Cost-sensitive learning for credit risk

Jorge C. Rella

Doctoral Thesis UDC 2024

Advisors: Juan M. Vilar, Ricardo Cao

Doctoral Programme in Statistics and Operations Research

Cost-sensitive learning for credit risk

Jorge C. Rella

Doctoral Thesis UDC 2024

Advisors: Juan M. Vilar, Ricardo Cao

Doctoral Programme in Statistics and Operations Research

Department of Mathematics

Faculty of Computer Science

The undersigned, Juan Manuel Vilar Fernández and Ricardo Cao Abad, certify that they are the advisors of the Doctoral Thesis with International Doctorate Mention and Industrial Doctorate Mention entitled "Cost-sensitive learning for credit risk", developed by Jorge Castiñeiras Rella at the University of A Coruña (Department of Mathematics), as part of the interuniversity doctoral program (UDC, USC and UVigo) of Statistics and Operational Research, and hereby give the author authorization to proceed with the presentation of the thesis and its subsequent defense.

A Coruña, 30th May 2024.

Advisors:

Juan Manuel Vilar Fernández

Ricardo Cao Abad

PhD candidate:

Jorge Castiñeiras Rella

"I saw the angel in the marble
and carved until I set him free."

Michelangelo Buonarroti

Agradecimientos

En primer lugar, gracias a mí. Gracias por atreverte con este reto, por mover cielo y tierra hasta conseguir lo que te habías propuesto y por todo el trabajo y dedicación que has puesto para dar forma a esta tesis y a la persona en la que te has convertido estos últimos 3 años. Ha sido una montaña rusa, pero ha valido la pena por todo lo que he crecido, aprendido y disfrutado.

Dicho lo anterior, Juan, Ricardo, evidentemente este trabajo no habría sido posible sin vosotros. Gracias por todo lo que me habéis enseñado en todos los sentidos y por contagiarme vuestra pasión por la investigación. Ha sido un placer compartir este camino con vosotros y un orgullo haber crecido de la mano de dos personas a las que admiro enormemente. Gracias por la confianza que tuvisteis en este proyecto, que por momentos parecía una locura, así como toda la dedicación para que fuese un éxito. También quiero destacar el papel de Ricardo en hacerlo posible. Busqué todo tipo de alternativas y tú fuiste el único que sacó tiempo para buscar soluciones para hacer un doctorado. De no ser por ti, nada de esto existiría.

Gracias también a ASF por creer en este proyecto, en especial a Gonzalo por siempre apostar y confiar en mí. Has sido como un padre estos años, siempre preocupándote en que crezca como profesional y como persona.

Gracias al programa de doctorado por todo el soporte que nos da a los investigadores. Y gracias al tribunal de seguimiento (Salva, Rosa y Maca) por la dedicación y feedback que me habéis dado. Sin duda, ha sido una gran ayuda para la confección de esta tesis.

Thank you, Gerda, for inviting me to work with you in Leuven, for taking care of me (and my luggage), and for making my first experience of living abroad feel like I was still in the city where I grew up. I take with me many good moments, a chapter of this thesis and a second home thanks to you.

Gracias David por invitarme a colaborar contigo en Londres. Desde el primer momento noté que eras de esas personas que motiva y estimula a los que le rodean, pero conmigo lo has hecho a otro nivel. Gracias por todo lo que has aportado tanto a esta tesis como a mí, por la inspiración a siempre pensar fuera de la caja y por demostrarme que la suerte sonríe a la gente que trabaja y se atreve. Gracias también por la ópera, las cañas y todas las cosas que todavía tenemos pendientes.

Tengo que agradecer también a toda la gente que me inspiró a comenzar este camino de la tesis. Siempre me ha gustado aprender y darle vueltas a las cosas, pero nunca pensé que esto fuese para mí. Recuero hablar muchas veces con Ángel que nunca nos quedaríamos en la universidad. Es gracioso a veces como son las cosas. Poco después, Ángel era quien me motivaba a empezar con la tesis y ahora yo estoy presentando la mía, siempre inspirado por él. También gracias a Rosa y Belén, que me inspiraron durante mi TFM, dándome una primera muestra de cómo era el mundo de la investigación y me motivaron a seguir en él. Tengo la suerte y orgullo de consideraros dos referentes. Gracias Vicente por enseñarme todo lo que se puede impactar con la estadística y por inspirarme a buscar siempre algo más.

Hay muchísima gente que de una forma u otra han aportado también a esta tesis. Gracias a toda la gente que me ha mantenido cuerdo, con los pies en el suelo y que ha sido una fuente de inspiración y desconexión siempre que la he necesitado. Sin vosotros, esto tampoco habría sido posible. Perdón también a la gente, aficiones y todo a lo que le he robado tiempo y atención por dedicársela a esto. Espero que esta obra sirva de disculpa.

Gracias a todos mis compañeros y amigos del CITIC, por todos los cafés, comidas y charlas en las que siempre podía disfrutar un buen momento o aprender algo nuevo. En especial, gracias a Ángel, Rebeca, Alberto y Bea, que aparte de ser grandes amigos, han puesto también su granito de arena en la confección de esta tesis.

Gracias a toda la gente que he conocido a lo largo de este camino. En el trabajo, estancias y congresos he tenido la suerte de cruzarme con gente maravillosa que ha hecho que disfrute cada segundo. Me alegra saber que no solo me llevo una tesis, si

no un círculo que no ha hecho más que expandirse con gente increíble.

Por supuesto, gracias al círculo que traía de antes, que siempre me apoyó y se alegró por mis alegrías. Gracias a TN, a los Despojos y a los CHAVALES. Gracias Nerea, Andrea, Adrián, Raquel, Nora, Román y Nacho. Gracias a todas las personas que considero familia y sin las que no me imagino mi vida bajo ninguna circunstancia. Sin vosotros, este camino no habría sido tan divertido y enriquecedor. Y sobre todo, no tendría sentido sin vosotros para celebrarlo.

Y por último, gracias familia por darme siempre todo el apoyo y cariño del mundo para impulsarme a conseguir lo que me proponga. Con un respaldo así, es difícil tener límites. Gracias papá por inculcarme el amor a las matemáticas y por demostrarme que si disfrutas de lo que haces, nunca tendrás que trabajar. Gracias mamá por enseñarme a afrontar todo de forma positiva y a confiar en mí mismo (aquí creo te pasaste de hecho). Gracias Iván por ser esa persona en la que sé que siempre podré confiar. Y gracias por motivarme a ser mejor para ser un buen ejemplo para ti. Espero que esta tesis te motive también a encontrar tu pasión. Gracias abuela por recordarme siempre que estoy en los huesos, así no me relajo en los otros aspectos de mi vida. Y por supuesto, gracias por hacerme rosquillas siempre que voy a casa.

Los logros carecen de sentido si no tienes con quien celebrarlos, y yo tengo la suerte de verme rodeado de caras felices por mis alegrías. Tengo amigos en todo el mundo, haciendo cosas increíbles, siendo una inspiración y motivación constante así como una fuente de felicidad. No podría ser más afortunado de contar con un círculo así de talentoso, especial y sublime, que me apoya incondicionalmente y que quiere que me vaya bien en todo lo que haga, incluidas locuras como esta tesis.

A pesar de que no soy la persona más expresiva ni cariñosa del mundo, a todos vosotros, mil gracias desde el fondo de mi corazón. Esta tesis la he hecho porque era lo que quería y me llenaba, pero por favor, vedla también como mi forma de daros las gracias y demostrar que todo lo que me habéis dado ha tenido sus frutos.

Funding

The PhD candidate's research was supported by the Axencia Galega de Innovación Industrial PhD Grant 14-IN606D-2021-2607768. The work has been partially carried out during two stays, one at KU Leuven (Belgium), which was supported by the INDITEX-UDC mobility grant 04.00.47.00.01 422D 48001, and another at DataSpartan (England), which was supported by the Axencia Galega de Innovación Industrial PhD Grant 14-IN606D-2021-2607768.

This research has also been partially financed by the Grant PID2020-113578RB-I00, funded by MCIN/AEI/10.13039/501100011033/. It has also been supported by the Xunta de Galicia (Grupos de Referencia Competitiva ED431C-2020/14) and by CITIC that is supported by Xunta de Galicia, convenio de colaboración entre la Consellería de Cultura, Educación, Formación Profesional e Universidades y las universidades gallegas para el refuerzo de los centros de investigación del Sistema Universitario de Galicia (CIGUS).

Consent for publication

Abanca Servicios Financieros consents to the publication of this work after verifying that no type of confidential or sensitive information related to the company or its clients is provided.

Abstract

This thesis addresses the problem of fraud detection and credit risk from a cost-sensitive perspective, exploring techniques that maximize the benefits to a financial institution while minimizing the probability of loss. First, an algorithm is proposed to estimate the optimal decision given a score. Once the optimal decision rule problem is solved, the estimation of the score is addressed from a cost-sensitive perspective. A parametric model is proposed to estimate the score, and its consistency and asymptotic normality are obtained. A cost-sensitive semi-parametric model is also proposed, which is more robust and flexible. Finally, credit risk is approached from a reinforcement learning perspective. An online learning algorithm and a bandit algorithm are proposed to obtain updated models with the latest available information, optimizing decisions from a cost-sensitive perspective. The good performance of all proposals is demonstrated through extensive simulation studies and the analysis of various real credit risk data sets.

Resumen

Esta tesis aborda el problema de detección de fraude y riesgo de crédito desde una perspectiva sensible al coste. Se investigan técnicas que maximicen los beneficios de una entidad financiera al mismo tiempo que se minimiza la probabilidad de incurrir en pérdidas. En primer lugar se propone un algoritmo para la estimación de la decisión óptima dada una puntuación. Una vez obtenida la regla de decisión óptima, se aborda la estimación de la puntuación desde un enfoque sensible al coste. Se propone un modelo paramétrico sensible al coste, para el cual se obtiene su consistencia y normalidad asintótica. También se propone un modelo semi paramétrico sensible al coste, el cual es más robusto y flexible. Finalmente se afronta el riesgo de crédito desde una perspectiva de aprendizaje por refuerzo. En particular, se proponen un algoritmo de aprendizaje en línea y un algoritmo de bandido, de forma que se obtienen modelos actualizados con la última información disponible, optimizando las decisiones desde una perspectiva sensible al coste. El buen funcionamiento de todas las propuestas es demostrado a través de estudios de simulación y el análisis de varios conjuntos de datos reales de riesgo de crédito.

Resumo

Esta tese aborda o problema da detección do fraude e do risco de crédito desde unha perspectiva sensible ao custo. Investíganse técnicas que maximizan os beneficios dunha entidade financeira e minimizan a probabilidade de sufrir perdas. En primeiro lugar, proponse un algoritmo para estimar a decisión óptima dada unha puntuación. Unha vez resolto o problema da obtención da regra de decisión óptima, a estimación da puntuación afróntase desde un enfoque sensible ao custo. Proponse un modelo paramétrico sensible ao custo, para o que se obtén a súa consistencia e normalidade asintótica. Tamén se propón un modelo semiparamétrico sensible ao custo, que é máis robusto e flexible. Finalmente, o risco de crédito abórdase desde unha perspectiva de aprendizaxe por reforzo. En particular, proponse un algoritmo de aprendizaxe en liña e un algoritmo de bandido, de xeito que se obteñan modelos actualizados coa última información dispoñible, optimizando as decisións desde unha perspectiva sensible ao custo. O bo rendemento de todas as propostas demóstrase mediante estudos de simulación e a análise de varios conxuntos de datos reais de risco de crédito.

Contents

Introduction

This industrial thesis has been carried out in collaboration with *Abanca Servicios Financieros* (ASF), a Spanish consumer finance company, as part of an industrial thesis. Thus, all the problems addressed are motivated by real problems of ASF, as well as the proposed techniques. In an industrial thesis, the objective is to research and provide innovative solutions adapted to the needs of the company, which are not currently found in the state-of-the-art. Nevertheless, all the proposed techniques can be generalized to any problem, as indicated throughout the thesis.

As the global economy constantly evolves, financial companies have had to expand their product catalog to meet new demands from customers and points of sale. One example is financing for online businesses, which has required new processes for analyzing and approving lines of credit. While these changes offer lucrative opportunities for increased capital acquisition and profit growth, they also present new challenges for assessing, mitigating, and managing the risks associated with lending activities. One prominent example is the increased risk of fraud in e-commerce, where threats such as document falsification and identity theft are much more difficult to prevent. In addition, banks have had to streamline their processes to meet the growing demand for faster application responses and higher transaction volumes.

The increase in the volume of applications, the new risks faced and the need for faster response times have led companies to rely more heavily on automated decision-making tools. This industrial thesis delves into this area of credit risk management, focusing on practical strategies, methodologies, and technological advancements aimed at maximizing profits while strengthening risk management. Thereby

ensuring the stability and growth of financial companies.

Credit can be defined as a financial agreement of a deferred payment extended by a creditor, also known as a lender, to a debtor, also known as a borrower, based on trust. The creditor extends money or resources to the debtor with the agreement that the debtor will repay the equivalent at a later date, creating a debt that the borrower is obligated to fulfill. The resources provided may be financial (e.g., a loan) or may consist of goods or services (e.g., consumer credit). In exchange for providing the loan and assuming the risk that the debt will not be repaid, the lender receives a commission on the capital borrowed. This is the lender's reward for assuming the credit risk in exchange for providing the loan. Consumer finance companies such as ASF provide credit to individuals for the purchase of goods such as a car, treatment, or appliance (see Committee (2001)). This allows points of sale to operate without having to finance their customers, with all the capital and regulatory implications that this entails.

Credit risk measures the probability of loss incurred by a financial institution as a result of a borrower failing to repay a loan in accordance with the terms of the loan agreement. This implies that a lender may not receive the principal and interest owed, resulting in an interruption of cash flows, allocation of additional provisions and increased capital recovery costs after the default is produced. This can directly affect the company's profitability and financial health, as well as its reputation from the perspective of both investors and the regulator.

The 2007 financial crisis is a stark example of the importance of robust risk management in the financial sector. Subprime mortgages granted in the US to borrowers with insufficient solvency, triggered widespread defaults and liquidity crises, eventually leading to a severe recession. This applies not only to financial institutions and banks, but also extends to companies, markets, and organizations in various sectors. As a result, financial firms are subject to a very strict regulatory framework that imposes capital requirements, maximum risk thresholds, and responsible lending policies, as summarized in Committee (2001).

Although it's impossible to know exactly who will default on obligations, properly assessing and managing credit risk can lessen the probability and the severity of a loss. Banks mitigate this risk by requiring guarantees or imposing additional clauses based on a client's risk profile, such as higher interest rates or credit limits. This involves assessing the applicant's creditworthiness, specifically their solvency and payment capacity, to determine their ability to fulfill their financial obligations under a loan agreement.

The five Cs of credit are composed by capacity, capital, conditions, character, and collateral. These are the factors that lenders can analyze about a borrower to help predict the likelihood that the applicant will default on a loan and reduce credit risk. By analyzing past approved applications, subsequent default patterns can be observed, enabling the construction of structured datasets. These datasets include customer and loan characteristics collected at the application stage, focusing on factors relevant to credit risk assessment. In addition, customer behavior is observed for approved loans, enabling the application of classification techniques to build defaulter profiles to prevent future losses.

Modern approaches often focus on individual-level factors and their historical patterns, assuming that poor individual decisions contribute significantly to default. This implies that certain individual characteristics can be used to explain why some borrowers are more likely to default than others. Typical information collected at the time of application includes socio-demographic information (e.g., age and occupation), financial information (e.g., income and number of loans), and information about the borrower's credit history (e.g., missed payments and prior defaults), which is often obtained from a credit bureau, as discussed in Petrides et al. (2022).

In general, lenders analyze a potential creditor's capacity, or the income they have in relation to the amount of debt they are requesting. Most lenders employ models (credit scorecards) to rank potential and existing customers according to risk and extend credit (or not) accordingly. They may use in-house programs as well as information provided by third parties, such as external rating agencies for PD estimation, for a fee. Due to the role of accurate credit risk assessment in enabling

financial institutions to effectively manage risk and optimize profitability, credit scoring has long attracted significant attention from both academia and industry.

Consumer finance involves the management of two risk factors: the legitimacy of the loan application (fraud risk) and the risk that the borrower will default on the loan (default risk). Both components create the potential for lenders to incur losses and impact their overall profitability. Credit default occurs when a customer fails to meet its contractual payment obligations for a predetermined period of time. Typically, default is defined as 90 days of payment delay, as specified in Petrides et al. (2022) and Siddiqi (2006). Fraud, on the other hand, is a more severe risk, defined as a credit application with no intent to pay, leading to the total loss of the financed credit.

The main difference between default and fraud is that the latter corresponds to an operation in which the client has no intention of repaying the loan, usually involving falsification of information, identity theft and other types of fraudulent strategies. For its part, default, corresponds to operations in which the client intends to repay the loan at the time of application, but due to a poor credit profile or a subsequent deterioration, it makes the client unable to meet the formalized debt obligations. Thus, fraud is observed from the first receipt issued, since the client does not pay from the first moment. On the other hand, the default is observed after a period of time, even years, after the formalization of the loan.

By definition and concept, the risk of fraud and the risk of default are two different threats. Therefore, financial institutions typically manage credit risk through a sequential two-step evaluation process to mitigates these two risks. This two-step approach effectively manages credit risk by prioritizing fraud prevention and ensuring borrower solvency for responsible lending and minimizing potential losses.

The first evaluation stage focuses on fraud detection, which is the most dangerous risk because of the potential for complete loss of capital if fraudulent operations go undetected. In addition, fraud is a growing risk for financial institutions as noted in Lucas and Jurgovsky (2020) and Yusuf and Ekrem (2011). Fraud detection is an

extremely challenging problem because fraudsters adapt to risk policies, rendering models trained on historical information useless in detecting new fraudulent customer profiles. Also, fraudsters usually change their information or commit identity theft, leading to class overlap as claimed in King and Zeng (2002). This makes it difficult to identify patterns when training supervised models. Furthermore, fraudulent cases are rare, leading to an extraordinarily imbalanced problem as observed in Almhaithawi et al. (2020), which makes the task of fraud detection even more difficult. Finally, regulatory constraints limit the use of complex models, such as neural networks, because all decisions made must be explainable to the regulator and the customer, as noted in Altman et al. (1994); Lucas and Jurgovsky (2020) and Min and Lee (2008).

Given these difficulties, financial institutions typically mitigate the risk of fraud by applying certain admission filters based on expert judgment and past fraud. This stage uses various tools to assess the legitimacy of an application, including the study of historical data, risk flags based on previous fraud profiles, and document verification. While this methodology detects the most obvious and historically recurring frauds, it falls short in evolving scenarios such as consumer finance. Only applications that are deemed legitimate after this fraud screening process proceed to the next stage of solvency assessment.

Once it has been verified that the requested loan is a legitimate operation, the second step is to assess the applicant's ability to repay the loan, taking into account several factors such as the client's socio-economic characteristics and the characteristics of the loan. When evaluating loan applications, a critical factor influencing the decision on whether or not to approve the proposed transaction is the estimated probability of default (PD). This metric represents the likelihood that a borrower will default on the loan. Based on this evaluation, a credit decision is made after considering both the potential benefits and associated risks of the loan request. If the client profile and loan request meet the company's risk criteria, the transaction is approved and the loan is extended.

In credit risk problems, it is extremely costly to manually evaluate each borrower.

Therefore, there have been many attempts to develop automated credit scoring models, known as *scorecards*, which predict the fraud/default probabilities given individual attributes. Statistically, credit scoring is concerned with the development of predictive models to support decision making, as introduced in Crook et al. (2007); Lessmann et al. (2015) and Petrides et al. (2022). Thus, credit scoring is a supervised learning task designed to distinguish between good and bad credit applicants based on their known characteristics, as defined in Thomas et al. (2002) and Verbraken et al. (2014). A "bad" label indicates a high risk of default, meaning that the borrower may not fully repay the loan. The industry standard for banks today is considered to be the logistic regression model, even though several advanced models outperform it, as noted in Lessmann et al. (2015) and Visantavarakul (2022). This is mainly due to the interpretability of the logistic model, something that is sought by financial firms and required by regulators. The logistic model for credit scoring has been studied by Thomas (2000); Siddiqi (2006) and Samreen et al. (2013), among others.

The primary objective in developing credit scorecards is typically to accurately risk-rank loan applicants, allowing to set a minimum creditworthiness level below which applicants are rejected and above which applicants are accepted (see Petrides et al. (2022)). Beyond simply granting or denying loans, scorecards can also inform risk-sensitive pricing strategies, tailoring loan terms (interest rates and fees) based on the borrower's risk profile. This enables more nuanced and profitable lending decisions. In addition, for Basel-compliant financial institutions, credit scorecards and PD estimation play a critical role for regulatory capital requirements as outlined by Bequé et al. (2017).

A wide variety of classification methods have been proposed and applied to credit scoring in the literature. Comprehensive experimental benchmarking studies have recently been conducted by Lessmann et al. (2015) and Vanderschueren et al. (2022b) as an update to an earlier study in Baesens et al. (2003). They compare the performance of several classification models on eight and four credit scoring datasets, respectively, and show that the industry standard, logistic regression, is significantly

outperformed by more advanced methods. In addition, in Lessmann et al. (2015) it is highlighted a lack of real-world financial data available for academic research on credit scoring, and that existing datasets are relatively small, averaging only $6,167$ observations. This limits the scope and generalizability of research findings. In addition, publicly available datasets often lack details about the costs and benefits associated with credit, information that is essential for conducting cost-based evaluations of scoring models, a key objective in this area. These limitations, primarily due to financial institutions' concerns about privacy and competitive advantage, make large-scale benchmark studies with diverse datasets nearly impossible, as also noted in Petrides et al. (2022) and Visantavarakul (2022). As a result, the literature, while extensive, lacks state-of-the-art references due to the lack of publicly available datasets and the resulting comparisons.

There is another drawback in the credit scoring literature. Classification models are typically trained in terms of statistical performance measures that do not take into account the actual business objective, which is to maximize benefits while minimizing financial losses. For example, it would be better to detect one $10,000€$ fraud than five $1,000€$ frauds, although the accuracy will be lower. Several authors have pointed out that decisions based only on an estimated probability have a worse performance in problems where not all error types have the same impact (Lucas and Jurgovsky (2020); Höppner et al. (2021); Bahnsen et al. (2013, 2014a); Elkan (2001); Nikolaou et al. (2016); Vanderschueren et al. (2022b); Verbraken et al. (2014); Wang et al. (2012)). Consequently, credit risk should be acknowledged as an *instance-dependent cost-sensitive* problem as claimed in Höppner et al. (2021). This introduces an additional degree of complexity, since the costs depend not only on the class, but also on the characteristics of the instance (the loan amount).

Cost-sensitive (CS) methods address problems where different misclassification errors imply different impacts. These cost-aware techniques help decision makers prioritize and optimize decisions based on the specific costs associated with different classification errors, maximizing benefits while preventing susceptible losses.

In CS learning, models are trained by considering different costs for false positives

(incorrectly identifying an event) and false negatives (missing an event) as defined in Elkan (2001). This setting is observed in all sort of real-world problems. In fraud detection, it is much more expensive to miss a fraudulent case than to falsely flag a legitimate transaction (Lessmann et al. (2015); Höppner et al. (2021)). Similarly, in customer churn prediction, misclassifying a loyal customer is less costly than missing a churner (Verbeke et al. (2012); Verbraken et al. (2012)). Credit scoring also faces this challenge, as misclassifying good and bad loan applicants leads to different costs: rejecting a good applicant means lost revenue, while accepting a bad applicant can lead to significant financial losses, as noted in Verbraken et al. (2014). Thus, as stated in the Technological Roadmap of the MLnet II project (European Network of Excellence in Machine Learning), incorporating cost into learning has been identified as one of the most relevant topics for future machine learning research (Saitta and Geibel (2001); Zhou and Liu (2010)).

Among the state-of-the-art CS approaches, there are two distinct philosophies, as introduced in Vanderschueren et al. (2022b). The first, known as *predict-and-optimize*, consists of constructing a classifier with a cost-sensitive perspective, tuning the estimated probabilities as proposed in King and Zeng (2002); Yih et al. (2006) and Vanderschueren et al. (2022a), or using weighted versions of logistic regression (Bahnsen et al. (2014a); Höppner et al. (2021)), boosting algorithms (Nikolaou et al. (2016); Höppner et al. (2021)), or neural networks (Shu et al. (2023)), among others. The second approach, known as *predict-then-optimize* or *thresholding*, focuses on the decision phase. A predictive model is trained with the goal of maximizing accuracy, and then the decision is optimized by minimizing losses, as proposed in Almhaithawi et al. (2020); Bahnsen et al. (2013); Elkan (2001); Höppner et al. (2021); Vanderschueren et al. (2022a,b) and Sheng and Ling (2006).

While traditional credit risk systems such as those mentioned above are effective, they have their limitations. Their reliance on historical data from pre-selected customers results in a lack of information about potential borrowers who were previously deemed ineligible for credit. This can lead to customers being unfairly misjudged and missed profit opportunities from new customer profiles. Similarly, static models

are subject to inherent bias and the risk of concept drift due to their reliance on historical data.

Credit risk is a dynamic problem, as customer behavior changes with the economic cycle, market trends and other factors. As a result, models must be continually updated as new information becomes available. This is known as reinforcement learning. This approach avoids concept drift and bias from historical data, and keeps models up to date in changing scenarios.

Likewise, in order to collect enough data for model fitting, lenders have to face credit risk and incur financial losses by collecting the labels of defaulting borrowers. If a lender tries to avoid such an exposure by collecting only a few observations, the estimated model would have a high variance problem. Conversely, if a lender collects too many observations, the improvement in the performance may not justify the financial losses of lending to defaulting borrowers. This leads to an exploration vs. exploitation dilemma as noted in Sutton and Barto (2018): should a new client be employed to gain more information and improve predictions at the potential cost of incurring losses, or should the current model's assessment dictate the decision? As a result, financial institutions face two challenges that are often overlooked: adapting to a dynamic environment while optimizing both immediate risk and long-term model performance. Therefore, agents seeking optimal credit risk management must adapt to dynamic environments while addressing the exploration-exploitation dilemma, optimizing both immediate risks and long-term model performance.

Online learning (OL) copes with dynamic environments, continuously updating models as new observations become available. This enables timely and context-aware decisions, as highlighted in Lattimore and Szepesvári (2020); Sutton and Barto (2018) and Hoi et al. (2018). OL algorithms continuously refine their predictions by sequentially observing data, making a decision, receiving feedback on those decisions, and adapting their prediction mechanisms with the newly obtained information. OL embraces a continuous learning-while-doing approach, making it the preferred choice for scenarios where data arrives continuously and adaptation is paramount, such as the credit risk problem or online advertising.

Online learning algorithms deal with dynamic environments but do not consider exploration. Within the reinforcement learning paradigm, bandit algorithms are designed to balance the process of seeking new information (exploration) with maximizing rewards based on existing knowledge (exploitation) in sequential decision making under dynamic conditions. Bandit algorithms receive their name from a very illustrative example. Imagine a gambler (agent) staring at a row of one-armed bandits (actions), each promising unknown payoffs. The agent is faced with the dilemma of pulling the familiar lever that's currently spitting out coins (exploitation) or trying a new one for potentially bigger winnings (exploration). In the stochastic setting, each source (arm) is associated with a distribution generating random rewards as described in Achab et al. (2018). The goal of the agent is to learn the reward distribution of each arm and find a strategy in order to maximize the cumulative reward over time. One option would be to pull each available arm once and then exploit the arm with the highest reward. However, this could be just a lucky fluke. On the other hand, if the agent decides to keep exploring, it may lose out on the profits from the optimal arm. Bandit algorithms aim to solve this exploitation-exploration trade-off.

As highlighted above, the various effects of misclassification errors significantly affect the results in cost-sensitive problems such as credit risk. Consequently, CS reinforcement learning algorithms have been recently proposed and investigated. These algorithms incorporate error costs into their learning process, allowing them to prioritize updates that minimize the overall cost of misclassification. This adaptation allows models to make cost-sensitive decisions in real time, especially when dealing with constantly evolving data streams. The drawback of these approaches is that they are not context-aware and/or consider a fixed impact for each decision error, as claimed in Zhang et al. (2018) and Achab et al. (2018). While this may be reasonable in some scenarios, such as online advertising, it is a restrictive constraint in instance-dependent problems, such as credit risk, and may lead to suboptimal results as mentioned in Bahnsen et al. (2014a); Höppner et al. (2021) and Vanderschueren et al. (2022b).

Given the aforementioned drawbacks of the state-of-the-art approaches and the lack of alternatives available to deal with cost-sensitive problems such as the credit risk problem presented by ASF, this thesis proposes new techniques to address the problem of maximizing benefits and reducing potential losses in credit risk. Therefore, we adopt a cost-sensitive perspective towards the development of credit scorecards and decision-making techniques. Thus, Chapter 1 is devoted to introducing the reader in detail to the cost-sensitive problem to be addressed throughout this thesis.

We consider the two philosophies for cost-sensitive classification (thresholding and predict-and-optimize as introduced in Vanderschueren et al. (2022b)) so that we provide solutions from both perspectives.

In Chapter 2, a new cost-sensitive thresholding algorithm is introduced to estimate the optimal decision rule over a data set given a score, the loan amount, and a cost specification. A novel decision space is constructed using the variables on which losses depend: an estimated fraud/default probability and the loan amount. In this expanded space, the optimal decision rule is searched with a new proposed algorithm with complete freedom. The algorithm search incorporates and extends all previous thresholding approaches, thus providing a consistent improvement. Furthermore, the algorithm allows the search for constrained decision rules, something that no previous approach can solve as noted in Vanderschueren et al. (2022a) and Verbraken et al. (2014).

Chapter 3 extends the proposed thresholding algorithm to the multi-class setting. With this algorithm, the best possible decision rule is empirically obtained for multi-class responses given a single score, a previously unexplored challenge. In this way, the optimal decision rule estimation given a score is solved for any CS problem. Thus, all that remains is to focus on the estimation of the score.

For the score estimation, we use both a parametric and a semiparametric approach. In Chapter 4, cost-sensitive parameter estimators for parametric classifiers are proposed and their consistency and asymptotic normality under general condi-

tions are obtained. The theoretical results, combined with an extensive performance study, provide a theoretical basis and practical justification for CS parameter estimation in binary classification tasks.

Chapter 5 proposes cost-sensitive semi-parametric classifiers extending the single-index model proposed in Klein and Spady (1993) to estimate the score. For this, the classifier is estimated in a two-step iterative process minimizing the expected losses. Thus, the model is specifically trained to optimize cost-sensitive decisions. This, combined with the flexibility and robustness provided by SIMs, leads to proficient results.

With these proposals, the problem of estimating the score and obtaining the best possible decision is solved from a static perspective. However, the credit risk problem is by definition dynamic. This problem has not yet been solved from the perspective of reinforcement learning, so it is considered interesting to study the improvements that can be obtained in credit risk considering a dynamic approach.

In Chapter 6 different techniques are proposed to address the cost-sensitive classification problem from a reinforcement learning perspective. Proficient models are obtained by combining the improvement obtained by considering a CS approach with the adaptability provided by the reinforcement learning framework to adapt to changing scenarios. A cost-sensitive online learning algorithm is proposed that extends the passive-aggressive (PA) algorithm introduced in Crammer et al. (2006) by considering a loss function for model updating. In addition, a CS version of the logistic bandit approach is proposed considering an instance-dependent reward function for model fitting and exploration.

The CS bandit model is expected to perform better than the OL algorithm because the former takes exploration into account. However, there are entities for which the strategy of not sticking to the optimal decision given by the current model is too aggressive. Therefore, we want to provide a more conservative CS alternative with an OL algorithm. Thus, the problem of score estimation is solved from all possible perspectives.

To empirically demonstrate the good performance of the proposed techniques throughout this work, we conduct an extensive set of experiments on some benchmark datasets from different real-world application domains and several simulated scenarios. Promising experimental results demonstrate the effectiveness and efficiency of the proposed models, especially when compared to the state-of-the-art. Furthermore, these studies provide valuable insights into the algorithmic performance in terms of case proportions, dependency relationships, and cost specifications in both simulated and real scenarios for the proposals in this work and state-of-the-art approaches. In this way, we also fill the gap regarding the lack of comparative studies claimed in Petrides et al. (2022) and extend previous results in Lessmann et al. (2015); Höppner et al. (2021); Baesens et al. (2003) and Vanderschueren et al. (2022b). Finally, Chapter 7 gives some concluding remarks and hints for future research.

With the proposed techniques, the initial objective is achieved, which is to optimize the admission and risk management flows of our collaborating entity, ASF. Nevertheless, all the proposals can be extended to any cost-sensitive problem. The chapters have been written in such a way that they can be read individually, allowing to understand the developments in the different chapters without having to read the others. The main chapters that make up this work have been published in international scientific journals or are currently under review, so the reader may wish to explore them independently.

To implement the methods proposed in this thesis, several functions programmed in the open-source statistical software R (R Core Team (2021)) have been developed by us and are available in **github.com/C-Rella**.

Chapter 1

Cost-sensitive classification

In the field of artificial intelligence and data science, classification is one of the most important machine learning tasks. Classification is a supervised learning task that involves the categorization of input data into predefined classes or categories based on their intrinsic features and attributes. Classification tasks are found in a wide range of challenges in different domains and applications. Examples of classification tasks include spam detection, image classification, medical diagnosis, sentiment analysis, and fraud detection, among others. Classification techniques in machine learning play a critical role in extracting actionable insights from complex data sets and facilitating informed decision making.

Binary classification addresses the problem of predicting a binary dependent variable $Y \in \{0, 1\}$ from a set of independent covariates $\mathbf{X} = (X_1, \ldots, X_d)$. For this, a model provides a continuous score, $s(\mathbf{x})$, which is used for decision making. For the rest of this work, let $s(\mathbf{x}) = E[Y \mid \mathbf{X} = \mathbf{x}] = P(Y = 1 \mid \mathbf{X} = \mathbf{x})$, i.e. the optimal theoretical predictor (in the sense of minimizing the mean square prediction error) of the class label, Y, given the covariate vector $\mathbf{X}$. The assigned classification is determined by setting a cutoff value, h, which determines the predicted class as $\hat{Y} = \mathbb{I}[s(\mathbf{x}) > h]$.

Within the classification paradigm, *cost-sensitive* (CS) classification addresses

the prediction of a binary dependent variable $Y \in \{0, 1\}$ from a set of independent covariates $\mathbf{X} = (X_1, \ldots, X_d)$, taking into account the costs of prediction error and potentially other costs, as introduced in Elkan (2001). In practice, these costs typically depend on an exogenous variable, W, such as the monetary amount in financial settings or life expectancy in medical treatments.

An example of a CS classification problem is represented in Figure 1.1, motivated by Shen et al. (2020). It shows the different estimation approaches in a CS setting, and how accuracy and loss reduction are two different objectives in settings where misclassification error costs are not constant. For a binary classification problem, cases are represented by red boxes and non-cases by green dots, with the corresponding misclassification costs superimposed. Two different classifiers are considered, labeling the instances at the right of the dotted line as cases. The left classifier achieves an accuracy of 91% and a loss of 6 units, while the right classifier reduces the total loss to 4 units, which is preferred in a cost-sensitive setting, even though it has a lower accuracy (82%).

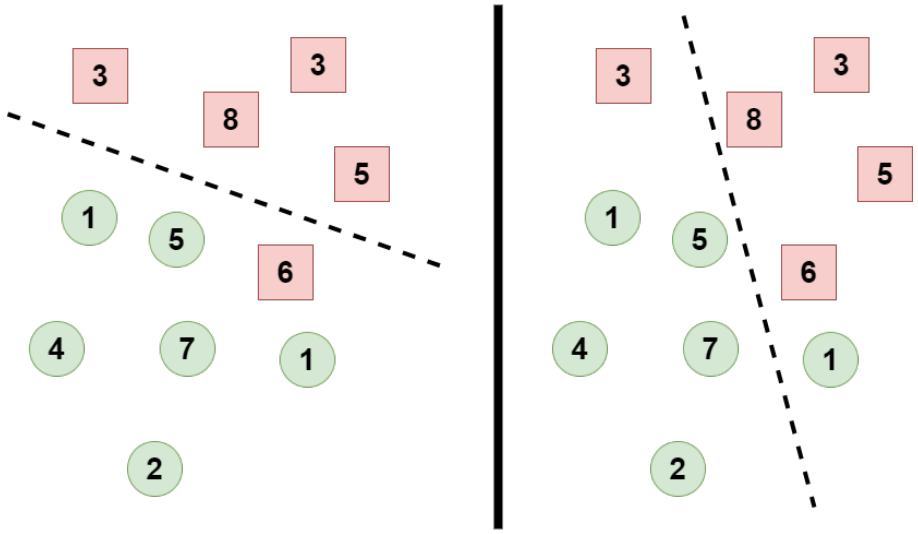

Figure 1.1: Two different classifiers over the same data set. Cases are represented in red boxes and non-cases in green dots, with each observation misclassification cost superimposed.

In CS settings, the objective is loss reduction (or benefit maximization). Thus, as highlighted in Figure 1.1, model performance must be evaluated considering the different costs of misclassification error in order to obtain reasonable results. The

first cost-sensitive classification techniques consider a cost matrix based on the true class Y and the predicted class $\hat{Y}$. These approaches assume that misclassification error costs are class-dependent, as in Elkan (2001) and Nikolaou et al. (2016). Nevertheless, this means a hazardous overlook of information as illustrated in Figure 1.1 and as noted in Bahnsen et al. (2014a, 2013); Elkan (2001) and Zadrozny and Elkan (2001). For example, considering the average misclassification cost for the positive and negative classes in the example in Figure 1.1, the error cost would be 3.3 for the positive class and 5 for the negative class. Thus, the best classifier would be the left classifier with a total loss of 5 compared to a loss of 8.3 for the right classifier. In practice, however, as explained above, the right classifier is the best, because it has the smallest cumulative loss.

In CS classification, the impact of a prediction is determined by the exogenous variable W and the prediction $\hat{Y}$, which depends on the estimated probability, $\hat{s}(\mathbf{x})$, of the conditional probability, $P(Y = 1 \mid \mathbf{X} = \mathbf{x})$, and the thresholding strategy through $\hat{Y} = \mathbb{I}[\hat{s}(\mathbf{x}) > h]$.

To accurately capture the impact of misclassification costs, it is critical to use instance-dependent cost-sensitive (IDCS) metrics to measure model performance. To address this, throughout this work it is considered a *loss function* to evaluate a prediction, constructed from a cost matrix such as the one represented in Table 1.1. This is in line with the proposals in Bahnsen et al. (2013, 2014a); Elkan (2001); Höppner et al. (2021) and Verbraken et al. (2012).

	$\hat{y}_i = 0$	$\hat{y}_i = 1$
$y_i = 0$	$C_i^{TN} = 0$	$C_i^{FP} = aw_i + b$
$y_i = 1$	$C_i^{FN} = cw_i$	$C_i^{TP} = b$

Table 1.1: Amount-dependent cost matrix for the fraud detection problem.

The cost matrix in Table 1.1 is constructed for the fraud detection problem to make more understandable what we introduce in this chapter. However, it can be

extended to any problem by changing the various entries in the matrix. Throughout the work, other cost matrices will be considered in the different chapters, as required by the problem being treated. For example, in the case of default risk, the potential benefits of granting good operations are also included. We will even extend this matrix to the case of multi-class classification in Chapter 3.

To construct the cost matrix in Table 1.1, it is assumed that there is a fixed cost $b \in \mathbb{R}^+$ corresponding to the cost of the analysis that the organization must perform when faced with a fraud alert. Furthermore, if a transaction is flagged as fraudulent, it will be rejected. Consequently, the credit interest earned when a legitimate operation is rejected is no longer earned, so it is considered a loss. An average profit per operation of aW is assumed, with $a \in (0, 1)$ and W is the credit amount. Finally, it is assumed that an undetected fraud implies a loss of cW, where $c \in (0, 1]$. In Table 1.1, C_i^{FN} encloses the cost of an undetected fraud, i.e., labeling a fraudulent operation as legitimate ($\hat{y} = 0$). The lost benefit when classifying a legitimate client as a fraud ($\hat{y} = 1$) is summarized in C_i^{FP}. The fixed cost of investigating the transaction, b, is included in both C_i^{FP} and C_i^{TP}, i.e. whenever $\hat{y}_i = 1$. The impact of false negatives is assumed to be greater than of false positives ($c > a$), following the reasonableness condition introduced in Elkan (2001). Gains could be introduced, but they do not appear because there is only the possibility of loss when dealing with fraud. Lastly, costs are assumed to be independent of the covariate vector $\mathbf{X}$ in accordance with Zadrozny and Elkan (2001).

Despite the cost matrix in Table 1.1 is very intuitive, we need a metric to evaluate model performance that summarizes the results obtained. By aggregating all the costs in Table 1.1 into a function, the loss function is defined as:

$$\begin{aligned}\ell(\hat{y}_i, w_i, y_i) =& (1 - y_i)(1 - \hat{y}_i)C_i^{TN} + (1 - y_i)\hat{y}_i C_i^{FP} + y_i(1 - \hat{y}_i)C_i^{FN} + y_i\hat{y}_i C_i^{TP} \\ =& y_i(1 - \hat{y}_i)cw_i + (1 - y_i)\hat{y}_i(aw_i + b) + y_i\hat{y}_i b.\end{aligned} \tag{1.1}$$

This is the function that is intended to be optimized throughout the entire work. This metric captures the impact of a prediction made on an observation in terms of

cost/benefit. Consequently, the optimal prediction for an instance i will be the one that minimizes the loss function in equation (1.1).

An example of the different results obtained when considering the loss function (1.1) for model fitting or the classical maximum likelihood estimator is shown in Figure 1.2. The upper graph represents a cloud of points in $\mathbb{R}^2$, with cases represented in red, non-cases in green, and each observation's misclassification cost superimposed. The bottom graphs represent the score obtained with a linear classifier, $s(\mathbf{X};\theta) = \theta'\mathbf{X}$. For the left one (ML), θ is estimated by maximizing the likelihood function, obtaining $\theta = (-1,-1)$. For the right one (CS), with the objective of minimizing losses, $\theta = (-1,1)$ is estimated. Considering the decision rule $\hat{Y} = \mathbb{I}[s(\mathbf{X};\theta) > 0]$ (dashed lines), the ML classifier achieves an accuracy of 90% and a loss of 9 units. The CS classifier reduces the total loss to 5 units, which is preferred in a CS setting, although it has a lower accuracy (70%). It is concluded that in CS problems, maximizing accuracy and minimizing cost are two different objectives. Therefore, cost-sensitive metrics need to be considered for model fitting and evaluation in CS problems.

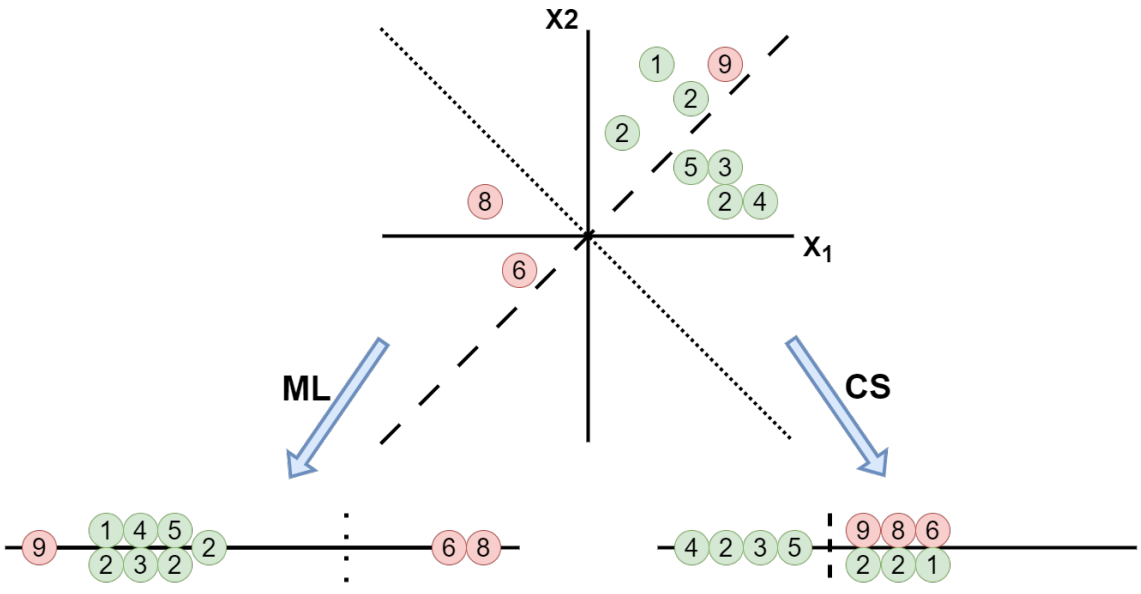

Figure 1.2: Two different linear classifiers fitted with the aim of maximize accuracy (ML) and minimize total losses (CS). Cases are represented in red and non-cases in green, with each observation misclassification cost superimposed.

The loss function in equation (1.1) is an individual metric. However, the goal is to reduce losses over the entire population. Consequently, the performance metric considered to evaluate model performance is *savings*, which is widely used in the literature (Almhaithawi et al. (2020); Bahnsen et al. (2014a, 2013)). It corresponds to the standardized sum of misclassification error costs committed over a data set. Thus, savings is a more appropriate metric for evaluating model performance because it considers results over the entire sample under study.

For a sample $(\hat{y}_i, w_i, y_i)_{i=1}^n$, savings are defined as:

$$\text{Savings} = 1 - \frac{\sum_{i=1}^n \ell(\hat{y}_i, w_i, y_i)}{\sum_{i=1}^n y_i w_i} \tag{1.2}$$

where ℓ is as defined in equation (1.1) and the denominator is the total loss assumed if no preventive action were taken, in order to have a base reference as suggested in Verbraken et al. (2014). This metric provides a standardized measure of prevented losses relative to the base case scenario, allowing for reasonable comparison between different models and problems. The objective in CS classification is to minimize $\sum_{i=1}^n \ell(\hat{y}_i, w_i, y_i)$, equivalently maximize (1.2), along possible classifiers. Note that considering a different cost specification in the cost matrix in Table 1.1, a different loss function (1.1) is obtained and consequently savings are also defined differently. Therefore, the introduced CS metrics can be adapted to any problem at hand.

To show the importance of considering a CS metric as the savings introduced in equation (1.2) in CS classification problems, a logistic model is fitted to the ASF dataset to be introduced in Section 2.5.1. Its PP $= \sum_{i=1}^n \hat{y}_i/n$, accuracy $= \sum_{i=1}^n I(\hat{y}_i = y_i)/n$, recall $= \sum_{i=1}^n I(\hat{y}_i = y_i = 1)/\sum_{i=1}^n y_i$, and savings (1.2) are represented in Figure 1.3 for different decision thresholds over the estimated fraud probability. The "Score" is referred as the escalation of the estimated fraud probability between 0 and 10, for the sake of confidentiality, and the metrics are represented as percentages. This notation is followed throughout the work. It can be seen the nonlinear relationship between the different metrics. In this way, detecting more frauds does not necessarily lead to an increase in savings (1.2) due to analysis costs. Also, the extreme class imbalance ($\bar{Y} \approx 0.6\%$) biases the accuracy, since the highest accuracy is achieved labeling all operations as legitimate. Consequently, the

fraud detection problem (as the credit risk problem) should be addressed from a CS perspective, both conceptually and in order to obtain better practical results.

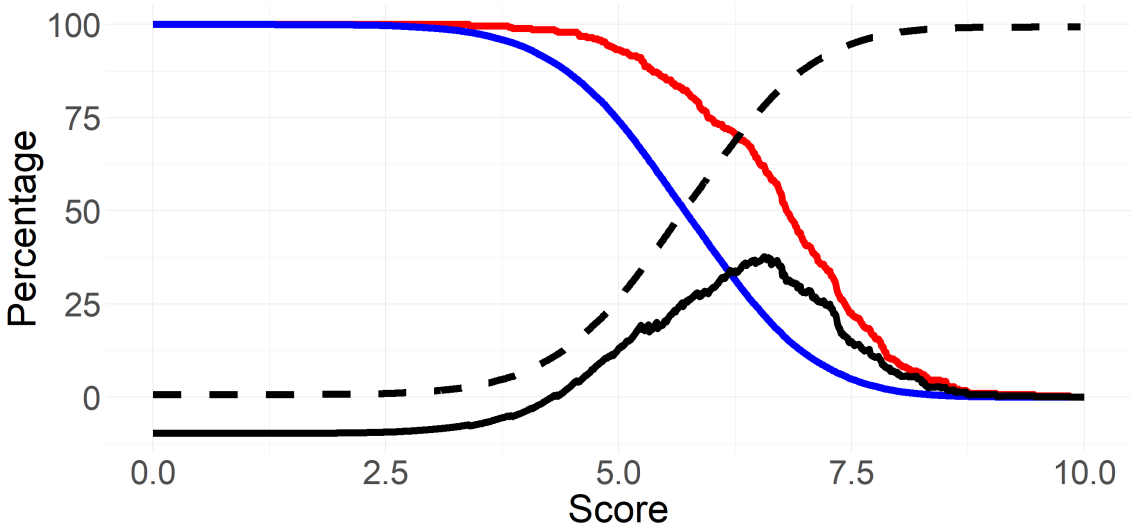

Figure 1.3: Accuracy (dashed black), recall (red), PP (blue) and savings (solid black) considering different cut-off decision points over the estimated score for the ASF data set.

It is worth noting that cost-sensitive approaches are valuable not only when misclassification errors incur different costs, but also when datasets exhibit class imbalance. This imbalance occurs when the number of observations in each class differs significantly, as is common in fraud detection and credit risk. In these domains, the positive class (representing "bads") is often much smaller than the negative class ("goods"). Such an imbalance can impede learners from building models that accurately predict the minority class. In such scenarios, simply classifying all instances as the majority class would yield high accuracy, but wouldn't provide valuable predictions, as shown in Figure 1.3. This can be corrected by constructing the cost matrix in Table 1.1 depending on the class imbalance of the data set.

Similarly, resampling techniques such as oversampling, undersampling, or SMOTE are often used to address cost-sensitive learning (Chawla et al. (2002); Almhaithawi et al. (2020); Dal Pozzolo et al. (2015); Elkan (2001); Lucas and Jurgovsky (2020)). However, while these techniques are appropriate for unbalanced settings, they imply bias when undersampling or an increase in variance when oversampling the data set, as claimed in He and A. Garcia (2009); Dal Pozzolo et al. (2015); Bahnsen et al.

(2014b) and Zhou and Liu (2006).

Regarding cost-sensitive proposals, there are two main methodologies as introduced in Vanderschueren et al. (2022b). The first one, known as *predict-then-optimize*, consists in building a predictive model maximizing accuracy. Then, the result of the model is used to optimize the CS decision making with some thresholding strategy. A drawback of this approach is that costs are only incorporated in the second stage. Conversely, the second approach, known as *predict-and-optimize*, proposes to train a classifier by minimizing a task-specific loss function that takes into account the various misclassification costs. Classifiers are constructed considering a cost-sensitive objective function by tuning the estimated probabilities (Bahnsen et al. (2014a); King and Zeng (2002); Yih et al. (2006); Vanderschueren et al. (2022a)) or using weighted versions of logistic regression (Bahnsen et al. (2014a); Höppner et al. (2021)), boosting algorithms (Nikolaou et al. (2016); Höppner et al. (2021)), or neural networks (Shu et al. (2023)), among others. Thus, the decisions are expected to be of superior quality, as the model is specifically trained to optimize cost-sensitive decisions in subsequent thresholding stages.

In the literature on cost-sensitive learning, which offers a plethora of methods, there is a lack of guidance as to which methods can be considered state-of-the-art. This is partly due to the lack of publicly available real-world datasets that would allow for a comprehensive comparison, as noted in Petrides et al. (2022). In Dmochowski et al. (2010), it is shown that the *predict-and-optimize* approach is more effective under model misspecification, and Vanderschueren et al. (2022b); Höppner et al. (2021) and Bahnsen et al. (2014a), among others, empirically demonstrate a better performance of classifiers when considering a CS thresholding strategy. Therefore, in this work, the CS classification problem is addressed from the two perspectives: thresholding and predict-and-optimize. Furthermore, extensive simulation studies and real data sets are evaluated to fill the gap regarding comparative studies for the performance of the different alternatives across different problem settings. Thus, providing valuable insights regarding the performance of the different approaches depending on the problem.

Chapter 2

Cost-sensitive thresholding over a two-dimensional decision region

2.1 Introduction

In this chapter it is addressed the fraud detection problem of minimizing losses given an estimated score. This was the first problem to be addressed with ASF because it was the most urgent and least developed problem in the company. In addition, the limitations of ASF meant that there was no state-of-the-art alternative to solve the problem. The company only had some filters to prevent previously detected fraud typologies and some alerting tools. The proposal presented in this chapter provides an adequate tool to optimize fraud detection, focusing on the operations with the highest risk and the highest probability of being fraudulent. As shown in the results section, a significant improvement is obtained compared to previous approaches, especially when compared to the company's strategy. Consequently, the proposal in this chapter has been introduced into the company's risk management workflow. In addition, the generalizability of the proposed algorithm, as well as the fact that it only requires an estimated score (ASF did not have to build a new model), has led ASF to apply the proposed algorithm in other areas to optimize decision making.

Fraud is a significant and growing risk for financial institutions as stated in Lucas and Jurgovsky (2020) and Yusuf and Ekrem (2011). Fraud is defined as an operation that intentionally results in the total loss of the financed loan. As a result, all transactions must undergo an initial fraud screening before following the standard risk analysis and approval procedures. This results in operational costs and restrictions that cannot be overlooked, as noted in Almhaithawi et al. (2020); Vanderschueren et al. (2022a) and Dal Pozzolo et al. (2015), for which a percentage of operations to be analyzed (predicted positive, PP) restriction of ideally less than 5% and no more than 10% is imposed by ASF. Fraud detection in consumer finance presents several challenges. Fraudsters adapt to risk policies and often modify their information, resulting in class overlap as noted in King and Zeng (2002). This makes it difficult to identify patterns when training supervised models. In addition, fraudulent cases are rare, resulting in an extraordinarily unbalanced problem (Almhaithawi et al. (2020)). Finally, as pointed out in Altman et al. (1994); Lucas and Jurgovsky (2020) and Min and Lee (2008), regulatory constraints limit the use of complex models such as neural networks in consumer finance, since all decisions made must be explainable to the regulator and the customer.

As introduced in Chapter 1, fraud should be acknowledged as an *instance-dependent cost-sensitive* problem. Consequently, the misclassification error setting introduced in Table 1.1 for the fraud detection problem is considered. To measure model performance, *savings* in equation (1.2) is considered as an aggregated metric of the estimated classification over a sample. The objective is to minimize $\sum_{i=1}^{n} \ell(\hat{y}_i, w_i, y_i)$, equivalently to maximize savings (1.2), along possible classifiers.

The literature, although extensive, lacks state-of-the-art references due to the absence of publicly available datasets and consequent comparisons as mentioned above. To overcome this, a wide range of models and methods are presented and tested in this chapter. Among the cost-insensitive approaches, undersampling and oversampling techniques as the proposed in Almhaithawi et al. (2020); Dal Pozzolo et al. (2015); Elkan (2001) and Lucas and Jurgovsky (2020), although appropriate given the problem imbalance, imply a bias or an increase in variance respectively as

indicated in He and A. Garcia (2009); Dal Pozzolo et al. (2015) and Bahnsen et al. (2014b). Classification techniques as support vector machines (SVM) (Yusuf and Ekrem (2011)), data envelopment analysis (DEA) methods (Min and Lee (2008)) or fuzzy models (Malhotra and Malhotra (2002); Piramuthu (1999)) do not consider the loan amount in the decision making, which is an important drawback. In addition, given the difficulties presented, more complex models are unlikely to achieve a better classification than state-of-the-art models as stated in Altman et al. (1994); Desai et al. (1996); Nikolaou et al. (2016); Dal Pozzolo et al. (2015) and West (2000).

Regarding cost-sensitive approaches, there are two different philosophies as introduced in Vanderschueren et al. (2022b). The first one, known as *predict-and-optimize*, consists in fitting classifiers to minimize the expected loss. The second one, known as *predict-then-optimize* or *thresholding*, is to focus on the decision making. A predictive model is trained with the goal of maximizing accuracy, and then decision-making is optimized by minimizing losses. The drawback of these techniques is that losses are considered only in the second stage and classification rules are individual, overlooking aggregated losses.

In this section and the next, the CS classification problem is addressed from the perspective of thresholding in order to solve the problems of the previous techniques. A novel decision space is constructed using the variables on which losses depend: an estimated fraud probability and the credit amount. In this expanded space, there is more freedom to search for the optimal decision rule, which is estimated with a newly proposed algorithm. It incorporates and extends all previous thresholding approaches, so that an improvement is obtained. It also allows the constrained optimal decision search, something that any previous approach solve, as mentioned in Vanderschueren et al. (2022a) and Verbraken et al. (2014). Since the algorithm works with a given cost specification and an estimated probability, it can be generalized to any other cost-sensitive problem, such as churn prediction, credit risk, medical diagnosis, or logistic planning.

The rest of the chapter is as follows. Section 2.2 presents state-of-the-art approaches for the fraud probability estimation. Section 2.3 list the available thres-

holding strategies, emphasizing their drawbacks and motivating the proposed methodology, which is explained in Section 2.4. Finally, Section 2.5 studies the performance of the different combinations of classifiers and thresholding strategies over two real data sets. One was provided by ASF and the other one is a widely used open fraud data set. Conclusions and future extensions are included in Section 2.6.

2.2 Classification methods

Probability estimation methods are presented in this section, so that the thresholding strategies introduced in the next section can be applied afterwards. Different non- and cost-sensitive approaches are introduced, so that in the practical application it could be tested if the latter help in the posterior thresholding, in line with Vanderschueren et al. (2022b). Boosting approaches are introduced to provide a measure of the predictive power that can be achieved by considering a more complex model. Other complex methods, such as bagging and SVM, are not considered, since boosting has been shown to have better behavior in practical applications due to its simplicity, flexibility, and performance, as observed in Yusuf and Ekrem (2011) and Nikolaou et al. (2016). In addition, they cannot be used in practice due to the aforementioned interpretability limitations, so only boosting seems to be sufficient as a benchmark.

Recently, nonparametric models have been proposed to estimate PD in credit risk problems in Cao et al. (2009); Peláez et al. (2021) and Peláez et al. (2020), with promising results. However, this branch of the literature is not considered in this work because the computational time is too high for the financial context in which this problem is framed.

2.2.1 Logistic regression

In practice, fraud detection is often addressed as a mere classification problem. Logistic regression is the *de facto* model in credit risk, modeling the probability as:

$$P(Y = 1 \mid \mathbf{X} = \mathbf{x}) = p_\theta(\mathbf{x}_i) = \frac{1}{1 + \exp^{-(\beta_0 + \boldsymbol{\beta}' \mathbf{x})}} \tag{2.1}$$

The problem is to estimate the parameter $\boldsymbol{\theta} = (\beta_0, \boldsymbol{\beta})$ that maximizes the log-likelihood function in equation (2.2) below with $\omega_i = 1/n$ for $i = 1, \cdots, n$. It gives equal weight to both types of classification errors, which is not the case in many real-world applications such as fraud detection. The goal is to maximize accuracy, which does not necessarily imply minimizing losses, as shown in the Chapter 1.

2.2.2 Weighted logistic regression

To improve and adapt logistic regression to the cost-sensitive setting, weights are introduced into the log-likelihood function as suggested in Bahnsen et al. (2014a); King and Zeng (2002) and Pesántez-Narváez and Guillén (2020):

$$\mathcal{L}(\theta) = -\sum_{i=1}^{n} \omega_i \left[y_i \log(p_\theta(\mathbf{x}_i)) + (1 - y_i) \log(1 - p_\theta(\mathbf{x}_i)) \right] \tag{2.2}$$

where $p_\theta(\mathbf{x})$ is as defined in (2.1) and ω_i is the weight of the i-th instance. The probability modeling is the same as in equation (2.1), but it is considered a different objective function. This affects the model parameter estimation, and thus the classification. Weights are introduced either to balance the data set (Pesántez-Narváez and Guillén (2020)) or to give more emphasis to an operation depending on its amount (Bahnsen et al. (2014a); Höppner et al. (2021)), which is expected to improve classification error costs.

2.2.3 Boosting algorithms

Boosting (see Hastie et al. (2009)) is an ensemble learning algorithm that outputs the estimated probability as a sum of T weak classifiers, whose performance is

slightly better than random guessing. In this work, shallow binary decision trees are considered as weak classifiers. Given a sample $(y_i, \mathbf{x}_i)_{i=1}^n$, a tree m_1 with Q terminal leaves is fitted depending on $\mathbf{x}$. This tree has associated a weight vector $\beta^{(1)} \in \mathbb{R}^Q$, which assigns the weight $\beta_q^{(1)}$ to each terminal node $q \in \{1, \ldots, Q\}$. Next, a tree m_2 is fitted with weights $\beta^{(2)}$ to improve the previous prediction. This is iterated T times and the estimated conditional probability is computed as:

$$\hat{p}(\mathbf{x}) = \sum_{t=1}^{T} m_t(\mathbf{x}) \tag{2.3}$$

Adaboost and XGBoost are two of the most outstanding boosting algorithms, the main difference being the regularization function present in XGBoost. They are introduced below.

AdaBoost

In Adaboost (see Nikolaou et al. (2016); Freund and Schapire (1997)), each weak classifier is trained on a new weighted data set that gives more weight to observations that were misclassified in previous models. Thus, on each iteration, the focus is on refining the classification of all points as the algorithm progresses. Given a sample $(y_i, \mathbf{x}_i)_{i=1}^n$, the weights for each observation i are initialized equally $v_i^1 = 1/n$, and a weak classifier m_1 is fitted maximizing accuracy, acc_1. At each successive step t, m_t is fitted with updated weights $v_i^t \propto v_i^{t-1} e^{-y_i \alpha_t m_{t-1}(\mathbf{x}_i)}$, where $\alpha_t = \log(acc_{t-1}/(1 - acc_{t-1}))$.

XGBoost

Extreme Gradient Boosting (XGBoost) (see Chen and Guestrin (2016) and Nikolaou et al. (2016)) is a boosting algorithm that trains the model with respect to the objective function:

$$\mathcal{L}(\theta) = \sum_{i=1}^{n} \ell(\hat{p}(\mathbf{x}_i), y_i) + \sum_{t=1}^{T} \Omega(m_t) \tag{2.4}$$

where ℓ is a differentiable convex objective function, as the log-likelihood in equation (2.2) with $\omega_i = 1/n$ for $i = 1, \ldots, n$, $\Omega(m_t) = \gamma T + \frac{1}{2}\lambda \|\beta\|^2$ and $\lambda, \gamma \geq 0$ constants

that penalize model complexity. The regularization function aims to choose a model that uses simple functions by penalizing the complexity of each tree, which prevents overfitting. In each step, a tree is trained to improve the previous prediction. For a prediction, $\hat{p}(\mathbf{x}_i)^{(t-1)}$, of the i-th observation at the $(t-1)$-th iteration, m_t is trained to minimize:

$$\mathcal{L}^{(t)} = \sum_{i=1}^{n} \ell(\hat{p}(\mathbf{x}_i)^{(t-1)} + m_t(\mathbf{x}_i), y_i) + \Omega(m_t)$$

with a gradient descent algorithm. This process is iterated T times and the final prediction computed as in (2.3).

LightGBM

The light gradient-boosting machine (LightGBM), proposed in Ke et al. (2017), is another boosting algorithm similar to XGBoost. The main difference lies in the way it builds trees. XGBoost grows a tree level by level. Instead, LightGBM grows trees leaf-wise, so in each step it selects the leaf that is expected to improve the objective function more and grows the tree from that leaf. In addition, the split point for the decision trees is selected using a histogram-based approach, yielding a greater efficiency. It implements more refinements to run faster than previous approaches while maintaining a high level of accuracy, which is the main advantage of this algorithm.

2.2.4 Instance-dependent cost-sensitive logistic regression

In this approach, introduced in Höppner et al. (2021) as *cslogit*, and studied in Chapter 4, the novelty is that the parameter estimation is carried out in a cost-sensitive manner, minimizing the average expected cost (AEC) of a given loss function (1.1):

$$AEC(\boldsymbol{\theta}; \mathbf{x}_i, w_i, y_i) = \frac{1}{n} \sum_{i=1}^{n} \ell(p_\theta(\mathbf{x}_i), w_i, Y_i) \tag{2.5}$$

where $p_\theta(\mathbf{x}_i)$ is as in equation (2.1). Then, $\boldsymbol{\theta}$ is estimated as the minimizer of (2.5), which can be found using a gradient descent algorithm. By incorporating the AEC

into the model fitting, there is a higher probability of detecting high amount frauds, leading to an improvement in terms of savings.

2.2.5 Instance-dependent cost-sensitive boosting

Introduced in Höppner et al. (2021), this method is constructed as the XGBoost model introduced in Section 2.2.3 but optimizing AEC, instead of the accuracy. To do this, $\ell(\mathbf{x}_i, w_i, Y_i) = AEC(\boldsymbol{\theta}; \mathbf{x}_i, w_i, Y_i)$ is considered in the objective function (2.4). Since the AEC is introduced in the model fitting, it is more likely to rely on high amount frauds and therefore obtain greater savings.

2.3 Previous thresholding approaches

All classifiers generate probability estimates. Thresholding selects a decision strategy based on the cost of misclassification, converting cost-insensitive learning algorithms into cost-sensitive ones, as stated in Sheng and Ling (2006) and Vanderschueren et al. (2022b). Most approaches rely on the estimated probability, for which calibrated probabilities are needed, as noted in Bahnsen et al. (2014b). This adds a degree of complexity as these are not always easy to obtain, especially in an unstable setting such as fraud detection. State-of-the-art approaches are presented in this section and represented in Figure 2.1. This figure shows their differences and shortfalls. For the sake of confindentiality, the amount is shown in logarithm, rescaled to the interval $[0, 10]$. This notation is followed throughout the work.

2.3.1 Youden's J statistic

Youden's J statistic, introduced in Youden (1950), is often used in dichotomous decision problems, taking its maximizer as the optimal classification threshold. It is

defined as:

$$J = \text{recall} + \text{specifity} \ - 1 = \frac{\sum_{i=1}^{n} y_i \hat{y}_i}{\sum_{i=1}^{n} y_i} + \frac{\sum_{i=1}^{n} (1 - y_i)(1 - \hat{y}_i)}{\sum_{i=1}^{n} (1 - y_i)} - 1 \tag{2.6}$$

where $\hat{y}_i = \mathbb{I}(\hat{p}(\mathbf{x}_i) > h)$. The cut-off point h, which maximizes J, (2.6) minimizes the false positive and false negative rates. However, as shown in Chapter 1, maximizing accuracy may be suboptimal when the goal is loss reduction.

2.3.2 Brute force threshold

In order to scrutinize the best strategy considering only the estimated fraud probability, an empirical exhaustive search is considered. A grid is constructed dividing the one-dimensional decision space in $1,000$ equally spaced intervals. The savings (1.2) obtained by considering each cutoff point h in the grid is computed. The one that produces the maximum is taken as the classification threshold, as suggested in Sheng and Ling (2006). Since the resulting savings are computed for each cutoff, the restricted search can also be implemented by considering only those thresholds that satisfy the positive predicted (PP) constraint.

2.3.3 Bayes minimum risk

The Bayes minimum risk (BMR) approach (see Bahnsen et al. (2013) and Elkan (2001)) is the theoretical optimal cost-sensitive decision rule. Given an exogenous variable, W, and an estimated probability, $\hat{p}(\mathbf{x})$, the *risk* of a data point is defined as:

$$R(\hat{y}, w \mid \mathbf{x}) = \ell(\hat{y}, w, 0)(1 - \hat{p}(\mathbf{x})) + \ell(\hat{y}, w, 1)\hat{p}(\mathbf{x})$$

where $\hat{y} \in \{0, 1\}$ and ℓ is a loss function as in (1.1). Then, an operation is labeled as fraud if $R(1, w \mid \mathbf{x}) \leq R(0, w \mid \mathbf{x})$, i.e. if the risk of classifying it as fraud is lower than the risk of classifying it as legitimate. This leads to the decision rule $\hat{y}_i = \mathbb{I}(\hat{p}(\mathbf{x}_i) > h_i)$ with:

$$h_i = \frac{C_i^{FP} - C_i^{TN}}{C_i^{FP} - C_i^{TN} + C_i^{FN} - C_i^{TP}} = \frac{aw_i + b}{(1 + a)w_i} \tag{2.7}$$

Note that although this approach is theoretically optimal, it may not be when considering the aggregated sample results as outlined in Höppner et al. (2021). For example, for $c = 1$, $b = 10e$ and $a = 0.004$, a data point with $w_i = 300e$ is analyzed for fraud if $\hat{p}(\mathbf{x}_i) \geq 0.037$. Suppose that there are 40 operations, one of which is a fraud, with $\hat{p}(\mathbf{x}_i) = 0.04$ and $w_i = 300e$, likely to occur in an unbalanced problem as fraud detection. The aggregated cost would be $40 \cdot 10e$ versus a $300e$ fraud. Thus, a global strategy is more likely to produce better practical results. Moreover, the frontier defined by (2.7) does not allow any flexibility in order to adapt the decision region, so the method cannot be used in practice because it does not fulfill the imposed PP constraints.

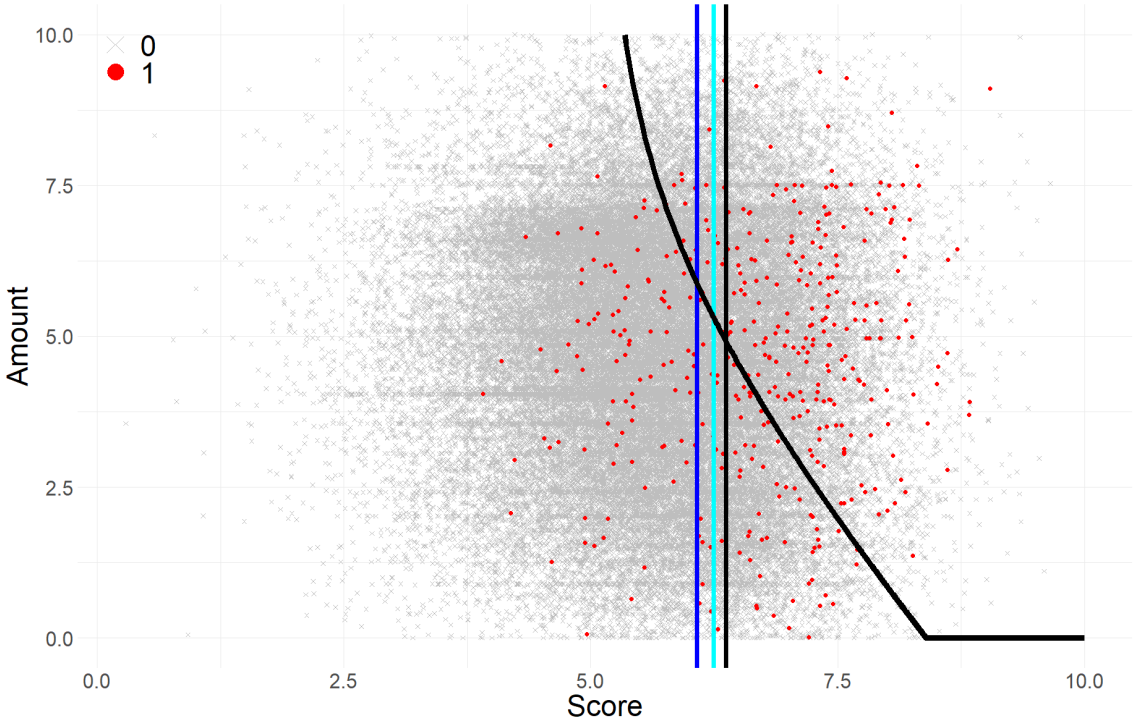

Figure 2.1: State-of-the-art thresholding approaches: Youden's J Statistic (cyan), brute force (black), BMR (black parabola), fixed cost matrix threshold (blue).

2.3.4 Fixed cost matrix

Classical approaches consider a fixed value in the four entries of the cost matrix introduced in Table 1.1 for all i. The optimal threshold in terms of misclassification

costs is the same as that defined in equation (2.7), but with a fixed cut-off for each instance, as proposed in Elkan (2001). Thus, it can be considered as the fixed-threshold version of the BMR approach. Considering the mean of the instance-dependent cost matrix as proposed in Vanderschueren et al. (2022b), the optimal decision cut-off point becomes $h = \frac{1}{n}\sum_{i=1}^{n} h_i$, where h_i is as in equation (2.7).

2.4 Two-dimensional thresholding

In order to overcome the limitations of the thresholding approaches presented in Section 2.3, we propose to extend the decision space to a two-dimensional map generated by the estimated probability, $\hat{p}(\mathbf{x})$, and the loan amount, W, as represented in Figure 2.1. In this space a more flexible and effective decision region can be explored, which is done with a newly proposed algorithm.

For numerical optimization, given a sample $\{(\hat{p}(\mathbf{x}_i), w_i)\}_{i=1}^{n}$, a grid, $G(k)$, is defined depending on a k parameter that controls the smoothness of the search:

$$G(k) = \{(\hat{p}_{\min} + s\delta_1, w_{\min} + t\delta_2)\}_{s,t=0}^{k-1} \tag{2.8}$$

where

$$\begin{gathered} \delta_1 = (\hat{p}_{\max} - \hat{p}_{\min})/k \\ \delta_2 = (w_{\max} - w_{\min})/k \\ \hat{p}_{\min} = \min\{\hat{p}(\mathbf{x}_i)\}_{i=1}^{n},\ \hat{p}_{\max} = \max\{\hat{p}(\mathbf{x}_i)\}_{i=1}^{n}, \\ w_{\min} = \min\{w_i\}_{i=1}^{n},\ w_{\max} = \max\{w_i\}_{i=1}^{n} \end{gathered} \tag{2.9}$$

When considering a two-dimensional decision space, a simple generalization is to take a threshold in each dimension. Thus, the decision region consists of an upper right quadrant $Q_{\mathbf{r}} = \{(\hat{p}, w) \in \mathbb{R}^2 \mid \hat{p} > r_1, w > r_2\}$ defined by a point $\mathbf{r} = (r_1, r_2)$. Given a set of points, $R = \{\mathbf{r}_j\}_{j=1}^{m}$, a decision region is constructed as the union of the upper right quadrants $Q_{\mathbf{r}_j}$,

$$D(R) = \bigcup_{j=1}^{m} Q_{\mathbf{r}_j} \tag{2.10}$$

Given an instance $(\hat{p}_i, w_i)$ and a decision region $D(R)$ as (2.10), its labeling is defined as $\hat{y}_i^R = \mathbb{I}((\hat{p}_i, w_i) \in D(R))$. For a sample $\mathcal{X} = \{(\hat{p}_i, w_i, y_i)\}_{i=1}^n$, savings is defined as:

$$\mathcal{S}(R \mid \mathcal{X}) = 1 - \frac{\sum_{i=1}^n \ell(\hat{y}_i^R, w_i, y_i)}{\sum_{i=1}^n y_i w_i} \tag{2.11}$$

Algorithm 2-DDR(k) (2-dimensional decision region algorithm depending on the parameter k) is proposed for the optimal decision-making estimation. It starts with a decision region defined by the northeasternmost point of the grid $G(k)$, the one with the highest estimated fraud probability and amount. Recursively, each of the points of $G(k)$ surrounding the current decision region is added to the current region as in (2.10) and savings are calculated as in (2.11). The point whose inclusion gives the largest savings increase is added. If there is no savings improvement over the previous decision region, the next surrounding points of $G(k)$ are explored. The algorithm stops when the minimum in the data support is reached.

An example of the first iterations of Algorithm 2-DDR(k) is shown in Figure 2.2. Starting with a preliminary decision region, the savings are calculated taking into account the surrounding points of the grid. Since no improvement is obtained, the next surrounding vertices are explored. This time an improvement is obtained, so it is updated.

The algorithm works on a given probability/score and loss function (1.1). Consequently, it can be generalized to any cost-sensitive problem by considering a different loss function. The search is performed over the entire space, so if the optimal decision has the shape of one of the thresholding proposals introduced in Section 2.3, it will be found except for some roughness depending on the parameter k. Thus, Algorithm 2-DDR(k) is expected to improve (or at least reproduce) state-of-the-art thresholding approaches in terms of savings. Furthermore, calibrated probabilities are not needed because the proposed approach relies only on the points ordering, thus reducing the complexity of the problem. In addition, it allows the exploration of constrained decision rules by iterating until the PP constraint (e.g., 10% or 5%) is satisfied. Thus, the optimal decision rule can be estimated in any CS problem.

Algorithm 2-DDR(k) Two-dimensional decision region algorithm

Data $\mathcal{X} = \{(\hat{p}_i, w_i, y_i)\}_{i=1}^{n}$

Input k parameter

Compute

  Steps δ_1 and δ_2 as (2.9) and the grid $G(k)$ as defined in (2.8);

  $R := (\hat{p}_{\max}, w_{\max})$ as defined in (2.9);

$F \leftarrow R$

while $\min(F) \neq (\hat{p}_{\min}, w_{\min})$ **do**

  $R_{old} \leftarrow R$

  $s \leftarrow \mathcal{S}(R_{old} \mid \mathcal{X})$, as defined in (2.11)

  $t \leftarrow 1$

  while $R = R_{old}$ **do**

    $F :=$ subset of $G(k)$ at distance $t\delta_1$ in the first dimension or $t\delta_2$ in the second dimension from $D(R)$

    $\mathbf{f}_m = \arg\max_{f \in F} \{\mathcal{S}(R \cup \{f\} \mid \mathcal{X})\}$

    if $\mathcal{S}(R \cup \{\mathbf{f}_m\} \mid \mathcal{X}) > s$ **then**

      $R \leftarrow R \cup \{\mathbf{f}_m\}$

    end if

    $t \leftarrow t + 1$

  end while

end while

Output A decision region $D(R)$ defined as in (2.10)

The resulting regions for the ASF dataset are shown in Figure 2.3 over the proposed two-dimensional decision space. The algorithm focuses, as expected, on high fraud probability and high amount operations. It also avoids regions with a high density of legitimate points, where the cost of analysis does not compensate for the fraudulent amount detected when aggregated costs are considered. The strength of the algorithm is that this intuitive logic is developed automatically, without the need for additional estimation or parameter tuning.

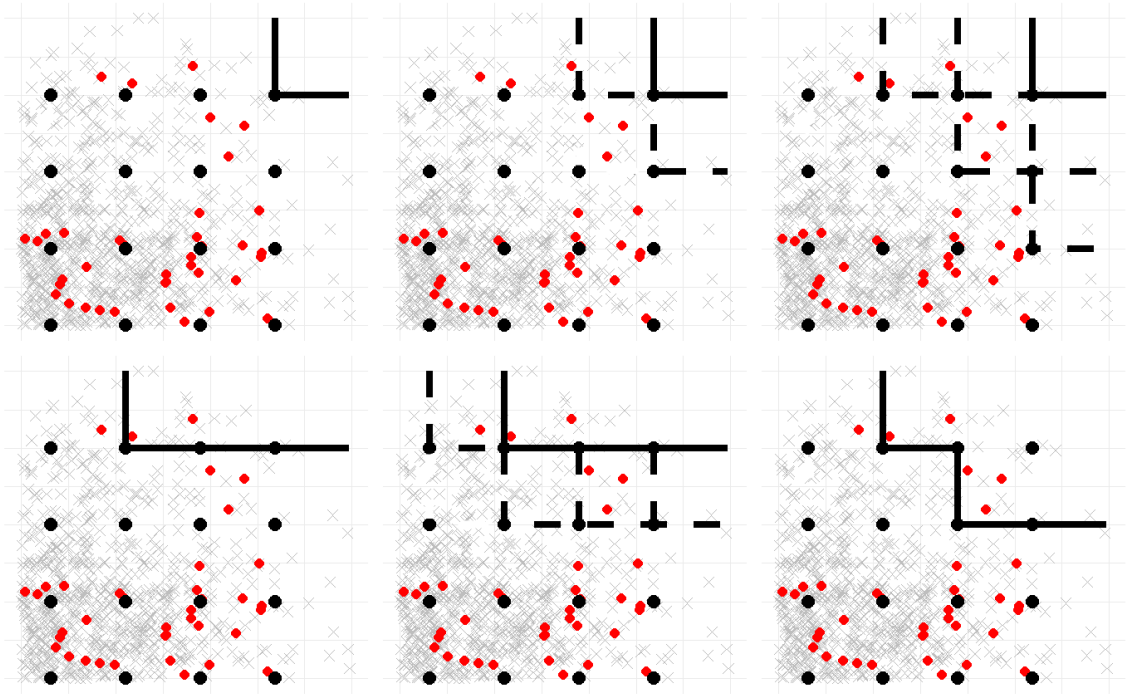

Figure 2.2: Algorithm 2-DDR representation along the grid $G(k)$ (black dots), evaluated decision region (dashed lines) and the updated decision region (solid lines).

Algorithm 2-DDR(k) depends only on one parameter, k, which determines how thorough the search is. A larger k implies a finer grid, which gives more flexibility to the search. In practice, we would suggest starting with a small value of k and trying larger values until a plateau in savings in a validation set is reached. In our experience, this is achieved with k smaller than the number of cases. As for the computational complexity, it depends mainly on k^2 as discussed in Section 2.5.3. This makes the algorithm suitable for scaling to very large datasets, since it does not depend on the sample size.

Finally, note that the algorithm is inherently stable and robust to outliers. Figure 2.4 (left) shows the estimated frontiers for 5 simulated data from the same model, with $40,000$ instances and a fraud percentage of 0.5% . It can be seen that there are slight differences depending on the sample, but the frontiers have a stable shape. Figure 2.4 (right) shows the estimated decision region when considering four types of possible fraud outliers, placing 20 of each type in the score-amount space. Only high amount frauds are susceptible of influencing the algorithm. If they also have a

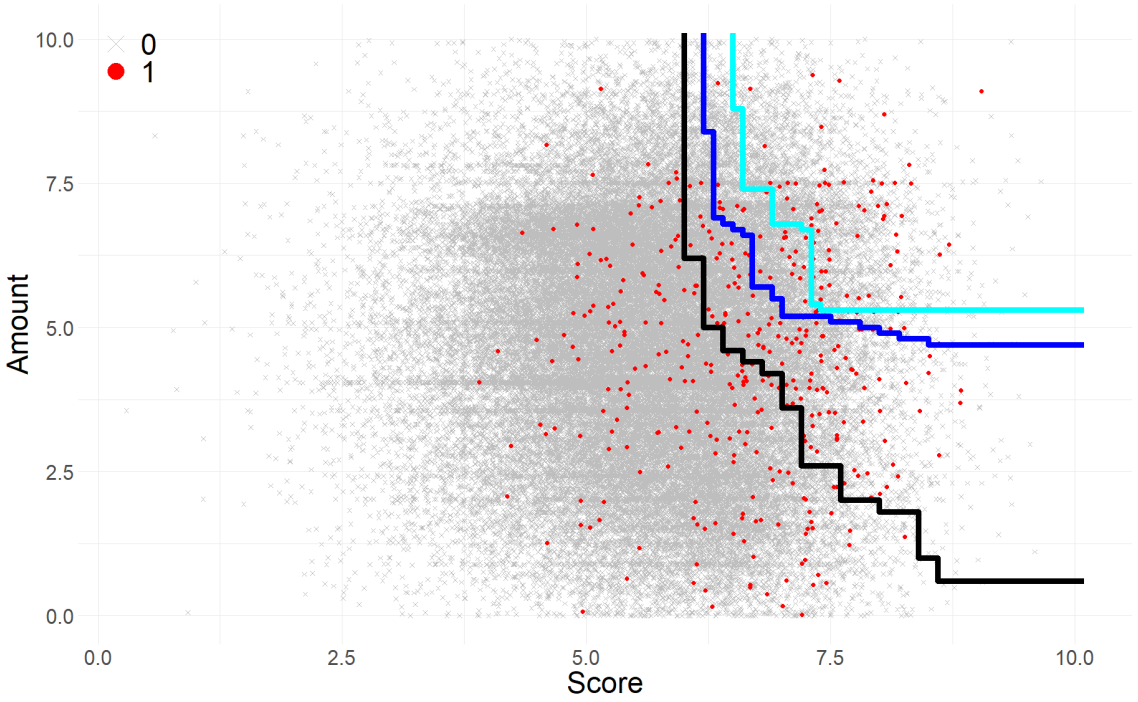

Figure 2.3: Optimal decision frontiers in terms of savings estimated running Algorithm 2-DDR(k) with $k = 100$, fulfilling that PP $\leq 100\%$ (black), $\leq 10\%$ (blue), $\leq 5\%$ (cyan).

high estimated probability of being fraudulent, they are included in the first steps of the decision region and the same frontier is obtained. If they have a low estimated probability, the same optimal decision region is obtained, but these outliers are included (as they imply a significant loss), which would have no consequences in practice.

This proposal, Algorithm 2-DDR(k), belongs to the class of greedy algorithms. The algorithm iteratively makes an optimally local choice after another, reducing the problem into a smaller one. There is no a universal theoretical result that guarantees the optimality of greedy algorithms. Huffman trees for discrete/categorical variable simulation, Dijkstra's algorithm for shortest path search, and Kruskal algorithm for finding minimum spanning trees are examples of optimal greedy algorithms. Although, there exist other well-known examples of greedy algorithms (as the nearest unvisited city algorithm for the traveling salesman problem) that may perform much worse in pathological examples. Just to examine the practical performance of Algorithm 2-DDR(k), some experimental scenarios are studied in Section 2.5.

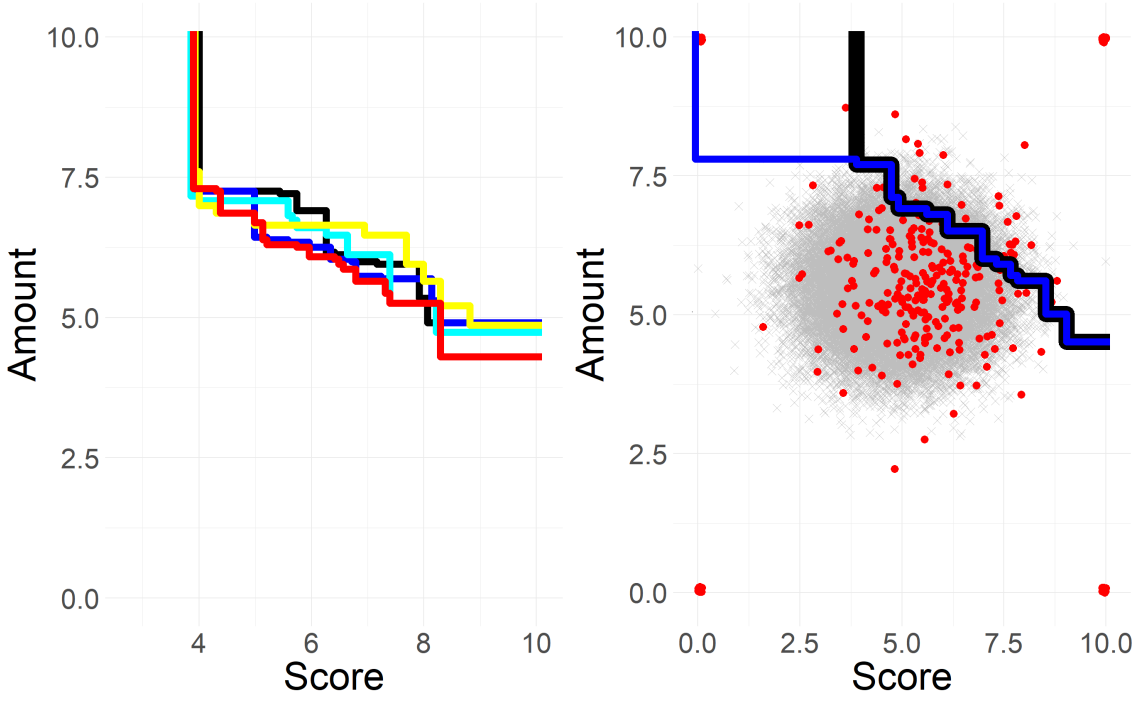

Figure 2.4: Estimated decision frontiers running Algorithm 2-DDR over 5 simulated data from the same model (left) and over a sample with fraudulent outliers situated in the four corners of the graph (right), when not considering the outliers (black) and when considering the outliers (blue).

2.5 Experimental results

Two real data sets are presented to evaluate the performance of all the approaches introduced in this chapter. Both consist of real fraud datasets, according to which they exhibit the difficulties presented in Section 2.1 such as extreme imbalance and class overlap. To evaluate the classifiers performance independently from the thresholding strategy, we rely on threshold-independent metrics, namely AUC, Gini index, the Kolmogorov-Smirnov statistic (KS) and H-measure (H). The latter may be more informative given the high degree of class imbalance as indicated in Vanderschueren et al. (2022b). Expected savings (ES) $= 1 - \frac{\sum_{i=1}^{n} \ell(\hat{p}_i, w_i, Y_i)}{\sum_{i=1}^{n} Y_i w_i}$, inspired from (2.5), are also summarized. To evaluate the impact of the threshold strategy, the accuracy (Acc), recall (Rec), specificity (Spec), and F-score (F) are reported along with the objective function, savings (Sav), defined in (1.2).

The classifiers introduced in Section 2.2 are trained over the two data sets.

These are logistic regression (LR), weighted logistic regression (WLR), Adaboost (AB), XGBoost (XGB), LightGBM (LGBM), cslogit (CSL), and csboost (CSB). The thresholding methods introduced in Section 2.3 and the algorithm proposed in Section 2.4 with $k = 25, 50, 100$ are applied over the estimated probabilities, previously calibrated following Elkan (2001). Thus, the thresholding approaches that rely on probabilities are not distorted. For each data set, a 5-fold cross validation is performed, stratified by the proportion of frauds and the amount following Höppner et al. (2021). The average results over the test sets are summarized in Section 2.5.2.

2.5.1 Datasets

The Credit Card data set[1], previously considered in Dal Pozzolo et al. (2015) and Höppner et al. (2021), consists of 284,807 credit card transactions over two days, with 492 (0.17%) frauds. It consists of 28 variables resulting from a PCA, along with a "Time" variable (seconds since first use) and the "Amount" of each transaction. The "Class" variable indicates whether an operation is legitimate (0) or fraudulent (1). The ASF data set is a real-world data set of 210,180 loan requests lend by ASF, collected between January 2018 and December 2021, with a fraud rate of 0.67%. In order to preserve confidentiality, the number of registers is truncated and so is the fraud proportion. The variables considered are summarized in Table 2.1 in terms of their information value (a measure of the relationship between a variable and the odds ratio, see Siddiqi (2006)). Available variables are limited to information provided by the customer at the time of the request and information about the point of sale. There is no behavioral history or extensive databases to draw on, as is the case with credit cards, which is another handicap for modeling fraud. Only formalized requests are considered, since nothing can be assured about a non-formalized operation. Note that these are the operations of interest and the most difficult to detect, as they are the ones that have passed all the filters and controls.

[1] `kaggle.com/mlg-ulb/creditcardfraud`

Variable	Type	IV
Commerce activity	Categorical	0.227
Client activity sector	Categorical	0.162
Housing situation	Categorical	0.152
Marital status	Categorical	0.136
Profession	Categorical	0.112
Region	Categorical	0.108
Commerce class	Categorical	0.032
Previous request indicator	Categorical	0.007
Commerce monthly amount	Continuous	0.007
Age	Continuous	0.004
Commerce mean default rate	Continuous	0.001
Loan term	Continuous	0.001

Table 2.1: Summary of the ASF data set variables.

2.5.2 Results

The results of the classifiers obtained on the Credit Card data set over the test samples are summarized in Table 2.2. The prediction performance of all classifiers is outstanding in terms of classification, as it can be seen in the high values of the AUC. This is due to the number of available variables and their discrimination power. As expected, the highest ES is obtained with a cost-sensitive model (CSB). The flexibility offered by boosting combined with a cost-sensitive approach provides a model capable of detecting high-amount frauds, which leads to the highest ES.

Classifier	AUC	Gini	H	KS	ES
LR	97.56	95.13	87.13	88.99	44.12
WLR	96.81	93.62	85.45	87.73	42.80
AB	97.57	95.15	87.62	89.89	68.13
XGB	97.34	94.68	87.29	89.19	85.97
LGBM	97.62	95.23	82.63	86.87	49.50
CSL	93.85	87.70	81.13	83.54	67.51
CSB	95.45	90.91	82.22	86.05	92.49

Table 2.2: Classification metrics for the different classifiers over the credit card data set.

Table 2.4 summarizes the performance of thresholding approaches for the different classifiers. In this case, the restricted search is not taken into account since it is a public dataset without any imposed restriction. The first highlight is that detecting more fraud does not imply an increase in savings, as can be seen with JS. Since costs are not considered in the decision making, the highest frauds are not detected and there is an increase in PP, leading to high analysis costs and consequently lower savings. This is a clear example of the importance of selecting the right operations to inspect in a cost-sensitive problem.

Csboost and LightGBM obtain the best metrics in the train set, but a much lower performance in the test set, due to a clear overfitting that could become dangerous in practice. This can be seen for all boosting approaches. In the test set, the best results in terms of savings are achieved with significantly smaller PP with the new proposed approach: Algorithm 2-DDR(k). It focuses on detecting high amount frauds, so it obtains the smallest PP with each classifier as well as the highest savings, for which it is considered the outperforming approach.

The smoothness effect of the parameter k is clear, with a direct relation with the savings obtained in the train set and some overfitting in some classifiers. We highlight the cslogit approach, which outperforms all other existing methods in the test set with an understandable model and low PP. Thus, cslogit with Algorithm 2-DDR(k) is the chosen approach for this dataset.

For the ASF data set, the results of the classifiers are summarized in Table 2.3. As expected, cost-sensitive approaches again outperform classical approaches in terms of ES. This setting is more difficult, which is reflected in the smaller values in Table 2.3 compared to the previous data set.

Table 2.5 summarizes the results of the different thresholding strategies. Csboost gives a very good performance, probably due to the scarcity of variables that require more complex modeling. However, it seems to fall into overfitting again, which leads to worse results in the test samples. Algorithm 2-DDR(k) again gives the best results under each classifier in terms of savings (1.2). It is also worth noting

Classifier	AUC	Gini	H	KS	ES
LR	76.11	52.22	21.64	41.15	0.99
WLR	76.11	52.22	21.77	41.47	0.98
AB	77.06	54.12	22.63	42.48	1.14
XGB	77.13	54.27	23.59	42.75	2.56
LGBM	77.18	54.35	26.81	43.88	1.20
CSL	72.34	44.66	15.15	37.15	47.11
CSB	77.00	54.00	23.88	42.88	67.13

Table 2.3: Classification metrics for the different classifiers over the ASF data set.

the fact that Algorithm 2-DDR(k) outperforms all classical approaches in terms of classification metrics except recall, making clear how the algorithm focuses on the more profitable operations to analyze. As a consequence, PP tends to be significantly smaller, which is another advantage. The highest savings are obtained with Adaboost and Algorithm 2-DDR(k) in the test samples, but almost the same savings can be obtained considering logistic regression, a simpler model. This shows how much the results can be improved by considering the proposed algorithm.

The restricted search is performed as it was required by ASF. PP restrictions of 10% and 5% are considered in Table 2.6. The results are parallel to those obtained in the unrestricted case. It is worth noting the 40.25% of savings achieved with the 10% PP restriction compared to the 47.65% of savings achieved with the best unrestricted approach. In this case, the best results are obtained with CSB. Thus, a complex model combined with Algorithm 2-DDR(k) produces satisfactory results, thanks to the ability of the model to discriminate fraud and the flexibility of Algorithm 2-DDR(k) to detect high amount frauds. Likewise, similar results are obtained considering LightGBM, a complex cost-insensitive model, but with the proposed approach costs are taken into account in the decision-making. Nevertheless, taking into account the explainability restriction in the financial context, the model finally selected is the logistic regression together with Algorithm 2-DDR(k). Thus, ASF can have a model that satisfies the interpretability constraints and its own workload constraints and obtains a 33% reduction in fraud losses.

		TRAIN set						TEST set					
Classifier	Threshold	Sav	Acc	Rec	FS	Spec	PP	Sav	Acc	Rec	FS	Spec	PP
LR	JS	-5.43	98.33	**88.98**	99.15	98.34	1.80	-4.48	98.32	**89.25**	99.15	98.34	1.81
	BF	67.65	**99.93**	80.91	99.96	99.96	0.18	63.07	**99.91**	78.49	99.95	99.95	0.18
	CM	66.73	**99.93**	77.96	**99.97**	99.97	0.16	48.21	**99.91**	72.04	**99.96**	99.96	0.16
	BMR	68.80	99.86	38.98	99.93	99.96	0.10	65.98	99.85	36.56	99.92	99.95	0.11
	2-DDR(25)	70.78	99.88	40.05	99.94	**99.98**	0.09	**67.76**	99.87	35.48	99.93	**99.97**	**0.08**
	2-DDR(50)	71.16	99.88	39.78	99.94	**99.98**	**0.08**	66.12	99.86	34.41	99.93	**99.97**	**0.08**
	2-DDR(100)	**73.30**	99.86	42.20	99.93	99.96	0.11	63.94	99.83	36.56	99.91	99.93	0.13
WLR	JS	17.55	98.78	**87.63**	99.38	98.80	1.35	16.91	98.77	**88.17**	99.38	98.79	1.36
	BF	68.35	99.86	81.99	99.93	99.89	0.24	49.09	99.83	79.57	99.92	99.87	0.26
	CM	66.86	**99.92**	72.58	**99.96**	99.97	0.15	49.00	**99.91**	67.74	**99.95**	99.96	0.15
	BMR	72.52	99.85	39.78	99.93	99.95	0.11	**65.46**	99.83	36.56	99.91	99.93	0.13
	2-DDR(25)	70.68	99.88	38.17	99.94	**99.98**	**0.08**	51.58	99.86	32.26	99.93	**99.98**	**0.08**
	2-DDR(50)	72.42	99.87	38.71	99.94	99.97	0.09	50.34	99.84	32.26	99.92	99.95	0.10
	2-DDR(100)	**73.07**	99.86	38.98	99.93	99.96	0.11	62.66	99.83	33.33	99.91	99.94	0.12
AB	JS	-139.44	95.60	**93.82**	97.75	95.61	4.54	-116.50	95.72	**93.55**	97.81	95.73	4.42
	BF	64.07	**99.93**	73.12	**99.97**	**99.98**	0.14	55.78	**99.92**	72.04	**99.96**	99.96	0.16
	CM	62.68	99.92	77.42	99.96	99.95	0.17	55.36	**99.92**	75.27	**99.96**	99.96	0.17
	BMR	64.34	99.83	41.40	99.92	99.93	0.14	54.01	99.83	41.94	99.92	99.93	0.14
	2-DDR(25)	67.54	99.88	39.52	99.94	**99.98**	**0.08**	**57.63**	99.87	39.78	99.93	**99.97**	**0.10**
	2-DDR(50)	68.25	99.87	40.32	99.93	99.97	0.10	56.14	99.86	39.78	99.93	99.96	0.11
	2-DDR(100)	**68.44**	99.87	40.32	99.94	99.97	0.10	56.22	99.86	39.78	99.93	99.96	0.11
XGB	JS	9.79	98.56	**90.86**	99.28	98.58	1.57	4.33	98.58	**88.17**	99.28	98.60	1.54
	BF	71.26	**99.95**	84.95	**99.97**	99.97	0.17	63.78	**99.92**	82.80	**99.96**	99.94	0.19
	CM	71.24	**99.95**	85.22	**99.97**	99.97	0.17	63.78	**99.92**	82.80	**99.96**	99.94	0.19
	BMR	74.14	99.84	44.09	99.92	99.94	0.14	63.37	99.81	40.86	99.90	99.91	0.16
	2-DDR(25)	74.96	99.90	42.20	99.95	**99.99**	**0.08**	64.13	99.87	35.48	99.93	**99.98**	**0.08**
	2-DDR(50)	**76.80**	99.89	44.35	99.94	99.98	0.10	63.39	99.86	37.63	99.93	99.96	0.10
	2-DDR(100)	**76.80**	99.89	44.35	99.94	99.98	0.10	**65.25**	99.86	38.71	99.93	99.96	0.10
LGBM	JS	85.81	99.94	**96.77**	99.97	99.94	0.22	49.60	99.89	**77.42**	99.95	99.93	0.20
	BF	87.42	99.97	96.51	**99.99**	99.98	0.18	46.45	99.93	76.34	99.96	99.97	0.16
	CM	81.72	**99.98**	93.82	**99.99**	**99.99**	0.17	47.65	**99.95**	74.19	**99.97**	**99.99**	0.13
	BMR	89.91	99.90	46.77	99.95	**99.99**	**0.09**	**54.74**	99.87	32.26	99.94	**99.99**	**0.07**
	2-DDR(25)	90.99	99.91	49.19	99.96	**99.99**	**0.09**	50.04	99.88	34.41	99.94	**99.99**	**0.07**
	2-DDR(50)	90.99	99.91	49.19	99.96	**99.99**	**0.09**	49.95	99.88	33.33	99.94	**99.99**	**0.07**
	2-DDR(100)	**91.01**	99.91	49.73	99.96	**99.99**	**0.09**	50.04	99.88	34.41	99.94	**99.99**	**0.07**
CSL	JS	-1.28	98.48	**85.22**	99.23	98.50	1.64	1.94	98.51	**86.02**	99.25	98.53	1.61
	BF	69.40	**99.95**	79.84	**99.97**	99.98	0.15	64.94	**99.92**	76.34	**99.96**	99.96	0.16
	CM	69.23	99.94	80.65	**99.97**	99.97	0.16	66.48	**99.92**	77.42	**99.96**	99.96	0.17
	BMR	71.22	99.89	44.09	99.94	99.98	0.09	67.42	99.86	43.01	99.93	99.95	0.12
	2-DDR(25)	**72.23**	99.89	42.47	99.95	**99.99**	**0.08**	**68.82**	99.87	40.86	99.94	**99.97**	**0.10**
	2-DDR(50)	**72.23**	99.89	42.47	99.95	**99.99**	**0.08**	**68.82**	99.87	40.86	99.94	**99.97**	**0.10**
	2-DDR(100)	**72.23**	99.89	42.47	99.95	**99.99**	**0.08**	**68.82**	99.87	40.86	99.94	**99.97**	**0.10**
CSB	JS	32.35	98.96	**84.68**	99.48	98.98	1.15	34.89	98.89	**83.87**	99.44	98.92	1.22
	BF	91.39	99.89	48.39	**99.95**	99.98	0.10	62.37	99.86	45.16	99.93	99.95	0.13
	CM	92.29	**99.91**	46.77	**99.95**	99.99	**0.08**	**62.97**	**99.88**	40.86	**99.94**	99.97	0.09
	BMR	92.31	**99.91**	46.77	**99.95**	**100.00**	**0.08**	62.76	99.87	40.86	**99.94**	99.97	0.10
	2-DDR(25)	**92.36**	**99.91**	46.77	**99.95**	**100.00**	**0.08**	62.89	99.87	34.41	**99.94**	**99.98**	**0.07**
	2-DDR(50)	**92.36**	**99.91**	46.77	**99.95**	**100.00**	**0.08**	**62.97**	**99.88**	34.41	**99.94**	**99.98**	**0.07**
	2-DDR(100)	**92.36**	**99.91**	46.77	**99.95**	**100.00**	**0.08**	**62.97**	**99.88**	34.41	**99.94**	**99.98**	**0.07**

Table 2.4: Mean results summary in the credit card data set for the combination of all the approaches introduced throughout this chapter.

		TRAIN set						TEST set					
Classifier	Threshold	Sav	Acc	Rec	FS	Spec	PP	Sav	Acc	Rec	FS	Spec	PP
LR	JS	40.33	72.27	68.01	83.81	72.30	27.97	39.56	72.24	66.05	83.79	72.28	27.98
	BF	40.59	64.82	**74.57**	78.44	64.76	35.51	40.27	64.77	**74.85**	78.39	64.70	35.57
	CM	40.26	65.62	73.73	79.12	65.56	34.70	41.19	65.55	74.16	79.06	65.49	34.78
	BMR	45.21	**74.12**	59.36	**85.06**	**74.22**	**26.01**	45.19	**74.14**	58.98	**85.08**	**74.25**	**25.98**
	2-DDR(25)	44.36	72.37	57.53	83.86	72.47	27.73	41.70	72.31	55.85	83.81	72.43	27.77
	2-DDR(50)	46.52	73.23	58.14	84.47	73.33	26.88	**46.73**	73.25	58.17	84.49	73.36	26.86
	2-DDR(100)	**47.02**	73.37	58.81	84.56	73.47	26.75	46.18	73.36	58.63	84.56	73.47	26.75
WLR	JS	40.82	71.49	68.85	83.27	71.50	28.77	40.66	71.51	67.78	83.29	71.54	28.73
	BF	40.54	66.73	72.94	79.89	66.69	33.58	40.96	66.71	72.76	79.87	66.66	33.61
	CM	40.20	65.42	**73.79**	78.97	65.37	34.90	41.01	65.33	**74.04**	78.91	65.27	34.99
	BMR	45.50	**74.05**	59.62	**85.02**	**74.15**	**26.08**	44.51	**74.04**	59.09	**85.01**	**74.14**	**26.08**
	2-DDR(25)	44.11	72.13	57.85	83.69	72.23	27.98	42.91	72.07	56.54	83.65	72.18	28.02
	2-DDR(50)	46.49	73.67	57.65	84.76	73.79	26.43	**45.72**	73.59	56.66	84.70	73.71	26.50
	2-DDR(100)	**47.21**	73.95	58.08	84.95	74.06	26.16	45.51	73.92	57.01	84.93	74.04	26.18
AB	JS	46.09	69.63	73.70	81.96	69.60	30.69	43.46	69.72	70.68	82.03	69.71	30.57
	BF	46.14	66.82	**75.70**	79.99	66.76	33.53	44.14	66.92	**73.23**	80.06	66.88	33.40
	CM	46.21	67.88	74.69	80.75	67.84	32.45	44.50	67.91	72.65	80.77	67.87	32.40
	BMR	49.11	74.53	61.71	85.34	74.63	25.62	44.97	74.62	60.95	85.40	74.72	25.53
	2-DDR(25)	48.30	72.56	61.51	83.99	72.64	27.59	45.73	72.77	59.79	84.14	72.86	27.36
	2-DDR(50)	49.63	74.07	60.93	85.03	74.16	26.08	**47.65**	74.31	59.33	85.19	74.42	25.82
	2-DDR(100)	**50.36**	**75.00**	60.38	**85.64**	**75.10**	**25.14**	47.02	**75.16**	57.71	**85.75**	**75.28**	**24.94**
XGB	JS	48.80	73.27	74.57	84.46	73.26	27.07	42.91	73.15	67.91	84.40	73.19	27.10
	BF	48.87	70.64	**76.90**	82.68	70.60	29.73	42.87	70.50	70.22	82.59	70.50	29.78
	CM	48.42	70.54	76.78	82.61	70.49	29.83	43.43	70.12	**70.91**	82.33	70.12	30.17
	BMR	51.33	**76.33**	64.30	**86.51**	**76.42**	**23.86**	45.70	76.13	58.86	86.38	76.25	23.99
	2-DDR(25)	49.46	74.02	63.45	84.99	74.09	26.16	44.84	73.83	60.26	84.87	73.93	26.31
	2-DDR(50)	50.55	73.06	63.92	84.35	73.13	27.13	44.76	72.88	59.56	84.23	72.97	27.25
	2-DDR(100)	**51.75**	76.28	62.64	86.47	76.37	23.90	**46.29**	**76.17**	59.09	**86.41**	**76.29**	**23.95**
LGBM	JS	52.38	78.07	69.14	87.57	78.13	22.19	45.18	77.84	61.77	87.42	77.95	22.33
	BF	52.51	78.92	68.01	88.14	78.99	21.33	44.96	78.72	60.84	88.02	78.84	21.43
	CM	48.73	67.99	**77.01**	80.82	67.93	32.38	42.43	67.71	**70.69**	80.63	67.69	32.57
	BMR	44.36	70.54	63.22	82.64	70.59	29.64	39.82	70.35	56.32	82.52	70.45	29.74
	2-DDR(25)	52.61	84.67	57.59	91.66	84.86	15.43	43.54	84.43	47.85	91.53	84.68	15.54
	2-DDR(50)	53.11	84.58	56.75	91.60	84.77	15.52	**45.35**	84.34	46.70	91.47	84.60	15.61
	2-DDR(100)	**53.61**	**86.84**	55.65	**92.92**	**87.05**	**13.24**	43.93	**86.56**	44.72	**92.77**	**86.84**	**13.37**
CSL	JS	48.18	71.50	66.74	83.28	71.53	28.73	44.01	71.47	63.84	83.27	71.53	28.71
	BF	48.31	71.91	66.10	83.56	71.95	28.31	44.25	71.89	63.26	83.55	71.95	28.29
	CM	47.33	70.39	**67.32**	82.52	70.41	29.85	43.47	70.34	**64.77**	82.49	70.38	29.86
	BMR	47.27	70.69	66.65	82.72	70.72	29.54	43.94	70.66	64.31	82.71	70.71	29.53
	2-DDR(25)	48.49	73.92	59.07	84.93	74.03	26.20	44.44	73.98	56.77	84.97	74.10	26.11
	2-DDR(50)	48.60	74.19	58.43	**85.11**	74.30	**25.92**	**44.56**	**74.23**	55.97	**85.14**	**74.36**	**25.85**
	2-DDR(100)	**48.61**	**74.20**	58.49	**85.11**	**74.31**	**25.92**	44.55	**74.23**	56.55	85.13	74.35	25.86
CSB	JS	72.62	85.22	75.99	91.97	85.28	15.14	44.63	85.16	50.87	91.95	85.40	14.85
	BF	72.64	**86.57**	73.90	**92.75**	**86.65**	**13.76**	44.68	**86.68**	47.85	**92.83**	**86.95**	**13.29**
	CM	66.80	76.03	**82.44**	86.29	75.99	24.41	44.61	75.84	**65.47**	86.19	75.91	24.37
	BMR	66.78	77.36	79.80	87.15	77.33	23.05	44.17	77.22	63.15	87.08	77.31	22.97
	2-DDR(25)	72.87	86.00	74.57	92.42	86.08	14.34	**44.87**	86.05	48.67	92.47	86.31	13.93
	2-DDR(50)	72.90	86.26	74.28	92.58	86.34	14.07	44.59	86.40	47.97	92.67	86.66	13.57
	2-DDR(100)	**72.92**	86.11	74.43	92.49	86.19	14.22	44.57	86.18	48.32	92.54	86.44	13.80

Table 2.5: Mean results summary in the ASF data set for the combination of all the approaches introduced throughout this chapter.

		TRAIN set (5% PP)						TEST set (5% PP)					
Classifier	Threshold	Sav	Acc	Rec	FS	Spec	PP	Sav	Acc	Rec	FS	Spec	PP
LR	BF	18.20	94.75	**23.34**	97.30	95.25	4.88	16.95	94.72	**21.78**	97.28	95.22	4.90
	2-DDR(25)	21.83	**95.17**	12.80	**97.52**	**95.73**	**4.33**	19.36	**95.13**	11.35	**97.50**	**95.70**	**4.35**
	2-DDR(50)	22.91	94.99	16.40	97.43	95.53	4.55	**22.23**	95.03	15.88	97.45	95.57	4.51
	2-DDR(100)	**24.90**	94.85	14.66	97.35	95.40	4.67	21.07	94.87	11.93	97.36	95.44	4.61
WLR	BF	18.24	94.77	**23.31**	97.31	95.26	4.86	16.84	94.74	**21.78**	97.30	95.25	4.87
	2-DDR(25)	21.32	**94.80**	13.79	**97.33**	**95.36**	**4.70**	20.99	**94.82**	13.21	**97.34**	**95.38**	**4.68**
	2-DDR(50)	25.56	94.60	15.73	97.22	95.14	4.93	22.45	94.60	13.67	97.22	95.15	4.91
	2-DDR(100)	**26.05**	94.75	14.37	97.30	95.31	4.76	**23.63**	94.75	12.52	97.31	95.32	4.73
AB	BF	20.36	94.79	**26.18**	97.32	95.27	4.88	19.29	94.84	**24.10**	97.34	95.32	4.81
	2-DDR(25)	24.52	**95.33**	14.86	**97.61**	**95.88**	**4.19**	22.01	**95.40**	13.09	**97.64**	**95.96**	**4.10**
	2-DDR(50)	26.85	95.00	17.01	97.43	95.54	4.55	24.64	94.99	15.29	97.43	95.54	4.53
	2-DDR(100)	**28.84**	94.64	17.07	97.24	95.17	4.91	**26.87**	94.62	15.52	97.24	95.17	4.90
XGB	BF	26.60	94.90	**33.38**	97.38	95.33	4.87	19.13	94.91	**26.07**	97.39	95.39	4.76
	2-DDR(25)	27.66	**95.97**	15.88	**97.94**	**96.52**	**3.56**	18.76	**95.91**	10.78	**97.91**	**96.50**	**3.56**
	2-DDR(50)	29.05	95.28	15.56	97.58	95.83	4.25	21.66	95.21	11.59	97.55	95.79	4.26
	2-DDR(100)	**32.04**	95.01	18.55	97.44	95.54	4.56	**25.74**	94.96	14.25	97.41	95.52	4.55
LGBM	BF	33.57	94.95	**39.62**	97.40	95.33	4.90	18.74	94.77	**27.11**	97.31	95.24	4.91
	2-DDR(25)	35.11	**95.74**	28.97	**97.81**	**96.19**	**3.98**	22.99	**95.61**	18.19	**97.75**	**96.14**	**3.95**
	2-DDR(50)	38.20	95.02	29.06	97.44	95.48	4.69	25.91	94.81	18.89	97.33	95.33	4.77
	2-DDR(100)	**40.48**	94.85	29.87	97.35	95.29	4.88	**29.95**	94.62	19.93	97.23	95.14	4.97
CSL	BF	0.00	**99.32**	0.00	**99.66**	**100.00**		0.00	**99.31**	0.00	**99.66**	**100.00**	
	2-DDR(25)	9.70	95.51	4.33	97.70	96.14	**3.86**	11.14	95.59	4.87	97.74	96.21	**3.80**
	2-DDR(50)	15.70	94.71	7.40	97.29	95.32	4.70	15.93	94.80	7.30	97.33	95.41	4.61
	2-DDR(100)	**16.89**	94.59	**7.87**	97.21	95.18	4.84	**16.49**	94.67	**7.65**	97.26	95.26	4.76
CSB	BF	42.24	**95.67**	**30.51**	**97.78**	**96.12**	**4.06**	21.16	**96.11**	**17.95**	**98.01**	**96.65**	**3.45**
	2-DDR(25)	48.81	95.11	25.84	97.49	95.58	4.56	27.62	95.64	14.94	97.77	96.19	3.88
	2-DDR(50)	50.12	94.94	27.11	97.40	95.41	4.75	29.97	95.45	16.91	97.67	95.99	4.09
	2-DDR(100)	**51.96**	94.76	29.09	97.31	95.22	4.95	**30.45**	95.27	17.61	97.58	95.81	4.28

		TRAIN set (10% PP)						TEST set (10% PP)					
Classifier	Threshold	Sav	Acc	Rec	FS	Spec	PP	Sav	Acc	Rec	FS	Spec	PP
LR	BF	28.82	90.07	**39.97**	94.76	90.41	9.80	27.72	90.11	**38.36**	94.78	90.46	9.73
	2-DDR(25)	30.46	**91.15**	19.39	**95.36**	**91.65**	**8.43**	30.23	**91.14**	18.54	**95.35**	**91.64**	**8.43**
	2-DDR(50)	32.59	90.04	21.45	94.75	90.52	9.57	27.63	89.99	17.85	94.73	90.49	9.57
	2-DDR(100)	**33.85**	90.55	23.92	95.03	91.01	9.09	**32.93**	90.64	22.83	95.09	91.11	8.99
WLR	BF	28.67	89.93	**39.97**	94.69	90.28	9.93	27.89	89.98	**38.70**	94.71	90.33	9.87
	2-DDR(25)	29.73	**91.72**	18.78	**95.67**	**92.22**	**7.85**	**30.32**	**91.75**	18.19	**95.69**	**92.25**	**7.82**
	2-DDR(50)	32.53	90.00	21.36	94.73	90.47	9.61	27.90	89.97	18.08	94.71	90.47	9.59
	2-DDR(100)	**33.31**	90.16	22.90	94.82	90.62	9.47	29.01	90.15	19.81	94.81	90.63	9.44
AB	BF	31.03	90.10	**39.71**	94.78	90.44	9.76	28.06	90.13	**36.73**	94.79	90.50	9.69
	2-DDR(25)	33.68	**91.35**	21.71	**95.47**	**91.82**	**8.27**	32.84	**91.43**	20.74	**95.51**	**91.91**	**8.17**
	2-DDR(50)	38.66	89.93	27.72	94.69	90.36	9.77	**37.13**	89.97	25.60	94.71	90.41	9.70
	2-DDR(100)	**38.89**	89.99	26.18	94.72	90.43	9.68	35.06	89.99	23.52	94.72	90.44	9.65
XGB	BF	35.51	90.31	**46.13**	94.89	90.62	9.64	26.65	90.30	**37.65**	94.89	90.66	9.53
	2-DDR(25)	37.48	**90.35**	24.59	**94.92**	**90.81**	**9.30**	32.79	**90.38**	21.67	**94.94**	**90.85**	**9.24**
	2-DDR(50)	40.09	90.23	27.02	94.85	90.66	9.46	31.71	90.29	21.79	94.89	90.76	9.33
	2-DDR(100)	**42.71**	89.86	29.67	94.65	90.27	9.86	**34.25**	89.91	24.22	94.68	90.36	9.74
LGBM	BF	44.74	90.17	**52.10**	94.81	90.43	9.86	33.05	89.99	**41.83**	94.71	90.32	9.90
	2-DDR(25)	44.20	**92.57**	35.99	**96.13**	**92.97**	**7.23**	33.17	**92.43**	26.54	**96.06**	**92.89**	**7.24**
	2-DDR(50)	47.79	90.89	39.27	95.21	91.24	8.97	38.71	90.71	30.13	95.11	91.13	9.02
	2-DDR(100)	**50.39**	90.03	43.25	94.74	90.35	9.88	**40.16**	89.80	33.60	94.61	90.19	9.97
CSL	BF	0.00	**99.32**	0.00	**99.66**	**100.00**		0.00	**99.31**	0.00	**99.66**	**100.00**	
	2-DDR(25)	28.54	91.04	16.46	95.30	91.55	**8.50**	25.63	91.08	14.83	95.32	91.60	**8.44**
	2-DDR(50)	30.19	90.12	18.43	94.80	90.61	9.45	27.84	90.15	16.92	94.81	90.65	9.40
	2-DDR(100)	**31.72**	89.81	**19.39**	94.62	90.29	9.77	**28.19**	89.83	**17.38**	94.64	90.33	9.72
CSB	BF	66.67	**90.45**	**59.51**	**94.96**	**90.66**	**9.68**	38.39	**90.97**	**37.42**	**95.26**	**91.33**	**8.86**
	2-DDR(25)	66.11	90.24	49.20	94.85	90.52	9.75	39.53	90.44	29.89	94.97	90.86	9.29
	2-DDR(50)	67.36	90.21	51.35	94.83	90.48	9.81	39.68	90.48	31.16	94.99	90.89	9.27
	2-DDR(100)	**67.80**	90.06	52.16	94.75	90.32	9.97	**40.25**	90.28	31.86	94.87	90.68	9.48

Table 2.6: Mean results summary in the ASF data set considering the 5% and 10% PP restriction for the combination of all the approaches introduced throughout this chapter.

2.5.3 Sensitivity analysis

Undertaking a performance study for the proposed algorithm is fundamental to assessing its viability and practical utility. To better understand the effect of the parameter k and to justify its choice, a sensitivity analysis of k is presented. The results obtained for the ASF data set in terms of savings are summarized in Figure 2.5. The upper plot shows the direct relationship between k and the savings in the train sets. This is due to the additional flexibility. Nevertheless, the performance clearly reaches an "horizontal asymptote" for all models. In the test set (bottom graph in Figure 2.5), a larger value of k is needed for the simpler models, with a peak obtained mainly at $k = 50$ for all classifiers. This suggests that a larger value of k is not needed to get better results, as these are limited by the ranking provided by the classifier.

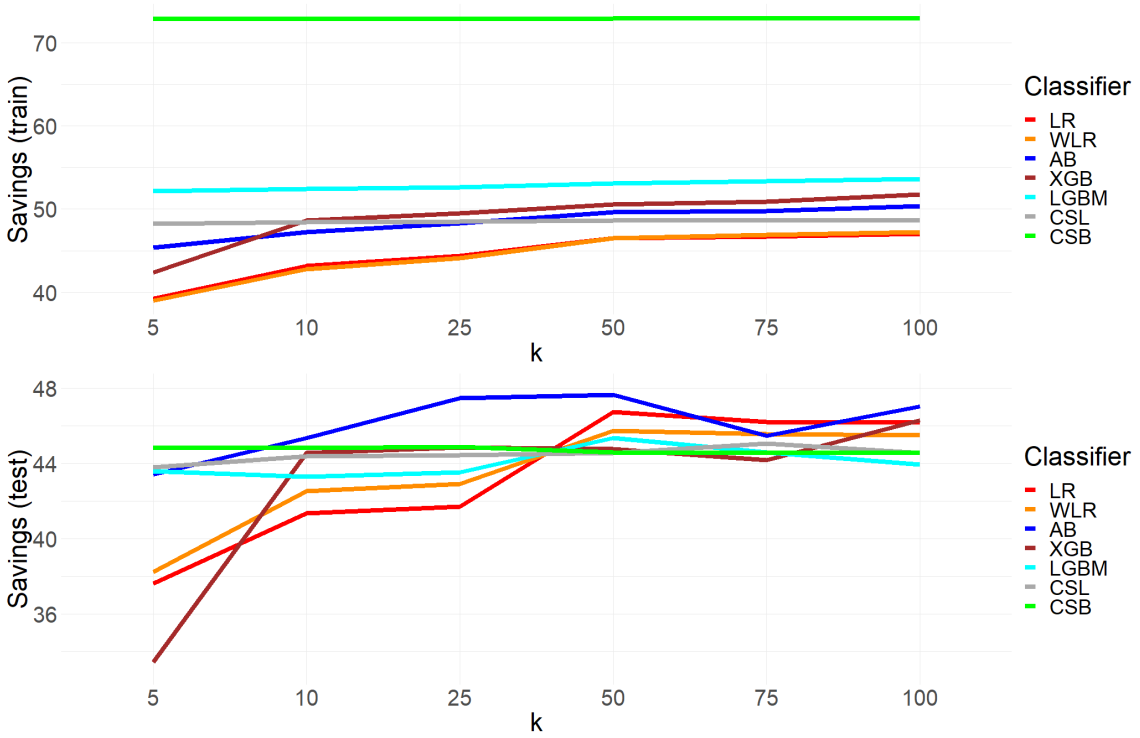

Figure 2.5: Mean savings obtained in the ASF data set for the combination of all the classifiers introduced throughout the chapter and the proposed 2-DDR(k) algorithm with different values of k for the train sets (upper graph) and test sets (bottom graph).

Figure 2.6 shows the time (minutes) required to run the algorithm for different

values of k. There is a quadratic correlation between the computation time and k. As a visual check, the curve $0.004x^2 - 0.798$ is represented by a dashed black line. The cost-sensitive classifiers and the LightGBM were faster, probably due to the good ranking of high-value fraud offered by these models, which facilitates the thresholding task.

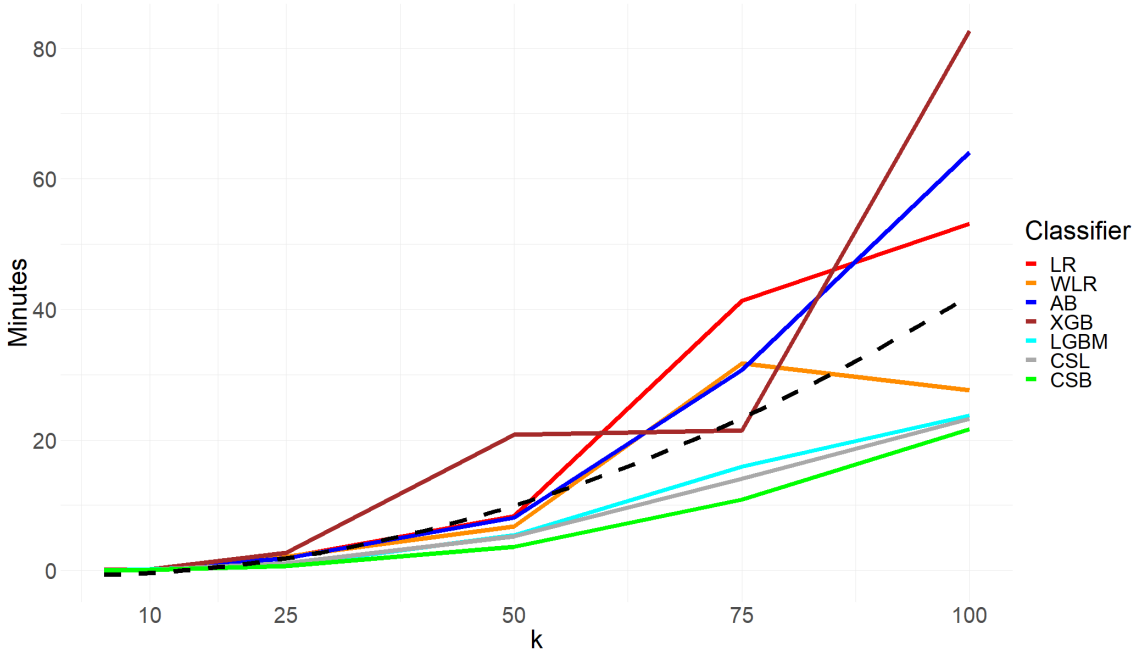

Figure 2.6: Mean computational times in minutes for the Algorithm 2-DDR(k) considering the classifiers introduced throughout this chapter fitted over the ASF data set. A quadratic curve is superimposed with a dashed black line.

2.6 Summary and conclusions

In this chapter it is introduced a new cost-sensitive thresholding method for reducing aggregate losses, the main concern in any business. Algorithm 2-DDR(k) yields the optimal classification rule given an estimated probability and a cost specification. In addition, it has the advantage that any PP constraint can be considered. Thus, it can be generalized to any cost-sensitive setting, with potential in other settings such as churn prediction, credit risk, medical diagnosis, or logistic planning.

Previous thresholding approaches are included in Algorithm 2-DDR(k) search,

so it extends and improves previous thresholding approaches, without the need for further estimation. Thus, a consistent improvement was expected. This has been verified with the results on two real fraud data sets, summarized in Tables 2.4 and 2.5. Although some thresholding approaches outperform the proposed methodology in terms of classification, given a classifier they are always beaten by the new proposal in terms of the objective function, savings in (1.2). This illustrates the contrast between minimizing costs or minimizing classification error during training, indicating that these are two different objectives. For each data set a different classifier was selected. Given the no-free-lunch theorem, there is never an overall winner and some experimentation will always be required to optimize performance.

Regarding the computational time, empirical results show that it depends mainly on k quadratically, as it controls the search grid size, and not on the sample size, making this approach suitable for scalability to larger datasets. Furthermore, it has been empirically verified that a larger value of k is not necessary to obtain satisfactory and stable results. In fact, using a large k can be counterproductive and lead to overfitting.

Other extensions can be considered. More dimensions could be introduced, as an additional default probability dimension to globally optimize the credit approval strategy. More complex loss functions could also be introduced considering more than one exogenous variable defining losses. These are just a few examples of possible extensions that can be made from the flexibility offered by the proposed method, which has shown satisfactory performance in amount-dependent problems.

The algorithm proposed in this chapter and all the results provided have been published in Information Sciences in C-Rella et al. (2024b).

Chapter 3

Flexible cost-sensitive thresholding for multi-class problems

3.1 Introduction

In this chapter, a new version of the Algorithm 2-DDR(k) is proposed considering a different search grid. In particular, the grid $G(k)$ in equation (2.8) is modified so that it is constructed based on the quantiles of the empirical distribution of the variables on which the decision rule is estimated. This is expected to speed up the search process when the distributions are skewed, since low-density regions of space are omitted. Likewise, since a finer search is performed, it is expected to obtain better results adapted to the data distribution structure.

In addition, the proposed algorithm is extended to other problems. ASF, satisfied with the results attained in Chapter 2, wanted to apply the algorithm to other problems. In this case, the multi-class classification problem was of interest in the credit rating stage. The problem is that the algorithm, as defined in the previous chapter, needed some adjustments to adapt to this multi-response context. Furthermore, to ensure that the algorithm would work properly, it is also developed an extensive simulation study to empirically verify its good performance. Theoretical

results are not explored because there is no universal result that guarantees the optimality of greedy algorithms such as the one proposed.

As introduced in the previous chapters, classification tasks are typically divided into two well-defined stages. In the first, a classifier returns an estimated probability or score that measures the likelihood of an event of interest occurring. The second step involves determining a prediction rule from the previously computed score to optimize a performance metric, such as accuracy. In credit risk, the second evaluation step performed by ASF, the goal is to distinguish creditworthy borrowers from those with a higher risk of default.

In general, in credit risk problems, as in most classification tasks, models prioritize accurately classifying observations. However, they ignore the various costs of misclassification errors. This can lead to suboptimal estimates that jeopardize both profitability and risk management, as discussed earlier in Chapter 1. For example, misclassifying a bad transaction doesn't have the same impact as misclassifying a good transaction, which depends on the amount of the credit.

Another layer of complexity is that in practical problems, the answer is not necessarily binary. In the creditworthiness evaluation step, banks tend to consider multiple responses depending on the risk of the transaction. The most common approach is a ternary label, where the transactions with the best profile are approved, those with the highest probability of default are rejected, and those with an "average" PD are sanctioned as a doubt to be decided by an expert agent. This is a very specific problem because it aims to provide a multi-label prediction (approve, doubt or reject) from a single score fitted to a binary response (PD). This setting is shown in Figure 3.1, where three labels can be predicted from the estimated PD.

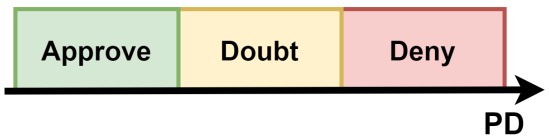

Figure 3.1: Ternary predictions (approve, doubt or deny) from the estimated probability of default (PD) for the credit risk problem.

To address the credit risk problem, a different misclassification cost specification is considered than that introduced in Chapter 1 for the fraud detection problem. In the same way, it addresses the prediction of a binary dependent variable $Y \in \{0, 1\}$ from a set of independent variables $\mathbf{X} = (X_1, \ldots, X_p)$, taking into account the costs associated with the different prediction errors and possibly other costs. For the credit risk problem, $Y = 0$ is assumed to be the correct payment of the debt and $Y = 1$ is the default event. The objective is to maximize the total benefit, which depends on the true class, Y, the amount of credit, W, and the prediction, $\hat{Y}$. The difference between the fraud detection problem and the credit risk problem lies in the available answers. For the credit risk problem, three possible predictions are considered, $\hat{Y} \in \{0, \frac{1}{2}, 1\}$, where 0 means that the operation is approved, $\frac{1}{2}$ labels it as doubtful, and 1 means that the credit request is rejected.

Traditional multi-label approaches assume that all errors of the same type have the same impact, which can be a dangerous oversimplification as introduced in Chapter 1. To account for the different costs of misclassification errors depending on the instance characteristics, we consider an instance-dependent cost matrix like the one introduced above in Table 3.1. This approach generalizes the CS binary classification problem introduced in Chapter 1 to the multi-response setting. It assumes a benefit of aW per legitimate transaction, where $W > 0$ is the loan amount and a is the interest earned. It assumes a loss of cW when granting an operation that ends in default, where c is the loss given default. We consider the reasonableness condition proposed in Elkan (2001), which enforces $c > a$. This implies that the impact of false negatives (granting a default operation) is greater than the impact of false positives (denying a legitimate operation). Finally, there is a fixed cost, b, of doubting an operation due to the cost of analysis, in line with Elkan (2001); Vanderschueren et al. (2022a) and Höppner et al. (2021). It is assumed that the expert analysis will correctly determine whether the customer will pay. Negative values in the cost matrix represent losses, positive values represent benefits and all constants are assumed to be positive.

	$\hat{y}_i = 0$	$\hat{y}_i = \frac{1}{2}$	$\hat{y}_i = 1$
$y_i = 0$	$C_i^{TN} = aw_i$	$C_i^{DN} = aw_i - b$	$C_i^{FP} = -aw_i$
$y_i = 1$	$C_i^{FN} = -cw_i$	$C_i^{DP} = -b$	$C_i^{TP} = 0$

Table 3.1: Amount-dependent cost matrix for the multi-class response credit risk problem.

In order to evaluate model performance, from Table 3.1, a *reward function* is defined as:

$$r(\hat{y}_i, w_i, y_i) = \begin{cases} C_i^{TN},\ \hat{y}_i = 0, y_i = 0 \\ C_i^{DN},\ \hat{y}_i = \frac{1}{2}, y_i = 0 \\ C_i^{FP},\ \hat{y}_i = 1, y_i = 0 \\ C_i^{FN},\ \hat{y}_i = 0, y_i = 1 \\ C_i^{DP},\ \hat{y}_i = \frac{1}{2}, y_i = 1 \\ C_i^{TP},\ \hat{y}_i = 1, y_i = 1 \end{cases} \tag{3.1}$$

where C_i^* are as given in Table 3.1.

The reward function in the equation (3.1) evaluates only a single instance. In order to have a metric to measure classification performance over an entire sample, we consider the *average expected reward* (AER), which extends the AEC proposed in Höppner et al. (2021) and Bahnsen et al. (2014a). For a sample $\{(\hat{y}_i, w_i, y_i)\}_{i=1}^n$, the AER is defined as follows:

$$\text{AER} = \frac{1}{n} \sum_{i=1}^{n} r(\hat{y}_i, w_i, y_i) \tag{3.2}$$

which serves as an estimate of the expected benefit per operation and facilitates comparison between different models because it is a standardized metric.

State-of-the-art multi-label classification techniques assume that there is a score for each possible response. Multi-label algorithms can be divided into two main approaches, one-vs-rest (OVR) and one-vs-one (OVO), as noted in Alotaibi et al. (2014). Both approaches extend the binary classification problem to the multi-label setting. The former is a heuristic method that splits the multi-class data set

into multiple binary classification problems of one class versus the rest, over which classifiers are trained. Predictions are made using the most confident label. For its part, OVO splits the problem into a data set for each class versus each other class. Similarly, the argmax of the sum of the scores (class with the largest sum score) is the predicted label. A potential drawback of these approaches is that they require a model for each class, which can be a problem for large datasets, slow models, or a very large number of classes.

Regarding multi-label CS thresholding, O'Brien et al. (2008); Collell et al. (2018); Alotaibi and Flach (2021) and Huang and Lin (2017) consider a different decision threshold for each class, allowing for multiple predicted labels. Bourke et al. (2008) and Zhou and Liu (2010) suggest reweighting the estimated scores to account for the different costs of misclassification. However, the IDCS multi-class classification problem has not yet been fully solved, as noted in Alotaibi et al. (2014) and Alotaibi and Flach (2021). Furthermore, to the best of our knowledge, there is no proposal that considers all classes together in a single decision process from a single score. Consequently, there is no alternative in the state of the art to address the credit risk problem as presented by ASF (see Figure 3.1). This creates a gap in the literature for a common problem in classification tasks such as the credit risk problem, as mentioned in C-Rella and Vilar (2024).

The proposed Algorithm 2-DDR(k) in Section 2.4 empirically estimates the optimal prediction in binary classification tasks from a given score. Its main advantage is the flexibility it provides to adapt the decision rule to any prediction constraint. However, this flexibility has not been fully exploited. In this chapter, Algorithm 2-DDR(k) is extended to obtain the optimal decision rule in multi-class problems from a single score. The proposed approach can be extended to any multi-class classification problem as well as to any CS problem.

Moreover, the Algorithm 2-DDR(k) as proposed in in Section 2.4 is further modified in order to explore the decision space more efficiently, taking into account the quantiles of the decision variables.

Another gap in the literature addressed in this chapter is the comparison of state-of-the-art thresholding techniques. Apart from the fact that the available approaches are narrow, their performance is only investigated on a limited number of datasets. This makes it difficult to understand their behavior and how they relate to each other, as noted in Vanderschueren et al. (2022b); Alotaibi and Flach (2021) and Lessmann et al. (2015). This chapter presents a simulation study that examines the performance of thresholding algorithms under different levels of class overlap, cost specification and positive class proportion. This provides valuable insights into the advantages and drawbacks of the different techniques available depending on the problem characteristics.

The chapter is organized as follows. Section 3.2 introduces the extension of the Algorithm 2-DDR(k) considering a quantile grid, and Section 3.3 extends the proposed algorithm to the multi-class setting. Section 3.4 investigates the performance of the different thresholding strategies on a wide range of simulations and four real data sets. Conclusions and future extensions are presented in Section 3.5.

3.2 Two-dimensional thresholding considering a quantile grid

The proposed Algorithm 2-DDR(k) empirically searches for the optimal classification rule over a two-dimensional space. This space is generated by the two variables of interest. For the credit risk problem, these are the estimated probability of default, $\hat{p}(\mathbf{x})$, and the credit amount, W, on which the losses depend. The algorithm greedily explores the decision space with complete freedom, iteratively making one optimal local choice after another. A global decision rule is obtained, on which any restrictions can be imposed. In this section, an extension of Algorithm 2-DDR(k) is presented that considers a different grid than the one introduced in Section 2.4 for the decision rule search, as introduced in C-Rella and Vilar (2024). The goal is to reduce the computational complexity and to obtain a finer decision strategy.

For the credit risk problem, one goal is to identify a region of the two-dimensional decision space where credit applications are automatically rejected, i.e. a region to which the prediction $\hat{Y} = 1$ is assigned. This region targets the least profitable operations, characterized by both a higher probability of default (PD) and a smaller loan amount, W.

The extension proposed in this chapter considers a search grid constructed from the quantiles of the decision variables. Given a sample $\{(\hat{p}(\mathbf{x}_i), w_i)\}_{i=1}^n$, a grid, $G^q(k)$, is defined considering the $1/k$ quantiles of each variable:

$$G^q(k) = (p_t^q, w_{t'}^q) \in \{q(\hat{p}, t/k)\}_{t=0}^k \times \{q(w, t'/k)\}_{t'=0}^k \tag{3.3}$$

where $q(x, p)$ is the p-quantile of the continuous variable $x \in \mathbb{R}$. By considering a quantile grid, it is expected to achieve better performance and a more efficient search by omitting regions of the space with few points and prioritizing regions with higher points density.

Motivated by the credit risk problem, the decision region (the subspace where instances are predicted to be cases) is defined as the union of quadrants of the form:

$$Q_{\mathbf{r}} = \left\{(\hat{p}, w) \in \mathbb{R}^2 \mid \hat{p} > r_1, w < r_2\right\}$$

where $\mathbf{r} = (r_1, r_2)$ characterizes the quadrant. For a set of points, $R = \{\mathbf{r}_j\}_{j=1}^m$, a decision region is constructed as the union of the associated $Q_{\mathbf{r}_j}$,

$$D(R) = \bigcup_{j=1}^{m} Q_{\mathbf{r}_j} \tag{3.4}$$

Given an instance $(\hat{p}_i, w_i)$, for a decision region $D(R)$ as in (3.4), it is predicted $\hat{y}_i^R = \mathbb{I}\left((\hat{p}_i, w_i) \in D(R)\right)$. Likewise, for a sample $\mathcal{X} = \{(\hat{p}_i, w_i, y_i)\}_{i=1}^n$, the AER (3.2) obtained with the decision region $D(R)$ in (3.4) is defined as:

$$AER(R \mid \mathcal{X}) = \frac{1}{n} \sum_{i=1}^{n} r(\hat{y}_i^R, w_i, y_i) \tag{3.5}$$

Extending Algorithm 2-DDR(k) to the grid $G^q(k)$ in equation (3.3), it is obtained Algorithm 2-$DDR_q(k)$ (2-dimensional decision region depending on the parameter

k over a quantile grid). The objective is to obtain a region of the two-dimensional decision space in which loan applications are rejected. The least profitable operations are those with a higher PD (greater risk of default) and those with smaller amount. Consequently, the algorithm starts in the south-easternmost region of the decision space to construct the rejection region. Recursively, each of the points of $G^q(k)$ surrounding the current decision region is added to the current region as in (3.4) and the AER is calculated as in (3.5). If the AER is increased, the decision region is updated with the point whose associated quadrant produces the largest AER. If there is no improvement, the next surrounding points of $G^q(k)$ are explored. This is iterated until the data support limit is reached. An example of the first steps is shown in Figure 3.2, where the cyan lines represent the newly explored regions and the black lines represent the estimated decision region at each step.

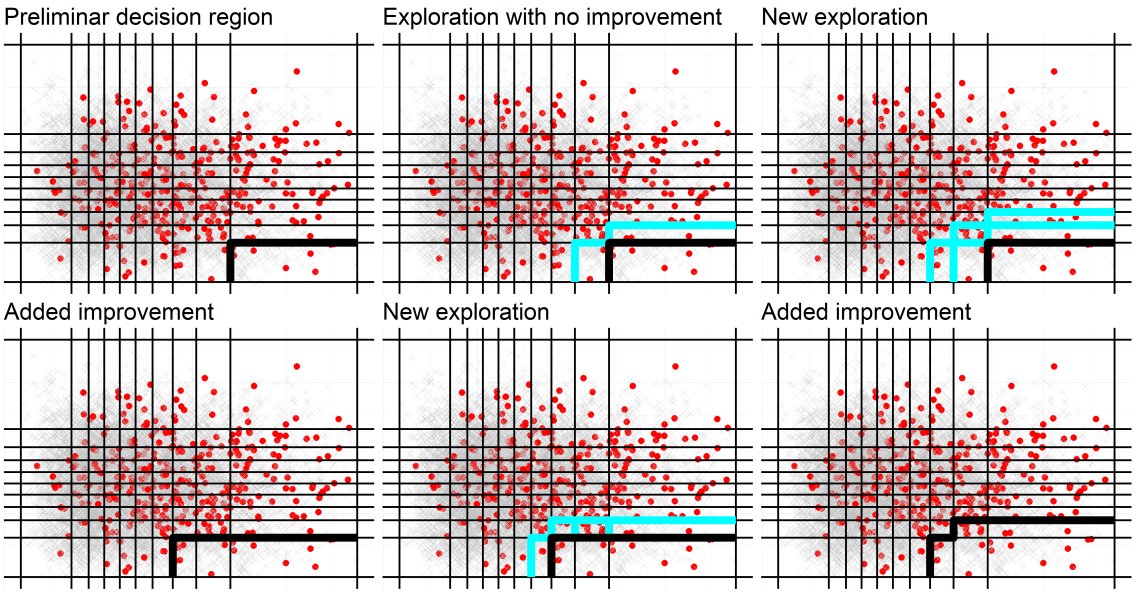

Figure 3.2: Algorithm 2-DDR representation along the grid $G^q(k)$ in (3.3) (black thin lines), explored decision regions (cyan lines) and the estimated decision region (black wide line).

Algorithm 2-$DDR_q(k)$ has all the advantages of the proposal in Section 2.4. These are the generalizability to any CS problem and the possibility to construct decision rules considering constraints. Likewise, the decision rule search is performed over the whole space. Thus, all previous thresholding proposals are included in the algorithm search, except for some roughness depending on the k parameter.

Consequently, Algorithm 2-$DDR_q(k)$ is expected to improve (or at least reproduce) state-of-the-art thresholding approaches in terms of AER.

Algorithm 2-DDRq(k) Two-dimensional decision region algorithm over a quantile grid

Data $\mathcal{X} = \{(\hat{p}_i, w_i, y_i)\}_{i=1}^{n}$
Input k parameter
Compute
  Grid $G^q(k)$ as defined in (3.3);
  $R := (p_k^q, w_0^q)$;
$F \leftarrow R$
while $(p_0^q, w_k^q) \notin F$ **do**
  $R_{old} \leftarrow R$
  $r \leftarrow AER(R_{old} \mid \mathcal{X})$, as defined in (3.5)
  $t \leftarrow 1$
  while $R = R_{old}$ **do**
    $F :=$ subset of $G^q(k)$ surrounding $D(R)$ at a step t
    $\mathbf{f}_m = \arg\max_{f \in F} \{AER(R \cup \{f\} \mid \mathcal{X})\}$
    if $AER(R \cup \{\mathbf{f}_m\} \mid \mathcal{X}) > r$ **then**
      $R \leftarrow R \cup \{\mathbf{f}_m\}$
    end if
    $t \leftarrow t + 1$
  end while
end while
Output A decision region as defined in (3.4)

3.3 Two-dimensional multi-class labeling algorithm

This section extends the previously introduced Algorithm 2-$DDR_q(k)$ to handle multi-class classification problems such as the credit risk problem presented in Section 3.1. The objective is to estimate the optimal CS strategy for credit applications

(approve, doubt, deny) based on a single score measuring the probability of default, as shown in Figure 3.1. Extending the flexibility of Algorithm 2-$DDR_q(k)$ to the multi-class setting is expected to yield good practical results as in Section 2.5.2. Furthermore, to the best of our knowledge, no previous approach addresses the multi-class classification problem considering a single score. Thus, the proposed approach fills a gap in multi-response problems such as the credit risk problem summarized in Figure 3.1.

The Algorithm 2-$DDR_q(k)$ is adapted to the multi-class problem by starting the algorithm search from different corners of the two-dimensional map, one for each of the possible responses as suggested in C-Rella and Vilar (2024). Then, each estimated decision region is then assigned to each of the classes. The proposed approach is shown in Figure 2.2. For the credit risk problem, the possible responses are approve, doubt and deny (see Figure 3.1). The algorithm is started from two different corners, one for approve and the other for deny operations. If the estimated decision regions do not intersect, the doubt region is the complementary of the union of the approve and deny regions. Conversely, if the decision regions intersect, the doubt region is assigned as the intersection. If both occur, the doubt region is the union of these regions. In this way, the decision regions for approving or rejecting an operation are obtained, and the regions where either decision is predicted to be optimal/suboptimal are decided by an expert.

The decision regions obtained for the ASF dataset introduced in Section 2.5.1 and the cost setting $a = 0.01$ are shown in Figure 3.3. The grid $G^q(k)$ in equation (3.3) is represented by black lines over the two-dimensional space of the score and the loan amount. Gray crosses represent good loans and red dots represent defaults. The approval region (green line) is estimated starting the algorithm in the upper left corner, which corresponds to the highest amount loans with lower estimated risk in terms of PD. These are the more profitable operations. The rejection region is estimated starting Algorithm 2-$DDR_q(k)$ from the opposite corner (lower right corner), which corresponds to those that the company is most interested in avoiding. The strip between the two decision regions creates an area where it does not compensate

to approve or reject an operation directly. Therefore, this area is marked as a zone of doubt for an expert to decide. Thus, the multi-class extension of Algorithm 2-$DDR_q(k)$ allows to handle multi-class CS classification problems considering a single score and the exogenous variable on which losses depend, maximizing the potential benefits without the need to estimate a score comparing each pair of classes.

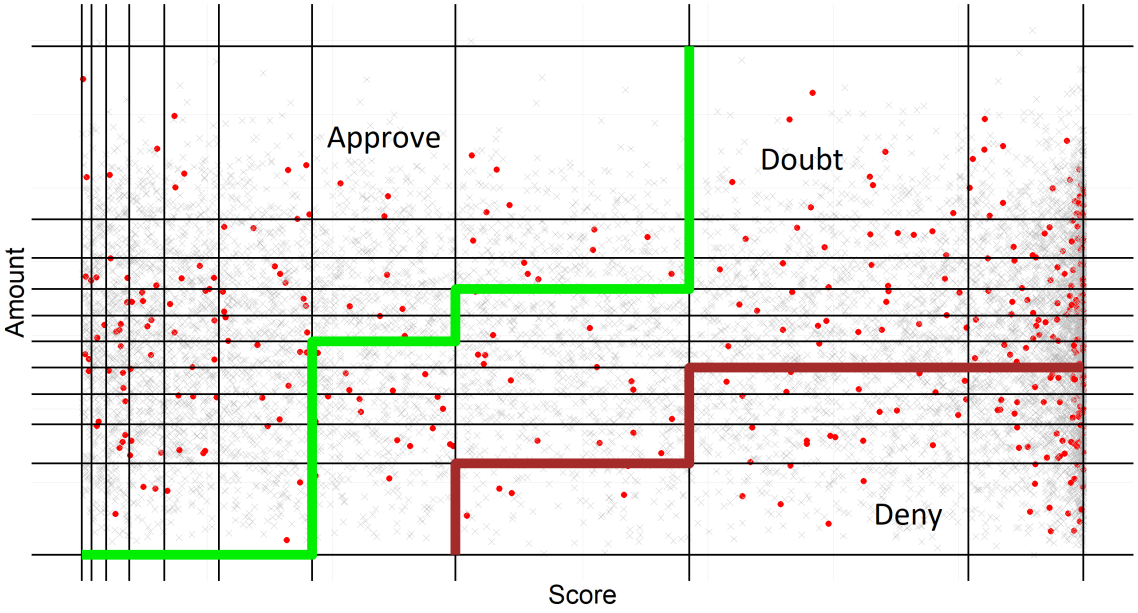

Figure 3.3: Algorithm 2-DDR representation for the multi-class classification problem. The region delimited by the green line represents the approval space, the red line the rejection space and the thin black lines the grid $G(k)$.

The approach in Figure 3.3 can be extended to more classes using the same reasoning. In addition, since Algorithm 2-$DDR_q(k)$ allows restrictions, it could be further extended to consider more than one loss function. For example, a stricter loss function could be considered to select the top 10% more profitable transactions, to select a portion of the portfolio to offer better terms, and then another loss function for the remaining loans. Similarly, the algorithm can be extended to consider more dimensions in the decision rule search by introducing more than one score or more exogenous variables. The different extensions that can be considered are innumerable thanks to the flexibility of Algorithm 2-$DDR_q(k)$ combined with the proposed approach.

3.4 Experimental results

This section evaluates the behavior of the thresholding algorithms presented throughout the work. The performance of the different thresholding approaches is evaluated considering four real credit risk datasets and an extensive set of simulations. It is considered the credit risk problem, where we face the task of estimating a multi-class response to a binary variable, as introduced in Section 3.1 and illustrated in Figure 3.1.

State-of-the-art multi-class classification techniques are designed to predict the optimal class given a score or probability that measures the likelihood of each response. In the problem presented, however, we only have a single score (the estimated PD). To the best of our knowledge, there is no other alternative to tackle multi-class classification from a single score. Therefore, to compare the proposed multi-class method in Section 3.3 with the state-of-the-art, the thresholding algorithms introduced in Section 2.3 and Section 2.4 are trained assuming a binary response. That is, only approving or rejecting an operation ($\hat{Y} = \{0, 1\}$).

In this study, we consider the Youden thresholding approach using equation (2.6) (Youden), the brute force approach introduced in Section 2.3.2 (Brute force), the Bayes minimum risk threshold in equation (2.7) (Bayes), and its fixed threshold version as introduced in Section 2.3.4 (Elkan). In addition, the 2-DDR algorithm is trained considering a regular grid as introduced in Algorithm $2 - DDR(k)$ and a quantile grid as proposed in Algorithm 2-$DDR_q(k)$ with $k \in \{5, 10, 25, 50\}$. For both algorithms, it is considered the binary problem as in Section 2.4 (2-DDR and 2-DDR_q, respectively) and a multiclass threshold as proposed in Section 3.3 (2-$DDRM$ and 2-$DDRM_q$, respectively). For the sake of brevity, the results are summarized for the optimal k value chosen in advance by cross-validation. A study of the effect of the parameter k on the performance of the algorithm has already been done in Chapter 2, which is expected to be valid for the multiclass case as well.

The parameters in the cost matrix in Table 3.1 are varied in order to explore different cost specification scenarios. The values of $b = 10$ (analysis cost) and $c = 1$

(loss given default) are fixed and $a \in \{0.01, 0.1, 0.33\}$ (benefit for a good loan). These values were suggested by ASF. The goal is to understand how the thresholding techniques behave as the proportion of cases, the dependency relationship and the cost specification vary, providing valuable insights into the performance of each technique. In addition, it is further investigated whether the fact of considering a multi-class approach can improve the results in credit risk problems.

3.4.1 Simulation study

For the simulation study, a two-dimensional space is constructed by simulating the two variables of interest for decision making, the score and the exogenous variable. Since the thresholding algorithms operate on a given score, we focus on simulating the score rather than estimating it from a set of covariates.

The response variable Y is simulated from a Bernoulli distribution with probability $P(Y = 1) = (1 + \exp(-S + U + \epsilon))^{-1}$, where $S \in \mathbb{R}$ and $\epsilon \sim N(0, \sigma)$ is an error term. The term $U \in \{-1, 0, 1\}$ is a bias variable introduced to assess the effect of under- or overestimating the estimated probability on classification performance. The variable S is simulated as a random variable $S \sim N(\mu, 1)$, with $\mu \in \{-2.5, -5\}$ in order to explore different case proportion scenarios. The corresponding case proportions are 10% (balanced scenario) and 1% (unbalanced scenario). We also consider two values for the variance of the error term, $\sigma^2 \in \{0.25, 16\}$. Increasing the value of the variance in the error term, imposes a greater degree of noise and overlap between the two classes, making the thresholding task more difficult.

The exogenous variable is assumed to be independent of the score, in line with Zadrozny and Elkan (2001). It is simulated as $W \sim 200\chi_2$, in order to replicate the asymmetry commonly found in credit risk problems. We also consider different sample sizes $n = \{1000, 10000, 100000\}$ for the training samples. The test samples, over which the classification performance is evaluated, are simulated with $n = 100000$. Thresholding algorithms are trained considering the estimated probability, $\hat{P}(Y = 1) = (1 + \exp(-S))^{-1}$, and the exogenous variable W.

To gain insight into the structure of the different simulation scenarios, Figure 3.4 shows the datasets generated for different values of σ and U. It shows the two-dimensional map generated by the estimate probability and the exogenous variable, W, log-scaled. Non-cases are represented by gray crosses, cases by red dots, and the curve levels for both classes are overlaid with the respective colors. When the bias is $U = \{-1, 1\}$, the score tends to either underestimate or overestimate the probabilities across the samples. In addition, increasing the standard deviation in the error term, σ, leads to increased overlap between classes, making the thresholding task more difficult.

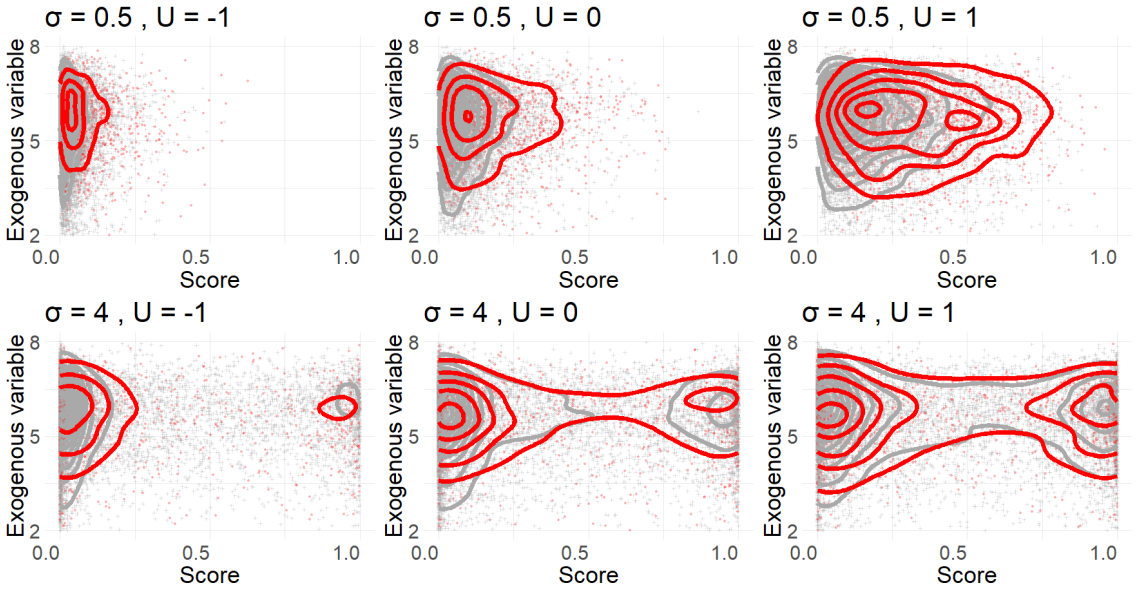

Figure 3.4: Simulated datasets for the different values of the standard deviation σ and bias U.

In this way, two case proportion scenarios, two class overlap scenarios, three score bias scenarios and three cost specifications are considered. Thus, the behavior of each of the thresholding algorithms is analyzed in a broad and in-depth manner.

All experiments were performed over 50 simulations for each simulation scenario. The results are reported by averaging the results of the test samples over these 50 runs. Model performance is evaluated by considering the AER (3.2), accuracy, recall $= TP/(TP + FN)$, precision ($PPV = TP/PP$), specificity $= TN/N$, proportion of observations labeled as case (PP), and F-score (FS). Thus, model performance is evaluated using both cost-sensitive and classification metrics. Boxplots of the

obtained AER are shown in Figures 3.5-3.7 for the training sample size $n = 100,000$. For brevity, the metrics introduced are summarized in detail for all scenarios in the appendix in Tables 3.5-3.13.

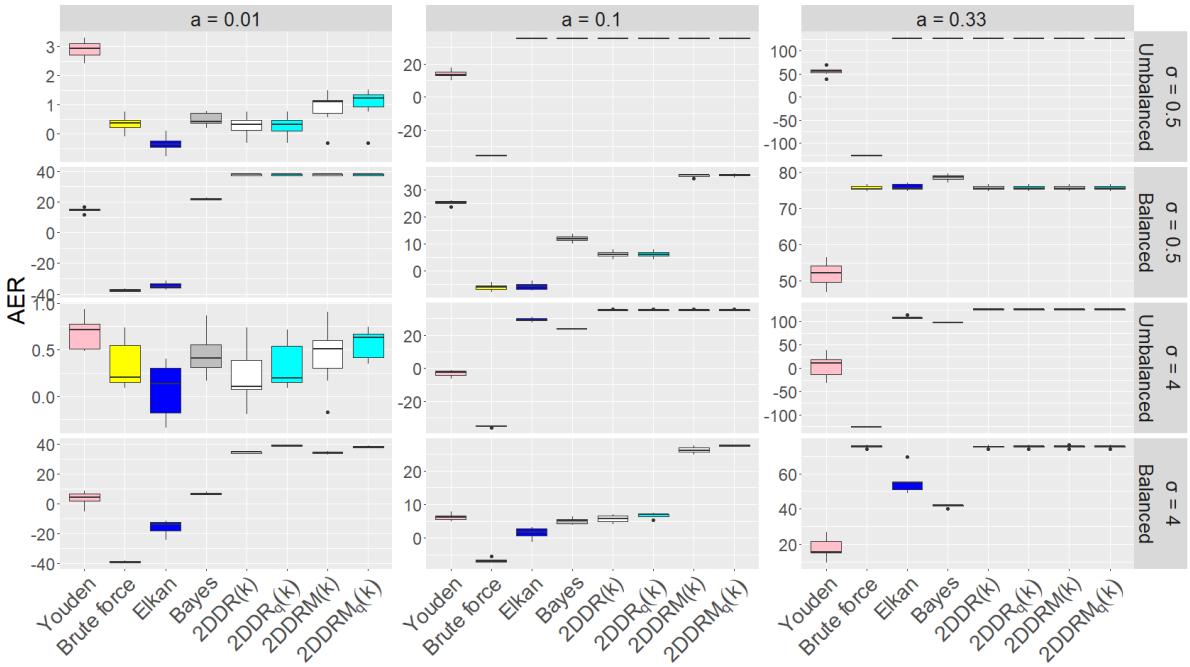

Figure 3.5: Simulation results for the scenario with bias $U = -1$ and $n = 100000$.

The first observation regarding the results is the poor performance of the Youden statistic approach (2.6), the only cost-insensitive alternative considered. As expected, it is concluded that the use of a cost-sensitive approach yields better results in scenarios where the impact of misclassification errors varies. Among the cost-sensitive alternatives, Bayes' approach (2.7) consistently outperforms Elkan's approach as expected. This is due to the nuanced classification resulting from individual decision rules for each point with the Bayes minimum risk threshold (2.7).

The brute force approach achieves satisfactory results despite considering only the score for decision rule estimation. Its empirical nature allows it to outperform the Elkan and Bayes approaches, especially in biased scenarios, where it excels by considering only the score rather than the estimated probability. When this empirical search is extended by introducing the exogenous variable in Algorithm 2-$DDR_q(k)$, it consistently outperforms previous alternatives across all simulations

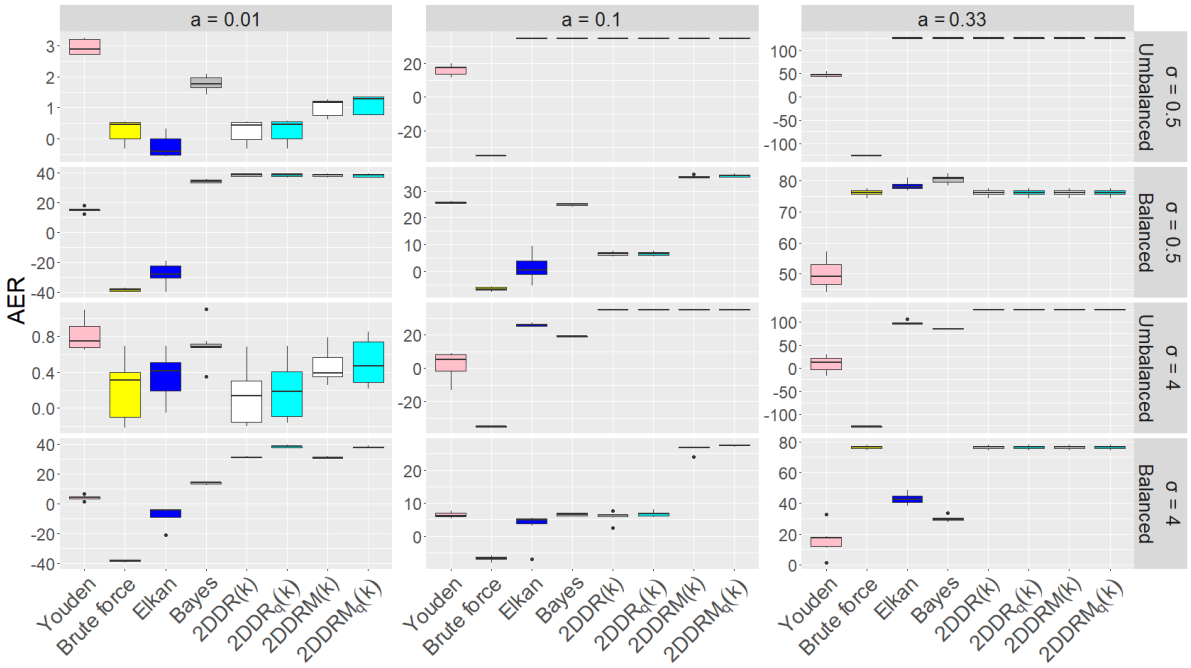

Figure 3.6: Simulation results for the scenario with bias $U = 0$ and $n = 100000$.

and cost scenarios. These conclusions hold for all sample sizes analyzed in the simulation study, as can be seen in Tables 3.5-3.13. Therefore, it can be concluded that the proposed algorithm does not require much data to obtain a satisfactory classification. Nevertheless, as the sample size increases, the empirical nature of the 2-DDR algorithm allows for a more informed estimation of the decision region, leading to improved results, as can be seen in Tables 3.5-3.13.

The comparison between using a uniform grid (algorithm 2-DDR(k)) and using quantiles (algorithm 2-$DDR_q(k)$) favors the latter due to its finer grid construction adapted to the data distribution, leading to equal or better results in all scenarios, as can be seen in Figures 3.5-3.7. This is true when comparing the results considering both a binary and a multinomial response.

Regarding the multi-class approaches (2-$DDRM$ and 2-$DDRM_q$), in all simulations, they obtain results equal to or better than the alternatives considering a binary response, except in the umbalanced scenario in which the potential benefit is more limited ($a = 0.01$), where the algorithm seems to fall into some overfitting. In some scenarios, the improvements are very significant, thanks to having more

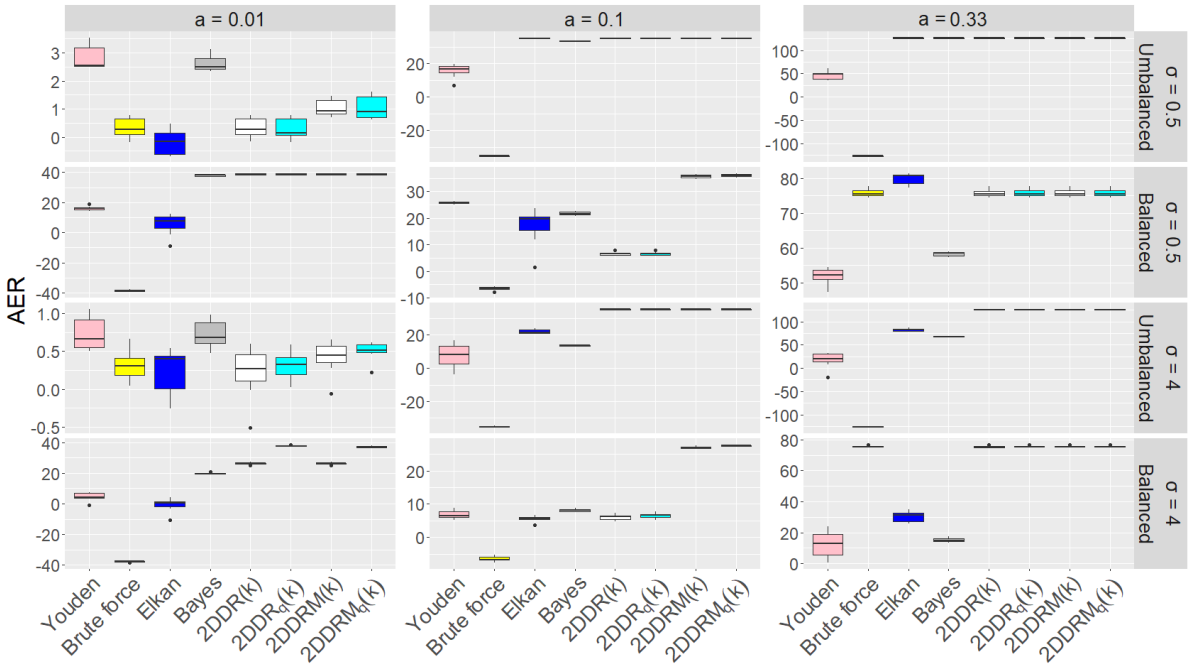

Figure 3.7: Simulation results for the scenario with bias $U = 1$ and $n = 100000$.

alternatives to classify the observations, which allows them to further maximize the AER considering the risk and the amount of the operations. This is especially seen in the scenarios with more noise ($\sigma = 4$), which are more complicated and require a more subtle classification to obtain good results.

In this way, we first conclude that the best performance when considering a binary response is obtained whit Algorithm 2-$DDR_q(k)$. Second, we conclude that considering a multi-class response leads to better cost-sensitive results. This is achieved with the proposal made in Section 3.3, which to the best of our knowledge is the only one available to solve this problem. Therefore, it is concluded that the Algorithm 2-$DDR_q(k)$ is the best proposal to solve CS classification given a score, both considering a binary or a multi-class response.

3.4.2 Real datasets study

The first one (ASF data set) is a data set lend by ASF, consisting of 130, 482 loan requests collected between January 2021 and December 2022 with a default rate of

2.06%. The number of registers of the original data set is truncated in order to modify the default rate and thus preserve confidentiality. It consists of 14 variables containing information about the client and the operation. The second one (GCD data set[1]) is a real credit risk data set of size $n = 1,000$ with a case proportion of 30% and 24 variables available to model the default indicator. The third one (GSC data set[2]) consists of $112,398$ instances with 11 covariates and a case proportion of 5.75%. The fourth one (PAKDD data set[3]) consists of $11,111$ potential customers to be selected for a loan, for which 51 variables are available to model their PD. There are 3047 (27.4%) defaults in the sample. For this data set, the loan amount is simulated as $W \sim 100\chi_3$ in order to replicate the asymmetry commonly encountered in credit risk problems. These datasets are chosen in order to study the effect of different proportions of cases, sample size, and number of variables used to model the score on the thresholding performance.

For all datasets, the probability of default (PD) is estimated using a logistic regression model trained with a stepwise approach. Other classifiers are not explored, as the goal is to evaluate the classification performance of the different thresholding algorithms given a score. To evaluate the classification performance, standard metrics such as accuracy, recall, specificity and F-score are used together with the objective function, AER as defined in equation (3.2). The obtained results are summarized in Tables 3.2-3.4.

Looking at the results in Tables 3.2-3.4, the first notable observation is the variance in the average results depending on the data set. This variance is due to differences in the proportion of cases and the amount distribution in the different datasets. It should also be emphasized that a difference of just one euro in the AER (3.2) can translate into millions of euros when scaled up to larger datasets. Therefore, even small improvements can have a significant impact on practice.

[1] `https://github.com/JLZml/Credit-Scoring-Data-Sets/tree/master/1.%20UCI%20Repository/German`

[2] `https://www.kaggle.com/datasets/brycecf/give-me-some-credit-dataset`

[3] `https://github.com/JLZml/Credit-Scoring-Data-Sets/tree/master/2.%20PAKDD%202009%20Data%20Mining%20Competition`

Dataset	Threshold	AER	Acc	Rec	Spec	PP	FS
Bank	Youden	14.89	63.54	63.01	63.55	36.86	5.06
	Brute force	14.82	51.06	75.30	50.68	49.72	4.53
	Elkan	-5.80	92.14	1.69	93.56	6.37	0.66
	Bayes	-4.29	90.23	4.70	91.57	8.37	1.46
	2-DDR	15.42	47.16	78.55	46.66	53.72	4.38
	2-DDRq	14.77	45.30	80.00	44.76	55.62	4.32
	2-DDRM	17.38	45.05	70.00	37.36	55.01	4.44
	2-DDRMq	16.86	36.04	78.07	29.67	62.81	4.26
GCD	Youden	-3.51	75.60	54.55	83.15	26.80	54.14
	Brute force	26.29	36.80	98.48	14.67	88.80	45.14
	Elkan	-22.94	58.00	4.55	77.17	18.00	5.41
	Bayes	-28.39	66.00	0.00	89.67	7.60	
	2-DDR	28.22	26.40	100.00	0.00	100.00	41.77
	2-DDRq	28.22	26.40	100.00	0.00	100.00	41.77
	2-DDRM	22.63	27.93	93.94	0.00	94.40	43.06
	2-DDRMq	23.67	33.88	93.94	0.00	86.60	49.80
GSC	Youden	20.83	84.63	73.61	18.09	35.54	
	Brute force	26.07	20.29	96.47	15.64	85.06	12.23
	Elkan	-23.39	24.24	20.26	24.49	72.33	2.99
	Bayes	-26.16	37.24	13.51	38.69	58.56	2.42
	2-DDR	26.12	5.77	100.00	0.01	99.99	10.89
	2-DDRq	26.23	15.55	97.89	10.52	89.96	11.78
	2-DDRM	26.13	5.80	100.00	0.05	99.95	10.89
	2-DDRMq	26.23	15.55	97.89	10.52	89.96	11.78
PAKDD	Youden	6.14	57.08	55.40	57.80	46.13	43.43
	Brute force	54.58	29.98	100.00	0.34	99.76	45.94
	Elkan	-54.58	70.16	0.00	99.87	0.09	
	Bayes	-54.39	68.56	3.65	96.03	3.87	6.46
	2-DDR	54.62	30.12	100.00	0.54	99.62	45.99
	2-DDRq	54.58	29.79	100.00	0.07	99.95	45.87
	2-DDRM	54.62	30.08	100.00	0.47	99.67	45.97
	2-DDRMq	54.40	29.86	99.84	0.07	99.76	45.93

Table 3.2: Results for the real data sets for the multi-class classification problem for the setting $a = 0.01$ considering the different thresholding algorithms introduced throughout the work.

Dataset	Threshold	AER	Acc	Rec	Spec	PP	FS
Bank	Youden	59.14	63.54	63.01	63.55	36.86	5.06
	Brute force	202.15	98.41	0.24	99.95	0.05	0.47
	Elkan	177.47	92.14	1.69	93.56	6.37	0.66
	Bayes	193.55	90.23	4.70	91.57	8.37	1.46
	2-DDR	202.12	98.37	0.48	99.91	0.10	0.91
	2-DDRq	202.47	98.39	0.24	99.93	0.08	0.46
	2-DDRM	202.47	98.45	0.24	99.91	0.06	0.48
	2-DDRMq	202.54	98.46	0.00	99.93	0.04	
GCD	Youden	9.00	75.60	54.55	83.15	26.80	54.14
	Brute force	17.87	43.60	95.45	25.00	80.40	47.19
	Elkan	-17.03	58.00	4.55	77.17	18.00	5.41
	Bayes	-14.40	66.00	0.00	89.67	7.60	
	2-DDR	20.46	51.20	92.42	36.41	71.20	50.00
	2-DDRq	21.41	50.00	93.94	34.24	73.20	49.80
	2-DDRM	20.87	57.30	80.30	26.63	64.40	55.50
	2-DDRMq	22.15	59.49	80.30	34.24	62.20	54.64
GSC	Youden	33.22	84.63	73.61	85.30	18.09	35.54
	Brute force	33.56	88.06	68.03	89.28	14.02	39.62
	Elkan	-29.51	24.24	20.26	24.49	72.33	2.99
	Bayes	-11.48	37.24	13.51	38.69	58.56	2.42
	2-DDR	33.49	89.80	64.19	91.37	11.83	42.02
	2-DDRq	33.74	88.31	67.72	89.57	13.73	40.01
	2-DDRM	32.88	70.22	60.41	68.97	31.92	19.09
	2-DDRMq	33.60	65.33	63.26	63.00	37.25	17.66
PAKDD	Youden	9.18	57.08	55.40	57.80	46.13	43.43
	Brute force	36.02	32.48	98.10	4.70	96.13	46.36
	Elkan	-35.43	70.16	0.00	99.87	0.09	
	Bayes	-35.34	68.56	3.65	96.03	3.87	6.46
	2-DDR	35.99	30.26	100.00	0.74	99.48	46.04
	2-DDRq	36.29	31.49	98.73	3.02	97.50	46.16
	2-DDRM	33.23	30.84	94.60	1.28	96.03	45.81
	2-DDRMq	35.80	30.88	92.38	0.00	94.50	46.28

Table 3.3: Results for the real data sets for the multi-class classification problem for the setting $a = 0.10$ considering the different thresholding algorithms introduced throughout the work.

Dataset	Threshold	AER	Acc	Rec	Spec	PP	FS
Bank	Youden	172.24	63.54	63.01	63.55	36.86	5.06
	Brute force	729.19	98.44	0.12	99.98	0.02	0.24
	Elkan	645.82	92.14	1.69	93.56	6.37	0.66
	Bayes	699.15	90.23	4.70	91.57	8.37	1.46
	2-DDR	729.25	98.45	0.12	99.99	0.01	0.24
	2-DDRq	726.87	96.80	3.13	98.26	1.76	2.93
	2-DDRM	729.71	98.46	0.00	99.99	0.01	
	2-DDRMq	729.61	98.48	0.00	98.26	0.88	
GCD	Youden	40.96	75.60	54.55	83.15	26.80	54.14
	Brute force	41.29	77.20	51.52	86.41	23.60	54.40
	Elkan	-1.93	58.00	4.55	77.17	18.00	5.41
	Bayes	21.34	66.00	0.00	89.67	7.60	
	2-DDR	42.03	71.20	75.76	69.57	42.40	58.14
	2-DDRq	43.57	75.20	62.12	79.89	31.20	56.94
	2-DDRM	46.24	82.05	48.48	69.57	31.40	54.70
	2-DDRMq	46.21	81.65	46.97	79.89	24.80	55.36
GSC	Youden	64.87	84.63	73.61	85.30	18.09	35.54
	Brute force	86.13	93.98	38.17	97.39	4.66	42.19
	Elkan	-45.15	24.24	20.26	24.49	72.33	2.99
	Bayes	26.04	37.24	13.51	38.69	58.56	2.42
	2-DDR	87.13	93.74	42.57	96.87	5.40	43.93
	2-DDRq	85.11	94.22	33.21	97.95	3.85	39.82
	2-DDRM	88.80	95.40	23.85	96.87	3.97	33.15
	2-DDRMq	90.43	86.21	2.91	82.70	15.06	2.08
PAKDD	Youden	16.93	57.08	55.40	57.80	46.13	43.43
	Brute force	14.05	66.29	17.46	86.96	14.35	23.55
	Elkan	13.50	70.16	0.00	99.87	0.09	
	Bayes	13.36	68.56	3.65	96.03	3.87	6.46
	2-DDR	13.45	63.64	26.51	79.37	22.38	30.25
	2-DDRq	14.87	65.39	23.65	83.06	18.93	28.90
	2-DDRM	23.54	70.40	7.94	77.35	15.20	13.40
	2-DDRMq	23.88	71.78	0.00	82.06	9.84	

Table 3.4: Results for the real data sets for the multi-class classification problem for the setting $a = 0.33$ considering the different thresholding algorithms introduced throughout the work.

It is striking that the decision rule defined by the Youden's J statistic in equation (2.6) often outperforms the Elkan and Bayes proposals in many scenarios. This phenomenon occurs because in real data, the estimated probabilities are not necessarily well calibrated, which affects the performance of these proposals, as shown

in Tables 3.2-3.4, and as previously noted in Höppner et al. (2021), among others. It is also worth noting the good performance of the brute force algorithm as in the simulation study. Using an empirical rule based on point order rather than on calibrated probabilities, results in a decision rule that generalizes well and produces good results.

The multi-class approach considering Algorithm 2-$DDR_q(k)$ proposed in Section 3.3 (2-$DDRM_q(k)$) consistently achieves the best results across all data sets and cost scenarios as can be seen in Tables 3.2-3.4. This is consistent with the results of the simulation study.

Two conclusions can be drawn from these results. First, considering a multi-class response in problems such as credit risk helps to achieve better results. Second, the proposed multi-class approach yields the optimal decision rule for multiple responses given a score, outperforming all previous alternatives and solving a problem that has not been addressed before.

3.5 Summary and conclusions

A new methodology to optimize decision making in cost-sensitive classification problems is proposed. Extending Algorithm $2 - DDR(k)$ proposed in Section 2.4, an optimal decision rule is searched more efficiently over a two-dimensional space generated by a positive class probability estimate and the exogenous variable on which misclassification costs depend. Algorithm 2-$DDR_q(k)$ explores the decision space with complete freedom to optimize a given reward function. As a result, it achieves better results than any previous approach, regardless of the difficulty of the problem, both in terms of class overlap and the proportion of positive cases.

An extension of the algorithm to allow for multi-class responses given a single score, a previously unexplored challenge, is also proposed. It yields a flexible and satisfactory solution, as evidenced by the results in the simulation study (Figures 3.5-3.7) and in the real datasets study (Tables 3.2-3.4). This results also suggest that

addressing the credit risk problem with a multi-response approach yields significantly improved results compared to the binary response approach traditionally used.

The proposed technique can be extended to several other problems. It is enough to tune the reward function in equation (3.1) and the decision rule search to extend the proposed approach to any CS problem. The flexibility of Algorithm 2-$DDR_q(k)$ for the decision region search in the two-dimensional decision space, in conjunction with the proposed multi-class approach, makes the proposed technique a satisfactory methodology for multi-class classification problems. Moreover, it can be further extended to consider more classes or various loss functions. Given the good results obtained and the number of possible extensions, it is concluded that the proposed technique is a promising proposal for addressing practical challenges such as credit risk.

Appendix

			Umbalanced							Balanced						
σ	a	Thresholding	AER	Acc	Rec	PPV	Spec	PP	FS	AER	Acc	Rec	PPV	Spec	PP	FS
0.5	0.01	Youden	2.47	57.70	71.87	2.14	57.55	42.77	4.10	13.56	67.49	64.07	19.37	67.87	35.50	29.52
		Brute force	0.42	1.17	100.00	1.08	0.10	99.91	2.13	-38.02	89.47	1.04	62.91	99.89	0.21	1.99
		Elkan	-0.41	98.93	0.00		100.00	0.00		-35.14	89.40	4.41	48.72	99.42	0.99	7.95
		Bayes	0.39	98.08	5.57	6.21	99.09	0.96	5.87	23.19	67.94	55.72	17.67	69.38	33.27	26.83
		2-DDR	-0.25	76.51	21.72	0.99	77.10	22.89	1.89	37.62	10.69	98.17	10.41	0.37	99.48	18.83
		2-DDRq	-0.19	81.59	17.17	0.99	82.28	17.71	1.89	38.29	11.83	96.17	10.36	1.88	97.92	18.71
		2-DDRM	-0.19	82.10	16.10	1.15	78.84	16.43	1.83	34.87	10.69	96.74	10.42	0.36	97.98	18.81
		2-DDRMq	0.08	86.72	14.01	2.04	82.28	10.91	2.26	36.21	16.78	89.85	10.86	7.13	87.40	19.37
	0.10	Youden	5.38	54.94	68.95	2.13	54.79	45.47	4.06	24.82	65.02	67.53	18.58	64.72	38.66	29.01
		Brute force	-35.07	1.24	100.00	1.10	0.15	99.86	2.17	-5.56	89.51	0.69	57.75	99.92	0.14	1.34
		Elkan	35.18	98.91	0.00		100.00	0.00		-4.26	89.50	2.57	51.38	99.69	0.54	4.78
		Bayes	35.19	98.89	0.12	9.40	99.99	0.01	0.48	11.63	86.57	21.17	30.13	94.24	7.38	24.87
		2-DDR	34.56	98.74	0.07	0.49	99.84	0.16	0.59	4.03	21.46	84.78	10.39	14.04	85.84	17.52
		2-DDRq	35.18	98.91	0.00		100.00	0.00		4.78	30.54	73.99	10.44	25.46	74.48	18.05
		2-DDRM	35.18	98.90	0.00		99.84	0.00		28.74	35.74	36.95	23.64	14.10	20.27	24.16
		2-DDRMq	35.18	98.91	0.00		100.00	0.00		32.12	49.50	47.61	18.04	26.13	28.72	24.97
	0.33	Youden	44.37	66.41	65.00	2.26	66.42	33.92	4.34	50.88	65.68	66.82	19.00	65.55	37.87	29.37
		Brute force	-125.89	1.20	100.00	1.09	0.12	99.88	2.15	76.59	89.46	0.75	63.36	99.93	0.14	1.46
		Elkan	126.22	98.91	0.00		100.00	0.00		76.54	89.46	0.56	66.91	99.96	0.10	1.10
		Bayes	126.22	98.91	0.00		100.00	0.00		78.73	89.34	4.41	45.11	99.37	1.03	8.03
		2-DDR	126.22	98.91	0.00		100.00	0.00		76.19	89.44	0.00		100.00	0.00	
		2-DDRq	126.22	98.91	0.00		100.00	0.00		76.19	89.44	0.00		100.00	0.00	
		2-DDRM	126.22	98.91	0.00		100.00	0.00		76.45	89.44	0.00	100.00	99.87	0.00	0.04
		2-DDRMq	126.22	98.91	0.00		100.00	0.00		76.19	89.44	0.00		100.00	0.00	
4	0.01	Youden	0.71	47.16	60.86	1.29	47.00	53.09	2.50	3.36	53.92	54.19	12.34	53.89	46.95	19.58
		Brute force	0.41	2.18	99.21	1.11	1.10	98.90	2.19	-37.00	88.67	1.93	18.11	98.82	1.26	2.63
		Elkan	0.05	89.48	14.72	1.67	90.31	9.74	3.03	-13.13	70.08	32.90	13.09	74.43	26.34	18.72
		Bayes	0.56	77.41	29.45	1.47	77.95	22.13	2.81	6.63	57.37	49.87	12.27	58.25	42.60	19.69
		2-DDR	-0.16	50.20	46.84	1.03	50.24	49.73	1.95	37.93	12.42	96.24	10.37	2.61	97.27	18.72
		2-DDRq	-0.18	71.79	26.20	1.00	72.31	27.68	1.81	38.21	12.20	97.07	10.41	2.27	97.66	18.81
		2-DDRM	-0.01	52.30	41.81	1.04	50.02	44.38	1.96	35.15	12.49	94.61	10.37	2.65	95.62	18.69
		2-DDRMq	-0.14	79.79	16.63	1.66	74.35	17.18	1.56	35.41	14.91	91.55	10.48	5.42	91.55	18.80
	0.10	Youden	3.84	54.13	56.31	1.40	54.10	46.02	2.71	5.15	59.01	47.20	12.54	60.37	40.43	19.46
		Brute force	-34.52	1.88	99.66	1.11	0.79	99.22	2.19	-6.83	89.33	0.15	28.71	99.93	0.08	0.32
		Elkan	28.72	89.49	15.25	1.72	90.31	9.75	3.09	1.48	72.38	28.84	13.26	77.55	23.13	18.12
		Bayes	23.78	84.43	21.82	1.60	85.13	14.95	2.99	4.74	64.95	39.43	12.77	67.98	32.81	19.29
		2-DDR	35.10	98.90	0.00		100.00	0.00		0.97	51.03	47.08	10.28	51.52	48.33	14.54
		2-DDRq	35.10	98.90	0.00		100.00	0.00		3.08	48.12	51.67	10.38	47.66	52.27	16.44
		2-DDRM	35.10	98.90	0.00		100.00	0.00		15.48	70.78	11.81	13.06	51.46	10.23	10.44
		2-DDRMq	35.10	98.90	0.00		100.00	0.00		20.07	69.25	13.73	12.53	47.70	12.30	12.85
	0.33	Youden	-10.07	46.08	62.16	1.32	45.89	54.20	2.58	10.07	52.65	55.58	12.44	52.34	48.50	19.79
		Brute force	-114.88	5.52	97.19	1.15	4.48	95.54	2.26	75.04	89.16	0.61	21.05	99.66	0.37	1.12
		Elkan	108.47	91.96	10.61	1.66	92.89	7.15	2.85	53.44	75.97	23.81	13.70	82.15	18.48	17.28
		Bayes	98.22	88.93	14.41	1.58	89.78	10.27	2.84	41.71	70.84	31.75	13.30	75.47	25.29	18.75
		2-DDR	126.19	98.88	0.00		100.00	0.00		74.90	89.37	0.05	12.69	99.96	0.04	0.91
		2-DDRq	126.19	98.88	0.00		100.00	0.00		75.23	89.40	0.00		100.00	0.00	
		2-DDRM	126.19	98.88	0.00		100.00	0.00		75.90	89.40	0.00		99.72	0.00	
		2-DDRMq	126.19	98.88	0.00		100.00	0.00		75.50	89.39	0.00		99.64	0.00	

Table 3.5: Results for the simulated multi-class classification problems for the sample size $n = 1,000$ and bias $U = -1$ for the different thresholding algorithms introduced throughout the work.

			Umbalanced							Balanced						
σ	a	Thresholding	AER	Acc	Rec	PPV	Spec	PP	FS	AER	Acc	Rec	PPV	Spec	PP	FS
0.5	0.01	Youden	2.81	60.91	71.14	2.05	60.80	39.55	3.97	14.00	66.41	65.23	19.25	66.54	36.81	29.33
		Brute force	0.47	1.17	99.98	1.09	0.09	99.91	2.15	-38.00	89.46	0.54	68.75	99.97	0.09	1.07
		Elkan	-0.46	98.91	0.07	25.00	100.00	0.00	0.48	-23.62	87.38	18.12	38.57	95.57	5.88	21.89
		Bayes	1.65	93.36	19.79	3.60	94.17	5.99	6.09	34.50	46.21	77.36	13.72	42.53	59.57	23.31
		2-DDR	-0.06	41.29	55.10	1.04	41.14	58.82	2.02	37.53	10.68	97.87	10.40	0.38	99.44	18.80
		2-DDRq	0.16	53.77	45.47	1.07	53.86	46.13	2.08	37.89	12.04	96.08	10.39	2.11	97.70	18.75
		2-DDRM	0.07	43.74	46.45	1.07	41.14	49.19	2.04	35.37	10.76	96.71	10.49	0.37	97.51	18.92
		2-DDRMq	0.35	64.24	31.62	1.59	55.38	28.79	2.49	35.59	16.84	89.74	10.97	7.03	86.49	19.55
	0.10	Youden	9.59	59.93	72.12	2.19	59.80	40.55	4.20	24.17	65.69	65.39	18.97	65.73	37.56	28.98
		Brute force	-34.95	1.52	99.96	1.09	0.45	99.56	2.15	-5.62	89.45	1.28	59.80	99.86	0.26	2.42
		Elkan	35.30	98.91	0.09	25.00	100.00	0.00	0.37	1.11	88.60	10.97	45.36	97.75	3.17	17.22
		Bayes	35.23	98.66	2.78	9.49	99.71	0.31	4.29	24.14	73.16	52.13	20.16	75.65	27.29	29.07
		2-DDR	35.30	98.92	0.00		100.00	0.00		2.44	34.95	67.65	10.58	31.12	68.75	16.35
		2-DDRq	35.30	98.92	0.00		100.00	0.00		2.86	41.15	60.58	10.46	38.87	61.07	18.21
		2-DDRM	35.30	98.92	0.00		100.00	0.00		23.83	45.14	36.20	21.84	31.13	21.00	23.07
		2-DDRMq	35.30	98.92	0.00		100.00	0.00		24.90	59.71	35.41	19.70	40.87	20.73	25.22
	0.33	Youden	56.90	71.25	58.12	2.54	71.38	28.93	4.75	51.47	65.98	66.51	18.90	65.92	37.49	29.30
		Brute force	-123.98	1.99	99.92	1.07	0.93	99.08	2.13	76.54	89.49	0.66	76.32	99.94	0.12	1.27
		Elkan	126.42	98.93	0.02	33.33	100.00	0.00	0.38	79.80	89.24	7.10	44.76	98.91	1.72	12.01
		Bayes	126.41	98.93	0.29	22.47	99.99	0.01	0.82	81.25	86.65	21.54	30.84	94.32	7.35	25.36
		2-DDR	126.42	98.94	0.00		100.00	0.00		75.88	89.45	0.04	12.34	99.97	0.03	0.69
		2-DDRq	126.42	98.94	0.00		100.00	0.00		76.14	89.47	0.00		100.00	0.00	
		2-DDRM	126.42	98.94	0.00		100.00	0.00		76.53	89.47	0.00	0.00	99.82	0.00	
		2-DDRMq	126.42	98.94	0.00		100.00	0.00		76.15	89.46	0.00		99.83	0.00	
4	0.01	Youden	0.57	71.45	34.92	1.40	71.84	28.24	2.56	-3.66	60.50	44.95	12.89	62.35	38.43	19.00
		Brute force	0.47	3.01	98.74	1.10	1.95	98.05	2.18	-37.73	89.01	0.98	25.87	99.42	0.62	1.76
		Elkan	0.13	86.54	17.59	1.51	87.30	12.75	2.79	-6.08	64.23	41.23	12.86	66.95	33.91	19.58
		Bayes	0.71	70.01	37.55	1.38	70.37	29.71	2.66	14.07	50.30	58.91	12.08	49.28	51.59	20.04
		2-DDR	-0.13	79.31	19.86	1.17	79.95	20.04	2.04	37.95	13.38	95.13	10.46	3.71	96.17	18.85
		2-DDRq	0.12	65.19	33.93	1.06	65.52	34.48	1.95	38.21	12.79	96.60	10.52	2.89	97.06	18.98
		2-DDRM	0.03	77.04	20.88	1.04	75.90	21.12	2.17	35.96	13.41	94.15	10.46	3.71	95.17	18.83
		2-DDRMq	0.08	68.93	25.59	1.34	65.54	25.49	1.95	36.63	13.21	94.68	10.59	3.26	94.55	19.05
	0.10	Youden	-3.67	44.18	63.25	1.27	43.96	56.12	2.47	6.13	51.85	56.15	12.36	51.34	49.45	19.73
		Brute force	-31.68	5.88	96.57	1.13	4.87	95.14	2.23	-6.44	89.04	0.75	17.23	99.55	0.48	1.36
		Elkan	26.83	86.64	18.11	1.59	87.41	12.65	2.94	0.85	73.35	26.69	13.23	78.90	21.69	18.50
		Bayes	19.14	78.36	28.29	1.47	78.92	21.16	2.80	6.70	57.98	49.12	12.50	59.04	41.83	19.92
		2-DDR	35.21	98.90	0.00		100.00	0.00		0.96	49.70	48.65	10.30	49.84	50.00	15.05
		2-DDRq	35.21	98.90	0.00		100.00	0.00		3.43	40.65	61.45	10.39	38.17	61.79	17.43
		2-DDRM	35.21	98.90	0.00		100.00	0.00		16.26	70.23	12.76	13.06	48.01	11.51	12.03
		2-DDRMq	35.21	98.90	0.00		99.92	0.00		20.75	67.30	13.06	12.47	38.72	11.51	12.34
	0.33	Youden	44.80	67.13	40.43	1.40	67.42	32.67	2.70	19.98	57.83	49.02	12.44	58.88	41.95	19.54
		Brute force	-120.04	3.46	97.91	1.11	2.40	97.60	2.19	75.13	89.17	0.79	19.12	99.58	0.46	1.76
		Elkan	96.41	87.24	17.26	1.59	88.03	12.03	2.91	41.09	68.97	34.32	13.04	73.05	27.72	18.88
		Bayes	84.74	84.02	21.29	1.54	84.72	15.35	2.86	29.56	64.62	40.46	12.77	67.47	33.37	19.41
		2-DDR	125.98	98.90	0.00		100.00	0.00		74.64	89.38	0.12	11.33	99.89	0.11	2.18
		2-DDRq	125.98	98.90	0.00		100.00	0.00		75.50	89.47	0.00		100.00	0.00	
		2-DDRM	125.98	98.90	0.00		100.00	0.00		76.23	89.47	0.00		99.63	0.00	
		2-DDRMq	125.98	98.90	0.00		100.00	0.00		76.04	89.47	0.00		99.70	0.00	

Table 3.6: Results for the simulated multi-class classification problems for the sample size $n = 1,000$ and bias $U = 0$ for the different thresholding algorithms introduced throughout the work.

			Umbalanced							Balanced						
σ	a	Thresholding	AER	Acc	Rec	PPV	Spec	PP	FS	AER	Acc	Rec	PPV	Spec	PP	FS
0.5	0.01	Youden	2.40	67.36	62.85	2.37	67.41	32.92	4.50	18.27	61.98	70.62	17.93	60.94	42.39	28.37
		Brute force	0.42	2.67	99.75	1.09	1.62	98.40	2.16	-37.55	89.47	1.06	59.91	99.90	0.20	2.03
		Elkan	-0.16	98.72	2.26	10.56	99.77	0.25	3.35	7.92	72.22	57.28	20.91	73.98	29.32	30.42
		Bayes	2.42	78.58	44.80	2.27	78.95	21.30	4.31	37.71	28.63	89.22	11.82	21.49	79.64	20.87
		2-DDR	-0.31	53.54	43.52	0.98	53.66	46.31	1.91	36.88	10.66	96.93	10.30	0.48	99.24	18.63
		2-DDRq	-0.23	69.17	29.09	0.99	69.62	30.37	1.92	37.79	11.12	97.39	10.39	0.94	98.88	18.78
		2-DDRM	-0.04	80.56	17.15	1.56	74.83	15.73	2.44	33.19	10.96	94.41	10.59	0.47	94.09	19.04
		2-DDRMq	0.10	76.51	22.91	1.34	72.45	20.03	2.46	35.54	16.23	89.46	10.95	6.08	86.84	19.47
	0.10	Youden	9.05	59.08	75.13	1.97	58.91	41.46	3.84	22.98	68.60	60.89	20.09	69.51	33.67	29.20
		Brute force	-35.10	1.42	99.98	1.09	0.34	99.66	2.15	-5.20	89.57	0.98	53.82	99.90	0.19	1.90
		Elkan	35.41	98.85	1.10	9.29	99.92	0.09	2.76	17.01	81.37	39.38	25.53	86.28	16.40	30.53
		Bayes	33.56	96.14	14.03	4.90	97.03	3.09	7.26	21.13	48.28	79.72	14.38	44.61	57.93	24.36
		2-DDR	35.05	98.85	0.03	0.55	99.93	0.07	0.42	2.93	33.62	68.92	10.23	29.57	70.27	17.16
		2-DDRq	35.37	98.92	0.00		100.00	0.00		4.58	26.44	79.02	10.39	20.32	79.62	18.12
		2-DDRM	35.38	98.92	0.00		99.72	0.00		28.07	48.38	35.17	19.38	29.57	20.98	21.52
		2-DDRMq	35.38	98.92	0.00		99.73	0.00		32.22	52.71	40.28	20.31	24.98	22.06	24.85
	0.33	Youden	52.93	69.79	61.06	2.34	69.87	30.45	4.46	53.28	67.34	63.27	19.69	67.84	35.43	29.44
		Brute force	-124.01	1.85	99.94	1.06	0.81	99.20	2.10	76.18	89.49	0.51	64.73	99.96	0.09	1.00
		Elkan	126.10	98.93	0.51	21.00	99.98	0.03	1.10	80.38	85.44	26.78	30.73	92.34	9.67	27.27
		Bayes	125.52	98.73	2.72	10.33	99.75	0.28	4.30	57.85	72.95	54.62	20.51	75.10	28.03	29.82
		2-DDR	126.10	98.95	0.00		100.00	0.00		75.86	89.47	0.00		100.00	0.00	
		2-DDRq	126.10	98.95	0.00		100.00	0.00		75.86	89.47	0.00		100.00	0.00	
		2-DDRM	126.10	98.95	0.00		100.00	0.00		76.06	89.47	0.00	50.00	99.85	0.00	0.04
		2-DDRMq	126.10	98.95	0.00		100.00	0.00		75.86	89.47	0.00		100.00	0.00	
4	0.01	Youden	0.85	47.69	60.90	1.26	47.55	52.54	2.47	6.99	49.07	58.72	12.28	47.97	52.73	18.27
		Brute force	0.39	3.96	98.01	1.08	2.95	97.06	2.13	-37.00	89.15	0.90	15.35	99.46	0.57	1.73
		Elkan	0.47	77.98	29.01	1.44	78.51	21.57	2.74	-0.95	59.18	47.89	12.41	60.51	40.37	20.12
		Bayes	0.79	61.93	46.20	1.30	62.10	37.99	2.52	19.92	42.97	67.44	11.62	40.11	60.68	19.83
		2-DDR	-0.31	65.32	32.54	1.04	65.66	34.32	1.97	36.97	15.00	92.40	10.30	5.96	93.87	18.53
		2-DDRq	-0.33	72.37	25.02	0.95	72.88	27.10	1.72	37.29	12.28	96.63	10.37	2.43	97.47	18.73
		2-DDRM	-0.08	71.22	25.94	0.90	67.73	27.16	1.89	33.94	15.12	90.27	10.34	5.91	91.33	18.55
		2-DDRMq	-0.21	79.99	17.29	0.96	75.55	17.45	1.75	35.26	13.91	92.61	10.41	4.32	93.04	18.72
	0.10	Youden	9.19	61.64	46.39	1.34	61.80	38.28	2.58	4.92	56.47	49.90	12.46	57.25	43.50	19.09
		Brute force	-33.19	3.81	98.22	1.08	2.79	97.22	2.13	-5.74	89.18	0.83	20.51	99.56	0.49	1.53
		Elkan	21.73	79.20	27.56	1.44	79.75	20.32	2.74	5.47	59.96	46.83	12.50	61.50	39.38	19.71
		Bayes	13.66	71.05	37.62	1.40	71.41	28.68	2.69	7.45	50.68	58.15	11.97	49.81	51.03	19.86
		2-DDR	35.22	98.93	0.00	0.00	99.99	0.01		1.25	52.22	46.62	10.26	52.81	47.13	15.67
		2-DDRq	35.27	98.94	0.00		100.00	0.00		0.63	54.11	42.93	10.03	55.33	44.48	16.22
		2-DDRM	35.28	98.94	0.00		99.98	0.00		15.57	67.55	14.08	12.91	52.22	11.83	15.76
		2-DDRMq	35.28	98.94	0.00		99.92	0.00		15.69	74.15	10.75	13.46	54.65	9.14	12.84
	0.33	Youden	22.95	58.72	48.24	1.27	58.83	41.24	2.46	11.22	53.23	55.00	12.31	53.01	47.84	19.66
		Brute force	-113.30	6.00	96.43	1.08	5.02	94.99	2.14	74.21	88.61	1.82	16.25	98.85	1.22	3.23
		Elkan	89.28	84.57	19.65	1.43	85.27	14.78	2.64	30.72	63.45	41.78	12.74	66.00	34.82	19.33
		Bayes	67.70	77.92	27.79	1.37	78.46	21.61	2.60	15.45	57.47	49.75	12.35	58.38	42.47	19.79
		2-DDR	126.14	98.94	0.00		100.00	0.00		73.98	89.32	0.16	9.79	99.83	0.17	0.94
		2-DDRq	126.14	98.94	0.00		100.00	0.00		75.38	89.45	0.00		100.00	0.00	
		2-DDRM	126.14	98.94	0.00		100.00	0.00		76.87	89.45	0.00		99.30	0.00	
		2-DDRMq	126.14	98.94	0.00		100.00	0.00		75.77	89.45	0.00		99.68	0.00	

Table 3.7: Results for the simulated multi-class classification problems for the sample size $n = 1,000$ and bias $U = 1$ for the different thresholding algorithms introduced throughout the work.

			Umbalanced							Balanced						
σ	a	Thresholding	AER	Acc	Rec	PPV	Spec	PP	FS	AER	Acc	Rec	PPV	Spec	PP	FS
0.5	0.01	Youden	2.60	67.74	64.54	2.24	67.78	32.57	4.32	12.62	68.45	63.32	19.50	69.04	34.36	29.74
		Brute force	0.28	1.09	100.00	1.08	0.01	99.99	2.13	-38.29	89.49	0.09	71.71	100.00	0.01	0.18
		Elkan	-0.28	98.92	0.00		100.00	0.00		-34.18	89.39	5.07	47.54	99.29	1.16	9.01
		Bayes	0.60	98.09	6.00	6.77	99.10	0.96	6.36	22.28	67.93	55.16	17.48	69.43	33.16	26.55
		2-DDR	0.07	29.50	70.46	0.97	29.04	70.95	1.83	38.25	10.51	99.79	10.49	0.02	99.96	18.99
		2-DDRq	0.18	32.02	68.59	1.08	31.62	68.38	2.08	38.26	11.24	98.60	10.47	0.98	98.97	18.93
		2-DDRM	0.53	26.32	68.56	3.96	24.37	66.76	1.99	37.96	10.51	99.62	10.49	0.03	99.83	18.98
		2-DDRMq	0.76	42.43	56.09	1.51	37.42	50.99	2.32	38.16	12.31	97.34	10.55	2.22	96.97	19.04
	0.10	Youden	10.96	61.56	71.55	2.03	61.46	38.90	3.94	25.50	66.28	66.42	19.00	66.27	37.19	29.48
		Brute force	-35.10	1.09	100.00	1.08	0.01	99.99	2.14	-6.84	89.42	0.05	77.83	100.00	0.01	0.12
		Elkan	35.11	98.92	0.00		100.00	0.00		-5.59	89.41	1.66	60.13	99.79	0.36	3.18
		Bayes	35.14	98.91	0.27	22.93	99.99	0.01	0.65	11.96	86.48	21.55	30.43	94.17	7.50	25.23
		2-DDR	35.09	98.92	0.02	4.17	100.00	0.00	0.35	6.76	10.59	99.80	10.57	0.03	99.96	19.11
		2-DDRq	35.11	98.92	0.00		100.00	0.00		6.77	17.81	91.05	10.60	9.13	90.89	18.98
		2-DDRM	35.12	98.92	0.00		99.98	0.00		35.48	19.74	61.89	16.32	2.07	40.85	25.61
		2-DDRMq	35.11	98.92	0.00		100.00	0.00		35.84	43.18	48.65	20.42	12.83	26.19	28.21
	0.33	Youden	61.73	73.11	60.02	2.53	73.26	27.11	4.83	50.68	65.53	66.98	18.66	65.36	38.05	29.15
		Brute force	-126.36	1.10	100.00	1.09	0.01	99.99	2.16	75.78	89.44	0.06	87.78	100.00	0.01	0.19
		Elkan	126.38	98.91	0.00		100.00	0.00		75.96	89.45	0.38	72.37	99.98	0.06	0.75
		Bayes	126.38	98.91	0.00	0.00	100.00	0.00		78.78	89.38	4.66	47.35	99.39	1.04	8.48
		2-DDR	126.38	98.91	0.00		100.00	0.00		75.67	89.43	0.01	12.00	99.99	0.01	0.11
		2-DDRq	126.38	98.91	0.00		100.00	0.00		75.74	89.43	0.00		100.00	0.00	
		2-DDRM	126.38	98.91	0.00		100.00	0.00		75.78	89.43	0.00		99.99	0.00	
		2-DDRMq	126.38	98.91	0.00		100.00	0.00		75.74	89.43	0.00		100.00	0.00	
4	0.01	Youden	0.69	53.93	54.18	1.28	53.92	46.17	2.50	-1.48	59.21	47.34	12.53	60.63	40.22	19.58
		Brute force	0.35	1.54	99.60	1.07	0.48	99.52	2.11	-38.70	89.42	0.01	10.48	99.99	0.01	0.08
		Elkan	-0.05	91.54	10.97	1.52	92.41	7.62	2.74	-13.67	70.90	31.79	13.35	75.52	25.25	18.73
		Bayes	0.47	77.47	29.02	1.40	77.99	22.09	2.68	7.07	57.39	49.90	12.39	58.28	42.59	19.85
		2-DDR	-0.07	36.34	61.42	1.05	36.07	63.90	1.99	36.84	13.23	95.04	10.43	3.55	96.30	18.80
		2-DDRq	0.11	51.01	48.85	1.05	51.03	48.97	1.99	38.69	11.01	99.33	10.56	0.56	99.43	19.09
		2-DDRM	0.15	41.99	53.50	1.03	40.18	55.42	1.81	36.36	13.23	94.85	10.43	3.55	96.11	18.80
		2-DDRMq	0.17	62.50	33.12	1.01	57.61	33.25	1.80	38.24	12.03	97.93	10.60	1.81	97.68	19.13
	0.10	Youden	4.77	55.39	55.06	1.42	55.39	44.72	2.73	6.52	57.15	50.76	12.53	57.91	43.01	20.00
		Brute force	-34.91	1.39	99.88	1.10	0.30	99.70	2.18	-6.52	89.41	0.04	16.52	99.98	0.02	0.14
		Elkan	28.88	89.61	14.86	1.70	90.44	9.62	3.04	0.40	76.77	22.28	13.75	83.23	17.36	16.96
		Bayes	23.74	84.39	21.86	1.61	85.09	14.99	2.99	5.52	64.97	40.13	12.89	67.91	32.94	19.52
		2-DDR	35.14	98.90	0.00		100.00	0.00		6.24	15.12	92.86	10.46	5.92	93.95	18.80
		2-DDRq	35.14	98.90	0.00		100.00	0.00		6.44	16.20	92.63	10.56	7.14	92.84	18.95
		2-DDRM	35.14	98.90	0.00		100.00	0.00		27.03	32.41	28.46	12.08	8.99	25.83	16.28
		2-DDRMq	35.14	98.90	0.00		100.00	0.00		27.31	30.42	29.81	11.84	8.75	26.91	16.74
	0.33	Youden	21.50	58.07	50.40	1.31	58.15	41.94	2.53	20.74	58.09	49.05	12.54	59.15	41.72	19.65
		Brute force	-125.63	1.30	99.83	1.06	0.24	99.76	2.11	75.91	89.46	0.03	19.54	99.99	0.01	0.09
		Elkan	108.30	91.83	10.75	1.57	92.70	7.33	2.73	56.34	77.15	22.20	13.80	83.62	16.99	17.57
		Bayes	98.15	88.90	14.80	1.52	89.70	10.35	2.76	42.36	70.98	31.75	13.28	75.59	25.18	18.73
		2-DDR	126.26	98.94	0.00		100.00	0.00		75.69	89.45	0.02	22.79	99.98	0.02	0.08
		2-DDRq	126.26	98.94	0.00		100.00	0.00		75.91	89.47	0.00		100.00	0.00	
		2-DDRM	126.26	98.94	0.00		100.00	0.00		76.02	89.47	0.00		99.97	0.00	
		2-DDRMq	126.26	98.94	0.00		100.00	0.00		75.92	89.47	0.00		99.99	0.00	

Table 3.8: Results for the simulated multi-class classification problems for the sample size $n = 10,000$ and bias $U = -1$ for the different thresholding algorithms introduced throughout the work.

			Umbalanced							Balanced						
σ	a	Thresholding	AER	Acc	Rec	PPV	Spec	PP	FS	AER	Acc	Rec	PPV	Spec	PP	FS
0.5	0.01	Youden	2.81	68.04	66.07	2.26	68.06	32.31	4.37	14.23	66.98	65.63	19.24	67.14	36.33	29.67
		Brute force	0.35	1.09	100.00	1.08	0.00	100.00	2.14	-38.78	89.43	0.07	72.71	100.00	0.01	0.15
		Elkan	-0.34	98.91	0.09	26.86	99.99	0.01	0.61	-27.59	87.86	13.70	40.26	96.62	4.47	20.97
		Bayes	1.74	93.42	20.16	3.68	94.22	5.94	6.22	34.91	46.34	77.53	13.78	42.65	59.48	23.40
		2-DDR	0.10	40.07	59.98	1.10	39.85	60.15	1.92	38.72	10.57	99.81	10.56	0.02	99.96	19.10
		2-DDRq	0.06	57.41	42.40	1.07	57.58	42.42	2.08	38.76	12.09	97.94	10.56	1.94	98.05	19.07
		2-DDRM	0.30	40.31	55.64	0.83	39.47	54.83	2.18	38.45	10.59	99.64	10.57	0.02	99.69	19.11
		2-DDRMq	0.55	61.46	38.00	1.96	57.44	33.76	2.39	38.60	13.20	96.61	10.72	3.10	95.30	19.30
	0.10	Youden	17.05	70.16	63.45	2.33	70.23	30.12	4.48	25.47	66.26	66.13	18.95	66.27	37.17	29.42
		Brute force	-35.50	1.07	100.00	1.06	0.01	99.99	2.09	-6.58	89.39	0.09	81.80	100.00	0.01	0.18
		Elkan	35.51	98.94	0.06	26.19	100.00	0.00	0.38	2.65	88.27	13.25	39.47	97.18	3.93	18.21
		Bayes	35.44	98.67	2.47	8.02	99.70	0.32	3.77	24.80	73.23	52.44	20.41	75.70	27.29	29.38
		2-DDR	35.49	98.94	0.02	4.55	100.00	0.00	0.38	6.46	10.61	99.68	10.59	0.03	99.94	19.15
		2-DDRq	35.51	98.94	0.00		100.00	0.00		6.58	13.15	96.78	10.62	3.21	96.79	19.14
		2-DDRM	35.51	98.94	0.00		100.00	0.00		35.14	16.00	63.21	15.96	0.03	42.07	25.47
		2-DDRMq	35.51	98.94	0.00		100.00	0.00		35.72	26.06	59.44	17.60	4.76	36.06	27.10
	0.33	Youden	36.87	63.40	70.01	2.08	63.32	37.04	4.03	48.29	64.28	68.48	18.30	63.79	39.61	28.83
		Brute force	-126.18	1.09	100.00	1.08	0.01	99.99	2.14	76.20	89.47	0.03	68.33	100.00	0.00	0.09
		Elkan	126.20	98.92	0.00		100.00	0.00		79.76	89.14	7.84	41.88	98.71	1.98	13.12
		Bayes	126.18	98.91	0.15	11.46	99.99	0.01	0.48	81.61	86.71	22.00	31.37	94.33	7.39	25.86
		2-DDR	126.20	98.92	0.00		100.00	0.00		76.17	89.47	0.00		100.00	0.00	
		2-DDRq	126.20	98.92	0.00		100.00	0.00		76.17	89.47	0.00		100.00	0.00	
		2-DDRM	126.20	98.92	0.00		100.00	0.00		76.28	89.47	0.00		99.97	0.00	
		2-DDRMq	126.20	98.92	0.00		100.00	0.00		76.17	89.47	0.00		100.00	0.00	
4	0.01	Youden	0.80	47.83	60.69	1.29	47.68	52.41	2.52	0.79	57.84	50.04	12.74	58.77	42.16	20.04
		Brute force	0.46	1.57	99.72	1.09	0.48	99.52	2.16	-38.82	89.40	0.01	12.93	99.99	0.01	0.05
		Elkan	0.17	85.19	19.31	1.50	85.92	14.14	2.77	-7.68	66.13	39.26	13.18	69.32	31.59	19.64
		Bayes	0.71	70.02	37.33	1.37	70.38	29.70	2.64	14.35	50.25	59.35	12.16	49.17	51.73	20.18
		2-DDR	0.02	57.05	41.90	1.12	57.23	42.76	1.97	37.64	13.73	94.35	10.45	4.18	95.67	18.82
		2-DDRq	0.16	40.41	58.45	1.06	40.22	59.77	2.06	38.81	10.92	99.37	10.58	0.44	99.54	19.12
		2-DDRM	0.16	59.98	37.41	1.54	59.37	38.08	1.83	37.35	13.74	94.24	10.45	4.18	95.55	18.82
		2-DDRMq	0.23	53.40	42.99	1.15	49.66	43.53	2.02	38.39	11.22	98.72	10.60	0.79	98.70	19.14
	0.10	Youden	5.32	56.52	52.26	1.36	56.57	43.52	2.64	7.34	50.00	59.75	12.21	48.86	52.05	20.17
		Brute force	-35.23	1.12	100.00	1.11	0.01	99.99	2.20	-6.39	89.42	0.02	20.24	99.99	0.01	0.08
		Elkan	25.72	85.17	20.17	1.59	85.90	14.17	2.94	4.57	64.96	40.36	12.94	67.87	33.00	19.59
		Bayes	18.93	78.35	28.08	1.48	78.92	21.16	2.81	6.95	58.13	49.48	12.53	59.16	41.76	20.00
		2-DDR	35.23	98.89	0.00		100.00	0.00		5.15	22.71	83.18	10.44	15.55	84.31	18.54
		2-DDRq	35.23	98.89	0.00		100.00	0.00		6.30	15.18	93.75	10.54	5.89	94.07	18.94
		2-DDRM	35.23	98.89	0.00		100.00	0.00		26.04	45.73	23.12	12.26	15.76	20.56	15.38
		2-DDRMq	35.23	98.89	0.00		100.00	0.00		27.41	28.85	29.66	11.86	7.00	26.74	16.37
	0.33	Youden	16.34	56.18	54.27	1.39	56.20	43.91	2.70	10.11	52.34	57.17	12.31	51.78	49.17	20.20
		Brute force	-125.68	1.12	100.00	1.11	0.01	99.99	2.20	75.87	89.44	0.02	12.49	99.99	0.01	0.05
		Elkan	97.90	87.98	16.81	1.65	88.78	11.29	3.00	41.78	69.03	34.99	13.28	73.04	27.81	19.24
		Bayes	84.34	84.02	22.02	1.59	84.72	15.36	2.97	30.07	64.71	40.86	12.92	67.52	33.36	19.64
		2-DDR	125.71	98.89	0.00		100.00	0.00		75.88	89.44	0.02	23.81	99.99	0.01	0.19
		2-DDRq	125.71	98.89	0.00		100.00	0.00		75.88	89.45	0.00		100.00	0.00	
		2-DDRM	125.71	98.89	0.00		100.00	0.00		75.98	89.45	0.00		99.99	0.00	
		2-DDRMq	125.71	98.89	0.00		100.00	0.00		75.90	89.45	0.00		99.99	0.00	

Table 3.9: Results for the simulated multi-class classification problems for the sample size $n = 10,000$ and bias $U = 0$ for the different thresholding algorithms introduced throughout the work.

			Umbalanced							Balanced						
σ	a	Thresholding	AER	Acc	Rec	PPV	Spec	PP	FS	AER	Acc	Rec	PPV	Spec	PP	FS
0.5	0.01	Youden	2.75	70.36	63.21	2.38	70.44	29.93	4.58	15.77	65.16	67.42	18.50	64.89	38.52	29.01
		Brute force	0.45	1.10	100.00	1.09	0.01	99.99	2.16	-38.42	89.45	0.03	46.72	100.00	0.01	0.11
		Elkan	-0.24	98.63	2.70	9.52	99.69	0.33	4.02	1.37	74.15	49.23	21.18	77.08	25.69	29.75
		Bayes	2.65	78.63	46.26	2.38	78.99	21.29	4.53	37.80	28.55	89.35	11.82	21.38	79.75	20.88
		2-DDR	-0.23	72.53	26.58	1.01	73.03	26.96	1.59	38.40	10.56	99.78	10.53	0.03	99.95	19.05
		2-DDRq	-0.10	75.70	23.99	1.11	76.28	23.73	2.15	38.40	12.02	97.88	10.53	1.89	98.08	19.01
		2-DDRM	0.10	81.96	17.87	2.03	78.71	15.68	2.06	37.90	10.60	99.56	10.59	0.02	99.22	19.14
		2-DDRMq	0.33	80.22	21.14	1.49	74.93	16.87	2.83	38.24	12.24	97.48	10.62	2.05	96.88	19.15
	0.10	Youden	18.23	71.76	61.64	2.44	71.87	28.49	4.68	25.52	62.96	69.85	17.90	62.15	41.22	28.48
		Brute force	-35.21	1.09	99.98	1.08	0.01	99.99	2.13	-6.20	89.46	0.04	70.48	100.00	0.00	0.17
		Elkan	35.23	98.86	0.81	11.72	99.93	0.08	2.73	15.87	81.33	37.38	26.30	86.49	16.03	29.25
		Bayes	33.49	96.14	15.02	5.21	97.03	3.10	7.74	21.31	48.25	79.62	14.47	44.55	58.00	24.49
		2-DDR	35.22	98.92	0.00		100.00	0.00		6.08	13.03	96.56	10.51	3.16	96.81	18.94
		2-DDRq	35.22	98.92	0.00		100.00	0.00		6.10	16.43	92.56	10.54	7.43	92.57	18.89
		2-DDRM	35.22	98.92	0.00		100.00	0.00		34.81	21.48	58.26	16.53	3.16	38.14	25.31
		2-DDRMq	35.22	98.92	0.00		100.00	0.00		35.12	33.10	53.40	18.01	9.62	32.17	26.38
	0.33	Youden	53.11	69.79	63.97	2.39	69.85	30.51	4.59	51.88	66.05	66.27	18.69	66.02	37.36	29.09
		Brute force	-126.34	1.09	100.00	1.08	0.01	99.99	2.14	76.49	89.52	0.06	67.76	100.00	0.01	0.13
		Elkan	126.33	98.90	0.45	14.40	99.97	0.03	0.96	80.72	85.32	27.47	29.86	92.08	9.97	27.65
		Bayes	125.68	98.67	2.40	8.86	99.73	0.30	3.77	57.79	72.89	54.33	20.33	75.07	28.01	29.58
		2-DDR	126.36	98.92	0.00		100.00	0.00		76.46	89.52	0.00		100.00	0.00	
		2-DDRq	126.36	98.92	0.00		100.00	0.00		76.46	89.52	0.00		100.00	0.00	
		2-DDRM	126.36	98.92	0.00		100.00	0.00		76.48	89.52	0.00		99.99	0.00	
		2-DDRMq	126.36	98.92	0.00		100.00	0.00		76.46	89.52	0.00		100.00	0.00	
4	0.01	Youden	0.67	60.05	48.31	1.35	60.18	39.91	2.61	3.79	53.91	54.41	12.27	53.83	47.03	19.77
		Brute force	0.35	1.32	99.89	1.09	0.23	99.78	2.16	-38.41	89.50	0.02	32.81	100.00	0.01	0.07
		Elkan	0.40	78.20	27.68	1.43	78.76	21.31	2.70	-5.17	62.46	42.60	12.44	64.79	35.99	19.76
		Bayes	0.85	61.84	46.96	1.35	62.01	38.09	2.62	19.82	42.98	67.29	11.65	40.13	60.65	19.85
		2-DDR	-0.21	79.56	19.38	1.12	80.23	19.76	1.66	37.77	15.12	92.53	10.35	6.04	93.81	18.62
		2-DDRq	-0.18	79.36	19.58	1.07	80.03	19.97	1.86	38.40	10.72	99.48	10.48	0.31	99.67	18.96
		2-DDRM	-0.09	74.25	22.16	1.16	73.15	23.00	1.77	37.38	15.13	92.38	10.36	6.04	93.64	18.62
		2-DDRMq	-0.13	84.07	12.71	1.03	80.38	12.84	1.55	37.92	10.75	99.09	10.49	0.34	99.18	18.97
	0.10	Youden	2.57	52.44	57.44	1.33	52.38	47.72	2.59	5.98	52.50	55.51	12.26	52.15	48.66	19.61
		Brute force	-35.03	1.23	99.96	1.10	0.13	99.87	2.17	-6.46	89.41	0.01	15.83	99.99	0.01	0.06
		Elkan	21.85	79.62	26.70	1.47	80.21	19.86	2.79	5.83	58.62	48.41	12.49	59.83	41.05	19.84
		Bayes	13.36	70.91	37.19	1.42	71.28	28.81	2.73	7.59	50.70	58.23	12.07	49.81	51.04	20.00
		2-DDR	35.09	98.90	0.00	0.00	99.99	0.01		6.06	26.55	79.28	10.54	20.31	79.64	18.60
		2-DDRq	35.12	98.90	0.00		100.00	0.00		6.40	14.61	94.57	10.56	5.14	94.83	18.99
		2-DDRM	35.12	98.90	0.00		99.99	0.00		26.79	60.86	15.62	12.80	20.63	13.78	12.04
		2-DDRMq	35.12	98.90	0.00		100.00	0.00		27.48	32.83	24.98	11.77	8.00	22.88	15.15
	0.33	Youden	36.19	63.81	42.95	1.31	64.02	36.05	2.50	10.57	52.82	55.68	12.22	52.49	48.37	19.86
		Brute force	-126.01	1.08	99.98	1.07	0.01	99.99	2.13	75.60	89.42	0.12	16.48	99.94	0.07	0.62
		Elkan	84.49	82.64	22.96	1.48	83.28	16.78	2.77	29.51	62.51	43.49	12.69	64.75	36.12	19.64
		Bayes	67.70	77.91	29.06	1.44	78.44	21.64	2.75	15.47	57.60	49.73	12.38	58.52	42.35	19.82
		2-DDR	126.03	98.93	0.00		100.00	0.00		75.67	89.46	0.00		100.00	0.00	
		2-DDRq	126.03	98.93	0.00		100.00	0.00		75.67	89.46	0.00		100.00	0.00	
		2-DDRM	126.03	98.93	0.00		100.00	0.00		75.80	89.46	0.00		99.94	0.00	
		2-DDRMq	126.03	98.93	0.00		100.00	0.00		75.68	89.46	0.00		99.99	0.00	

Table 3.10: Results for the simulated multi-class classification problems for the sample size $n = 10,000$ and bias $U = 1$ for the different thresholding algorithms introduced throughout the work.

			Umbalanced							Balanced						
σ	a	Thresholding	AER	Acc	Rec	PPV	Spec	PP	FS	AER	Acc	Rec	PPV	Spec	PP	FS
0.5	0.01	Youden	2.88	65.76	68.23	2.20	65.73	34.64	4.25	14.78	66.29	66.64	18.68	66.26	37.17	29.16
		Brute force	0.36	1.11	100.00	1.11	0.00	100.00	2.19	-37.72	89.59	0.01	100.00	100.00	0.00	0.04
		Elkan	-0.36	98.89	0.00	0.00	100.00	0.00		-34.58	89.51	3.91	47.40	99.46	0.89	7.07
		Bayes	0.49	98.09	6.03	7.07	99.12	0.94	6.51	21.94	67.98	55.14	17.34	69.47	33.09	26.39
		2-DDR	0.28	13.34	87.36	1.10	12.51	87.49	2.18	37.72	10.41	99.99	10.41	0.00	100.00	18.85
		2-DDRq	0.28	23.07	78.21	1.11	22.46	77.54	2.20	37.70	12.41	97.54	10.41	2.52	97.49	18.82
		2-DDRM	0.88	16.56	78.26	1.17	15.37	74.09	2.30	37.67	10.41	99.97	10.41	0.00	99.98	18.85
		2-DDRMq	1.02	23.11	75.00	1.22	21.23	67.82	2.41	37.66	12.49	97.46	10.48	2.52	96.82	18.92
	0.10	Youden	14.15	66.08	68.85	2.13	66.05	34.32	4.13	25.22	65.49	67.02	18.54	65.31	38.09	29.03
		Brute force	-35.33	1.06	100.00	1.06	0.00	100.00	2.09	-6.15	89.47	0.00	100.00	100.00	0.00	0.04
		Elkan	35.33	98.94	0.00		100.00	0.00		-5.80	89.48	0.47	59.22	99.95	0.09	1.09
		Bayes	35.36	98.94	0.24	16.26	99.99	0.01	0.75	11.88	86.58	21.46	30.50	94.25	7.41	25.19
		2-DDR	35.33	98.94	0.00		100.00	0.00		6.14	10.53	99.99	10.53	0.01	99.99	19.05
		2-DDRq	35.33	98.94	0.00		100.00	0.00		6.15	12.49	97.46	10.52	2.49	97.51	18.99
		2-DDRM	35.33	98.94	0.00		100.00	0.00		35.18	15.83	62.12	15.82	0.00	41.67	25.18
		2-DDRMq	35.33	98.94	0.00		100.00	0.00		35.49	27.28	55.98	17.61	5.25	33.63	26.74
	0.33	Youden	55.20	70.48	64.23	2.32	70.55	29.83	4.47	51.90	65.96	66.65	18.82	65.88	37.56	29.33
		Brute force	-126.79	1.07	100.00	1.07	0.00	100.00	2.11	75.61	89.41	0.01	75.00	100.00	0.00	0.05
		Elkan	126.80	98.93	0.00		100.00	0.00		75.84	89.43	0.37	64.84	99.97	0.06	0.85
		Bayes	126.80	98.93	0.02	100.00	100.00	0.00	0.37	78.51	89.31	4.48	45.36	99.36	1.05	8.15
		2-DDR	126.80	98.93	0.00		100.00	0.00		75.59	89.41	0.00		100.00	0.00	
		2-DDRq	126.80	98.93	0.00		100.00	0.00		75.59	89.41	0.00		100.00	0.00	
		2-DDRM	126.80	98.93	0.00		100.00	0.00		75.59	89.41	0.00		100.00	0.00	
		2-DDRMq	126.80	98.93	0.00		100.00	0.00		75.59	89.41	0.00		100.00	0.00	
4	0.01	Youden	0.67	53.30	55.34	1.29	53.28	46.82	2.51	3.39	54.63	54.05	12.39	54.71	46.22	20.11
		Brute force	0.35	1.08	100.00	1.08	0.00	100.00	2.13	-38.77	89.43	0.00	0.00	100.00	0.00	
		Elkan	0.06	89.51	13.95	1.56	90.33	9.71	2.80	-15.64	72.55	29.22	13.48	77.67	23.05	18.19
		Bayes	0.45	77.52	28.78	1.41	78.05	22.03	2.69	6.79	57.39	49.89	12.38	58.28	42.58	19.84
		2-DDR	0.22	30.96	68.70	1.33	30.55	69.44	1.96	34.18	13.42	94.11	10.37	3.88	95.90	18.68
		2-DDRq	0.34	13.84	86.05	1.07	13.06	86.93	2.11	38.75	12.58	97.49	10.57	2.55	97.46	19.08
		2-DDRM	0.44	32.42	61.56	1.07	29.97	61.81	2.10	34.12	13.42	94.09	10.37	3.88	95.89	18.68
		2-DDRMq	0.56	24.73	69.31	1.08	21.95	69.09	2.13	37.98	12.72	95.94	10.67	2.55	95.06	19.20
	0.10	Youden	-3.12	44.56	64.99	1.26	44.34	55.76	2.48	6.24	55.35	52.72	12.33	55.66	45.23	19.96
		Brute force	-35.17	1.08	99.97	1.08	0.00	100.00	2.14	-6.68	89.43	0.00	0.00	100.00	0.00	
		Elkan	29.38	90.36	12.91	1.60	91.21	8.84	2.83	1.59	73.63	27.79	13.60	79.04	21.68	18.13
		Bayes	23.73	84.35	20.78	1.50	85.05	15.02	2.80	5.05	65.03	39.84	12.82	68.01	32.82	19.40
		2-DDR	35.17	98.92	0.00		100.00	0.00		5.70	13.81	93.82	10.38	4.35	95.45	18.70
		2-DDRq	35.17	98.92	0.00		100.00	0.00		6.67	12.90	97.10	10.57	2.95	97.06	19.07
		2-DDRM	35.17	98.92	0.00		100.00	0.00		26.23	19.74	32.84	10.70	4.35	32.48	16.11
		2-DDRMq	35.17	98.92	0.00		100.00	0.00		27.51	21.06	32.16	11.64	4.05	29.24	17.06
	0.33	Youden	3.72	51.32	59.06	1.32	51.24	48.87	2.58	18.02	56.47	52.38	12.60	56.95	44.04	20.30
		Brute force	-126.24	1.08	100.00	1.08	0.00	100.00	2.14	75.31	89.41	0.01	66.67	100.00	0.00	0.04
		Elkan	108.14	91.81	11.12	1.64	92.70	7.34	2.85	55.09	76.72	23.55	14.28	83.03	17.67	17.20
		Bayes	98.20	88.94	15.30	1.61	89.74	10.31	2.92	41.73	70.84	31.89	13.33	75.45	25.33	18.80
		2-DDR	126.24	98.92	0.00		100.00	0.00		75.27	89.41	0.00	0.00	100.00	0.00	
		2-DDRq	126.24	98.92	0.00		100.00	0.00		75.31	89.41	0.00		100.00	0.00	
		2-DDRM	126.24	98.92	0.00		100.00	0.00		75.34	89.41	0.00		100.00	0.00	
		2-DDRMq	126.24	98.92	0.00		100.00	0.00		75.31	89.41	0.00		100.00	0.00	

Table 3.11: Results for the simulated multi-class classification problems for the sample size $n = 100,000$ and bias $U = -1$ for the different thresholding algorithms introduced throughout the work.

			Umbalanced							Balanced						
σ	a	Thresholding	AER	Acc	Rec	PPV	Spec	PP	FS	AER	Acc	Rec	PPV	Spec	PP	FS
0.5	0.01	Youden	2.95	66.52	67.97	2.11	66.50	33.86	4.10	15.26	65.95	66.90	18.76	65.85	37.60	29.28
		Brute force	0.28	1.05	100.00	1.05	0.00	100.00	2.08	-38.58	89.47	0.01	100.00	100.00	0.00	0.04
		Elkan	-0.27	98.94	0.19	14.74	99.99	0.01	0.59	-27.40	88.13	13.50	36.33	96.91	4.18	20.72
		Bayes	1.79	93.37	21.02	3.68	94.14	6.02	6.26	34.57	46.30	77.42	13.71	42.63	59.48	23.29
		2-DDR	0.27	1.05	99.83	1.05	0.00	99.99	2.08	38.58	10.53	99.99	10.53	0.00	100.00	19.05
		2-DDRq	0.28	11.95	90.05	1.07	11.11	88.90	2.11	38.55	12.50	97.57	10.54	2.48	97.52	19.02
		2-DDRM	1.03	1.13	91.99	1.13	0.01	86.04	2.22	38.51	10.53	99.97	10.53	0.00	99.95	19.06
		2-DDRMq	1.12	12.25	85.27	1.17	9.82	76.69	2.31	38.51	12.59	97.46	10.62	2.48	96.68	19.15
	0.10	Youden	16.26	69.14	63.88	2.27	69.20	31.16	4.38	25.76	65.88	67.00	18.68	65.75	37.69	29.20
		Brute force	-35.00	1.10	100.00	1.10	0.00	100.00	2.17	-6.56	89.51	0.00	0.00	100.00	0.00	
		Elkan	35.00	98.90	0.00	0.00	100.00	0.00		1.43	88.49	11.77	43.09	97.48	3.50	16.25
		Bayes	35.08	98.65	3.13	10.75	99.71	0.32	4.85	24.94	73.30	52.72	20.28	75.71	27.27	29.29
		2-DDR	34.99	98.90	0.00	0.00	100.00	0.00		6.55	10.49	99.98	10.49	0.00	99.99	18.99
		2-DDRq	35.00	98.90	0.00		100.00	0.00		6.55	12.64	97.34	10.50	2.71	97.29	18.95
		2-DDRM	35.00	98.90	0.00		100.00	0.00		35.38	15.77	63.19	15.76	0.00	42.07	25.23
		2-DDRMq	35.00	98.90	0.00		100.00	0.00		35.73	27.32	57.89	17.62	5.33	34.73	26.93
	0.33	Youden	47.60	67.82	67.21	2.28	67.83	32.55	4.41	49.99	65.41	67.73	18.58	65.14	38.31	29.12
		Brute force	-126.28	1.10	100.00	1.10	0.00	100.00	2.18	76.14	89.52	0.00	0.00	100.00	0.00	
		Elkan	126.29	98.90	0.00	0.00	100.00	0.00		78.42	89.31	5.34	52.51	99.13	1.34	9.03
		Bayes	126.26	98.89	0.16	12.42	99.99	0.01	0.49	80.47	86.65	21.65	30.61	94.26	7.41	25.37
		2-DDR	126.29	98.90	0.00		100.00	0.00		76.14	89.52	0.00		100.00	0.00	
		2-DDRq	126.29	98.90	0.00		100.00	0.00		76.14	89.52	0.00		100.00	0.00	
		2-DDRM	126.29	98.90	0.00		100.00	0.00		76.15	89.52	0.00		100.00	0.00	
		2-DDRMq	126.29	98.90	0.00		100.00	0.00		76.14	89.52	0.00		100.00	0.00	
4	0.01	Youden	0.81	58.03	51.00	1.28	58.10	41.99	2.50	4.11	54.17	54.81	12.26	54.10	46.83	20.03
		Brute force	0.20	1.04	100.00	1.04	0.00	100.00	2.06	-38.50	89.52	0.00	50.00	100.00	0.00	0.04
		Elkan	0.35	84.24	20.54	1.41	84.91	15.15	2.63	-7.91	65.76	39.08	12.87	68.87	31.97	19.16
		Bayes	0.70	70.17	37.23	1.31	70.52	29.56	2.54	13.75	50.22	59.34	12.02	49.15	51.74	19.98
		2-DDR	0.13	25.15	74.68	1.03	24.61	75.38	2.04	31.03	15.23	90.50	10.17	6.42	93.26	18.28
		2-DDRq	0.19	23.99	77.93	1.06	23.42	76.59	2.09	38.47	12.46	97.50	10.48	2.50	97.50	18.92
		2-DDRM	0.47	28.92	65.99	1.05	25.85	65.61	2.07	30.99	15.23	90.49	10.17	6.42	93.24	18.28
		2-DDRMq	0.51	23.75	71.90	1.09	20.59	68.85	2.14	37.86	12.59	96.21	10.57	2.50	95.38	19.04
	0.10	Youden	1.88	51.58	56.54	1.26	51.52	48.56	2.46	6.47	55.12	52.97	12.25	55.39	45.49	19.88
		Brute force	-35.13	1.07	100.00	1.07	0.00	100.00	2.11	-6.72	89.48	0.00	25.00	100.00	0.00	0.04
		Elkan	25.82	85.29	19.20	1.47	86.00	14.05	2.73	3.10	68.72	34.33	12.91	72.75	27.99	19.52
		Bayes	19.14	78.36	28.55	1.44	78.89	21.19	2.74	6.61	58.11	49.20	12.40	59.16	41.72	19.81
		2-DDR	35.13	98.93	0.00	0.00	100.00	0.00		5.95	19.03	87.57	10.37	10.97	88.88	18.53
		2-DDRq	35.13	98.93	0.00		100.00	0.00		6.71	12.47	97.50	10.51	2.48	97.52	18.98
		2-DDRM	35.13	98.93	0.00		100.00	0.00		26.56	36.23	24.56	11.50	10.97	23.66	14.73
		2-DDRMq	35.13	98.93	0.00		100.00	0.00		27.43	24.78	29.19	11.33	5.61	27.08	16.24
	0.33	Youden	9.16	53.10	56.71	1.29	53.06	47.04	2.51	16.08	55.52	52.25	12.24	55.91	44.94	19.79
		Brute force	-126.59	1.07	100.00	1.07	0.00	100.00	2.11	76.52	89.50	0.00	50.00	100.00	0.00	0.04
		Elkan	97.72	87.68	16.63	1.53	88.45	11.61	2.79	42.91	69.89	33.04	13.08	74.21	26.55	18.72
		Bayes	85.09	84.09	21.56	1.50	84.76	15.31	2.80	29.93	64.51	40.25	12.64	67.35	33.45	19.23
		2-DDR	126.59	98.93	0.00		100.00	0.00		76.53	89.50	0.00		100.00	0.00	
		2-DDRq	126.59	98.93	0.00		100.00	0.00		76.53	89.50	0.00		100.00	0.00	
		2-DDRM	126.59	98.93	0.00		100.00	0.00		76.53	89.50	0.00		100.00	0.00	
		2-DDRMq	126.59	98.93	0.00		100.00	0.00		76.53	89.50	0.00		100.00	0.00	

Table 3.12: Results for the simulated multi-class classification problems for the sample size $n = 100,000$ and bias $U = 0$ for the different thresholding algorithms introduced throughout the work.

			Umbalanced							Balanced						
σ	a	Thresholding	AER	Acc	Rec	PPV	Spec	PP	FS	AER	Acc	Rec	PPV	Spec	PP	FS
0.5	0.01	Youden	2.83	68.28	65.77	2.23	68.31	32.06	4.32	16.19	64.87	67.99	18.51	64.50	38.93	29.08
		Brute force	0.31	1.08	100.00	1.08	0.00	100.00	2.14	-38.64	89.42	0.01	100.00	100.00	0.00	0.08
		Elkan	-0.19	98.73	2.02	9.00	99.78	0.24	3.96	5.31	73.71	54.33	21.63	76.00	27.21	30.46
		Bayes	2.63	78.66	45.93	2.34	79.01	21.26	4.45	37.87	28.55	89.11	11.83	21.38	79.73	20.88
		2-DDR	0.32	4.69	96.25	1.08	3.69	96.31	2.13	38.62	10.58	99.96	10.58	0.00	99.99	19.13
		2-DDRq	0.28	24.76	74.62	1.06	24.22	75.77	2.09	38.62	12.56	97.38	10.57	2.53	97.46	19.08
		2-DDRM	1.05	5.25	90.52	1.15	3.69	85.20	2.27	38.54	10.60	99.91	10.60	0.00	99.78	19.16
		2-DDRMq	1.04	29.12	67.64	1.18	24.59	61.73	2.32	38.54	12.64	97.26	10.64	2.53	96.74	19.18
	0.10	Youden	15.55	68.14	65.03	2.20	68.17	32.19	4.25	25.77	66.22	66.67	18.77	66.17	37.28	29.28
		Brute force	-35.17	1.08	100.00	1.07	0.00	100.00	2.13	-6.51	89.51	0.01	66.67	100.00	0.00	0.06
		Elkan	35.16	98.85	0.92	13.16	99.92	0.09	1.85	16.61	80.42	39.87	26.35	85.18	17.44	28.89
		Bayes	33.45	96.18	14.06	4.95	97.07	3.05	7.32	21.67	48.27	80.05	14.46	44.55	58.03	24.50
		2-DDR	35.16	98.93	0.00	0.00	100.00	0.00		6.47	10.48	99.94	10.48	0.00	99.99	18.97
		2-DDRq	35.17	98.93	0.00		100.00	0.00		6.50	12.46	97.41	10.48	2.51	97.48	18.92
		2-DDRM	35.17	98.93	0.00		100.00	0.00		35.70	15.64	63.85	15.64	0.00	42.83	25.12
		2-DDRMq	35.17	98.93	0.00		100.00	0.00		36.00	23.50	58.91	17.29	3.37	35.75	26.73
	0.33	Youden	46.52	67.23	66.91	2.19	67.24	33.13	4.24	51.91	65.81	67.20	18.79	65.64	37.83	29.36
		Brute force	-126.44	1.08	100.00	1.08	0.00	100.00	2.13	75.81	89.43	0.00	100.00	100.00	0.00	0.04
		Elkan	126.43	98.91	0.25	9.06	99.98	0.02	0.65	79.82	85.42	25.49	32.06	92.51	9.39	25.27
		Bayes	125.95	98.70	3.09	11.56	99.74	0.29	4.87	58.24	72.94	54.88	20.65	75.08	28.09	30.00
		2-DDR	126.44	98.92	0.00		100.00	0.00		75.79	89.43	0.01	40.00	100.00	0.01	0.08
		2-DDRq	126.44	98.92	0.00		100.00	0.00		75.81	89.43	0.00		100.00	0.00	
		2-DDRM	126.44	98.92	0.00		100.00	0.00		75.86	89.43	0.00		100.00	0.00	
		2-DDRMq	126.44	98.92	0.00		100.00	0.00		75.81	89.43	0.00		100.00	0.00	
4	0.01	Youden	0.74	53.81	54.91	1.25	53.80	46.30	2.43	4.49	53.96	54.75	12.16	53.86	47.04	19.88
		Brute force	0.31	1.05	100.00	1.05	0.00	100.00	2.08	-37.73	89.57	0.00	0.00	100.00	0.00	
		Elkan	0.22	83.09	21.63	1.41	83.74	16.32	2.63	-1.11	59.56	47.59	12.47	60.96	39.94	19.68
		Bayes	0.73	61.79	46.50	1.28	61.95	38.14	2.49	19.70	43.04	67.25	11.59	40.22	60.56	19.77
		2-DDR	0.21	32.63	66.75	1.02	32.26	67.73	1.99	26.28	17.94	85.32	9.95	10.09	89.43	17.83
		2-DDRq	0.31	15.58	85.65	1.06	14.83	85.17	2.08	37.71	12.41	97.48	10.43	2.50	97.50	18.84
		2-DDRM	0.41	35.20	60.75	1.04	32.45	61.50	1.96	26.25	17.94	85.31	9.95	10.09	89.42	17.83
		2-DDRMq	0.50	15.41	78.57	1.05	13.37	78.40	2.07	37.08	12.54	96.20	10.53	2.50	95.35	18.98
	0.10	Youden	7.58	59.34	50.07	1.35	59.43	40.67	2.63	6.82	55.28	53.07	12.40	55.53	45.38	20.04
		Brute force	-35.12	1.09	100.00	1.09	0.00	100.00	2.16	-6.44	89.42	0.00	33.33	100.00	0.00	0.04
		Elkan	21.89	79.44	27.57	1.50	80.01	20.07	2.84	5.55	60.72	45.95	12.71	62.47	38.42	19.82
		Bayes	13.51	70.93	37.40	1.42	71.30	28.79	2.73	8.13	50.60	58.74	12.13	49.63	51.25	20.11
		2-DDR	35.11	98.91	0.00	0.00	100.00	0.00		5.95	29.61	75.85	10.59	24.12	75.87	18.56
		2-DDRq	35.12	98.91	0.00		100.00	0.00		6.44	12.42	97.73	10.59	2.32	97.68	19.10
		2-DDRM	35.12	98.91	0.00		100.00	0.00		27.00	66.94	11.76	13.12	24.12	9.95	11.11
		2-DDRMq	35.12	98.91	0.00		100.00	0.00		27.58	22.94	30.25	11.64	4.66	27.60	16.72
	0.33	Youden	17.13	56.39	52.97	1.33	56.42	43.68	2.59	12.35	53.74	55.04	12.24	53.58	47.32	19.99
		Brute force	-126.37	1.09	100.00	1.09	0.00	100.00	2.16	75.56	89.51	0.00	0.00	100.00	0.00	
		Elkan	82.67	81.89	24.00	1.49	82.52	17.55	2.81	30.33	63.13	43.00	12.75	65.48	35.41	19.66
		Bayes	67.95	77.86	29.14	1.46	78.40	21.68	2.79	15.34	57.56	49.92	12.35	58.45	42.43	19.80
		2-DDR	126.37	98.91	0.00		100.00	0.00		75.54	89.50	0.01	17.14	100.00	0.00	0.04
		2-DDRq	126.37	98.91	0.00		100.00	0.00		75.56	89.51	0.00		100.00	0.00	
		2-DDRM	126.37	98.91	0.00		100.00	0.00		75.65	89.51	0.00		99.99	0.00	
		2-DDRMq	126.37	98.91	0.00		100.00	0.00		75.56	89.51	0.00		100.00	0.00	

Table 3.13: Results for the simulated multi-class classification problems for the sample size $n = 100,000$ and bias $U = 1$ for the different thresholding algorithms introduced throughout the work.

Chapter 4

Cost-sensitive parametric learning

4.1 Introduction

As previously introduced, classification is a supervised learning task involving the categorization of input data into predefined classes or categories based on their intrinsic features and attributes. These techniques are widely used across diverse domains such as economics, medicine, biology, and psychology, providing analytical tools for situations where decision-making processes are needed. Classification tasks are typically divided into two well-defined stages, as previously introduced. The first involves a classifier returning an estimated probability or score that measures the likelihood of an event of interest occurring. The second stage involves determining a prediction rule from the previously computed score to optimize a performance metric, e.g. accuracy.

Current standard classification models assume that all labeling errors have the same impact, fitting models by maximizing the likelihood or the accuracy over a validation set. However, this approach is suboptimal for problems where each observation has a different misclassification cost, as shown in Chapter 1.

Cost-sensitive (CS) classification addresses the problem of optimal learning and decision-making when different misclassification errors imply different costs. Thus,

classifiers are trained to recognize the different error costs, aligning them with the overall objective of minimizing losses, which is equivalent to maximize benefits. To attain this, there are two methodologies as presented in Chapter 1, predict-then-optimize and predict-and-optimize as introduced in Vanderschueren et al. (2022b). While there are no comprehensive comparative studies on the effectiveness of these two strategies, Dmochowski et al. (2010) shows that the *predict-and-optimize* approach is more effective under model misspecification, and Vanderschueren et al. (2022b); Höppner et al. (2021); C-Rella et al. (2024b) and Bahnsen et al. (2014a), among others, empirically demonstrates better performance of CS classifiers when considering a CS thresholding strategy.

The theoretical optimal CS threshold is the Bayes minimum risk rule introduced in equation (2.7), as noted in Höppner et al. (2021); Bahnsen et al. (2013) and Elkan (2001). Furthermore, an optimal empirical thresholding rule for given a score has been proposed in the Chapters 2 and 3, with good practical results. Therefore, the *predict-and-optimize* approach to estimate the score is the stream of literature which is pursued in the reminder of this work.

In this chapter it is considered a cost specification as the introduced in Table 1.1 for the fraud detection problem. However, the methodology introduced in this chapter can be extended to any cost setting. As introduced in Chapter 1, to measure model performance, *savings* is considered as an aggregated metric over a sample. The objective is to minimize the aggregated losses, $\sum_{i=1}^{n} \ell(\hat{y}_i, w_i, y_i)$, equivalently, to maximize the savings (1.2), along possible classifiers.

Regarding previous CS scoring approaches, Stripling et al. (2018) and Verbraken et al. (2012) focus on obtaining a more accurate cost-sensitive metric, expecting to achieve a better classification when using it for model fitting. Despite interesting, this stream of work is not considered as the results in this work can be applied considering any loss function. Among CS classifiers, it is worth highlighting the proposals of using weighted versions of logistic regression (see Bahnsen et al. (2014a); Höppner et al. (2021); Stripling et al. (2018)) or boosting algorithms (see Devi et al. (2019); Höppner et al. (2021)), among others. Although these methods achieve

satisfactory results, many of them are heuristic, task-dependent and often tested on a limited number of (typically proprietary) data sets. This makes it impossible to replicate findings and understand how they relate to each other as noted in Vanderschueren et al. (2022b). As a result, although good practical performance has been demonstrated on real data sets, there is a lack of evidence to support the use of these models in general CS scenarios.

In model fitting, the inclusion of costs often makes the objective function non-convex, making traditional convex optimization methods impractical. While genetic algorithms have been suggested as a means to address non-convexity (see Stripling et al. (2018)), this chapter proposes strategies that consider differentiable functions to handle non-convexity. This allows the optimization problem to be solved using classical gradient descent algorithms, eliminating the need for further exploration of optimization methods. Similarly, robust versions have been proposed in De Vos et al. (2023). However, non-convexity is usually not caused by the presence of outliers, but rather by the asymmetric distribution of the variables.

The main goal of this chapter is to provide theoretical grounds and practical justification for CS parameter estimation in binary classification tasks considering the proposals in Höppner et al. (2021) and Bahnsen et al. (2014a). The consistency and asymptotic distribution of the proposed CS estimators are obtained and their behavior is evaluated through extensive simulations and the analysis of two real datasets. Different simulation models and cost settings are considered, performing for the first time, as far as we know, a deep study of the performance of CS classifiers where the underlying structure of the problem is known. This provides a deeper understanding of the proposed estimators behavior depending on the distribution that generates the data, the degree of misspecification in the model and the cost specification. The proposed approach can be extended to various domains, including customer churn prediction (Stripling et al. (2018)), credit risk (Bahnsen et al. (2014a); Höppner et al. (2021); Vanderschueren et al. (2022a)), medical diagnosis, choice models, and logistic planning (Elmachtoub and Grigas (2020)), among others.

4.2 Cost-sensitive parameter estimation

Given a loss function ℓ as the one introduced in equation (1.1), the goal is to minimize $E[\ell(\hat{Y}, W, Y)]$ over $\hat{Y}$, which depends on the function s and the threshold h through $\hat{Y} = \mathbb{I}(s(\mathbf{x}) > h)$. The best classifier s independent of the thresholding strategy is sought, for which it is assumed that $\hat{Y} = \mathbb{I}(s(\mathbf{x}) > u)$, with $u \sim U[0, 1]$. Thus, $E[\hat{Y} \mid \mathbf{x}] = P(\hat{Y} = 1 \mid \mathbf{x}) = s(\mathbf{x})$. In practice, the true joint probability distribution of $\mathbf{Z} = (\mathbf{X}, W, Y)$ is unknown and therefore the expectation $E[\ell(\hat{Y}, W, Y)]$ is also unknown. Consequently, the learner has to rely on the empirical density to minimize the risk given the available training data, $\mathcal{Z}$.

Let $\mathcal{Z} = \{\mathbf{z}_i = (\mathbf{x}_i, w_i, y_i)\}_{i=1}^n$ be a set of n independent and identically distributed (i.i.d.) observations of the predictor, response and exogenous variables. For model fitting, the *average expected cost* (AEC), as proposed in Höppner et al. (2021); Verbraken et al. (2012) and Stripling et al. (2018), of a classifier s over the set $\mathcal{Z}$ is defined as:

$$AEC(s(\mathcal{Z})) = \frac{1}{n} E\Big[\sum_{i=1}^{n} \ell(\hat{y}_i, w_i, y_i) \mid \mathcal{Z}\Big] = \frac{1}{n} \sum_{i=1}^{n} \ell(s(\mathbf{x}_i), w_i, y_i) \tag{4.1}$$

where ℓ is the loss function introduced in equation (1.1)

Considering a parametric model, $s_\theta(\mathbf{x}) = s(\theta, \mathbf{x})$, e.g. logistic regression, where $s_\theta(\mathbf{x}) = (1 + e^{-\theta'\mathbf{x}})^{-1}$, the objective function on the right-hand side of equation (4.1) depends only on θ. Thus, the AEC can be expressed as:

$$\mathcal{L}_n(\theta) = \frac{1}{n} \sum_{i=1}^{n} l_i(\theta), \;\; l_i(\theta) = l(\theta, \mathbf{z}_i) = \ell(s_\theta(\mathbf{x}_i), w_i, y_i) = c y_i w_i + \xi_i s_\theta(\mathbf{x}_i) \tag{4.2}$$

where $\xi_i = -c y_i w_i + (1 - y_i) w_i a + b$ for the loss function introduced in (1.1).

The objective in CS classification is to obtain the population parameter, θ^*, that minimizes the expected loss:

$$\theta^* = \arg\min_{\theta \in \Theta} \mathcal{L}(\theta) \tag{4.3}$$

where $\mathcal{L}(\theta) = E[l(\theta, \mathbf{Z})] = E[\mathcal{L}_n(\theta)]$ and $\Theta \subset \mathbb{R}^d$. Since the unconditional expected loss, $\mathcal{L}$, cannot be computed, θ^* is estimated with the parameter that minimizes the

sample version of $\mathcal{L}$, the AEC:

$$\widehat{\theta}_n = \arg\min_{\theta \in \Theta} \mathcal{L}_n(\theta) \tag{4.4}$$

4.3 Regularized cost-sensitive classification

In general, neither the objective function, $\mathcal{L}_n$, nor the functions l_i have a unique minimum, nor are they always convex due to their dependence on W, Y, and θ through $\hat{Y}$. To address this issue and to allow the use of conventional gradient descent optimization algorithms, a regularizer is added to the objective function, following a standard practice in the machine learning literature.

4.3.1 Objective function convexity

An example of the non-convex shape of the objective function is shown in Figure 4.1, which is a contour plot of the objective function, $\mathcal{L}_n(\theta)$ as defined in equation (4.2), as a function of $\theta \in \mathbb{R}^2$ for the Klein & Spady model and the cost setting $a = 0$ presented in Section 4.5. The plot depends on $\theta \in \mathbb{R}^2$ for a simulated covariate vector $\mathbf{X} \in \mathbb{R}^2$ and Y modeled as a function of $\mathbf{X}$ by a logistic model. Lighter regions represent lower values of the AEC and darker regions represent higher values. The objective is to find the minimizer of $\mathcal{L}_n(\theta)$, which is located in the lighter regions of the Figure 4.1. It can be seen that the minimum is not reached within the considered support and that in the upper right part of the graph the function does not have a convex shape. Therefore there is no a global minimum defined for this function. In Figure 4.1, the maximum likelihood estimator is represented by a black dot. Note that this point does not coincide with anywhere near the clearest region in the AEC contour plot, demonstrating that maximizing likelihood and minimizing loss are not necessarily equivalent goals.

To analytically check the convexity of the objective function (4.2), the positive definiteness of its Hessian matrix, $H\mathcal{L}_n$, is investigated. For the logistic regression

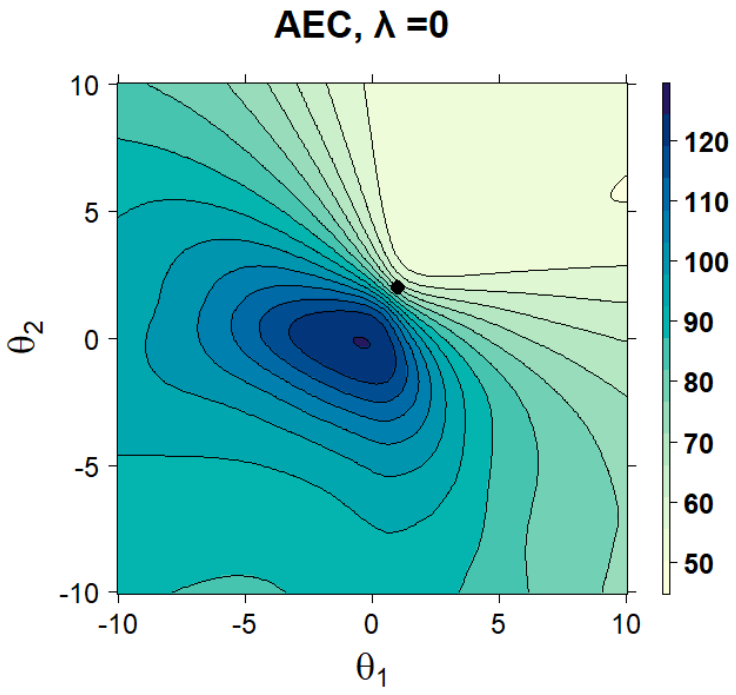

Figure 4.1: AEC (4.2) with respect to $\theta = (\theta_1, \theta_2)$ considering a logistic model for the Klein & Spady model simulation in Section 4.5. The black dot represents the maximum likelihood estimator.

case, the jk element of $H\mathcal{L}_n$ is equal to $\sum_{i=1}^{n} s_\theta(x_i)(1 - s_\theta(x_i))(1 - 2s_\theta(x_i))x_{ij}x'_{ik}\xi_i$, for $j, k \in \{1, \cdots, d\}$. The $d \times d$ matrix $x_i x'_i$ is positive semidefinite. The function $\Psi(u) = u(1-u)(1-2u)$ is zero for $u = 0, 0.5, 1$, $\Psi(u) > 0$ for $u \in (0, 0.5)$ and $\Psi(u) < 0$ for $u \in (0.5, 1)$. Also, $\xi_i > 0$ when $Y_i = 0$ and $\xi_i < 0$ when $Y_i = 1$. Therefore, there is no guarantee that the Hessian is positive or negative definite for all parameter values. Likewise, it is difficult to find a criterion that guarantees the convexity of the objective function or a region of $\Theta \subset \mathbb{R}^d$ where the loss function (4.2) is convex. A possible solution is to compute $H\mathcal{L}_n(\theta)$ for a grid in Θ and consider the convex hull of each connected region where the eigenvalues of $H\mathcal{L}_n$ are positive. If there is more than one connected component, an initial point in each region can be considered for optimization and finally the solution with the minimum value of the loss function is taken. This approach will not be explored further as another solution to deal with non-convexity is presented in the following subsection.

4.3.2 Regularized objective function

A regularizer is a convex function of the model parameters that aids in optimizing non-convex cost functions by enforcing a certain structure in the solution. A regularization component is introduced into the loss function (4.2) to enforce convexity as suggested in Höppner et al. (2021); Stripling et al. (2018) and Bahnsen et al. (2013). This facilitates numerical optimization techniques to avoid suboptimal solutions at saddle points or flat regions. Likewise, it can be viewed as a mechanism to control the complexity of the model by restricting the space of admissible solutions, thereby preventing overfitting and improving generalization performance. By introducing a regularizer into the loss function (4.2), the penalized loss function is defined as:

$$\mathcal{L}_{n,\lambda}(\theta) = \frac{1}{n}\sum_{i=1}^{n} l_i(\theta) + \mathrm{P}_\lambda(\theta), \tag{4.5}$$

where $\mathrm{P}_\lambda(\theta)$ is a regularizer that depends on a parameter $\lambda \in \mathbb{R}^+$. The regularization effect can be seen in Figure 4.2, which represents contour plots of the regularized AEC (4.5) as in Figure 4.1. It is considered $\mathrm{P}_\lambda(\theta) = \lambda\|\theta\|_1$ for different values of λ. Further regularizers could be considered, but since this is the one proposed in Höppner et al. (2021) and it is a common choice in the machine learning literature (lasso regularization), no other regularizers are explored, without loss of generality. The top left graph in Figure 4.2, which corresponds to the unregularized objective function shown in Figure 4.1, is not convex and has a local minimum near the global maximum. This makes up a truly difficult scenario to obtain the AEC (4.1) minimizer. By adding $\mathrm{P}_\lambda(\theta)$ to the objective function and increasing the regularization parameter (λ), the objective function becomes convex with $\lambda \geq 0.1$. Consequently, the optimization problem can be solved considering the regularized objective function (4.5), but at the cost of introducing bias in the parameter estimation. A bound on this bias is given in Section 4.4.

As in Section 4.2, the objective is to find the population parameter minimizing the expected cost:

$$\theta^*_\lambda = \arg\min_{\|\theta\|_1 < R} \mathcal{L}_\lambda(\theta). \tag{4.6}$$

where $\mathcal{L}_\lambda(\theta) = E[\mathcal{L}_{n,\lambda}(\theta)]$. It is considered the constrained optimization problem

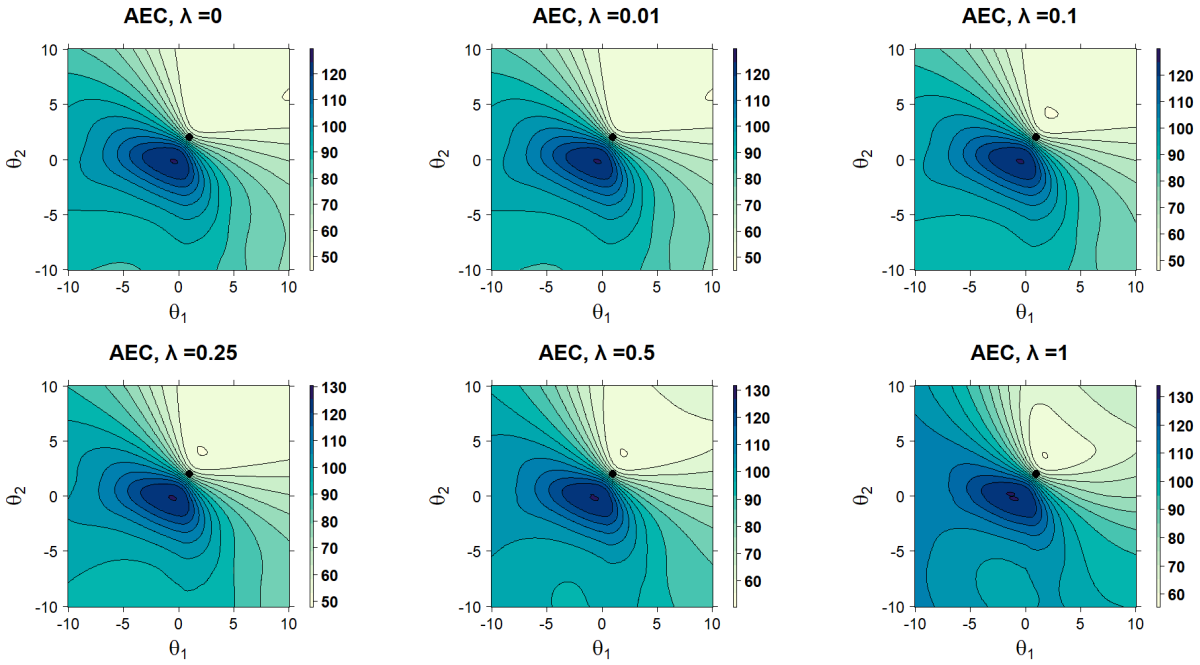

Figure 4.2: Regularized AEC (4.5) with $\mathrm{P}_\lambda(\theta) = \lambda\|\theta\|_1$ depending on $\theta = (\theta_1, \theta_2)$ considering a logistic model for the Klein & Spady model simulation in Section 4.5 and cost setting with $a = 0$. The black dot represents the maximum likelihood estimator.

with $\|\theta\|_1 < R$, for some $R > 0$, to simplify the optimization process. Again, the population value θ^*_λ in equation (4.6) cannot be obtained as we do not known the underlying joint distribution of the variables affecting $\mathcal{L}_{n,\lambda}$. Thus, θ^*_λ is estimated considering its sample version:

$$\widehat{\theta}_{n,\lambda} = \arg \min_{\|\theta\|_1 < R} \mathcal{L}_{n,\lambda}(\theta) \tag{4.7}$$

4.4 Properties of the AEC minimizer

Let's suppose $E[Y \mid \mathbf{X} = \mathbf{x}] = s_{\theta_0}(\mathbf{x})$, where θ_0 is an unknown parameter. In CS classification, the goal is to find the value θ^* that minimizes $\mathcal{L}_n$ in (4.2), which is unlikely to coincide with θ_0, the one that maximizes the expected likelihood. In this section, the consistency of $\widehat{\theta}_{n,\lambda}$ as an estimator of the expected loss minimizer, θ^*, and its asymptotic distribution are obtained. Note that θ^* as defined in (4.3) coincides with θ^*_0 as defined in equation (4.6) with $\lambda = 0$. Therefore, in this section

it is considered the regularized objective function (4.5) without loss of generality.

First of all, note that $\mathcal{L}_n$ converges uniformly to $\mathcal{L}$ by the uniform law of large numbers, where $\mathcal{L}$ can be expressed as:

$$\mathcal{L}(\theta) = E[\mathcal{L}_n(\theta)] = E\left[\frac{1}{n}\sum_{i=1}^{n} l_i(\theta)\right] = E[l_1(\theta)] = E[cy_1w_1 + \xi_1 s_\theta(\mathbf{x}_1)] =$$

$$cE[yw] - (c+a)E[w]\int s_\theta(\mathbf{x})s_{\theta_0}(\mathbf{x})dF(\mathbf{x}) + (E[w]a+b)\int s_\theta(\mathbf{x})dF(\mathbf{x})$$

and $F(\mathbf{x})$ is the distribution function of $\mathbf{X}$. In the third equality it is used that $\mathcal{Z} = \{\mathbf{z}_i = (\mathbf{x}_i, w_i, y_i)\}_{i=1}^n$ is assumed i.i.d.. With analogous reasoning, $\mathcal{L}_{n,\lambda}(\theta)$ converges uniformly to $\mathcal{L}_\lambda(\theta)$. The problem is that even when the underlying distribution of $\mathbf{Z} = (\mathbf{X}, W, Y)$ is known, this expression cannot be calculated analytically in general, but it can be approximated by numerical integration. Assuming that θ^*_λ is a "well separated" minimum of $\mathcal{L}_\lambda$, the consistency of the proposed estimator, $\widehat{\theta}_{n,\lambda}$, can be obtained with the following Theorem, derived from Theorem 5.7 in Vaart (1998).

Theorem 4.1. *Assuming*

A1. $\forall \epsilon > 0$, $\inf_{\theta:\|\theta-\theta^*_\lambda\|_2 \geq \epsilon} \mathcal{L}_\lambda(\theta) > \mathcal{L}_\lambda(\theta^*)$.

A2. *There exists a sequence of random variables $\zeta_n = o_p(1)$ such that the sequence of estimators $\widehat{\theta}_{n,\lambda}$ satisfy $\mathcal{L}_{n,\lambda}(\widehat{\theta}_{n,\lambda}) \leq \mathcal{L}_{n,\lambda}(\theta^*_\lambda) + \zeta_n$.*

Then $\widehat{\theta}_{n,\lambda}$ converges in probability to θ^*_λ.

Proof. The function $\mathcal{L}_{n,\lambda}$ is defined over a compact set, Θ. Furthermore, ℓ is continuous in $\mathbf{Z}$ and a measurable function of $\mathbf{Z}$ at each $\theta \in \Theta$. In addition, there exists a function M with $E[M(\mathbf{Z})] < \infty$ such that $\mid l(\theta, \mathbf{z}) \mid \leq M(\mathbf{z})\ \forall \theta \in \Theta$ given that l is bounded by $cW + \xi$. Therefore, the uniform law of large numbers can be applied to prove:

$$\sup_{\theta\in\Theta} \mid \mathcal{L}_{n,\lambda}(\theta) - \mathcal{L}_\lambda(\theta) \mid \xrightarrow{\mathrm{P}} 0 \tag{4.8}$$

By Condition A2, $\mathcal{L}_{n,\lambda}(\widehat{\theta}_n) - \mathcal{L}_{n,\lambda}(\theta^*_\lambda) \leq \zeta_n = o_p(1)$, and by the uniform convergence of $\mathcal{L}_{n,\lambda}$ to $\mathcal{L}_\lambda$ (4.8), $\sup_{\theta\in\Theta} \mid \mathcal{L}_{n,\lambda}(\theta) - \mathcal{L}_\lambda(\theta) \mid \leq \delta_n$, with $\delta_n = o_p(1)$.

Then it follows, adding and subtracting $\mathcal{L}_{n,\lambda}(\theta^*)$ and $\mathcal{L}_{n,\lambda}(\widehat{\theta}_{n,\lambda})$:

$$\mathcal{L}_\lambda(\widehat{\theta}_{n,\lambda})-\mathcal{L}_\lambda(\theta^*_\lambda) = \mathcal{L}_\lambda(\widehat{\theta}_{n,\lambda})-\mathcal{L}_{n,\lambda}(\widehat{\theta}_{n,\lambda})+\Big(\mathcal{L}_{n,\lambda}(\theta^*_\lambda)-\mathcal{L}(\theta^*_\lambda)\Big)+\Big(\mathcal{L}_{n,\lambda}(\widehat{\theta}_{n,\lambda})-\mathcal{L}_{n,\lambda}(\theta^*_\lambda)\Big) \leq$$

$$\leq \mathcal{L}_\lambda(\widehat{\theta}_{n,\lambda}) - \mathcal{L}_{n,\lambda}(\widehat{\theta}_{n,\lambda}) + \zeta_n + \delta_n \leq 2\delta_n + \zeta_n \xrightarrow{\mathrm{P}} 0 \tag{4.9}$$

By Assumption A1, $\forall \epsilon > 0$, $\exists \mu > 0$ such that $\mathcal{L}_\lambda(\theta) > \mathcal{L}_\lambda(\theta^*_\lambda) + \mu$ for every θ satisfying $\|\theta - \theta^*_\lambda\|_2 \geq \epsilon$, so the event $\{\|\widehat{\theta}_{n,\lambda} - \theta^*_\lambda\|_2 \geq \epsilon\}$ is contained in the event $\{\mathcal{L}_\lambda(\widehat{\theta}_{n,\lambda}) > \mathcal{L}_\lambda(\theta^*_\lambda) + \mu\}$. The probability of the latter event converges to 0 as a consequence of (4.9). Thus, $P\left(\|\widehat{\theta}_{n,\lambda} - \theta^*_\lambda\|_2 \geq \epsilon\right) \to 0$ □

Condition A1 guarantees that the minimum is "well separated", in the sense that $\mathcal{L}_\lambda$ has a unique minimizer, θ^*_λ, and only parameters close to θ^*_λ attain a value $\mathcal{L}_\lambda(\theta)$ near $\mathcal{L}_\lambda(\theta^*_\lambda)$. In Vaart (1998) it is suggested to check this condition with the plot of the *AEC* (4.2) with respect to θ, as represented in Figure 4.2.

Consider the M-estimator for θ^*_λ defined in (4.7), which is consistent whenever the assumptions of Theorem 4.1 hold. Now, considering Theorem 5.23 in Vaart (1998), the asymptotic distribution of $\widehat{\theta}_{n,\lambda}$ is given in the following theorem. This result can be used to make inference about the estimated parameter from a cost-sensitive perspective.

Theorem 4.2. *Assume that $\mathcal{L}_\lambda(\theta)$ in (4.5) admits a second-order Taylor expansion at a point of minimum θ^*_λ with a nonsingular second derivative matrix $H_{\theta^*_\lambda}$:*

$$\mathcal{L}_\lambda(\theta) = \mathcal{L}_\lambda(\theta^*_\lambda) + \frac{1}{2}(\theta - \theta^*_\lambda)' H_{\theta^*_\lambda}(\theta - \theta^*_\lambda) + o(\|\theta - \theta^*_\lambda\|_2^2)$$

If $\mathcal{L}_{n,\lambda}(\widehat{\theta}_{n,\lambda}) \leq \inf_\theta \mathcal{L}_{n,\lambda}(\theta) + o_p(n^{-1})$ and $\widehat{\theta}_{n,\lambda} \xrightarrow{P} \theta^*_\lambda$, then

$$\sqrt{n}(\widehat{\theta}_{n,\lambda} - \theta^*_\lambda) \xrightarrow{d} N(0, H^{-1}_{\theta^*} V_{\theta^*} H^{-1}_{\theta^*})$$

where V_{θ^*} is the variance-covariance matrix of $l'(\theta^*_\lambda, \mathbf{z})$, where $l'(\theta^*_\lambda, \mathbf{z}) = \frac{\partial l(\theta, \mathbf{z})}{\partial \theta}$ is the gradient of l, where l is as defined as in (4.2).

Proof. Let $\theta \in A$, with A an open subset of $\mathbb{R}^d$. The function $l_\theta(\mathbf{z}) = l(\theta, \mathbf{z})$ is continuous (consequently measurable). The function $l_{\mathbf{z}}(\theta) = l(\theta, \mathbf{z})$ is differentiable

at θ^*_λ for every $\mathbf{z}$. In addition, for every θ_1 and θ_2 in a neighborhood of θ^*_λ,

$$\mid l(\theta_1, \mathbf{z}) - l(\theta_2, \mathbf{z}) \mid \leq l'(\tilde{\theta}, \mathbf{z}) \|\theta_1 - \theta_2\|_2, \text{for some } \tilde{\theta} \text{ between } \theta_1 \text{ and } \theta_2.$$

Then, the conditions of Theorem 5.23 in Vaart (1998) are satisfied with $m_\theta(\mathbf{x}) = -l(\theta, \mathbf{x})$, $\theta_0 = \theta^*_\lambda$ and $Pm_\theta = \mathcal{L}_\lambda(\theta)$, which leads to the demonstration of the theorem. □

It is worth noting that the previous theorems can be considered for any regularizer, P_λ, and parametric model, s_θ. Since conditions are only imposed on the regularized objective function, $\mathcal{L}_{n,\lambda}$ (4.5), for a given objective function, $\mathcal{L}_n$ (4.2), the regularizer can be chosen to satisfy the assumptions in Theorems 4.1-4.2. Thus, Theorems 4.1-4.2 can be applied to any CS parametric estimator.

4.4.1 Regularized estimator properties

As explained in Section 4.3.1, the objective function $\mathcal{L}_n$ is usually non-convex and it is not expected that θ^* is a well-separated minimum of $\mathcal{L}$ in general, which is needed in order to apply Theorems 4.1 and 4.2. In addition, these conditions are more difficult to check when $d > 2$. These drawbacks are avoided by considering the regularized objective function (4.5) with $\lambda > 0$. However, this comes at the cost of introducing some bias in the parameter estimation, which needs to be measured.

As one would expect, the bias depends on the regularization introduced and on the degree of non-convexity present in the objective function (4.2). Rewriting Theorem 1 in Loh and Wainwright (2015), a theorem bounding the introduced bias is presented below. It also gives a criterion for choosing the parameter λ.

Assume, as in Loh and Wainwright (2015), that the unregularized objective function (4.2) satisfies the *restricted strong convexity* (RSC) conditions:

A3.

$$\langle \nabla\mathcal{L}_n(\theta^* + \triangle) - \nabla\mathcal{L}_n(\theta^*), \triangle \rangle \geq \begin{cases} \alpha_1 \|\triangle\|_2^2 - \tau_1 \frac{\log d}{n} \|\triangle\|_1^2, & \|\triangle\|_2 \leq 1 \\ \alpha_2 \|\triangle\|_2 - \tau_2 \sqrt{\frac{\log d}{n}} \|\triangle\|_1, & \|\triangle\|_2 \geq 1 \end{cases}$$

where $\triangle \in \mathbb{R}^d$ *and* α_j *are strictly positive and* τ_j *nonnegative constants.*

Assumption A3 implies that although the objective function, $\mathcal{L}_n$, is not globally convex, it is so in most directions, and in those in where it is not, its convexity can be forced by adding some slack terms, defined by τ_i. In fact, if $\mathcal{L}_n$ is convex, the RSC conditions are satisfied with $\tau_i = 0$.

Furthermore, Loh and Wainwright (2015) assume that the regularizer, P_λ, is "separable" across dimensions, i.e. that it can be expressed as $\mathrm{P}_\lambda(\theta) = \sum_{i=1}^d \rho_\lambda(\theta_i)$, satisfying:

A4.

(*i*) $\rho_\lambda(0) = 0$ *and* $\rho_\lambda(t) = \rho_\lambda(-t), \forall t \in \mathbb{R}$.

(*ii*) $\rho_\lambda(t)$ *is nondecreasing in* $\mathbb{R}^+$.

(*iii*) *For* $t > 0, \rho_\lambda(t)/t$ *is nonincreasing in* t.

(*iv*) $\rho_\lambda(t)$ *is differentiable for all* $t \neq 0$ *and subdifferentiable at* $t = 0$, *with nonzero subgradients at* $t = 0$ *bounded by* λL.

(*v*) *There exists* $\mu > 0$ *s.t.* $\rho_{\lambda,\mu}(t) := \rho_\lambda(t) + \mu t^2$ *is convex.*

Lastly, consider the *first-order necessary condition* to be a local minimum of $\mathcal{L}_{n,\lambda}$:

A5. $\langle\, \nabla \mathcal{L}_{n,\lambda}(\theta^*_\lambda),\ \theta - \theta^*_\lambda \,\rangle \geq 0, \quad \forall \theta : \|\theta\|_1 < R$

Theorem 4.3. *Assume the regularizer* ρ_λ *satisfies conditions A4,* $\mathcal{L}_n$ *satisfies the RSC conditions A3 with* $\alpha_1 > \mu$, *and* θ^*_λ *is feasible for the objective. Consider any* λ *s.t.*

$$\frac{2}{L} \max \left\{ \|\nabla \mathcal{L}_n(\theta^*)\|_\infty,\ \alpha_2 \sqrt{\frac{\log d}{n}} \right\} \leq \lambda \leq \frac{\alpha_2}{6RL} \tag{4.10}$$

and assume that $n \geq \frac{16R^2 \max\{\tau_1^2, \tau_2^2\}}{\alpha_2^2} \log d$. *Then any vector* $\widehat{\theta}_{n,\lambda}$ *satisfying the first-order necessary condition A5 satisfies the error bounds:*

$$\|\widehat{\theta}_{n,\lambda} - \theta^*\|_2 \leq \frac{7\lambda L \sqrt{\|\theta^*\|_0}}{4(\alpha_1 - \mu)}, \tag{4.11}$$

$$\|\widehat{\theta}_{n,\lambda} - \theta^*\|_1 \leq \frac{56\lambda L \|\theta^*\|_0}{4(\alpha_1 - \mu)}$$

Proof. It suffices to consider Theorem 1 in Loh and Wainwright (2015) with $\rho_\lambda = \mathrm{P}_\lambda$ and $\beta^* = \theta^*$. □

The parameters that satisfy the conditions A3-A5 are not unique, so by playing with their values, different bounds can be considered for different values of λ in 4.10. The conditions A4 on the penalization function are satisfied for a wide range of regularizers. These include those commonly used in the machine learning literature, such as in our case the penalization function $\rho_\lambda(\theta) = \lambda \sum_{j=1}^{d} \|\theta_j\|_1$.

Increasing the level of regularization comes at the cost of increasing the bias bounds in (4.11). The bounds depend directly on λ, so that the more regularization is imposed on the problem, the larger the bias. They also have an inversely proportional relationship with α_1, which measures the degree of convexity of the objective function, $\mathcal{L}_n$ in equation (4.2), near the true minimum, θ^* (4.3). Thus, the bounds are larger when $\mathcal{L}_n$ is flat near θ^*. On the other hand, the bias bounds grow as the parameter μ increases, which indicates the degree of convexity that must be added to the regularizer for it to satisfy the convexity condition A4 (v). Regarding the required sample size, it grows with τ_i, indicating that larger samples are needed as $\mathcal{L}_n$ deviates from convexity. It also depends on the dimension of the problem and on $1/\alpha_2^2$, so that larger sample sizes are needed when $\mathcal{L}_n$ is non-convex far from the true minimum θ^*, which increases the chances of finding a local minimum or of the optimization algorithm not converging.

For the example in Figure 4.2, which corresponds to the Klein & Spady model and the cost setting $a = 0$ in Section 4.5, the parameters for the assumptions A3-A5 are calculated for different values of λ. In Figure 4.3, the required sample size, represented as $log(n)$, and the L_2-norm bias bound in equation (4.11) are obtained following Theorem 4.3 and represented along the savings (1.2) over a test set as a function of λ. The introduced bias grows exponentially with λ within a reasonable range. Considering a low regularization (small λ), it is guaranteed that the estimator $\widehat{\theta}_{n,\lambda}$ is close to the true minimizer θ^*. More worrying is the required sample size, which reaches billions for small values of λ. As the degree of regularization increases,

the objective function becomes more convex and consequently a smaller sample size is required, as can be seen in Figure 4.3. The savings (1.2) increase with λ thanks to the help provided by the regularizer, so that the optimization algorithm converges more easily. Thus, for this problem, if a large sample size is available ($n \geq 1,788,264$) with a choice of $\lambda = 2$, the consistency of the estimator is guaranteed and it is close to the true minimizer. In particular, $\|\widehat{\theta}_{n,\lambda} - \theta^*\|_2 \leq 0.12$. If a large sample size is not available, considering $\lambda = 10$, with $n \geq 71,531$, the estimator satisfies that $\|\widehat{\theta}_{n,\lambda} - \theta^*\|_2 \leq 0.61$, which is reasonably small.

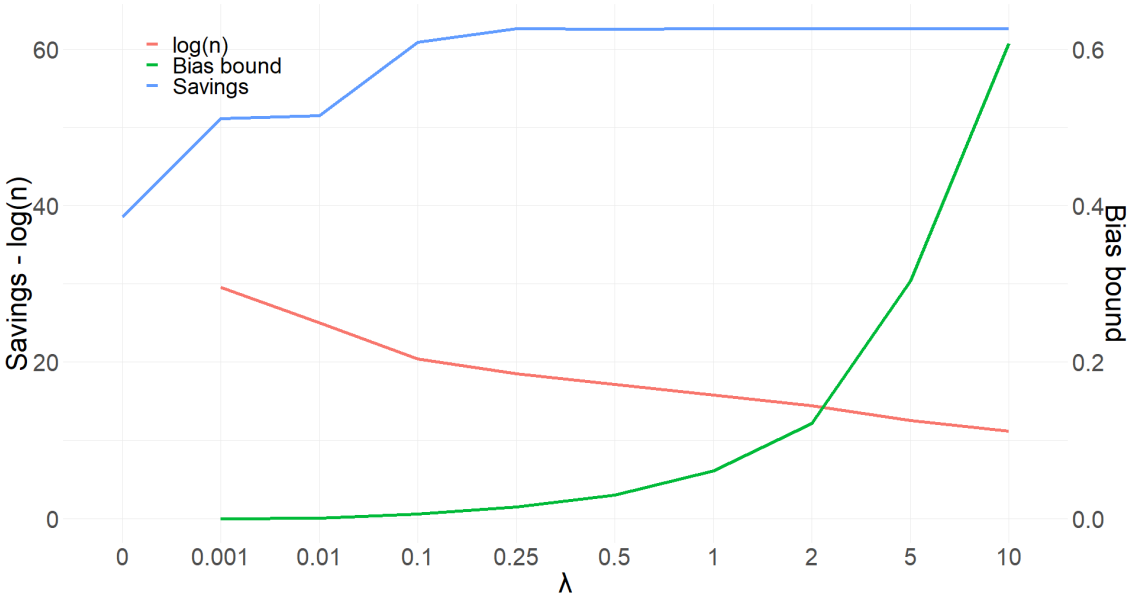

Figure 4.3: Required sample size ($\log(n)$) and L_2-norm bias bound from Theorem 4.3 along with the savings (1.2) obtained for different values of λ for the data simulated in Figure 4.2.

4.5 Experimental results

CS parameter estimators, along with conventional maximum likelihood (ML) estimators, are studied in this section over several scenarios and cost specifications, both simulated and real. A logistic model is considered, where $s(\theta, \mathbf{x}) = (1 + e^{-\theta'\mathbf{x}})^{-1}$, in line with Höppner et al. (2021) and Stripling et al. (2018). This choice is motivated by the fact that the logistic model is widely used in classification tasks, is fast to compute, and is easy to understand. The objective is to evaluate the performance

of a parametric classifier when fitted by maximum likelihood, compared to the introduced CS approach, for which other classifiers are not considered. However, it should be noted that all the developments in this chapter can be applied to any parametric model. In this study, the parameter vector, θ, of the logistic model is estimated by minimizing the regularized AEC (4.5) and by ML, and the performance of both approaches is evaluated using both cost-sensitive and cost-insensitive metrics.

4.5.1 Simulation study

For the simulation study, different dependence structures (non-monotone, heterocedastic, ...) and cost settings are considered in order to provide an in-depth study of the expected classifier performance under different levels of model misspecification and error cost impact. The data are simulated based on six different models, summarized in Table 4.1. For all simulations, the true model is given by $Y = \mu(V) + \epsilon$, where μ is a link function and $V = \theta_0' \mathbf{X}$ is a linear combination of the predictor covariates, $\mathbf{X} = (1, X_1, \ldots, X_6) \in \mathbb{R}^7$. For the Logit model, $X_i \sim N(0,1)$, $\mu(v) = \frac{e^v}{1+e^v}$ and $\epsilon \sim N(0,1)$. This corresponds to the case where the classifier (logistic model) is well-specified. The Squared model is analogous, except that $V = \theta_0' \mathbf{Q}$, with $Q = (1, Q_1, \ldots, Q_6), Q_j = X_j^2$. In this way, the probability of Y is symmetric with respect to the origin in the score V, breaking the monotonic dependence assumption of the classifier. The Interaction model differs from the Logit model defining the index as $V = \theta' X + X_1 X_2$. For the Ker & Sam model, introduced in Ker and Sam (2018), $X_j \sim N(0,1)$, $j = 1,3,5$ and $X_j \sim \chi_3^2$, $j = 2,4,6$ and ϵ is drawn from a normal mixture such that $\epsilon \sim N(3,1)$ or $\epsilon \sim N(-3,1)$ with probability 0.5. Thus, ϵ is not normally distributed as assumed in the logistic model, leading to model misspecification. The Scobit model is exactly like the Logit model, but with the link function $\mu(v) = 1 - \frac{1}{(1+e^v)^6}$. This is a skewed version of the logistic model. For the Klein & Spady model, introduced in Klein and Spady (1993), a heteroscedastic error term, $\epsilon \mid V \sim N(0, \sigma(V))$ is considered, with $\sigma^2(V) = 0.1(1+V^2)^2$. This violates the assumption of monotonic dependence between Y and V and

the assumption of homoscedasticity in logistic regression. In all the simulations, $\theta_0 = (\theta_{00}, \theta_{01}, \cdots, \theta_{06}) = (\theta_{00}, 0.001, 0.002, -0.001, -0.002, 0.003, -0.003)'$, where θ_{00} is chosen such that $\bar{y} \approx 0.01$, a case proportion in line with the typical fraud detection problem that motivates this chapter.

Model	θ_{00}	Index	$E[Y = 1 \mid \mathbf{x}]$
Logit	-4	$V = \theta' X$	$\frac{e^V}{1+e^V}$
Squared	-4.5	$V = \theta' Q$	$\frac{e^V}{1+e^V}$
Interaction	-5	$V = \theta' X + X_1 X_2$	$\frac{e^V}{1+e^V}$
Ker & Sam	-5	$V = \theta' X$	$\mathbb{P}(V + \zeta > 0 \mid V)$
Scobit	-6	$V = \theta' X$	$1 - \frac{1}{(1+e^V)^6}$
Klein & Spady	-4	$V = \theta' Q$	$\mathbb{P}(V + \eta > 0 \mid V)$

Table 4.1: Summary of the data generation models for the simulation study.

The exogenous variable W is simulated in two ways. For the first (W1), $W = 100\chi_1\chi_2$, with $\chi_1 \sim \chi^2_4$ and $\chi_2 \sim N(7, 0.8)$, where χ_1 and χ_2 are independent, in order to replicate the asymmetry commonly encountered in financial problems. For the second (W2), if $y = 0$ or $U < 0.8$, where $U \sim U[0, 1]$, W is simulated as in (W1), and W is drawn from $N(e^{10}, e^8)$ elsewhere, to impose some dependence of the amount on the response variable.

The parameters of the loss function (1.1) vary in order to study different cost scenarios. With the same a/c and b/c ratios, the same results are obtained since it is only the scale of the loss function that is changed. Therefore, the values of $b = 10$ (the analysis cost) and $c = 100$ (the lost percentage produced by an undetected case) in (1.1) are fixed and $a = \{0, 0.01, 0.1, 1\}$ (the benefit per operation). Thus, the impact of different false positive and false negative rates is thoroughly investigated.

Model performance is evaluated in terms of savings (1.2), proportion of observations labeled as cases (PP), precision ($PPV = TP/PP$), and recall $= TP/(TP + FN)$, considering the Bayes minimum risk threshold introduced in equation (2.7) (the theoretical optimal cutoff). By considering the Bayes minimum risk threshold and a cost-insensitive model (trained by maximizing the likelihood), the proposed CS estimator, included in the *predict-and-optimize* philosophy, is compared with a

predict-then-optimize strategy. Therefore, a comparative study of both strategies is also performed. There are other thresholding strategies, such as the empirically constructed rule proposed in Chapter 2, that show good performance in practical applications. However, in order to compare the results obtained in this section with those of previous works such as Höppner et al. (2021) and Bahnsen et al. (2014a), in what follows the Bayes minimum risk decision rule (2.7) is considered.

Six models, two choices for W, and four cost settings are combined to generate 48 simulation scenarios. For each scenario, training and test samples are simulated with $n = 10,000$ and $n = 20,000$, respectively. A logistic regression model is fitted maximizing the likelihood (ML) and minimizing the regularized AEC (4.5) (CS) over the train set, with $\rho_\lambda(\theta) = \lambda\|\theta\|_1$. This provides a comprehensive test of the performance of the proposed approach compared to its cost-insensitive counterpart.

For brevity sake, results are summarized only for the value of λ maximizing savings (1.2), selected by cross-validation (CV) for each scenario over the hyperparameter grid $\lambda \in (0, 0.001, 0.01, 0.1, 1, 2, 5, 10, 20, \lambda_l, \lambda_u)$, where λ_l, λ_u are the lower and upper bounds in (4.10). Note that the first value of λ coincides with the unregularized loss function (4.2). The parameter R, which controls the restricted search in the optimization process (4.7), is set at $R = 50$. The obtained models are applied to the test sets and the metrics introduced above are calculated. This is repeated 100 times for each setting.

The optimization process is implemented using the open-source statistical software R R Core Team (2021). The Broyden–Fletcher–Goldfarb–Shanno algorithm (BFGS), named after Broyden (1970); Fletcher (1970); Goldfarb (1970) and Shanno and Kettler (1970), is used, which is available in the maxlik package (see Henningsen and Toomet (2011)). The BFGS algorithm was chosen because it is easy to implement and allows the inclusion of parameter search constraints, as introduced in equation (4.7). Furthermore, BFGS updates avoid matrix inversion, resulting in a computational complexity of $\mathcal{O}(n^2)$, which is better than the $\mathcal{O}(n^3)$ complexity of Newton's method. The BFGS algorithm starts with the parameters estimated by fitting a logistic regression model using maximum likelihood. At each iteration t, the

current estimate $\widehat{\theta}_t$ is updated using the gradient of $\mathcal{L}_{n,\lambda}$ in equation (4.5) evaluated at $\widehat{\theta}_t$ and an approximation of its Hessian matrix. The algorithm continues until the objective value or parameter estimator converges, or after a user-specified number of iterations, set to $10,000$ in this study.

The average results for each simulation are summarized in Table 4.2. Figures 4.4 and 4.5 display boxplots of the savings obtained with each classifier. The consistent improvement obtained with the CS approach, which fits the model minimizing the regularized AEC (4.5), is confirmed, with a significant benefit in several settings. Results in Table 4.2 and in Figures 4.4 and 4.5 are consistent with the results obtained in previous work, such as Höppner et al. (2021); Bahnsen et al. (2014a) and Vanderschueren et al. (2022b). The differences in CS classification performance across the different models are more pronounced when W has some dependence on Y (W2), as the classifier can take this into account in order to make cost-aware decisions.

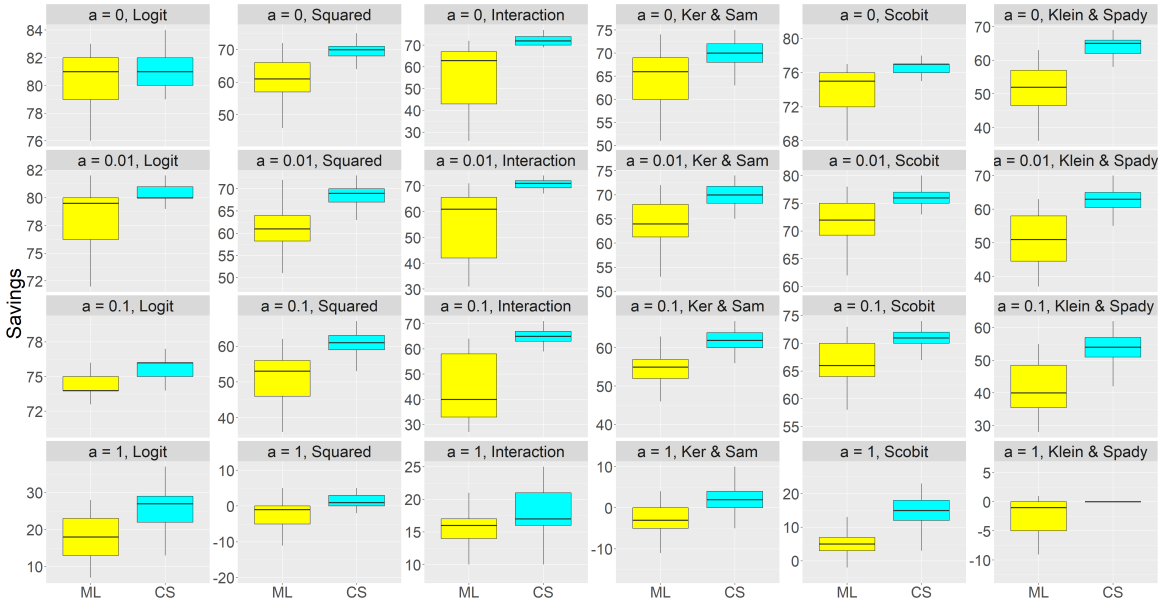

Figure 4.4: Savings (1.2) for the test sets considering the optimal decision threshold (2.7) by cost setting, underlying data model simulation on Table 4.1 and W simulated as (W1). It is considered a logistic model with parameters estimated by maximum likelihood (ML, in yellow) and minimizing the regularized AEC (4.5) (CS, in cyan)

Although the classifier is misspecified except in the logit simulation, the CS

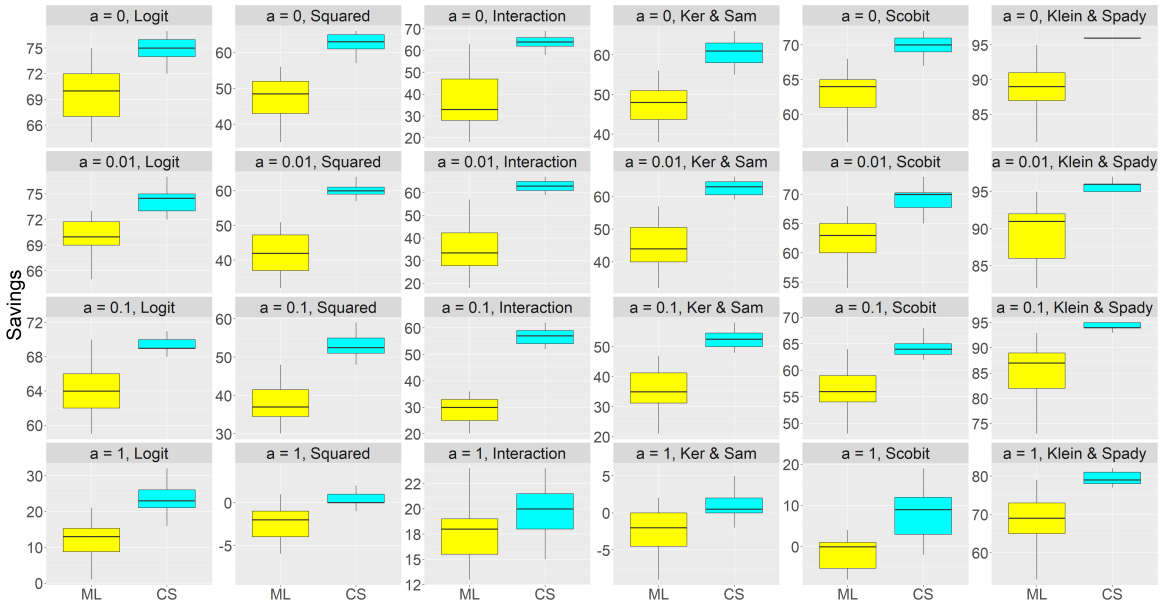

Figure 4.5: Savings (1.2) for the test sets considering the optimal decision threshold (2.7) by cost setting, underlying data model simulation on Table 4.1 and W simulated as (W2). It is considered a logistic model with parameters estimated by maximum likelihood (ML, in yellow) and minimizing the regularized AEC (4.5) (CS, in cyan)

proposal is able to significantly reduce losses, especially when compared to its cost-insensitive counterpart estimated by maximum likelihood. Regardless of the underlying structure of the data and the impact of misclassification errors, it can be seen that fitting the model with a CS approach leads to a significant increase in savings in all scenarios, even in the scenario where the ML classifier is well specified.

It is worth noting the scenario where $a = 0$, where the savings are significantly lower than in the other scenarios. This is due to the large impact of false positives respect to false negatives, which makes the labeling of cases much more sensitive.

The ML estimator focuses on detecting more cases regardless of the exogenous variable. Thus, in some cases it will randomly detect high cost cases and in other cases it will not. This leads to lower savings and higher variance in this metric, as can be seen in Figures 4.4 and 4.5. Since the proposed CS estimator focuses on detecting large impact cases, it consistently achieves higher savings with lower variance, for which it is considered the outperforming approach.

			W1				W2			
Model	a	Estimator	Sav	PP	PPV	Rec	Sav	PP	PPV	Rec
Logit	0	ML	79.64	98.89	1.81	98.92	71.73	96.76	1.79	96.54
	0	CS	80.81	94.05	1.82	94.32	76.43	82.22	1.82	83.44
	0.01	ML	77.91	97.76	1.78	97.99	68.85	96.54	1.78	96.39
	0.01	CS	79.83	94.22	1.78	94.33	73.78	82.33	1.79	82.43
	0.1	ML	74.14	98.83	1.81	98.86	66.66	95.41	1.82	95.76
	0.1	CS	75.32	93.58	1.81	93.50	71.52	82.21	1.82	82.77
	1	ML	17.32	70.41	1.80	71.06	18.08	66.26	1.81	66.50
	1	CS	25.21	79.20	1.79	79.44	28.89	64.08	1.81	64.72
Squared	0	ML	60.20	89.00	1.12	89.18	50.86	83.93	1.13	84.10
	0	CS	69.07	85.79	1.11	85.53	65.22	71.23	1.15	73.20
	0.01	ML	59.84	90.57	1.09	90.62	41.72	81.18	1.07	80.09
	0.01	CS	67.96	85.39	1.10	85.75	59.40	71.21	1.08	71.01
	0.1	ML	51.49	86.43	1.12	86.95	41.15	81.06	1.10	81.23
	0.1	CS	60.53	83.42	1.11	83.37	56.32	71.02	1.11	71.74
	1	ML	-3.31	14.87	2.84	15.00	3.50	17.57	2.91	17.68
	1	CS	0.77	11.40	3.84	13.02	8.41	6.49	4.12	11.77
Interaction	0	ML	53.66	74.00	1.26	75.35	41.16	59.31	1.32	62.35
	0	CS	71.81	88.27	1.22	88.58	66.32	75.35	1.21	75.80
	0.01	ML	54.69	78.68	1.19	79.23	35.86	54.94	1.41	58.88
	0.01	CS	70.14	88.56	1.18	89.35	62.07	74.59	1.19	74.46
	0.1	ML	42.47	65.09	1.29	66.40	35.11	45.91	1.55	52.93
	0.1	CS	64.32	87.50	1.25	88.41	60.10	73.52	1.23	75.32
	1	ML	15.06	9.39	3.53	25.77	21.25	14.16	4.64	30.64
	1	CS	17.06	14.87	2.84	33.17	25.08	12.01	3.84	33.96
Ker & Sam	0	ML	64.02	94.16	1.12	94.41	50.33	85.52	1.11	85.99
	0	CS	69.46	86.53	1.13	87.07	63.31	72.83	1.11	73.48
	0.01	ML	63.27	92.96	1.14	92.56	43.29	78.87	1.15	78.26
	0.01	CS	69.68	87.55	1.15	87.54	61.86	71.78	1.17	72.27
	0.1	ML	53.06	88.02	1.14	87.97	39.45	78.67	1.12	78.30
	0.1	CS	61.22	85.03	1.14	84.43	55.99	69.83	1.14	70.53
	1	ML	-3.42	20.01	5.08	19.58	3.22	19.30	4.65	19.54
	1	CS	1.57	16.35	5.12	17.69	8.85	6.84	4.77	12.28
Scobit	0	ML	73.48	97.47	1.45	97.38	65.72	95.12	1.47	95.31
	0	CS	76.14	91.07	1.46	91.19	71.68	79.24	1.49	80.37
	0.01	ML	70.91	95.34	1.47	95.21	61.46	94.95	1.44	94.73
	0.01	CS	75.53	91.06	1.47	91.00	68.62	78.76	1.46	79.55
	0.1	ML	65.54	93.90	1.50	93.79	58.88	92.10	1.49	92.31
	0.1	CS	70.41	90.42	1.50	90.39	66.75	77.33	1.52	78.71
	1	ML	4.78	43.26	1.49	42.82	5.52	39.37	1.43	38.88
	1	CS	14.40	54.62	1.52	55.11	14.89	43.31	1.56	46.91
Klein & Spady	0	ML	51.16	83.57	0.93	82.91	88.27	92.65	0.95	92.84
	0	CS	63.72	81.61	0.94	81.78	95.30	79.88	1.06	89.95
	0.01	ML	49.89	83.19	0.92	83.40	89.04	92.93	0.94	93.50
	0.01	CS	62.17	81.73	0.92	81.88	95.08	80.42	1.05	90.65
	0.1	ML	41.08	79.48	0.93	79.46	84.44	91.23	0.92	90.97
	0.1	CS	53.83	78.37	0.93	78.20	93.72	78.17	1.06	89.15
	1	ML	36.09	6.22	4.33	6.15	67.46	84.59	0.92	84.39
	1	CS	42.24	0.80	11.72	1.11	80.49	86.81	13.46	93.04

Table 4.2: Simulation results for the different models in Table 4.1, cost setting depending on a in (1.1) and W dependence on Y. A logistic model is considered and its parameters estimated by maximum likelihood (ML) and minimizing the regularized AEC (4.5) (CS).

4.5.2 Real datasets study

Two real fraud datasets are considered to study the performance of the proposed estimator in practice. The first one, introduced in Section 2.5.1 (credit card[1]), consists of 284, 807 credit card transactions made over two days, containing 492 (0.17%) frauds. It includes 28 variables resulting from a PCA along with a "Time" variable (seconds elapsed from the first use) and the "Amount" of each transaction. The "Class" variable indicates whether an operation is legitimate (0) or fraudulent (1). The second one is a data set lend by ASF, consisting of 384, 181 loan requests collected between January 2020 and December 2022 with a fraud percentage of 0.62%. In order to preserve confidentiality, the number of registers of the original data set was truncated and so the fraud proportion. It consists of 17 variables that include socioeconomic information about the customer, and characteristics and historical behavior of the point of sale where the financing is requested. The available variables are limited to the information provided at the time of the request, since there is no history of the customer's behavior or extensive databases to draw on, unlike in the case of credit cards. Only formalized requests are considered, since nothing can be guaranteed about an unformalized operation. These frauds are also the most difficult to detect as they have passed all risk controls. For both datasets, the loss function in (1.1) is considered with $a = 0.1$, $b = 10$ and $c = 100$, as specified by ASF.

The proposed CS model introduced in this chapter and the maximum likelihood model are estimated and tuned as in the previous section. For each data set, a 5-fold cross-validation is performed following Höppner et al. (2021), stratified by the fraud rate and the amount. The average results over the test sets are summarized in tables 4.3 and 4.4.

As in the simulation study, for both datasets the greatest savings are achieved considering the proposed cost-sensitive model. The recall is also higher, indicating that it tends to focus on detecting more frauds, the operations with the greatest

[1] `kaggle.com/mlg-ulb/creditcardfraud`

Estimator	Sav	PP	PPV	Rec
ML	68.22	0.11	56.90	37.20
CS	71.45	0.09	76.56	42.58

Table 4.3: Results for the credit card data set considering a logistic model fitted by maximum likelihood (ML) and minimizing the regularized AEC (4.5) (CS) and the optimal decision threshold (2.7).

Estimator	Sav	PP	PPV	Rec
ML	46.53	20.01	1.79	58.24
CS	49.28	34.87	1.32	72.20

Table 4.4: Results for the ASF data set considering a logistic model fitted by maximum likelihood (ML) and minimizing the regularized AEC (4.5) (CS) and the optimal decision threshold (2.7).

impact on savings (1.2). It is worth noting the PP and PPV resulting from the CS model in the second data set. There are significantly more cases labeled as fraud and consequently a lower accuracy. However, the higher amount of fraud detected offsets the FP cost, resulting in greater savings. This perfectly illustrates the difference between a cost-oriented and an accuracy-oriented approach, and why the former produces better results in CS settings.

4.6 Summary and conclusions

This chapter extends the recent stream of literature known as *predict-and-optimize*, addressing CS parameter estimation in classification models and providing a theoretical and practical underpinning for IDCS learning. For a generic parametric model, different estimation approaches are proposed that allow estimating the optimal parameter vector under any convexity scenario of the objective function. The consistency and asymptotic normality of the estimators are proved. In addition, a bound for the introduced bias is obtained when considering the regularized version of the problem, which is often needed in practice, as shown in Section 4.3.1.

The proposed estimator demonstrates its good practical behavior over a wide

range of simulations and two real data sets, confirming the theoretical results presented earlier. Regardless of the problem difficulty and model misspecification, the CS approach consistently yields larger savings. Thus, it is empirically demonstrated that including costs in the downstream model optimization problem helps to improve posterior decision making. For decision making, the optimal cost-sensitive thresholding strategy included in the *predict-then-optimize* approach is considered. Consequently, it is shown that in CS problems, it is more profitable to consider the *predict-and-optimize* approach with a CS thresholding strategy than a mere *predict-then-optimize* approach.

Theoretical results, an extensive simulation study, and two real data analyses support the application of the proposed method to CS problems. Thus, the gap between theory and practice is effectively bridged by providing both theoretical and practical justification for the implementation of CS models. A parametric model is considered in this chapter, but it is natural and of practical interest to extend this to single index models, where $s_\theta(\mathbf{x}) = g(\theta'\mathbf{x})$ with a nonparametric link function g. This problem will be discussed in the next chapter.

Chapter 5

Cost-sensitive single-index model

5.1 Introduction

Capturing the underlying relationship between a response variable and a set of explanatory covariates is crucial for understanding complex systems, as previously introduced. In classification tasks, the dependence between a random binary variable, $Y \in \{0, 1\}$, on a d-dimensional covariate, $\mathbf{X} = (X_1, \ldots, X_d)$, is modeled as:

$$Y = \mathbb{I}\left(\theta'\mathbf{X} - U \geq 0\right) \tag{5.1}$$

where θ is a d-dimensional parameter vector and U is an unobserved random error term. Denoting s as the conditional distribution function of U, equation (5.1) can be expressed as:

$$P(Y = 1 \mid \mathbf{X}) = \mathbb{E}\left(Y \mid \mathbf{X}\right) = \mathbb{E}\left(\mathbb{I}\left(U \leq \theta'\mathbf{X}\right) \mid \mathbf{X}\right) = s(\mathbf{X}; \theta) \tag{5.2}$$

It is often assumed that the error term, U, follows either a normal or a logistic distribution, becoming (5.2) the probit or logistic model, respectively. In practice, however, there is often no guidance on the functional form of the distribution of the error term. Parametric models, such as those considered in Chapter 4, entail restrictive functional dependence assumptions between the response variable and the covariates. This creates the danger of introducing misspecifications, which generally

leads to inconsistent estimates, as noted in Rothe (2009); Ker and Sam (2018) and Dmochowski et al. (2010). This can also be seen in the results in Section 4.5, with a decrease in classification performance when parametric models are misspecified.

To overcome the challenge of specifying the distribution of U, researchers have sought classifiers that do not require assumptions about the function s. Non-parametric regression provides a solution by allowing complete freedom in modeling the dependence between the predictor variables, $\mathbf{X}$, and the error term, U. However, it faces two major challenges. First, as the number of covariates increases, non-parametric estimation becomes more difficult due to the curse of dimensionality. Parametric models also struggle with large numbers of covariates due to computational limitations and the risk of overfitting. The second challenge is that any interpretation of the effects of the explanatory variables on the response cannot be made directly. This implies another drawback, especially in fields where understanding the underlying dependency relationship is crucial, such as medicine, economics, and social sciences, as mentioned in Hristache et al. (2000).

As a trade-off, single index models (SIMs) relax parametric restrictions by assuming that s is completely unknown, but still impose some parametric restrictions by assuming that the distribution of U depends on $\mathbf{X}$ only through an index $\theta'\mathbf{X}$ (index restriction, as defined in Klein and Spady (1993)). On the one hand, by making no assumption about s, SIMs provide a robust and flexible framework for accurate classification. On the other hand, the parametric assumption reduces the complexity of the relationship between Y and $\mathbf{X}$ to a single index, simplifying the modeling process while preserving the essential information. In this way, overfitting is avoided with a more parsimonious representation of the data and the the computational burden is significantly reduced. Finally, the estimated linear combination, $\theta'\mathbf{X}$, allows interpretation of the model results obtained while providing a flexible framework for handling complex dependencies.

There are two estimation problems in SIMs; the estimation of the parameter θ and the estimation of the link function s. In Peng and Huang (2011) it is stated that if θ can be estimated efficiently, it can be plugged into the SIM and subsequently a

good estimate for the link function can be obtained. Therefore, several techniques have been developed to estimate the parameter vector in SIMs, including Semi-parametric Least Squares (Ichimura (1993)), Semiparametric Maximum Likelihood (Klein and Spady (1993); Rothe (2009); Ker and Sam (2018); Cosslett (1983)), M-estimators (Delecroix et al. (2006)), Average Derivative estimators (Hristache et al. (2000); Horowitz and Härdle (1996); Powell et al. (1989)), and penalized approaches (Peng and Huang (2011)), to mention a few.

Classical classification models estimate the probability of a point belonging to the positive class, usually maximizing accuracy. However, decision-making based solely on minimizing the probability of incorrect classification can lead to poor performance in problems where different types of error have different impacts, as introduced in Chapter 1. In the cost-sensitive (CS) learning domain, classifiers are estimated by minimizing the various misclassification costs. This results in classifiers that make better decisions, as reported in Vanderschueren et al. (2022b); Verbeke et al. (2023) and Höppner et al. (2021). Parametric models have already been extended to the CS approach in Bahnsen et al. (2014a); Höppner et al. (2021); Stripling et al. (2018); C-Rella et al. (2023) and Vanderschueren et al. (2022b) with satisfactory performance. However, they are expected to perform suboptimally when the model is misspecified (see Dmochowski et al. (2010)) or when there are complex relationships between the response, covariates, and costs.

Recently, non-parametric models have been proposed to estimate the PD in Cao et al. (2009); Peláez et al. (2021) and Peláez et al. (2020), with promising results. However, to the best of our knowledge, the CS approach has not been extended to non-parametric or semi-parametric models, for which the proposed model opens the door to a new paradigm in CS classification. Likewise, no previous work has compared in a single work parametric and semi-parametric models both with a CS and cost-insensitive approach, a gap where valuable insights are provided with the results in Section 5.3.

This chapter extends the CS learning approach developing the CS version of the SIM. Considering the fraud detection problem introduced in Chapter 1, the objective

is to minimize losses as specified in the cost matrix in Table 1.1. To do this, the parameter θ and the link function s are estimated in a two-step iterative process minimizing the expected losses. Thus, the model is specifically trained to optimize cost-sensitive decision-making. The CS approach, combined with the flexibility and robustness of SIMs, leads to proficient results. This is empirically demonstrated in Section 5.3 through a simulation study and the analysis of three real credit risk data sets.

The proposed approach can be extended to various domains, including customer churn prediction (Stripling et al. (2018); Verbraken et al. (2012)), credit risk (Bahnsen et al. (2014a); Vanderschueren et al. (2022a)), fraud detection (Höppner et al. (2021)), medical diagnosis (Ker and Sam (2018)), housing decisions (Rothe (2009)) and choice models (Cosslett (1983)) among others.

The plan for this chapter is as follows. Section 5.2 introduces state-of-the-art approaches and develops the estimation of the cost-sensitive SIM in Section 5.2.3, whose performance is analyzed in a thorough simulation study and the analysis of three real data sets in Section 5.3. Conclusions and future extensions are included in Section 5.4.

5.2 Classifiers estimation

Classification models provide a continuous score $s(\mathbf{X}, \theta)$, given the covariate $\mathbf{X}$, reflecting their confidence that an observation belongs to the positive class. Then, in the thresholding stage, the predicted class is estimated as $\hat{Y} = \mathbb{I}(s(\mathbf{X}, \theta) > h)$, where h is the decision threshold. In this section, four classifiers to estimate the score are presented, considering the combination of parametric and semiparametric models with a cost-sensitive and maximum likelihood approach.

In practice, the true joint probability distribution of the variable $\mathbf{Z} = (\mathbf{X}, W, Y)$ is unknown. Therefore, models have to be estimated empirically, relying on the available training data as documented in Vapnik (2000). Formally, given a sample

$\mathcal{Z} = \{\mathbf{z}_i = (\mathbf{x}_i, w_i, y_i)\}_{i=1}^n$ of the random variable $\mathbf{Z}$, the goal is to estimate $s(\mathbf{x}, \theta)$ in equation (5.2) optimizing an objective function, $\mathcal{L}(\theta; s \mid \mathcal{Z})$, over s and the parameter θ. The assumptions about the functional form of s define the parametric or semiparametric nature of the classifier. Likewise, the objective function determines whether the estimator is designed to minimize losses (cost-sensitive) or maximize accuracy (maximum likelihood).

5.2.1 Maximum likelihood parametric classifiers

For a parametric classifier, it is assumed that the function s belongs to a family of parametric functions, e.g. $s(\mathbf{x}; \theta) = (1 + \exp(-\theta'\mathbf{x}))^{-1}$ in the case of logistic regression. Thus, the objective function $\mathcal{L}(\theta \mid s; \mathcal{Z}) = \mathcal{L}(\theta)$ depends only on the parameter θ, which has to be estimated (as introduced in Section 4.2). When the objective is accuracy maximization, the binomial log-likelihood, introduced in equation (2.2), is considered as the objective function:

$$\mathcal{L}_L(\theta) = \mathcal{L}(\theta \mid s; \mathcal{Z}) = \frac{1}{n} \sum_{i=1}^{n} \{y_i \ln[s(\mathbf{x}_i; \theta)] + (1 - y_i) \ln[1 - s(\mathbf{x}_i; \theta)]\} \tag{5.3}$$

and the model is estimated as $\hat{s}(\mathbf{x}) = s(\mathbf{x}, \widehat{\theta}_{ML})$, where:

$$\widehat{\theta}_{ML} = \arg\max_{\theta} \ \mathcal{L}_L(\theta)$$

5.2.2 Parametric cost-sensitive classifiers

As introduced in Section 4.2, the goal of CS learning is to minimize the expected loss, $E[\ell(\hat{Y}, W, Y)]$, where ℓ is the loss function defined in equation (1.1). The expectation $E[\ell(\hat{Y}, W, Y)]$ depends on the parameter θ, the function s and the threshold h through $\hat{Y} = \mathbb{I}(s(\mathbf{X}) > h)$. As introduced in Chapter 4, assuming that $\hat{Y} = \mathbb{I}(s(\mathbf{X}) > \epsilon)$, with $\epsilon \sim U[0, 1]$, then $E[\hat{Y} \mid \mathbf{X} = \mathbf{x}] = P(\hat{Y} = 1 \mid \mathbf{x}) = s(\mathbf{x}; \theta)$. Thus, the *average expected cost* (AEC) of a classifier s over the sample set $\mathcal{Z} = \{\mathbf{z}_i = (\mathbf{x}_i, w_i, y_i)\}_{i=1}^n$ can be defined as introduced in equation (4.1):

$$\mathcal{L}_{CS}(\theta) = \frac{1}{n} E\Big[\sum_{i=1}^{n} \ell(\hat{y}_i, w_i, y_i) \mid \mathcal{Z}\Big] = \frac{1}{n} \sum_{i=1}^{n} \ell(s(\mathbf{x}_i; \theta), w_i, y_i)$$

which only depends on θ when the function s is assumed to belong to a family of parametric functions, e.g. $s(\mathbf{x};\theta) = (1+\exp(-\theta'\mathbf{x}))^{-1}$.

A drawback of the AEC introduced above is that, in practice, it is usually not convex or has no single minimum, as shown in Figure 4.1. As a result, the optimization process may give inconsistent results or fail to converge as demonstrated in Chapter 4. To address this, a regularizer is added to the objective function (4.1), following a common practice in the machine learning literature.

Considering a lasso penalty, as in Section 4.3.2 without loss of generality, the objective function to be optimized is:

$$\mathcal{L}_\lambda(\theta) = \mathcal{L}_{CS}(\theta) + \mathrm{P}_\lambda(\theta) = \frac{1}{n}\sum_{i=1}^{n} \ell(s(\mathbf{x}_i;\theta), w_i, y_i) + \lambda\|\theta\|_1 \tag{5.4}$$

where $\mathrm{P}_\lambda(\theta) = \lambda\|\theta\|_1$ is the regularizer depending on a parameter $\lambda > 0$ that controls the degree of regularization. The lasso penalty does not guarantee an unique minimum, but it creates a valley in the objective function, which helps gradient descent optimization methods to find a solution more efficiently. The regularization parameter λ cannot be estimated directly from the data and has to be chosen by means of hyperparameter optimization strategies such as cross-validation.

As in the previous subsection, assuming some parametric functional form over s, the AEC depends only on θ. Consequently, the goal in CS parametric learning is to minimize the regularized AEC (5.4) with respect to θ given a sample $\mathcal{Z}$. Thus, as introduced in equation (4.7), the classifier is estimated as $\hat{s}(\mathbf{x}) = s(\mathbf{x}, \widehat{\theta}_{CS})$, where:

$$\widehat{\theta}_{CS} = \arg\max_{\theta} \ \mathcal{L}_\lambda(\theta)$$

This CS parametric approach has recently received a lot of attention, and its good performance has been demonstrated empirically in Bahnsen et al. (2014a); Höppner et al. (2021); Stripling et al. (2018) and C-Rella et al. (2023) as well as theoretically in Chapter 4. However, there is still a risk of misspecification in the functional form of the link function, which could lead to inconsistent estimates. By extending the good performance of CS parametric learning to semi-parametric models, it is expected to obtain the best of both approaches.

5.2.3 Cost-sensitive single-index model

The CS single-index model is introduced in this section, and its practical behavior is evaluated in Section 5.3. Assuming that Y depends on $\mathbf{X}$ only through an index $\theta'\mathbf{X}$ and no functional form is imposed over s, the SIM arises as defined in equation (5.2). It is necessary to estimate s and θ in the SIM, which is solved iteratively in two steps as proposed in Hristache et al. (2000); Klein and Spady (1993); Cosslett (1983) and Peng and Huang (2011). For an iteration t, given the estimator of θ in the previous step, $\widehat{\theta}_{t-1}$, the unknown link function s is estimated as an infinite-dimensional nuisance parameter, $\widehat{s}_t$, considering $\widehat{\theta}_{t-1}$ as the true parameter vector. Formally, for an objective function $\mathcal{L}_L(\theta; s)$ as in equation (5.3), it can be expressed as:

$$\widehat{s}_t = \arg\max_s \mathcal{L}_L(\widehat{\theta}_{t-1}; s)$$

In the second step, $\widehat{\theta}_t$ is computed by minimizing a functional with respect to θ, when replacing s in the objective function by the previously computed estimator, $\widehat{s}_t$. In particular, the concentrated objective function, $\widehat{\mathcal{L}}_\lambda(\theta) = \mathcal{L}_\lambda(\theta; \widehat{s}_t)$ is obtained introducing the previously estimated link function into the regularized AEC (5.4). Thus, at step t, the parameter θ is estimated as the minimizer of $\widehat{\mathcal{L}}_\lambda(\theta)$ over θ:

$$\widehat{\theta}_t = \arg\max_\theta \mathcal{L}_\lambda(\theta; \widehat{s}_t)$$

This two-step process is iterated until some stopping criteria is met, for example, the difference between two consecutive estimates is less than a given tolerance. The starting value for θ_0 is the maximum likelihood estimator for the logistic regression model. The conceptually difficult part is the minimization over s, but its computation is simple. On the other hand, minimizing $\widehat{\mathcal{L}}_\lambda(\theta)$ is straightforward because it is analogous to the optimization process in parametric models. However, it is the step that consumes most of the computational effort.

In the following, the iterative estimation process of the cost-sensitive single-index model (CSSIM) introduced in C-Rella et al. (2024a) is detailed. This method generalizes the SIM proposed in Hristache et al. (2000); Klein and Spady (1993);

Cosslett (1983) and Peng and Huang (2011) to the CS setting.

For the first step, $s(\mathbf{X};\theta)$ is expressed, considering Klein and Spady (1993), as:

$$s(\mathbf{X}=\mathbf{x};\theta) \equiv P(Y=1)\frac{f_{\theta'\mathbf{X}|Y=1}(\theta'\mathbf{x})}{f_{\theta'\mathbf{X}}(\theta'\mathbf{x})} \tag{5.5}$$

where $f_{\theta'\mathbf{X}|Y=1}$ is the density of $\theta'\mathbf{X}$ conditioned on $Y=1$ and $f_{\theta'\mathbf{X}}$ the unconditional density of $\theta'\mathbf{X}$. The true underlying densities $f_{\theta'\mathbf{X}|Y=1}$ and $f_{\theta'\mathbf{X}}$ are unknown. Thus, it is necessary to estimate these two densities in order to plug them into equation (5.5) in order to obtain an estimator of the function s. Following Klein and Spady (1993), let $f_{Y=y}(\mathbf{X};\theta) = P(Y=y)f_{\theta'\mathbf{X}|Y=y}(\mathbf{X};\theta)$ and define the estimator of $f_{Y=y}$ as:

$$\hat{f}_{Y=y}(\mathbf{X}_i;\theta;h) \equiv \frac{1}{n-1}\sum_{j\neq i}^{n}\frac{y_j}{h}K\left[\frac{\theta'\mathbf{X}_i-\theta'\mathbf{X}_j}{h}\right] \tag{5.6}$$

where K is a kernel symmetric function with bounded second moments and h is the smoothing parameter or bandwidth that determines the degree of smoothness in the estimated link function. It can be selected a priori with some auxiliary criteria as suggested in Peng and Huang (2011) and Klein and Spady (1993) or introduced in the minimization process in the second step as suggested in Rothe (2009); Hardle et al. (1993) and Delecroix et al. (2006). In this work, the bandwidth is selected by cross-validation. This is done in order to simplify the estimation process and because no significant improvements were observed in preliminary simulations when the bandwidth h was introduced in the optimization process.

Introducing (5.6) in (5.5), following Klein and Spady (1993), the estimator of the function s is defined as:

$$\hat{s}(\mathbf{X};\theta) = \frac{\hat{f}_{y=1}(\mathbf{X};\theta;h)}{\hat{f}_{y=0}(\mathbf{X};\theta;h)+\hat{f}_{y=1}(\mathbf{X};\theta;h)} \tag{5.7}$$

Introducing (5.7) in (5.4), the estimator of the concentrated AEC is expressed as:

$$\widehat{\mathcal{L}}_\lambda(\theta) = \frac{1}{n}\sum_{i=1}^{n}\ell(\hat{s}(\mathbf{x}_i;\theta),w_i,y_i) \tag{5.8}$$

where $\hat{s}$ is as defined in (5.7). Thus, in the second step, given $\hat{s}$, the parameter θ is estimated as the minimizer of the concentrated loss function (5.8):

$$\widehat{\theta}_{CS} = \underset{\theta \mid \theta_1=1}{\arg\min}\,\widehat{\mathcal{L}}_\lambda(\theta) \tag{5.9}$$

where $\theta = (\theta_1, \cdots, \theta_d)$ and the constraint $\theta_1 = 1$ is imposed in order to identify the index coefficients, since there are infinite solutions up to a multiplicative constant as noted in Klein and Spady (1993); Horowitz and Härdle (1996); Rothe (2009); Delecroix et al. (2006) and Horowitz (1992). The iterative process for estimating the CSSIM is summarized in Algorithm CSSIM.

Algorithm CSSIM Cost-sensitive single-index model estimation

1: **Data** $\mathcal{Z} = \{\mathbf{z}_i = (\mathbf{x}_i, w_i, y_i)\}_{i=1}^n$

2: **Input** Regularization parameter $\lambda \geq 0$

3: **Input** Objective function $\mathcal{L}_\lambda(\theta)$ as defined in equation (5.4)

4: **Compute** θ_0 as the ML estimator of the logistic model

5: $t \leftarrow 1$

6: **while** Stopping criteria is not met **do**

7: **Compute** $\widehat{s}_t$ as defined in equation (5.7) with $\theta = \widehat{\theta}_{t-1}$

8: $\widehat{\mathcal{L}}_\lambda(\theta) = \mathcal{L}_\lambda(\theta; \widehat{s}_t)$ as in equation (5.8)

9: **Compute** $\widehat{\theta}_t = \arg\max_{\theta \mid \theta_1 = 1} \widehat{\mathcal{L}}_\lambda(\theta)$ as in equation (5.9)

10: $t \leftarrow t + 1$

11: **end while**

12: **Output** A SIM defined as $s(\mathbf{x}) = \widehat{s}_t(\mathbf{x}; \widehat{\theta}_t)$

5.2.4 Maximum likelihood single-index model

For the classical SIM as introduced in Hristache et al. (2000); Klein and Spady (1993); Cosslett (1983) and Peng and Huang (2011), the process is analogous to that introduced in the previous section, with the log-likelihood (5.3) as the objective function instead of the regularized AEC (5.4). The model fitting is performed as in the iterative process introduced Algorithm CSSIM, where $\mathcal{L}_{CS}$ is replaced by $\mathcal{L}_L$ in the equation (5.3) in lines 3 and 8.

5.3 Experimental results

In this section, the introduced parametric and semiparametric models, as well as boosting algorithms, are studied in their cost-sensitive and cost-insensitive versions over several simulation scenarios and cost specifications. In addition, a study of three real data sets is performed. The objective is to evaluate the performance of these models according to the dependence of the response on the regressors, the distribution of the data, and the different costs of misclassification errors. Since in the simulations it is known the underlying dependence relationship, for the parametric model it is also studied its performance under different degrees of model misspecification. Finally, it is worth highlighting the comparison that is made between parametric, semi-parametric and boosting models. This provides valuable insights that allow for a better understanding of the advantages of each approach depending on the problem.

A logistic model is considered as a parametric benchmark, motivated by its widespread use in classification tasks, its fast computation, and its interpretability as described in Höppner et al. (2021) and Stripling et al. (2018). The logistic model is fitted with the two approaches introduced in Chapter 4 (and recalled in Section 5.2), one cost-insensitive, maximizing the likelihood (5.3) (MLLogit) and another cost-sensitive, minimizing the regularized AEC (5.4) (CSLogit). In addition, two SIMs are fitted, one maximizing the likelihood (5.3) (MLSIM) and, the other, minimizing the regularized AEC (5.4) (CSSIM). Both SIMs are fitted using the two-step iterative process introduced in Algorithm CSSIM. A Gaussian kernel function, K, is considered in equation (5.6) for both SIMs.

In addition, for a more exhaustive performance study, two boosting classifiers introduced in Section 2.2, are considered. These are the Light gradient-boosting machine (LightGBM), introduced in Section 2.2.3, trained with the goal of minimizing cross-entropy (in preliminary simulations this model with this metric gave the best results), and the CS version of the XGBoost algorithm (CSBoost), introduced in Section 2.2.4, fitted minimizing the regularized AEC (5.4). No other

classification models are considered, since the aim of the study is to compare the performance of parametric models, complex classifiers and SIM models in their CS and cost-insensitive versions. This is achieved with the selected classifiers.

The parameters λ and h for the different classifiers are selected by cross-validation (CV) with the goal of maximizing savings (1.2) for each simulation model and cost specification. For the cost sensitive models (CSLogit, CSBoost and CSSIM), the parameter λ in the regularized objective function (5.4) is selected by CV over the grid $\lambda \in \{0, 0.01, 0.1, 1, 10, 100, 1000\}$. Note that the first value of the grid coincides with the unregularized AEC (4.1). For the bandwidth of the SIMs, the MLSIM is fitted by jointly maximizing the likelihood with respect to the parameter, θ, and the bandwidth, h, in the second fitting step (line 8) in Algorithm CSSIM. For the CSSIM, it is necessary to select the parameters λ and h, which increases the computational load. To compensate for this, and to reduce the dimension of the optimization problem in the second step (which causes most of the computational load), the bandwidth is selected beforehand by CV, following the suggestion in Klein and Spady (1993). The CSSIM bandwidth is chosen by CV from a set of bandwidths $h \in \left\{n^{-1/2}, n^{-1/3}, n^{-1/5}, n^{-1/7}, n^{-1/20}, n^{-1/100}, 10, 100, 1000\right\}$. The bandwidths are taken as powers of the sample size, n, in line with Klein and Spady (1993) and as powers of 10 to further explore different degrees of smoothness. When bandwidth estimation is not introduced into the optimization process, the computational burden is significantly reduced without observing a worsening of model performance.

The optimization process is implemented in the open-source statistical software R (R Core Team (2021)), using the Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm (see Broyden (1970); Fletcher (1970); Goldfarb (1970); Shanno and Kettler (1970)) from the maxlik package. This algorithm is easy to implement and avoids matrix inversion, making it computationally faster than Newton's method. This is because at each iteration, the current parameter estimate is updated using the gradient and an approximation of the Hessian of the objective function. The algorithm iterates until convergence or after a specified number of iterations, which in this study is set at $1,000$.

Model performance is evaluated using both cost-sensitive and classification metrics, including savings (1.2), accuracy (Acc), recall, precision (PPV), the proportion of observations labeled as case (PP) and the F-Score (FS). Given an estimated probability for the event of interest, an instance is classified considering the theoretically optimal CS threshold given by the Bayes minimum risk rule in equation (2.7). The probabilities are calibrated using the method proposed in Elkan (2001) in order to ensure that the decision rule is not compromised by uncalibrated probabilities. Using the Bayes minimum risk threshold (2.7) and three cost-insensitive models (MLLogit, MLSIM, and LightGBM), the CS models, that follow the *predict-and-optimize* philosophy, are compared with CS thresholding approaches that only consider costs in the thresholding stage (predict-then-optimize). Likewise, a comparison is made of the improvement obtained by considering a semi-parametric model instead of a parametric one. Finally, the performance of the proposed CSSIM is compared with the main existing approaches in the state of the art, that is, parametric (MLLogit) and semi-parametric (MLSIM) models, complex models (LightGBM) and CS parametric learners (CSLogit, CSBoost).

5.3.1 Simulation study

Simulations are generated from four different models, summarized in Table 5.1. The general model $Y = \mathbb{I}(\mu(V) + U > 0)$ is considered, where μ is a link function, $V = \theta_0'\mathbf{X}$ is a linear combination of four predictor variables, $\mathbf{X} = (1, X_1, \ldots, X_4) \in \mathbb{R}^5$, through the parameter vector θ_0 and U is a random error term. For all simulation models, $\theta_0 = (\theta_{00}, 1, 2, -1, -2)$, with θ_{00} as indicated in Table 5.1. For the Logit model, $\mathbf{X}_i \sim N(0,1)$ and $\mu(V) = \frac{e^V}{1+e^V}$ with $U \sim N(0,1)$. This corresponds to the case where the parametric classifier (logistic model) is well specified. For the Ker & Sam simulation model, introduced in Ker and Sam (2018), $\mu(V) = V$, $\mathbf{X}_j \sim N(0,1), j = 1, 3$ and $\mathbf{X}_j \sim \chi^2_3, j = 2, 4$ and U is drawn from a normal mixture such that $U \sim N(3,1)$ or $U \sim N(-3,1)$ with probability 0.5. For the Qubic simulation model, $V = \theta_0' X/5$ and $\mu(V) = V(V-1)(V+2) - 2$. For the Klein & Spady simulation model (Klein and Spady (1993)), $\mu(V) = V$ and a heteroscedastic error

term, $U \mid V \sim N(0, \sigma(V))$ is considered, with $\sigma^2(V) = 0.1(1+V^2)^2$.

Model	θ_{00}	Index	$\mu(V)$	U
Logit	-6.5	$V = \theta' X$	$\frac{e^V}{1+e^V}$	$N(0,1)$
Ker & Sam	1	$V = \theta' X$	V	$.5N(3,1) + .5N(-3,1)$
Qubic	0	$V = \theta' X/5$	$V(V-1)(V+2) - 2$	$N(0,1)$
Klein & Spady	-7	$V = \theta' X$	V	$N(0, \sqrt{0.1}(1+V^2))$

Table 5.1: Data generation models for the simulation study.

The link function for the four introduced models are represented in Figure 5.1. The index V is scaled to be able to compare the different models. Except for Logit and Ker & Sam models, $P(Y \mid V)$ is not monotone, which contradicts the monotonic dependence assumption of the logistic model (and most parametric classifiers).

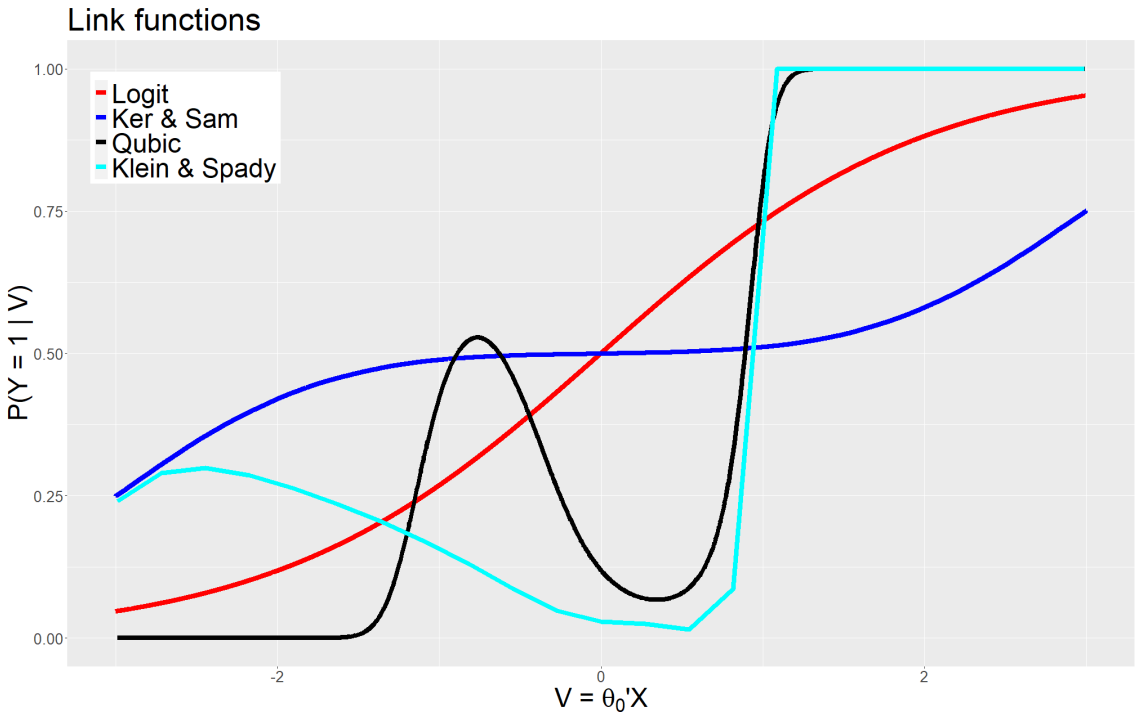

Figure 5.1: Link functions for the different simulation models summarized in Table 5.1.

The exogenous variable is simulated as $W = 100 W_1 W_2$, with $W_1 \sim \chi^2_3$ and $W_2 \sim N(7, 0.8)$, independent. In this way, the asymmetry usually encountered in real problems, such as credit risk, is replicated. The parameters of the loss function (1.1) vary to study different cost scenarios. Note that with the same a/c and b/c

ratios, the same results are obtained since it is only the scale of the loss function that is changed. Therefore, the values $c = 1$ (cost of a false negative) and $b = 10$ (cost of labeling an observation as a case) are fixed in (1.1) and $a = \{0, 0.01, 0.1\}$ (cost of a false positive). In this way, the effects of different false positive and false negative cost ratios are thoroughly investigated.

Four models and three cost settings are combined to generate 12 simulation scenarios, providing a comprehensive test of the performance of the proposed approach. The training and testing data sets of each scenario are simulated with a sample size of $n = 3,000$. This ensures sufficient data to train and evaluate the models, while maintaining a manageable computational load to run the simulations within a reasonable timeframe. Computational times are discussed in Section 5.3.3. The fitted models over the train sets, with the parameters λ and h previously selected by CV, are evaluated over the test sets using the metrics introduced above. This is repeated 50 times for each scenario. The results are summarized in Tables 5.2-5.4 and with a box plot of the savings (1.2) in Figure 5.2.

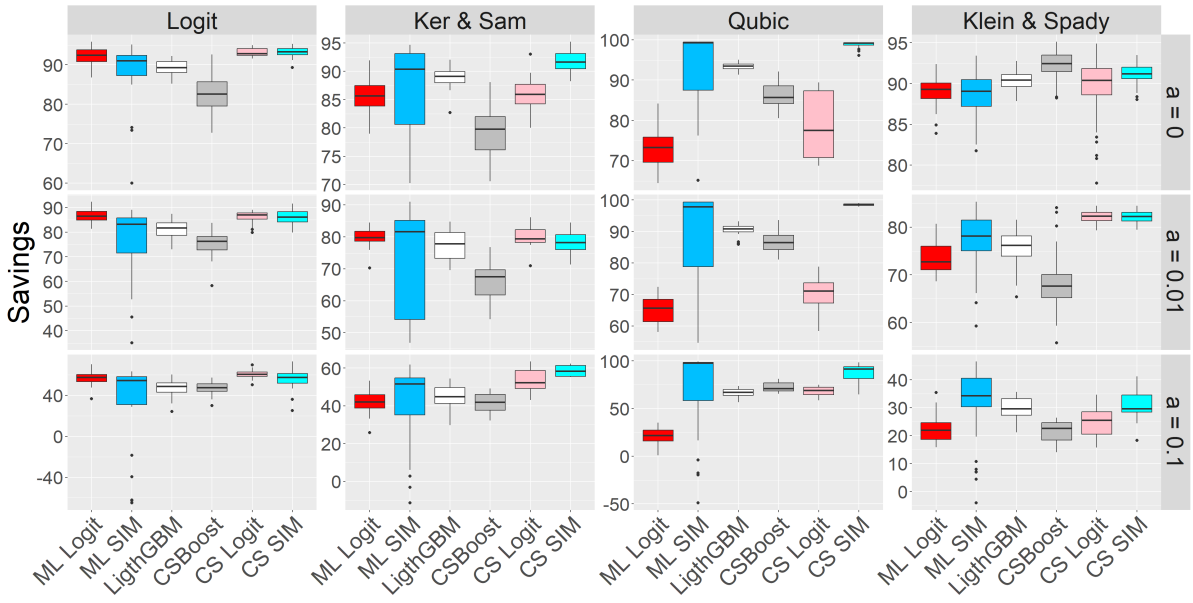

Figure 5.2: Savings (1.2) for the test sets considering the optimal decision threshold (2.7) by cost setting and simulation model on Table 5.1 for the classifiers previously introduced: ML Logit (red), ML SIM (blue), LightGBM (white), CSBoost (grey), CS Logit (pink) and CS SIM (cyan).

Simulation	Model	Sav	Acc	Rec	PPV	PP	FS
Logit	ML Logit	91.80	82.60	91.60	16.70	20.50	35.50
	CSBoost	82.80	81.20	84.90	15.10	21.30	25.10
	LigthGBM	89.30	52.10	90.20	6.60	50.90	31.60
	ML SIM	88.30	78.10	87.10	14.40	24.70	31.50
	CS Logit	93.10	58.60	99.00	8.20	45.10	33.90
	CS SIM	93.10	65.40	94.60	9.30	37.90	33.60
Ker & Sam	ML Logit	85.50	62.50	85.30	10.60	41.10	35.30
	CSBoost	79.30	74.00	80.60	14.50	29.10	24.90
	LigthGBM	88.90	48.00	84.70	7.70	55.50	30.00
	ML SIM	86.90	68.10	83.90	13.20	35.30	28.80
	CS Logit	85.70	68.40	88.90	12.90	35.60	34.50
	CS SIM	91.70	76.00	89.60	16.30	28.00	33.50
Qubic	ML Logit	73.50	44.10	74.50	7.20	58.70	59.70
	CSBoost	86.10	86.50	85.60	28.50	17.50	39.50
	LigthGBM	93.50	52.80	93.10	10.10	52.10	45.50
	ML SIM	93.70	80.20	90.60	65.80	24.30	73.20
	CS Logit	78.30	38.60	77.70	6.90	64.50	41.60
	CS SIM	98.80	97.10	99.10	67.10	8.50	82.10
Klein & Spady	ML Logit	89.00	31.50	76.10	10.60	73.90	24.90
	CSBoost	92.20	11.90	96.60	10.20	97.70	18.40
	LigthGBM	90.50	35.90	78.60	11.60	70.00	26.70
	ML SIM	88.50	45.20	78.00	13.60	60.50	26.40
	CS Logit	89.60	22.30	83.00	10.10	84.50	17.40
	CS SIM	91.20	31.30	78.60	10.90	74.60	25.80

Table 5.2: Mean results obtained in the test sets for the cost setting $a = 0$, for the different simulation models in Table 5.1, with the different classifiers introduced considering the optimal decision threshold (2.7).

In view of the results in Tables 5.2-5.4, the performance of the models varied significantly depending on the characteristics of the data, the complexity of the underlying model, and the error cost specification. As the cost of false positives (FPs) increases, the problem becomes more difficult, and a good trade-off between detection and a low FP rate becomes more important than simple case detection. This complexity is reflected in the decreasing savings as a (FP cost) grows.

In all simulations, savings are greater with the CS version of the logistic model and SIM compared to their cost-insensitive counterparts, as one would expect. However, for the boosting algorithms, LightGBM, adapted to minimize cross-entropy, outperforms CSBoost in some settings. This is because LightGBM, although not trained in a CS way, detects a high percentage of cases, as can be seen from the recall

Simulation	Model	Sav	Acc	Rec	PPV	PP	FS
Logit	ML Logit	86.50	86.60	91.00	20.80	16.50	39.20
	CSBoost	75.20	90.00	77.20	24.50	12.00	32.10
	LigthGBM	81.30	83.10	85.80	16.50	19.60	32.40
	ML SIM	76.10	79.50	81.60	16.50	22.90	31.90
	CS Logit	86.30	78.00	96.20	14.60	25.50	37.80
	CS SIM	86.10	85.40	91.20	19.60	17.70	39.00
Ker & Sam	ML Logit	80.00	73.80	85.60	15.00	29.90	35.50
	CSBoost	66.40	85.90	69.30	22.40	16.10	29.90
	LigthGBM	77.90	78.50	79.70	16.80	24.60	31.40
	ML SIM	71.50	73.80	75.40	14.50	28.80	28.90
	CS Logit	80.00	74.40	87.60	15.50	29.50	35.20
	CS SIM	78.00	73.60	82.10	14.50	29.70	27.60
Qubic	ML Logit	65.20	60.50	75.90	10.40	42.40	65.90
	CSBoost	86.70	91.10	88.10	38.40	13.30	48.60
	LigthGBM	90.50	83.00	93.10	24.30	22.00	46.90
	ML SIM	87.60	79.90	89.70	56.30	24.70	64.90
	CS Logit	70.20	29.00	76.70	6.00	74.10	28.10
	CS SIM	98.50	97.60	99.40	70.80	8.00	87.40
Klein & Spady	ML Logit	73.30	38.80	70.80	10.90	65.40	24.60
	CSBoost	68.20	47.50	75.60	13.50	57.60	23.70
	LigthGBM	75.60	51.90	71.30	13.70	52.40	26.20
	ML SIM	77.10	51.00	74.20	14.20	53.80	26.20
	CS Logit	82.20	28.50	76.30	10.00	76.70	19.10
	CS SIM	82.20	28.60	76.40	10.00	76.70	24.40

Table 5.3: Mean results obtained in the test sets for the cost setting $a = 0.01$, for the different simulation models in Table 5.1, with the different classifiers introduced considering the optimal decision threshold (2.7).

in Tables 5.2-5.4. By detecting most of the cases, the savings are consequently high, especially in scenarios where the FP cost does not have as much impact. In addition, by considering the CS threshold (2.7), LightGBM can achieve a satisfactory CS classification taking costs into account in the thresholding stage. The probabilities are calibrated beforehand, but since CSBoost tends to output probabilities that are too extreme, worse results are obtained in the test sets due to overfitting, as can be seen in Figure 5.2. In conclusion, the combination of a good classifier and a CS threshold can improve the results compared to a CS model, consistent with previous findings in Vanderschueren et al. (2022b).

Regarding the comparison between parametric and non-parametric models, ML-SIM achieves overall a higher accuracy than MLLogit due to its flexibility in mod-

Simulation	Model	Sav	Acc	Rec	PPV	PP	FS
Logit	ML Logit	57.20	88.00	91.30	22.40	15.00	37.40
	CSBoost	46.60	94.60	58.50	35.90	6.00	30.60
	LigthGBM	46.90	86.70	84.90	19.70	15.90	31.00
	ML SIM	32.20	82.80	80.00	18.10	19.40	30.40
	CS Logit	60.70	91.40	84.40	28.10	11.10	36.30
	CS SIM	55.50	87.50	90.70	22.00	15.50	36.90
Ker & Sam	ML Logit	41.80	77.90	85.90	17.60	26.00	36.20
	CSBoost	41.40	90.50	55.60	29.30	10.10	31.40
	LigthGBM	45.10	82.70	77.90	20.40	20.30	31.70
	ML SIM	40.60	79.40	80.60	18.50	23.90	31.00
	CS Logit	53.40	86.50	78.00	25.60	16.40	36.00
	CS SIM	58.60	84.10	85.50	23.50	19.80	35.00
Qubic	ML Logit	21.00	69.60	78.50	13.30	33.60	65.20
	CSBoost	72.70	94.90	80.70	53.60	8.50	54.70
	LigthGBM	66.80	86.70	91.10	28.80	18.00	46.70
	ML SIM	70.70	90.10	89.30	68.10	14.40	76.60
	CS Logit	68.40	93.70	78.70	46.50	9.60	77.10
	CS SIM	88.00	96.30	95.00	63.90	8.80	73.60
Klein & Spady	ML Logit	22.20	56.40	63.40	13.90	46.30	24.80
	CSBoost	21.50	73.20	43.20	17.30	25.40	24.90
	LigthGBM	29.50	63.20	62.90	16.20	39.40	26.40
	ML SIM	31.40	59.50	69.20	16.10	44.40	26.50
	CS Logit	24.80	57.80	65.80	14.90	45.40	25.50
	CS SIM	31.20	60.40	68.40	16.10	43.30	27.20

Table 5.4: Mean results obtained in the test sets for the cost setting $a = 0.1$, for the different simulation models in Table 5.1, with the different classifiers introduced considering the optimal decision threshold (2.7).

eling data dependence. The drawback, as explained in Chapter 1, is that there is no direct relationship between accuracy and losses. Thus, there are some scenarios, such as the Klein & Spady simulation, where the MLSIM is the worst approach in terms of savings despite having the highest accuracy. When the MLLogit is compared with the boosting algorithms, it is clearly outperformed by both boosting algorithms in terms of all the metrics considered, except in the Logit simulation, the scenario where the logistic regression model is correctly specified.

Maximum likelihood models focus on obtaining the best possible classification, regardless of the exogenous variable. Therefore, in some cases these models will randomly detect observations with high costs and in other cases they will not. This leads to lower savings and higher variance in this metric, as can be seen in Figure

5.2. The CSLogit model, by focusing on reducing the cost of misclassification error, tends to have larger savings with less variance, as shown in Section 4.5. Special cases are simulations where the link function is not monotonic (Qubic and Klein & Spady simulation models, as can be seen in Figure 5.1). In these simulations, by imposing a logistic shape on the link function, the model adjusts one of the two monotone regions of the underlying link function. Thus, in some cases the fit will produce good results despite the model misspecification, and in other cases it will not, resulting in lower savings and higher variance.

Combining the flexibility of SIMs with a CS approach for model fitting, the CSSIM proposed in Algorithm CSSIM achieves the best results compared to all other approaches considered in this study, except in the case of the Logit simulation. This can be seen in Tables 5.2-5.4 and Figure 5.2. Because the CSSIM model focuses on detecting large cost cases and has the freedom to adapt to any shape of the link function, it performs better in terms of savings with lower variance. In this way, CSSIM provides a flexible and general model for minimizing losses, while other models struggle with overfitting or do not explicitly consider misclassification costs.

It is demonstrated that fitting a SIM with a cost-sensitive approach improves the model performance in terms of savings compared to its cost-insensitive version and compared to boosting and logistic regression models in their CS and cost-insensitive versions. In almost all the simulations and cost settings the highest savings with lower variance are obtained with the CSSIM, and in the worse performance scenarios, similar results are obtained, as can be seen in Figure 5.2. For all these reasons, it is concluded that the proposed method, CSSIM, presents the best results of the study.

5.3.2 Real datasets study

Three real credit risk data sets are considered in order to study the proposed CSSIM performance in practical scenarios. Some of the data sets contain continuous and categorical variables, but since SIMs are designed to work with continuous variables, categorical variables are ignored in the modeling. All datasets correspond to a real

credit risk problem where some covariates are available to model a response variable (fraud or default). The objective is to minimize losses, which depend on the amount of the transaction.

The first data set (Credit Card data set[1]), introduced in Section 2.5.1, is a fraud data set containing 28 variables resulting from a PCA along with a "Time" variable (seconds elapsed since the first use). The data set is undersampled so that they remain $5,752$ transactions containing 101 (1.76%) frauds. The second one (ASF data set) is lend by ASF. It is composed of $4,300$ loan requests for a new product collected between January 2022 and December 2022 with a default percentage of 10.32%. To preserve confidentiality, the number of registers of the original data set is truncated in order to change the default proportion. It consists on 11 numerical variables which include socioeconomic information of the client, characteristics of the operation and attributes of the point of sale where the financing is requested. The third one, (PAKDD data set[2], introduced in Linhart et al. (2009)), consists of $11,111$ prospective clients to be selected for granting a loan for which 4 numerical variables are available to model their probability of default. There are 3047 (27.42%) defaults in the sample. The amount of the loan is simulated as $W \sim 100\chi_3$ in order to replicate the asymmetry commonly encountered in credit risk problems.

These datasets are selected to examine different proportions of cases, sample sizes, and number of covariates in the sample. The goal is to investigate the impact of class imbalance and number of covariates on model performance. Since none of the state-of-the-art datasets provide guidance on the impact of misclassification error costs, and to preserve the confidentiality of our collaborating institution, several loss function (1.1) specifications are considered. Also, the effect of different ratios of false positive and false negative costs on the model performance is investigated. The values of the loss function are taken in line with those considered in the previous section, i.e. the loss function in equation (1.1) is considered with $a \in \{0, 0.01, 0.1\}$, $b = 10$ and $c = 1$.

[1] `https://kaggle.com/mlg-ulb/creditcardfraud`

[2] `https://github.com/JLZml/Credit-Scoring-Data-Sets/tree/master/2.%20PAKDD%202009%20Data%20Mining%20Competition`

The models used in the real data analysis are fitted and estimated as described in the previous section. For each data set, a 5-fold cross-validation is performed according to Höppner et al. (2021), stratified by case proportion and amount. The mean results over the test sets are summarized in Table 5.5. As in the previous section, when comparing the $a = 0$ and $a = 0.1$ scenarios, most models showed a decrease in savings as the impact of false positives increased.

On the credit card dataset, the proposed CSSIM outperforms all other approaches in terms of savings, even outperforming complex models such as LightGBM and CSBoost. This is due to its high recall (correctly identifying cases) while minimizing the number of operations labeled as cases. This translates to fewer false positives and consequently, higher savings.

As the proportion of cases in the training sample increases, complex models such as LightGBM show improvement. This is likely due to having more cases to rely on for accurate predictions. This can be observed in the ASF dataset, where state-of-the-art models achieve better performance than the CSSIM. Nevertheless, the proposed CSSIM performs better overall when the impact of false positives is high. In these scenarios, correct case prediction becomes more critical, where the CSSIM proves to be proficient. CSBoost achieves the best performance in the $a = 0.1$ setting, but performs poorly in the other settings. This significant variability in performance highlights the need for caution when considering complex models.

In the PAKDD dataset, all models produced similar results, possibly due to the limited number of covariates, which didn't provide enough information for complex models to outperform simpler ones. In addition, the high number of defaults in the dataset reduced the risk of false positives, making it more difficult for cost-sensitive approaches to demonstrate substantial improvement over cost-insensitive models. It's noteworthy that CSBoost again performed significantly worse at $a = 0.1$, possibly due to overfitting (as evidenced by the high accuracy in this scenario).

Setting	Simulation	Model	Sav	Acc	Rec	PPV	PP	FS
$a = 0$	Creditcard	ML Logit	82.41	98.68	52.38	55.00	1.39	53.66
		CSBoost	82.53	99.10	47.62	83.33	0.83	60.61
		LightGBM	66.75	97.77	47.62	32.26	2.16	38.46
		ML SIM	73.06	98.82	47.62	62.50	1.11	54.05
		CS Logit	80.72	98.40	52.38	45.83	1.67	48.89
		CS SIM	83.25	98.82	52.38	61.11	1.25	56.41
	ASF	ML Logit	96.37	12.28	100.00	11.04	98.60	19.88
		CSBoost	92.09	52.56	90.60	17.52	56.28	29.36
		LightGBM	96.32	10.88	100.00	10.88	100.00	19.63
		ML SIM	96.65	19.63	100.00	11.93	91.26	21.31
		CS Logit	96.32	10.88	100.00	10.88	100.00	19.63
		CS SIM	95.82	28.65	98.29	13.07	81.86	23.07
	PAKDD	ML Logit	87.93	30.13	93.69	27.79	94.31	42.86
		CSBoost	85.98	33.87	94.98	29.10	91.29	44.55
		LightGBM	87.95	30.13	93.95	27.82	94.46	42.93
		ML SIM	87.60	29.52	93.82	27.62	95.00	42.68
		CS Logit	87.89	29.48	93.56	27.58	94.89	42.60
		CS SIM	87.83	30.31	93.18	27.77	93.84	42.79
$a = 0.01$	Creditcard	ML Logit	90.30	98.68	56.52	59.09	1.53	57.78
		CSBoost	88.30	99.10	47.83	91.67	0.83	62.86
		LightGBM	85.47	98.75	52.17	63.16	1.32	57.14
		ML SIM	70.08	97.77	34.78	32.00	1.74	33.33
		CS Logit	87.11	98.33	52.17	48.00	1.74	50.00
		CS SIM	95.51	99.30	56.52	100.00	0.90	72.22
	ASF	ML Logit	88.38	14.79	100.00	11.33	96.09	20.35
		CSBoost	88.33	55.26	91.45	18.51	53.77	30.79
		LightGBM	88.13	10.88	100.00	10.88	100.00	19.63
		ML SIM	82.03	48.84	88.03	16.12	59.44	27.25
		CS Logit	88.13	10.88	100.00	10.88	100.00	19.63
		CS SIM	89.05	25.02	97.44	12.43	85.30	22.05
	PAKDD	ML Logit	85.20	29.30	94.16	26.51	94.10	41.37
		CSBoost	82.00	31.43	94.70	27.19	92.26	42.26
		LightGBM	85.20	29.12	94.57	26.51	94.49	41.42
		ML SIM	85.16	28.87	94.43	26.43	94.67	41.30
		CS Logit	85.21	28.98	94.84	26.51	94.78	41.44
		CS SIM	85.15	29.66	94.16	26.61	93.74	41.50
$a = 0.1$	Creditcard	ML Logit	57.75	99.03	54.17	81.25	1.11	65.00
		CSBoost	85.62	99.03	50.00	85.71	0.97	63.16
		LightGBM	42.11	98.89	41.67	83.33	0.83	55.56
		ML SIM	20.22	97.98	29.17	36.84	1.32	32.56
		CS Logit	90.46	99.17	50.00	100.00	0.83	66.67
		CS SIM	91.48	99.24	54.17	100.00	0.90	70.27
	ASF	ML Logit	31.89	66.88	66.67	19.75	36.74	30.47
		CSBoost	64.34	82.98	76.07	36.48	22.70	49.31
		LightGBM	36.96	67.63	72.65	21.20	37.30	32.82
		ML SIM	22.89	59.91	54.70	14.48	41.12	22.90
		CS Logit	36.72	64.28	66.67	18.44	39.35	28.89
		ML SIM	38.26	68.19	70.94	21.23	36.37	32.68
	PAKDD	ML Logit	64.19	32.61	92.42	28.40	91.14	43.44
		CSBoost	58.09	39.42	87.02	29.97	81.32	44.58
		LightGBM	64.16	31.97	93.06	28.28	92.15	43.38
		ML SIM	64.25	32.76	92.03	28.39	90.78	43.39
		CS Logit	64.13	31.57	93.83	28.26	92.98	43.44
		CS SIM	64.13	32.90	92.42	28.49	90.86	43.55

Table 5.5: Results for the real data sets with the models introduced throughout the chapter considering the optimal decision threshold (2.7).

The proposed CSSIM shows consistent and satisfactory results across different settings, regardless of data characteristics, case proportion, or cost specification. Conversely, complex models such as CSBoost, while achieving the best results in some settings, also exhibit significant variability and potential risks of overfitting. Given its stability and overall performance in both the simulated and real datasets, the proposed CSSIM is considered a good alternative to address CS classification problems. Furthermore, it offers the additional advantage of interpretability through the estimated index.

5.3.3 Cost-sensitive single-index model performance study

After establishing the effectiveness of the proposed CSSIM, it is relevant to perform an analysis to investigate the impact of the parameters λ and h on its performance. In order to better understand the effect of the parameters and to justify their selection, a sensitivity analysis is presented, considering the main objective of maximizing the savings (1.2) on the test set. The computational time is also studied, since it could be a drawback in practice. The results obtained for the Klein & Spady simulation and the cost setting $a = 0$ are summarized in Figures 5.3-5.4 for the different λ and h introduced earlier for the CV. Figure 5.3 is constructed by fixing one of the parameters and representing the results when varying the other parameter. For example, the left graph represent a boxplot of the savings obtained with each fixed value of h and varying λ. In Figure 5.4, the savings obtained for each combination of λ and h are represented to examine how their combination affects model performance. To isolate the effect of the parameters from the sample size, the sample size in Figures 5.3 and 5.4 is fixed at $n = 5,000$.

In Figure 5.3, it can be seen that the λ parameter does not seem to significantly affect the savings obtained in the test set. However, the bandwidth parameter has a significant impact on the model performance. It can be seen that small bandwidth values lead to lower savings, probably due to an overfitting on the training sample. Similarly, large bandwidth values lead to excessive smoothness and consequently

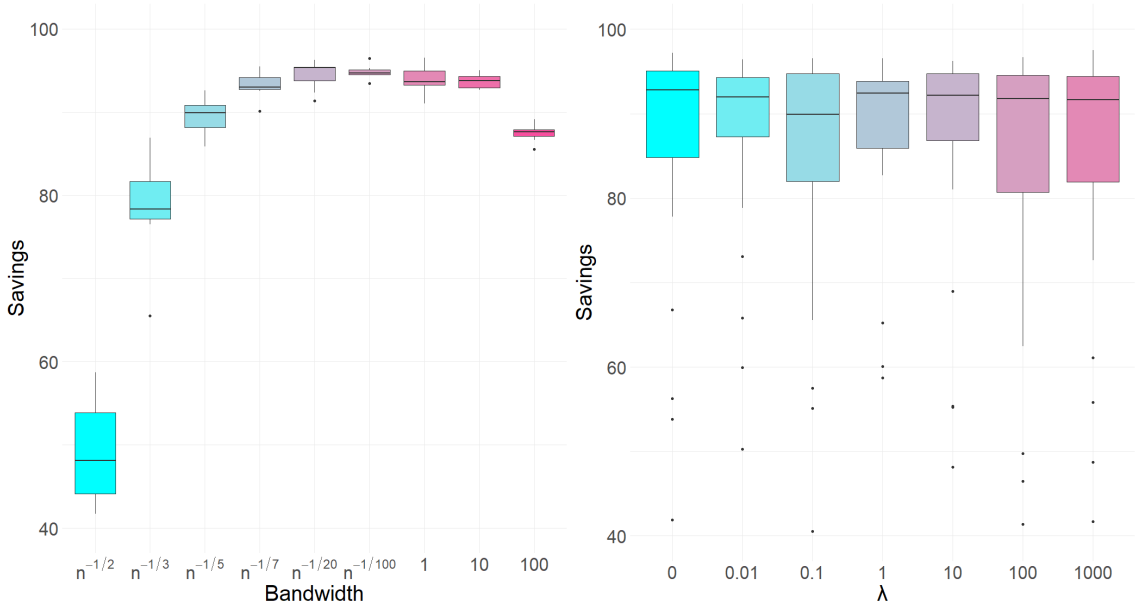

Figure 5.3: Summary for the effect of the parameters h (left) and λ (right) for the Klein & Spady simulation in Table 5.1 and cost setting $a = 0$. The graphs represent a boxplot of the savings for different values of the bandwidth h and the regularization parameter λ.

lower savings. This effect of the bandwidth could justify that the λ parameter does not have as much influence on model performance. Since the bandwidth controls the smoothness of the link function, it gives the model enough flexibility to correct for the lack of convexity in the objective function and prevent overfitting. It is therefore concluded that the choice of the bandwidth h is more important than the parameter λ. Furthermore, considering the results in Figures 5.3 and 5.4, it could be suggested that regularization is not necessary when considering the proposed CSSIM.

As for the computational time depending on the sample size, Figure 5.5 summarizes the time needed to fit the CSSIM as introduced in Algorithm CSSIM for the different values of λ and h considered for CV and for training sample sizes $n \in (100, 200, 500, 1000, 5000, 10000, 50000)$. As expected, the computation time increases as the sample size increases, with little variance across different choices of λ and h.

Finally, the computation time of the models considered in this chapter are compared. For the Klein & Spady simulation and the cost setting $a = 0$, all considered models are fitted as introduced earlier. The computational time required for the en-

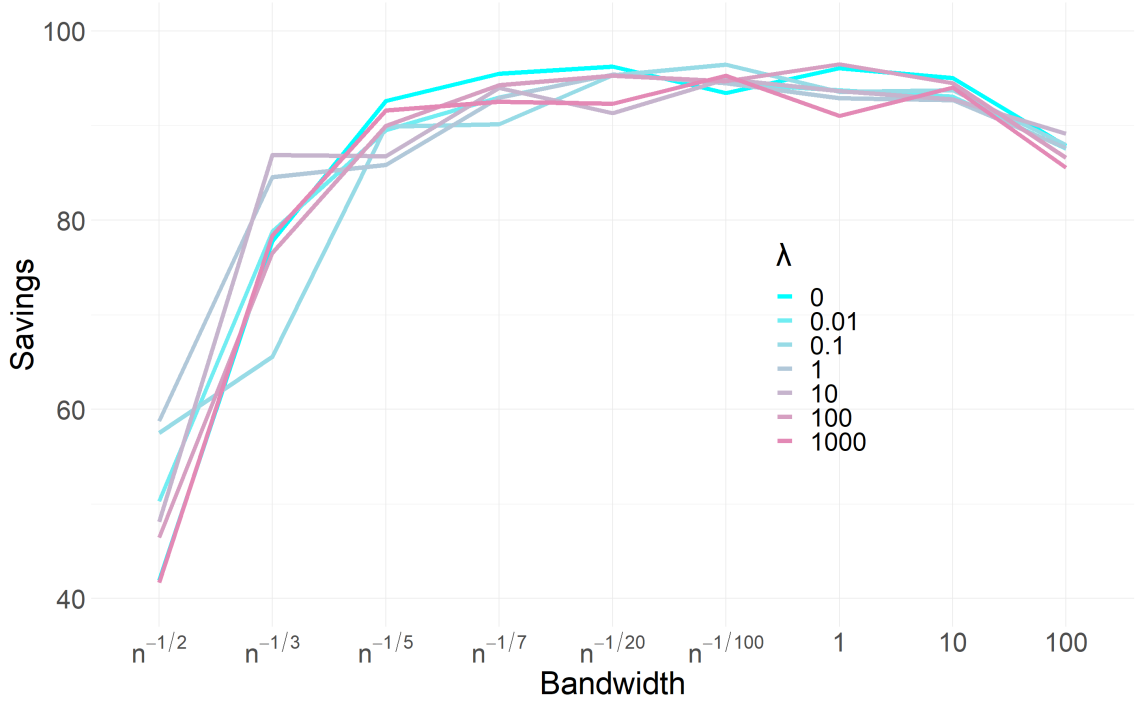

Figure 5.4: Savings for the different values of the hyperparameters λ and h in the CSSIM for the Klein & Spady simulation in Table 5.1 and cost setting $a = 0$.

tire estimation process (parameter tuning plus model fitting) is shown in Figure 5.6 for the different sample sizes. Both parametric and semi-parametric models include the M-estimation step to obtain the estimator of θ, which requires an optimization algorithm with a computational complexity of $\mathcal{O}(n^2)$. However, semi-parametric approaches also require the estimation of the link function (an infinite-dimensional functional). As a result, SIMs are computationally more expensive than parametric models, and the computational burden increases sharply with sample size. Boosting algorithms are, as expected, more computationally expensive than parametric models, but still faster than SIMs.

CSSIM requires a more complex estimation process than MLSIM, since the parameter λ must also be tuned. However, not introducing the parameter h in the second step of the iterative estimation process (step 8 in Algorithm CSSIM), seems to reduce the computational load. This can be deduced from the fact that CSSIM, despite having to tune two parameters, has computational times similar to the MLSIM model. The MLSIM, despite not doing any parameter tuning, introduces h in

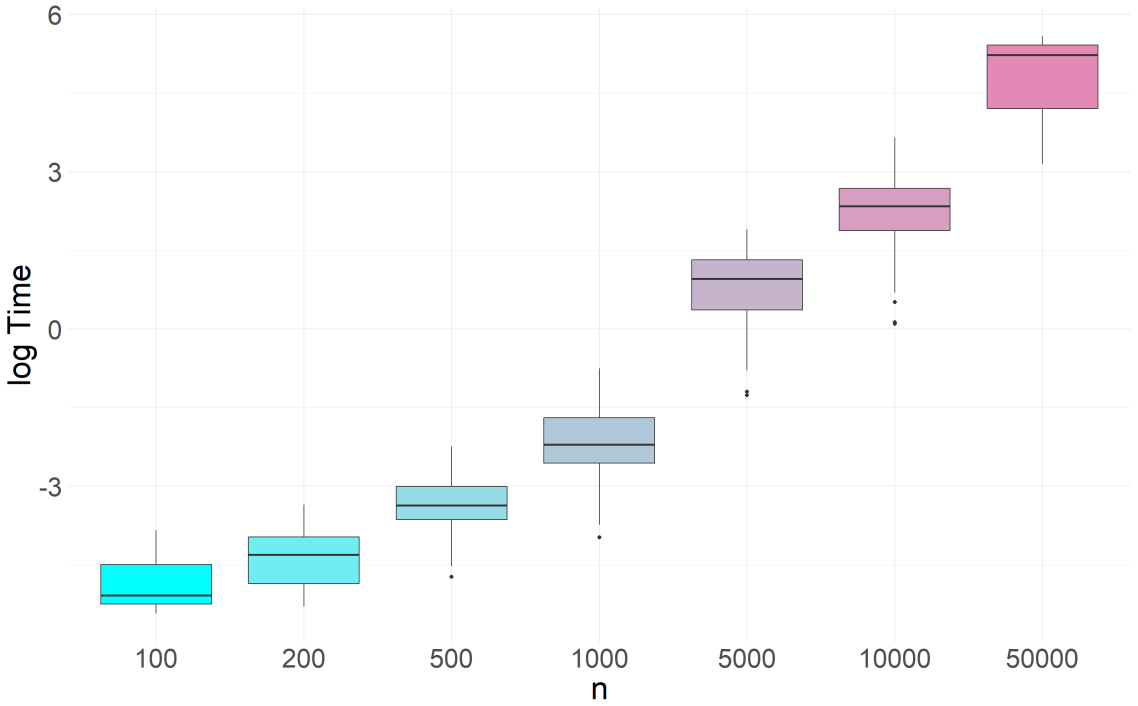

Figure 5.5: Boxplot of the logarithm of the fitting time (minutes) of the CSSIM for different values of the bandwidth h and parameter λ aggregated by the sample size, n, for the Klein & Spady simulation and cost setting $a = 0$.

the M-estimation step, which increases the dimension of the optimization problem. The computational time for the CSSIM is the largest of all the models, with a total estimation time of about 3 days when $n = 50,000$. However, the average time to estimate the model for a fixed λ and h with $n = 50,000$ is about two hours. Moreover, considering the results in Figures 5.3 and 5.4, the dimension of the parameter tuning step could be reduced by considering only the tuning of the bandwidth h, leading to a potential reduction in computation time. Although it is still a high computation time in some practical cases, in several real problems, such as the credit risk problem motivating this work, it is an acceptable time. Especially considering its good performance.

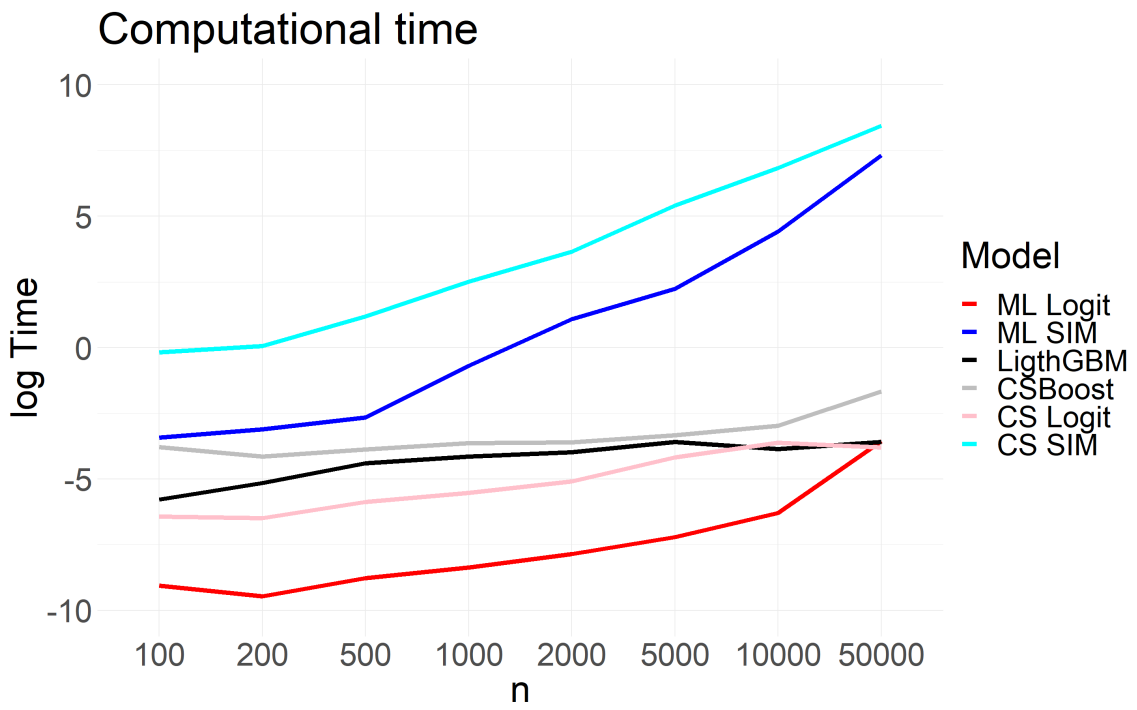

Figure 5.6: Total time (minutes) required for the hyperparameter tuning and fitting of the introduced models by sample size for the Klein & Spady simulation and cost setting $a = 0$.

5.4 Summary and conclusions

The SIM offers a compelling alternative to parametric models and other classification techniques. Its flexibility, high-dimensional data handling capabilities, interpretability, and ability to model complex relationships make it a powerful tool for accurate classification. In this chapter, the SIM is extended to the cost-sensitive classification problem, obtaining a superior model combining the flexibility of SIMs with a cost-sensitive approach. This is demonstrated through a simulation study and on three real data sets, testing different models and cost settings. In addition, valuable insights are obtained about the strengths and limitations of different statistical models under controlled conditions and real-world settings, comparing both parametric, boosting, semi-parametric and cost-sensitive and cost-insensitive approaches.

In general, the savings obtained in the simulation study with the CSSIM are larger and with less variance than the state-of-the-art approaches. With respect to the real data sets, the proposed model outperforms all previous approaches over-

all, demonstrating its good performance in real applications regardless of the data structure and cost specification. This study also highlights the potential pitfalls of relying on complex models in certain scenarios. The fact that the most complex models don't always produce the best results underscores the importance of carefully selecting models that match the nature of the problem and the available data.

A disadvantage of the proposed classifier is the computational time, as is usually the case in semi-parametric and non-parametric modeling. Nevertheless, it has been shown that the parameter tuning step can be alleviated, since the main parameter influencing the model performance is the bandwidth h. Likewise, the proposed model is able to deal with non-convexity in the objective function, as there is no performance improvement when introducing regularization. Thus, it provides a more robust model against suboptimal estimation and model misspecification.

The CSSIM not only achieves superior savings, but also provides valuable insights into the underlying data relationships, as the model provides us with a linear combination of the covariates, something that boosting techniques do not allow. Consequently, the proposed model is considered a promising classifier for CS classification problems in terms of both interpretability and performance, making CSSIM a valuable tool for real-world applications.

Chapter 6

Cost-sensitive reinforcement learning

6.1 Introduction

Lenders and investors face the challenge of selecting potential borrowers to lend to in order to control risk while maximizing profit. Traditional credit scoring systems, such as those presented throughout this work, assign scores based on socio-economic characteristics, with higher scores indicating a higher risk of default (see Höppner et al. (2021); Bahnsen et al. (2014a) and Lessmann et al. (2015)). Despite their effectiveness, these techniques have limitations. Their reliance on historical data from pre-selected customers results in a lack of information about potential borrowers who were previously deemed ineligible for credit. This can lead to customers being unfairly misjudged. Similarly, as models rely on historical data, model performance declines if the underlying relationships between variables change (concept drift). In addition, credit risk leads to an exploration vs. exploitation dilemma as described in Sutton and Barto (2018): should a new client be employed to gain more information and improve predictions at the potential cost of incurring losses, or should the current model's assessment dictate the decision? As a result, financial institutions face two challenges that are often overlooked: adapting to dynamic environments

while optimizing both immediate risk and long-term model performance.

Online learning (OL) addresses the first challenge of dealing with dynamic environments by continuously updating models as new information becomes available. The fundamental difference between OL and classical offline methods lies in their approach to data handling. Static learning algorithms, such as logistic regression, support vector machines or decision trees, use a fixed data set for model training, achieving high accuracy in offline settings. However, they cannot adapt to evolving patterns. OL processes data incrementally, learning continuously as new observations become available. As a result, models remain relevant and effective over time, even as data distributions and patterns evolve as documented in Lattimore and Szepesvári (2020); Sutton and Barto (2018) and Hoi et al. (2018). This continuous learning loop also leads to superior resource efficiency with minimal retraining requirements as noted in Li et al. (2010). The most prominent algorithms in the OL literature are the Online Gradient Descent algorithm (Zhao et al. (2015)), the Passive-Aggressive algorithm (Crammer et al. (2006)), the Perceptron (Rosenblatt (1958)), and many other recently proposed algorithms that follow the principle of large-margin learning as in Kivinen et al. (2004); Dredze et al. (2008); Wang et al. (2012) and Shalev-Shwartz (2012), among others.

The previous classification approaches to credit risk consider the model estimation process separately from the exploitation step. This approach is referred to in the reinforcement learning literature as a greedy algorithm, which is known to be suboptimal because it lacks exploration of the environment, as noted in Sutton and Barto (2018). For its part, OL deals with dynamic settings, but does not consider exploration. Consequently, the credit risk problem is not fully solved.

Within the reinforcement learning paradigm, bandit algorithms are designed to balance exploration and exploitation. Bandit algorithms assume that multiple actions with different rewards are available. Then, an agent explores the different actions to decide on the optimal strategy that maximizes its reward in the long run. Due to the increasing number of problems in this context, several bandit algorithms have been developed in recent years, such as Epsilon-Greedy algorithms

(Lattimore and Szepesvári (2020); Sutton and Barto (2018)), Upper Confidence Bound (UCB) algorithms (Auer et al. (2002); Kaufmann et al. (2014); Lattimore and Szepesvári (2020)), and sampling approaches (Lattimore and Szepesvári (2020); Kaufmann et al. (2012); Korda et al. (2013); Russo et al. (2020)). The drawback of these algorithms is that they do not take into account any information other than the expected reward of each action. In practice, however, auxiliary information is usually available, as noted in Zhou (2016); Langford and Zhang (2007); Pandey et al. (2007) and Wang et al. (2005). For example, in credit risk, do you lend money to people regardless of their socioeconomic characteristics? Contextual bandits extend bandit algorithms to include the additional context of each action to make informed decisions. This approach has gained recognition in applications such as online advertising (Tang et al. (2013); Li et al. (2010)), recommender systems (Lai and Robbins (1985); Li et al. (2010)), personalized medicine (Rabbi et al. (2015)), and credit risk (Shen et al. (2015); Visantavarakul (2022)), where tailoring decisions to specific situations leads to improved performance.

Since the goal of reinforcement learning is for models to learn and adapt to dynamic scenarios, this chapter considers the credit risk problem from the perspective of a binary classification problem. By not having a response with a doubt label (as in the credit risk problem addressed in Chapter 3), the algorithms have to decide for themselves what is the best action to take, without the need for human intervention. Note that the decision about which action is most profitable in the long run becomes particularly important when considering bandit algorithms.

Most classification models, both online and offline, assume a uniform impact of misclassification errors. However, this assumption can lead to suboptimal results in scenarios where error costs vary across error types and instances as introduced in Chapter 1. By explicitly training models to adapt to different misclassification costs, they are aligned with the goal of maximizing benefits, leading to better results as demonstrated in the previous chapters and as shown in previous work such as Höppner et al. (2021); Bahnsen et al. (2014a, 2013); Vanderschueren et al. (2022b) and Wang et al. (2012).

Extending the proposals introduced in Chapter 1 and Chapter 3, an instance-dependent *reward function* is defined from the cost matrix in Table 6.1 below considering the problem of credit risk. It is assumed that approving a good loan application will earn a percent of the amount (W), where a is the interest rate after deducting the operational costs. Regarding the cost of misclassification, the impact of false negatives (FN) is given by cW, where c is the loss given default. Following the reasonableness condition introduced in Elkan (2001), it is assumed that $c > a$, i.e. that the impact of FN is greater than that of false positives (FP). Finally, there is a fixed cost, b, of labeling an observation as a case regardless of its true class, due to the administrative costs incurred by the bank for rejected transactions. Negative values in the cost matrix represent losses, positive values represent profits and all constants are assumed to be positive.

	$\hat{y}_i = 0$	$\hat{y}_i = 1$
$y_i = 0$	$C_i^{TN} = aw_i$	$C_i^{FP} = -aw_i - b$
$y_i = 1$	$C_i^{FN} = -cw_i$	$C_i^{TP} = -b$

Table 6.1: Instance-dependent cost matrix summarizing the benefit/cost of a true negative (TN), false positive (FP), false negative (FN) and a true positive (TP) for the credit risk problem.

From the cost matrix in Table 6.1, the reward function is defined as in equation (3.1). The only difference is that in this chapter the binary response is considered ($\hat{Y} \in \{0, 1\}$). To evaluate model performance over an entire sample, in this chapter it is considered the average expected reward (AER) introduced in equation (3.2).

Traditional batch learning methods are surpassed by reinforcement learning algorithms, which offer adaptability, efficiency, and real-time decision making capabilities. However, there is a lack of approaches for optimizing dynamic CS problems. Recently, cost-sensitive online learning (CSOL) and cost-sensitive bandit techniques have been proposed in the reinforcement learning literature. However, these approaches do not take context into account and/or consider a fixed impact for each

error type, as claimed in Zhang et al. (2018) and Achab et al. (2018). This imposes restrictive assumptions in instance-dependent problems such as credit risk, as reported in Bahnsen et al. (2014a) and Höppner et al. (2021).

To address these drawbacks, in this chapter two novel algorithms to tackle the CS problem from a reinforcement learning perspective are proposed as suggested in C-Rella et al. (2024c). A CSOL algorithm is proposed, which extends the passive-aggressive (PA) algorithm introduced in Crammer et al. (2006). To account for exploration, a CS version of the logistic bandit is also proposed. By combining a CS approach to model fitting with the adaptability of reinforcement learning, it is expected to achieve better results than current classification algorithms.

Exploration is generally expected to improve long-term performance. However, some companies may be more conservative or operate under stricter regulations that limit their ability to explore beyond the predictions given by the current model. Therefore, both approaches are of practical interest. To empirically demonstrate the good performance of the proposed algorithms, an extensive set of experiments on five real-world benchmark datasets and several simulations are performed.

The chapter is structured as follows. Section 6.2 introduces the online learning framework and the cost-sensitive passive-aggressive algorithm proposed in this work. Section 6.3 presents state-of-the-art bandit algorithms and introduces the novel cost-sensitive logistic bandit proposed in this work. Section 6.4 summarizes the results obtained in a performance study considering both simulated and real settings. Conclusions and future extensions are included in Section 6.5.

6.2 Online learning

Online learning (OL) algorithms incrementally learn a predictive model and refine their predictions by sequentially observing data, making a prediction, receiving feedback, and adapting the model with the newly obtained information. This chapter focuses on the Passive-Aggressive (PA) algorithm, a maximum margin classifier as

introduced in Crammer et al. (2006); Wang et al. (2012) and Shalev-Shwartz (2012). This is motivated by the fact that it is generally a more effective approach than the Perceptron algorithm (see Zhao et al. (2015)).

Regarding previous CS online learning approaches, the main proposals in the literature, which will be presented below in this section, are the prediction-based passive-aggressive algorithm (PAPB) proposed in Crammer et al. (2006) and extended in Wang et al. (2012) and Zhang et al. (2018), the cost-sensitive online gradient-descend (CSOGD) algorithm studied in Wang et al. (2012) and Zhao et al. (2015), and the perceptron algorithm with uneven margin (PAUM) introduced in Li et al. (2002). These approaches are designed to deal with class unbalance and consider constant entries in the cost matrix in Table 6.1. This is an oversimplification that can lead to sub-optimal results as shown in Chapter 1. Therefore, an instance-dependent cost-sensitive (IDCS) version of the PA algorithm is proposed in this section to overcome the shortcomings of previous algorithms.

Throughout this section, for convenience, the binary response variable of interest is represented as $Y \in \{-1, 1\}$, where 1 represents cases (default) and -1 non-cases (good operations). We restrict our discussion to classification functions of the form $\hat{Y} = sign(\theta'\mathbf{X})$, where $\mathbf{X} = (X_1, \ldots, X_d)$ is the covariate vector and $\theta \in \mathbb{R}^d$ is the parameter vector to be fitted. At round t, the learner is given an observation $\mathbf{x}_t$ of the covariate vector $\mathbf{X}$ and makes a prediction $\hat{y}_t = sign(\theta'\mathbf{x}_t)$. The vector $\theta \in \mathbb{R}^d$ is estimated considering all the information available up to step t, denoted as $H_t = \{(\mathbf{x}_i, y_i, w_i)\}_{i=1}^{t-1}$. After the prediction $\hat{y}_t$ is made, the correct answer $y_t \in \{-1, 1\}$ is revealed and the learner suffers a loss $\ell(\theta'\mathbf{x}_t, y_t)$, which measures the discrepancy between his prediction and the correct label. The algorithm uses the newly obtained instance-label pair $(\mathbf{x}_t, y_t)$ to update the parameter vector θ for the next rounds, and the process is repeated. The goal of the learner is to minimize the cumulative loss along its path up to the horizon T, $\sum_{t=1}^{T} \ell(\hat{y}_t, w_t, y_t)$.

6.2.1 Passive-aggressive algorithm

The parameter θ in the linear classifier $sign(\theta'\mathbf{X})$ can be interpreted as the normal vector of an hyperplane dividing the data space into two half-spaces, where each side of the hyperplane corresponds to a class $\hat{Y} \in \{-1, 1\}$. The PA algorithm iteratively adjusts the parameter vector θ by maximizing the distance between the prediction for the last instance received, $\theta'\mathbf{x}_t$, and the hyperplane $\theta'\mathbf{X} = 0$. The goal is to obtain the maximum distance while maintaining proximity to the previous classifier, thereby ensuring a balance between adapting to new data and maintaining stability.

The value of $|\theta'\mathbf{x}_t|$ represents the degree of confidence in the prediction as stated in Crammer et al. (2006). Likewise, the loss, $\ell(\theta_t'\mathbf{x}_t, y_t)$, is proportional to the degree to which the prediction was wrong. If a prediction is correct, $y_t\hat{y}_t$ is positive. Thus, the goal is to find $\theta \in \mathbb{R}^d$ such that $y_t\theta'\mathbf{x}_t > 0$. However, the focus remains on obtaining a classifier capable of predicting with high confidence. By imposing a margin on the classification, the following *hinge loss* is obtained:

$$\ell_H(\theta; (\mathbf{x}_t, y_t)) = \begin{cases} 0, & y_t\theta'\mathbf{x}_t \geq 1 \\ 1 - y_t\theta'\mathbf{x}_t, & \text{otherwise.} \end{cases} \tag{6.1}$$

The objective is to correctly classify the last received instance, $\mathbf{x}_t$, while maintaining proximity to the previous classifier. Given the hinge loss in equation (6.1), the parameter of the PA algorithm is updated after each step t as the solution to the constrained optimization problem:

$$\theta_{t+1} = \underset{\theta \in \mathbb{R}^d}{\arg\min} \frac{1}{2}\|\theta - \theta_t\|^2 \text{ s.t. } \ell_H(\theta; (\mathbf{x}_t, y_t)) = 0 \tag{6.2}$$

where $\|\cdot\|$ represents the Euclidean L^2 norm. This notation is used throughout the work for ease of reading.

If the prediction with the current parameter θ_t is correct, the hinge loss (6.1) is zero and the estimate of θ does not vary, i.e. the algorithm remains passive and keeps θ_t unchanged. If the prediction is wrong, $\ell_H(\theta_t; (\mathbf{x}_t, y_t)) > 0$ and the algorithm corrects θ_t in order to classify the instance with margin, but changing the previous parameter θ_t as little as possible. This balances learning from new

data while retaining knowledge from previous observations. The solution to the optimization problem in equation (6.2) is obtained in Crammer et al. (2006) as:

$$\theta_{t+1} = \theta_t + \tau_t y_t \mathbf{x}_t, \quad \text{where} \quad \tau_t = \frac{\ell_H(\theta_t; (\mathbf{x}_t, y_t))}{\|\mathbf{x}_t\|^2} \tag{6.3}$$

When an observation is misclassified, the update strategy in equation (6.3) updates the current estimate as necessary to assign the correct class to the last observation. Although reasonable, this can be too aggressive and is not robust to outliers. Therefore, following Vapnik (1998), it is considered a soft-margin approach that introduces a slack variable ξ into the optimization problem (6.2):

$$\theta_{t+1} = \underset{\theta \in \mathbb{R}^d}{\arg\min} \frac{1}{2}\|\theta - \theta_t\|^2 + C\xi \quad \text{s.t.} \quad \ell_H(\theta; (\mathbf{x}_t, y_t)) \leq \xi \text{ and } \xi \geq 0 \tag{6.4}$$

where C is a positive constant that controls the effect of the slack variable ξ. The updating process has the same closed form as in the equation (6.3), but with a different τ_t term, as demonstrated in Crammer et al. (2006):

$$\theta_{t+1} = \theta_t + \tau_t y_t \mathbf{x}_t, \quad \text{where} \quad \tau_t = \min\left\{C, \frac{\ell_H(\theta_t; (\mathbf{x}_t, y_t))}{\|\mathbf{x}_t\|^2}\right\} \tag{6.5}$$

The iterative process of the soft-margin PA algorithm in equation (6.5) is summarized in Algorithm PA. Note that the regular PA algorithm in equation (6.3) can be obtained by considering a sufficiently large value of C in equation (6.5).

The learner's ultimate goal is to minimize the cumulative loss suffered along its run, $\sum_{t=1}^{T} \ell_H(\theta; (\mathbf{x}_t, y_t))$. For the soft margin PA classifier in Algorithm PA it is demonstrated in Crammer et al. (2006) that if $\|\mathbf{x}_t\| \leq R$ for all t:

$$\sum_{t=1}^{T} \ell_H(\theta_t; (\mathbf{x}_t, y_t)) \leq \max\left\{R^2, 1/C\right\}\left(\|\mathbf{u}\|^2 + 2C\sum_{t=1}^{T} \ell_H(\mathbf{u}; (\mathbf{x}_t, y_t))\right)$$

where $\mathbf{u} \in \mathbb{R}^d$ and $C > 0$ is the parameter that controls the aggressiveness of the algorithm as introduced in equation (6.4).

Algorithm PA consists of a linear predictor, which may seem restrictive. However, it can be further extended to non-linear predictors using Mercer kernels, as shown in Vapnik (1998). Recall that the PA classifier can be represented as a sum

Algorithm PA (Soft-margin) passive-aggressive algorithm

1: **Input** Parameter $C > 0$

2: **Initialize** $\theta_1 = (0, \cdots, 0)$

3: **for** $t = 1, \cdots, T$ **do**

4: **Receive** covariate, $\mathbf{x}_t$

5: **Compute** prediction, $\hat{y}_t = sign(\theta_t' \mathbf{x}_t)$

6: **Receive** feedback, y_t

7: **Compute** $\ell_H(\theta_t; (\mathbf{x}_t, y_t))$ as defined in equation (6.1)

8: **Compute** $\tau_t = \min\left\{C, \frac{\ell_H(\theta_t;(\mathbf{x}_t,y_t))}{\|\mathbf{x}_t\|^2}\right\}$

9: **Update** $\theta_{t+1} = \theta_t + \tau_t y_t \mathbf{x}_t$

10: **end for**

11: **Output** Parameter vector of the PA classifier, θ_T

of inner products:

$$\theta_t' \mathbf{x}_t = \sum_{i=1}^{t-1} \tau_i y_i (\mathbf{x}_i' \mathbf{x}_t),$$

and consider a Mercer kernel, K, such that $K(\mathbf{X}, \mathbf{X}') = \Phi(\mathbf{X})'\Phi(\mathbf{X}')$, where Φ is a "feature map" mapping $\mathbf{X}$ into another space (possibly with higher dimensions). An explicit representation for Φ is not necessary as long as it defines an inner product. Replacing the inner products in Algorithm PA with $K(\mathbf{X}, \mathbf{X}')$ gives a nonlinear version of the PA algorithm, represented as:

$$\hat{y}_t = sign(\theta_{t-1}' \Phi(\mathbf{x}_t)), \text{ where } \theta_{t-1}' \Phi(\mathbf{x}_t) = \sum_{i=1}^{t-1} \tau_i y_i K(\mathbf{x}_i, \mathbf{x}_t) \tag{6.6}$$

where τ_i is defined as in equation (6.5).

6.2.2 State-of-the-art cost-sensitive online learning algorithms

The PA algorithm solves the problem of updating the estimate of θ as new information becomes available. However, it overlooks the effect of different misclassification error costs, which can lead to suboptimal results in cost-sensitive settings as shown

in Chapter 1. As a result, CS algorithms appear in the OL literature. In this subsection, we review all CS proposals in the OL literature to the best of our knowledge.

Passive-aggressive prediction-based algorithm

The passive-aggressive prediction-based algorithm (PAPB), proposed in Crammer et al. (2006), extends the Algorithm PA to the CS setting. This algorithm takes into account the different misclassification costs considering a classification margin defined by an asymmetric loss function. The PAPB algorithm considers the following optimization problem:

$$\theta_{t+1} = \underset{\theta \in \mathbb{R}^d}{\arg\min} \frac{1}{2}\|\theta - \theta_t\|^2 \text{ s.t. } y_t \theta' \mathbf{x}_t - \hat{y}_t \theta' \mathbf{x}_t \geq \sqrt{\ell_{PB}(\hat{y}_t, y_t)} \tag{6.7}$$

where ℓ_{PB} is a loss function with different values depending on the error type (FN or FP). Like the regular PA algorithm, the PAPB algorithm is passive-aggressive, since the current estimate of θ is only updated when an error is made, i.e. when $\hat{y}_t \neq y_t$. Considering a soft-margin approach, as in equation (6.4), in the optimization problem (6.7) leads to the update strategy (see Crammer et al. (2006)):

$$\theta_{t+1} = \theta_t + \tau_t(y_t - \hat{y}_t)\mathbf{x}_t, \quad \tau_t = \min\left\{C, \frac{(\hat{y}_t - y_t)\theta_t' \mathbf{x}_t + \sqrt{\ell_{PB}(\hat{y}_t, y_t)}}{\|(\hat{y}_t - y_t)\mathbf{x}_t\|^2}\right\} \tag{6.8}$$

Cost-sensitive online gradient-descend algorithm

The cost-sensitive online gradient descend (CSOGD) algorithm, studied in Wang et al. (2012) and Zhao et al. (2015), extends the online gradient descent approach introduced in Hazan et al. (2007) by considering a loss function in the parameter updating process:

$$\theta_{t+1} = \begin{cases} \theta_t + \lambda \rho_t y_t \mathbf{x}_t, & \ell_{CH}(\theta_t; (\mathbf{x}_t, y_t)) > 0 \\ \theta_t, & \text{otherwise.} \end{cases} \tag{6.9}$$

where $\rho_t = \rho I(y_t = 1) + I(y_t = -1)$, $\rho = \frac{C^{FN}}{C^{FP}}$ is the ratio between the impact of a FN error and a FP error as defined in Table 6.1, $\lambda > 0$ is a learning rate parameter,

and $\ell_{CH}(\theta_t; (\mathbf{x}_t, y_t))$ is the cost-sensitive version of the hinge loss function (6.1):

$$\ell_{CH}\left(\theta_t; (\mathbf{x}_t, y_t)\right) = \left(\rho I(y_t = 1) + I(y_y = -1)\right) \max\left\{0, 1 - y_t \theta_t' \mathbf{x}_t\right\}$$

Perceptron algorithm with uneven margin

The perceptron algorithm with uneven margin (PAUM) proposed in Li et al. (2002) consists in the CS version of the perceptron algorithm introduced in Rosenblatt (1958), a maximum margin classifier. The predicted class for an observation $\mathbf{x}_t$ is assigned as:

$$\hat{y}_t = sign(\theta' \mathbf{x}_t + b)$$

where $\theta \in \mathbb{R}^d$ and $b \in \mathbb{R}$. Considering different margins for the positive and negative classes, τ_{+1} and τ_{-1} respectively, the CS version of the perceptron algorithm is proposed in Li et al. (2002). For a given learning rate η, starting with all variables equal to 0, the updating process of the PAUM algorithm is given by:

$$\theta_{t+1} = \begin{cases} \theta_t + \eta y_t \mathbf{x}_t, & y_t(\theta_t' \mathbf{x}_t + b_t) \leq \tau_{y_t} \\ \theta_t, & \text{otherwise.} \end{cases} \tag{6.10}$$

$$b_{t+1} = \begin{cases} b_t + \eta y_t R^2, & y_t(\theta_t' \mathbf{x}_t + b_t) \leq \tau_{y_t} \\ b_t, & \text{otherwise.} \end{cases}$$

where $R = \max_t \|\mathbf{x}_t\|$.

6.2.3 Instance-dependent cost-sensitive passive-aggressive algorithm

So far, the OL alternatives in the state of the art have been introduced. The PA algorithm has shown good performance in some settings, such as online advertising, where the agent is only concerned with whether the customer clicks or not, as shown in Li et al. (2010). Regarding the CS OL approaches available, they are rather designed to solve the class imbalance problem (few positive cases in the population), considering constant error impacts depending on the type of error (FP

or FN). However, in settings where the cost of misclassification depends on the characteristics of the instance, these assumptions are too restrictive and naive, as highlighted in Bahnsen et al. (2014a); Elkan (2001); Höppner et al. (2021) and Vanderschueren et al. (2022b).

This section proposes a novel instance-dependent cost-sensitive passive-aggressive (IDCSPA) algorithm that extends the previously introduced PAPB algorithm to the IDCS setting. This approach retains the advantages of OL by updating predictions with the latest available information, but further incorporates the instance-dependent impact of misclassification errors. This is expected to result in a proficient model with better results as shown in C-Rella et al. (2024c).

Since the PA algorithm, by definition, focuses only on losses, it is considered the loss function (1.1) considering only the misclassification costs in the cost matrix in Table 6.1:

$$\ell_{CS}(\hat{y}_t, w_t, y_t) = I(y_t = -1)I(\hat{y}_t = 1)C_t^{FP} + I(y_t = 1)I(\hat{y}_t = -1)C_t^{FN} \tag{6.11}$$

where C_i^{FP} and C_i^{FN} are the FP and FN costs, respectively, as defined in Table 6.1. The loss function ℓ_{CS} in equation (6.11) is introduced into the PAPB optimization problem (6.7) to obtain the proposed IDCSPA algorithm. Thus, at each step t, the parameter θ_t is fitted by solving the optimization problem:

$$\theta_{t+1} = \underset{\theta \in \mathbb{R}^d}{\arg\min} \frac{1}{2}\|\theta - \theta_t\|^2 \text{ s.t. } y_t\theta'\mathbf{x}_t - \hat{y}_t\theta'\mathbf{x}_t \geq \sqrt{\ell_{CS}(\hat{y}_t, w_t, y_t)} \tag{6.12}$$

where $\hat{y}_t = sign(\theta_t'\mathbf{x}_t)$ is defined as in Algorithm PA. The optimization problem in equation (6.12) maintains the idea of varying the current estimate as little as possible while satisfying a certain margin, which in this case depends on the loss function ℓ_{CS} (6.11). The optimization problem in equation (6.12) has a close form solution generalizing the solution developed in Crammer et al. (2006):

$$\theta_{t+1} = \theta_t + \tau_t(y_t - \hat{y}_t)\mathbf{x}_t, \text{ where } \tau_t = \frac{(\hat{y}_t - y_t)\theta_t'\mathbf{x}_t + \sqrt{\ell_{CS}(\hat{y}_t, w_t, y_t)}}{\|(\hat{y}_t - y_t)\mathbf{x}_t\|^2} \tag{6.13}$$

As for the PA algorithm in equation (6.3), the update strategy in (6.13) may be too aggressive, especially since we include the loss function in the estimation.

To consider a soft-margin approach, extending the approach in equation (6.5), the IDCSPA updating rule is defined by:

$$\theta_{t+1} = \theta_t + \tau_t(y_t - \hat{y}_t)\mathbf{x}_t, \;\; \tau_t = \min\left\{C, \frac{(\hat{y}_t - y_t)\theta_t'\mathbf{x}_t + \sqrt{\ell_{CS}(\hat{y}_t, w_t, y_t)}}{\|(\hat{y}_t - y_t)\mathbf{x}_t\|^2}\right\} \tag{6.14}$$

where $C > 0$ is the parameter that controls the aggressiveness of the algorithm. The resulting classifier is shown in Algorithm IDCSPA. Lastly, as for the PA algorithm, although it is considered a linear classifier in Algorithm IDCSPA, it can be extended to non-linear predictors by replacing the inner products with a Mercer kernel, similar to what was presented in equation (6.6).

Algorithm IDCSPA Instance-dependent cost-sensitive (soft-margin) passive-aggressive algorithm

1: **Input** Parameter $C > 0$

2: **Initialize** $\theta_1 = (0, \cdots, 0)$

3: **for** $t = 1, \cdots, T$ **do**

4: **Receive** covariate, $\mathbf{x}_t$

5: **Compute** prediction, $\hat{y}_t = sign(\theta_t'\mathbf{x}_t)$

6: **Receive** feedback, y_t

7: **Compute** $\tau_t = \min\left\{C, \frac{(\hat{y}_t - y_t)\theta_t'\mathbf{x}_t + \sqrt{\ell_{CS}(\hat{y}_t, w_t, y_t)}}{\|(\hat{y}_t - y_t)\mathbf{x}_t\|^2}\right\}$, where ℓ_{CS} is as in (6.11)

8: **Update** $\theta_{t+1} = \theta_t + \tau_t(y_t - \hat{y}_t)\mathbf{x}_t$

9: **end for**

10: **Output** Vector classifier θ_T

6.3 Bandit algorithms

The OL framework introduced in the previous section aims to optimize exploitation with all available information, but does not consider exploration. However, many real-world situations require choosing between multiple options and learning the best one through successive trial and error. This is true for problems such as credit

risk (Achab et al. (2018); Visantavarakul (2022); Russo et al. (2020)) and online advertising (Li et al. (2010)).

Bandit algorithms are a class of reinforcement learning algorithms designed to balance the process of seeking new information (exploration) with maximizing rewards based on existing knowledge (exploitation) in sequential decision problems. Since exploration and exploitation cannot be done simultaneously, Sutton and Barto (2018) concludes that there is a trade-off between these two goals. Thus, at each step, the agent must decide whether to exploit the known option with the highest expected payoff or to explore a less known option, potentially discovering better strategies for future gains.

Bandit algorithms receive their name from a very explanatory example. Imagine a gambler (agent) staring at a row of one-armed bandits (actions), each promising unknown payouts. The agent faces the dilemma of pulling the familiar lever that's currently spitting out coins (exploitation) or trying a new one for potentially bigger wins (exploration). In the stochastic setting, each option (arm) is associated with a distribution that generates random rewards as mentioned in Achab et al. (2018). For example, in Bernoulli bandit problems (see Russo et al. (2020)), the reward of each arm is Bernoulli distributed with a different parameter on each arm. Figure 6.1 shows an example of the problem where there are multiple bandit machines with different (unknown) reward distributions depending on the machine.

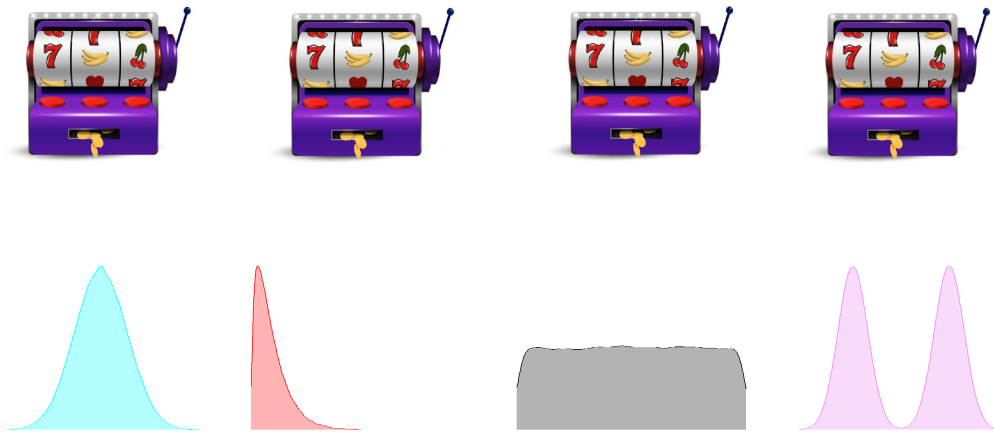

Figure 6.1: Bandit machines with different reward distributions (bottom graphs).

The objective of the agent is to learn the reward distribution of each arm and find a strategy (or policy) in order to maximize the cumulative reward over time. A naive approach would be to try all arms once and then choose the one with the highest reward. However. this can be misleading, as the initial result could be pure luck. On the other hand, endless exploration will lead to missing the benefits of the best option. Bandit algorithms address this exploitation-exploration trade-off by strategically choosing between exploring less known options and exploiting the seemingly best one. Considering that, instead of multiple slot machines, there is a single machine with multiple arms to pull, the problem is commonly known as the multi-arm bandit (MAB) problem.

In this section, there are presented state-of-the-art MAB algorithms and their drawbacks are commented, motivating the novel CS logistic bandit algorithm proposed in this chapter. There are three main bandit algorithms in the literature. The Epsilon-Greedy algorithm (Sutton and Barto (2018)), a greedy algorithm allowing for exploration, is an intuitive way of forcing exploration. At each step, it simply exploits the most lucrative arm (given the information obtained so far) with probability $1 - \epsilon$, and explores a random new arm with probability ϵ. However, this algorithm does not explore the environment efficiently, since it selects new actions with equal probability, regardless of whether it is actually a good action to explore. Two approaches are proposed to overcome this drawback. The first is the Upper Confidence Bound (UCB) algorithm (Auer et al. (2002); Kaufmann et al. (2014); Lattimore and Szepesvári (2020)), which prioritizes regions with high potential rewards while accounting for uncertainty through confidence bounds. Based on optimism in the face of uncertainty, the action with the highest upper confidence bound for the expected reward is selected. In this way, UCB not only exploits the best arm, but also explores those with promising rewards, minimizing the risk of overlooking hidden riches. The other approach is Thompson Sampling (TS), originally proposed in Thompson (1933) and further studied in Kaufmann et al. (2012); Korda et al. (2013) and Russo et al. (2020). It embraces the probabilistic nature of the problem by treating models as probabilistic terms rather than deterministic

truths. TS samples model estimators from their posterior distribution, updated with the observations collected from all the arms up to time t. In the early stages, there is greater uncertainty about the distribution of each arm, which leads to greater variance when resampling the estimators, and hence to exploration. As more information becomes available, it is used to update the estimated posterior distribution. This results in a more concentrated probability mass around the estimator, which favors exploitation. This uncertainty-aware approach allows TS to adapt to dynamic and changing scenarios while exploring the arms for which there is less information.

6.3.1 Contextual bandits

The previously introduced MABs blindly explore actions with unknown rewards. Nevertheless, real-world scenarios often provide additional information about each option. For example, the socioeconomic characteristics of the customer in loan application, or the search history in advertisement recommendation. This additional knowledge is called *context*, and it can change the optimal choice at each step, since the arm with the highest expected reward may differ in different contexts as noted in Zhou (2016) and Langford and Zhang (2007).

Contextual multi-armed bandits (CMAB), as formulated in Langford and Zhang (2007); Pandey et al. (2007) and Wang et al. (2005), are a class of bandit algorithms that take the context into account. Thus, predictions are tailored to each unique situation, improving performance. Furthermore, the context allows the generalization of the information across actions as highlighted in Russo et al. (2020). As a result, CMAB algorithms have found applications in many different areas, such as recommendation engines (Lai and Robbins (1985)), advertising (Tang et al. (2013); Li et al. (2010)), personalized healthcare (Rabbi et al. (2015)), and portfolio selection (Shen et al. (2015)), among others.

A CMAB, B, is defined by a set of available arms $\mathcal{A} \subseteq \{1, \cdots, K\}$. The set of arms $\mathcal{A}$ is assumed fixed at each step $t = 1, \cdots, T$. Each arm $k \in \mathcal{A}$ is described by a reward function, r_k, mapping d-dimensional context vectors, $\mathbf{x}_t$, to some reward

$r_{t,k} = r_k(\mathbf{x}_t)$. A policy, π, observes the current state of the world, represented by the context feature $\mathbf{x}_t$. Using some arm-selection strategy based on the previous observation sequence, $H_t = \{(\mathbf{x}_i, r_{i,a_i})\}_{i=1}^{t-1}$, the policy chooses one of the available arms, defined as taking action $a_t \in \mathcal{A}$. Then, the policy gets the reward of the chosen arm, r_{t,a_t}. With the new observation, $(\mathbf{x}_t, r_{t,a_t})$, the policy updates its strategy. This is iterated until horizon T. The goal of the policy π is to maximize its cumulative reward, $\sum_{t=1}^{T} r_{t,a_t}$. This is the unscaled version of the average expected reward (3.2).

One approach when having different contexts might be to apply a non-contextual MAB algorithm to each context. However, the relationship between the contexts is lost, so learning one context does not help learning the others as noted in Zhou (2016). In addition, this approach assumes that the contexts can be enumerated, which is not true when they are continuous.

The most extended approach to solving CMAB problems is to consider stochastic bandit algorithms, which assume that the reward of each arm follows an unknown probability distribution depending on the context. The contextual Epoch-Greedy (CEG) algorithm studied in Langford and Zhang (2007) address bandits as a classification task solving an empirical risk minimization problem to find the best policy given a context. Linear UCB (LinUCB), as proposed in Li et al. (2010) and Chu et al. (2011), extends the UCB algorithm (Auer et al. (2002)) to accommodate to context information. Lastly, logistic bandits model the probability of getting a reward, extending the classical logistic regression model to the bandits setting.

Contextual Epoch-Greedy

The Contextual Epoch-Greedy (CEG) algorithm considers two well differentiated steps of exploration and exploitation. During exploration, it gathers information by randomly selecting each of the available arms to obtain information about their rewards along the contextual variables. In this step, the goal is to create unbiased samples by randomly pulling all arms to improve the accuracy of learning. This step leads to large immediate regret, but can potentially be reduced for the exploitation

step. With the information obtained, a model $s : \mathbf{X} \mapsto \mathcal{A}$ is estimated in a class $\mathcal{S}$ of models to select the optimal action a_t given a context $\mathbf{x}$.

Once a satisfactory regret is obtained with the estimated model, the algorithm switches to the exploitation phase up to the horizon T. For a Bernoulli bandit, the expected regret in the exploration step compared to the optimal model can be as large as $O(1)$. This is the case when all chosen actions have a reward of 0 and the optimal arm has a reward of 1. In the exploitation step, the regret is expected to be much smaller. Suppose there are n exploration steps, where the average regret is e_n. Thus, the total regret is bounded by $n + (T - n)e_n$. Consequently, the optimal strategy is to switch from exploration to exploitation at the step n that minimizes the previous expression, as demonstrated in Langford and Zhang (2007).

LinUCB

The Linear Upper Confidence Bound (LinUCB) algorithm, studied in Li et al. (2010) and Chu et al. (2011), assumes that the expected reward of an arm k is linear with respect to the feature vector $\mathbf{x}_t$ through an unknown coefficient vector θ_k:

$$E[r_{t,k} \mid \mathbf{x}_t] = \theta_k' \mathbf{x}_t \tag{6.15}$$

In Li et al. (2010) it is shown that, under the linear assumption in equation (6.15), a confidence interval for the expected reward can be efficiently computed. Considering ridge regression, the parameter vectors θ_k are estimated as:

$$\widehat{\theta}_k = \left(\mathbf{D}_k' \mathbf{D}_k + I_d\right)^{-1} \mathbf{D}_k' \mathbf{r}_k$$

where $\mathbf{D}_k \in \mathcal{M}^{(t-1)\times d}$ is the design matrix whose rows correspond to the previously received instances $\{\mathbf{x}_i\}_{i=1}^{t-1}$ for the arm k, $\mathbf{r}_k$ the analog for the rewards obtained, and I_d is the $d \times d$ identity matrix. Then, as shown in Li et al. (2010):

$$\mathbb{P}\left(|\widehat{\theta}_k \mathbf{x}_t - E[r_{t,k} \mid \mathbf{x}_t]| \leq \alpha \sqrt{\mathbf{x}_t' (\mathbf{D}_k' \mathbf{D}_k + I_d)^{-1} \mathbf{x}_t}\right) \geq 1 - \delta \tag{6.16}$$

for any $\delta > 0$, where $\alpha = 1 + \sqrt{\ln(2/\delta)/2}$. The inequality in equation (6.16) gives a bound on the expected reward of an arm k. Extending the UCB logic of optimism

in the face of uncertainty, the policy in LinUCB selects the arm with the largest upper bound on the expected reward, as given by equation (6.16). The LinUCB algorithm is shown in Algorithm LinUCB.

Algorithm LinUCB Linear Upper Confidence Bound algorithm

Input Confidence parameter $\delta \in [0, 1]$
Initialize $\mathbf{A}_a \leftarrow I_d$
Initialize $\mathbf{b}_a \leftarrow \mathbf{0}_d$
for $t = 1, \cdots, T$ **do**
    for $a \in \mathcal{A}$ **do**
        Compute $\widehat{\theta}_a \leftarrow \mathbf{A}_a^{-1}\mathbf{b}_a$
        Compute $r_{t,a} \leftarrow \widehat{\theta}_a'\mathbf{x}_t + \left(1 + \sqrt{\ln(2/\delta)/2}\right)\sqrt{\mathbf{x}_t'(\mathbf{x}_t\mathbf{x}_t' + I_d)^{-1}\mathbf{x}_t}$
    end for
    Compute $a_t = \arg\max_{a \in \mathcal{A}} r_{t,a}$
    Receive Reward r_{a_t}
    Update $\mathbf{A}_a \leftarrow \mathbf{A}_a + \mathbf{x}_t\mathbf{x}_t'$
    Update $\mathbf{b}_a \leftarrow \mathbf{b}_a + r_{a_t}\mathbf{x}_t$
end for
Output Reward parameters $\widehat{\theta}_a$

Logistic bandits

The logistic bandit (LB) model extends Bernoulli bandits by incorporating generalization across actions through the context $\mathbf{x}_t$. In the Bernoulli bandit problem, the agent chooses an action, a_t, and receives a binary feedback, $y_t \in \{0, 1\}$, Bernoulli distributed. As in logistic regression, the conditional probability of the response variable is modeled with a logistic function through a linear combination of the context vector. For each arm k, the probability of receiving a reward is modeled as $P(Y = 1 \mid \mathbf{X} = \mathbf{x}) = s(\mathbf{x}; \theta_k) = (1 + \exp(-\theta_k'\mathbf{x}))^{-1}$, where $\theta_k \in \mathbb{R}^d$ is the parameter vector to be estimated for each arm $k \in \{1, \cdots, K\}$.

For the logistic bandit, at each step t, the parameter θ_k associated with each arm k is estimated as the maximizer of the likelihood function given the available information (as in the classical logistic model introduced in Section 2.2.1):

$$\widehat{\theta}_{k,t} = \arg\max_{\theta} \ \mathcal{L}^L_{k,t}(\theta) \tag{6.17}$$

where

$$\mathcal{L}^L_{k,t}(\theta) = \frac{1}{t}\sum_{i=1}^{t} I(a_i = k)\left\{y_i \ln[s(\mathbf{x}_i;\theta)] + (1-y_i)\ln[1-s(\mathbf{x}_i;\theta)]\right\} \tag{6.18}$$

The main difference with the classical logistic model is the exploration carried out by the logistic bandit and the continuous updating as more information is available. The exploration is imposed by considering Thompson Sampling (TS) as introduced in Thompson (1933). In addition to TS, other proposals are available to allow the agent to explore. For example, OL2M (Zhang et al. (2016)), PG-TS (Dumitrascu et al. (2018)), or Logistic-UCB (Faury et al. (2020)), among others. These algorithms are based on Upper Confidence Bound (UCB) strategies, except for PG-TS, which is based on Gibbs sampling. However, in Sutton and Barto (2018), Thompson Sampling is found to be more effective than these strategies, for which further approaches are not considered.

In TS, at each step t, the parameter vector, $\theta_{k,t}$, is treated as a random variable sampled from the distribution of distribution of $\widehat{\theta}_{k,t}$, estimated from the available information. The sampled estimators are used to estimate the probability of receiving a reward on each arm k as $s(\mathbf{x};\theta_{k,t})$, and the action that maximizes the expected reward is selected. Feedback on the selected action is received, and the new information is used to update the posterior distribution of $\widehat{\theta}_{k,t}$ using Bayes' rule as as shown in Russo et al. (2020) and Dumitrascu et al. (2018). For unexplored arms, the tails of the distribution are heavier, favoring exploration. As more information is obtained, the uncertainty is reduced, favoring the exploitation of the most profitable arms.

Figure 6.2 shows an example of the resampling distribution considered at different steps of TS. The dotted line represents the estimator at each step and the curves

represent the density over which it is resampled. In the early steps, there is a lot of uncertainty, so the density is wider, favoring exploration. As more information becomes available, the density of the estimator becomes more concentrated, favoring exploitation. Likewise, as more information becomes available, the estimator is expected to be closer to the true parameter maximizing the expected reward.

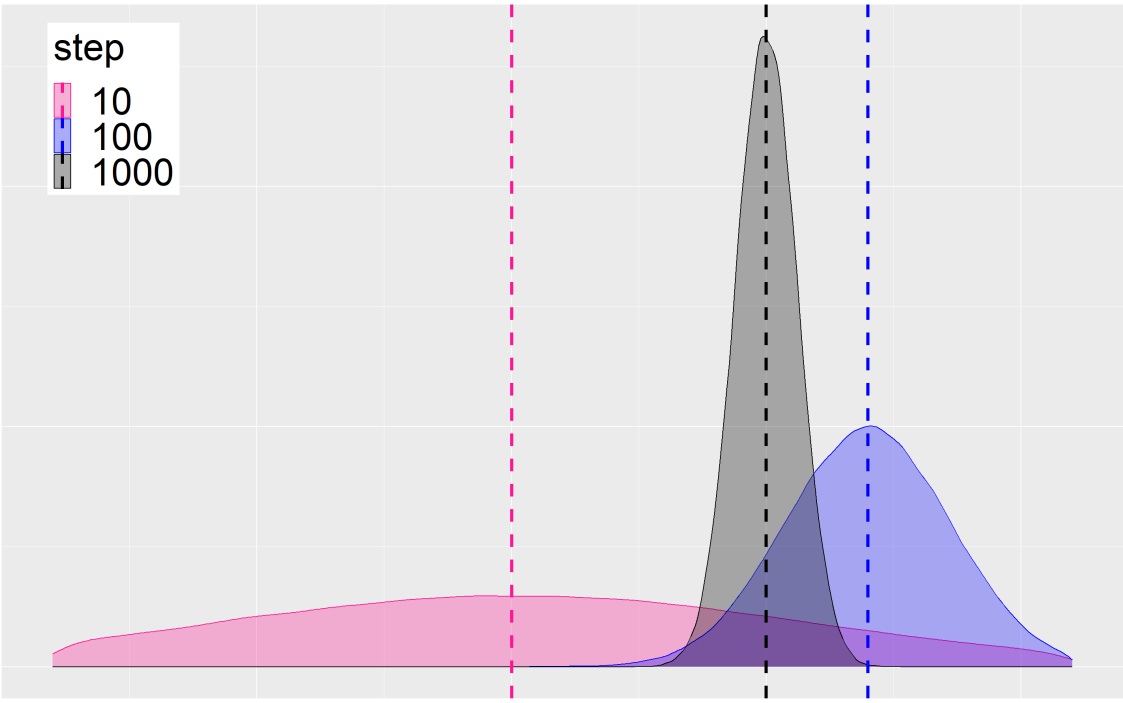

Figure 6.2: Thompson sampling resampling densities for different steps.

The drawback of the logistic bandit algorithm is that it results in a computationally intractable posterior, which makes the implementation of TS challenging, as noted in Dumitrascu et al. (2018). To address this, the Laplace approximation is used, which is based on a Taylor series expansion of the distribution function up to the second order. For the Laplace approximation to work well, the distribution of θ should be close in shape to a normal distribution, as the approximation would ignore the skewness and the secondary mode as noted in Gamerman and Lopes (2006). This is expected in the logistic framework, with a smooth density peaked around its mode, as stated in Dumitrascu et al. (2018). There are other approaches to performing the sampling process, such as Metropolis Hasting or Langevin Monte Carlo Markov Chain. However, in Visantavarakul (2022), no significant improve-

ment is observed with these approaches, for which no sampling algorithms other than the Laplace approximation are considered.

After estimating the parameter θ_k with $\widehat{\theta}_{k,t}$ as in equation (6.17), considering the Laplace sampling approximation, the parameter used in the step $t+1$ is resampled from the distribution:

$$\theta_{k,t+1} \sim N_d(\widehat{\theta}_{k,t}, H_{k,t}) \tag{6.19}$$

where $H_{k,t}$ is the Hessian of the likelihood function in equation (6.18) evaluated at $\widehat{\theta}_{k,t}$. This process is analogous to the one represented in Figure 6.2. Considering the Laplace sampling approximation, the logistic bandit algorithm is illustrated in Algorithm LB.

Algorithm LB Logistic bandit algorithm

Initialize $\theta_{k,0} \leftarrow \mathbf{0}_d$
Initialize $H_0 \leftarrow I_d$
for $t = 1, \cdots, T$ **do**
    Compute $\theta_{k,t} \sim N_d(\widehat{\theta}_{k,t-1}, H_{k,t-1}),\ k \in A_t$
    Compute $a_t = \arg\max_{a \in \mathcal{A}} s(\theta_{a,t}; \mathbf{x})$
    Receive Result y_t
    Update $\widehat{\theta}_{k,t} = \arg\max_{\theta}\ \mathcal{L}^L_{k,t}(\theta)$, where $\mathcal{L}^L_{k,t}$ as defined in (6.18)
    Update $H_{k,t} = H\mathcal{L}^L_{k,t}(\widehat{\theta}_{k,t})$
end for
Output Parameters $\widehat{\theta}_{k,T}$

6.3.2 Cost-sensitive logistic bandit

So far, the bandit algorithms in the state-of-the-art have been introduced. In this subsection, we propose the cost-sensitive logistic bandit (CSLB) algorithm to solve the credit risk problem introduced in Section 6.1, extending the logistic bandit introduced in Algorithm LB as suggested in C-Rella et al. (2024c). It is assumed that a certain number of borrowers apply for loans and an agent has to sequentially

select which borrowers to approve for credit. The applications are studied one by one, individually, and the decision is binary: loan approval or denial. Consequently, the credit risk problem is reasonably framed within the logistic bandit framework.

The set of actions for the credit risk problem considered in this chapter is binary, $\hat{Y} \in \{0, 1\}$, where 0 stands for granting the loan and 1 for denying it. This can be thought of as a Bernoulli one-armed bandit, where the agent has to decide whether to pull the arm or not. If the operation is approved, the customer's behavior is observed, otherwise no information is obtained for this application. The goal is to maximize the total reward, which depends on the characteristics of each instance, as represented by the reward function in equation (3.1).

Since the set of actions is binary (approve or deny the application), we can model the problem with a single parameter θ such that $\hat{y}_t = I(s(\theta; \mathbf{x}_t) > h_t)$, where h_t is the decision threshold for the instance $(\mathbf{x}_t, w_t)$. Considering the expected reward of an action,

$$R(\hat{y} \mid \mathbf{x}, w) = r(\hat{y}, w, 0)(1 - s(\mathbf{x})) + r(\hat{y}, w, 1)s(\mathbf{x})$$

the arm is pulled if the expected reward of this action is greater than not pulling the arm, i.e. if $R(0 \mid \mathbf{x}, w) \geq R(1 \mid \mathbf{x}, w)$. This logic leads to the Bayes decision threshold introduced in equation (2.7):

$$h_i = \frac{C_i^{FP} - C_i^{TN}}{C_i^{FP} - C_i^{TN} - C_i^{TP} + C_i^{FN}}$$

Considering the logistic classifier, assume that $E[Y \mid \mathbf{X} = \mathbf{x}] = s(\mathbf{x}; \theta) = (1 + \exp(-\theta'\mathbf{x}))^{-1}$, where $\theta \in \mathbb{R}^d$. The objective is to estimate the parameter θ that maximizes the expected reward $E[r(\hat{Y}, W, Y)]$. As introduced in Section 4.2, assuming that $\hat{Y} = \mathbb{I}(s(\mathbf{X}) > u)$, with $u \sim U[0, 1]$, then $E[\hat{Y} \mid \mathbf{X} = \mathbf{x}] = P(\hat{Y} = 1 \mid \mathbf{x}) = s(\mathbf{x}; \theta)$. Thus, extending the proposals in Höppner et al. (2021) and Stripling et al. (2018), the *average expected reward* (AER) of a classifier s over the set sample $\mathcal{Z}_t = \{\mathbf{z}_i = (\mathbf{x}_i, w_i, y_i)\}_{i=1}^t$ can be defined as:

$$\mathcal{L}_t^{CS}(\theta) = \frac{1}{t} E\Big[\sum_{i=1}^{t} r(\hat{y}_i, w_i, y_i) \mid \mathcal{Z}_t\Big] = \frac{1}{t}\sum_{i=1}^{t} r(s(\mathbf{x}_i; \theta), w_i, y_i) \tag{6.20}$$

which depends only on θ. The AER in equation (6.20) corresponds to the empirical version of $E[r(\hat{Y}, W, Y)]$ given the sample $\mathcal{Z}_t = \{\mathbf{z}_i = (\mathbf{x}_i, w_i, y_i)\}_{i=1}^t$. Thus, the estimator of θ at each step $t+1$ is defined as:

$$\widehat{\theta}_t = \arg\max_{\theta} \mathcal{L}_t^{CS}(\theta) \tag{6.21}$$

When introducing mislcassification costs in the objective function (6.20), in practice it is usually not convex nor has a unique minimum due to its dependence on W, Y, and θ through $\hat{Y}$ as shown in Section 4.3.2. To allow the optimization problem to be solved, a regularizer is added to the objective function to force convexity as suggested in Höppner et al. (2021); Stripling et al. (2018) and Bahnsen et al. (2013). This facilitates numerical optimization techniques to avoid suboptimal solutions. Thus, it is considered the regularized AER:

$$\mathcal{L}_t^{\lambda}(\theta) = \frac{1}{t}\sum_{i=1}^{t} r(s(\mathbf{x}_i;\theta), w_i, y_i) - \lambda\|\theta\|_1 \tag{6.22}$$

where $\lambda \in \mathbb{R}^+$ is the parameter controlling the regularizer effect. The regularizer considered is the lasso regularizer following the proposal in Höppner et al. (2021). Other regularizers are not explored without loss of generality. The corresponding estimator is defined as:

$$\widehat{\theta}_t = \arg\max_{\theta} \mathcal{L}_t^{\lambda}(\theta) \tag{6.23}$$

Introducing the regularized AER (6.22) as the objective function in the logistic bandit in Algorithm LB and considering the Bayes decision rule in equation (2.7), the CS logistic bandit (CSLB) is represented in Algorithm CSLB. At each step, the optimal action from a CS perspective is selected by considering the estimator in equation (6.23) and the decision threshold in equation (2.7). Similarly, by applying the sampling strategy provided by TS, all contexts are efficiently explored to further improve the long-term reward. Consequently, the CSLB algorithm is expected to improve the performance of previous models by combining the advantages of exploration with a CS approach. The performance of the proposed algorithm is evaluated in the next section.

Algorithm CSLB Cost-sensitive logistic bandit algorithm

Input Reward function, $r(\hat{y}_i, w_i, y_i)$ as in equation (3.1)

Initialize $\theta_0 \leftarrow \mathbf{0}_d$

Initialize $H_0 \leftarrow I_d$

for $t = 1, \cdots, T$ **do**

  Compute $\theta_t \sim N_d(\widehat{\theta}_{t-1}, H_{t-1})$

  Receive Context, $(\mathbf{x}_t, w_t)$

  Compute $h_t = \frac{C_t^{FP} - C_t^{TN}}{C_t^{FP} - C_t^{TN} - C_t^{TP} + C_t^{FN}}$

  Compute $a_t = I\left(s(\theta_t; \mathbf{x}_t) \geq h_t\right)$

  Receive Feedback, y_t

  Update $\widehat{\theta}_t = \arg\max_\theta \ \mathcal{L}_t^\lambda(\theta)$, where $\mathcal{L}_t^\lambda$ as defined in (6.22)

  Update $H_t = H\mathcal{L}_t^\lambda(\widehat{\theta}_t)$

end for

Output Reward parameter $\widehat{\theta}_T$

6.4 Experimental results

The performance of the introduced OL and bandit algorithms are evaluated considering a wide range of simulated settings and the analysis of five real credit risk data sets. The objective is to understand how the different algorithms behave as the proportion of cases, the dependency relationship and the cost specification vary.

Online learning and bandit algorithms represent two different methodologies, particularly due to the exploration performed by bandit algorithms. Therefore, the algorithms are evaluated separately. After introducing the datasets, the results obtained for the different algorithms are presented, together with some comments on the performance of the classifiers.

Regarding OL algorithms, the passive-aggressive (PA) algorithm detailed in Algorithm PA, the CSOGD algorithm in equation (6.9), the PAUM algorithm as depicted in equation (6.10), the PAPB algorithm described in (6.8) and the proposed

Algorithm IDCSPA are considered. The selection of the aggressiveness parameter C in the PA algorithms is done by cross-validation (CV) over the grid $C \in (0, 0.001, 0.01, 0.1, 1, 10, 100)$. In addition, we evaluate a static approach (Static) considering a linear model such as the PA algorithm in equation (6.5), previously fitted to a static training sample, in line with the classical approach in credit risk. For this model, the parameter θ that minimizes the hinge loss (6.1) over a training sample is estimated beforehand. Thus, it is evaluated the improvement obtained with online learning algorithms with respect to a static approach. Similarly, by considering a cost-insensitive model such as the PA algorithm, a comparison is also made of the improvement that can be obtained by considering a CS approach in problems with different misclassification effects. Finally, this study examines in detail the behavior of the proposed IDCSPA algorithm and compares it with the state-of-the-art alternatives.

Regarding bandit algorithms, we consider the logistic bandit described in Algorithm LB, the LinUCB algorithm summarized in Algorithm LinUCB and the CEG algorithm described in Section 6.3.1. We also consider the classical (Static) approach in the simulation study. This consists of training a logistic model on a training sample and then considering the fitted model on the new samples. This provides a comparison of the improvement that can be obtained by considering a dynamic approach with exploration versus a static approach. All these algorithms are compared with the CS logistic bandit proposed in Algorithm CSLB. For the proposed algorithm, the regularization parameter λ is estimated by cross-validation over the grid $\lambda \in (0, 0.001, 0.01, 0.1, 1, 2, 5, 10, 20)$. When considering a cost-insensitive model such as the logistic bandit algorithm, a comparison is also made of the improvement that can be obtained by considering a cost-sensitive approach in bandit algorithms. Finally, this study examines in detail the behavior of the proposed CSLB model and compares it with the state-of-the-art alternatives.

The optimization process for all the algorithms is implemented using the open source statistical software R (R Core Team (2021)).

For the OL algorithms, for a set $\{(\hat{y}_i, y_i, w_i)\}_{i=1}^n$, the savings, introduced in equa-

tion (1.2), is considered to evaluate model performance with ℓ as defined in equation (6.11). This metric provides a standardized measure of the losses prevented compared to the base case scenario, allowing for a reasonable comparison across different models and problems. The maximum value this metric can take is 1, in which case all observations are well classified. For the bandit algorithms, we evaluate model performance considering the AER introduced in equation (3.2). Since a scaled version of the same objective function is obtained with the same a/c and b/c ratios, the values $b = 10$ (the analysis cost) and $c = 0.75$ (the lost percentage produced by an undetected case) are fixed and $a = \{0.05, 0.1, 0.2\}$ (the benefit per operation) in the cost matrix in Table 6.1. This provides valuable insights into model performance under different false positive and false negative rates.

Model performance is evaluated over the entire sample, taking into account the savings (1.2) or the AER (3.2) as appropriate, the positive predicted proportion (PP), the precision ($PPV = TP/PP$), the recall $= TP/(TP+FN)$ and the F score. Thus, model performance is evaluated using both cost-sensitive and classification metrics.

6.4.1 Simulation study

This section explores the behavior of the algorithms presented in this chapter in a simulation study. By controlling the data generation process and varying both the data dependency structure and the cost specification, the goal is to gain insight into how each approach performs under different problem conditions. This simulated environment allows us to observe and compare the strengths and weaknesses of each algorithm, providing valuable guidance for selecting the most appropriate method for specific real-world applications. The simulated scenarios consider the credit risk problem introduced in section 6.1.

Simulation models

For the simulation study the data is simulated based on six different models, which are summarized in Table 6.2. For all simulations, the underlying model is given by $Y = \mu(v) + \epsilon$, where μ is a link function and $V = \theta_0'\mathbf{X}$ is a linear combination of the predictor covariates, $\mathbf{X} = (1, X_1, \ldots, X_6) \in \mathbb{R}^7$. For the Logit model, $X_i \sim N(0,1)$, $\mu(v) = \frac{e^v}{1+e^v}$ and $\epsilon \sim N(0,1)$. For the Squared model, $V = \theta_0'\mathbf{S}$, where $\mathbf{S} = (1, S_1, \ldots, S_6), S_j = X_j^2$. The Qubic model is constructed similarly, but defines $V = \theta_0'\mathbf{Q}$, with $\mathbf{Q} = (1, Q_1, \ldots, Q_6), Q_j = X_j^3/5$. For the models with polynomial terms, the dependence between the response variable and the covariates $\mathbf{X}$ is not monotonous. This violates the hypotheses of the classical classification models and makes modeling more difficult. For the Ker & Sam model, introduced in Ker and Sam (2018), $X_j \sim N(0,1)$, $j = 1,3,5$, $X_j \sim \chi_3^2$, $j = 2,4,6$ and ϵ is drawn from a normal mixture such that $\epsilon \sim N(3,1)$ or $\epsilon \sim N(-3,1)$ with probability 0.5. This corresponds to an homoscedastic model but with a bimodal error term. The Scobit model is defined as the skewed version of the Logit model, considering the link function $\mu(v) = 1 - \frac{1}{(1+e^v)^6}$. For the Klein & Spady model, introduced in Klein and Spady (1993), a heteroskedastic error term, $\epsilon \mid V \sim N(0, \sigma(V))$, is considered, with $\sigma^2(V) = 0.1(1+V^2)^2$. For all simulations, $\theta_0 = (\theta_{00}, \theta_{01}, \cdots, \theta_{06}) = (\theta_{00}, 1, 2, -1, -2, 3, -3)'$. Two scenarios are considered. In the normal scenario, θ_{00} is chosen so that $\bar{y} \approx 0.1$. In the risky scenario, θ_{00} is chosen so that $\bar{y} \approx 0.25$. Thus, the performance of the model at different levels of case proportions is also explored.

Model	Index	$\mu(V)$	ϵ
Logit	$V = \theta_0'\mathbf{X}$	$\frac{e^V}{1+e^V}$	$N(0,1)$
Squared	$V = \theta_0'\mathbf{S}$	$\frac{e^V}{1+e^V}$	$N(0,1)$
Qubic	$V = \theta_0'\mathbf{Q}/5$	$V(V-2)(V+2)$	$N(0,1)$
Ker & Sam	$V = \theta_0'\mathbf{X}$	V	$.5N(3,1) + .5N(-3,1)$
Scobit	$V = \theta_0'\mathbf{X}$	$1 - \frac{1}{(1+e^V)^10}$	$N(0,1)$
Klein & Spady	$V = \theta_0'\mathbf{X}$	V	$N(0, \sqrt{0.1}(1+V^2))$

Table 6.2: Data generation models for the simulation study.

The exogenous variable is assumed to be independent from the covariates following Zadrozny and Elkan (2001). It is simulated as $W = 100W_1W_2$, with $W_1 \sim \chi^2_4$ and $W_2 \sim N(7, 0.8)$, where W_1 and W_2 are independent. The idea is to replicate the asymmetry commonly encountered in financial problems such as the one motivating this work.

Six models (as introduced in Table 6.2), three cost settings ($a = \{0.05, 0.1, 0.2\}$) and two scenarios (normal and risky) are combined to generate 36 simulation settings. This provides a comprehensive test of the performance of the introduced classifiers under different dependency relationships and cost settings. For each scenario, samples are simulated with $n = 10,000$ and the algorithms are trained as introduced in Sections 6.2 and 6.3. This is repeated 20 times for each simulation scenario.

The results obtained are summarized in the next two subsections, reported by averaging over the 20 runs performed on each setting. Due to the number of simulations performed, the results for the OL and bandit algorithms are summarized in Tables 6.6-6.11 in the Appendix. A summary of the results obtained with each of the algorithms is also included in the form of graphs in Figures 6.3-6.6. In this way we have a quick and visual representation of the performance of the models and the details are left for the appendix.

Online learning algorithms

This section evaluates the performance of the online learning algorithms presented in Section 6.2 over the simulated datasets presented in Table 6.2. Figure 6.3 shows the box plot of the savings (1.2) obtained with the 6 algorithms compared for the 6 simulation models and the 3 cost specifications for the normal scenario. Figure 6.4 is analogous for the risky scenario. The first thing that can be observed in Figures 6.3 and 6.4 is how the performance of the models varies significantly depending on the value of the parameter a. This is due to the greater impact of false positives. Since the online learning algorithms only take losses into account, their performance

decreases as the value of a increases. Something similar can be observed depending on the case proportion scenario. As expected, when there are more bad requests in the sample, the savings obtained are lower.

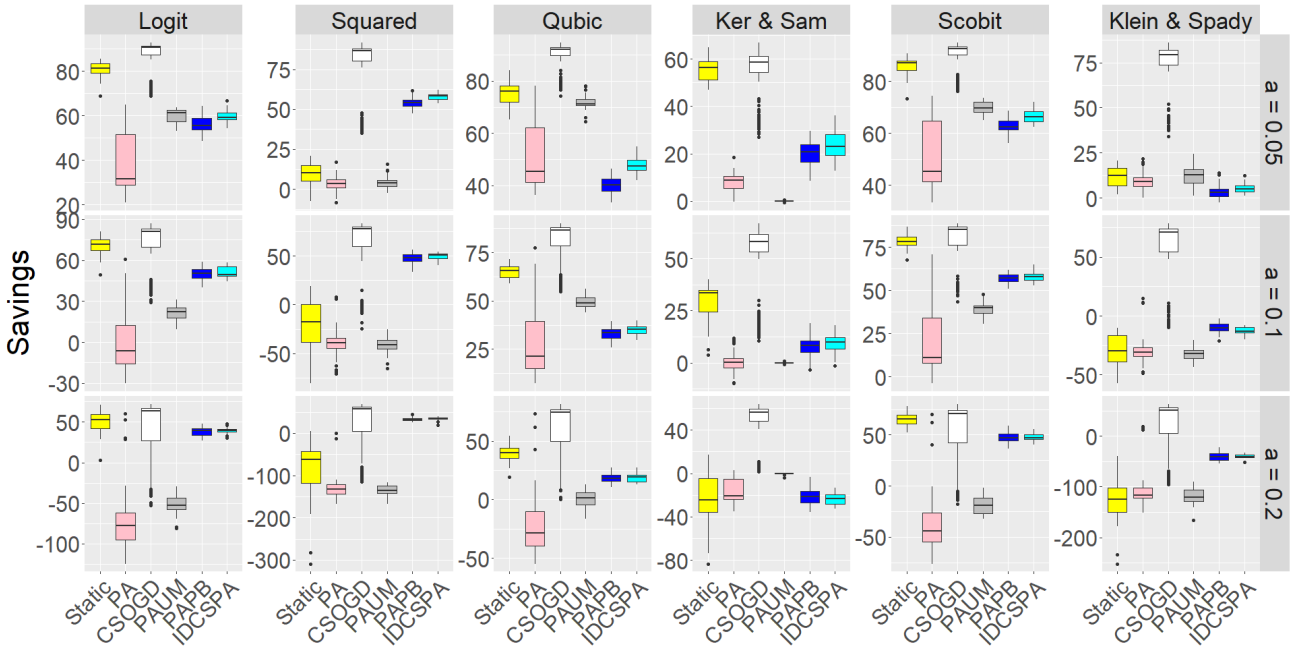

Figure 6.3: Box plots of the savings (1.2) for the OL algorithms in the normal scenario.

The static alternative performs well in many scenarios. However, especially in the normal scenario, the fact that the model is not updated with the new observations received means that the model perform poorly in scenarios where the underlying model deviates from linearity. Regarding the performance of the different OL classifiers, as expected, the cost-insensitive model (PA algorithm) has a worse overall performance.

Regarding the CSOGD model, we can see that it is the model that performs best in some scenarios, but still has a large variance in some settings. This is particularly the case for simulation models that are furthest from a linear model and for scenarios where the impact of errors is greater, i.e. as the value of a increases. This variability across scenarios and cost specifications means that while it may be a good alternative in certain scenarios, it must be treated with caution. Something similar happens with the PAUM model, although to a lesser extent.

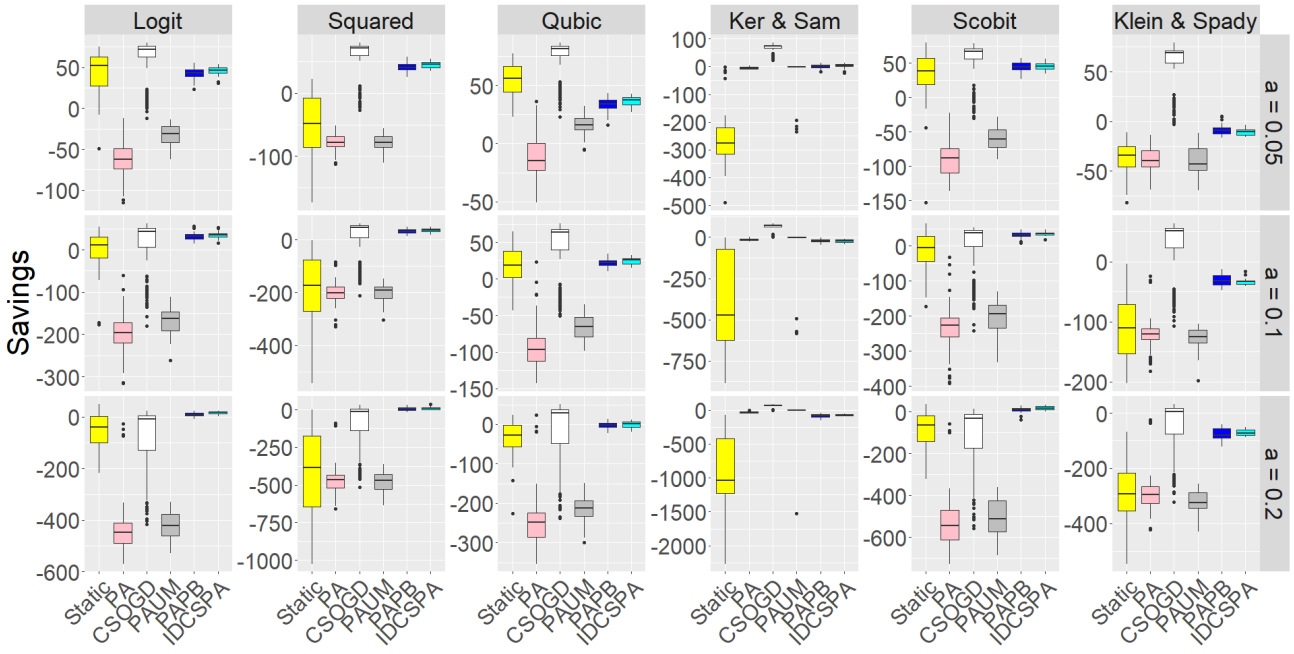

Figure 6.4: Box plots of the savings (1.2) for the OL algorithms in the risky scenario.

As for the CS passive-aggressive algorithms, namely PAUM, PAPB, and IDCSPA, their behavior is generally quite similar, with a general improvement with the instance-dependent algorithm. Thus, considering the effect of each observation individually leads to better results. The IDCSPA algorithm does not perform better than the state-of-the-art in the normal scenario. This is because the model tends to mark many observations as cases relative to the true proportion of cases, especially when the proportion of cases in the sample is low. As the proportion of cases increases, the model has more information to update its predictions and achieves better performance. Therefore, the proposed model should be considered with caution in highly imbalanced scenarios.

Although the proposed IDCSPA model is not the best approach in all scenarios, it is the best in most of them and gives stable results regardless of the problem setting, as it can be seen in Tables 6.6-6.8. Likewise, it is also the model with the least variability in its performance as shown in Figures 6.3 and 6.4. In particular, it is worth noting that the proposed IDCSPA algorithm consistently performs better than the other passive-aggressive algorithms. Taking this into account, Algorithm

IDCSPA is considered to be the best performing model in the simulation study. Therefore, it is an interesting model when dealing with dynamic problems, especially in scenarios where the interpretability of the predictions is important, such as the credit risk problem, due to the simplicity of the model, which can be reduced to a linear classifier.

Bandit algorithms

This section evaluates the behavior of the bandit algorithms presented throughout this chapter over the simulated datasets presented in Table 6.2. Results are summarized in Figures 6.5 and 6.6 and detailed in the Appendix in Tables 6.9-6.11. Figure 6.5 shows the mean cumulative reward obtained over the 20 simulations with the 5 considered algorithms for the 6 simulation models and the 3 cost specifications for the normal scenario. The X-axis represents the step and the Y-axis represents the cumulative reward up that step. Figure 6.6 is analogous for the risky scenario.

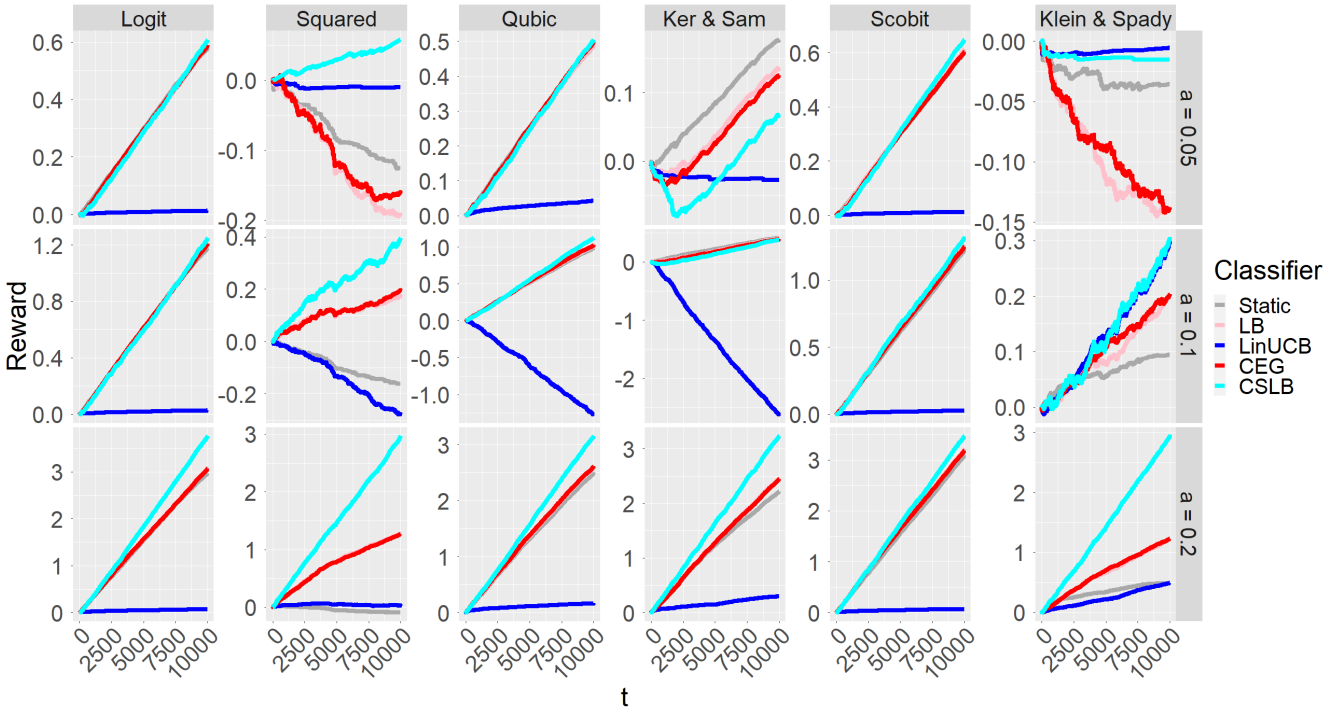

Figure 6.5: Simulation results (cumulative reward) for the bandit algorithms in the normal scenario.

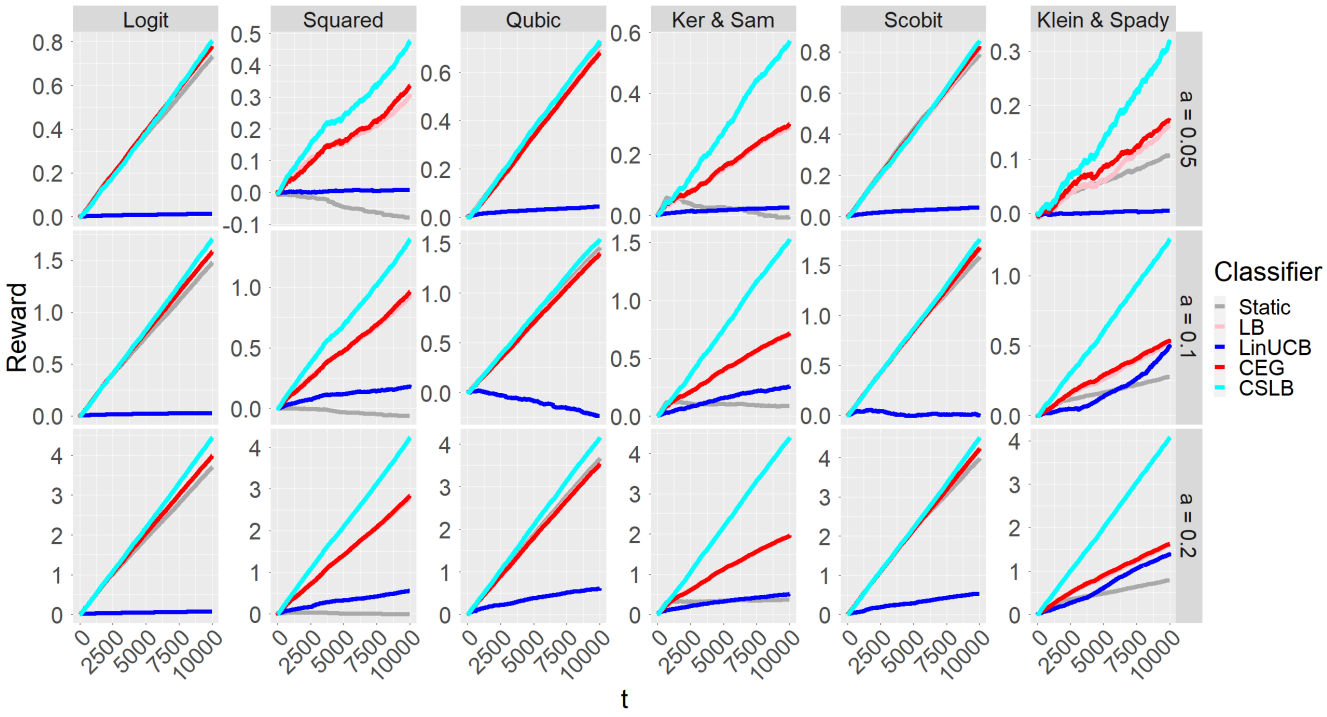

Figure 6.6: Simulation results (cumulative reward, in millions) for the bandit algorithms in the risky scenario.

For the bandit algorithms, as for the OL algorithms, we can see how the proportion of cases and the cost specification significantly affect the performance of the models. In this case, since we include both profits and losses in the objective function, the average reward increases as a does, which is the profit made on an approved good operation, increases.

The first thing that stands out from Figures 6.5 and 6.6 is the poor behavior of the LinUCB algorithm. This is explained by the fact that this algorithm only considers the context to model the expected reward. The exogenous variable is assumed to be independent of the covariates. Therefore, the model makes a naive prediction in all scenarios. Nevertheless, the optimistic approach seems to yield good results in some scenarios.

It is worth highlighting the good performance of the static model in some scenarios, such as Logit, Qubic or Scobit, despite the fact that it does not update predictions or take exploration into account. In the case of the Logit scenario, this good behavior is justified because the model is well specified. In the other two,

despite the fact that the model is not well specified, the dependence relationship is similar to the assumed by the logistic model as can be seen in Figure 5.1, which also makes the model behave well. As the misspecification of the logistic model is further broken down, its behavior deteriorates significantly, as can be seen in the Squared or Klein & Spady scenarios. In these simulations, the exploration performed by the LB and CEG models allows them to adapt much better to the problem. This is reflected in the higher expected rewards despite not considering a CS approach for model fitting.

With respect to the proposed CSLB algorithm, it can be seen that it practically achieves the best results in all simulations when compared to the state-of-the-art algorithms, both cost-insensitive and cost-sensitive. The biggest differences are observed as the value of a increases, where the algorithm seems to be able to increase the profits even more than the other models, thanks to a higher profit per good operation. It is noteworthy that the proposed CSLB model, unlike the other models, obtains better results in the most risky scenario. This is due to the fact that the model explores a greater number of cases in the first steps. This means that in the long run, the CSLB algorithm has a more representative sample. Thus, the model is more accurate and better results are obtained.

The only simulation scenario where the proposed model performs worse than state-of-the-art models is the Ker & Sam simulation with $a \in \{0.05, 0.1\}$ and the normal scenario. In this setting, the model produces a fairly significant loss in the early iterations, which is corrected as the model progresses. The problem is that these initial losses do not seem to be compensated in the run considered. Perhaps, given the results obtained in the other simulation models, the CSLB will outperform the other approaches in a larger time frame.

It is worth highlighting the Klein & Spady simulation, where all the algorithms seem to have problems to obtain a good classification. In fact, for the normal case with $a = 0.05$, none of the models is able to obtain profits. This is because it is the most difficult setting to model and corresponds to a scenario with limited potential benefits (small a).

Except for the Ker & Sam simulation, where the proposed model performs slightly worse than the state-of-the-art models, the proposed CSLB obtains the best results in all simulations, regardless of the proportion of cases, relationship dependencies, and cost specification. Taking this into account, it is considered that the proposed CSLB algorithm achieves satisfactory performance and is an interesting alternative for cost-sensitive dynamic problems considering exploration.

6.4.2 Real datasets study

This section explores the behavior of the reinforcement learning algorithms presented over various real credit risk data sets. The objective is to asses the performance of the different algorithms in real scenarios once the good performance has been shown in simulated settings. After introducing the datasets, the results obtained for the different algorithms are presented, together with some comments on the performance of the classifiers.

The different datasets are selected to investigate the impact of class imbalance and the number of covariates on model performance. As none of the state-of-the-art data sets provide guidance on the misclassification error costs, and to preserve the confidentiality of our collaborating entity, several reward function (3.1) specifications are considered. Likewise, the effect of different ratios of false positive and false negative costs on model performance is studied. The values of the reward function are taken in line with those considered in previous chapters, i.e. the cost matrix in Table 6.1 is considered with $a \in \{0.05, 0.10, 0.20\}$, $b = 10$ and $c = 0.75$.

For each data set, a 5-fold cross-validation is performed according to Höppner et al. (2021), stratified by case proportion and amount. The mean results over the test sets are summarized in Tables 6.3-6.5 for the OL algorithms and in Tables 6.12-6.14 for the bandit algorithms in the Appendix. A summary of the results obtained with the bandit algorithms is also included in Figure 6.7.

Datasets

Five real credit risk data sets are considered in order to study the introduced algorithms performance in practical scenarios. All datasets correspond to a real credit risk problem where some covariates are available to model a response variable (fraud or default). The objective is to minimize losses/maximize benefits, which depend on the amount of the transaction and the cost specification.

The first data set (Credit Card data set[1]), introduced in Section 2.5.1, is a fraud data set containing 28 variables resulting from a PCA along with a "Time" variable (seconds elapsed since the first use). The data set is undersampled so that they remain $15,000$ transactions containing 465 (3.19%) frauds. The second one (GCD data set[2]) is a real credit risk data set of size $n = 1,000$ with a case proportion of 30% and 24 variables available to model the default indicator. The third one, (PAKDD data set[3], as introduced in Linhart et al. (2009)), consists of $11,111$ prospective clients for which 4 numerical variables are available to model their probability of default. There are $3,047$ (27.42%) defaults in the sample. The amount of the loan is simulated as $W \sim 100\chi_3$ in order to replicate the asymmetry commonly encountered in credit risk problems. The fourth one (ASF data set) is lend by ASF. It is composed of $23,377$ loan requests for a new product collected between January 2022 and December 2022 with a default percentage of 3.06%. To preserve confidentiality, the number of registers of the original data set is truncated in order to change the default proportion. It consists on 13 numerical variables which include socioeconomic information of the client, characteristics of the operation and attributes of the point of sale where the financing is requested. The fifth dataset consists of a set of data synthetically generated from the ASF dataset, where concept and covariate drifts are introduced to test the performance of the different models in the presence of drifts. To do this, we follow Nowok et al. (2016) considering the `syn` function of

[1] `https://kaggle.com/mlg-ulb/creditcardfraud`

[2] `https://github.com/JLZml/Credit-Scoring-Data-Sets/tree/master/1.%20UCI%20Repository/German`

[3] `https://github.com/JLZml/Credit-Scoring-Data-Sets/tree/master/2.%20PAKDD%202009%20Data%20Mining%20Competition`

the R package synthpop. First, a sample of size 15,000 is synthetically generated from the ASF data. To this data set we add 15,000 observations generated from a different distribution for the covariates, but preserving the dependence between $\mathbf{X}$ and Y. Another 15,000 instances are added to this sample, introducing concept drift in this step, generating the response variable with a different model than the one considered for the first 30,000 observations. Finally, 15,000 synthetically generated observations from the original ASF data set are added to form the final sample.

Online learning algorithms

This section evaluates the performance of the online learning algorithms presented in Section 6.2 over the real datasets presented in the previous subsection. Tables 6.3-6.5 summarize the savings obtained with the 6 algorithms compared for the 5 datasets and the 3 cost specifications.

It can be observed in Tables 6.3-6.5 how the performance of the models varies significantly depending on the value of the parameter a. This is due to the greater impact of false positives. Since the online learning algorithms only take losses into account, their performance decreases as the value of a increases. Something similar can be observed depending on the case proportion scenario. As expected, when there are more bad requests in the sample, the savings obtained are lower.

In line with the findings of the simulation study, the CSOGD algorithm consistently demonstrates the best overall performance. The proposed IDCSPA model is the next best performing algorithm across all scenarios. Thus, the CSOGD algorithm seems to be the most reasonable choice for this type of problem. Although the IDCSPA algorithm does not outperform the CSOGD, it achieves the best results within the passive-aggressive models. This suggests that incorporating the instance-dependent loss function into the training process improves the performance of OL algorithms. These results corroborate in practical scenarios the results of the simulation study and indicate the potential for extending other online learning algorithms to cost-sensitive cases to achieve further improvements.

Data set	Classifier	Sav	Acc	Rec	PPV	PP	FS
Credit card	Static	-52.36	43.23	11.18	0.66	54.30	1.24
	OL	-20.05	52.65	52.90	3.55	47.54	6.65
	CSOGD	63.71	73.93	98.44	11.36	29.16	20.28
	PAUM	-10.85	50.14	68.39	4.27	51.03	8.04
	PAPB	62.44	96.72	79.78	49.14	5.17	60.82
	IDCSPA	63.68	96.91	78.28	50.98	4.89	61.75
GCD	Static	27.03	53.65	25.00	23.96	31.33	24.47
	OL	31.82	65.57	35.67	41.47	25.83	38.35
	CSOGD	74.96	91.42	79.58	90.69	26.35	84.78
	PAUM	0.49	69.67	0.67	28.57	0.70	1.30
	PAPB	34.17	63.16	45.00	39.94	33.83	42.32
	IDCSPA	34.78	62.86	45.33	39.65	34.33	42.30
PAKDD	Static	34.96	54.85	42.63	28.21	41.13	33.95
	OL	22.76	61.01	26.28	27.43	26.08	26.84
	CSOGD	76.10	86.98	76.77	75.72	27.60	76.24
	PAUM	-0.01	72.77	0.00	0.00	0.01	
	PAPB	0.04	72.77	0.03	33.33	0.02	0.05
	IDCSPA	19.95	63.78	22.08	28.60	21.02	24.92
ASF	Static	-32.71	68.59	32.03	3.23	30.31	5.87
	OL	-17.03	61.64	58.47	5.05	38.93	9.31
	CSOGD	58.72	82.74	87.96	13.83	19.59	23.9
	PAUM	-3.85	63.75	77.78	6.31	37.96	11.67
	PAPB	-4.44	90.56	14.81	6.27	7.27	8.82
	IDCSPA	-4.40	90.50	14.81	6.23	7.33	8.77
Drift	Static	-22.70	83.35	5.33	1.44	13.41	2.27
	OL	-3.78	92.18	3.81	3.10	4.47	3.42
	CSOGD	72.87	98.03	71.95	73.36	3.56	72.65
	PAUM	-1.42	95.27	0.55	1.75	1.14	0.84
	PAPB	27.44	92.99	35.77	21.71	5.98	27.02
	IDCSPA	39.52	93.00	50.92	26.16	7.07	34.56

Table 6.3: Savings (Sav), accuracy (Acc), recall (Rec), positive predicted value (PPV), positive predicted proportion (PP) and F-score (FS) for the OL algorithms for the real datasets and cost setting $a = 0.05$.

Data set	Classifier	Sav	Acc	Rec	PPV	PP	FS
Credit card	Static	-151.97	42.72	12.69	0.74	54.90	1.39
	OL	-103.49	51.51	50.32	3.31	48.51	6.20
	CSOGD	30.12	72.32	98.82	10.77	30.80	19.35
	PAUM	-69.31	48.34	72.26	4.34	53.08	8.19
	PAPB	54.48	97.22	79.78	54.32	4.68	64.63
	IDCSPA	59.12	97.10	79.14	53.03	4.76	63.50
GCD	Static	5.74	59.26	15.67	23.38	20.12	18.76
	OL	21.18	64.96	35.33	40.46	26.23	37.72
	CSOGD	70.22	90.22	75.00	90.82	24.80	82.15
	PAUM	-0.13	69.87	0.00	0.00	0.10	
	PAPB	26.96	62.36	43.00	38.62	33.43	40.69
	IDCSPA	27.35	62.76	43.67	39.22	33.43	41.32
PAKDD	Static	12.69	60.18	23.01	24.93	25.12	23.93
	OL	17.52	61.42	25.76	27.63	25.38	26.66
	CSOGD	73.77	86.98	77.24	75.51	27.84	76.36
	PAUM	-0.01	72.74	0.03	14.29	0.05	0.05
	PAPB	-0.00	72.77	0.00	0.00	0.01	
	IDCSPA	14.16	63.76	21.45	28.21	20.70	24.37
ASF	Static	-64.39	78.35	12.31	1.95	19.34	3.36
	OL	-91.36	60.75	61.98	5.34	40.07	9.83
	CSOGD	30.67	84.11	86	13.67	17.94	23.59
	PAUM	-75.38	66	66	5.39	34.91	9.97
	PAPB	-12.8	91.19	18	7.35	6.99	10.43
	IDCSPA	-6.13	91.99	21	9.42	6.36	13
Drift	Static	-49.56	83.35	5.33	1.44	13.41	2.27
	OL	-10.99	92.18	3.81	3.10	4.47	3.42
	CSOGD	71.37	98.03	71.95	73.36	3.56	72.65
	PAUM	-3.21	95.27	0.55	1.75	1.14	0.84
	PAPB	17.72	92.54	36.96	20.59	6.52	26.45
	IDCSPA	29.05	92.69	52.11	25.35	7.46	34.10

Table 6.4: Savings (Sav), accuracy (Acc), recall (Rec), positive predicted value (PPV), positive predicted proportion (PP) and F-score (FS) for the OL algorithms for the real datasets and cost setting $a = 0.1$.

Data set	Classifier	Sav	Acc	Rec	PPV	PP	FS
Credit card	Static	-178.65	56.18	9.68	0.75	41.25	1.39
	OL	-239.81	51.99	50.32	3.34	48.03	6.26
	CSOGD	-53.32	71.78	98.06	10.55	31.28	18.97
	PAUM	-187.69	47.95	80.43	4.75	53.99	8.97
	PAPB	35.53	97.05	78.92	52.43	4.80	63.00
	IDCSPA	45.16	97.59	80.65	58.96	4.36	68.12
German	Static	14.14	59.76	44.00	36.07	36.64	39.64
	OL	13.62	65.17	34.67	40.62	25.63	37.41
	CSOGD	70.71	90.47	77.83	89.04	26.25	83.06
	PAUM	-0.16	69.77	0.00	0.00	0.20	
	PAPB	15.26	63.96	43.33	40.62	32.03	41.94
	IDCSPA	14.65	61.46	48.67	38.73	37.74	43.13
PAKDD	Static	7.96	56.07	33.49	26.09	34.94	29.33
	OL	7.75	61.79	25.38	27.84	24.81	26.56
	CSOGD	69.83	86.75	76.28	75.36	27.55	75.82
	PAUM	0.24	72.73	0.35	39.39	0.25	0.70
	PAPB	-0.00	72.77	0.00	0.00	0.01	
	IDCSPA	6.89	63.03	21.48	27.27	21.44	24.03
ASF	Static	-235.03	65.91	29.65	2.76	32.85	5.05
	OL	-265.14	60.87	59.84	5.23	39.82	9.62
	CSOGD	-42.39	82.89	84.3	14.93	19.48	25.37
	PAUM	-243.08	64.06	68.6	6.36	37.22	11.64
	PAPB	-49.54	89.79	17.36	7.53	7.96	10.5
	IDCSPA	-42.01	90.7	12.4	6.38	6.7	8.43
Drift	Static	-103.27	83.35	5.33	1.44	13.41	2.27
	OL	-25.41	92.18	3.81	3.10	4.47	3.42
	CSOGD	68.37	98.03	71.95	73.36	3.56	72.65
	PAUM	-6.80	95.27	0.55	1.75	1.14	0.84
	PAPB	-2.16	92.54	36.96	20.59	6.52	26.45
	IDCSPA	7.93	92.82	51.56	25.67	7.29	34.27

Table 6.5: Savings (Sav), accuracy (Acc), recall (Rec), positive predicted value (PPV), positive predicted proportion (PP) and F-score (FS) for the OL algorithms for the real datasets and cost setting $a = 0.2$.

Bandit algorithms

This section evaluates the behavior of the bandit algorithms presented throughout this chapter. Results are summarized in Figure 6.7 and detailed in the Appendix in Tables 6.12-6.14. Figure 6.7 shows the mean cumulative reward obtained with the 5 considered algorithms for the 5 datasets and the 3 different cost specifications. The X-axis represents the step and the Y-axis represents the cumulative reward up that step.

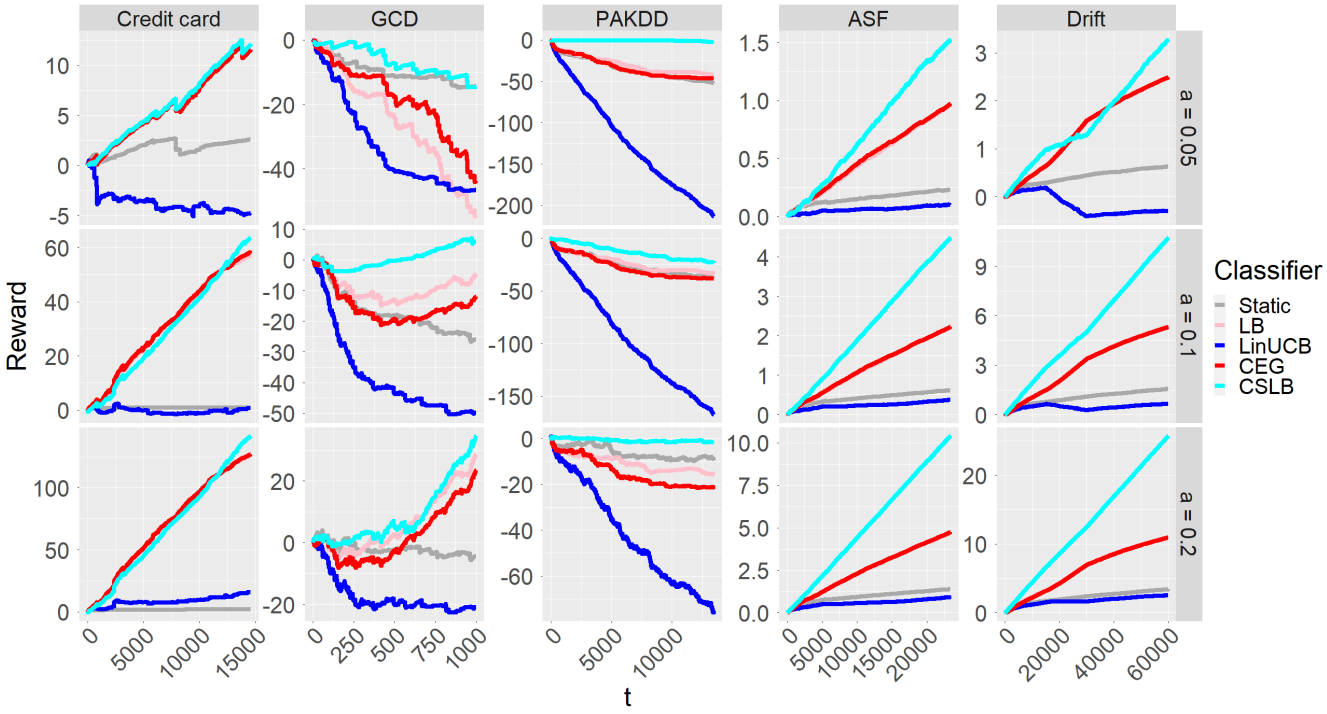

Figure 6.7: Real datasets results (cumulative reward, in thousands for the credit card, GCD and PAKDD datasets, and in millions for the ASF and drift datasets) for the bandit algorithms.

For the bandit algorithms, as for the OL algorithms, we can see how the proportion of cases, the number of covariates and the cost specification significantly affect the performance of the models. In this case, since we include both profits and losses in the objective function, the average reward increases as a, which is the profit made on an approved good operation, increases.

The first thing that stands out from Figure 6.7 is the poor behavior of the LinUCB algorithm again. Regarding the performance of the Static approach, excluding

the LinUCB algorithm, its performance is the worse. Thus, it is concluded that updating the models with all the available information and exploration significantly improves model performance, especially in real problems as the considered in this study.

With respect to the proposed CSLB algorithm, it can be seen that it achieves the best results in all the datasets and cost settings when compared to the state-of-the-art algorithms, both cost-insensitive and cost-sensitive. As for the simulation study, the biggest differences are observed as the value of a increases, where the CSLB algorithm seems to be able to increase the profits even more than the other models, thanks to a higher profit per good operation granted. This was observed also in the simulation study.

It is worth highlighting the PAKDD data set, where all the algorithms seem to have problems to obtain a good classification. In fact, none of the models is able to obtain profits in any cost setting. This is because it is the most difficult setting to model (only 4 covariates available and a default proportion of 27%). Something similar happens in the GCD data set with $a = 0.05$. Nevertheless, the proposed CSLB algorithm attains the best results among all the considered models.

Regarding the drift data set, it can be clearly seen a decline in model performance for all the algorithms as the drifts appear, specially in the step where it starts the concept drift. The proposed CSLB has a decline in performance, but thanks to the exploration performed, it is able to correct its estimates and obtain the best aggregated results in all the cost settings.

The proposed CSLB obtains the best results in all the problems, regardless of the proportion of cases, sample size, relationship dependencies, and cost specification. Taking this into account, it is considered that the proposed CSLB algorithm achieves satisfactory performance and is an interesting alternative for cost-sensitive dynamic problems as the credit risk problem.

6.5 Summary and conclusions

Motivated by the credit risk problem, this chapter presents two novel algorithms designed to tackle cost-sensitive dynamic classification. The PA algorithm, as introduced in Crammer et al. (2006), is extended to the CS setting in Algorithm IDCSPA. In addition, the logistic bandit approach is extended with Algorithm CSLB for efficient dynamic learning considering exploration. The performance of the proposed models is empirically tested against the state-of-the-art alternatives, considering a wide range of simulation settings and the study of five real datasets. Promising experimental results demonstrate the effectiveness and efficiency of the proposed models compared to state-of-the-art algorithms. Furthermore, this study provides valuable insights into the algorithmic performance in terms of case proportions, dependency relationships, and cost specifications.

For online learning algorithms, the cost-insensitive model (PA) shows inferior overall performance as it can be seen in the results in Figures 6.3 and 6.4 and in Tables 6.3-6.5. CSOGD and PAUM perform well in some scenarios, but poorly in others, especially where the impact of errors is greater. While PAPB and IDCSPA show similar behavior, the proposed IDCSPA algorithm emerges as the best performing model overall, offering consistent performance across different scenarios, as can be seen from the results in the Appendix. Thus, IDCSPA is considered a valuable alternative for dynamic problems, especially in contexts prioritizing interpretability, such as credit risk. Finally, it should be noted that the proposed IDCSPA algorithm performs consistently better than the other passive-aggressive algorithms. In this way, it is concluded that extending online learning algorithms with a cost-sensitive approach improves the results obtained, opening the door to the extension of more algorithms to the CS setting.

As for the bandit algorithms, the LinUCB algorithm performs poorly because it only considers context for modeling expected rewards. Despite not updating its predictions, the static approach yields a good performance in simulation models such as Logit, Qubic, and Scobit due to a (nearly) well-specified model. Nevertheless,

it suffers a performance decline in the real scenarios. Bandit algorithms such as ML and CEG correct model misspecification through exploration, leading to better adaptation and higher expected rewards in all simulations and real datasets. The proposed CS logistic bandit consistently outperforms state-of-the-art algorithms, both cost-insensitive and cost-sensitive, particularly excelling as the parameter a increases. This show that the exploration performed by CSLB combined with a CS approach can significantly improve the potential benefits. Thus, CSLB is considered an interesting choice for cost-sensitive dynamic problems.

The CS algorithms proposed in this chapter prove to be effective and stable in most scenarios, making them good approaches for cost-sensitive dynamic problems. By proposing one approach that considers exploration and another that does not, the full range of alternatives is covered from the perspective of reinforcing learning in cost-sensitive classification problems. Thus, solving the dynamic cost-sensitive learning problem.

Appendix

		Normal						Risky					
Model	Classifier	Sav	Acc	Rec	PPV	PP	FS	Sav	Acc	Rec	PPV	PP	FS
Logit	Static	82.81	90.80	94.55	54.03	18.45	68.46	68.42	93.02	94.33	34.51	10.29	50.41
	PA	39.26	55.65	77.83	16.42	50.26	27.11	-62.73	50.83	68.03	5.11	50.54	9.50
	CSOGD	87.74	87.67	99.03	49.18	22.71	65.02	61.12	86.71	97.19	25.19	16.87	39.44
	PAUM	61.37	60.25	97.98	20.78	49.91	34.28	-30.05	52.81	96.60	7.22	50.73	13.43
	PAPB	57.86	89.71	62.47	51.25	12.95	56.22	43.25	93.99	54.38	32.92	6.35	40.71
	IDCSPA	58.66	88.02	66.03	44.67	15.31	53.28	44.47	93.90	55.30	32.08	6.50	40.34
Squared	Static	3.67	74.67	30.81	16.85	21.74	19.80	-6.55	88.18	30.99	11.53	10.49	16.57
	PA	2.50	49.46	50.07	9.87	50.56	16.49	-82.80	49.99	50.64	3.72	50.06	6.92
	CSOGD	79.10	86.68	92.34	45.98	21.77	60.84	57.62	86.72	94.58	24.55	16.56	38.52
	PAUM	3.39	50.24	49.97	10.01	49.75	16.67	-87.50	49.66	47.42	3.48	50.15	6.49
	PAPB	54.88	89.43	60.74	47.72	12.72	53.36	42.72	94.05	54.17	32.13	6.28	40.11
	IDCSPA	59.19	88.19	66.64	43.94	15.13	52.96	48.35	94.03	61.25	34.03	6.83	43.74
Qubic	Static	73.08	81.65	88.66	43.94	29.69	58.63	62.73	87.95	94.42	29.84	16.70	45.24
	PA	51.55	58.17	76.87	22.70	49.72	35.04	-17.16	52.17	68.77	7.65	49.90	13.75
	CSOGD	90.72	88.40	98.23	58.47	25.75	72.72	73.22	87.05	96.85	33.37	18.15	48.88
	PAUM	73.16	64.44	96.97	28.84	49.34	44.45	11.86	54.91	94.53	10.47	50.01	18.83
	PAPB	39.66	81.21	47.21	38.57	17.98	42.41	33.73	90.45	43.78	27.65	8.90	33.65
	IDCSPA	48.76	79.39	57.79	37.31	22.91	45.34	33.50	90.78	41.42	28.29	8.31	33.31
Ker & Sam	Static	53.61	79.36	84.52	25.84	26.15	39.56	-3.21	86.32	57.76	10.41	14.03	17.60
	PA	8.10	87.22	16.33	24.87	7.38	16.01	-6.62	95.60	1.86		2.14	
	CSOGD	56.82	96.31	56.79	95.15	4.78	70.69	67.31	99.00	66.39	89.38	1.76	74.78
	PAUM	-0.17	91.62	0.33	1.38	0.42		-44.11	80.86	17.46	0.55	17.77	
	PAPB	20.71	83.48	35.95	20.18	14.27	25.75	-0.34	91.35	25.49	8.03	7.49	12.10
	IDCSPA	23.49	81.43	43.00	19.53	17.48	26.86	-0.32	92.49	19.91	7.92	6.07	11.24
Scobit	Static	85.76	92.67	92.91	65.38	18.39	76.66	68.31	92.71	96.28	32.73	10.24	47.81
	PA	50.80	58.22	80.90	20.95	49.74	33.28	-94.51	51.21	66.64	4.22	49.84	7.94
	CSOGD	90.00	88.15	99.04	54.71	24.49	69.86	52.86	86.73	97.88	21.94	16.30	35.39
	PAUM	70.26	62.64	98.96	25.50	49.97	40.54	-61.87	53.15	98.25	6.24	49.91	11.72
	PAPB	61.34	89.19	66.58	56.97	15.09	61.35	47.78	95.22	57.96	35.16	5.30	43.47
	IDCSPA	65.69	88.10	72.28	52.24	17.54	60.64	46.95	95.35	56.80	35.62	5.08	43.53
Klein & Spady	Static	-1.47	49.33	51.32	11.43	50.96	18.39	-47.29	60.62	43.17	5.90	38.71	10.21
	PA	11.67	50.45	49.93	11.15	49.53	18.22	-38.73	51.55	49.57	5.40	48.39	9.73
	CSOGD	73.50	85.32	85.60	44.93	22.57	58.46	58.25	85.56	85.28	27.84	18.16	41.51
	PAUM	15.99	49.82	55.43	11.94	51.36	19.62	-39.25	50.61	51.17	5.48	49.54	9.89
	PAPB	4.12	74.76	18.77	11.33	18.33	14.05	-7.81	83.66	13.14	5.63	12.45	7.77
	IDCSPA	11.40	66.92	27.23	12.70	27.28	17.31	-1.70	87.43	10.76	6.72	8.43	8.27

Table 6.6: Savings (Sav), accuracy (Acc), recall (Rec), positive predicted value (PPV), positive predicted proportion (PP) and F-score (FS) for the OL algorithms for the cost setting $a = 0.05$ in the simulated datasets.

		Normal						Risky					
Model	Classifier	Sav	Acc	Rec	PPV	PP	FS	Sav	Acc	Rec	PPV	PP	FS
Logit	Static	82.59	91.59	91.41	57.26	17.32	70.19	75.78	94.57	91.72	42.21	8.64	57.28
	PA	3.08	56.32	78.43	16.23	49.52	26.89	-201.20	50.76	67.35	4.50	50.39	8.43
	CSOGD	72.13	86.48	98.96	47.07	23.56	62.87	4.84	85.09	97.60	21.86	18.13	35.12
	PAUM	19.45	60.19	97.62	20.18	49.58	33.44	-204.16	52.49	94.33	6.30	50.50	11.80
	PAPB	50.13	90.00	62.59	51.00	12.62	56.11	27.37	94.31	53.34	30.82	5.92	38.78
	IDCSPA	53.35	89.67	65.80	50.27	13.66	56.92	29.90	94.77	51.45	33.49	5.32	40.43
Squared	Static	9.64	80.10	30.05	21.71	16.24	23.24	-28.40	81.77	34.35	9.56	17.11	14.26
	PA	-26.22	50.23	50.44	10.26	49.86	17.04	-174.26	49.69	50.14	3.89	50.33	7.22
	CSOGD	63.96	85.07	90.00	44.38	23.03	58.78	14.68	85.00	91.79	23.83	18.27	37.33
	PAUM	-36.17	50.28	51.15	10.38	49.96	17.25	-194.67	50.43	53.48	4.21	49.86	7.79
	PAPB	48.48	89.71	61.84	49.53	12.69	54.91	33.12	94.06	56.14	34.29	6.44	42.37
	IDCSPA	51.10	89.29	64.79	48.53	13.74	55.44	35.35	94.33	56.16	36.40	6.17	44.05
Qubic	Static	74.91	82.85	89.08	45.84	28.65	60.48	64.01	87.11	92.48	29.55	17.53	44.44
	PA	31.04	58.01	77.12	23.04	50.12	35.48	-69.26	52.35	69.67	8.10	49.93	14.51
	CSOGD	81.52	87.51	97.57	57.86	26.74	71.88	50.15	85.80	97.25	33.11	19.68	48.54
	PAUM	49.88	63.88	97.30	28.98	50.30	44.65	-49.41	55.95	97.51	11.42	49.57	20.44
	PAPB	31.16	80.73	46.54	38.27	18.23	41.95	24.60	90.44	46.12	29.63	9.10	35.88
	IDCSPA	33.62	79.15	52.24	36.38	21.53	42.88	25.15	90.40	46.24	29.15	9.17	35.69
Ker & Sam	Static	51.51	77.39	88.73	24.05	28.47	37.69	-15.34	87.35	46.71	11.17	12.45	17.15
	PA	1.07	87.45	15.03	16.93	7.12		-13.01	95.26	3.09		2.26	
	CSOGD	50.86	96.06	52.33	94.09	4.30	65.72	63.02	98.81	63.82	89.39	1.94	70.12
	PAUM	-0.16	92.16	0.04	1.43	0.07		-113.28	82.71	15.34	0.49	15.39	
	PAPB	10.02	84.20	36.93	20.93	13.75	26.60	-26.87	91.05	26.17	9.17	7.71	13.46
	IDCSPA	10.60	83.84	38.27	20.58	14.34	26.62	-20.54	92.91	17.92	8.33	5.47	11.36
Scobit	Static	86.04	92.75	92.50	66.02	17.95	76.65	78.40	96.53	92.60	47.58	6.11	62.42
	PA	22.12	58.09	81.15	20.61	49.79	32.87	-181.28	50.58	67.14	4.48	50.58	8.40
	CSOGD	78.17	87.04	99.08	52.82	25.38	68.08	12.56	85.01	97.54	21.76	18.20	35.01
	PAUM	37.06	62.64	98.66	25.12	49.68	40.03	-170.94	52.62	95.78	6.41	50.48	12.01
	PAPB	56.96	89.65	67.76	57.74	14.86	62.30	32.68	94.65	55.54	33.22	5.72	41.28
	IDCSPA	57.81	89.03	70.34	54.76	16.08	61.58	35.69	95.33	53.58	37.55	4.91	43.91
Klein & Spady	Static	-5.89	69.60	24.67	9.68	25.31	13.20	-60.74	57.44	44.28	5.20	42.07	9.03
	PA	-24.46	51.43	47.22	10.57	47.99	17.26	-106.75	50.51	46.95	4.96	49.17	8.97
	CSOGD	57.93	84.25	83.02	43.10	22.84	56.19	29.12	84.00	82.40	26.06	19.37	39.09
	PAUM	-32.25	48.83	52.28	10.85	51.66	17.97	-124.72	48.55	52.23	5.26	51.69	9.55
	PAPB	-8.89	75.48	19.80	11.76	18.06	14.67	-30.33	83.58	11.71	4.79	12.45	6.72
	IDCSPA	-8.33	75.20	20.68	12.11	18.47	15.11	-20.52	86.40	10.53	6.59	9.10	8.11

Table 6.7: Savings (Sav), accuracy (Acc), recall (Rec), positive predicted value (PPV), positive predicted proportion (PP) and F-score (FS) for the OL algorithms for the cost setting $a = 0.10$ in the simulated datasets.

		Normal						Risky					
Model	Classifier	Sav	Acc	Rec	PPV	PP	FS	Sav	Acc	Rec	PPV	PP	FS
Logit	Static	81.48	90.78	93.37	53.77	18.11	67.90	67.02	92.29	93.07	34.87	10.96	49.80
	PA	-57.09	56.50	79.02	16.42	49.47	27.17	-393.00	51.67	66.37	4.94	49.55	9.19
	CSOGD	42.10	85.18	98.78	45.39	24.85	61.11	-86.76	83.74	97.44	22.82	19.77	36.17
	PAUM	-55.21	60.19	98.96	20.40	49.88	33.80	-441.41	52.71	93.45	6.84	50.52	12.74
	PAPB	36.03	89.38	60.65	48.72	12.84	53.95	10.94	94.25	55.83	33.72	6.20	41.75
	IDCSPA	37.19	89.79	59.90	50.22	12.27	54.56	19.41	94.84	53.71	36.44	5.43	43.29
Squared	Static	13.99	82.04	31.72	20.07	14.64	24.46	-31.40	83.01	26.35	6.92	15.29	10.56
	PA	-99.85	50.63	51.48	10.32	49.70	17.18	-394.73	49.88	54.04	3.79	50.41	7.08
	CSOGD	26.00	83.51	88.48	42.14	24.14	56.28	-90.93	83.28	90.50	21.25	19.58	33.82
	PAUM	-130.53	50.23	49.39	9.90	49.67	16.48	-492.41	49.09	48.82	3.39	50.81	6.34
	PAPB	32.61	89.51	60.97	47.89	12.69	53.55	-3.81	93.66	50.24	28.20	6.35	35.87
	IDCSPA	32.75	89.80	57.44	48.89	11.71	52.75	0.48	94.14	45.36	30.78	5.49	36.59
Qubic	Static	75.33	81.90	90.53	44.38	30.00	59.52	65.21	86.91	95.12	30.93	18.45	46.51
	PA	-8.59	57.98	77.71	23.02	50.28	35.51	-191.01	53.20	73.42	8.76	49.58	15.63
	CSOGD	61.68	86.15	97.82	55.72	28.09	70.04	-9.77	84.36	97.53	32.37	21.27	47.51
	PAUM	1.97	63.85	97.73	28.88	50.35	44.58	-203.64	55.91	98.19	11.66	49.79	20.83
	PAPB	17.07	80.96	47.49	38.73	18.29	42.62	-5.76	89.68	42.65	26.88	9.46	32.79
	IDCSPA	18.21	80.70	47.14	39.03	18.45	42.62	-4.17	90.06	39.10	27.20	8.68	32.04
Ker & Sam	Static	50.78	80.68	80.24	26.43	24.01	39.50	-20.45	85.81	51.14	9.89	14.25	16.26
	PA	-13.29	87.32	13.63	20.25	7.08		-26.63	95.40	3.42		2.23	
	CSOGD	47.98	95.93	49.60	94.84	4.03	62.29	57.90	98.78	60.89		1.78	
	PAUM	-0.66	92.16	0.16	1.73	0.14		-305.93	78.35	20.00	0.44	19.97	
	PAPB	-23.87	83.88	35.44	19.80	13.86	25.28	-85.31	90.99	28.55	9.36	7.92	13.98
	IDCSPA	-20.31	85.08	32.40	20.46	12.21	25.02	-56.68	92.77	24.06	10.63	5.89	14.74
Scobit	Static	85.18	93.58	91.35	69.79	17.25	78.97	73.57	95.33	91.63	42.03	7.30	56.93
	PA	-24.53	57.82	80.60	20.32	49.89	32.44	-456.24	51.24	64.69	4.07	49.70	7.65
	CSOGD	54.19	85.60	99.06	50.71	26.74	66.07	-110.00	83.80	97.40	20.24	19.16	32.76
	PAUM	-21.29	61.69	98.63	24.55	50.55	39.31	-511.68	52.55	94.87	5.92	50.25	11.12
	PAPB	47.11	89.44	66.34	56.97	14.67	61.25	6.45	94.91	53.84	31.71	5.36	39.59
	IDCSPA	53.41	90.50	67.11	60.87	13.77	63.80	16.23	95.60	47.87	35.16	4.27	40.54
Klein & Spady	Static	2.59	61.62	38.57	12.17	36.05	17.56	-59.06	54.38	51.33	5.83	45.78	10.34
	PA	-90.00	50.44	49.32	10.83	49.41	17.75	-228.43	51.99	47.45	5.31	47.73	9.55
	CSOGD	23.84	82.24	82.40	40.95	24.79	54.01	-36.87	82.67	81.83	25.71	20.74	38.54
	PAUM	-116.74	48.39	52.85	10.97	52.21	18.15	-301.98	48.86	54.23	5.63	51.60	10.19
	PAPB	-40.58	75.11	18.22	10.89	18.00	13.56	-77.89	83.11	11.24	4.66	12.74	6.51
	IDCSPA	-39.83	75.05	16.28	10.92	17.15	13.00	-54.13	86.63	8.13	4.87	8.90	6.09

Table 6.8: Savings (Sav), accuracy (Acc), recall (Rec), positive predicted value (PPV), positive predicted proportion (PP) and F-score (FS) for the OL algorithms for the cost setting $a = 0.20$ in the simulated datasets.

		Normal						Risky					
Model	Classifier	AER	Acc	Rec	PPV	PP	FS	AER	Acc	Rec	PPV	PP	FS
Logit	Static	57.82	71.23	99.02	25.97	38.73	41.15	73.05	79.35	98.36	14.85	24.18	25.80
	LB	58.94	73.78	97.93	27.67	35.96	43.15	77.74	85.36	95.89	19.46	17.99	32.35
	LinUCB	-63.82	89.84	0.00		0.00		-0.61	5.86	94.52	3.54	97.39	6.83
	CEG	58.85	73.34	98.43	27.40	36.50	42.86	77.87	85.45	96.44	19.62	17.94	32.61
	CSLB	60.80	87.86	89.57	45.09	20.18	59.99	80.37	95.37	81.37	42.92	6.92	56.20
Squared	Static	-12.54	12.73	88.19	8.99	94.64	16.32	-7.95	4.56	81.90	2.92	97.66	5.64
	LB	-19.22	43.18	61.45	10.05	59.03	17.27	30.38	63.76	37.36	3.68	35.36	6.69
	LinUCB	-54.88	90.35	0.00		0.00		-2.63	5.57	92.53	3.31	97.39	6.38
	CEG	-16.15	42.60	63.63	10.23	60.03	17.62	33.55	64.38	39.08	3.90	34.86	7.09
	CSLB	5.80	19.95	96.99	10.50	89.12	18.95	47.66	92.29	16.95	10.91	5.41	13.27
Qubic	Static	48.86	66.30	99.25	30.34	48.22	46.47	71.86	80.85	98.27	22.94	24.72	37.19
	LB	48.97	68.49	98.37	31.68	45.77	47.93	69.37	78.49	97.05	20.79	26.94	34.24
	LinUCB	-129.75	85.26	0.00		0.00		-1.39	7.73	95.84	5.67	97.56	10.70
	CEG	49.80	68.77	98.58	31.90	45.55	48.20	68.07	78.46	96.01	20.63	26.85	33.97
	CSLB	50.41	75.72	95.32	37.33	37.64	53.65	73.00	89.43	86.66	33.78	14.80	48.61
Ker & Sam	Static	17.67	48.74	98.29	33.24	76.12	49.68	-0.57	11.56	75.48	2.19	89.77	4.26
	LB	13.56	51.46	96.62	34.28	72.54	50.61	28.90	44.08	66.67	3.06	56.79	5.86
	LinUCB	-305.73	74.26	0.00		0.00		1.03	4.73	96.93	2.59	97.72	5.04
	CEG	12.51	51.15	96.23	34.09	72.65	50.35	29.93	44.15	72.03	3.30	57.00	6.31
	CSLB	6.80	50.74	95.07	33.77	72.46	49.84	57.19	96.35	2.68	5.93	1.18	3.69
Scobit	Static	60.08	76.37	99.06	35.06	36.20	51.79	78.65	84.45	98.73	16.66	18.61	28.51
	LB	61.23	78.92	98.13	37.62	33.41	54.39	81.72	89.56	94.90	22.47	13.26	36.34
	LinUCB	-103.19	87.19	0.00		0.00		-0.36	5.39	94.27	3.04	97.39	5.89
	CEG	60.20	78.56	98.28	37.24	33.81	54.01	82.69	89.67	96.50	22.87	13.25	36.97
	CSLB	64.68	89.47	94.46	55.20	21.92	69.68	85.32	94.77	92.36	36.76	7.89	52.58
Klein & Spady	Static	-3.56	22.14	87.64	10.22	85.35	18.30	10.77	21.53	88.47	4.60	81.74	8.75
	LB	-13.83	41.93	67.34	10.89	61.52	18.75	16.28	40.33	68.00	4.72	61.20	8.83
	LinUCB	-58.43	90.05	0.00		0.00		-0.15	6.71	97.18	4.24	97.30	8.13
	CEG	-13.81	41.82	66.43	10.76	61.45	18.52	17.22	40.55	69.65	4.84	61.12	9.06
	CSLB	-1.52	12.07	97.49	9.96	97.38	18.08	31.97	90.66	6.12	4.63	5.61	5.27

Table 6.9: Average expected reward (AER), accuracy (Acc), recall (Rec), positive predicted value (PPV), positive predicted proportion (PP) and F-score (FS) for the bandit algorithms for the cost setting $a = 0.05$ in the simulated datasets.

		Normal						Risky					
Model	Classifier	AER	Acc	Rec	PPV	PP	FS	AER	Acc	Rec	PPV	PP	FS
Logit	Static	117.55	71.23	99.02	25.97	38.73	41.15	147.27	79.35	98.36	14.85	24.18	25.80
	LB	121.22	73.78	97.93	27.67	35.96	43.15	157.93	85.36	95.89	19.46	17.99	32.35
	LinUCB	-1.18	12.06	96.46	10.06	97.38	18.23	1.75	5.86	94.52	3.54	97.39	6.83
	CEG	120.78	73.34	98.43	27.40	36.50	42.86	157.91	85.45	96.44	19.62	17.94	32.61
	CSLB	124.96	92.01	74.41	58.38	12.95	65.43	170.03	95.80	80.82	45.74	6.45	58.42
Squared	Static	-16.24	10.76	86.22	8.65	96.23	15.72	-6.34	4.56	81.90	2.92	97.66	5.64
	LB	17.43	43.18	61.45	10.05	59.03	17.27	92.07	63.76	37.36	3.68	35.36	6.69
	LinUCB	-5.23	11.24	94.72	9.38	97.39	17.08	-0.40	5.57	92.53	3.31	97.39	6.38
	CEG	19.50	42.58	63.52	10.21	60.03	17.59	95.94	64.38	39.08	3.90	34.86	7.09
	CSLB	39.58	78.20	22.80	13.29	16.55	16.79	139.42	95.85	3.16	12.36	0.89	5.03
Qubic	Static	99.08	66.30	99.25	30.34	48.22	46.47	145.41	80.85	98.27	22.94	24.72	37.19
	LB	101.54	68.49	98.37	31.68	45.77	47.93	140.71	78.49	97.05	20.79	26.94	34.24
	LinUCB	-3.19	16.27	96.95	14.65	97.57	25.45	0.77	7.73	95.84	5.67	97.56	10.70
	CEG	102.64	68.77	98.58	31.90	45.55	48.20	139.33	78.46	96.01	20.63	26.85	33.97
	CSLB	112.41	79.71	92.40	41.54	32.79	57.31	153.93	90.85	77.99	36.35	12.38	49.59
Ker & Sam	Static	41.18	48.74	98.29	33.24	76.12	49.68	8.85	11.56	75.48	2.19	89.77	4.26
	LB	39.96	51.46	96.62	34.28	72.54	50.61	70.39	44.08	66.67	3.06	56.79	5.86
	LinUCB	-2.57	27.07	98.17	25.86	97.73	40.93	3.14	4.73	96.93	2.59	97.72	5.04
	CEG	38.84	51.15	96.23	34.09	72.65	50.35	71.46	44.15	72.03	3.30	57.00	6.31
	CSLB	37.19	60.03	90.75	38.33	60.95	53.89	152.71	96.93	1.92	8.93	0.56	3.15
Scobit	Static	122.28	76.37	99.06	35.06	36.20	51.79	158.33	84.45	98.73	16.66	18.61	28.51
	LB	126.03	78.92	98.13	37.62	33.41	54.39	166.88	89.56	94.90	22.47	13.26	36.34
	LinUCB	-2.86	14.50	96.33	12.67	97.37	22.40	2.01	5.39	94.27	3.04	97.39	5.89
	CEG	124.60	78.56	98.28	37.24	33.81	54.01	167.78	89.67	96.50	22.87	13.25	36.97
	CSLB	131.42	91.18	86.34	61.00	18.13	71.49	176.07	95.32	90.76	39.36	7.24	54.91
Klein & Spady	Static	9.40	22.14	87.64	10.22	85.35	18.30	27.69	21.53	88.47	4.60	81.74	8.75
	LB	20.23	41.93	67.34	10.89	61.52	18.75	52.37	40.33	68.00	4.72	61.20	8.83
	LinUCB	-0.78	12.11	97.19	9.94	97.28	18.04	2.27	6.71	97.18	4.24	97.30	8.13
	CEG	20.34	41.82	66.43	10.76	61.45	18.52	53.44	40.55	69.65	4.84	61.12	9.06
	CSLB	30.48	85.61	6.03	10.64	5.64	7.70	126.11	95.05	0.47	2.70	0.74	0.80

Table 6.10: Average expected reward (AER), accuracy (Acc), recall (Rec), positive predicted value (PPV), positive predicted proportion (PP) and F-score (FS) for the bandit algorithms for the cost setting $a = 0.10$ in the simulated datasets.

		Normal						Risky					
Model	Classifier	AER	Acc	Rec	PPV	PP	FS	AER	Acc	Rec	PPV	PP	FS
Logit	Static	296.74	71.23	99.02	25.97	38.73	41.15	369.93	79.35	98.36	14.85	24.18	25.80
	LB	308.08	73.78	97.93	27.67	35.96	43.15	398.50	85.36	95.89	19.46	17.99	32.35
	LinUCB	5.41	12.06	96.46	10.06	97.38	18.23	8.84	5.86	94.52	3.54	97.39	6.83
	CEG	306.55	73.34	98.43	27.40	36.50	42.86	398.03	85.45	96.44	19.62	17.94	32.61
	CSLB	376.03	93.20	72.15	64.87	11.30	68.31	444.48	96.77	68.77	54.57	4.60	60.85
Squared	Static	-9.34	10.76	86.22	8.65	96.23	15.72	-1.50	4.56	81.90	2.92	97.66	5.64
	LB	127.38	43.18	61.45	10.05	59.03	17.27	277.15	63.76	37.36	3.68	35.36	6.69
	LinUCB	0.63	11.24	94.72	9.38	97.39	17.08	6.32	5.57	92.53	3.31	97.39	6.38
	CEG	126.64	42.58	63.52	10.21	60.03	17.59	283.09	64.38	39.08	3.90	34.86	7.09
	CSLB	296.66	89.18	1.76	11.26	1.51	3.05	423.09	96.16	2.01	14.00	0.50	3.52
Qubic	Static	249.77	66.30	99.25	30.34	48.22	46.47	366.08	80.85	98.27	22.94	24.72	37.19
	LB	259.24	68.49	98.37	31.68	45.77	47.93	354.71	78.49	97.05	20.79	26.94	34.24
	LinUCB	2.52	16.27	96.95	14.65	97.57	25.45	7.26	7.73	95.84	5.67	97.56	10.70
	CEG	261.16	68.77	98.58	31.90	45.55	48.20	353.11	78.46	96.01	20.63	26.85	33.97
	CSLB	315.19	83.03	86.97	46.00	27.87	60.17	415.04	93.49	61.70	45.29	7.86	52.24
Ker & Sam	Static	221.66	54.23	96.67	15.18	53.63	26.24	37.11	11.56	75.48	2.19	89.77	4.26
	LB	245.58	59.03	92.99	16.24	48.21	27.65	194.85	44.08	66.67	3.06	56.79	5.86
	LinUCB	7.66	10.33	97.86	8.43	97.73	15.53	9.49	4.73	96.93	2.59	97.72	5.04
	CEG	244.16	59.27	93.23	16.35	48.01	27.82	196.06	44.15	72.03	3.30	57.00	6.31
	CSLB	323.20	91.37	0.48	13.79	0.29	0.92	439.25	97.10	0.77	6.06	0.33	1.36
Scobit	Static	308.88	76.37	99.06	35.06	36.20	51.79	397.39	84.45	98.73	16.66	18.61	28.51
	LB	320.44	78.92	98.13	37.62	33.41	54.39	422.34	89.56	94.90	22.47	13.26	36.34
	LinUCB	3.45	14.50	96.33	12.67	97.37	22.40	9.15	5.39	94.27	3.04	97.39	5.89
	CEG	317.79	78.56	98.28	37.24	33.81	54.01	423.06	89.67	96.50	22.87	13.25	36.97
	CSLB	346.91	93.34	62.14	81.47	9.77	70.50	450.84	97.22	66.56	54.71	3.82	60.06
Klein & Spady	Static	48.31	22.14	87.64	10.22	85.35	18.30	78.45	21.53	88.47	4.60	81.74	8.75
	LB	122.39	41.93	67.34	10.89	61.52	18.75	160.65	40.33	68.00	4.72	61.20	8.83
	LinUCB	5.90	12.11	97.19	9.94	97.28	18.04	9.54	6.71	97.18	4.24	97.30	8.13
	CEG	122.80	41.82	66.43	10.76	61.45	18.52	162.09	40.55	69.65	4.84	61.12	9.06
	CSLB	293.99	89.88	0.30	13.04	0.23	0.59	407.19	95.52	0.24	4.00	0.25	0.44

Table 6.11: Average expected reward (AER), accuracy (Acc), recall (Rec), positive predicted value (PPV), positive predicted proportion (PP) and F-score (FS) for the bandit algorithms for the cost setting $a = 0.20$ in the simulated datasets.

Data set	Classifier	AER	Acc	Rec	PPV	PP	FS
Credit card	Static	0.18	10.77	97.42	3.37	92.25	6.51
	LB	0.80	52.82	89.25	5.72	49.68	10.76
	LinUCB	-0.33	5.15	64.73	2.15	95.79	4.17
	CEG	0.79	53.23	88.82	5.75	49.24	10.80
	CSLB	0.83	52.51	86.67	5.54	49.83	10.42
GCD	Static	-14.34	45.20	85.67	33.73	76.20	48.40
	LB	-54.57	55.50	64.33	36.35	53.10	46.45
	LinUCB	-46.57	43.90	67.67	30.43	66.70	41.99
	CEG	-43.84	59.00	70.00	39.62	53.00	50.60
	CSLB	-14.50	41.40	92.67	33.02	84.20	48.69
PAKDD	Static	-3.87	32.22	91.81	27.60	90.54	42.44
	LB	-3.12	30.09	93.97	27.26	93.84	42.26
	LinUCB	-15.85	41.39	66.95	26.86	67.83	38.34
	CEG	-3.46	29.66	93.50	27.07	94.03	41.98
	CSLB	-0.18	27.46	99.54	27.23	99.50	42.76
ASF	Static	9.87	15.37	90.21	3.17	87.09	6.12
	LB	41.24	50.80	77.34	4.65	50.88	8.77
	LinUCB	4.23	9.90	88.39	2.92	92.45	5.66
	CEG	41.56	51.16	77.90	4.71	50.55	8.89
	CSLB	65.03	96.91	0.00	0.00	0.03	
Drift	Static	10.57	16.57	94.08	3.94	86.63	7.57
	LB	41.46	41.78	91.60	5.43	61.24	10.25
	LinUCB	-4.87	12.87	71.90	2.94	88.72	5.65
	CEG	41.50	41.51	91.51	5.40	61.50	10.20
	CSLB	54.51	96.21	0.09	1.96	0.17	0.18

Table 6.12: Average expected reward (AER), accuracy (Acc), recall (Rec), positive predicted value (PPV), positive predicted proportion (PP) and F-score (FS) for the bandit algorithms for the cost setting $a = 0.05$ in the real datasets.

Data set	Classifier	AER	Acc	Rec	PPV	PP	FS
	Static	0.07	4.57	98.28	3.18	98.51	6.16
	LB	3.91	59.72	86.02	6.44	42.57	11.98
Credit card	LinUCB	0.05	5.23	62.37	2.08	95.56	4.03
	CEG	3.99	59.39	86.88	6.45	42.96	12.00
	CSLB	4.36	51.41	91.40	5.69	51.23	10.71
	Static	-26.01	42.10	78.00	31.33	74.70	44.70
	LB	-4.82	57.30	79.33	39.47	60.30	52.71
GCD	LinUCB	-49.88	42.00	62.67	28.66	65.60	39.33
	CEG	-11.96	57.90	76.00	39.51	57.70	52.00
	CSLB	5.97	41.90	96.33	33.64	85.90	49.87
	Static	-2.79	32.22	91.81	27.60	90.54	42.44
	LB	-2.47	30.09	93.97	27.26	93.84	42.26
PAKDD	LinUCB	-12.44	41.39	66.95	26.86	67.83	38.34
	CEG	-2.83	29.66	93.50	27.07	94.03	41.98
	CSLB	-1.74	30.44	94.43	27.42	93.74	42.50
	Static	26.26	15.37	90.21	3.17	87.09	6.12
	LB	94.80	50.80	77.34	4.65	50.88	8.77
ASF	LinUCB	15.82	9.90	88.39	2.92	92.45	5.66
	CEG	95.37	51.16	77.90	4.71	50.55	8.89
	CSLB	191.73	96.91	0.00	0.00	0.03	
	Static	25.87	16.57	94.08	3.94	86.63	7.57
	LB	88.59	41.78	91.60	5.43	61.24	10.25
Drift	LinUCB	10.79	12.87	71.90	2.94	88.72	5.65
	CEG	88.27	41.51	91.51	5.40	61.50	10.20
	CSLB	178.46	95.24	0.55	1.71	1.17	0.83

Table 6.13: Average expected reward (AER), accuracy (Acc), recall (Rec), positive predicted value (PPV), positive predicted proportion (PP) and F-score (FS) for the bandit algorithms for the cost setting $a = 0.10$ in the real datasets.

Data set	Classifier	AER	Acc	Rec	PPV	PP	FS
Credit card	Static	0.13	4.57	98.28	3.18	98.51	6.16
	LB	8.64	59.72	86.02	6.44	42.57	11.98
	LinUCB	1.07	5.23	62.37	2.08	95.56	4.03
	CEG	8.69	59.39	86.88	6.45	42.96	12.00
	CSLB	9.71	53.41	89.25	5.79	49.09	10.88
GCD	Static	-4.39	42.10	78.00	31.33	74.70	44.70
	LB	28.48	57.30	79.33	39.47	60.30	52.71
	LinUCB	-20.90	42.00	62.67	28.66	65.60	39.33
	CEG	23.43	57.90	76.00	39.51	57.70	52.00
	CSLB	34.59	57.80	82.00	40.07	61.40	53.83
PAKDD	Static	-0.64	32.22	91.81	27.60	90.54	42.44
	LB	-1.16	30.09	93.97	27.26	93.84	42.26
	LinUCB	-5.63	41.39	66.95	26.86	67.83	38.34
	CEG	-1.58	29.66	93.50	27.07	94.03	41.98
	CSLB	-0.13	28.24	98.28	27.28	98.05	42.71
ASF	Static	59.04	15.37	90.21	3.17	87.09	6.12
	LB	201.92	50.80	77.34	4.65	50.88	8.77
	LinUCB	39.01	9.90	88.39	2.92	92.45	5.66
	CEG	202.99	51.16	77.90	4.71	50.55	8.89
	CSLB	445.12	96.91	0.00	0.00	0.03	
Drift	Static	56.48	16.57	94.08	3.94	86.63	7.57
	LB	182.85	41.78	91.60	5.43	61.24	10.25
	LinUCB	42.10	12.87	71.90	2.94	88.72	5.65
	CEG	181.82	41.51	91.51	5.40	61.50	10.20
	CSLB	428.17	95.24	0.55	1.71	1.17	0.83

Table 6.14: Average expected reward (AER), accuracy (Acc), recall (Rec), positive predicted value (PPV), positive predicted proportion (PP) and F-score (FS) for the bandit algorithms for the cost setting $a = 0.20$ in the real datasets.

Chapter 7

Conclusions and future research lines

The credit risk problem presented by our collaborating company, ASF, was defined by a number of very specific characteristics, which meant that there were no alternatives in the state-of-the-art for solving it. The main characteristic of the problem is its cost-sensitive nature, since misclassification errors have different effects depending on the type of error and on the characteristics of each transaction. In addition, no work has addressed the problem of credit risk as a dynamic problem, which may be of interest, as has been commented on throughout the report.

There are two approaches to address cost-sensitive classification problems as stated in Vanderschueren et al. (2022b): thresholding and predict-and-optimize. This work introduces new cost-sensitive methods from both perspectives to maximize benefits and reduce aggregate losses, the main concern in any business. The proposed techniques are motivated by the credit risk problem, but all proposals can be generalized to any cost-sensitive problem.

With the proposed Algorithm 2-DDR(k), extended in Algorithm 2-$DDR_q(k)$, the optimal decision rule is obtained given a score and an error cost specification. Previous thresholding approaches are included in the algorithm search, for which

a consistent improvement is obtained without the need for further estimation or parameter tuning. The algorithm also allows estimation of the optimal decision rule in multi-class response problems given a single score, a previously unexplored challenge. Thus, Algorithm 2-$DDR_q(k)$ provides a flexible and satisfactory solution to CS classification problems. It also has the advantage that any constraint can be considered. Therefore, Algorithm 2-$DDR_q(k)$ can be used to solve any CS classification problem given an estimated score.

Given a score, the CS classification problem is solved with Algorithm 2-$DDR_q(k)$ from the thresholding perspective. This leaves only the estimation of the score. For this, the cost-sensitive logistic model introduced in Höppner et al. (2021) is considered first. For a generic parametric model, several estimation approaches are proposed. The consistency and asymptotic normality of the estimators are proved. Furthermore, a bound on the introduced bias is obtained when considering the regularized version of the problem, which is often needed in practice. In addition, an extensive simulation study and the analysis of two real data sets are performed for the first time to assess the satisfactory performance of the model. Thus, a theoretical and practical foundation for cost-sensitive parametric learning is provided.

Given the satisfactory performance of the predict-and-optimize approach considering a parametric learner, the CS approach is extended to the semi-parametric setting considering the single-index model (SIM) introduced in Klein and Spady (1993). The SIM offers a compelling alternative to parametric models and other classification techniques due to its flexibility, high-dimensional data handling capabilities, interpretability, and ability to model complex relationships. Combining the flexibility of SIMs with a cost-sensitive approach yields a superior model for cost-sensitive problems. This is demonstrated through a simulation study and the analysis of three real data sets. The cost-sensitive SIM (CSSIM) not only achieves superior results, but also provides valuable insights into the underlying data relationships, as the model provides us with a linear combination of the covariates, something that is sought in, for example, financial settings. Consequently, the proposed model is considered as a promising classifier for CS classification problems,

both in terms of interpretability and performance.

With the proposals mentioned above, the CS classification problem is solved from both perspectives (thresholding and predict-and-optimize) and considering both parametric and semi-parametric models. The only thing that remains at this point is to solve the problem from a dynamic approach, as should be done by the definition of the problem. This work presents two novel algorithms designed to tackle cost-sensitive dynamic classification problems. The PA algorithm, as introduced in Crammer et al. (2006) is extended to the CS setting as described in Algorithm IDC-SPA. In addition, the logistic bandit approach is extended with Algorithm CSLB for efficient exploration. The performance of the proposed models is empirically tested against the state-of-the-art alternatives, considering a wide range of simulation settings and various real-world data sets. Promising experimental results demonstrate the effectiveness and efficiency of the proposed models compared to state-of-the-art algorithms. Furthermore, this study provides valuable insights into the algorithmic performance in terms of case proportions, dependency relationships, and cost specifications. Finally, it is noted that by proposing one approach that considers exploration and another that does not, the full range of alternatives is covered from a reinforcement learning perspective.

Interesting challenges and open problems remain to be addressed in the future:

Further extensions of the proposed Algorithm 2-$DDR_q(k)$ can be explored. For example, a stricter loss function to select the top 10% more profitable operations could be considered to select a portion of the portfolio to offer better terms, and then a different loss function for the remaining loans. Similarly, the algorithm can be extended to consider more dimensions in the decision rule search by introducing more than one score or more exogenous variables. The different extensions that can be considered are innumerable, thanks to the flexibility of Algorithm 2-$DDR_q(k)$.

Another problem that remains open is to obtain the theoretical properties (consistency and asymptotic normality) of the proposed CSSIM, which would provide greater justification for the application of this model in practice.

The computational time of the CSSIM can be a drawback in some settings. Another interesting line of research would be to try to reduce the computational time for model estimation.

Another drawback of the CSSIM is the introduction of categorical variables into the model, which results in a loss of continuity in the score, making it difficult to estimate the link function. There are alternatives in the state-of-the-art to introduce categorical variables in the SIM, so it is expected that a solution can be found to introduce categorical variables in the CSSIM. This would make the CSSIM an even more flexible and interesting model in areas where there are categorical variables, such as credit risk itself.

Other reinforcement learning techniques could be extended to the CS setting. In this report, the PA algorithm and the logistic bandit algorithm are extended, but currently there are many unexplored models with potential in CS settings. The proposed algorithms could even be extended by introducing weights in the objective function to give more weight to the most recent observations, for example. Similarly, the theoretical properties, such as regret bounds or the expected reward, of the proposed CS algorithms could be investigated. This would give more weight to the empirical results obtained and further justify their use.

By approaching the problem of credit risk from the perspective of reinforcement learning, the door is open to even extend static algorithms to this methodology. For example, consider thresholding algorithms that are updated as more information becomes available to adapt not only models but also decision making strategies as more information becomes available.

Finally, the development of a R package to make the developed software more accessible to researchers and a wider user base would be an interesting prospect, given the good results obtained with the programmed functions for the proposals in this work.

Appendix A

Resumen en castellano

Esta tesis se ha realizado en colaboración con *Abanca Servicios Financieros* (ASF), una entidad española de financiación al consumo, como parte de una tesis industrial. En una tesis industrial el objetivo es investigar y aportar soluciones innovadoras ajustadas a las necesidades de la empresa. En particular, soluciones que no se encuentran todavía en el estado del arte. De todas formas, a pesar de que las metodologías desarrolladas están motivadas por problemas propuestos por ASF, todas las propuestas en este trabajo pueden generalizarse a cualquier otro problema.

A medida que la economía global evoluciona, las empresas financieras han tenido que ampliar su catálogo de productos para satisfacer las nuevas demandas de clientes y puntos de venta. El aumento en el volumen de solicitudes, los nuevos riesgos que emergen a medida que evolucionan los mercados y la necesidad de tiempos de respuesta más rápidos han empujado a las entidades financieras a depender más de herramientas automatizadas para la toma de decisiones. Esta tesis industrial profundiza en este ámbito de la gestión del riesgo de crédito, enfocándose en estrategias prácticas, metodologías y avances destinados a maximizar las ganancias y al mismo tiempo fortalecer la gestión del riesgo, asegurando así la estabilidad y el crecimiento de las entidades.

El crédito puede definirse como un acuerdo financiero de pago diferido, extendido

por un acreedor, también conocido como prestamista, a un deudor, también conocido como prestatario, basado en un fideicomiso. El acreedor extiende dinero o recursos al deudor bajo el acuerdo de que el deudor pagará el equivalente en una fecha posterior, creando una deuda que el prestatario está obligado a cumplir. Los recursos proporcionados pueden ser financieros (por ejemplo, la concesión de un préstamo) o pueden consistir en bienes o servicios (por ejemplo, crédito al consumo). A cambio de conceder el préstamo y asumir el riesgo de que la deuda no sea devuelta, el prestamista recibe una comisión sobre el capital prestado. Esta es la recompensa del prestamista por asumir el riesgo crediticio.

Entidades financieras de consumo como ASF otorgan crédito a personas físicas por medio de un punto de venta para la compra de bienes como un automóvil, un tratamiento o un electrodoméstico según se indica en Committee (2001). De esta forma, los establecimientos pueden desarrollar su actividad sin tener que financiar a sus clientes, con todo lo que ello implica tanto a nivel de capital como de regulación.

El riesgo de crédito mide la probabilidad de pérdida como resultado de que un prestatario no pague un préstamo de acuerdo con los términos del contrato. Esto implica que un prestamista puede no recibir el principal y los intereses adeudados, lo que resulta en una interrupción en los flujos de caja, asignación de provisiones adicionales y mayores costos de recuperación de capital después de que se produce el incumplimiento. Esto puede afectar directamente la rentabilidad y la estabilidad de la empresa, así como su reputación desde la perspectiva tanto de los inversores como del regulador.

Aunque es imposible saber exactamente quién incumplirá sus obligaciones, evaluar y gestionar adecuadamente el riesgo de crédito puede reducir la probabilidad y la gravedad de una pérdida. Los bancos mitigan este riesgo exigiendo garantías o imponiendo cláusulas adicionales basadas en el perfil del cliente, como mayores tasas de interés o límites de crédito. El objetivo es evaluar la solvencia y capacidad de pago del solicitante para determinar su capacidad de cumplir con sus obligaciones en virtud de un contrato de préstamo y conceder crédito o no según esta evaluación.

La financiación al consumo implica la gestión de dos factores de riesgo: la legitimidad de la solicitud de préstamo (riesgo de fraude) y la probabilidad de que un cliente incumpla su compromiso de pago (riesgo de impago). Ambos componentes pueden suponer que las entidades financieras sufran pérdidas, afectando su rentabilidad. La mora se produce cuando un cliente no cumple con sus obligaciones de pago contractuales durante un período de tiempo predeterminado. Normalmente, este valor se define como 90 días de retraso en el pago, como se especifica en Siddiqi (2006). Por otro lado, el fraude es un riesgo más grave, definido como una solicitud de préstamo sin ninguna intención de pago, lo que conduce a la pérdida total del crédito financiado.

El riesgo de fraude y el riesgo de impago son dos amenazas diferentes. El fraude viene provocado por clientes que falsifican información, suplantan identidades, etc. Por su parte, el riesgo de mora viene provocado por un mal perfil crediticio (o un empeoramiento de este) de un cliente que pretende pagar pero no tiene la solvencia suficiente para ello. Por lo tanto, las instituciones financieras normalmente gestionan el riesgo crediticio a través de un proceso de evaluación de dos etapas. En la primera, se estudia la legitimidad de la solicitud, comprobando la documentación aportada y consultando posibles alertas en otras entidades. Para evaluar el riesgo de insolvencia del prestatario, se tienen en cuentas sus variables socioeconómicas para diferenciar entre clientes solventes e insolventes. Este enfoque de dos etapas gestiona eficazmente el riesgo crediticio al priorizar la prevención del fraude y garantizar la solvencia del cliente para otorgar préstamos responsables y minimizar las pérdidas.

Estadísticamente, el riesgo de crédito se ocupa del desarrollo de modelos predictivos para respaldar la toma de decisiones, como se define en Crook et al. (2007); Lessmann et al. (2015) y Petrides et al. (2022). Por lo tanto, el riesgo de crédito es una tarea de aprendizaje supervisada diseñada para diferenciar entre solicitantes de préstamos buenos y malos en función de sus características conocidas, según se indica en Thomas et al. (2002) y Verbraken et al. (2014). Una etiqueta “mala” indica un alto riesgo de incumplimiento, lo que significa que es posible que el prestatario no pague el préstamo en su totalidad. El estándar de la industria para los bancos hoy

en día se considera el modelo de regresión logística, aunque varios modelos más complejos lo superan, como se indica en Lessmann et al. (2015) y Visantavarakul (2022). Esto se debe principalmente a la interpretabilidad del modelo logístico respecto a estos modelos más complejos, algo que se busca en las empresas financieras y que exige el regulador. El modelo logístico para el scoring crediticio ha sido estudiado en Thomas (2000); Siddiqi (2006) y Samreen et al. (2013), entre otros.

El objetivo principal al desarrollar modelos para la gestión del riesgo de crédito suele ser clasificar con precisión el riesgo de los solicitantes de préstamos, lo que permite establecer un nivel mínimo de solvencia crediticia a partir del cual los solicitantes son aceptados y por debajo del cual son rechazados. Más allá de simplemente otorgar o denegar préstamos, estos modelos también pueden definir estrategias de fijación de precios sensibles al riesgo, adaptando los términos del préstamo (tasas de interés y tarifas) según el perfil de riesgo del cliente. Esto permite tomar decisiones crediticias más ajustadas y rentables. Además, para las instituciones financieras que cumplen con Basilea, la estimación de la probabilidad de mora (PD, por sus siglas en inglés) desempeñan un papel vital para el cumplimiento normativo de las disposiciones de capital, como se describe en Bequé et al. (2017).

En la literatura se han propuesto y aplicado una amplia variedad de métodos de clasificación para la calificación crediticia. Lessmann et al. (2015) y Vanderschueren et al. (2022b) han realizado recientemente un completo estudio de evaluación como actualización de un estudio anterior realizado en Baesens et al. (2003). Lessmann et al. (2015) resaltan la escasez de datos financieros reales disponibles para la investigación académica en riesgo de crédito, y que los conjuntos de datos existentes son relativamente pequeños, con un promedio de sólo 6.167 observaciones. Esto limita el alcance y la generalización de los resultados de investigación. Además, los conjuntos de datos disponibles públicamente a menudo carecen de detalles sobre los costes y beneficios asociados con los préstamos, una información esencial para evaluar los modelos de clasificación basados en costes, algo clave en esta área. Estas limitaciones, principalmente debidas a las inquietudes de las instituciones financieras sobre la privacidad de los datos y su ventaja competitiva, hacen que los estudios de referencia

a gran escala con diversos conjuntos de datos sean casi imposibles, como se señala en Petrides et al. (2022). En consecuencia, la literatura, aunque extensa, carece de referentes debido a la ausencia de conjuntos de datos públicos y las consiguientes comparaciones.

Otra laguna en la literatura es que los modelos de clasificación en riesgo de crédito, comúnmente se entrenan en términos de medidas estadísticas que no tienen en cuenta el objetivo principal real, que es maximizar los beneficios y minimizar las pérdidas. Por ejemplo, sería mejor detectar un fraude de 10.000€ que cinco fraudes de 1.000€, aunque la precisión sea menor. Varios autores han señalado que la toma de decisiones basada únicamente en una probabilidad estimada tiene un peor desempeño en problemas donde no todos los tipos de error tienen el mismo impacto (Lucas and Jurgovsky (2020); Höppner et al. (2021); Bahnsen et al. (2014a); Elkan (2001); Vanderschueren et al. (2022b); Bahnsen et al. (2013); Wang et al. (2012)). En consecuencia, el riesgo de crédito debe abordarse como un problema sensible al coste, tal como se recalca en Höppner et al. (2021). Esto crea un grado adicional de complejidad, ya que los costes dependen no sólo de la clase de la observación si no también de sus características (el importe del préstamo).

Los métodos sensibles al coste (cost-sensitive) abordan problemas en los que diferentes errores de clasificación implican diferentes impactos. De esta forma, los modelos se ajustan teniendo en cuenta diferentes costes para los falsos positivos (identificar incorrectamente un evento) y los falsos negativos (perder un evento) como se define en Elkan (2001). Los métodos sensibles al coste ayudan a priorizar y optimizar las decisiones en función de los costes asociados con los diferentes errores de clasificación. Este enfoque ayuda a los prestamistas a administrar el riesgo crediticio de manera efectiva, maximizando los beneficios y evitando pérdidas susceptibles.

Entre los enfoques más modernos para abordar los problemas sensibles al coste, se distinguen dos filosofías según se indica en Vanderschueren et al. (2022b). La primera, conocida como predecir y optimizar (predict-and-optimize), consiste en construir un clasificador con una perspectiva sensible a los costes, ajustando las

probabilidades estimadas como se propone en Bahnsen et al. (2014a); King and Zeng (2002); Yih et al. (2006) y Vanderschueren et al. (2022a) o usando versiones ponderadas de la regresión logística (Bahnsen et al. (2014a); Höppner et al. (2021)) o algoritmos de boosting (Nikolaou et al. (2016); Höppner et al. (2021)), entre otros. El segundo enfoque, conocido como *thresholding* o predecir y luego optimizar (predict-then-optimize), consiste en centrarse en la etapa de toma de decisiones. De esta forma, se entrena un modelo predictivo con el objetivo de maximizar la precisión y luego se optimiza la toma de decisiones minimizando las pérdidas, como se propone en Almhaithawi et al. (2020); Bahnsen et al. (2013); Elkan (2001); Höppner et al. (2021); Vanderschueren et al. (2022a,b) y Sheng and Ling (2006).

Los sistemas de riesgo de crédito como los mencionados anteriormente, a pesar de ser efectivos, tienen limitaciones. Su dependencia en datos históricos de clientes preseleccionados tiene como resultado una falta de información sobre prestatarios potenciales que anteriormente no se consideraban elegibles para recibir crédito. Esto puede llevar a que se juzgue injustamente a los clientes y que se pierdan nuevas oportunidades. Del mismo modo, los modelos estáticos enfrentan sesgos inherentes y el peligro de pérdidas debido a un cambio de tendencia dada su dependencia en datos históricos. El riesgo crediticio es un problema dinámico, ya que el comportamiento de los clientes cambia según el ciclo económico y otros factores. Por lo tanto, es necesario actualizar continuamente los modelos a medida que haya nueva información disponible. Esto se conoce como *aprendizaje por refuerzo* (reinforcement learning). Con este enfoque, se evitan el sesgo de los datos históricos y los modelos se mantienen actualizados en escenarios dinámicos.

Del mismo modo, para recopilar datos para el ajuste del modelo, los prestamistas tienen que afrontar riesgo de crédito e incurrir en pérdidas financieras para recopilar las etiquetas de prestatarios morosos. Si un prestamista intenta evitar esa exposición recopilando sólo unas pocas observaciones, el modelo estimado tendrá un problema de alta varianza. Por el contrario, si un prestamista recopila demasiadas observaciones, la mejora en el desempeño del modelo puede no justificar las pérdidas por otorgar préstamos a clientes morosos. Esto conduce a un dilema entre exploración

y explotación, como se define en Sutton and Barto (2018): ¿debería emplearse un nuevo cliente para obtener más información y mejorar las predicciones al coste potencial de incurrir en pérdidas, o debería ser la evaluación del modelo la que dicte la decisión? Como resultado, las instituciones financieras enfrentan dos desafíos que a menudo se pasan por alto: adaptarse a entornos dinámicos y al mismo tiempo optimizar tanto los riesgos inmediatos como el desempeño del modelo a largo plazo.

El aprendizaje en línea (online learning) se enfrenta a entornos dinámicos y actualiza continuamente los modelos a medida que hay nuevas observaciones disponibles. De esta manera, se obtienen decisiones actualizadas y ajustadas al contexto actual, como se destaca en Lattimore and Szepesvári (2020); Sutton and Barto (2018) y Hoi et al. (2018). Los algoritmos de online learning refinan continuamente sus predicciones observando datos, tomando una decisión, recibiendo retroalimentación sobre estas decisiones y adaptando sus mecanismos de predicción con la nueva información obtenida. El online learning adopta un enfoque continuo de "aprender mientras se hace", lo que lo convierte en la opción más razonable para escenarios donde los datos llegan de manera constante y la adaptación es primordial, como el problema del riesgo crediticio.

El online learning se ocupa de entornos dinámicos pero no considera la exploración. Dentro del paradigma del aprendizaje por refuerzo, los algoritmos de bandidos (bandit algorithms) están diseñados para equilibrar el proceso de búsqueda de nueva información (exploración) con la maximización de recompensas basadas en el conocimiento actual (explotación) bajo condiciones dinámicas. Los algoritmos de bandidos reciben su nombre de un ejemplo muy explicativo. Imagine a un jugador (agente) mirando una fila de tragaperras de una palanca (acciones), cada una de los cuales promete una recompensa desconocida. El agente tiene el dilema de tirar de la palanca familiar que actualmente devuelve monedas (explotación) o probar una nueva para obtener ganancias potencialmente mayores (exploración). En el entorno estocástico, cada brazo está asociado con una distribución que genera recompensas aleatorias, como se describe en Achab et al. (2018). El objetivo del agente es conocer la distribución de recompensas de cada palanca y encontrar una estrategia

para maximizar la recompensa acumulada. Una opción sería tirar de cada palanca disponible una vez y de ahí en adelante explotar la palanca que ofreciera el premio más alto. Sin embargo, esto podría no ser más que un golpe de suerte. Por otro lado, si el agente decide seguir explorando, puede perder las ganancias del brazo óptimo. Los algoritmos de bandido tienen como objetivo resolver este equilibrio entre explotación y exploración.

Como se destacó anteriormente, los diferentes impactos provocados por una clasificación errónea afectan significativamente a los resultados en problemas sensibles al coste. En consecuencia, recientemente se han propuesto e investigado algoritmos de aprendizaje por refuerzo sensibles al coste, cuyo objetivo es optimizar medidas sensibles al coste en tareas de clasificación dinámicas. El inconveniente de estos modelos es que no consideran el contexto y/o consideran un impacto fijo para cada error de decisión, como se afirma en Zhang et al. (2018) y Achab et al. (2018). Aunque esto puede ser razonable en algunos escenarios, como la publicidad online, en problemas como el riesgo de crédito, esto impone suposiciones que podrían llevar a resultados subóptimos tal como se indica en Bahnsen et al. (2014a); Elkan (2001); Höppner et al. (2021) y Vanderschueren et al. (2022b).

Considerando los inconvenientes presentes en los enfoques del estado del arte y la falta de alternativas disponibles para abordar problemas sensibles al coste como el problema del riesgo crediticio presentado por ASF, esta tesis propone nuevas técnicas para abordar el problema de maximizar beneficios y reducir pérdidas potenciales en financiación al consumo. Por tanto, se adopta una perspectiva sensible al coste para desarrollar modelos de clasificación y técnicas para la toma de decisiones.

En este trabajo se consideran las dos filosofías existentes para la clasificación sensible al coste (thresholding y predict-and-optimize) para brindar soluciones desde ambas perspectivas. En primer lugar, se introduce un nuevo algoritmo de thresholding sensible al coste (algoritmo 2-DDR) para estimar la regla de decisión óptima sobre un conjunto de datos dada una puntuación, el importe del préstamo y una especificación de costes. Se construye un espacio de decisión bidimensional tomando una puntuación estimada de que se produzca la mora/fraude y el importe

del préstamo. Sobre este espacio, el algoritmo explora, con total libertad, la regla de decisión óptima, buscando minimizar las pérdidas agregadas sobre una muestra representada sobre el espacio de decisión bidimensional. El objetivo es que al considerar conjuntamente la puntuación estimada y el importe de la operación, el algoritmo 2-DDR encuentre un equilibrio entre el riesgo de una operación y la pérdida potencial. En la búsqueda del algoritmo, están contenidas todas las técnicas de thresholding previas, por lo que siempre se obtienen resultados mejores o iguales a los obtenidos por las técnicas del estado del arte.

Posteriormente, el algoritmo 2-DDR se extiende al problema con respuesta múltiple considerando una sola puntuación. Por ejemplo, para el problema del riesgo de crédito, en las entidades financieras lo normal es contar con varias respuestas para una solicitud. Las posibles respuestas para una solicitud son aprobar, duda y denegar, por lo general. En caso de que la PD de la operación sea baja, se aprueba, si implica demasiado riesgo se deniega, y aquellas con una PD media se marcan como duda para que las estudie un analista interno. Sin embargo, las posibles clases de una operación son bueno o malo, por tanto hay que dar una respuesta múltiple (ternaria) a una variable binaria. Este problema no ha sido abordado previamente, ya que en clasificación multiclase se supone que se cuenta con una puntuación para cada clase. Iniciando el algoritmo 2-DDR desde distintas regiones del espacio de decisión, se generaliza para considerar respuestas múltiples. Con esta propuesta se obtiene empíricamente la mejor regla de decisión posible dada una puntuación para cualquier problema de clasificación. Por tanto, sólo queda centrarse en la estimación de la puntuación.

Para la estimación de la puntuación, utilizamos un enfoque paramétrico y semiparamétrico en aras de cubrir todos los enfoques posibles. Primero, se proponen estimadores sensibles al coste para clasificadores paramétricos. Para esto, se estima el vector de parámetros de un modelo paramétrico (un modelo logístico por ejemplo, como se propone en Höppner et al. (2021)) minimizando una función de pérdida en lugar de maximizando la verosimilitud como se suele hacer en los métodos clásicos. Con esta propuesta, el modelo obtenido está entrenado para minimizar las pérdidas

en lugar de para maximizar la precisión, dos objetivos que por lo general no son equivalentes. Para el modelo propuesto, se obtiene su consistencia y normalidad asintótica bajo condiciones generales.

Un inconveniente con el que nos encontramos con esta propuesta es que la función objetivo no es convexa en general, a diferencia de la función de verosimilitud. Esto hace que sea necesario introducir alguna penalización (regularización lasso por ejemplo) en la función objetivo para que los algoritmos de optimización puedan encontrar el mínimo. Introduciendo la penalización se solventa el problema de la falta de convexidad, pero esto tiene el precio de introducir sesgo en la estimación de los parámetros. Con el objetivo de evaluar el impacto que tiene la penalización, también se obtienen cotas para el sesgo introducido bajo condiciones bastante generales.

Con estos resultados, se garantiza teóricamente el buen funcionamiento del modelo propuesto. Este buen funcionamiento también se justifica en este trabajo por medio de un extensivo estudio de simulación y el análisis de dos bases de datos reales. Con todas estas evidencias, se concluye que el modelo propuesto es una alternativa interesante para resolver problemas sensibles al coste.

Una vez contamos con una técnica para estimar la puntuación considerando modelos paramétricos, se propone un clasificador semiparamétrico sensible al coste extendiendo el modelo de single-index propuesto en Klein and Spady (1993) para estimar la puntuación. Este modelo relaja las restricciones paramétricas dando total libertad a la función de enlace, pero impone ciertas condiciones paramétricas asumiendo que la variable respuesta depende del vector de covariables solo a través de una combinación lineal de este. De esta forma, se relajan las restricciones impuestas en los modelos paramétricos, pero dota al modelo de flexibilidad para modelar las relaciones de dependencia entre la variable respuesta y el vector de covariables. Así, se cuenta con un modelo flexible y robusto para tareas de clasificación. El enfoque que se sigue para desarrollar la version sensible al coste del modelo single-index es equivalente al considerado para el modelo paramétrico, solo que en este caso también se tiene que estimar de forma no paramétrica la función de enlace. Para esto, se extiende la propuesta en Klein and Spady (1993) estimando el vector de paráme-

tros de forma sensible al coste y se obtiene un modelo satisfactorio combinando la flexibilidad de los modelos single-index con un enfoque sensible al coste.

Con estas propuestas se soluciona el problema de estimar la puntuación y obtener la mejor decisión posible desde una perspectiva estática. Estos fueron los problemas planteados inicialmente por ASF. Sin embargo, el problema del riesgo crediticio es dinámico por definición. No obstante, este problema no ha sido resuelto previamente desde la perspectiva del aprendizaje por refuerzo, por lo que se considera de interés estudiar las mejoras que se pueden obtener en riesgo de crédito considerando un enfoque dinámico. Por tanto, se proponen varias técnicas para abordar el problema de clasificación sensible al coste desde una perspectiva de aprendizaje por refuerzo. De esta forma, se obtienen modelos eficaces combinando la mejora obtenida al considerar un enfoque sensible al coste con la flexibilidad proporcionada por el aprendizaje por refuerzo para adaptarse a escenarios dinámicos.

Se propone un algoritmo de online learning sensible al coste extendiendo el algoritmo pasivo-agresivo (PA) introducido en Crammer et al. (2006). Introduciendo una función de pérdida para cada observación según su clase e importe, se obtiene un clasificador que discrimina las operaciones satisfactoriamente teniendo en cuenta su impacto en caso de error. De esta forma, se obtienen mejores resultados en términos de pérdidas que con los algoritmos del estado del arte.

También se propone una versión sensible al coste del bandido logístico propuesto en Sutton and Barto (2018). Este modelo se entrena considerando el impacto de aprobar o no una solicitud de crédito para estimar la acción óptima, que puede consistir en explorar una operación arriesgada, aprobarla por tener beneficios potenciales o denegarla por tener demasiado riesgo de terminar en impago. La exploración realizada por el algorimo de bandido, combinado con un enfoque sensible al coste resulta en un modelo competente en clasificación dinámica sensible al coste.

Se espera que el modelo de bandido tenga un mejor rendimiento que el algoritmo PA, ya que el primero considera la exploración. Sin embargo, hay entidades para las que la estrategia de no ceñirse a la decisión óptima dada por el modelo actual

resulta demasiado agresiva. Por lo tanto, queremos ofrecer una alternativa sensible al coste más conservadora proponiendo un algoritmo de online learning (sin considerar exploración).

Con las técnicas propuestas se cumple el objetivo original que es optimizar los flujos de admisión y gestión de riesgos de la entidad colaboradora, ASF. Sin embargo, todas las propuestas pueden extenderse a cualquier problema sensible al coste.

Para demostrar empíricamente el buen desempeño de las técnicas propuestas a lo largo de este trabajo, se llevan a cabo un extenso número de experimentos sobre escenarios simulados y sobre conjuntos de datos reales, tanto del estado del arte como cedidos por ASF. Los resultados experimentales son prometedores y demuestran la eficacia y eficiencia de los modelos propuestos, especialmente si se comparan con las alternativas disponibles en el estado del arte. Además, estos estudios proporcionan información valiosa sobre el rendimiento de los distintos algoritmos disponibles tanto en escenarios simulados como reales. De esta manera, también llenamos el vacío relacionado con la falta de estudios comparativos manifestada en Petrides et al. (2022) y ampliamos los resultados de estudios anteriores como los obtenidos en Lessmann et al. (2015); Höppner et al. (2021); Baesens et al. (2003) y Vanderschueren et al. (2022b).

Por último, se dan conclusiones y algunos comentarios sobre futuras líneas de investigación a raíz de todas las técnicas propuestas y experimentos realizados en esta obra.

Bibliography

Achab, M., Clémençon, S., and Garivier, A. (2018). Profitable bandits. Proceedings of The 10th Asian Conference on Machine Learning, 95:694–709.

Almhaithawi, D., Jafar, A., and Aljnidi, M. (2020). Example-dependent cost-sensitive credit cards fraud detection using SMOTE and Bayes minimum risk. SN Applied Sciences, 2(9):1574–1585.

Alotaibi, R. and Flach, P. (2021). Multi-label thresholding for cost-sensitive classification. Neurocomputing, 436:232–247.

Alotaibi, R., Flach, P. A., and Kull, M. (2014). Multi-label classification: A comparative study on threshold selection methods.

Altman, E. I., Marco, G., and Varetto, F. (1994). Corporate distress diagnosis: Comparisons using linear discriminant analysis and neural networks (the italian experience). Journal of Banking & Finance, 18 (3):502–529.

Auer, P., Cesa-Bianchi, N., and Fischer, P. (2002). Finite-time analysis of the multiarmed bandit problem. Machine Learning, 47:235–256.

Baesens, B., Gestel, T. V., Viaene, S., Stepanova, M., Suykens, J., and Vanthienen, J. (2003). Benchmarking state-of-the-art classification algorithms for credit scoring. The Journal of the Operational Research Society, 54(6):627–635.

Bahnsen, A. C., Aouada, D., and Ottersten, B. (2014a). Example-dependent cost-sensitive logistic regression for credit scoring. Proceedings - 2014 13th International Conference on Machine Learning and Applications, ICMLA 2014, 1:263–269.

Bahnsen, A. C., Stojanovic, A., Aouada, D., and Ottersten, B. (2013). Cost sensitive credit card fraud detection using Bayes minimum risk. Proceedings - 2013 12th International Conference on Machine Learning and Applications, ICMLA 2013, 1:333–338.

Bahnsen, A. C., Stojanovic, A., Aouada, D., and Ottersten, B. (2014b). Improving credit card fraud detection with calibrated probabilities. Proceedings of the 2014 SIAM International Conference on Data Mining. Pennsylvania, USA.

Bequé, A., Coussement, K., Gayler, R., and Lessmann, S. (2017). Approaches for credit scorecard calibration: An empirical analysis. Knowledge-Based Systems, 134:213–227.

Bourke, C., Deng, K., Scott, S., Schapire, R., and Vinodchandran, N. (2008). On reoptimizing multi-class classifiers. Machine Learning, 71:219–242.

Broyden, C. G. (1970). The convergence of a class of double-rank minimization algorithms 1. General considerations. IMA Journal of Applied Mathematics, 6:76–90.

C-Rella, J., Cao, R., and Vilar, J. M. (2024a). Cost-sensitive single index classification model. Technical Report.

C-Rella, J., Cao, R., and Vilar, J. M. (2024b). Cost-sensitive thresholding over a two-dimensional decision region for fraud detection. Information Sciences, 657:119956.

C-Rella, J., Claeskens, G., Cao, R., and Vilar, J. M. (2023). Instance-dependent cost-sensitive parametric learning for fraud detection. Technical Report.

C-Rella, J., Martinez Rego, D., and Vilar, J. M. (2024c). Cost-sensitive reinforcement learning for credit risk. Technical Report.

C-Rella, J. and Vilar, J. M. (2024). Flexible multi-class cost-sensitive thresholding. Technical Report.

Cao, R., Vilar, J. M., and Devia Rivera, A. E. (2009). Modelling consumer credit risk via survival analysis. Statistics and Operations Research Transactions (SORT), 33(1):3–30.

Chawla, N., Bowyer, K., Hall, L., and Kegelmeyer, W. (2002). SMOTE: Synthetic minority over-sampling technique. Journal of Artificial Intelligence Research (JAIR), 16:321–357.

Chen, T. and Guestrin, C. (2016). Xgboost: a scalable tree boosting system. KDD '16: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 785–794.

Chu, W., Li, L., Reyzin, L., and Schapire, R. (2011). Contextual bandits with linear payoff functions. Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, 15:208–214.

Collell, G., Prelec, D., and Patil, K. R. (2018). A simple plug-in bagging ensemble based on threshold-moving for classifying binary and multiclass imbalanced data. Neurocomputing, 275:330–340.

Committee, B. (2001). The new basel capital accord. Bank for International Settlements.

Cosslett, S. (1983). Distribution-free maximum likelihood estimator of the binary choice model. Econometrica, 51:765–782.

Crammer, K., Dekel, O., Keshet, J., Shalev-Shwartz, S., and Singer, Y. (2006). Online passive-aggressive algorithms. Journal of Machine Learning Research, 7(19):551–585.

Crook, J. N., Edelman, D. B., and Thomas, L. C. (2007). Recent developments in consumer credit risk assessment. European Journal of Operational Research, 183(3):1447–1465.

Dal Pozzolo, A., Caelen, O., Johnson, R. A., and Bontempi, G. (2015). Calibrating probability with undersampling for unbalanced classification. 2015 IEEE Symposium Series on Computational Intelligence, 2015:159–166.

De Vos, S., Vanderschueren, T., Verdonck, T., and Verbeke, W. (2023). Robust instance-dependent cost-sensitive classification. Advances in Data Analysis and Classification, 17:1057–1079.

Delecroix, M., Hristache, M., and Patilea, V. (2006). On semiparametric m-estimation in single-index regression. Journal of Statistical Planning and Inference, 136:730–769.

Desai, V. S., Crook, J. N., and Overstreet, G. A. (1996). A comparison of neural networks and linear scoring models in the credit union environment. European Journal of Operational Research, 95(1):24–37.

Devi, D., Biswas, S. K., and Purkayastha, B. (2019). A cost-sensitive weighted random forest technique for credit card fraud detection. 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT), 2019:1–6.

Dmochowski, J., Sajda, P., and Parra, L. (2010). Maximum likelihood in cost-sensitive learning: Model specification, approximations, and upper bounds. Journal of Machine Learning Research, 11:3313–3332.

Dredze, M., Crammer, K., and Pereira, F. (2008). Confidence-weighted linear classification. Proceedings of the 25th International Conference on Machine Learning, pages 264–271.

Dumitrascu, B., Feng, K., and Engelhardt, B. E. (2018). Pg-ts: Improved thompson sampling for logistic contextual bandits. Proceedings of the 32nd International Conference on Neural Information Processing Systems, pages 4629–4638.

Elkan, C. (2001). The foundations of cost-sensitive learning. International Joint Conference on Artificial Intelligence, Lawrence Erlbaum Associates Ltd., 2001:973–978.

Elmachtoub, A. N. and Grigas, P. (2020). Smart "predict, then optimize". ArXiv, abs/1710.08005.

Faury, L., Abeille, M., Calauzenes, C., and Fercoq, O. (2020). Improved optimistic algorithms for logistic bandits. Proceedings of the 37th International Conference on Machine Learning, 119:3052–3060.

Fletcher, R. (1970). A new approach to variable metric algorithms. The Computer Journal, 13(3):317–322.

Freund, Y. and Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences, 55(1):119–139.

Gamerman, D. and Lopes, H. (2006). Markov Chain Monte Carlo: Stochastic Simulation for Bayesian Inference, Second Edition (2nd ed.). Chapman and Hall/CRC.

Goldfarb, D. (1970). A family of variable metric updates derived by variational means. Mathematics of Computation, 24(109):23–26.

Hardle, W., Hall, P., and Ichimura, H. (1993). Optimal smoothing in single-index models. The Annals of Statistics, 21(1):157 – 178.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition. Springer Series in Statistics.

Hazan, E., Rakhlin, A., and Bartlett, P. (2007). Adaptive online gradient descent. Advances in Neural Information Processing Systems, 20.

He, H. and A. Garcia, E. (2009). Learning from imbalanced data. IEEE Transactions on knowledge and data engineering, 21 (9):1263–1284.

Henningsen, A. and Toomet, O. (2011). maxlik: A package for maximum likelihood estimation in R. Computational Statistics, 26(3):443–458.

Hoi, S. C. H., Sahoo, D., Lu, J., and Zhao, P. (2018). Online learning: A comprehensive survey. ArXiv, abs/1802.02871.

Höppner, S., Baesens, B., Verbeke, W., and Verdonck, T. (2021). Instance-dependent cost-sensitive learning for detecting transfer fraud. European Journal of Operational Research, 297:291–300.

Horowitz, J. L. (1992). A smoothed maximum score estimator for the binary response model. Econometrica, 60(3):505–531.

Horowitz, J. L. and Härdle, W. (1996). Direct semiparametric estimation of single-index models with discrete covariates. Journal of the American Statistical Association, 91(436):1632–1640.

Hristache, M., Juditsky, A., and Spokoiny, V. (2000). Direct estimation of the index coefficient in a single-index. The Annals of Statistics, 29:593–623.

Huang, K.-H. and Lin, H.-T. (2017). Cost-sensitive label embedding for multi-label classification. Machine Learning, 106(9-10):1725–1746.

Ichimura, H. (1993). Semiparametric least squares (SLS) and weighted sls estimation of single-index models. Journal of Econometrics, 58(1):71–120.

Kaufmann, E., Cappé, O., and Garivier, A. (2014). On the complexity of best-arm identification in multi-armed bandit models. Journal of Machine Learning Research, 17:1–42.

Kaufmann, E., Korda, N., and Munos, R. (2012). Thompson sampling: An asymptotically optimal finite time analysis. ArXiv, 1205.4217.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., and Liu, T.-Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. Advances in Neural Information Processing Systems, 30.

Ker, A. P. and Sam, A. G. (2018). Semiparametric estimation of the link function in binary-choice single-index models. Computational Statistics, 33:1429–1455.

King, G. and Zeng, L. (2002). Logistic regression in rare events data. Political Analysis, 9 (2):137–163.

Kivinen, J., Smola, A., and Williamson, R. (2004). Online learning with kernels. IEEE Transactions on Signal Processing, 52(8):2165–2176.

Klein, R. and Spady, R. (1993). An efficient semiparametric estimator of the binary response models. Econometrica, 61:387–421.

Korda, N., Kaufmann, E., and Munos, R. (2013). Thompson sampling for 1-dimensional exponential family bandits. Advances in Neural Information Processing Systems, 26.

Lai, T. and Robbins, H. (1985). Asymptotically efficient adaptive allocation rules. Advanced in Applied Mathematics, 6:4–22.

Langford, J. and Zhang, T. (2007). The epoch-greedy algorithm for multi-armed bandits with side information. Advances in Neural Information Processing Systems, 20:661–670.

Lattimore, T. and Szepesvári, C. (2020). Bandit Algorithms. Cambridge University Press.

Lessmann, S., Baesens, B., Seow, H.-V., and Thomas, L. (2015). Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research. European Journal of Operational Research, 247:124–136.

Li, L., Chu, W., Langford, J., and Schapire, R. (2010). A contextual-bandit approach to personalized news article recommendation. WWW '10: Proceedings of the 19th international conference on World wide web, pages 661–670.

Li, Y., Zaragoza, H., Herbrich, R., Shawe-Taylor, J., and Kandola, J. (2002). The perceptron algorithm with uneven margins. International Conference on Machine Learning.

Linhart, C., Harari, G., Abramovich, S., and Buchris, A. (2009). PAKDD data mining competition 2009: new ways of using known methods. Proceedings of the 13th Pacific-Asia International Conference on Knowledge Discovery and Data Mining: New Frontiers in Applied Data Mining, PAKDD'09:99–105.

Loh, P.-L. and Wainwright, M. (2015). Regularized m-estimators with nonconvexity: Statistical and algorithmic theory for local optima. Journal of Machine Learning Research, 16:559–616.

Lucas, Y. and Jurgovsky, J. (2020). Credit card fraud detection using machine learning: A survey. ArXiv, abs/2010.06479.

Malhotra, R. and Malhotra, D. K. (2002). Differentiating between good credits and bad credits using neuro-fuzzy systems. European Journal of Operational Research, 136(1):190–211.

Min, J. H. and Lee, Y.-C. (2008). A practical approach to credit scoring. Expert Systems with Applications, 35(4):1762–1770.

Nikolaou, N., Edakunni, N., Kull, M., Flach, P., and Brown, G. (2016). Cost-sensitive boosting algorithms: Do we really need them? Machine learning, 104:359–384.

Nowok, B., Raab, G. M., and Dibben, C. (2016). Synthpop: Bespoke creation of synthetic data in R. Journal of Statistical Software, 74:1–26.

O'Brien, D. B., Gupta, M. R., and Gray, R. M. (2008). Cost-sensitive multi-class classification from probability estimates. Proceedings of the 25th International Conference on Machine Learning, pages 712–719.

Pandey, S., Agarwal, D., Chakrabarti, D., and Josifovski, V. (2007). Bandits for taxonomies: A model-based approach. Proceedings of the 2007 SIAM International Conference on Data Mining (SDM), pages 216–227.

Peláez, R., Cao, R., and Vilar, J. M. (2020). Probability of default estimation in credit risk using a nonparametric approach. TEST, 30:383–405.

Peláez, R., Cao, R., and Vilar, J. M. (2021). Nonparametric estimation of the probability of default with double smoothing. Statistics and Operations Research Transactions (SORT), 45(2):93–120.

Peng, H. and Huang, T. (2011). Penalized least squares for single index models. Journal of Statistical Planning and Inference, 141:1362–1379.

Pesántez-Narváez, J. and Guillén, M. (2020). Penalized logistic regression to improve predictive capacity of rare events in surveys. Journal of Intelligent & Fuzzy Systems, 38:1–11.

Petrides, G., Moldovan, D., Coenen, L., Guns, T., and Verbeke, W. (2022). Cost-sensitive learning for profit-driven credit scoring. Journal of the Operational Research Society, 73(2):338–350.

Piramuthu, S. (1999). Financial credit-risk evaluation with neural and neurofuzzy systems. European Journal of Operational Research, 112(2):310–321.

Powell, J. L., Stock, J. H., and Stoker, T. M. (1989). Semiparametric estimation of index coefficients. Econometrica, 57(6):1403–1430.

R Core Team (2021). R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria.

Rabbi, M., Aung, M. H., Zhang, M., and Choudhury, T. (2015). Mybehavior: Automatic personalized health feedback from user behaviors and preferences using smartphones. Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing, pages 707–718.

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. Psychological Review, 65(6):368–408.

Rothe, C. (2009). Semiparametric estimation of binary response models with endogenous regressors. Journal of Econometrics, 153(1):51–64.

Russo, D., Roy, B. V., Kazerouni, A., Osband, I., and Wen, Z. (2020). A tutorial on thompson sampling. ArXiv, abs/1707.02038.

Saitta, L. and Geibel, P. (2001). Machine learning: A Technological Roadmap [2nd ed.].

Samreen, A., Zaidi, S. F., and Sarwar, A. (2013). Design and development of credit scoring model for the commercial banks in pakistan: Forecasting creditworthiness of corporate borrowers. International Journal of Business and Social Science, 3:155–166.

Shalev-Shwartz, S. (2012). Online learning and online convex optimization. Foundations and Trends in Machine Learning, 4:107–194.

Shanno, D. and Kettler, P. (1970). Optimal conditioning of quasi-newton methods. Mathematics of Computation, 24:657–657.

Shen, F., Wang, R., and Shen, Y. (2020). A cost-sensitive logistic regression credit scoring model based on multi-objective optimization approach. Technological and Economic Development of Economy, 26(2):405–429.

Shen, W., Wang, J., Jiang, Y.-G., and Zha, H. (2015). Portfolio choices with orthogonal bandit learning. Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI 2015).

Sheng, V. and Ling, C. (2006). Thresholding for making classifiers cost sensitive. Proceedings of the National Conference on Artificial Intelligence, 1:476–481.

Shu, J., Yuan, X., Meng, D., and Xu, Z. (2023). CMW-Net: Learning a class-aware sample weighting mapping for robust deep learning. ArXiv, abs/2202.05613.

Siddiqi, N. (2006). Credit risk scorecards, developing and implementing intelligent credit scoring. John Wiley & Sons, Inc., Hoboken, NJ.

Stripling, E., Broucke, S. v., Antonio, K., Baesens, B., and Snoeck, M. (2018). Profit maximizing logistic model for customer churn prediction using genetic algorithms. Swarm and Evolutionary Computation, 40:116–130.

Sutton, R. S. and Barto, A. G. (2018). Reinforcement Learning: An Introduction. A Bradford Book, Cambridge, MA, USA. ISBN : 978-0-262-19398-6.

Tang, L., Rosales, R., Singh, A., and Agarwal, D. (2013). Automatic ad format selection via contextual bandits. Proceedings of the 22nd ACM International Conference on Information and Knowledge Management, (CIKM '13):1587–1594.

Thomas, L., Edelman, D., and Crook, J. (2002). Credit scoring and its applications.

Thomas, L. C. (2000). A survey of credit and behavioural scoring: forecasting financial risk of lending to consumers. International Journal of Forecasting, 16(2):149–172.

Thompson, W. R. (1933). On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. Biometrika, 25(3/4):285–294.

Vaart, A. W. (1998). Asymptotic statistics. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press.

Vanderschueren, T., Baesens, B., Verdonck, T., and Verbeke, W. (2022a). A new perspective on classification: optimally allocating limited resources to uncertain tasks. arXiv, abs/2202.04369.

Vanderschueren, T., Verdonck, T., Baesens, B., and Verbeke, W. (2022b). Predict-then-optimize or predict-and-optimize? An empirical evaluation of cost-sensitive learning strategies. Information Sciences, 594:400–415.

Vapnik, V. (2000). The nature of statistical learning theory, volume 8. Springer, New York.

Vapnik, V. N. (1998). Statistical Learning Theory. Wiley. ISBN: 978-0-471-03003-4.

Verbeke, W., Dejaeger, K., Martens, D., Hur, J., and Baesens, B. (2012). New insights into churn prediction in the telecommunication sector: A profit driven data mining approach. European Journal of Operational Research, 218(1):211–229.

Verbeke, W., Olaya, D., Guerry, M., and Van Belle, J. (2023). To do or not to do? Cost-sensitive causal classification with individual treatment effect estimates. European Journal of Operational Research, 305(2):838–852.

Verbraken, T., Bravo, C., Weber, R., and Baesens, B. (2014). Development and application of consumer credit scoring models using profit-based classification measures. European Journal of Operational Research, 238:505–513.

Verbraken, T., Verbeke, W., and Baesens, B. (2012). A novel profit maximizing metric for measuring classification performance of customer churn prediction models. IEEE Transactions on Knowledge and Data Engineering, 25:961 – 973.

Visantavarakul, K. (2022). An application of reinforcement learning to credit scoring based on the logistic bandit framework. Chulalongkorn University Theses and Dissertations (Chula ETD 6050).

Wang, C.-C., Kulkarni, S., and Poor, H. (2005). Bandit problems with side observations. IEEE Transactions on Automatic Control, 50(3):338–355.

Wang, J., Zhao, P., and Hoi, S. C. (2012). Cost-sensitive online classification. 2012 IEEE 12th International Conference on Data Mining, pages 1140–1145.

West, D. (2000). Neural network credit scoring models. Computers & Operations Research, 27(11):1131–1152.

Yih, W., Goodman, J., and Hulten, G. (2006). Learning at low false positive rates. The Third Conference on Email and Anti-Spam, 2006.

Youden, W. J. (1950). Index for rating diagnostic tests. Cancer, 3 (1):32–35.

Yusuf, S. and Ekrem, D. (2011). Detecting credit card fraud by decision trees and support vector machines. IMECS 2011 - International MultiConference of Engineers and Computer Scientists 2011, 1:442–447.

Zadrozny, B. and Elkan, C. (2001). Learning and making decisions when costs and probabilities are both unknown. Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2001:204–213.

Zhang, L., Yang, T., Jin, R., Xiao, Y., and Zhou, Z. (2016). Online stochastic linear optimization under one-bit feedback. Proceedings of The 33rd International Conference on Machine Learning, 48:392–401.

Zhang, Y., Wu, M., Hu, X., and Zhu, Y. (2018). An online transfer learning algorithm with adaptive cost. Proceedings of the 2018 International Conference on Signal Processing and Machine Learning.

Zhao, P., Zhuang, F., Wu, M., Li, X., and Hoi, S. C. H. (2015). Cost-sensitive online classification with adaptive regularization and its applications. IEEE International Conference on Data Mining ICDM 2015, pages 649–658.

Zhou, L. (2016). A survey on contextual multi-armed bandits. ArXiv, 1508.03326.

Zhou, Z. and Liu, X. (2006). Training cost-sensitive neural networks with methods addressing the class imbalance problem. IEEE Transactions on Knowledge and Data Engineering, 18:63–77.

Zhou, Z. and Liu, X. (2010). On multi-class cost-sensitive learning. Computational Intelligence, 26(3):232–257.