Statistical Learning in Complex and Temporal Data

Distances, Two-sample testing, Clustering, Classification and Big Data.

UNIVERSIDADE DA CORUÑA

Pablo Montero Manso

Doctoral Thesis

November 2018

Statistical Learning in Complex and Temporal Data

Distances, Two-sample testing, Clustering, Classification and Big Data.

UNIVERSIDADE DA CORUÑA

Pablo Montero Manso

Supervisor: José A. Vilar Fernández

Doctoral Program in Statistics and Operational Research
University of A Coruña

Doctoral Thesis

November 2018

Declaration

The undersigned certify that he is the advisor of the Doctoral Thesis entitled “Statistical Learning in Complex and Temporal Data: Distances, Two-sample testing, Clustering, Classification and Big Data”, developed by Pablo Montero Manso at the University of A Coruña (Department of Mathematics), as part of the interuniversity PhD program (UDC, USC and UVigo) of Statistics and Operational Research, opting for the International Doctorate, and hereby gives his consent to the author to proceed with the thesis presentation and the subsequent defense.

El abajo firmante hace constar que es el directores de la Tesis Doctoral titulada “Statistical Learning in Complex and Temporal Data: Distances, Two-sample testing, Clustering, Classification and Big Data”, realizada por Pablo Montero Manso en la Universidade da Coruña (Departamento de Matemáticas) en el marco del programa interuniversitario (UDC, USC y UVigo) de doctorado en Estadística e Investigación Operativa, optando a la Mención Internacional de Doctorado, y da su consentimiento para que el autor proceda a su presentación y posterior defensa.

O abaixo asinante fai constar que é os director da Tese de Doutoramento titulada “Statistical Learning in Complex and Temporal Data: Distances, Two-sample testing, Clustering, Classification and Big Data”, desenvolta por Pablo Montero Manso na Universidade da Coruña (Departamento de Matemáticas) no marco do programa interuniversitario (UDC, USC e UVigo) de doutoramento en Estatística e Investigación de Operacións, optando á Mención Internacional de Doutoramento, e da o seu consentimento para que o autor proceda á súa presentación e posterior defensa.

A Coruña, November 21, 2018

Supervisor:	PhD Student:
José A. Vilar Fernández	Pablo Montero Manso

Acknowledgements

Properly thanking my supervisor José A. Vilar would require too much space. For the sake of the trees and environment in general, it is left as an exercise for the imaginative reader. I would like to extend my thanks to the MODES Research group, for funding my predoctoral research, and to the many reviewers of the work in this thesis, especially to Rosa M. Crujeiras Casais, Javier Roca Pardiñas and Rubén Fernández Casal.

This work has received financial support from the Spanish Ministerio de Economía y Competitividad (grant MTM2014-52876-R and MTM2017-82724-R), the Xunta de Galicia (Centro Singular de Investigación de Galicia accreditation ED431G/01 2016-2019 and Grupos de Referencia Competitiva ED431C2016-015) and the European Union (European Regional Development Fund - ERDF)

Abstract

This thesis deals with the problem of statistical learning in complex objects, with emphasis on time series data. The problem is approached by facilitating the introduction of domain knoweldge of the underlying phenomena by means of distances and features. A distance-based two sample test is proposed, and its performance is studied under a wide range of scenarios. Distances for time series classification and clustering are also shown to increase statistical power when applied to two-sample testing. Our test compares favorably to other methods regarding its flexibility against different alternatives. A new distance for time series is defined by considering an innovative way of comparing lagged distributions of the series. This distance inherits the good empirical performance of existing methods while removing some of their limitations. A forecast method based on times series features is proposed. The method works by combining individual standard forecasting algorithms using a weighted average. These weights come from a learning model fitted on a large training set. A distributed classification algorithm is proposed, based on comparing, using a distance, the empirical distribution functions between the dataset that each computing node receives and the test set.

Resumen

Esta tesis trata sobre aprendizaje estadístico en objetos complejos, con énfasis en series temporales. El problema se aborda introduciendo conocimiento del dominio del fenómeno subyacente, mediante distancias y características.

Se propone un test de dos muestras basado en distancias y se estudia su funcionamiento en un gran abanico de escenarios. La distancias para clasificación y clustering de series temporales consiguen un incremento de la potencia estadística cuando se aplican al tests de dos muestras. Nuestro test se compara favorablemente con otros métodos gracias a su flexibilidad antes diferentes alternativas.

Se define una nueva distancia entre series temporales mediante una manera innovadora de comparar las distribuciones retardadas de la series. Esta distancia hereda el buen funcionamiento empírico de otros métodos pero elimina algunas de sus limitaciones.

Se propone un método de predicción basado en características de las series. El método combina diferentes algoritmos estándar de predicción mediante una suma ponderada. Los pesos de esta suma salen de un modelo que se ajusta a un conjunto de entrenamiento de gran tamaño.

Se propone un método de clasificación distribuida, basado en comparar, mediante una distancia, las funciones de distribución empírica del conjuto de prueba común y las de los datos que recibe cada nodo de cómputo.

Resumo

Esta tesis trata sobre aprendizaxe estatístico en obxetos complexos, con énfase en series temporais. O problema abórdase introducindo coñecemento sobre o dominio do fenómeno subxacente, mediante distancias e características.

Proponse un contraste de dúas mostras basado en distancias e estúdase o seu funcionamento nun gran abanico de escenarios. As distancias para clasificación e clustering de series temporais acadan un incremento da potencia estatística cando se aplican a contrastes de dúas mostras. O noso test compárase de xeito favorable con outros métodos gracias á súa flexibilidade ante diferentes alternativas.

Defínese unha nova distancia entre series temporais mediante un xeito innovador de comparar as distribucións retardadas das series. Esta distancia herda o bo funcionamento empírico doutros métodos pero elimina algunhas das súas limitacións.

Proponse un método de predicción baseada en características das series. O método combina diferentes algoritmos estándar de predicción mediante unha suma ponderada. Os pesos desta suma veñen dun modelo que se axusta a un conxunto de entrenamento de gran tamaño.

Propónse un método de clasificación distribuida, baseado en comparar, mediante unha distancia, as funcións de distribución empíricas do conxuto de proba común e as dos datos que recibe cada nodo de cómputo.

Preface

This thesis deals with the problem of statistical learning in complex objects, with emphasis on time series data. The studied learning tasks are two-sample testing, supervised classification, clustering and forecasting. Complex objects are data for which standard statistical methods, when applied in a straighforward manner, fail to achive satisfactory results. Complex data appear in multiple domains such as time series, shapes, images or text. The problem is approached by facilitating the introduction of domain knoweldge of the underlying phenomena by means of distances and features.

In Chapter 2, a distance-based two sample test is proposed, which shows a high degree of flexibility on the kinds of distances it accepts and high power under different alternatives, particularly for both location and scale alternatives. In Chapter 3, the proposed test is specifically analyzed in the context of time series, where special tools are needed due to the complexity that temporal dependence introduces. In this scenario, distances designed to deal with time series classification and clustering are shown to be also very useful when applied to the two-sample testing, greatly increasing statistical power when compared to standard procedures based on the Euclidean distance. Due to its flexibility, our test also compares favorably to other methods in this setting.

In Chapter 4, a new distance for time series is defined by considering an innovative way of comparing lagged distributions of the series. This distance inherits the good empirical performance of existing methods while removing some of their limitations. The distance can also be considered as a nonparametric extension of model-based time series distances, with the additional difference that it selects the meaningful lags based on maximizing discrimination between groups of series, not prediction.

A forecast method is proposed in Chapter 5. The method works by combining individual standard forecasting algorithms using a weighted average. These weights come from a feature-based learning model fitted on a large training set.

In Chapter 6, a distributed classification algorithm is proposed to minimize the impact on classification accuracy that comes when data is fragmented in different computing nodes and cannot be joined. The approach is based on comparing the empirical distribution functions between the dataset at each node and the test set, using a distance between distributions. Weights are assigned to each observation in the test set for each node in order to minimize this distance. To compute the final output of the distributed system, predictions done at each node are combined using these weights, using a weighted sum. This method improves the classification accuracy of the distributed system without requiring the sharing of the data that each node has received.

Table of contents

List of figures xxi

List of tables xxv

1 Introduction 1

2 Two-sample testing by comparing interpoint distances distributions 11

2.1 Introduction . . . 11

2.1.1 Two-sample homogeneity test . . . 13

2.1.2 Related work . . . 13

2.1.3 Overview . . . 15

2.2 Methodology . . . 15

2.2.1 The test statistic . . . 15

2.2.2 Theoretical foundation . . . 17

2.2.3 Motivation . . . 19

2.3 Experiments . . . 21

2.3.1 Some alternative test procedures . . . 22

2.3.2 Multivariate Gaussian distributions . . . 23

2.3.3 Differences in skewness and kurtosis . . . 26

2.3.4 Effect of large location shift combined with small covariance difference . . . 30

2.3.5 Comparing grids of bivariate Gaussians distributions . . . 31

2.3.6 Time series . . . 32

2.3.7 Discrete data . . . 34

2.3.8 Some benchmark datasets in classification . . . 35

2.4 Concluding remarks . . . 39

3 Time series two-sample testing 43

3.1 Introduction . . . 43

3.2 A rationale for a new test in the time series setting 45
3.3 Simulation study 47
3.3.1 Results 50
3.4 Applicative example 55
3.5 Conclusions and further research 56

4 A distance for time series and shapes based on lagged distributions 59
4.1 Introduction 59
4.2 A distance between lag embeddings 62
4.2.1 Lag embedding of a time series 62
4.2.2 Comparing lag embeddings 63
4.2.3 Lag and embedding dimension selection 64
4.3 A motivating example 64
4.4 Experimental results 68
4.4.1 Real data for classification 69
4.4.2 Simulation 69
4.5 Conclusions and future work 73

5 FFORMA: Feature-based FORecasting Model Averaging 75
5.1 Introduction 75
5.2 Methodology 77
5.2.1 Intuition and overview of FFORMA 77
5.2.2 Algorithmic description 78
5.3 Implementation and application to the M4 competition 79
5.3.1 Reference set 79
5.3.2 Time series features 79
5.3.3 Pool of forecasting methods 81
5.3.4 Forecast loss measure 81
5.3.5 Meta-learning model implementation 82
5.3.6 Prediction intervals 83
5.3.7 Results in the M4 Competition 85
5.4 Discussion and conclusions 87

6 Distributed classification using distances between probability distributions 89
6.1 Introduction 89
6.2 A novel distributed learning model 92

6.2.1 Background 92
6.2.2 An overview of our approach 93
6.2.3 Outline of the methodology 95
6.2.4 Measuring dissimilarity between high-dimensional distributions . 96
6.2.5 Weighting the belief values generated by the single classifiers . . 97
6.2.6 Combining the belief values generated by the single classifiers . 98
6.2.7 Some remarks 100
6.3 Experiments 102
6.3.1 Experimental setup 102
6.3.2 Some motivating experiments 106
6.3.3 Results 108
6.4 Computational cost 113
6.5 Conclusions and future work 114

7 Conclusions 117
7.1 Contributions by topic 120
7.2 Future work 121

References 123

Appendix A Proofs of theoretical results in Chapter 2 133

Appendix B Additional results for distributed classification 137

Appendix C Resumen extendido en castellano 147

List of figures

1.1 Simulated scenario of linear autoregressive processes. When looking for differences between these two groups using a two-sample hypothesis test, we highlight the fact that taking the series as multivariate observations produces much less statistical power that the same test over the coefficients extracted from the series. While the left panel series have been generated from the AR(1) process defined by $X_t = -0.3X_{t-1} + \varepsilon_t$, the ones on the right panel come from the AR(2) process $X_t = -0.3X_{t-1} + 0.5X_{t-2} + \varepsilon_t$. In both cases, independent standard Gaussian innovations ε_t were used. Each series is of length 50 and all of them were normalized to have zero mean and unity variance. . . . 2

1.2 Algorithmic definition of the TWED distance between two time series. Three different operations can be applied to the observations of the time series, in order. Each operation has a cost associated and the distance is defined as the minimum possible cost to make the two time series match exactly. This is an example of a distance that can be defined algorithmically but does not have a feature based equivalent. 5

1.3 Map of learning algorithms 8

2.1 Two samples, originally simulated from the von Mises-Fisher distribution and then scaled to produce the two concentric “circles”. 20

2.2 Means and density estimates for within- and between-samples distances using: (a) interpoint Euclidean distances, and (b) quantities defined in RKHSs by computing the Gaussian kernel with median window on pairs of points. Vertical lines represent the means of each density. 21

2.3 Empirical power vs dimension for several two-sample test procedures using multidimensional Gaussian distributions with mean shift. 24

2.4 Empirical power vs dimension for several two-sample test procedures using multidimensional Gaussian distributions with variance change. . . 25

2.5 Empirical power vs skewness level controlled by ν for several two-sample tests. Skewness increases with ν (symmetry if $\nu = 0$). 27
2.6 Empirical power vs kurtosis level controlled by τ for several two-sample tests. While $\tau = 1$ corresponds to the standard normal, $\tau > 1$ ($\tau < 1$) means greater (smaller) kurtosis than the standard normal. 27
2.7 Empirical power vs dimension for several two-sample tests when asymmetry on the first variable moves is set to $\nu = 3$. 29
2.8 Empirical power vs dimension for several two-sample tests when asymmetry on the first variable is set to $\tau = 0.2$. 29
2.9 Power vs dimension for large mean, small covariance alternative (Wei Simulation 3). 31
2.10 A realization of the Gaussian grids forming Scenario S8. 32

3.1 (a) Densities of within- and between-samples for series generated from AR(1) $\varphi = 0.3$ (A) and AR(1) $\varphi = 0.8$ (B) processes. (b) Kernel densities of the Euclidean distances between samples and within samples when the latter is obtained by joining the withinA and withinB distances. Vertical lines represent the mean values of both densities. While means may be close, the densities are still easily distinguised. 46
3.2 (a) EEG reading of an alcoholic subject. (b) EEG reading of a control subject. 56

4.1 (a) An otolith picture that has been transformed into a matrix of binaries. Red dots represent points randomly sampled inside the shape. The density formed by all pairwise distances between sample points is used to represent the original shape. (b) The results of representing the whole dataset of otoliths as densities. Color represents target class. . . 66
4.2 (a) Arc length representation of the edges of the otolith shape of Figure 4.1a. (b) Interpoint Euclidean distance representation of the shapes of the dataset, after applying the arc length parametrization to them. . . 67
4.3 Classification accuracy in the otolith example usign lag embedding limited to only one lag. We can see that all possible values of lag improve (or never decreases) the accuracy. 68

5.1 Flowchart of the meta-learning process applied to the M4 dataset. . . . 84
5.2 Importance of the features of the `xgboost` model fitted to the M4 dataset. 86

6.1 Graphical illustration of the per-Instance Weighting (pIW) criterion. . . 98

6.2 Distributed approaches schemes. 101

6.3 Plot of a simulated trial from a 2-dimensional version of Simul-C2 scenario. Color identifies the class. 105

6.4 Correlation between a distributional distance (the energy statistic) and classifier accuracy. 107

6.5 Accuracy-based interaction plot to check the joint effect of classifier, learning model and scenario. 110

6.6 Average classification accuracy values aggregated by decision rules. The horizontal red line indicates the average accuracy for the ND model. . . 111

6.7 Average accuracy as function of the partition size. The horizontal red lines indicate the average accuracy for the ND model. 112

6.8 Average accuracy as function of the partition size for each data set. The horizontal blue lines indicate the average accuracy for the ND model. . 113

B.1 Recall-based interaction plot to check the joint effect of classifier, learning model and scenario. 138

B.2 Average recall values aggregated by decision rules. The horizontal red line indicates the average recall for the ND model. 139

B.3 Average recall as function of the partition size. The horizontal red lines indicate the average recall for the ND model. 140

B.4 Average recall as function of the partition size for each data set. The horizontal blue lines indicate the average recall for the ND model. . . . 141

B.5 Precision-based interaction plot to check the joint effect of classifier, learning model and scenario. 142

B.6 Average precision values aggregated by decision rules. The horizontal red line indicates the average precision for the ND model. 143

B.7 Average precision as function of the partition size. The horizontal red lines indicate the average precision for the ND model. 144

B.8 Average precision as function of the partition size for each data set. The horizontal blue lines indicate the average precision for the ND model. . 145

List of tables

2.1 Empirical power for data simulated from the concentric circular Gaussian distributions. 21

2.2 Empirical power in experiments involving time series 34

2.3 Empirical nominal levels for the discrete data scenarios A and B, where A: $\eta = (1,2,3,4,5), \theta = 1$, B: $\eta = (1,2,3,4,5,6,7,8,9), \theta = 1$ 34

2.4 Empirical power in discrete scenarios. 35

2.5 Benchmark datasets 36

2.6 Empirical power in experiments with benchmark datasets using the interpoint Euclidean distance 38

2.7 Empirical power in experiments with benchmark datasets using the interpoint Chebyshev distance 39

3.1 Rejection rates at level $\alpha = 0.05$ for comparing two groups of time series simulated from AR(1) processes with autoregressive coefficients φ_X and φ_Y, respectively (Scenario S1), and the same models after standardization (Scenario S2). 51

3.2 Rejection rates at level $\alpha = 0.05$ in Scenario S3. T denotes the length of the series and “Norm.” indicates if the series were or not normalized. 53

3.3 Rejection rates at level $\alpha = 0.05$ in Scenario S4 involving real datasets from the UCR Time series archive. 54

3.4 Power results for comparing alcoholic and control subjects EEG readings. Nearest Neighbors, Energy and the proposed Distance Distribution methods are combined with Euclidean, Periodogram and CID distances to produce 9 testing approaches. 56

4.1 Classification accuracies in the UCR/UEA Time Series datasets. Out distance, dLAGDISTR, achieves or ties for the best distance in 11 out of the 43 datasets and has the highest average accuracy. 70

4.2 Classification accuracy of the distances in the simulation scenario . . . 72
4.3 Clustering results of the simulation scenario when the number of clusters is known. 73
4.4 Clustering results when the number of clusters is not known. It is searched using the Calinski-Harabasz index. 74

5.1 Features used in FFORMA framework. 80
5.2 M4 Competition: Point forecast error results. Benchmark is a combination of methods of the exponential smoothing family, simple, Holts and damped. 85
5.3 M4 Competition: Prediction Interval Results. 85
5.4 Average weights received by each indidividual forecast method under FFORMA in the M4 Reference Dataset. 87

6.1 Data sets characteristics 103
6.2 Distribution (in percentage) of normal activities and kinds of attacks in KDD Cup 99 data set. 104
6.3 Proportion of times that the smallest distributional distance leads to the best classifier node ($p_{\min(d_i)}$) against the test sample size (t). 107
6.4 Average classification accuracy values conditional on classifier type. Rows under the last sub-table (MEAN) show the averages over all trials, including balanced and unbalanced scenarios. The lack of results for ND in the first two sub-tables is due to the ND model assumes non-distributed data, i.e. no partitions (balanced or unbalanced) are considered. 109
6.5 Average classification accuracy values conditional on the decision rules. 110

B.1 Average recall values conditional on classifier type. 138
B.2 Average recall values conditional on the decision rules. 139
B.3 Average precision values conditional on classifier type. 141
B.4 Average precision values conditional on the decision rules. 142

Chapter 1

Introduction

This thesis presents contributions in supervised and unsupervised statistical learning, applied to complex data objects such as time series and shapes, among others. We study two-sample hypothesis testing, classification and clustering.

We use the term complex data to imply data objects for which general statistical methods fail to achieve satisfactory results when applied to them. The following example illustrates a hypothetical scenario for one of the statistical problems we address in this thesis.

> In an econometric setting we want to check whether firms in construction and firms in agriculture have different *behavior* during a period of time, according to one economic indicator. We have a sample of 20 time series from each group, depicted in Figure 1.1. We identify the question as a two-sample homogeneity test, since we think of the term *behavior* as the distribution function that generates the series in each sample, and time series are just multivariate observations. When we apply a standard multivariate two-sample test, such as an extension to the multivariate case of the Kolmogorov-Smirnov test [45], we find that it does not detect significant differences. This is puzzling because visually they look different, but still we would like to give a quantitative measure of their difference. After consulting a statistician, we model each series as a linear autoregressive process and then apply the same test, this time to the fitted coefficients, concluding clearly significant differences.

This example illustrates the following general scenario: We have a set of multivariate observations from a complex phenomenon and a statistical question about them. Standard multivariate tools, although when given enough data will be able to solve the

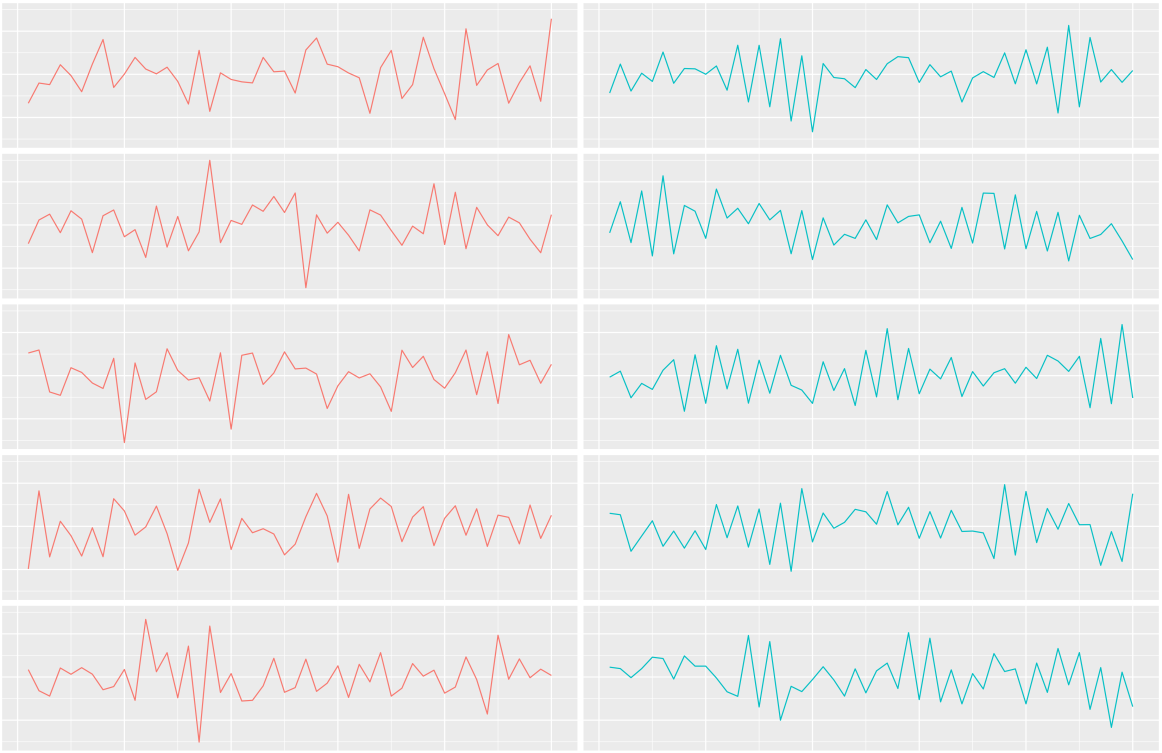


Fig. 1.1 Simulated scenario of linear autoregressive processes. When looking for differences between these two groups using a two-sample hypothesis test, we highlight the fact that taking the series as multivariate observations produces much less statistical power that the same test over the coefficients extracted from the series. While the left panel series have been generated from the AR(1) process defined by $X_t = -0.3X_{t-1} + \varepsilon_t$, the ones on the right panel come from the AR(2) process $X_t = -0.3X_{t-1} + 0.5X_{t-2} + \varepsilon_t$. In both cases, independent standard Gaussian innovations ε_t were used. Each series is of length 50 and all of them were normalized to have zero mean and unity variance.

problem, fail to achieve satisfactory results at realistic sample sizes. By introducing some "domain knowledge" about our underlying phenomenon, in this case just a simple assumption about the nature of their temporal behavior, we are able to drastically improve results. Quantitatively, in the scenario of Figure 1.1, the standard approach needs over 16 times the amount of data to achieve the same statistical power than the specialized approach, and over 250 times the amount of computing time. *There is clearly a need to develop tools to facilitate the introduction of domain knowledge.* We propose a two-sample test which is able to work with domain knowledge introduced in a wide range of ways and study how these ideas can be applied to time series in Chapers 2 and 3, respectively.

Other concrete examples for the major statistical tasks addressed in this thesis are:

- Classification: Identify the species of a fish from the shape (silhouette) of one of its bones, when a database of already identified bone shapes is available.
- Clustering: Grouping genes of a bacteria by how they vary their expression with time, after an alteration has been applied to the cell, such as adding an antibiotic to the substrate.

Distances for data analysis

One way of approaching these learning problems is by defining features (variables, attributes, characteristics) to be extracted from the data, and then apply standard multivariate methods over these features. In the above scenario about firms, using the fitted autoregressive coefficients is an example of feature extraction. When applicable, this is the preferred way, since it is often superior to alternatives in terms of interpretability of results and computational efficiency. However, in many scenarios we do not know which features to use. This approach is also limited in its generalization capability, a set of features that work well in one setting may not do so in another, even though they are related.

Another way of approaching learning with complex objects is through the use of similarities between them, this is, by introducing a function that assigns a numerical value to how similar two objects are. After we introduced this function, we apply a statistical procedure designed to work directly on object-to-object similarity comparisons, not the objects themselves.

This comparative function can take the form of a similarity, increasing when the two compared objects are more alike, or a dissimilarity or distance, which decreases instead of increasing the more alike two objects are. The use of one or the other forms is usually dictated by the context, and it is easy to get one from the other. Following this distance + method paradigm, the above example of classification could be approached by using a elastic measure (explained later) as the distance function and the nearest neighbors algorithm as the algorithm defined on distances between objects, i.e. "Assign new objects to the family of the most similar among the already classified, according to this notion of similarity"

Distance-based approaches are interesting because they address limitations of feature-based approaches. The most critical limitation being that in some domains, we can find a suitable distance but not a set of features.

The use of domain-specific distances has multiple interpretations in addition to the straightforward, one of them is as a form of feature extraction. Many of the distances that have been proposed for comparing complex objects are indeed a combination of two steps: feature extraction and then Euclidean distance between the features. An example in the time series setting is to extract linear autoregressive coefficients from the series and use Euclidean distance between vectors of coefficients afterwards. This distance was proposed by Piccolo in 1990 [101]. One could argue that there is little benefit in considering a distance instead of just using the AR coefficients directly as the set features and work from there with a feature-based approach. The distance of an object to another fixed reference object, called "landmark", is also a feature. However, distance-based approaches are more general than feature extraction, allowing us to get truly non-parametric: we can use a distance for which the feature-based equivalent would require an infinite set of features, therefore losing any advantage in interpretability, computational complexity or even memory complexity/storage space.

We will show an example of a distance between time series which does not have a practical feature-based equivalent, unlike the distance by Piccolo. This distance is called Time Warp Edit Distance (TWED), was proposed by Marteau in 2009 [90], and improved over the state-of-the-art methods in time series classification at the time. This included simultaneously improving classification accuracy on non related problems such as the aforementioned example of fish family identification, and others such as cardiopathy detection based on electrocardiogram readings. TWED belongs to a family of distances called "elastic measures" [80], based on the idea that the similarity between two time-evolving phenomena should not be affected by accelerations or decceleration in these phenomenons. The original domain of application of this familiy is speech recognitition [112] with a distance called Dynamic Time Warping (DTW). TWED is defined intuitively between two given series, lets say series A and B, by allowing modifications in these two series until they become equal. Roughly speaking, we can

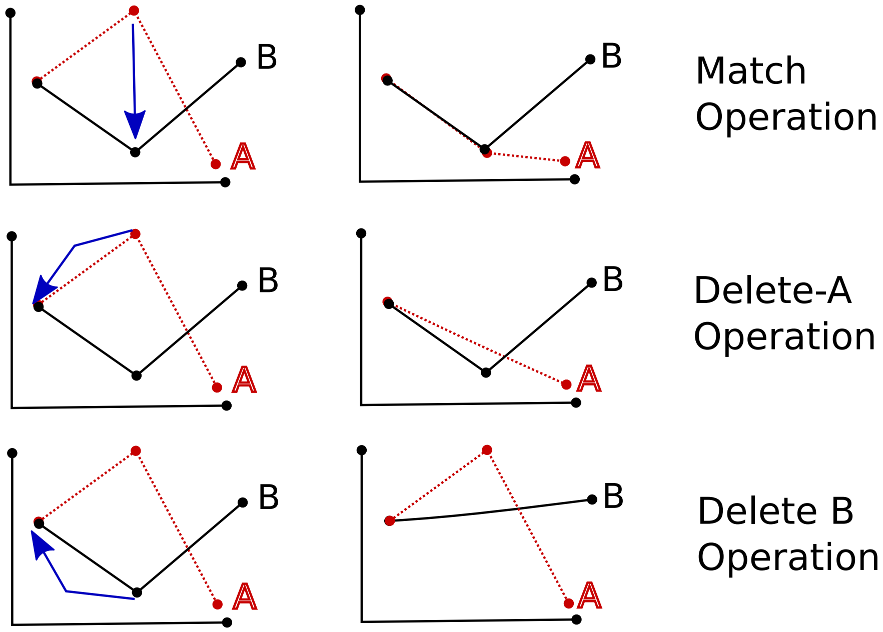


Fig. 1.2 Algorithmic definition of the TWED distance between two time series. Three different operations can be applied to the observations of the time series, in order. Each operation has a cost associated and the distance is defined as the minimum possible cost to make the two time series match exactly. This is an example of a distance that can be defined algorithmically but does not have a feature based equivalent.

remove points in A, in B, or move them until they match, as depicted in Figure 1.2. Moreover, the operations must be in the order of the arrow of time, once an operation has been applied to a series we must move to the next time step and cannot go back in time. Each operation has an associated cost, and the distance is defined by the minimal cost required to completely superimpose the two series.

We can define TWED algorithmically, but not as a set of features to be extracted from the series. The “domain knowledge” introduced by TWED has no feature based equivalent.

The distances also have a bayesian intepretation. In the Gaussian Processes modeling framwork, distances may be used to define the covariance function. Thus, a distance may help define a prior distribution over the functions that fit the data. A DTW based covariance function was proposed in [119] to identify iron ore deposits signatures using Gaussian Processes.

The benefits of introducing appropiate distances extend beyond the traditional objectives of statistical learning and can be used for:

- Information retrieval: Search in a database for objects similar to a given one, possibly ranking the relevance of the findings. A colourful example is music retrieval, where we want to find songs “similar” to the input one, or different interpretations of the same musical piece. Foster et al. [43] explore similarity measures based on information theory applied to this task. They also argue for the possiblity of extending the applicability of their studied measures to other domains.

- Anomaly detection: An umbrella term for detecting patterns such as outliers, suprises or novelties, aberrations, etc. Chandola et al. [25] conclude that distance-based approaches are the best alternative for anomaly detection in an empirical study involving 19 datasets and 9 techniques.
- Visualization / Summarization: Pairwise distances can be used coupled with projection/dimensionality reduction techniques, such as multidimensional scaling, to generate visualizations of samples of complex objects. These techniques work based on the fact that a suitable distance usually induces a lower-dimensional space than the raw observation space. Single objects such as time series may have thousands of dimensions. A recent example of this approach can be seen in [26].

Distances arise naturally in data analysis. In ecology, Jaccard proposed in 1901 to analyze the alpine ecosystem by comparing relative populations of flora between all pairs of areas that define the Alpine region [66]. This distance is now being used much more generally, to compare objects which are themselves sets.

These examples attest for the reliance on distances. When we define a distance for a specific scenario, such as clustering time series, there is a good chance that this distance can be used again to solve other tasks in the same domain, and even other domains.

Distance-based methods

There is a bi-directional relationship between distances and the learning methods that can leverage them: We want to define distances that can be used with existing methods, and we want to define new methods that can accept a wide range of distances.

Consider the K-means clustering algorithm. It is defined for the Euclidean distance: the arithmetic mean of a set of observations minimizes the average Euclidean distance to the members of the set. When one naively substitutes the Euclidean distance for a domain-specific one but keeps using the arithmetic mean to generate the centers, the results are likely to be of little utility. This happens because the cluster centers created by the arithmetic mean, are centers in the raw data space and not in the feature space induced by the distance, intragroup average distance is not being minimized. The *mean of the AR coefficients of a set of time series* is different to the *AR coefficients of the mean of the set of time series*, and for clustering we usually want the former, not the latter. This fact limits the applicability of the K-means algorithm, it can not work with arbitrary measures of similarity. In order to apply K-means with arbitrary distances, we need to provide a way of computing the center (usually called barycenter) of a group of objects, i.e. the object that minimizes the average distance to the group. This is a complex problem so the usability of K-means is effectively reduced. A barycenter-calculating method for DTW, one of the most widely used distances in time series, was proposed in [100], 17 years after DTW was introduced to the data mining community [15]. There is also no clear best way for computing these barycenters for DTW, it is still a subject of research [95].

There are clustering algorithms which can be used with arbitrary distances, such as Partitioning Against Medoids, similar to K-means but avoiding using centers, or hierachical clustering.

Most distance-based learning methods have limitations on the distance they are able to use, such as requiring additional properties that distances do not fulfil in general. Perhaphs the most common of this requirement is the triangular inequality, in other words, they need metrics. The energy family of statistics, used for two-sample and independence testing, requires its distances to be conditionally negative definite [129].

To fully benefit from the domain knowledge that may be introduced via a distance, it is therefore of special interest to define methods that can work with a wide range of them.

Limitations of distance-based methods

- Computational Complexity: The requirement of computing all parwise distances between the objects of the dataset induces quadratic time complexity, which renders these approaches inapplicable to very large datasets. On the other hand, pairwise distance calculations are easy to parallelize, and finding efficient approximate methods is an active field of research, e.g. a review of approximations for nearest neighbor search can be seen in [142].

- Rigidness compared to data-driven approaches: Both distance and feature-based approaches, while very powerful, can be outperformed by data-driven approaches when sufficient data is available. Data-driven approaches can be

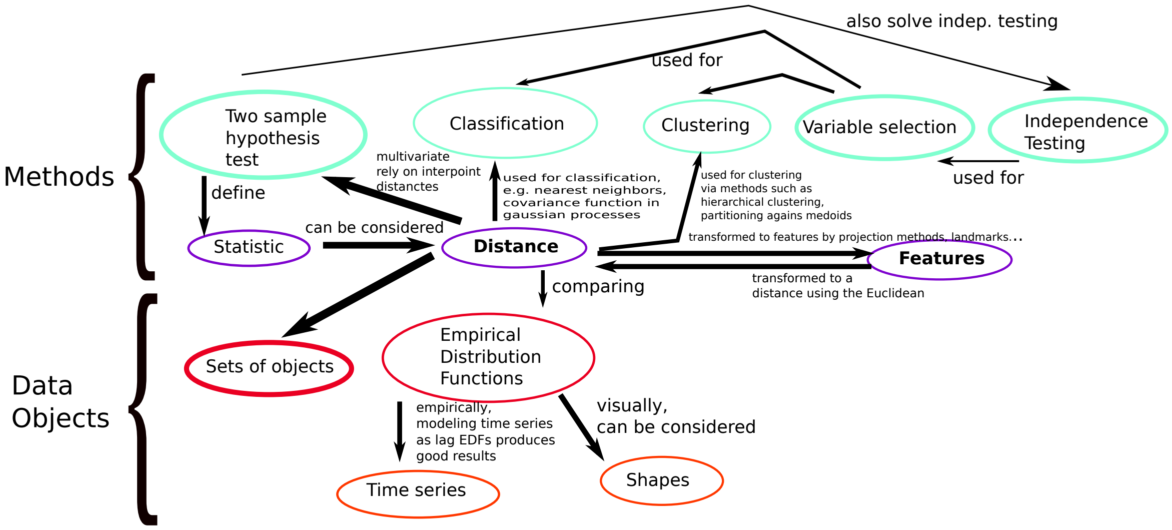

also solve indep. testing
used for
Methods
Two sample hypothesis test
Classification
Clustering
Variable selection
Independence Testing
multivariate rely on interpoint distanctes
used for classification, e.g. nearest neighbors, covariance function in gaussian processes
used for clustering via methods such as hierarchical clustering, partitioning agains medoids
used for
define
can be considered
Statistic
Distance
transformed to features by projection methods, landmarks...
Features
comparing
transformed to a distance using the Euclidean
Data Objects
Sets of objects
Empirical Distribution Functions
empirically, modeling time series as lag EDFs produces good results
visually, can be considered
Time series
Shapes

Fig. 1.3 Map of learning algorithms

seen as "automatically" constructing the set of features or distance metric to use based on the available data, instead of introduced by the expert to express some "domain knowledge". Image classification is a clear example of this effect, the yearly ImageNet challenge [111], was dominated in 2010 by feature-based methods and SVM, but it has since been won by data-driven neural network approaches. It can be argued that "domain knowledge" is still being introduced by humans in the form of network architecture and other hyperparameters.

A map of distance-based learning (Thesis structure)

The contributions of this thesis follow from two main intertwined ideas:

- Distances are a useful vehicle for introducing domain knowledge about complex objects to statistical problems. It is interesting to define both new distances between complex objects (such as time series) as well as statistical methods that work well with a variety of distances.
- Some two sample testing statistics can be considered distances between empirical distribution functions, so they can be used as the base to define distances between objects such as sets of multivariate observations.

Figure 1.3 shows a visual representation of how the topics of this thesis are related and how they fit in a larger context of statistical learning tasks. This figure also highlights how having a good performing distance may be used for solving a range of problems in the domain. We start from **Chapter 2**, a distance-based two-sample hypothesis test. This is a general procedure, it works with multivariate observations and a wide range of distances. In addition to the wide range of supported distances thanks to the light assumptions required from them, it shows good performance on the kinds of distributional differences that may be induced by the distances. All the methods we analyzed in our experiments are good against either location or scale differences, a fact that has already been highlighted in [27], but our method achieved competitive results on both types of differences. When we introduce a distance to take advantage of some “domain knowledge”, we can interpret it as inducing a different feature space. The distributions of the two samples in this new feature space may differ in several ways (mean, scale, etc.), therefore is of special interest that a test has good performance all-around and is not too focused on one kind of alternative.

We study the test in the context of time series in **Chapter 3**, combining it with some well-known time series dissimilarities originated from the time series clustering literature. We show that the introduction of these distances drastically improves statistical power in most scenarios, for all the studied two sample tests, compared to the baseline Euclidean distance. When comparing tests, the one we propose also performs better on average than the alternatives.

The relevance and range of application of two-sample test statistics is expanded by realizing that some of these statistics are also a form of distances, measuring dissimilarity between sets of objects or distribution functions. They can be used as distances for other statistical tasks such as classification and clustering of these kinds of objects. Shape objects and time series have been effectively modeled as distribution functions for information retrieval, classification and clustering [98, 113]. Previous works have been modeling shapes and time series as univariate distributions, often applying strong dimensionality reduction techniques to achieve this unidimensionality. Coupled with simple dissimilarities for histograms they achieved state-of-the art results. In **Chapter 4** we leverage modern two-sample statistics to lift the unidimensionality requirement. We propose a distance for time series and shapes based on modeling them as multivariate distributions, using lag embedding representations of these objects. As a distance, it achives excellent empirical results, while arguably being more parsimonious than the alternatives.

Two sample tests are also useful for testing independence. Testing independence between variables is an objective in itself, but is also used for variable selection in classification. Variable selection for classification using two sample testing is perfomed by finding a set of variables that are independent, called *filter* methods [19]. While our test can be applied directly to independence testing and variable selection, we do not explore its performance in this thesis.

Chapter 5 presents a meta-learning algorithm for time series forecasting. The scenario is forecasting a large dataset of time series, and the idea is to improve forecasting accuracy by exploiting information of the whole dataset, instead of treating each series individually. Our approach is similar to a time series classification problem, where we want predict which method is the best for forecasting a given time series. Instead of selecting the best method, we train our model to produce weights which then are used to average the forecasts of all the methods that are being considered. The method is based on extracting time series features, may of them have also been used in distances between time series.

In **Chapter 6** we apply the notion of similarity between probability distributions via two-sample test statistics to the context of classification when the dataset is partitioned in subsets which cannot be merged. This restriction may come into play for numerous reasons, each subset may reside in computing nodes located in different physical locations and the dataset may be too big to store in only one machine, communication between nodes may be too costly. Privacy or regulations may impede nodes to share the data, such as hospitals not able to share their patients records but that still want to develop some predicitive model. In our approach, classification models are fitted to each subset and their predictions are incorporated in a weighted voting system, where the weights are dependent on how similar each training subset is to the test dataset.

Chapter 2

Two-sample homogeneity testing: A procedure based on comparing distributions of interpoint distances

2.1 Introduction

A classical problem in statistical inference consists in checking whether two independent samples have been generated from the same probability distribution. This problem, commonly referred to as two-sample homogeneity testing problem, has been extensively studied, particularly in the univariate setting. Traditional nonparametric approaches to address two-sample homogeneity include the comparison of empirical distributions (two-sample Kolmogorov-Smirnov and Cramér-von Mises tests) and distribution free tests based on runs (Wald-Wolfowitz test) and ranks (Wilcoxon-Mann-Whitney test). Nevertheless, extending these procedures to deal with multivariate data is not simple. Unlike the univariate two-sample problem, most of the extensions to the multivariate setting are no longer distribution free. Even in the parametric setting, natural extensions of univariate tests to check equality of mean vectors (Hotelling's T^2 test) and variance-covariance homogeneity (e.g. Box test) are very sensitive to departures from distributional assumptions and show a substantial power loss as dimension increases [7]. Despite these difficulties, with the steadily growing availability of huge databases, developing efficient inference procedures for large dimensions is still a major challenge requiring further research.

A useful approach to address the comparison of probability distributions in large dimensions is to consider interpoint distances, see e.g. the early work by Bickel and

Breiman [17]. The distances (commonly the Euclidean distance) between all possible pairs of points in the data are calculated, and then these pairwise distances are used to discriminate between the samples. In addition to reducing the dimension of the problem, distance-based approaches exhibit appealing properties to be used in multivariate two-sample testing. First, they are not limited to dealing with continuous data since there is a variety of distances or dissimilarities for different kinds of data objects, including categorical data [34], time series objects [94], functional data [42], and also complex structured data such as graphs [51] and image data [36]. Moreover, whenever interpoint distances are available, the two-sample test can be tackled even though the original observations are not accessible. Versatility to choose the proper distance facilitates the introduction of prior domain knowledge such as invariance against rotation or shift effects in shapes [57]. For instance, it is well-known that some metrics are preferred to others when clustering functional data depending on the type of curves at hand [42] or when clustering time series data according to the clustering purpose [94]. In a similar way, it is intuitively expected that employing a suitable distance to define the statistic test in two-sample problems should increase the test power.

In this chapter, a new test statistic using interpoint distances is proposed to address the multivariate two-sample homogeneity problem. The idea is to compare the empirical distributions of both the intra-group and inter-group pairwise distances by using a Cramér-von Mises-type statistic and approximate the critical values by means of a permutation procedure. Maa et al. [85] establish conditions under which testing for equal distributions of these pairwise distances is equivalent to testing for the equality of the original distributions, thus providing theoretical support to ensure the consistency of the proposed test. Our proposal intends to take advantage of comparing the whole empirical distributions of interpoint distances instead of just a few moments. Based on this intuition, it is expected that the proposed test is more powerful for detecting alternatives based on differences in shape or other relevant distributional feature. Like other distance-based tests, our test is not restricted to continuous data. In fact, regarding the mild regularity conditions required in the results by Maa et al. [85], the proposed test applies to a broad range of dissimilarities in both discrete and continuous cases, including some popular distances not considered by other alternative test statistics. The asymptotic analysis of the proposed test is carried out within the classical setting of fixing the dimensionality d while letting the sample size n tend to infinity, i.e. the asymptotic power of the test as both n and d tend to infinity is not studied.

An extensive Monte Carlo power study shows the good performance of the proposed test compared to other popular distance-based tests. Overall, satisfactory results are reached in all simulated scenarios, including very different types of alternative distributions. In particular, it is highly competitive in power in HDLSS (High Dimension, Low Sample Size) scenarios.

2.1.1 Two-sample homogeneity test

The two-sample homogeneity testing problem can be formally stated as follows. Let $\Xi_X \equiv \{\mathbf{X}_1, \ldots, \mathbf{X}_n\}$ and $\Xi_Y \equiv \{\mathbf{Y}_1, \ldots, \mathbf{Y}_m\}$ be independent random samples of $\mathbb{R}^d$-valued random vectors, $d \geq 1$, with multivariate distributions F_X and F_Y, respectively. Then, the two-sample homogeneity problem consists in testing the equality of the distributions

$$H_0 : F_X = F_Y \quad \text{versus} \quad H_1 : F_X \neq F_Y. \tag{2.1}$$

Note that no assumptions on the kind of the distributions are imposed, and in particular they are not restricted to belonging to a specific family indexed by a finite dimensional parameter. Thus, the problem is nonparametric in nature.

2.1.2 Related work

Recent approaches for testing (2.1) involve statistics regarding interpoint distances. Following the works by Wei et al. [143] and Chen and Friedman [27], these approaches could be loosely classified into three categories, namely graph-based methods, projection-based methods and statistics comparing within- and between-sample interpoint distances.

Graph-based. Methods in this category rely on properties of the graph created when connecting all the observations (nodes) using interpoint distances as weighted non-directional edges. Friedman and Rafsky [45] proposed to count the edges connecting nodes from different samples in the minimum spanning tree of the graph so that the null is rejected if the number of between-sample edges is significantly small. Another approach in this category consists in using k-nearest neighbor (k-NN) graphs [58, 59, 115]. The test statistics evaluates, for all observations, the proportion of neighbours belonging to the same sample, rejecting the null when these proportions are large. The nearest neighbor tests are both consistent and asymptotically distribution-free under the null, but they require continuity of the distributions. Hall and Tajvidi [53] propose a method related to the nearest neighbors which is valid for infinite dimensional Euclidean spaces and can take any dissimilarity function as distance. More recently, Biswas et al. [18] propose counting the number of edges connecting different samples in the shortest Hamiltonian path of the graph.

Projection-based. The aim is to project the data onto a lower dimensional space where traditional statistics can be applied. Ghosh and Biswas [47] and Wei et al. [143] propose to use the projection direction learned from a linear classifier such as support vector machine (SVM) or distance-weighted discrimination (DWD) [89]. The method in Ghosh and Biswas [47] achieves the distribution-free property by splitting the data into training and testing sets so that the direction learned by the classifier is independent from the testing data, and then applying a distribution-free univariate test to the projected data. The method in [143], DiProPerm, is not distribution-free but uses the whole available data for training the classifier. A univariate statistic such as the t-statistic is calculated on the projected data and its null distribution is approximated by repeating the whole process of training the projection and calculating the statistic on permutations of the data labels. The projection approach has also been applied to checking for equality of means for two multivariate normal distributions [84, 132, 123].

Based on within and between-sample interpoint comparisons. Székely and Rizzo [127] and Baringhaus and Franz [10] propose to test the equality of multivariate distributions by comparing averages of interpoint Euclidean distances evaluated within and between samples. Large values of the statistic suggest that the samples come from different distributions. The test statistic is called *energy* statistic due to Newton's potential energy. In the univariate case, the energy statistic is equivalent to the L^2 distance between empirical distribution functions [10] and it can be seen as the treatment sum of squares in a ANOVA interpretation of the problem [108]. Tests based on the empirical multivariate characteristic functions introduced by Alba-Fernández et al. [6] and by Hušková and Meintanis [62] have also been linked to the energy statistic [62].

From the machine learning community, Gretton et al. [50] proposed a two-sample test statistic called Maximum Mean Discrepancy (MMD) based on evaluating the difference between embeddings of the distributions into reproducing kernel Hilbert spaces (RKHS) . A proper choice of the kernel used to perform the embeddings is crucial for obtaining a powerful test. Sejdinovic et al. [116] analyze the link between the distance- and the kernel-based approaches by considering the close relation between negative-type semimetrics and symmetric positive definite kernels, and in particular the equivalence between the energy and MMD statistics is studied.

The energy and MMD statistics have been widely studied because of their desirable properties such as simplicity, straightforward implementation and nice interpretation in terms of generalization of previously proposed statistics. Their computational complexity is dominated by the computation of interpoint distances or kernels $O((n+m)^2)$ (as opposed to, for instance, computing nearest neighbors $O((n+m)^2 \log(n+m))$. A linear time variant of the MMD has been proposed in [51] and the power-computational cost tradeoff has been studied in [105].

The procedure proposed in this chapter also falls in this category of methods comparing interpoint distances within and between-samples. However, unlike the energy and MMD methods, our test statistic compares the whole distributions of these distances instead of a few specific moments.

2.1.3 Overview

The rest of the chapter is structured as follows. Our approach is presented in detail in Section 2.2. The test statistic is formally introduced and motivated by illustrative examples highlighting the main differences with other approaches based on interpoint distances such as the energy statistic. Regularity conditions required to satisfy the results by Maa et al. [85] are also established, thus giving theoretical support for the consistency of the procedure. Section 2.3 is devoted to showing results from a broad Monte Carlo power study comparing the proposed method to other procedures representative of each of the aforementioned categories. Sensitivity of the tests is examined for different types of data and against a range of alternatives, including changes in location, scale, symmetry, kurtosis and particularly hard two-sample scenarios. Further, the effect of increasing the dimension is assessed and some benchmark datasets are used to remark the usefulness of our approach. Lastly, the main conclusions are summarized in Section 2.4.

2.2 Methodology

2.2.1 The test statistic

Our test statistic is based on pairwise distances between sample points, and hence a symmetric real-valued non-negative function $D(x,y)$ defined on $\mathbb{R}^d \times \mathbb{R}^d$ must be employed to compute these distances. Although the function D will be required to fulfill mild regularity conditions later on (see Lemma 1), D does not need to satisfy the triangle inequality. Based on both the distance D and the random samples Ξ_X and Ξ_Y, the following sets of pairwise distances are considered.

(a) The sets denoted by $\mathcal{W}_X = \{\omega_{ij} = D(\mathbf{X}_i, \mathbf{X}_j)\,;\, i,j = 1,\ldots,n;\, i < j\}$ and $\mathcal{W}_Y = \{\omega_{ij} = D(\mathbf{Y}_{i-n}, \mathbf{Y}_{j-n})\,;\, i,j = n+1,\ldots,n+m;\, i < j\}$ gather the distances between points within the samples Ξ_X and Ξ_Y, respectively.

(b) The set $\mathcal{W} = \mathcal{W}_X \cup \mathcal{W}_Y = \{\omega_{ij};\, i,j = 1,\ldots,n+m;\, i < j\}$ gathers together all the within-sample distances.

(c) The set denoted by $\mathcal{B} = \{b_{ij} = D(\mathbf{X}_i, \mathbf{Y}_j)\,;\ i = 1,\ldots,n;\, j = 1,\ldots,m\}$ groups the pairwise distances between elements from different samples.

Thus, $\mathcal{W}_X$, $\mathcal{W}_Y$, $\mathcal{W}$ and $\mathcal{B}$ are formed by realizations of random variables with univariate probability distribution functions denoted by $F_{\mathcal{W}_X}$, $F_{\mathcal{W}_Y}$, $F_{\mathcal{W}}$ and $F_{\mathcal{B}}$, respectively. Maa et al. [85] establish conditions under which testing for the equality of the univariate distributions $F_{\mathcal{W}_X}$, $F_{\mathcal{W}_Y}$ and $F_{\mathcal{B}}$ is equivalent to testing for the equality of the original multivariate distributions F_X and F_Y. Therefore, a way of approaching the multivariate two-sample homogeneity testing problem (2.1) as a univariate two-sample problem is as follows.

Let $\widehat{F}_{N,\mathcal{W}}$ and $\widehat{F}_{M,\mathcal{B}}$ be the usual empirical cumulative distribution functions (ecdf) based on the $N = n(n+1)/2 + m(m+1)/2$ and $M = nm$ elements forming $\mathcal{W}$ and $\mathcal{B}$, respectively. Then, the Cramér-von Mises statistic for testing

$$H_0' : F_{\mathcal{W}} = F_{\mathcal{B}} \quad \text{versus} \quad H_1' : F_{\mathcal{W}} \neq F_{\mathcal{B}} \tag{2.2}$$

is given by

$$T_{n,m}(u) = \frac{NM}{N+M} \int_{-\infty}^{\infty} \left(\widehat{F}_{N,\mathcal{W}}(u) - \widehat{F}_{M,\mathcal{B}}(u) \right)^2 d\widehat{H}_{N+M}(u), \tag{2.3}$$

where $\widehat{H}_{N+M}(u)$ is the ecdf associated with the pooled sample $\mathcal{W} \cup \mathcal{B}$, i.e. $(N+M)\widehat{H}_{N+M}(u) = N\widehat{F}_{N,\mathcal{W}}(u) + M\widehat{F}_{M,\mathcal{B}}(u)$, and the test criterion is to reject H_0' for large values of $T_{n,m}$.

At the standard framework of empirical cumulative distribution functions based on i.i.d. observations, the $T_{n,m}$ statistic becomes asymptotically distribution free under the null hypothesis [see e.g 71]. In order to obtain tables for using the test statistic at some conventional significance levels for small sample sizes, Anderson [8] established an alternative form based on ranks for $T_{n,m}$ as follows. Consider the pooled sample $\mathcal{W} \cup \mathcal{B}$ with all the ordered interpoint distances, and let $r_1, \ldots, r_M$ and $s_1, \ldots, s_N$ be the ranks in this ordered set of the elements in $\mathcal{B}$ and $\mathcal{W}$, respectively. Then

$$T_{n,m}(u) = \frac{U}{NM(N+M)} - \frac{4NM-1}{6(N+M)}, \tag{2.4}$$

with

$$U = M\sum_{i=1}^{M}(r_i - i)^2 + N\sum_{j=1}^{N}(s_i - i)^2.$$

Our proposal consists of selecting a proper distance D, constructing the sets $\mathcal{W}_X$, $\mathcal{W}_Y$, $\mathcal{W}$ and $\mathcal{B}$ based on D and the original multivariate samples, and then using the Cramér-von Mises statistic $T_{n,m}$ given in (2.4) to testing $F_{\mathcal{W}} = F_{\mathcal{B}}$, which is equivalent to testing $F_X = F_Y$ under mild constraints on D as it is shown in Section 2.2.2. The critical values of the test are approximated by using the traditional permutation approach. A large number B of random permutations of the samples Ξ_X and Ξ_Y are generated, $T_{n,m}$ is computed for each permuted sample, and the critical value is determined by the corresponding critical value of the permutational distribution. Since the procedure is based on comparing distributions of distances, hereafter we refer to the proposed test statistic as the DD statistic.

2.2.2 Theoretical foundation

First, we show that testing H_0: $F_X = F_Y$ is equivalent to testing H_0': $F_{\mathcal{W}} = F_{\mathcal{B}}$, thus reducing the dimensionality of the problem. To do this, the Theorem 2 by Maa et al. [85] is properly adjusted in Lemma 1 below.

Lemma 1. *Let* $\mathbf{X}$ *and* $\mathbf{Y}$ *be independent d-dimensional random vectors with continuous distribution functions* F_X *and* F_Y*, respectively. Assume that the corresponding density functions* f_X *and* f_Y *satisfy*

(A1) $\int_{\mathbb{R}^d} f_X^2(\mathbf{u})\, d\mathbf{u} < \infty$ *and* $\int_{\mathbb{R}^d} f_Y^2(\mathbf{v})\, d\mathbf{v} < \infty$.

(A2) The function $m(\mathbf{v}) = \int_{\mathbb{R}^d} f_Y(\mathbf{u}+\mathbf{v}) f_X(\mathbf{u})\, d(\mathbf{u})$ *has a Lebesgue point in* $\mathbf{0}$.

Let $D : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ *be a nonnegative continuous function such that*

(A3) $D(\mathbf{u},\mathbf{v}) = 0$ *if and only if* $\mathbf{u} = \mathbf{v}$.

(A4) $D(a\mathbf{u}+\mathbf{w}, a\mathbf{v}+\mathbf{w}) = |a|\, D(\mathbf{u},\mathbf{v})$*, for all* $a \in \mathbb{R}$ *and* $\mathbf{w} \in \mathbb{R}^d$.

Then, if $\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3$ *are iid random vectors with distribution* F_X*,* $\mathbf{Y}_1, \mathbf{Y}_2, \mathbf{Y}_3$ *are iid random vectors with distribution* F_Y*, and the* $\mathbf{X}$*'s and* $\mathbf{Y}$*'s are independent, it holds* *that*

$$F_X = F_Y \quad \text{if and only if} \quad F_{X_1Y_1} = \frac{1}{2}\left(F_{X_2X_3} + F_{Y_2Y_3}\right),$$

where $F_{X_1Y_1}$, $F_{X_2X_3}$ and $F_{Y_2Y_3}$ denote the distribution functions of the univariate random variables $D(\mathbf{X}_1,\mathbf{Y}_1)$, $D(\mathbf{X}_2,\mathbf{X}_3)$, and $D(\mathbf{Y}_2,\mathbf{Y}_3)$, respectively.

Note that conditions (A1) and (A2) are non-restrictive regularity requirements for the underlying probability densities f_X and f_Y. While (A1) ensures the existence of second-order moments, (A2) is a technical requirement which holds if, for example, one of the densities is bounded or continuous, see Remark 1 in [85]. Conditions (A3) and (A4) for the function D are also satisfied for a wide range of distances, including whatever function of the Euclidean metric.

Maa et al. [85] claim that their results do not require the independence of all the interpoint distances (which is also true for Lemma 1). This is particularly useful in our framework since the interpoint distances forming $\mathcal{W}$ and $\mathcal{B}$ are calculated using the same sample points so that the independence assumption is not fulfilled. Nevertheless, by treating with dependent data, the null asymptotic distribution of the Cramér-von Mises statistic involves complex quantities depending on the dependence structure of the interpoint distances via sums of covariances (see e.g. [118] or [88]). Overall, it is cumbersome to establish specific approximations for these quantities, which accounts for determining the critical values of the test using a permutation or bootstrap procedure. In any case, we have explored conditions ensuring the consistency under the null of the proposed statistic based on the samples of interpoint distances, such as is established in Theorem 1 below.

Theorem 1. *Let $\mathbf{X}_1,\ldots,\mathbf{X}_n$ and $\mathbf{Y}_1,\ldots,\mathbf{Y}_m$ be independent random samples of d-dimensional random vectors $\mathbf{X}$ and $\mathbf{Y}$ with respective continuous distribution functions F_X and F_Y. Let $D:\mathbb{R}^d\times\mathbb{R}^d\to\mathbb{R}$ be a symmetric function having bounded support $[0,D_{Max}]$. If conditions (A1)-(A4) in Lemma 1 hold and H_0: $F_X=F_Y\ (=F)$ is true, then, as $\min(n,m)\to\infty$, the statistic $T_{n,m}$ converges weakly to the distribution of*

$$T = \int \left(\mathcal{G}_{\mathcal{W}}(u) - \mathcal{G}_{\mathcal{B}}(u)\right)^2 dF_{\mathcal{D}}(u), \tag{2.5}$$

where $F_{\mathcal{D}}$ denotes the common distribution of the interpoint distances within ($F_{\mathcal{W}}$) and between samples ($F_{\mathcal{B}}$), and $\mathcal{G}_{\mathcal{W}}$ and $\mathcal{G}_{\mathcal{B}}$ are Gaussian processes with zero mean and covariance function $C_{\mathcal{D}}(\cdot,\cdot)$ given by

$$C_{\mathcal{D}}(u,v) = 4\left[P\left(D(\mathbf{Z}_i,\mathbf{Z}_j)\le u, D(\mathbf{Z}_s,\mathbf{Z}_t)\le v\right) - F_{\mathcal{D}}(u)F_{\mathcal{D}}(v)\right],$$

where $\mathbf{Z}_i$, $\mathbf{Z}_j$, $\mathbf{Z}_s$ and $\mathbf{Z}_t$ are distributed according to F, for pairs of indexes (i,j) and (s,t) having only one element in common.

Nevertheless, it is actually very complex to obtain the specific quantities determining the covariance structure of the limiting distribution. For this reason, a standard permutation test is proposed to obtain the critical values. Random permutations are performed on the pooled original sample, which is formed by i.i.d. observations under the null hypothesis. Hence, the observations are exchangeable, i.e. the joint distribution remains invariant under all possible rearrangements of the subscripts. As consequence of it, irrespective of the unknown distributions, the critical value obtained from the permutational distribution corresponds to the exact significance level (see e.g. Good [48]).

The proofs of Lemma 1 and Theorem 1 are provided in the Appendix A.

2.2.3 Motivation

This section is devoted to shed some light on the usefulness of a two-sample method comparing distributions of interpoint distances.

Like the popular energy statistic [127], the proposed DD statistic is based on calculating all the interpoint Euclidean distances and splitting them into two groups, distances between samples and distances within samples. However, while DD compares the whole distributions in both groups, energy evaluates the difference between their means. More specifically, if $\|\cdot\|$ denotes the Euclidean norm, then the energy statistic takes the form

$$\mathrm{E} = \frac{nm}{n+m}\left(\frac{2}{nm}\sum_{i=1}^{n}\sum_{j=1}^{m}\|\mathbf{X}_i-\mathbf{Y}_j\| - \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\|\mathbf{X}_i-\mathbf{X}_j\| - \frac{1}{m^2}\sum_{i=1}^{m}\sum_{j=1}^{m}\|\mathbf{Y}_i-\mathbf{Y}_j\|\right). \tag{2.6}$$

Other interesting approach comparing moments from these interpoint distance groups is provided by the maximum mean discrepancy (MMD) test. The MMD test statistic adopts the kernel-based approach and compares the mean embeddings of the multivariate distributions into reproducing kernel Hilbert spaces (RKHS) [50]. More formally, given a bounded continuous kernel κ, MMD is given by

$$\mathrm{MMD}^2 = \frac{nm}{n+m}\left(\frac{2}{nm}\sum_{i=1}^{n}\sum_{j=1}^{m}\kappa(\mathbf{X}_i-\mathbf{Y}_j)\right.$$
$$\left. -\frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\kappa(\mathbf{X}_i-\mathbf{X}_j)-\frac{1}{m^2}\sum_{i=1}^{m}\sum_{j=1}^{m}\kappa(\mathbf{Y}_i-\mathbf{Y}_j)\right). \tag{2.7}$$

Our intention is to get insight into the behavior of energy and MMD compared to DD by means of a motivating example. For illustrative purposes, consider samples drawn from two von Mises-Fisher distributions centered at $(0,1)$, one of them scaled, producing two concentric circumferences such as the ones depicted in Figure 2.1.

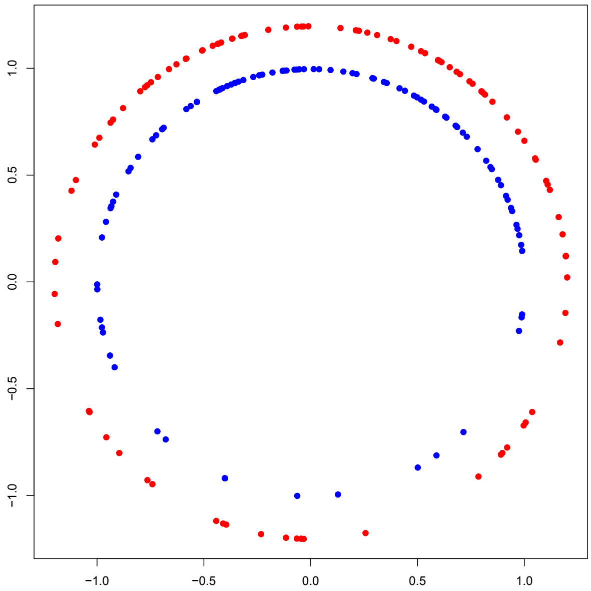


Fig. 2.1 Two samples, originally simulated from the von Mises-Fisher distribution and then scaled to produce the two concentric “circles”.

We have generated large samples and built up the sets of interpoint Euclidean distances between and within samples. The means (indicated by vertical lines) and density estimates for these sets of distances are shown in Figure 2.2a. The Gaussian kernel with median window was also used to observe the RKHS-distance between the mean embeddings of both distributions. The means and density estimates for the values of the kernel evaluated on pairs of elements from the same and different samples are shown in Figure 2.2b.

Both panels in Figure 2.2 show that the densities between and within-samples project very similar means, but their shapes are arguably more different. Therefore, it is expected that DD exhibits higher capability to distinguish the original distributions than energy and MMD statistics when the Euclidean distances and the kernel approach are respectively considered. For instance, the empirical power attained with these test statistics on T=2000 simulated trials using sample sizes $n = m = 40$ and a nominal level $\alpha = 0.05$ are provided in Table 2.1.

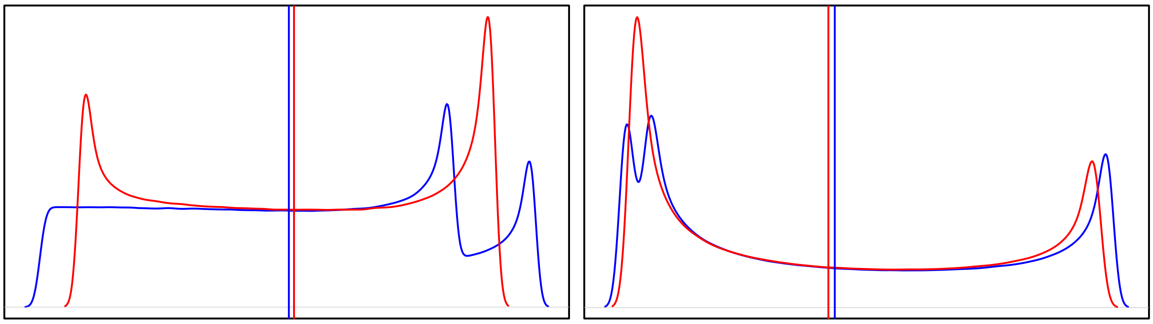
Energy	DD	MMD	DD (based on kernel approach)
0.19	0.8	0.18	0.79

Table 2.1 Empirical power for data simulated from the concentric circular Gaussian distributions.

Results in Table 2.1 fairly reveal that the distributional counterparts given by the DD statistic achieve greater power, thus confirming the intuition behind Figure 2.2. We argue that this effect, to a greater or lesser extent, can happen in practice in many situations. More examples will be seen in Section 2.3.

2.3 Experiments

A set of experiments was conducted to evaluate the power performance of the proposed test under different alternatives and with varying dimension. In addition to the Cramér-von Mises-type statistic DD, an alternative version, DDL1, considering the L^1 distance between kernel density estimates of interpoint distances within and between samples



(a) Euclidean distance

(b) Gaussian Kernel

Fig. 2.2 Means and density estimates for within- and between-samples distances using: (a) interpoint Euclidean distances, and (b) quantities defined in RKHSs by computing the Gaussian kernel with median window on pairs of points. Vertical lines represent the means of each density.

has been examined. Specifically, DDL1 is defined by

$$\mathrm{DDL1} = \int_{-\infty}^{\infty} \left| \widehat{f}_{h,\mathcal{W}}(u) - \widehat{f}_{g,\mathcal{B}}(u) \right| du, \tag{2.8}$$

where the bandwidths h and g are automatically determined with the data-driven bandwidth selector proposed by Sheather and Jones [117]. Note that reviewers have mentioned that h and g should be equal, see e.g. [54]. The behaviour in terms of power of DD and DDL1 is examined and compared with other popular two-sample test statistics regarding interpoint distances, which are enumerated below.

2.3.1 Some alternative test procedures

For comparison purposes, some test statistics representative of each of the three families described in Section 2.1.2 were selected. Among the procedures comparing interpoint distances within and between-samples, the energy (E) and maximum mean discrepancy (MMD) methods have been included in our experiments. These methods have been widely studied, can be used in different scenarios, and are versatile for considering a range of distances or kernel functions. Therefore, comparing their behavior in simulations is worthy.

As far as the graph-based family, three procedures were used, namely the k-th Nearest Neighbors (k-NN), the method proposed by Hall and Tajvidi [53] (HT), and the one based on the shortest Hamiltonian path (SHP) proposed by Biswas et al. [18]. The k-NN statistic calculates the quantity k-NN $= \sum_{i=1}^{n+m} \sum_{j=1}^{k} I_i(j)$, where $I_i(j)$ equals 1 if the i-the element in the pooled sample and its j-th nearest neighbor belong to the same sample, and 0 otherwise. Large values of k-NN indicate that the samples are well-separated. Note that k-NN depends on the choice of number of neighbors while the above tests do not require tuning parameters.

The test statistic HT proposed by Hall and Tajvidi [53], consists of evaluating a symmetric distance between data, ranking the pooled dataset with respect to this distance and, for any fixed datum $\mathbf{X}_i$ (analogously $\mathbf{Y}_j$), calculating $M_i(r)$ ($N_j(r)$) equal to the number of $\mathbf{Y}_j$ ($\mathbf{X}_i$) being smaller than the r-th nearest neighbor in the pooled sample, for $r = 1, \ldots, n+m-1$. Conditional on the sample Ξ_X (Ξ_Y) and assuming continuity, $M_i(r)$ ($N_j(r)$) follows an hypergeometric distribution under the null of equal distributions, and hence these quantities can be used to perform the test. In practice, the test is calibrated by using a standard permutation method. The HT test is also versatile to choose a suitable distance and has shown a nice performance in high-dimension (see numerical study reported in [53]).

The two-sample test based on the shortest Hamiltonian path (SHP) proposed by Biswas et al. [18] can be considered as a generalization of the univariate Wald–Wolfowitz run test. The pooled sample is seen as a complete graph with $n+m$ nodes connected by edges with associated cost equal to the Euclidean distance between their vertices. Then, if $\mathcal{H}$ denotes the Hamiltonian path (each vertex is visited only once) with the shortest cost sum, the SHP test statistic counts the number of runs along the $\mathcal{H}$, i.e. the number of consecutive edges in $\mathcal{H}$ connecting data from the same distribution. The SHP test has shown superiority over other existing nonparametric two-sample tests in high-dimension, low-sample-size settings.

In the projection category, we will take into consideration some simulation results obtained in the literature with the direction-projection-permutation (DiProPerm) procedure [143]. In essence, DiProPerm consists in working directly with one-dimensional projections of the data induced by a binary linear classifier trained from the samples, determining the critical level by a permutation approach of the labels of the pooled sample. Several possible choices of the univariate two-sample statistic have been proposed to work with the projected values, including differences between sample means (DiProPerm MD), the two-sample t-statistic (DiProPerm t) and the comparison of areas under the receiver operating curve (DiProPerm AUC).

2.3.2 Multivariate Gaussian distributions

The first experiment in our empirical power study considers multivariate Gaussian distributions. The gaussian setting is of great interest because of the ubiquity of this distribution and the fact that parametric approaches such as T^2 Hotelling do not work properly in high-dimension, low sample-size situations. To gain better insight into how the power of the tests degrades with increasing dimension, Ramdas et al. [106] suggest that the Kullback Leibler (KL) divergence between the compared distributions should remain constant as the dimension increases. Otherwise, for a given sample size, specific parameters for the distribution under the alternative could be chosen to attain a prefixed power. Thus, we adopt the criterion of keeping KL constant. In fact, our experiment followed similar lines to the one carried out by Ramdas et al. [106]. Two scenarios characterized by differences in location (S1) and in scale (S2) are determined.

For a given dimensionality d, multivariate Gaussians differing in the location of the mean for the first coordinate are compared in Scenario S1, where $F_X \equiv N(\mathbf{0}, I_d)$ and $F_Y \equiv N\left((1,0,\ldots,0)^t, I_d\right)$, with $I_d = diag(1,\ldots,1)$ the d-dimensional identity matrix. Scenario S2 involves the distributions $F_X \equiv N(\mathbf{0}, I_d)$ and $F_Y \equiv N\left(\mathbf{0}, diag(4,1,\ldots,1)\right)$, thus generating Gaussians with different scale in the first dimension.

For both scenarios, the remaining simulation features were as follows. Dimension d took the values $\{1,2,3,4,6,8,12,24,50,80,130,200\}$. Sample sizes were $n = m = 30$. The distance D was always the Euclidean distance, and the Gaussian kernel with median heuristic for the bandwidth was used to construct the MMD statistic. Each experiment was repeated $T = 2000$ times. The empirical power averaged over the 2000 trials for a nominal level $\alpha = 0.05$ was calculated with each of the examined test procedures. Plots of the attained power values against the increasing dimension are given in Figures 2.3 and 2.4 for Scenarios S1 and S2, respectively.

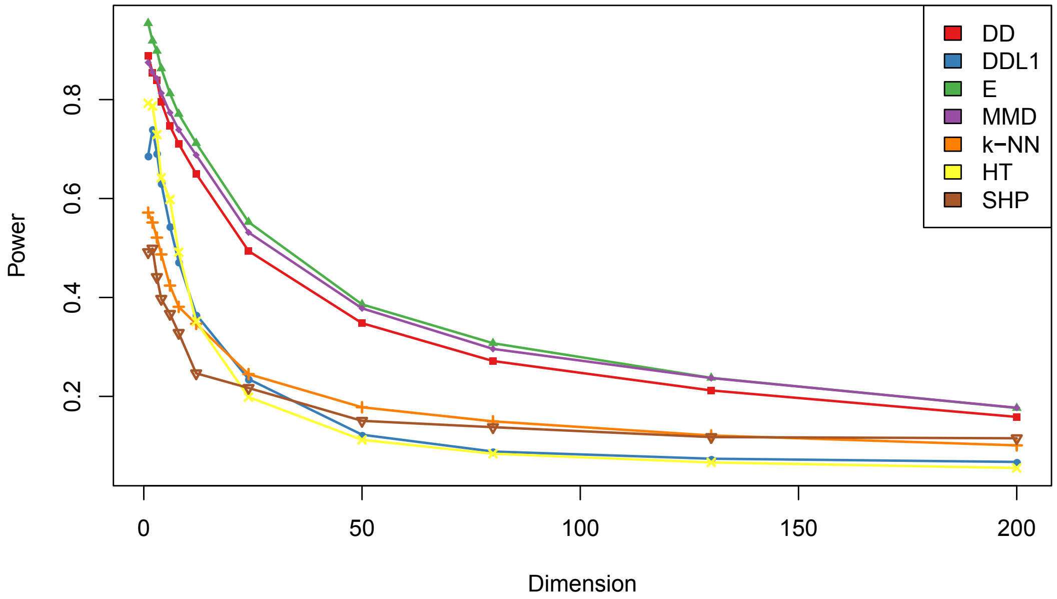


Fig. 2.3 Empirical power vs dimension for several two-sample test procedures using multidimensional Gaussian distributions with mean shift.

All the compared methods fairly show a degradation of power with increasing dimension in both S1 and S2, but different behaviors are observed for each scenario. In Figure 2.3, E and MMD have similar performance and exhibit greater power than all the alternative procedures. The proposed approach based on the CvM-type statistic, DD, presents a power profile very close to the best performing E and MMD. Regardless of dimension, E, MMD and DD form the group of methods producing the best results in power in the scenario with mean-separated Gaussian distributions. The alternative proposal based on measuring the L^1-distance within and between interpoint densities (DDL1) behaves similar to HT. Both of them lead to power values greater than k-NN and the shortest Hamiltonian path for lower dimensions, but their power degrades faster with dimension.

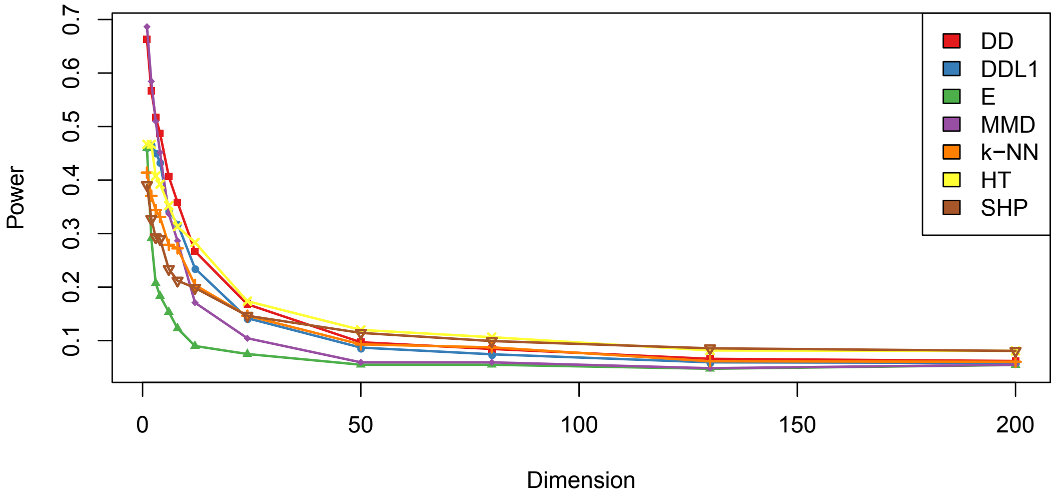


Fig. 2.4 Empirical power vs dimension for several two-sample test procedures using multidimensional Gaussian distributions with variance change.

As for the Scenario S2 with differences in scale, it is noticeable that E and MMD statistics perform in a different way. E produces the worst results, and MMD starts with the highest power at very low dimension but degrades very quickly to match the performance of energy. The proposed method DD together with MMD present the best power values in low dimension, but DD degrades much more slowly than MMD as dimension increases, keeping a competitive power for moderately large dimension. Actually, only HT and SHP achieve higher power in high dimensions.

If the area under the power curve is used to rank the methods at each scenario, it is observed that, except for DD and DDL1, the ranking of the remaining methods in S1 is exactly reversed in S2. Thus, it seems that no method is better than the other for both mean shift and scale change scenarios. Chen and Friedman [27] also draw the attention to the fact that graph-based methods are either good for location or scale alternatives.Only the proposed method DD achieves high positions in both rankings. Adding the two areas under the curves, DD is the best performer followed closely by MMD, E, k-NN, SHP, HT and DDL1. The poor performance of DDL1 might be accounted for the difficulty of selecting proper bands to estimate the kernel densities in high dimension. Note that DDL1 performs reasonably well in both scenarios with low dimension.

It is also worthy to analyze how differently the methods degrade with dimension. For instance, MMD at dimension 1 is clearly a good choice because is close to the best approach in location shift and the best in scale changes, but the quality of its performance decreases more rapidly than other tests when increasing the dimension.

2.3.3 Differences in skewness and kurtosis

In some situations, testing by location and scale differences is not the key issue. For example, two-sample testing has been suggested as a goodness-of-fit approach when samples can be simulated for the reference distribution [44]. This approach is particularly useful in high dimensions. However, in some goodness-of-fit tests, distributions are compared against a reference having by construction the same location and scale parameters. Therefore, besides alternative hypotheses based on different location and scale parameter, it is worthy to assess the capability of the two-sample tests to detect differences in skewness and weight of the tails.

To check for differences in higher order moments, some authors have included numerical results comparing normal and Student's t-distributions [127, 143]. We have considered the family of sinh-arcsinh distributions introduced by Jones and Pewsey [68]. By applying the sinh-arcsinh transformation to a generating distribution, usually the normal, a new class of four-parameter distributions controlling mean, variance, asymmetry and kurtosis is obtained. This way, symmetric and skewed distributions can be generated, but also distributions with heavier or lighter tailweight than those of the generating distribution. Skewness and tailweight can be smoothly changed by moving the respective parameters. We focused on the called *normal sinh-arcsinh* (NSAS) class of distributions, which is generated from the normal distribution. In its canonical version, a random variable $X_{\nu,\tau}$ of the NSAS class is defined by the sinh-arccsinh transformation

$$Z = \sinh\left\{\tau \sinh^{-1}(X_{\nu,\tau}) - \nu\right\},$$

where Z is the standard normal and parameters $\nu, \tau \in \mathbb{R}$, with $\tau > 0$, control skewness and tailweight, respectively. Specifically, skewness increases with ν (positive skewness when $\nu > 0$ and negative if $\nu < 0$), and tailweight decreases with τ (heavier tails than the normal distribution when $\tau < 1$ and lighter ones if $\tau > 1$). The four-parameter extension of $X_{\nu,\tau}$ is obtained by starting from a non-standard Gaussian.

The NSAS class offers a suitable framework to assess the empirical power of the two-samples tests when differences in skewness or tailweight are present. First, the univariate case ($d = 1$) was considered. Two new scenarios S3 and S4 were set up. In both cases, a sample was drawn out from the standard normal, while the other one came from variables $X_{\nu,1}$, for several values of ν in Scenario S3, and from variables $X_{0,\tau}$, for several values of τ in Scenario S4. The sample sizes were $n = m = 20$ in S3 and $n = m = 40$ in S4. Implementation was performed as in Stasinopoulos and Rigby [124]. Figures 2.5 and 2.6 show the average power attained in Scenarios S3 and S4.

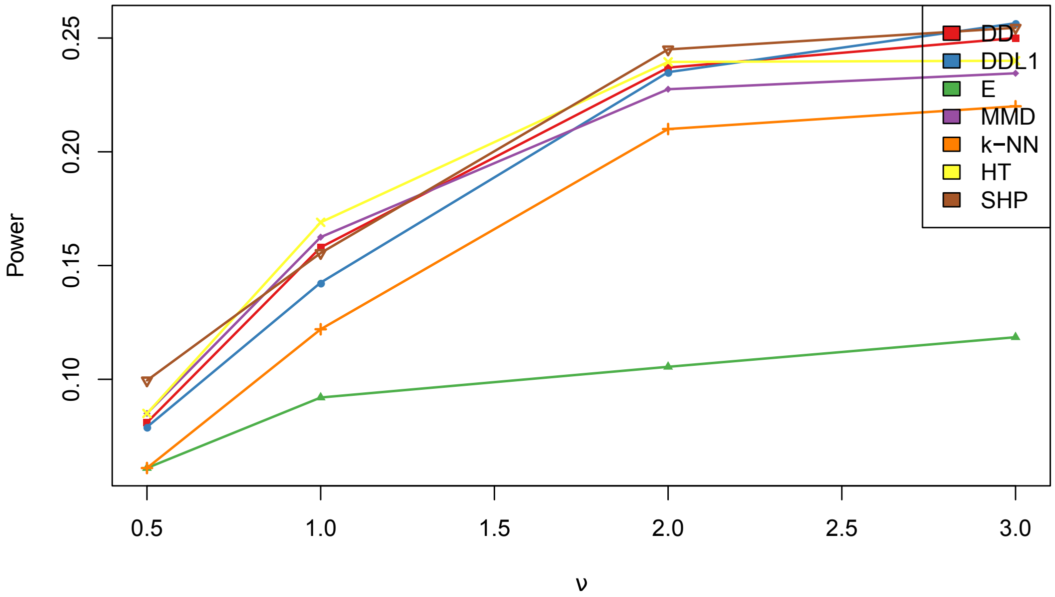


Fig. 2.5 Empirical power vs skewness level controlled by ν for several two-sample tests. Skewness increases with ν (symmetry if $\nu = 0$).

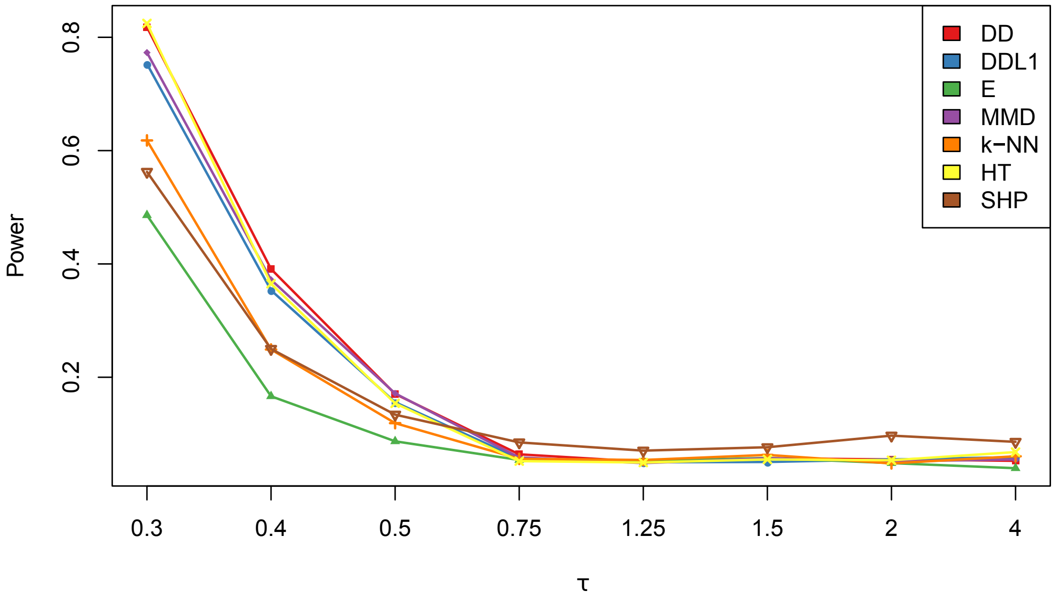


Fig. 2.6 Empirical power vs kurtosis level controlled by τ for several two-sample tests. While $\tau = 1$ corresponds to the standard normal, $\tau > 1$ ($\tau < 1$) means greater (smaller) kurtosis than the standard normal.

In Scenario S3, no test procedure performs uniformly better than the others since different behaviors are observed when the skewness level varies. For instance, the shortest Hamiltonian path (SHP) is one of the best performing but it is ranked fourth when $\nu = 1$. Indeed, the empirical power improves when skewness increases, and this improvement is more rapidly attained with the methods proposed in this chapter. For the largest ν, DDL1 and DD exhibit the first and third higher powers, respectively. The tests based on k-NN and particularly on the energy statistic led to the worst power values for all skewness levels. MMD performs substantially better than E in this scenario, although its power for high skewness is always surpassed by the proposed tests and the SHP and HT procedures. In Scenario S4 considering the effect of tailweight, all the methods showed higher power for values of τ less than 1 (heavier tails) than when τ is greater than 1 (lighter tails). Statistic DD together with MMD, DDL1 and HT fairly exhibit higher power than the remaining methods, particularly in the case of strongly heavy tails (τ close to zero). Note that SHP presents a poor behavior, also failing to achieve the nominal value (above 0.05 when $\tau = 1$). Our experiments definitely show that detecting alternatives characterized by light tails is a difficult task for all the examined procedures since very poor and similar power values were reach with all of them.

Next step consisted in analyzing the dimensionality effect when one wishes to detect departures from the null in skewness or kurtosis. As in Section 2.3.2, differences in skewness and kurtosis were only set for the first marginal of the compared multidimensional distributions. Sample sizes were $n = m = 100$ and again 2000 trials were run in order to generate average power values. Figures 2.7 and 2.8 show the average power results when varying the dimensionality, for fixed skewness $\nu = 3$ (Scenario S5) and kurtosis $\tau = 0.2$ (Scenario S6) parameters, respectively.

In both scenarios, S5 and S6, power results degrade fast and substantially with dimension, particularly in the Scenario S5 considering skewness. In fact, only for very small values of d, reasonable powers are attained in S5, being again noticeable the nice behavior of DD and DDL1. As d increases, all the methods exhibit very poor powers and DDL1 is somewhat superior to DD. The influence of dimension is also important in Scenario S6 considering heavy tails, but less marked than in S5. With low dimension, the graph-based procedures seem to work better and only DDL1 is able to show competitive results among the procedures based on within and between interpoint distances. In both scenarios, E and MMD are the procedures with the lowest power regardless of the considered dimension.

It is worth comparing the power behavior in Scenarios S1-S2, considering changes in location and scale, with Scenarios S5-S6, considering changes in skewness and kurtosis. It is observed that the degradation of power with dimension d is much faster and acute in S5 and S6. While all the methods have almost no discriminatory power at $d = 64$ to detect differences in skewness or kurtosis, similar power values are reached for some

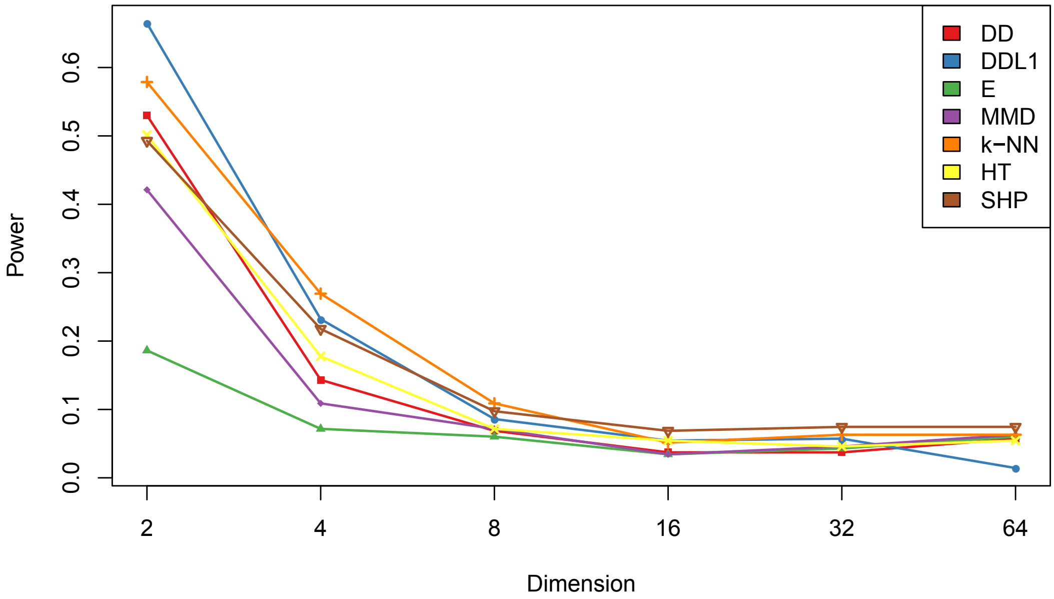


Fig. 2.7 Empirical power vs dimension for several two-sample tests when asymmetry on the first variable moves is set to $\nu = 3$.

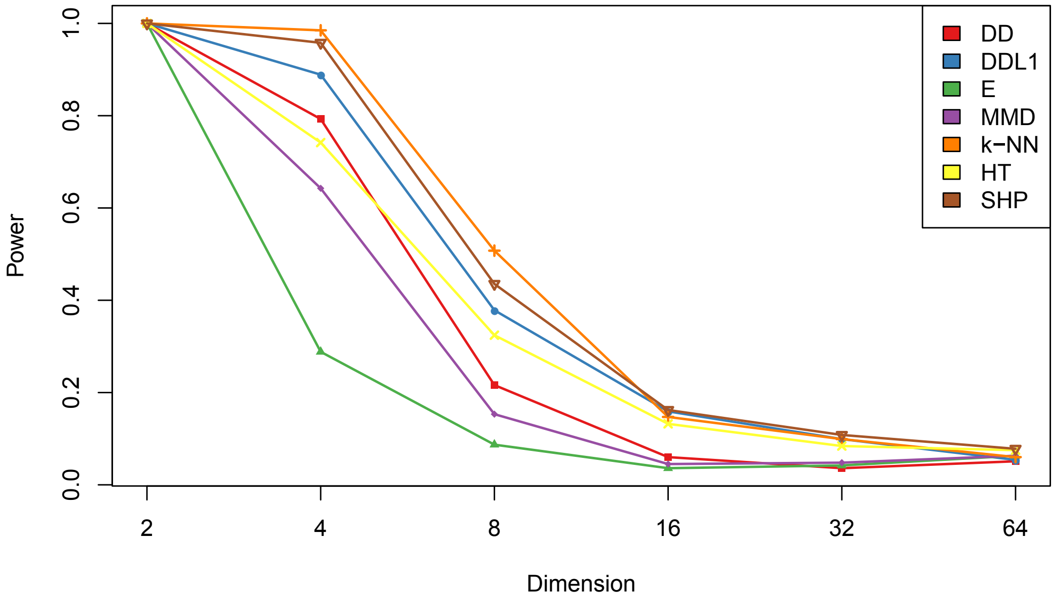


Fig. 2.8 Empirical power vs dimension for several two-sample tests when asymmetry on the first variable is set to $\tau = 0.2$.

methods at $d = 200$ by testing for differences in location or scale. The strong influence of the dimensionality is more noticeable at low dimensions. Many methods achieve powers of 100% at $d = 2$ in the kurtosis scenario S6, higher than the ones at the same dimensionality for scale and location. In sum, all the studied methods are affected by the noise introduced by adding noninformative dimensions.

2.3.4 Effect of large location shift combined with small covariance difference

To examine the joint effect of changes in location and scale, we have replicated one of the experiments studied by Wei et al. [143]. Simulation S3 in [143] proposes a scenario combining changes in location and scale related to the dimension in such a way that the location shift increases with dimension while the scale difference decreases. In addition, replicating this simulation allows us to compare the proposed approach with the DiProPerm procedure, which belongs to the projection-based two-sample test family.

The new scenario (hereinafter referred to as Scenario S7) involves samples of size $n = m = 100$ generated from multivariate Gaussian distributions $N(\mu_X, \Sigma_X)$ and $N(\mu_Y, \Sigma_Y)$, where μ_X is the zero vector and μ_Y is zero in the first 25% of the coordinates and $1/\sqrt{n}$ in the rest. Covariance matrices are constructed as follows. Denote by $\lambda_{min}(A)$ the minimum eigenvalue of a symmetric matrix A. Let $S = (s_{ij})$ be the $d \times d$ matrix such that $s_{ii} = 1$, $s_{i(i+1)} = s_{(i-1)i} = 0.2$, and $s_{ij} = 0$ otherwise. Let U be a $d \times d$ matrix with random entries in the upper triangle generated from a uniform distribution over $(0, 32/d^2)$, and the elements in the lower triangle determined by symmetry. Then, we set $\Sigma_X = S + \delta I_d$ and $\Sigma_Y = S + U + \delta I_d$, with $\delta = |\min[\lambda_{min}(S), \lambda_{min}(S+U)]| + 0.05$.

In order to compare our results with those obtained with the DiProPerm procedure in [143], identical sample size and values for dimension were considered, namely $m = n = 100$ and $d = (25, 50, 100, 200, 400, 800, 1600)$. The obtained empirical powers are shown in Figure 2.9.

DiProPerm MD, E and MMD produce the highest power values, very close to each other. By construction, the three procedures take advantage of the strong mean effect, which increases with dimension. Combined with the decreasing influence of the scale difference, the satisfactory behavior of these procedures is not surprising. Significantly, the proposed DD procedure is next, with very high power levels as well. The rest of methods present fairly worse results. Note that, unlike Scenarios S1 and S2 characterized by differences purely based on location or scale, the power increases with dimension in Scenario S7. This is a noticeable fact accounted for an increasing amount of dimensions contributing to discrimination, while in S1 and S2 the differences in location and scale are only present in the first coordinate direction. In addition,

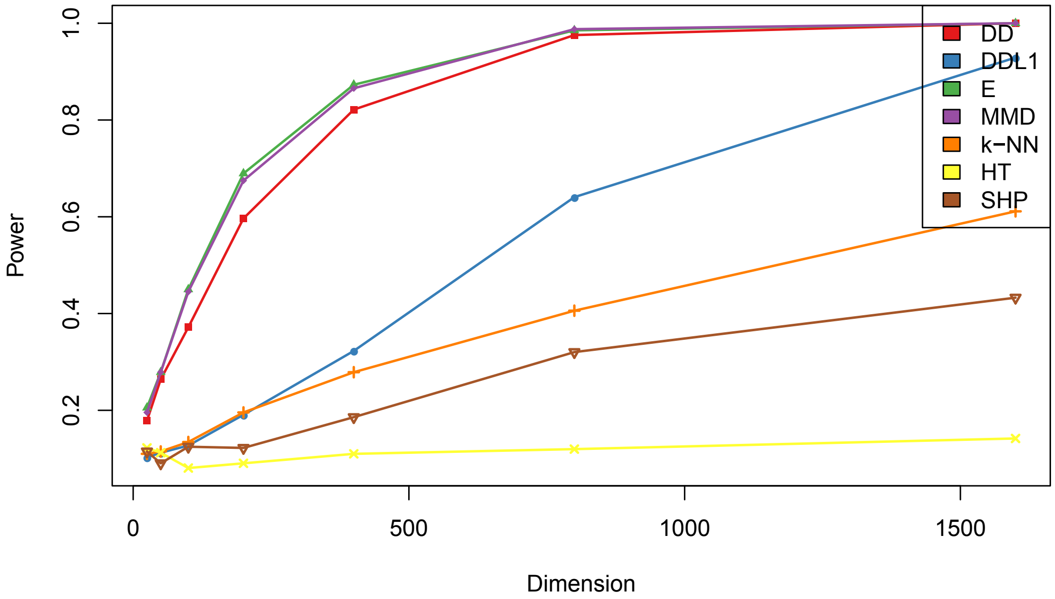


Fig. 2.9 Power vs dimension for large mean, small covariance alternative (Wei Simulation 3).

DiProPerm MD, E, MMD and DD exhibit rates of increase in power with dimension much higher than the other methods.

2.3.5 Comparing grids of bivariate Gaussians distributions

The next experiment has been proposed by Gretton et al. [52] and reproduced in [32]. In the new scenario, let us say S8, samples of two equally weighted mixtures of bivariate Gaussian distributions forming 5×5 grids are compared. Difference between the generating distributions lies in the correlation structure for the Gaussians forming the mixtures. The Gaussian components have uncorrelated coordinates for one of the mixtures but they are correlated for the other mixture. Gretton et al. [52] have considered different amounts of correlation, measured as the ratio between the largest and smallest eigenvalue of the covariance matrix. In our simulation, we set this ratio to 10. More specifically, the Gaussian mixtures in Scenario S8 are formed by 25 bivariate Gaussian distributions with equal weight, locations separated by 15 and with unit variance on both dimensions, and off-diagonal covariance equal to zero for one mixture and to 0.8182 for the other. A set of realizations from both mixtures is shown in Figure 2.10.

In this scenario, we are not particularly concerned about dimension, but it is considered to be a challenging setting for the two-sample testing problem, thus accounting for our interest in examining the behavior of the proposed approaches based on DD and

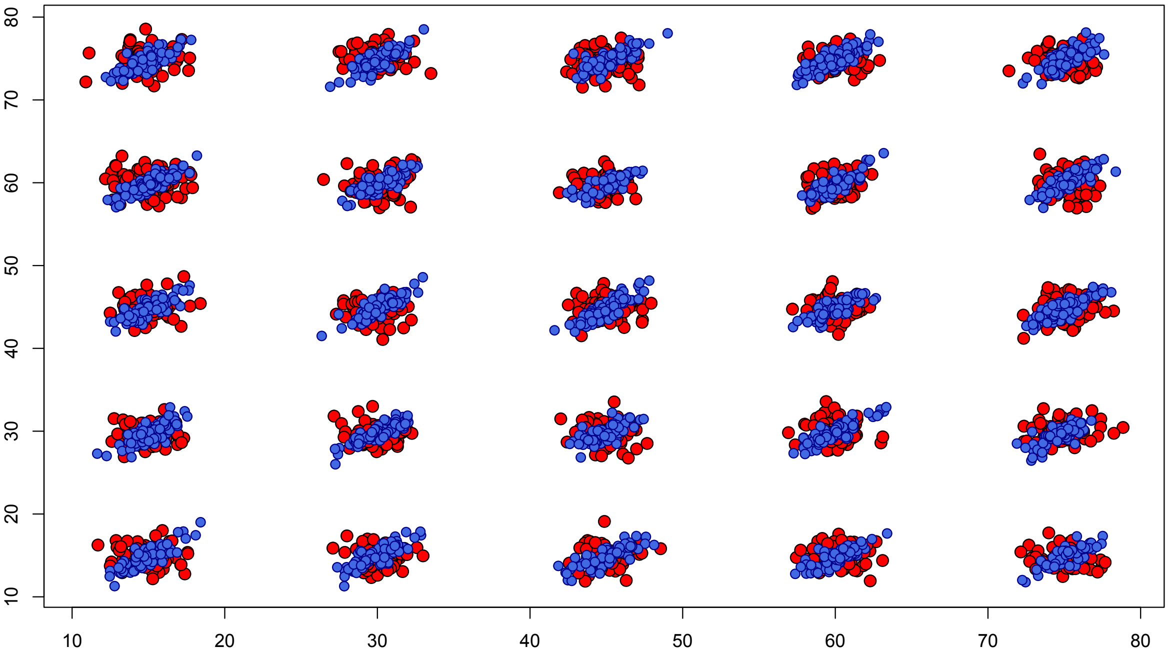


Fig. 2.10 A realization of the Gaussian grids forming Scenario S8.

DDL1 test statistics. The experiment was carried out with sample sizes $n = m = 1800$ and using E, MMD, DD and DDL1. The obtained results provide compelling evidence of the superior behavior of DD and DDL1. In fact, E and MMD showed absolute lack of discriminatory power in this setting, producing a empirical power equal to zero in both cases. On the contrary, our distribution and density based methods were close to full power, attained average empirical powers of 0.96 and 1, respectively. In conclusion, it has been necessary to compare the whole distributions of the interpoint distances to detect the differences in the underlying distributions. Indeed, a suitable and careful kernel choice can be performed in order to improve the behavior of MMD [52]. Nevertheless, using much larger sample sizes ($n = m = 10000$) in the same Gaussian grid scenario, the best linear MMD version reported power results below 0.8, still far from the power attained with DD and DDL1, see the right side of Figure 1 in [51].

2.3.6 Time series

Now, we assume that the random vectors are realizations of time series, i.e. $\Xi_X \equiv \{\mathbf{X}_1, \ldots, \mathbf{X}_n\}$ and $\Xi_Y \equiv \{\mathbf{Y}_1, \ldots, \mathbf{Y}_m\}$ are n and m independent realizations of length d from real-valued stochastic processes $X = \{X_t, t \in \mathbb{Z}\}$ and $Y = \{Y_t, t \in \mathbb{Z}\}$, respectively. Time series allow to model complex dependence structures and are inherently high-dimensional since d is usually large in this setting. In fact, dimensionality is often much greater than the sample size when dealing with time series. By setting the same length for all series, the two-sample test for equal distribution in high dimension posed in (2.1) is also of interest in this context. Furthermore, based on the mentioned features, scenarios consisting of time series become an interesting test bed for the examined statistics. We have carried out three experiments including autoregressive processes. The two first experiments, leading to Scenarios S9 and S10, replicate examples proposed by Biswas et al. [18], and the third one, generating the Scenario S11, combines features from S9 and S10. An extensive study of time series is presented in Chapter 3.

The three experiments consider $n = m = 20$ realizations of length $d = 50$ so that the differences lie in the generating processes. In the first experiment (Scenario S9), two linear autoregressive processes of order 1 with different parameter are considered. Specifically, $\mathbf{X}_i$ and $\mathbf{Y}_j$, for $i, j = 1, \ldots, 20$, are generated from the AR(1) models $X_t = 0.3X_{t-1} + \varepsilon_t$ and $Y_t = 0.5Y_{t-1} + \varepsilon_t$, respectively. In both cases, innovations ε_t are independent $N(0.25, 1)$ variables so that the joint distributions of the two samples present a constant location shift in all dimensions as well as exhibiting different variance structures. Scenario S10 consists of realizations of the AR(2) processes $X_t = 0.3X_{t-1} + 0.2X_{t-2} + \varepsilon_t$ and $Y_t = 0.4Y_{t-1} + 0.3Y_{t-2} + \varepsilon_t$, with standard Gaussian errors in both models. The joint distributions only differ in the variance structure in S10. The third experiment considers the AR(2) structures of S10 but with $N(0.25, 1)$ innovations as in S9. This enables us to compare the effect of the innovations at the same model complexity.

The experiments were repeated 500 times and the average powers with each statistic for $\alpha = 0.05$ are shown in Table 2.2. Results for Scenarios S9 and S10 are consistent with the ones in [18], where SHP led to powers higher than Energy and 3-NN. However, the proposed approaches DD and DDL1 work substantially better than SHP in these scenarios, and only HT presents higher power values. Scenario S11 replicates the AR(2) structures of Scenario S10 but adding mean to the innovations. This makes stronger marginal location differences (also bigger than in S9) and leads to the easiest scenario, thus accounting for the best performance of all the tests. In this case, E and MMD take advantage of the differences in location and outperform SHP. The difference in relative performance of SHP suggests that this method is comparatively better when changes in covariance are more subtle. Again DD achieves a very good result (0.82) comparable to the best one (0.83) provided by HT. It is interesting to note that the scenarios including time series are the only experiments where HT is the best performer.

Scenario	E	3-NN	MMD	SHP	HT	DDL1	DD
S9	0.20	0.18	0.26	0.24	0.59	0.34	0.44
S10	0.10	0.17	0.20	0.24	0.65	0.46	0.48
S11	0.67	0.48	0.75	0.46	0.83	0.69	0.82

Table 2.2 Empirical power in experiments involving time series

2.3.7 Discrete data

Discrete distributions may be problematic to some tests, such as graph-based tests when there are ties in the distance graph, as noted in [145]. This could end up in a non unique definition of a nearest neighbor or shortest hamiltonian path. To illustrate the performance of the proposed test, we pose several simulation scenarios generated using the following probability mass function (used in [145]):

$$P_{\theta,\eta}(x) = \frac{1}{\psi(\theta)} \exp(-\theta\rho(x,\eta)),$$

where η, the center of the distribution, is a list of numbers, θ controls the spread, ρ is the Spearman's distance and ψ a normalizing constant. The domain of this distribution is any permutation of the center η. Each experiment is repeated 1000 times considering a nominal level $\alpha = 0.05$. We check the empirical levels for the scenario with parameters: $n = m = 25, \eta_1 = \eta_2 = (1,2,3,4,5), \theta_1 = \theta_2 = 1$. Table 2.3 shows that our proposed methods, energy and MMD do not show problems when the domain is discrete, while the graph-based approaches produce incorrect nominal levels. NN works well because of the 3 first neighbors are being considered, its power differs from the expected when only one neighbor is considered. If the amount of possible values in the domain is increased, e.g. considering 9 numbers instead of 4, changing the η in the previous scenario to $\eta = (1,2,3,4,5,6,7,8,9)$, the graph based tests are closer to the expected power.

Scenario	E	3-NN	MMD	SHP	HT	DDL1	DD
A	0.053	0.048	0.050	0.141	0.063	0.052	0.050
B	0.053	0.054	0.054	0.073	0.056	0.052	0.052

Table 2.3 Empirical nominal levels for the discrete data scenarios A and B, where A: $\eta = (1,2,3,4,5), \theta = 1$, B: $\eta = (1,2,3,4,5,6,7,8,9), \theta = 1$

For differences between distributions, we consider the following scenarios:

- S12: A small change in spread:
  $N = M = 25, \eta_1 = \eta_2 = (1,2,3,4,5), \theta_1 = 0.5, \theta_2 = 2.$
- S13: A change of "center" with medium spread:
  $N = M = 25, \eta_1 = (1,2,3,4,5), \eta_2 = (5,4,3,2,1), \theta_1 = \theta_2 = 1.$
- S14: A large change in spread:
  $N = M = 25, \eta_1 = \eta_2 = (1,2,3,4,5), \theta_1 = 0.5, \theta_2 = 3.$
- S15: A change in center with small spread in both distributions:
  $N = M = 25, \eta_1 = (1,2,3,4,5), \eta_2 = (5,4,3,5,1), \theta_1 = \theta_2 = 0.5.$

The results are shown in Table 2.4. Graph based tests achieve less power than energy, MMD and the proposed approach. In S12, S13 and S14 our method achieves results very close to the best ones, while in the S15 it performs best. It is interesting to note that the density-based version of our tests, achieves the best result in S14. In this scenario, a simple plot of the densities of the groups shows a similar setting as in Figure 2.2: mean-based methods such as energy and MMD tend to confound the means while the differences in densities or cdf's are more clear.

Scenario	E	3-NN	MMD	SHP	HT	DDL1	DD
S12	0.400	0.180	0.398	0.277	0.381	0.233	0.399
S13	0.806	0.512	0.801	0.491	0.806	0.610	0.801
S14	0.808	0.432	0.809	0.537	0.758	0.802	0.807
S15	0.301	0.270	0.293	0.334	0.313	0.783	0.687

Table 2.4 Empirical power in discrete scenarios.

2.3.8 Some benchmark datasets in classification

Following [18], the two-sample test procedures were also examined on the basis of benchmark datasets extensively used in the literature on supervised classification. The authors argue that there is a reasonable separation between two competing classes, and therefore the alternative hypothesis of unequal underlying distributions can be assumed. We have replicated their experiment for a fixed sample size at each case and considering additional datasets, namely Spambase, Phoneme and Urbanland. All the considered datasets are enumerated and briefly described in Table 2.5. The Colon dataset is available within the R package **dprep** [4]. The Trace and Phoneme datasets are obtained from the University of California, Riverside's time series classification/clustering page, at www.cs.ucr.edu/~eamonn/time series data/ [30]. The rest of the datasets are taken from the University of California, Irvine's machine learning repository, at http://archive.ics.uci.edu/ml/datasets/ [79].

Name	Brief description
COLON	Microarray gene expression dataset containing expression levels of 2000 genes for each of 62 samples, 40 from colon cancer tissue and 22 from normal tissue.
ARCENE	Mass-spectrometry dataset for patients with ovarian or prostate cancer and healthy patients. It consists of 7000 features indicating abundance of proteins in human sera having a given mass value and 3000 additional distractor features called "probes".
IONOSPHERE	Radar dataset containing 34-dimensional observations on 126 "good" and 225 "bad" radar returns. A good return is that showing evidence of some type of structure in the ionosphere, while a bad return does not (its signal pass through the ionosphere).
SONAR	Formed by 111 patterns obtained by bouncing sonar signals off a metal cylinder and 97 patterns obtained from rocks. Each pattern is a set of 60 numbers in $(0,1)$, and each number represents the energy within a particular frequency band, integrated over a certain period of time.
SPAMBASE	Measures of 57 attributes for a collection of 4601 spam and non-spam e-mails. Most of the attributes indicate whether a particular word or character was frequently occurring in the e-mail.
PHONEME	Formed by 2000 discretized log-periodograms obtained from digitized speech outputs for five phonemes (aa, ao, dcl, iy, sh).
TRACE	Consists of four classes, each containing 50 instances of length 275. It is a time series dataset designed to simulate instrumentation failures in a nuclear power plant.
URBANLAND	148 variables extracted from high resolution hyperspectral aerial images of urban land cover. The classes of data can be trees, grass, concrete, buildings, asphalt...

Table 2.5 Benchmark datasets

As in [18], we consider two different two-sample problems for the Trace dataset, namely TRACE1 comparing the first and second classes and TRACE2 comparing the third and fourth classes. Besides checking by the equality of distributions of two classes for each dataset, we have also created new scenarios where mixtures of classes with different weights p are compared, i.e. the samples are formed by drawing observations from classes one and two of the dataset, but at different proportions.

This was carried out for the Ionosphere and Urbanland datasets, thus generating the scenarios called MIX-IONO and MIX-URBAN, respectively. Furthermore, the scenario called MIX-IONOSONAR combines, in different proportions, observations from class one in Ionosphere and class one in Sonar. In this case, we use the first 34 variables of the Sonar dataset to have the same dimension as observations in Ionosphere. This way, MIX-IONOSONAR scenario can be seen as a "contamination" scenario because the two distributions of the mixture have different nature. By including these mixed scenarios, we intend to examine the power of the test procedures when the distribution supports overlap significantly.

At each case, the experiment was repeated 2000 times to approximate the testing power. The distributions were simulated by sampling without replacement from each considered class, taking equal sample size for the two samples (train and test datasets were joined). For the mixture distributions, the proportions of random observations within each class were determined according to a binomial distribution. Specifically, the chosen binomial parameters were

- $p = 1.0$ vs $p = 0.7$ for the MIX-IONO scenario.
- $p = 0.3$ vs $p = 0.75$ for the MIX-URBAN scenario.
- $p = 0.5$ vs $p = 0.7$ for the MIX-IONOSONAR scenario.

In order to get insight into the effect of the distance between observations, the experiments were carried out considering two different interpoint distances, namely the Euclidean distance and the L_∞ or Chebyshev distance. Table 2.6 shows the power results for a nominal level $\alpha = 0.05$ for the Euclidean distance.

Scenario	E	3-NN	MMD	SHP	HT	DDL1	DD
COLON (n=15)	0.88	0.97	0.88	0.89	0.33	0.51	0.78
IONO (n=12)	0.87	0.83	0.91	0.96	0.78	0.84	0.86
SONAR (n=25)	0.62	0.80	0.67	0.78	0.23	0.36	0.55
ARCENE (n=20)	0.38	0.91	0.38	0.88	0.51	0.56	0.50
TRACE1 (n=17)	0.27	0.69	0.32	1.00	0.54	0.49	0.27
TRACE2 (n=25)	0.01	0.49	0.00	1.00	0.00	0.04	0.01
SPAMBASE (n=27)	0.81	0.74	0.85	0.62	0.82	0.81	0.90
PHONEME (n=20)	0.95	0.83	0.95	0.72	0.76	0.76	0.93
URBANLAND (n=20)	0.86	0.86	0.83	0.71	0.77	0.71	0.85
MIX-IONO (n=34)	0.18	0.15	0.20	0.27	0.21	0.17	0.18
MIX-IONOSONAR (n=34)	0.28	0.15	0.31	0.13	0.31	0.29	0.30
MIX-URBAN (n=40)	0.41	0.31	0.41	0.26	0.39	0.35	0.43

Table 2.6 Empirical power in experiments with benchmark datasets using the interpoint Euclidean distance

In the scenarios taken from [18], the first six rows of the Table 2.6, SHP and 3-NN are the best performing methods. Our method DD achieves lower power than Energy and MMD in the Colon and Sonar datasets, similar power in the Ionosphere dataset and higher power in the Arcene dataset. SHP is clearly the best method in the TRACE1 and TRACE2 datasets, where E, MMD and DD present similar and poor powers. It is worth remarking that in the TRACE2 scenario, most methods do not achieve even the nominal power. On the contrary, in the Spambase, Phoneme and Urbanland datasets, the best performing procedures are either Energy or the proposed DD, while SHP exhibits the lowest power. The mixtures introduce interesting effects. In the mixture of Ionosphere classes, while the relative performance of the methods is almost kept compared to the pure classes scenario (which can be understood also as mixtures with parameter $p = 1.0$), the power of the 3-NN method is impacted and ends up being the worst performer. Overall, HT is the least affected in the mixture scenarios.

The results using the Chebyshev distance as interpoint distance are given in Table 2.7. Compared to the Euclidean distance, the Chebyshev distance produces higher powers in some datasets such as Ionosphere, but also lower powers in others such as Colon. These increases and decreases are not uniform on all the methods in the sense that relative ordering of the methods by power is changed, e.g. DD achieved the best power in the Ionosphere dataset with the Chebyshev distance but not with the Euclidean. The performance in the mixtures is comparable to the Euclidean case.

Scenario	E	3-NN	MMD	SHP	HT	DDL1	DD
COLON (n=15)	0.54	0.72	0.48	0.56	0.09	0.09	0.44
IONO (n=12)	0.95	0.86	0.96	0.97	0.89	0.95	0.98
SONAR (n=25)	0.36	0.56	0.40	0.57	0.28	0.36	0.31
ARCENE (n=20)	0.16	0.34	0.19	0.19	0.34	0.28	0.24
TRACE1 (n=17)	0.59	0.96	0.75	1.00	1.00	1.00	1.00
TRACE2 (n=25)	0.01	0.78	0.00	1.00	0.01	1.00	0.01
SPAMBASE (n=27)	0.79	0.72	0.84	0.59	0.82	0.79	0.88
PHONEME (n=20)	1.00	1.00	1.00	1.00	1.00	0.50	0.50
URBANLAND (n=20)	0.87	0.73	0.82	0.57	0.77	0.68	0.83
MIX-IONO (n=34)	0.20	0.16	0.21	0.22	0.23	0.22	0.24
MIX-IONOSONAR (n=34)	0.26	0.15	0.29	0.13	0.38	0.25	0.29
MIX-URBAN (n=40)	0.43	0.27	0.39	0.22	0.38	0.34	0.42

Table 2.7 Empirical power in experiments with benchmark datasets using the interpoint Chebyshev distance

2.4 Concluding remarks

We have introduced a new class of multidimensional two-sample homogeneity test consisting of three steps, namely i) defining a suitable distance D between observations, ii) calculating all the interpoint distances, and iii) checking by the equality of the univariate distributions of interpoint distances between and within samples using the Cramér-von Mises test statistic. Our approach presents appealing properties, including reduction of dimensionality, being free of parameter tuning and inheriting the theoretical properties of the univariate Cramér-von Mises test. The nonrestrictive regularity conditions required for the distance D in Lemma 1 provide versatility for a suitable choice of D in order to increase the test power and get insight into the nature of the distributional differences. This is particularly interesting in the case of complex objects such as times series or functional data where different distance measures have been introduced in the literature (see e.g. [94, 42]).

Unlike other known test statistics based on within and between-samples interpoint comparisons aimed at evaluating differences in location, our proposal relies on comparing the whole interpoint distances distributions, thus providing higher capability to detect differences in shape or in other moments. This property has been motivated by means of illustrative examples in Section 2.2.3 and it is indeed a strength of the proposed test.

The power performance of our test has been examined through an extensive simulation study including multidimensional scenarios with changes in location, scale, skewness or kurtosis, scenarios formed by time series, and lastly replicating experiments with simulated and real datasets considered in related papers. Compared to the behavior of a broad range of alternative two-sample tests based on interpoint distances, all our experiments have revealed a very satisfactory performance of the proposed test.

Although no procedure has been uniformly more powerful than the others, the proposed test has shown a robust behavior by attaining reasonable power values in most scenarios. In fact, unlike of our proposal, we have always detected scenarios where some of the competitors have failed, i.e. they have presented poor power values compared to others. For instance, Energy in the concentric circular distributions shown in Section 2.2.3 or with changes in covariance, the nearest neighbor in large dimension scenarios or by comparing grids of bivariate Gaussian distributions, and so on.

As expected, our test has exhibited high power in the case of changes in location, although slightly worse than the best performing Energy and MMD. Also, our proposal presents the best power values for alternatives based on changes in scale and it is located between the most powerful by detecting departures from the null in skewness and kurtosis. As noted in [27] and confirmed in our experimental results, most methods are either sensitive against changes in location or scale. Unlike the rest of the compared methods, our proposed method DD achieves relatively good power in both scenarios.

In essence, our procedure takes advantage of the sensitivity of the interpoint distributions to detect subtle changes in the covariance and dependence structure in the multidimensional setting.

Our results have also shown that the statistic based on comparing empirical distributions (DD) exhibits a more robust behavior, while the one based on comparing densities (DDL1) is more sensitive to changes in shape. Note that a more sophisticated search of the bandwidth parameter would allow to improve the results with DDL1. These results are congruent with the detailed discussion and comparison between distribution and density based approaches provided by Martínez-Camblor and de Uña-Álvarez [91].

Although our work has focused on continuous variables, Theorem 1 in [85] (establishing the result (A.1) in Appendix A for discrete variables) together with the modifications for the univariate Cramér-von Mises statitic in order to deal with discrete distributions proposed by Choulakian et al. [31] and Lockhart et al. [83] would allow to extend our two-sample test to the case of discrete variables. The good performance of our test on discrete data is also illustrated in a short simulation in Section 2.3.7.

The main weakness of our proposal is the high computational cost, presenting a computational complexity $O((n+m)^2 \log(n+m))$. Though permutations increase computing time linearly, they can be computed fast since most calculations can be reused, the most time consuming step is the calculation of the pairwise distances and the statistic the first time.

In summary, we have presented a robust and powerful two-sample test procedure for high dimensional data, which is applicable to a broad range of types of multidimensional data. The choice of a suitable distance D in order to improve the power of the test is an open and interesting issue which deserves deeper study, particularly in the case of complex data objects such as time series, where in addition the dimension is frequently greater than the number of data objects.

Chapter 3

Time series two-sample testing

3.1 Introduction

The two-sample homogeneity testing problem addressed in Chapter 2 for general multidimensional data is specifically analyzed for time series data in the present chapter. As we argue later, two-sample tests with time series data are of particular interest in many applications, and furthermore, we intend to shed light on how the choice of a proper distance between time series may substantially improve the power of the testing procedures. Thus, Specifically, let $\{\mathbf{X}_1, \ldots, \mathbf{X}_n\}$ and $\{\mathbf{Y}_1, \ldots, \mathbf{Y}_m\}$ be n and m independent realizations of real-valued processes $\mathbf{X} \equiv \{X_t : t \in \mathbb{Z}\}$ and $\mathbf{Y} \equiv \{Y_t : t \in \mathbb{Z}\}$, respectively. Assuming equal length T for all observed series and denoting by F_X and F_Y the T-dimensional distributions of the random vectors $(X_1, \ldots, X_T)$ and $(Y_1, \ldots, Y_T)$, respectively, the two-sample test can be formally stated as

$$\begin{cases} H_0: & F_X = F_Y \\ H_1: & F_X \neq F_Y \end{cases} \tag{3.1}$$

If the null hypothesis in (3.1) is not refused, it can be concluded that all the time series follow the same temporal pattern over the observation period. Therefore, although the problem is posed in terms of a two-sample test for equal distributions in high dimension, the final target is to check for differences in the underlying dynamics of both groups of realizations. Note that two-sample problems involving partial realizations of time series frequently arise in applications from different fields. Some motivating examples in the literature are: examining whether two sets audio tracks sampled in short sequences form an homogenous group is a standard problem in audio segmentation [55], two-sample tests for detecting intervals of differential gene expression in microarray time series [126, 103], detecting structural changes in the flows of the river Nile by comparing annual flow series before and after the construction of the Aswan dam [77], comparison of abundance series of phytoplankton species in marine ecosystems studies [46], comparing temperature patterns [53] or brain signal analysis [97]. Nevertheless, classical multivariate approaches to handle (3.1) could face serious drawbacks in a time series framework due to the specific characteristics associated to series. In this context, dimensionality is often much greater than the sample size ($p \gg n$). Frequently, the observed time series have different length. In many problems the interest lies in the temporal behavior while differences in terms of location or dispersion tend to be considered as nuisance factors, thus series are normalized prior to analysis. Other relevant issues such as phase difference tend to be ignored too. In short, this kind of particularities limit the effectiveness of commonly used multivariate methods, requiring specific time series approaches, as can be seen in related problems such as time series clustering or classification.

Likewise distances or dissimilarities between time series objects play a key role in many successful methods of time series classification [94, 81], intuitively one would also expect that knowledge on the distances between time series should be useful to address the two-sample problem (3.1). In fact, a number of successful nonparametric multivariate two-sample tests based on distances have been introduced in the literature [59, 127, 51, 18]. All of them take advantage of being suited for the $p \gg n$ (a desirable property treating with time series) mainly due to relying on distances. Furthermore, an increasing research effort has been put on generalizing the class of distances that may be used with these methods [128, 116, 59]. We argue that different distances are expected to behave better than others, e.g. the supremum/infimum norm outperforms the Euclidean norm in multivariate scenarios where the difference lies only in the location of one of the variables (see Section 3.2 in [53]).

These evidences and the fact that distances are often used to convey invariance to time series nuisance factors (see [12] for a review of invariance types required in several time series domains) motivated us to study the behavior in two-sample problems of different combinations of distances and test statistics. There are two main contributions in this chapter. First, the test statistic proposed in Chapter 2 based on comparing the empirical distributions of pairwise distances is discussed and motivated for the setting of time series. Second, by means of an extensive numerical study, the effect of choosing suitable distances to increase the power of the test is shown. In particular, it is observed that the proposed testing method produces good results for a large set of distances.

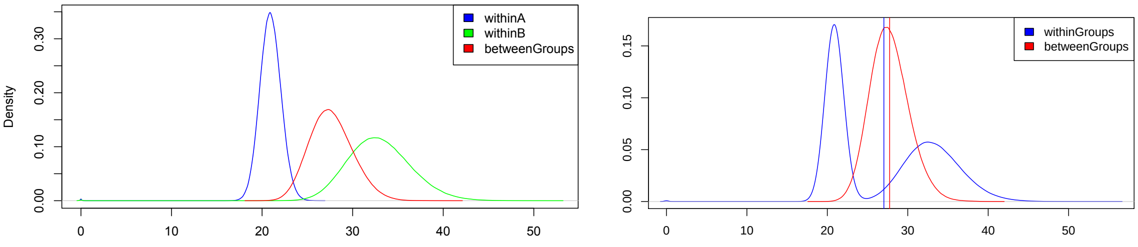
The rest of the chapter is structured as follows. We present a brief discussion on the new test statistic proposed in Chapter 2 intending to shed light on the intuition behind the test compared to a conceptually close alternative. Section 3.3 presents simulation and real data experiments comparing different test methods and dissimilarity measures between series. An applicative example is described in Section 3.4 in order to illustrate the usefulness and versatility of our proposal, and some conclusions and lines for further research are stated in Section 3.5.

3.2 A rationale for a new test in the time series setting

In Chapter 2 we propose a two-sample test statistic based on comparing distributions of interpoint distances. Like our proposal, the energy statistic can be understood in terms of within- and between-groups interpoint distances. Adding to Section 2.2.3, to motivate our approach, an intuition of the differences in performance between both methods in a case of time series is given below.

By definition, the energy statistic compares the average of the means of the within-groups distances against the mean of the between-groups distances. Nevertheless, in some scenarios, the value of this statistic might not be large enough to reject the null hypothesis, and larger sample sizes would be required. It is intuitively expected that comparing the whole estimated distributions instead of only their means lead to more powerful tests, and this is precisely the motivation behind of our proposal.

For illustration purposes, we have performed a simple simulated example, which consisted of generating two groups of time series from AR(1) with coefficient $\varphi = 0.3$ (group A) and AR(1) with coefficient $\varphi = 0.8$ (group B) processes, and then computing the interpoint Euclidean distances for all the generated series. Figure 3.1a shows kernel estimators of the corresponding within- and between-samples interpoint densities. Densities instead of distributions are depicted for a neater and clearer view of the differences. Figure 3.1b, where the within-groups distances are joint, provides valuable insight into the behavior of both test statistics. The energy statistic essentially evaluates the distance between the two vertical lines, i.e. it compares the means of the betweenGroups density and the density obtained by joining the withinA and withinB distances. Unlike the energy statistic, the proposed statistic directly compares the whole densities. Clearly, the density shapes differ more substantially than their means of the interpoint distances, thus the difference between densities seems to be still significant even though the energy statistic, based only on the difference between the means and not the whole densities, will not detect significant differences. In short, comparing interpoint densities can produce better discriminatory power than simply comparing their means.



(a)

(b)

Fig. 3.1 (a) Densities of within- and between-samples for series generated from AR(1) $\varphi = 0.3$ (A) and AR(1) $\varphi = 0.8$ (B) processes. (b) Kernel densities of the Euclidean distances between samples and within samples when the latter is obtained by joining the withinA and withinB distances. Vertical lines represent the mean values of both densities. While means may be close, the densities are still easily distinguised.

Scenarios where the mean-based approach works better than the distribution method can indeed appear, for instance when sampling distributions only differ in location. In these scenarios, the energy statistic outperforms the DD statistic in the same way that a two-sample mean test is superior to a general distribution test when only the means are different.

It is important to remark that if a complex distance is introduced, then the resulting interpoint distances may separate groups in more ways than the different location scenario. Indeed, a test robust against this possibility is desirable, even at the cost of discriminatory power in the previously mentioned scenario. Note that a broad range of different distances and dissimilarity measures have been proposed to deal with time series (see eg. [94]), most of them taking into account the dynamic character of the series.

In essence, our approach considers a Cramér-von Mises type statistic over interpoint distances, which is not to be confused with applying the Cramer-von Mises test to the underlying data. The Cramér-von Mises test has been recently proposed to address the problem of detecting change points in time series [110]. Their target is to test for equality of univariate distributions over two time segments, namely before and after the potential change point. This approach is not valid for the problem we are addressing here, which is multivariate homogeneity testing.

3.3 Simulation study

In this section, the empirical power of the nearest neighbors test (k-NN), the energy test (Energy) and the proposed test based on the distributions of interpoint distances (DD) are compared under different scenarios including both synthetic and real data sets. While Energy and DD are free of tuning parameters, the nearest neighbors test depends on the choice of the number k of neighbors. Székely and Rizzo [127] observed that the first two NN statistics, i.e. considering $k = 1$ and 2 neighbors, failed to achieve reasonable approximations to the nominal significance levels in simulations, thus they decided to use the third nearest neighbor test (3-NN). Based on these arguments, we have also considered $k = 3$ in our experiments. The specific scenarios for our empirical power study are described below.

Scenario S1 Time series realizations of length $T = 50$ are generated from first order autoregressive processes (AR(1)) with autoregressive parameter φ taking values in $\{-0.9, -0.6, -0.3, 0, 0.3, 0.6, 0.9\}$. In all cases, the error term follows a standard normal distribution. Based on these simulated realizations, all pairs of autoregressive models are compared with the three two-sample tests. The Euclidean distance is used as interpoint distance, since it is the one most commonly associated with the distance-based tests and the default choice when no information is available about the underlying distributions.

Scenario S2 Practitioners may be only interested in the temporal behavior of the series, thus they often standardize data to cancel shift and scale differences. In order to measure the sensibility of the tests in this setting, we have considered the same experimental framework as S1, but adjusting the models to have the same marginal mean and variance.

Scenario S3 The third experiment is conducted to assess the effect of using different dissimilarity measures specifically designed to deal with time series. Some of these dissimilarities are based on assuming specific generating models, e.g. ARMA models. For this reason, besides considering particularly interesting comparisons of normalized AR(1) models, GARCH(1,1) models are also included in order to examine the behavior of the test statistics under more complex dependence structures. Compared to the results from the above scenarios (S1 and S2) based on the Euclidean interpoint distance, results from S3 should show the positive effect on the two-sample tests of regarding specific time series distances. In this scenario, comparisons were carried out for series with lengths T =50, 100 and 1000.

Scenario S4 This scenario consists of real data examples taken from the UCR Time Series archive [30] using different dissimilarities. The selected datasets cover a range of series' lengths from 96 to 720, and do not require either very small or very large sample sizes to produce meaningful method and dissimilarity comparisons. Note that these data sets are primarily used for classification: each time series belongs to a class. In order to simulate a two sample test for a given dataset, two random subsets were selected for each repetition of the experiment, each subset belonging to one of the two classes of the dataset. When the datasets had more than two classes, the first and second classes (usually labeled using integers) were used.

To gain insight into the effect of the dissimilarity measure on the tests power, different dissimilarities between time series were considered in the Scenarios S3 and S4, which are briefly presented below.

- *Cepstral-based distance* (AR.LPC.CEPS). Euclidean distance between Linear Predictive Coding (LPC) cepstral coefficients, which are derived from the AR(p) models fitted to the series [69].

- *Distance based on local linear smoothers of spectra* (SPEC.LLR). Following Vilar and Pértega [140], we also consider the spectral disparity measure given by

$$d(\mathbf{X},\mathbf{Y}) = \frac{1}{4\pi}\int_{-\pi}^{\pi} W\left(\frac{\hat{f}_X(\lambda)}{\hat{f}_Y(\lambda)}\right) d\lambda,$$

  where $\hat{f}_X$ and $\hat{f}_Y$ are local lineal least squares smoothers of the corresponding periodograms and $W(\cdot)$ is the spectral divergence function $W(x) = W^*(x) + W^*(x^{-1})$, with $W^*(x) = \log(\alpha x + (1-\alpha)) - \alpha \log x$, for $0 < \alpha < 1$.

- *Integrated periodogram based distance* (INT.PER). Since the cumulative version of the periodogram completely determines the generating stochastic process, Casado de Lucas [23] propose to use the distance given by

$$d(\mathbf{X},\mathbf{Y}) = \int_{-\pi}^{\pi} |F_X(\lambda) - F_Y(\lambda)|\, d\lambda,$$

where $F_X(\lambda_j) = C_X^{-1} \sum_{i=1}^{j} I_X(\lambda_i)$, with $I_X(\lambda_i)$ the periodogram evaluated at the i-th Fourier frequency, and $C_X = \sum_i I_X(\lambda_i)$ (normalized version) or $C_X = 1$ (non-normalized version), and analogously for $\mathbf{Y}$.

- *Divergence between permutation distributions* (PDC). Dissimilarity is measured in terms of divergence between permutation distributions of order patterns in m-embedding of the original series. An m-dimensional embedding takes the form $\mathcal{X}'_m \equiv \left\{X'_m = (X_t, X_{t+1}, \ldots, X_{t+m}), t = 1, \ldots, T-m\right\}$. Then, for each $X'_m \in \mathcal{X}'_m$, permutation $\Pi\left(X'_m\right)$ obtained by sorting X'_m in ascending order (so-called *codeword of* X'_m) is recorded, and the distribution of these permutations on $\mathcal{X}'_m$, $P(\mathbf{X})$ (so-called *codebook of* X_T), is used to characterize the complexity of $\mathbf{X}$. PDC approach [20] consists in measuring the dissimilarity between $\mathbf{X}$ and $\mathbf{Y}$ as the α-divergence between their codebooks $P(\mathbf{X})$ and $P(\mathbf{Y})$, respectively.

- *A complexity-invariant dissimilarity measure* (CID). Batista et al. [12] propose to use information about complexity difference between two series as a correction factor for existing dissimilarity measures. Thus, CID takes the general form

$$d(\mathbf{X}, \mathbf{Y}) = CF(\mathbf{X}, \mathbf{Y}) \cdot d(\mathbf{X}, \mathbf{Y}),$$

where $d(\mathbf{X}, \mathbf{Y})$ denotes a conventional raw-data distance and $CF(\mathbf{X}, \mathbf{Y})$ is a complexity correction factor given by

$$CF(\mathbf{X}, \mathbf{Y}) = \frac{\max\{CE(\mathbf{X}), CE(\mathbf{Y})\}}{\min\{CE(\mathbf{X}), CE(\mathbf{Y})\}},$$

with $CE(\mathbf{X})$ a complexity estimator of $\mathbf{X}$. In Batista et al. [12], the complexity estimator is $CE(\mathbf{X}) = \sqrt{\sum_{t=1}^{T-1} (X_t - X_{t+1})^2}$.

- *Dynamic Time Warping distance* (DTWARP). This is one of the most widely employed distances between time series. DTWARP [15] is based on finding an alignment r (so-called *warping path*) between two series under certain restrictions such that a pre-specified point-to-point distance is minimized. For instance, if the L^1 distance is used as local distance, then DTWARP is given by

$$d(\mathbf{X}, \mathbf{Y}) = \min_{r \in \mathcal{M}} \left(\sum_{i=1,..,m} |X_{a_i} - Y_{b_i}| \right),$$

where $\mathcal{M}$ denotes the set of all sequences of m pairs $((X_{a_1}, Y_{b_1}), ..., (X_{a_m}, Y_{b_m}))$, with $a_i, b_j \in \{1, ..., T\}$ such that $a_1 = b_1 = 1$, $a_m = b_m = T$, and $a_{i+1} = a_i$ or $a_i + 1$ and $b_{i+1} = b_i$ or $b_i + 1$, for $i \in \{1, .., m-1\}$.

These dissimilarities have been selected as representatives of broad categories established in the literature on the topic. Thus, PDC belongs to the complexity-based dissimilarities group, AR.LPC.CEPS for the model-based, CID and DTWARP for their complexity and time warping invariances, and SPEC.LLR and INT.PER arise from comparisons in the frequency domain. A comprehensive overview of different approaches to measure dissimilarity between time series can be seen in [94] and references therein. It is worth to highlight that this study is not intended as a comparison of dissimilarities but as evidence of the benefits of introducing specialized dissimilarities in the context of two-sample testing with time series.

Sample size was $n = m = 20$ for Scenarios S1, S2 and S3. In S4, the sample sizes were $n = m = 15$ except for the *ScreenType* and *TwoPatterns* datasets, both with $n = m = 25$. Experiments were replicated 500 times to approximate the rejection proportion.

3.3.1 Results

Results from experiments in Scenarios S1 and S2 are summarized in Table 3.1, where the rejection rates at the nominal level $\alpha = 0.05$ are shown.

Results from S1 show that the proposed method is the best in this scenario, except for the comparisons -0.6 vs 0.6, where 3-NN takes a slight advantage, and -0.3 vs 0.3, where 3-NN fairly outperforms DD. The power obtained by the energy test is less than or equal to the one achieved by DD in all the considered situations. It is worth to analyze some particular results such as the power values attained in the comparisons -0.6 vs 0.3 and -0.6 vs 0.6. Despite the fact that the AR coefficients are more distant in the second case, energy and DD methods produce higher powers in S1. This is explained by the difference of variance of the generating processes, greater in the first scenario. It is observed that the larger variance difference the higher power, thus accounting for the power values equal to one when one of the autoregressive coefficients is $\varphi = 0.9$ or -0.9 and the largest variance differences are attained. The fact that under the Euclidean distance the differences in distributional properties such as variance overshadow stronger differences in temporal behavior highlights the importance of using a suitable interpoint distance regarding the underlying dynamic.

Table 3.1 Rejection rates at level $\alpha = 0.05$ for comparing two groups of time series simulated from AR(1) processes with autoregressive coefficients φ_X and φ_Y, respectively (Scenario S1), and the same models after standardization (Scenario S2).

Generating processes			Scenario S1			Scenario S2		
φ_X	vs	φ_Y	3-NN	Energy	DD	3-NN	Energy	DD
-0.9	vs	-0.9	0.05	0.05	0.06	0.05	0.06	0.05
		-0.6	0.97	0.96	1.00	0.18	0.09	0.14
		-0.3	1.00	1.00	1.00	0.29	0.13	0.33
		0.0	1.00	1.00	1.00	0.50	0.16	0.61
		0.3	1.00	1.00	1.00	0.84	0.22	0.81
		0.6	1.00	1.00	1.00	1.00	0.26	0.97
		0.9	1.00	0.36	1.00	1.00	0.38	1.00
-0.6	vs	-0.6	0.05	0.06	0.06	0.06	0.05	0.05
		-0.3	0.26	0.11	0.82	0.11	0.06	0.07
		0.0	0.68	0.27	0.99	0.39	0.08	0.17
		0.3	0.98	0.23	0.99	0.91	0.10	0.47
		0.6	1.00	0.10	0.90	1.00	0.15	0.90
		0.9	1.00	1.00	1.00	1.00	0.27	0.96
-0.3	vs	-0.3	0.06	0.03	0.04	0.06	0.06	0.06
		0.0	0.11	0.06	0.14	0.10	0.06	0.08
		0.3	0.43	0.07	0.16	0.43	0.07	0.16
		0.6	0.98	0.27	0.99	0.92	0.10	0.48
		0.9	1.00	1.00	1.00	0.86	0.19	0.81
0.0	vs	0.0	0.07	0.06	0.06	0.05	0.05	0.05
		0.3	0.10	0.06	0.12	0.10	0.07	0.08
		0.6	0.72	0.25	0.99	0.40	0.08	0.16
		0.9	1.00	1.00	1.00	0.52	0.18	0.64
0.3	vs	0.3	0.05	0.04	0.04	0.05	0.05	0.05
		0.6	0.28	0.14	0.83	0.11	0.06	0.07
		0.9	1.00	1.00	1.00	0.33	0.14	0.35
0.6	vs	0.6	0.06	0.07	0.06	0.04	0.05	0.05
		0.9	0.96	0.96	1.00	0.21	0.09	0.13
0.9	vs	0.9	0.04	0.06	0.06	0.06	0.05	0.05

By construction, the results from Scenario S2 are free from the effect of variance differences. Unlike Scenario S1, the power always increases with the difference between autoregressive coefficients in S2. Nevertheless, the lack of differences in variance complicates the discrimination task. In fact, a drastic reduction of the power of the tests is observed for all the procedures, thus illustrating the difficulty of comparing normalized time series. Beyond these arguments, the results of the nearest neighbor approach are superior on average to the ones obtained with the proposed test, although there exist comparisons where DD performs better than 3-NN. The energy statistic leads to the lowest powers. Note that the rejection rates under the null are very close to the nominal size for all statistics and experiments.

In Scenario S3, the two-sample tests were executed considering groups of time series generated from the following models:

- S3.1 Normalized series coming from two AR(1) models with autoregressive coefficients $\varphi_X = 0.2$ and $\varphi_Y = 0$, respectively. Therefore, we evaluate the capability of the test statistics to discriminate between a weak linear dependence structure and independent realizations.

- S3.2 Series coming from generalized autoregressive conditional heteroskedasticity GARCH(1,1) models with different parametric specifications. More specifically, the generating processes take the form $X_t = \sigma_t \varepsilon_t$, where the conditional variance is modeled by
$$\sigma_t^2 = \omega + \alpha\varepsilon_{t-1}^2 + \beta\sigma_{t-1}^2,$$
for different values of (ω, α, β). This way, more complex dependence structures are involved in our experiments.

Again, the errors are realizations from a white noise process with mean 0 and variance 1. In the present scenario, besides the Euclidean distance, we also use other dissimilarity measures aimed at comparing spectral approximations. The choice of these structure-based dissimilarities is supported by dealing with stationary models, without particular interest to detect differences in shape, complexity or specific invariances. The estimated rejection rates for a significance level $\alpha = 0.05$ are summarized in Table 3.2.

The effect of considering different dissimilarities in the normalized autoregressive scenario S3.1, AR(1) $\varphi = 0.2$ versus AR(1) $\varphi = 0$, is especially illustrative. When using the Euclidean distance, all methods present very low discriminatory capability. The power is substantially improved by considering distances like SPEC.LLR and AR.LPC.CEPS, which measure dissimilarity in terms of nonparametrically estimated spectra and estimated cepstral coefficients, respectively. The optimal properties of these distances to deal with these models are inherited by the three test statistics, resulting in higher powers. It is worthy to remark that, if the Euclidean distance is considered, the 3-NN method is the best one, but energy and D-D statistics outperform 3-NN when SPEC.LLR and AR.LPC.CEPS are used. This result, which can be extrapolated to

Table 3.2 Rejection rates at level $\alpha = 0.05$ in Scenario S3. T denotes the length of the series and "Norm." indicates if the series were or not normalized.

	T	Norm.	Dissimilarity	3-NN	Energy	DD
S3.1 AR(1) parameter φ						
0.2 vs 0.0	50	Yes	EUCL	0.06	0.03	0.03
	50	Yes	SPEC.LLR	0.62	0.92	0.83
	50	Yes	AR.LPC.CEPS	0.76	0.95	0.91
S3.2 GARCH(1,1) parameters (ω, α, β)						
(0.2, 0.05, 0.7) vs (0.1, 0.1, 0.8)	50	No	EUCL	0.05	0.07	0.28
(0.2, 0.1, 0.7) vs (0.1, 0.15, 0.7)	200	No	EUCL	0.10	0.27	0.88
	1000	No	EUCL	0.00	0.60	1.00
(0.1, 0.7, 0.2) vs (0.05, 0.65, 0.15)	1000	Yes	EUCL	0.04	0.03	0.04
	1000	Yes	SPEC.LLR	0.14	0.08	0.39
	1000	Yes	AR.LPC.CEPS	0.10	0.30	0.43

the whole of S3, shows that a dissimilarity measure may differentiate samples in ways which some tests are not sensible to. Coming back to the AR processes comparisons, it is observed that energy slightly outperforms DD. Mainly, energy takes a greater advantage of the dissimilarity properties. For instance, standardization coupled with the coefficients extracted by the AR.LPC.CEPS dissimilarity creates the ideal location-separated scenario mentioned in Section 3.2. On the other hand, in the GARCH scenarios (S3.2) DD outperforms energy because the dissimilarities do not produce this ideal situation.

Results by comparing non-normalized GARCH models reveal the importance of having substantially larger sample sizes to reach reasonable power levels. In particular, we have observed that 3-NN fails to achieve the nominal value, requiring fine-tuning of the parameter k, one of the drawbacks of this method. This effect occurs for other GARCH experiments of length $T = 200$, not shown here. In the normalized, length $T = 1000$ GARCH scenario, using other distances increases the discriminatory power and again DD is the method taking the most advantage of the new distances. Noticeably, 3-NN benefits more than energy with SPEC.LLR, while the opposite happens with AR.LPC.CEPS.

The results for the real data comparisons (Scenario S4) are given in Table 3.3. In S4, we face very different kinds of series, including non-stationary series and models characterized by differences in shapes, complexity, ... This accounts for the use of dissimilarity measures specifically designed to capture these features. We do not intend to determine the optimal dissimilarity for each single comparison, but highlighting the improvement in terms of power when non-Euclidean distances are regarded in the test statistics. Table 3.3 shows at a glance how this improvement in the rejection rates is attained with few exceptions. The results also show that the effect of a dissimilarity is not uniform across the studied methods, e.g. while all methods benefit from the PDC dissimilarity in the RefrigeratorDevices dataset, the 3-NN with PDC decreases its power in the TwoPatterns dataset, when compared with the Euclidean distance. Once again, it is shown that DD is either the best performing or highly competitive test in all the considered datasets when a suitable dissimilarity is considered.

The UCR time series datasets are used for classification, classes are presumed to be separable. Nevertheless the two-sample problem may include different scenarios, such as comparing groups consisting of the same two classes, but in different proportions. In the last row of Table 3.3, mixtures of the first classes of two datasets in different proportions are compared, showing that 3-NN method is not particularly suited for this scenario.

Table 3.3 Rejection rates at level $\alpha = 0.05$ in Scenario S4 involving real datasets from the UCR Time series archive.

Datasets	Dissimilarity	3-NN	Energy	DD
ScreenType	EUCL	0.10	0.10	0.08
	INT.PER	0.21	0.09	0.26
RefrigerationDevices	EUCL	0.06	0.04	0.04
	PDC	0.95	0.97	0.99
ShapeletSim	EUCL	0.05	0.04	0.04
	CID	0.86	0.98	0.97
ToeSegmentation1	EUCL	0.12	0.06	0.05
	DTWARP	0.86	0.73	0.78
TwoPatterns	EUCL	0.45	0.20	0.10
	PDC	0.15	0.22	0.21
	SPEC.LLR	0.40	0.58	0.60
ElectricDevices	EUCL	0.10	0.21	0.48
	PDC	1.00	1.00	1.00
ShapeletSim & Twopatterns Mixtures (0.6,0.4) vs (0.8,0.2)	SPEC.LLR	0.33	1.00	1.00

3.4 Applicative example

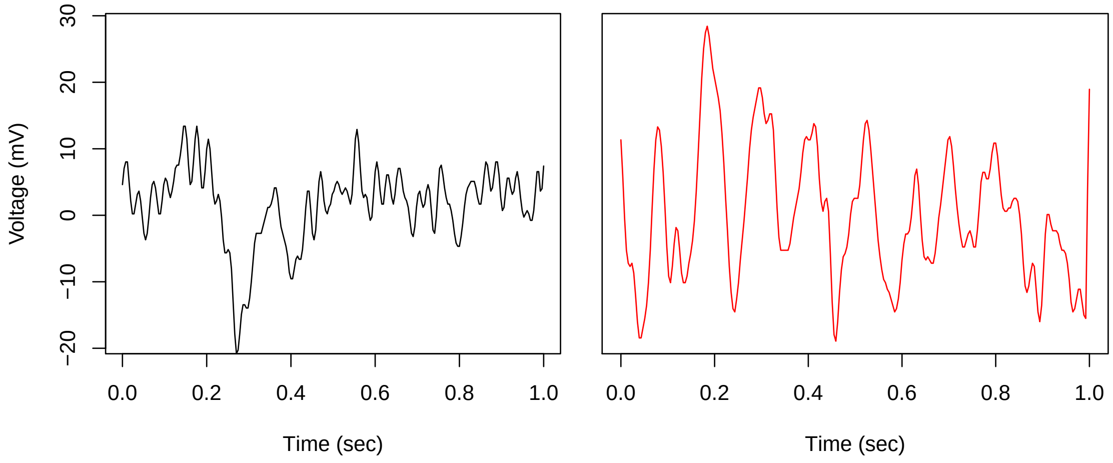
In this section we test the equality of distributions of electroencephalogram (EEG) time series coming from two groups: alcoholic and control subjects. In addition to the tests methods, time-series specific distances are introduced to illustrate the influence in test power of this parameter.

The data were gathered in a experiment to measure correlation of the EEG response with genetic predisposition to alcoholism (see [79] for data and description). Subjects belonged to either alcoholic or control groups and were exposed to different kinds of visual stimulus. Each EEG reading corresponds to a 1 second interval after the presentation of the stimulus, sampled at 256 Hz, therefore producing a 256 length time series. 64 positions (channels) in the subject's scalp were measured, producing a 64-variate series for each subject-stimulus presentation trial.

We divide the data into two classes, the EEGs produced by alcoholic subjects and the EEG produced by subjects in the control group. The two-sample tests are applied to random samples of these two groups. Therefore, we check whether EEG readings taken at random from alcoholic subjects differ in distribution from EEGs taken at random from control subjects. Note that we ignore subject, stimulus and channel dependencies (considering each of the 64 channels as individual time series) for the purposes of this experiment. The difference between alcoholic and control groups is one of the many that may be checked with this data, such as difference between two specific channels, test subjects, stimulus type, etc.

The division produces 450048 series in the alcoholic group and 257536 control group. Figure 3.2 illustrates series in each group. Missing data were assigned the series' median value and series with constant values were removed. We apply the three tests: Energy, Nearest Neighbors with parameter $r = 3$ and the proposed Distance Distribution (DD). Three distances are also combined with the methods. The Euclidean distance is used as reference, the Periodogram based distance proposed in [22] and the Complexity Invariant Distance (CID) [12] are included as examples of time series specific distances.

Table 3.4 shows the average power at sample sizes $N = M = 50$ and $\alpha = 0.05$ level for 1000 repetitions of the experiment. When comparing methods, our proposed approach DD achieves higher power for all distances. Distance-wise, the Peridogram based distance produces greater power when compared to the Euclidean and CID distances. Given the nature of the data, it is not surprising that a frequency domain distance is more suitable for finding differences in this dataset. On the other hand, the CID approach, designed to scale a given underlying distance (in this case the Euclidean) by the difference of complexity between series, decreased the power in relation to the raw



(a) (b)

Fig. 3.2 (a) EEG reading of an alcoholic subject. (b) EEG reading of a control subject.

Euclidean. In this scenario, weighting complexity over other series' features is similar to increasing noise, since it masks other differences that are better captured by the Periodogram based distance and even the Euclidean.

Table 3.4 Power results for comparing alcoholic and control subjects EEG readings. Nearest Neighbors, Energy and the proposed Distance Distribution methods are combined with Euclidean, Periodogram and CID distances to produce 9 testing approaches.

Distance	3-NN	Energy	DD
Euclidean	0.17	0.18	0.35
Periodogram	0.24	0.53	0.58
CID	0.09	0.11	0.14

3.5 Conclusions and further research

We have addressed the problem of testing for homogeneity of two sets of time series and studied the behavior of our test proposed in Chapter 2. Like other multivariate two-sample procedures based on distances, the proposed test presents the property of reducing the dimension of the problem, particularly desirable here because of the length of the series usually exceeds the number of series. The main novelty lies on how the test statistic handles the interpoint distances. Our strategy consists in comparing the whole empirical distributions of interpoint distances within- and between-groups. Considering the whole distributions allows us to obtain a more complete information than simply using a number of specific moments of these distributions. This way the test is not only sensitive to differences in location or spread but also allows to identify differences in the shapes of the distributions. Compared to other alternative statistics, our proposal attained the best performance in many of the simulated scenarios, and produced competitive results in the rest.

On the other hand, a testing procedure based on interpoint distances acquires particular relevance in the time series framework. This is due to the broad range of dissimilarity measures introduced in the literature to assess discrepancy between time series models. Thus, it is expected that a suitable choice of the distance between series will increase the discriminatory power of the proposed test. This intuition has been fully confirmed in our experiments. Compared to the standard multivariate approach based on the Euclidean distance, our experiments have shown that a well-selected time series dissimilarity can substantially improve the performance of the test. As an example, the proposed test was fairly the most powerful when complex dependence structures such as conditionally heteroskedastic models were considered. The importance of using a proper distance was also highlighted by the results obtained with all the real datasets involved in the experiments. In sum, similarly as distances between time series play a key role in tasks of clustering and classification, they are extremely useful to improve the power of the proposed test. In particular, the results presented in this work open the possibility of performing time series two-sample tests in situations where the discriminatory power is low.

There are a number of possible directions for further research. It is interesting to assess the performance in terms of power of the different ways of comparing interpoint distributions, for example by using nonparametric densities instead of empirical distributions. An automatic criterion to select a suitable dissimilarity in terms of testing power when no prior knowledge on the generating processes is available would provide a valuable tool. From a theoretical point of view, the results by Maa et al. [85] establish requirements on the interpoint distances and it is of great interest to analyze whether some commonly used time series dissimilarities satisfy these constraints.

Chapter 4

A distance for time series and shapes based on lagged distributions

4.1 Introduction

Time Series Classification and Clustering are two of the main tasks of time series analysis, spanning many fields of application. The main approach to Time Series Clustering consists of using a dissimilarity to compare time series and then applying general purpose distance-based clustering methods that work directly on the comparisons [5, 94]. Similarly, the state of the art approach in Time Series Classification [82] is an ensemble of 35 individual classifiers, 11 of which are distance-based. Many of the distances in time series can be applied to both tasks.

The claim that a method is state-of-the-art in classification is based on comparing its classification accuracies against other methods in the UCR/UEA time series repository [9]. This repository features classification problems from a wide range of domains including medicine, engineering, robotics, etc. Some of the datasets in this repository can be considered *shape* classification, dealing with silhouettes of objects, such as leaves or bones of animals, and curves coming from spectrography. We make the distinction of shapes from time series to highlight this possibility of application, even though it is implicit since these objects are considered as time series by the community, often under the umbrella term of *1D signals* or *sequential data.*

Methods in time series analysis can be loosely categorized as model-free or model-based.

One of the best performing "families" of model-free methods in time series classification are the so called "Bag of Patterns" [113] and "Bag of Features" [114]. These methods are based on the sliding window approach, extracting all subsequences of the series of a given length. Once the subsequences have been extracted, they are strongly quantized to transform them into categorical data. The purpose of this categorization is to represent each series by the relative frequency of these patterns in a way that is easily compared, i.e. the histogram. An example of such categorization process is the K-means algorithm, each subsequence is represented by the label of the cluster it has been assigned to. The term "bag" (i.e. multiset) comes from the fact that the order of these patterns inside the series is not considered. Once the series have been transformed to histograms, they are compared using distances between histograms, such as the L1 distance. While "Bag of Patterns" methods try to keep the subsequences, "Bag of Features" extracts features from them, such as their mean and variance, prior to the quantization step. These bag-based methods can be then considered as disimilarities between time series, since they compare histograms by means of distances.

Introducing models from the time series analysis literature to the classification and clustering domains has originated the model-based family of distances [69, 86, 139, 101]. Time series models are well understood, which translates into two main benefits: helps prediciting in which scenarios they will be useful and enhances the interpretability of results. These distances achieve very good performance when the underlying assumptions are met, such as economic time series after suitable transforms.

We observe that both bag-based and model-based methods have one thing in common, they rely on the idea of capturing the "generating process" or "autoregressive distribution" of the series (albeit in different ways), and then measuring the difference between these processes or distributions. Distances based on linear autoregressive models do it parametrically, describing the distribution with a few parameters, and "Bag of Patterns" methods do it non parametrically, the size of the sliding window serving a similar purpose to the lags selected in AR models.

All of these methods also perform very strong simplifications of the series, for several reasons:

- Noise Reduction: Information not considered relevant is discarded. Model-based methods separate the series into the deterministic and error parts, and the error part is not used for the similarity of the series. "Bag of Patterns" methods smooth the subsequences, e.g. by removing the high frequency Fourier coefficients.

- Computational Reasons: It can be unfeasible to analyze very long time series under these methods. Limiting the number of lags to test in model-based methods or the categories in the bag-based methods improves their computation.
- Technical limitations: Bag-methods were originally proposed to process text strings, "bag of words" [92], where there is no need for more complex tools beyond histogram. Adaptation of the "bag of words" methods to time series translate their problem to the original idea, even at the cost of discarding information. Model-based approaches often specify models even when they are not a perfect match to the data, for lack of a better solution.

We propose a new distance based on comparing the lagged distributions of the series. Our objective is to define a distance that inherits the good empirical performance of bag-based methods and model-based methods but in some sense is free from the limitations imposed by either family, unifying their underlying ideas into a single approach. At the same time it can arguably be considered more parsimonious: it does not assume a parametric model, does not extract features and does not apply simplifications such as strong quantizations which may be harmful in some domains of application. We also focus on discriminating instead of modeling: model-based approaches try to fit individual time series to the model that best predicts them. In the setting of classification and clustering, discrimination between series is required, not prediction, and therefore performance is expected to improve if we focus on the more specific objective. This is the same concept behind Support Vector Machines modeling the frontier between classes rather than the density function of the each of the classes. Vapnik [137] summarized this idea as "Do not solve a harder intermediate problem".

For parsimoniousness, instead of a sliding window, we use lag embedding with few lags. The lags for the embedding are selected in a data-driven way, growing a set of lags instead of using full subsequences of the sliding window. Biasing towards sparsity also helps with interpretability, as a set of lags maybe interpreted as some kind of seasonality or pattern. On the other hand, the full consecutive set of lags that is a sliding window can also be reached by our data-driven method. In the context of clustering, it is also possible that the practitioner is interested in introducing a given set of lags with some meaning in the domain of application.

To remove assumptions and strong simplifications required to produce histogram as the representation in the bag-based methods, we use a divergence between multivariate distributions, originally used as a statistic for two-sample hypothesis testing. This allows us to compare the lag embedding of two time series even when they are of high dimension (a large number of lags) and also keeps the notion of similarity between patterns that is lost when quantizing.

We detail the process of lag-embedding, the distance between lag embeddings and provide a data-driven method for lag selection in Section 4.2. We give an in-depth discussion of the application of our distance to shape classification in Section 4.3. Section 4.4 shows empirical results for clustering and classification applied to real and simulated data and Section 4.5 includes some conclusions and future work.

4.2 A distance between lag embeddings

The distance between time series proceeds by two steps:

1. Each series is transformed to its lag embedding representation (Section 4.2.1).

2. A distance between lag embeddings is calculated (Section 4.2.2).

The only input parameter is the set of lags to create the lag embedding, we also propose a data-driven method for finding the lags.

4.2.1 Lag embedding of a time series

Let $\mathbf{X}$ be a time series of length T, $\mathbf{X} = (X_1, X_2, \ldots, X_T)$. Then given an ordered set of lags $\mathbf{L}$, $\mathbf{L} = (L_1, \ldots, L_D)$ the lag embedding of X using $\mathbf{L}$ is defined as:

$$S_{\mathbf{X},\mathbf{L}} = \begin{pmatrix} X_1 & X_{1+L_1} & \cdots & X_{1+L_D} \\ X_2 & X_{2+L_1} & \cdots & X_{2+L_D} \\ \vdots & \vdots & \vdots & \vdots \\ X_{T-L_D} & X_{T-L_D+L_1} & \cdots & X_T \end{pmatrix}$$

For the special case of an empty set of lags, the lag embedding is just the series:

$$S_{\mathbf{X},\emptyset} = \begin{pmatrix} X_1 \\ X_2 \\ \vdots \\ X_T \end{pmatrix}$$

Therefore, for a given sequence of lags $\mathbf{L} = (L_1, \ldots, L_D)$, a lag embedding of an observed time series consists in representing the time series in a D-dimensional time delayed space whose k-th component contains the L_k-step-ahead realizations of the original data.

It can be seen that a sliding window approach can be recreated with the given definition of lag embedding by considering $\mathbf{L}$ to be a consecutive set of lags, e.g. for a window size of 10, $\mathbf{L} = (1,2,3,4,5,6,7,8,9)$.

The lag embedding we are proposing is related to the widely used technique in nonlinear time series analysis, most notably in chaotic systems, called time delay embedding for state space reconstruction. In state space reconstruction, scalar time series are embedded in a multidimensional space using time delayed versions of the measured quantities. Among the techniques from state space reconstruction, our proposal is more closely related to non-uniform time delay embedding [120].

4.2.2 Comparing lag embeddings

Given a set of lags $\mathbf{L}$ and two series $\mathbf{X}$ and $\mathbf{Y}$, represented by their respective lag embeddings $S_{\mathbf{X},\mathbf{L}}$ and $S_{\mathbf{Y},\mathbf{L}}$, we propose to measure the distance between the lag-embeddings by using the energy statistic [127], originally proposed for two-sample hypothesis testing and previously described in the above chapters. In this particular case, regarding that the two samples are actually the sets of the lag embeddings, the proposed distance takes the form:

$$\begin{aligned} E = \frac{(T-L_D)^2}{2(T-L_D)} \Bigg(& \frac{2}{(T-L_D)^2} \sum_{i=1}^{T-L_D} \sum_{j=1}^{T-L_D} d(S_{\mathbf{X},\mathbf{L}}i, S_{\mathbf{Y},\mathbf{L}}j) \\ & - \frac{1}{(T-L_D)^2} \sum_{i=1}^{T-L_D} \sum_{j=1}^{T-L_D} d(S_{\mathbf{X},\mathbf{L}}i, S_{\mathbf{X},\mathbf{L}}j) \\ & - \frac{1}{(T-L_D)^2} \sum_{i=1}^{T-L_D} \sum_{j=1}^{T-L_D} d(S_{\mathbf{Y},\mathbf{L}}i, S_{\mathbf{Y},\mathbf{L}}j) \Bigg) \end{aligned} \tag{4.1}$$

with d the Euclidean distance and $S_{\mathbf{X},\mathbf{L}}i$ the i-th row of $S_{\mathbf{X},\mathbf{L}}$.

We are interested in E not for testing, but as a measure of divergence between $S_{\mathbf{X},\mathbf{L}}$ and $S_{\mathbf{Y},\mathbf{L}}$, sets of multivariate observations of arbitrarily large but finite dimension. E is zero when the two sets are equal and generally grows the more different the two sets are. This allows us to use E in most distance-based methods, from classification (e.g. nearest neighbors) to clustering (e.g. hierarchical clustering).

Compared to approaches like “Bag of Patterns” or “Bag of Features”, the main benefit of using E is that it is not limited to univariate observations, so extra dimension reduction or categorization is not required for the lag embeddings to be compared.

Recently, a number of statistics from two-sample testing have been proposed that can be used as divergences in this step ([51, 143]), including the one we propose in Chapter 2. We use E due to its simplicity and computational efficiency, and a more thorough discussion of the choice of divergence is left for future work.

4.2.3 Lag and embedding dimension selection

Similarly to the sliding window size, the order of autoregressive processes or embedding dimension on state space reconstruction, our technique assumes a set of lags $\mathbf{L}$ to create the lag embedding. In some scenarios $\mathbf{L}$ may be given, e.g. in clustering we can be interested in how the data is clustered under a given lag that has some meaning in our context. Nevertheless, it is helpful to provide an automatic method for finding a good performing $\mathbf{L}$. Motivated by its good empirical evidence, interpretability and computational reasons, we propose a greedy approach for calculating $\mathbf{L}$. Whether for classification or clustering, our approach adds lags to the set if this addition improves an objective function. A pseudocode for lag selection is given in Algorithm 1.

Additional hyperparameters can be considered, such as declaring a maximum size of $\mathbf{L}$, a maximum lag value or fixing the number of clusters.

Unlike related techniques in state space reconstruction that try to find the set of lags that best models each individual series such as the false nearest neighbors [107], or dimension deviation [131], our method optimizes the performance measure of interest.

When the set of lags is the empty set, we are comparing the marginal distributions of the series with respect to time. When the set of lags is the whole series, E is equivalent to the Euclidean distance between the series.

4.3 A motivating example

We show a real data example in shape classification to illustrate a full process of analysis using our distance. We show how some of the ideas behind our distance relate to a reference approach in shape analysis, and measure how the performance is improved with regards to the reference.

The data are shapes of otoliths (bonelike structures in the ear) of fishes that have been digitalized by taking pictures. The goal is to determine which family of fish a

Algorithm 1 Greedy lag selection algorithm

```
Inputs:
    $M$: the dataset of time series of length $N$.
    $E$: the energy distance defined in Section 4.2.2.
Output:
    $\mathbf{L}$ the set of lags to be considered.
$\mathbf{L} \leftarrow \emptyset$
$g \leftarrow$ The "goodness" of the solution under $\mathbf{L}$.
repeat
    $continueSearch \leftarrow$ FALSE
    for $l = 1$ to $N$: do
        if $l$ not in $\mathbf{L}$ then
            $D \leftarrow$ the pairwise distance matrix, calculated from the $E$ distance between all possible pairs of time series in $M$ using the lag embedding $\mathbf{L} \cup l$.
            $gtemp \leftarrow$ A measure of "goodness" of the solution of $D$. In supervised classification, use Leave-one-out crossvalidation accuracy of a nearest neighbor classifier. For clustering, use a measure of goodness of clustering such as the average silhouette width or Calinski and Harabasz [56].
            if $gtemp > g$ then
                $g \leftarrow gtemp$
                $\mathbf{L} \leftarrow \mathbf{L} \cup l$
                $continueSearch \leftarrow$ TRUE
            end if
        end if
    end for
until $continueSearch ==$ FALSE
```

bone belongs to, given a picture of it. Each otolith belongs to one of three families. We use the AFORO database (http://isis.cmima.csic.es/aforo/) of already classified photos of otoliths. This is a supervised classification problem, with the pictures of otoliths as input and three possible classes, the three families of fish.

The first step is a preprocessing procedure, the pictures are transformed into a data structure that is easier to process. The pictures are transformed to a matrix of binary values, which each position in the matrix indicating whether it is bone or background, see Figure 4.1 for a result of one otolith.

The reference approach in shape analysis consists of randomly sampling points inside the shape (see the red dots in Figure 4.1a) and then representing the shape by the one dimensional density of their interpoint Euclidean distances. The representation of the shapes of the whole dataset as density objects, colored by their family, is shown in Figure 4.1b.

Note how this approach is similar to bag-based methods and our lag embedding in the sense that they represent the shapes by distributional objects, in this case a density. This transform has very desirable properties in shape analysis, it is invariant to rotations, translations and mirroring of the shapes. On the other hand, it only uniquely identifies a shape if it is convex [16].

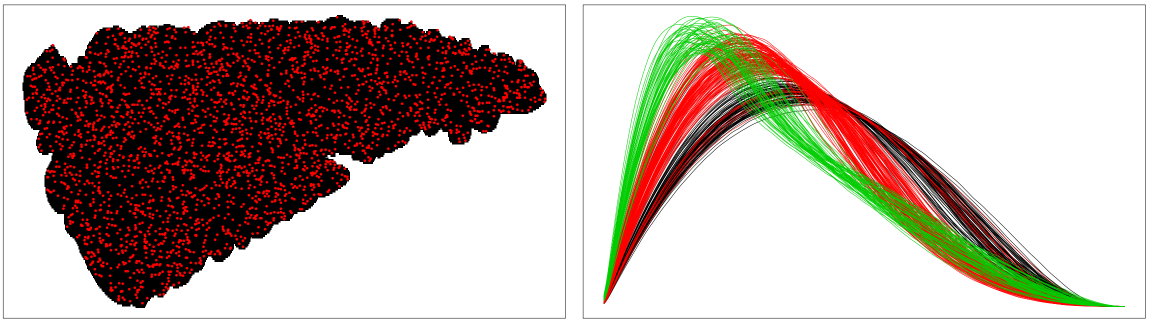


Fig. 4.1 (a) An otolith picture that has been transformed into a matrix of binaries. Red dots represent points randomly sampled inside the shape. The density formed by all pairwise distances between sample points is used to represent the original shape. (b) The results of representing the whole dataset of otoliths as densities. Color represents target class.

One way of using this representation in a classification context is by introducing the L1 distance between the densities [16], together with a distance-based algorithm such as nearest neighbors.

Our lag embedding representation is easier to apply to 1-dimensional curves instead of 2-dimensional shapes. To get curves from the 2D shapes, we consider only the edges of the shapes, and apply an arc length parametrization. The arc length parametrization of the 2D line formed by the edges of the shape consist of two 1D lines, each representing the relation between one of the 2 dimensions of the original curve at the given length of the curve. The arc length parametrization of the shape in Figure 4.1a can be seen in Figure 4.2a.

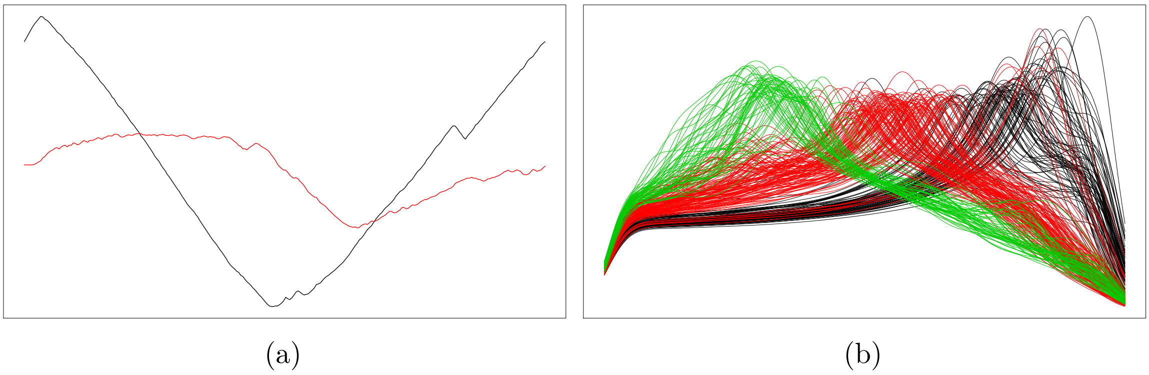
When we apply the reference approach, sampling inside the arc length curves and computing the density of the interpoint Euclidean distances results in the Figure 4.2b. To assess the effect of the arc length parametrization with respect to the original 2D shapes, we first compare the classification accuracy of the reference approach applied to the whole 2D shapes with the same approach but applied to the edges of the shapes, extrated by the arc length method. Accuracies are measured by leave-one-out crossvalidation.

- Reference approach sampling inside the wholeshape: 0.912.
- Reference approach sampling along the edges, parametrized as arc length: 0.923.

We see that the arc lenght parametrization prior to sampling already produces an improvement in classification accuracy in the reference approach. We attribute this effect to the fact than most discriminative information lies in the edges of the shapes, not inside them.

We now apply our distance based on lagged embeddings. A minor detail to considere in this particular example is that the original objects here are two one-dimensional "time series" rather than one. Our distance can be trivially extended in this case by considering each observation in a given time series $\mathbf{X}$ to be a two-dimensional point.

Our lagged distribution representation does not retain the useful invariances of the reference representation, but these invariances can be achieved by finding the best translation/rotation/mirror on a pair by pair basis, finding the best affine transformation from one shape to the other, prior to calculating the distance between the two shapes. The tradeoff here is the higher computational cost of calculating this transformation, though this cost is very small compared to the computation of the actual distance.


(a)

(b)

Fig. 4.2 (a) Arc length representation of the edges of the otolith shape of Figure 4.1a. (b) Interpoint Euclidean distance representation of the shapes of the dataset, after applying the arc length parametrization to them.

When no lags are considered, our approach is also similar to the reference in that they compare sets of points sampled from the curves. Instead of computing the densities for interpoint Euclidean distances of each shape and then comparing them via the L1 distance, our approach compares them directly via the energy statistic E, also based on interpoint Euclidean distances. We sample the curves in a deterministic way, using a grid.

When we consider no lags for the embedding with our distance, classification accuracy improves to 0.956 from the 0.923 of the reference approach that uses the L1 distance between densities interpoint Eculidean distances.

This improvement can be in some way attributed to focusing on discriminating instead of modeling, e.g. modeling the densities requires additional steps such as choosing bandwith that may not be optimal for discrimination.

After this, we consider introducing one lag for the lag embedding of the curves. Figure 4.3 shows the classification accuracy with respect to the time delay that is being considered. It is important to highlight that accuracy is improved for all possible lag values. This reflects the usefulness of lag embedding as a way of comparing shapes. The best accuracy found using only one lag is 0.968, considering two lags for the embedding, it can be improved to 0.97.

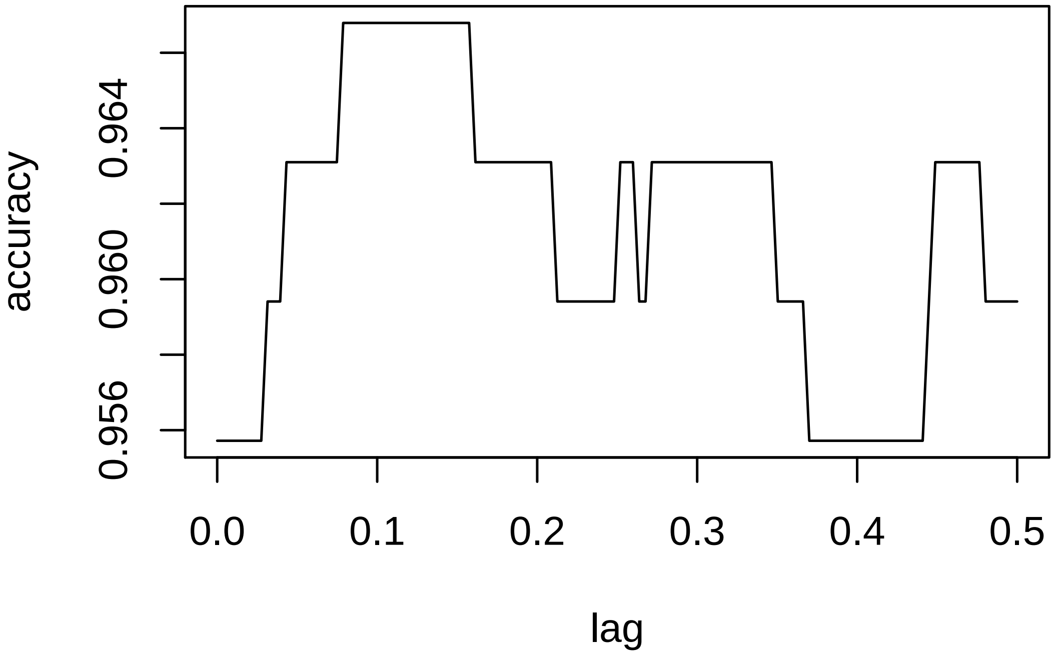


Fig. 4.3 Classification accuracy in the otolith example usign lag embedding limited to only one lag. We can see that all possible values of lag improve (or never decreases) the accuracy.

4.4 Experimental results

We show the empirical performance of our distance for both clustering and classification problems, under real data and synthetic scenarios. For computational reasons, the number of lags selected by the greedy method is limited to 2 in the experiments, and maximum lag is limited to 1/3 of the length of the series.

4.4.1 Real data for classification

We compare our approach with 12 other distances, using the UCR/UEA datasets. These distances have been used for classification in the literature and their results over these datasets are reported in the web page of [9] `timeseriesclassification.com`. All distances are used in a 1-nearest neighbor classifier. We use the same splits for train and test sets as in [9] and show the resulting classification accuracy of 4 of the 12 distances in Table 4.1. The 4 distances shown are the ones with the highest average accuracy. The rest may be seen in timeseriesclassification.com.

We compare only 43 out of the 85 datasets in the repository for computational reasons, these 43 are the fastest to test in time complexity taking into account training and testing sizes and length of the series.

The following groups of distances are included in our comparison

- Euclidean Distance.
- Five Variants of Dynamic Time Warping with different windows selection algorithms, including Weighted Dynamic Time Warping, WDTW [67].
- Longest Common Subsequence, LCSS [141].
- Edit Distance with Real Penalty, ERP [28].
- Complexity Invariant Dynamic Time Warping, dCIDDTW [12].
- Move Split Merge Metric [125].
- Time Warp Edit Distance, TWE [90].
- Derivative Dynamic Time Warping, DD_DTW [49].

Our distance, dLAGDISTR, achieves or ties for best position in 11 of the 43 datasets and has the best average accuracy among the 12.

4.4.2 Simulation

The UCR dataset is comprised of mostly deterministic times series, but it is also interesting to compare performance in series of stochastic nature, when the underlying

	DATASET	dLAGDISTR_1NN	WDTW_1NN	ERP_1NN	TWE_1NN	DD_DTW_1NN
1	Adiac	0.70	0.61	0.61	0.63	0.70
2	ArrowHead	0.80	0.82	0.80	0.79	0.79
3	Beef	0.60	0.70	0.67	0.60	0.67
4	BeetleFly	0.80	0.70	0.75	0.70	0.65
5	BirdChicken	0.95	0.75	0.75	0.85	0.85
6	Car	0.78	0.78	0.77	0.92	0.80
7	CBF	0.99	1.00	1.00	0.99	1.00
8	Coffee	1.00	1.00	1.00	1.00	1.00
9	DistalPhalanxOutlineAgeGroup	0.69	0.70	0.68	0.71	0.71
10	DistalPhalanxOutlineCorrect	0.72	0.72	0.71	0.74	0.73
11	DistalPhalanxTW	0.65	0.60	0.67	0.63	0.61
12	ECG200	0.83	0.88	0.88	0.89	0.83
13	ECGFiveDays	0.85	0.80	0.81	0.83	0.77
14	FaceAll	0.78	0.79	0.79	0.79	0.90
15	FaceFour	0.94	0.88	0.86	0.85	0.83
16	FacesUCR	0.92	0.92	0.92	0.92	0.90
17	GunPoint	0.98	0.98	0.95	0.95	0.98
18	Ham	0.54	0.58	0.60	0.51	0.48
19	Herring	0.56	0.53	0.62	0.52	0.55
20	ItalyPowerDemand	0.96	0.95	0.96	0.95	0.95
21	Lightning2	0.84	0.90	0.87	0.84	0.87
22	Lightning7	0.63	0.77	0.74	0.75	0.67
23	Meat	0.93	0.93	0.93	0.93	0.93
24	MedicalImages	0.71	0.74	0.68	0.71	0.74
25	MiddlePhalanxOutlineAgeGroup	0.52	0.52	0.48	0.52	0.54
26	MiddlePhalanxOutlineCorrect	0.75	0.75	0.77	0.76	0.73
27	MiddlePhalanxTW	0.50	0.51	0.50	0.49	0.49
28	MoteStrain	0.88	0.86	0.87	0.80	0.83
29	OliveOil	0.83	0.83	0.87	0.87	0.83
30	PhalangesOutlinesCorrect	0.75	0.75	0.76	0.76	0.74
31	Plane	0.96	1.00	1.00	1.00	1.00
32	ProximalPhalanxOutlineAgeGroup	0.79	0.80	0.80	0.80	0.80
33	ProximalPhalanxOutlineCorrect	0.84	0.78	0.81	0.81	0.79
34	ProximalPhalanxTW	0.71	0.75	0.70	0.75	0.77
35	SonyAIBORobotSurface1	0.87	0.74	0.69	0.68	0.74
36	SonyAIBORobotSurface2	0.88	0.83	0.82	0.85	0.89
37	SwedishLeaf	0.90	0.87	0.86	0.89	0.90
38	SyntheticControl	1.00	0.99	0.98	0.99	0.99
39	ToeSegmentation1	0.79	0.79	0.83	0.82	0.81
40	ToeSegmentation2	0.84	0.89	0.92	0.78	0.75
41	Trace	0.99	1.00	0.95	0.99	1.00
42	TwoLeadECG	0.84	0.91	0.94	0.97	0.98
43	Wine	0.63	0.57	0.61	0.57	0.57

Table 4.1 Classification accuracies in the UCR/UEA Time Series datasets. Out distance, dLAGDISTR, achieves or ties for the best distance in 11 out of the 43 datasets and has the highest average accuracy.

generating processes are understood. We generate an additional scenario of simulated time series from the standard time series analysis literature. We use this scenario both for clustering and classification. This serves as a way to compare distances performance in a particular scenario and also to compare them under both supervised and unsupervised versions of the same underlying problem, showing how flexible the distances are when additional information in the form of the true class labels is available.

This simulation scenario features traditional linear and nonlinear autoregressive processes and 9 models taken from [60], originally used for testing linearity vs nonlinearity, to make a total of 22 different models.

10 series of length 250 are simulated from each model, and the ground truth labels for purposes of measuring classification accuracy and the external clustering criteria are the 22 models. The following models are included in the simulation:

- Stationary Linear
  - AR(1) $\varphi = 0.9$
  - AR(2) $\varphi_1 = 0.95, \varphi_2 = -0.1$
  - MA(1) $\theta = -0.9$
  - ARMA(1,1) $\varphi = 0.95, \theta = 0.1$
- Nonstationary Linear
  - ARIMA(0,1,0)
  - ARIMA(1,1,0) $\varphi = -0.1$
  - ARIMA(0,1,1) $\theta = 0.1$
  - ARIMA(1,1,1) $\varphi = -0.1, \theta = 0.1$
- Nonlinearity in mean
  - TAR $X_t = 0.5X_{t-1}\mathbf{1}(X_{t-1} < 0) - 2X_{t-1}\mathbf{1}(X_{t-1} > 0) + e_t$
  - EXPAR $X_t = [0.3 - 10exp(-X_{t-1}^2)]X_{t-1} + e_t$
  - NLMA $X_t = -0.5e_{t-1} + 0.8e_{t-1}^2 + e_t$
- Nonlinear in variance
  - GARCH(2,1) $\alpha_1 = 0.1, \alpha_2 = 0.01, \beta = 0.9$
  - GARCH(2,1) $\alpha_1 = 0.1, \alpha_2 = 0.1, \beta = 0.1$
- Nonlinear models (b) to (j) from [60]
  - $X_t = (-0.9 - 0.1e_{t-1})X_{t_1} + e_t + 2$
  - $X_t = (0.3 - 0.2e_{t-1})X_{t_1} + e_t + 1$
  - $X_t = (0.9exp(-X_{t-1}^2) - 0.6)X_{t_1} + e_t$

- $X_t = (0.9exp(-X_{t-1}^2) - 0.6)X_{t_1} + e_t + 0.3$
- $X_t = (0.9exp(-X_{t-1}^2) - 0.6)X_{t_1} + e_t + 1$
- $X_t = -0.3X_{t-1}\mathbf{1}(X_{t-1} \geq 0.2) - 0.6X_{t-1}\mathbf{1}(X_{t-1} < 0.2) + e_t$
- $X_t = -0.3X_{t-1}\mathbf{1}(X_{t-1} \geq 0.2) + 0.6X_{t-1}\mathbf{1}(X_{t-1} < 0.2) + e_t$
- $X_t = (-0.3X_{t-1} - 1)\mathbf{1}(X_{t-1} \geq 0.2) - (0.3X_{t-1} + 0.5)\mathbf{1}(X_{t-1} < 0.2) + e_t$
- $X_t = (-0.3X_{t-1} + 1)\mathbf{1}(X_{t-1} \geq 0.2) - (0.3X_{t-1} - 1)\mathbf{1}(X_{t-1} < 0.2) + e_t$

All models use e_t standard Gaussian innovations, **1** is the indicator function.

The distances involved in the simulation scenario are not the same as in the real data example. In this case, model-based distances from the time series literature are added, one based on the cepstral components [69], dAR.LPC.CEPS, and other based on the spectral density [37], dSPEC.LLR. The rest are chosen because they are commonly used in clustering and are already implemented in the TSclust and TSdist R packages [94, 96].

Classification results on the Simulation Scenario

We show in Table 4.2 the leave-one-out nearest neighbour crossvalidation values on each distance. dLAGDISTR achieves the best classification accuracy by a large margin.

DISTANCE	LOO CV
dLAGDISTR	0.74
dSPEC.LLR	0.55
dAR.LPC.CEPS	0.40
dEUCL	0.17
dDTWARP	0.44
dERP	0.41
dLCSS	0.02
dCIDDTW	0.62

Table 4.2 Classification accuracy of the distances in the simulation scenario

Clustering results on the Simulation Scenario

We distinguish two settings, when the true number of clusters is known and when it is unknown. The distance-based clustering algorithm is Partitioning Around Medoids (PAM). We use Average Silhouette Width for the internal clustering criterion (non ground truth based) and the Adjusted Rand Index and Gavrilov Index for (external clustering criterion).

The results when the number of clusters is known are shown in Table 4.3. The best performing method in this case is the distance based on the Spectral Density of the series, dSPEC.LLR. Our proposed approach is second in this scenario. When the number of clusters is unknown, and is selected by the Calinski-Harabasz index generalized for arbitrary dissimilarities (see [56]), the results are similar but our distance achieves better performance according to both external and internal criterions (see Table 4.4). The number of clusters found by the distance based on the Spectral Density is closer to the true solution (10 clusters found, 22 real), but it is arguably less interpretable than K=44, which is exactly double the amount of clusters, possibly it is why it gets better external criteria. Note that the difference between the top performing methods and the rest is very significant according to the adjusted rand index.

It is interesting to note that the results are better when the number cluster is unknown in most scenarios, attributed to the the fact that some clusters being difficult to separate. The results of classification and cluster are consistent among themselves, the best performing distances are the dLAGDISTR, dSPEC.LLR, and dCIDDTW.

DISTANCE	ADJ-RAND	GAV
dLAGDISTR	0.41	0.55
dSPEC.LLR	0.45	0.61
dAR.LPC.CEPS	0.01	0.12
dEUCL	0.05	0.24
dDTWARP	0.17	0.37
dERP	0.15	0.34
dLCSS	0.01	0.15
dCIDDTW	0.41	0.53

Table 4.3 Clustering results of the simulation scenario when the number of clusters is known.

4.5 Conclusions and future work

We have presented a distance for time series with applications to classification and clustering. The distance shows competitive results in both areas, while at the same time focuses on simplicity and sparseness, helping with interpretability.

DISTANCE	K	AVG. SIL	ADJ-RAND	GAV
dLAGDISTR	44	0.60	0.54	0.64
dSPEC.LLR	10	0.56	0.37	0.54
dAR.LPC.CEPS	43	0.38	0.31	0.47
dEUCL	3	0.74	0.01	0.16
dDTWARP	6	0.73	0.02	0.19
dERP	3	0.80	0.01	0.16
dLCSS	44	-0.80	0.02	0.17
dCIDDTW	9	0.47	0.14	0.34

Table 4.4 Clustering results when the number of clusters is not known. It is searched using the Calinski-Harabasz index.

This distance is based on capturing the autoregressive distribution of the series in a simple way, without assuming models, extracting features or applying additional simplifications to the processes. The autoregressive distribution is captured via lag or time delay embeddings, and then the lag embeddings are compared using a divergence between multivariate sets, originally a statistic for two sample testing.

The main improvements come from focusing on discrimination rather than prediction when modeling the autoregressive distributions, and by comparing the distributions in a parsimonious way thanks to incorporating this modern divergences between multivariate distributions.

The main drawback of the method is its computational complexity, now $O(n^2T^2)$ with n the sample size and T the length of the series. For these computational and also for empirical performance reasons, we would like to explore further the use of alternative divergences for comparing the lag embeddings.

Regarding the data-driven method for lag selection, in the context of supervised classification, we are preparing a global optimization method that considers a set of lags simultaneously, using multiple kernel learning techniques.

Chapter 5

FFORMA: Feature-based FORecasting Model Averaging

5.1 Introduction

Forecasting future values with high accuracy is indeed one of the main objectives of time series analysis. It is interesting in the fields of business management, healthcare or tourism, among many others. It is becoming increasingly common to have a large set of related time series, which we want to forecast some or all of them, examples of such scenarios are inventory control or stock exchange. Exploiting domain information can help overcome the limitation of individual forecasting methods and improve overall forecasting accuracy. In this chapter we propose a general framework to time series forecasting that exploits domain information. It works by the linear combination of individual forecasting methods, the weights are generated from a model that is trained on a dataset of time series, using features extracted from them as input.

There are essentially two general approaches for forecasting a time series: (i) generating forecasts from a single model; and (ii) combining forecasts from many models (forecast model averaging). There has been a vast literature on the latter motivated by the seminal work of Bates and Granger [11] and followed by a plethora of empirical applications showing that combination forecasts are often superior to their individual counterparts [see 33, 133, for example]. Combining forecasts using a weighted average is considered a successful way of hedging against the risk of selecting a misspecified model. A major challenge is in selecting an appropriate set of weights, and many attempts to do this have been worse than simply using equal weights — something that has become known as the "forecast combination puzzle" [see for example, 121].

We address the problem of selecting the weights by using a meta-learning algorithm based on time series features.

There have been several previous attempts to use time series features combined with meta-learning for forecasting [see for example 102, 76, 74, 70]. Recently, Talagala et al. [130] proposed the FFORMS (Feature-based FORecast Model Selection) framework that uses time series features combined with meta-learning for forecast-model selection. That is, features are used to select a single forecasting model. In this chapter, we build on this framework by using meta-learning to select the weights for a weighted forecast combination. All candidate forecasting methods are applied, and the weights to be used in combining them are chosen based on the features of each time series. We call this framework FFORMA (Feature-based FORecast Model Averaging). FFORMA resulted in the second most accurate point forecasts and prediction intervals amongst all competitors in the M4 competition [87]. The M4 Competition is the new iteration of the foremost competition in time series forecasting, the M Competitions, highly influential to the research in the field. It featured forecasting a dataset of 100000 time series in two main subcategories, point forecasting and prediction intervals, with awards given to the most accurate methods in each category according to specific measures of error. The time series come from macro and microenocomics, finance, tourism, etc. and have periods of hourly, daily, weekly, monthly,... Participants were required to forecast the next two period of each series. Special emphasis was given to reproducibility of results and to the clarification of the reasons behind the performance of the methods. 50 teams participated in the final deadline, with methods ranging from classical statistical time series analysis to deep learning.

The rest of the chapter is organized as follows. In section 5.2 we describe the FFORMA framework in a general sense. section 5.3 gives the details of our implementation of FFORMA in the M4 competition for generating both point and interval forecasts. This includes the required preprocessing steps, the set of features and forecast methods, as well as the specific implementation of the meta-learning model. We show empirical evidence on the performance of the approach in section 5.4 by quantifying the difference between our proposed learning model and a traditional classifier approach. section 5.4 also provides some final remarks and conclusions.

5.2 Methodology

5.2.1 Intuition and overview of FFORMA

The objective of our meta-learning approach is to derive a set of weights to combine forecasts generated from a *pool of methods* (e.g., naïve, exponential smoothing, ARIMA, etc.). The FFORMA framework requires a set of time series we refer to as the *reference set*. Each time series in the reference set is divided into a training period and a test period. From the training period a set of *time series features* are calculated (e.g., length of time series, strength of trend, autocorrelations, etc.). These form the inputs to the meta-learning model. Each method in the pool is fitted to the training period, forecasts are generated over the test period, and *forecast errors* (the difference between actual and forecast values) are computed. From these, a summary forecast loss measure from a weighted combination forecast can be computed for any given set of weights.

The meta-learning model learns to produce weights for all methods in the pool, as a function of the features of the series to be forecasted, by minimizing this summary forecast loss measure. Once the model is trained, weights can be produced for a new series for which forecasts are required. It is assumed that the new series comes from a *generating process* that is similar to some of those that form the reference set.

A common meta-learning approach is to select the best method in the pool of methods for each series; i.e., the method that produces the smallest forecast loss. This approach transforms the problem into a traditional classification problem by setting the individual forecasting methods as the classes and the best method as the target class for each time series. However, there may be other methods that produce similar forecast errors to the best method, so the specific class chosen is less important than the forecast error resulting from each method. Further, some time series are more difficult to forecast than others, and hence have more impact on the total forecast error. This information is lost if the problem is treated as classification.

Consequently, we do not train our meta-learning algorithm using a classification approach. Instead, we pose the problem as finding a function that assigns *weights* to each forecasting method, with the objective of minimizing the expected loss that would have been produced if the methods were picked at random using these weights as probabilities. These are the weights in our weighted forecast combination. This approach is more general than classification, and can be thought of as classification with *per class weights* (the forecast errors) that vary per instance, combined with *per instance weights* that assign more importance to some series.

5.2.2 Algorithmic description

The operation of the FFORMA framework comprises two phases: (1) the offline phase, in which we train a meta-learner; and (2) the online phase, in which we use the pre-trained meta-learner to identify forecast combination weights for a new series. Algorithm 2 presents the pseudo-code of the proposed framework.

Algorithm 2 The FFORMA framework: Forecast combination based on meta-learning

```
OFFLINE PHASE: TRAIN THE LEARNING MODEL
Inputs:
    $\{x_1, x_2, \dots, x_N\}$: $N$ observed time series forming the reference set.
    $F$: a set of functions for calculating time series features.
    $M$: a set of forecasting methods in the pool, e.g., naïve, ETS, ARIMA, etc.
Output:
    FFORMA meta-learner: A function from the extracted features to a set of $M$
    weights, one for each forecasting method.

Prepare the meta-data
for $n = 1$ to $N$: do
    Split $x_n$ into a training period and test period.
    Calculate the set of features $\boldsymbol{f}_n \in F$ over the training period.
    Fit each forecasting method $m \in M$ over the training period and generate
    forecasts over the test period.
    Calculate forecast losses $L_{nm}$ over the test period.
end for

Train the meta-learner, w
Train a learning model based on the meta-data and errors, by minimizing:

    $\underset{w}{\operatorname{argmin}} \sum_{n=1}^{N} \sum_{m=1}^{M} w(\boldsymbol{f}_n)_m L_{nm}.$

ONLINE PHASE: FORECAST A NEW TIME SERIES
Input:
    FFORMA meta-learner from offline phase.
Output:
    Forecast the new time series $x_{new}$.
for each $x_{new}$: do
    Calculate features $\boldsymbol{f}_{new}$ by applying $F$.
    Use the meta-learner to produce $\boldsymbol{w}(\boldsymbol{f}_{new})$ an $M$-vector of weights.
    Compute the individual forecasts of the $M$ forecasting methods in the pool.
    Combine individual forecasts using $\boldsymbol{w}$ to generate final forecasts.
end for
```

5.3 Implementation and application to the M4 competition

5.3.1 Reference set

Our meta-learning scheme requires a dataset of time series to train the model. This dataset should come from a similar process to the the series we want to forecast. In the case of the M4 competition, the amount of series we want to forecast is large enough to use it as a reference set, provided we apply temporal holdout to get training and test periods. We have experimented with adding additional data from similar domains to the reference set (such as from other forecasting competitions) but the differences were not substantial.

The M4 dataset includes 100,000 time series of yearly, quarterly, monthly, weekly, daily and hourly data. All 100,000 series form the reference set. Each series is split into a training period and a test period. The length of the test period for each time series was set to be equal to the forecast horizon set by the competition. Series with training periods comprising fewer than two observations, or series that were constant over the training period, were eliminated from the reference set. In total, 4 series were eliminated using this process.

5.3.2 Time series features

Table 5.1 provides a brief description of the 42 features used in this experiment, (F in Algorithm 2). The functions to calculate these features are implemented in the `tsfeatures` R package by Hyndman et al. [64]. Most of the features (or variations of these) have been previously used in a forecasting context by Hyndman et al. [65] and Talagala et al. [130], and are described in more detail there. The ARCH.LM statistic was calculated based on the Lagrange Multiplier test of Engle [41] for autoregressive conditional heteroscedasticity (ARCH). The heterogeneity features 39–42 are based on two computed time series: the original time series is pre-whitened using an AR model resulting in z; a GARCH(1,1) model is then fitted to z to obtain the residual series, r.

Features corresponding only to seasonal time series are set to zero for non-seasonal time series. For the sake of generality, we have not used any of the domain-specific features such as macro, micro, finance, etc., even though this information was available in the M4 data set.

Table 5.1 Features used in FFORMA framework.

Feature	Description	Non-seasonal	Seasonal
1 T	length of time series	✓	✓
2 trend	strength of trend	✓	✓
3 seasonality	strength of seasonality	-	✓
4 linearity	linearity	✓	✓
5 curvature	curvature	✓	✓
6 spikiness	spikiness	✓	✓
7 e_acf1	first ACF value of remainder series	✓	✓
8 e_acf10	sum of squares of first 10 ACF values of remainder series	✓	✓
9 stability	stability	✓	✓
10 lumpiness	lumpiness	✓	✓
11 entropy	spectral entropy	✓	✓
12 hurst	Hurst exponent	✓	✓
13 nonlinearity	nonlinearity	✓	✓
13 alpha	ETS(A,A,N) $\hat{\alpha}$	✓	✓
14 beta	ETS(A,A,N) $\hat{\beta}$	✓	✓
15 hwalpha	ETS(A,A,A) $\hat{\alpha}$	-	✓
16 hwbeta	ETS(A,A,A) $\hat{\beta}$	-	✓
17 hwgamma	ETS(A,A,A) $\hat{\gamma}$	-	✓
18 ur_pp	test statistic based on Phillips-Perron test	✓	✓
19 ur_kpss	test statistic based on KPSS test	✓	✓
20 y_acf1	first ACF value of the original series	✓	✓
21 diff1y_acf1	first ACF value of the differenced series	✓	✓
22 diff2y_acf1	first ACF value of the twice-differenced series	✓	✓
23 y_acf10	sum of squares of first 10 ACF values of original series	✓	✓
24 diff1y_acf10	sum of squares of first 10 ACF values of differenced series	✓	✓
25 diff2y_acf10	sum of squares of first 10 ACF values of twice-differenced series	✓	✓
26 seas_acf1	autocorrelation coefficient at first seasonal lag	-	✓
27 sediff_acf1	first ACF value of seasonally differenced series	-	✓
28 y_pacf5	sum of squares of first 5 PACF values of original series	✓	✓
29 diff1y_pacf5	sum of squares of first 5 PACF values of differenced series	✓	✓
30 diff2y_pacf5	sum of squares of first 5 PACF values of twice-differenced series	✓	✓
31 seas_pacf	partial autocorrelation coefficient at first seasonal lag	✓	✓
32 crossing_point	number of times the time series crosses the median	✓	✓
33 flat_spots	number of flat spots, calculated by discretizing the series into 10 equal sized intervals and counting the maximum run length within any single interval	✓	✓
34 nperiods	number of seasonal periods in the series	-	✓
35 seasonal_period	length of seasonal period	-	✓
36 peak	strength of peak	✓	✓
37 trough	strength of trough	✓	✓
38 ARCH.LM	ARCH LM statistic	✓	✓
39 arch_acf	sum of squares of the first 12 autocorrelations of z^2	✓	✓
40 garch_acf	sum of squares of the first 12 autocorrelations of r^2	✓	✓
41 arch_r2	R^2 value of an AR model applied to z^2	✓	✓
42 garch_r2	R^2 value of an AR model applied to r^2	✓	✓

5.3.3 Pool of forecasting methods

We considered nine methods implemented in the `forecast` package in R [63] for the pool of methods, P in Algorithm 2:

1. naïve (`naive`);
2. random walk with drift (`rwf` with drift=TRUE);
3. seasonal naïve (`snaive`).
4. theta method (`thetaf`);
5. automated ARIMA algorithm (`auto.arima`);
6. automated exponential smoothing algorithm (`ets`);
7. TBATS model (`tbats`);
8. STLM-AR Seasonal and Trend decomposition using Loess with AR modeling of the seasonally adjusted series (`stlm` with model function `ar`).
9. neural networks autoregressive model (`nnetar`)

The R functions are given in parentheses. In all cases, the default settings are used. If any function returned an error when fitting the series (e.g. a series is constant), the `snaive` forecast method was used instead.

5.3.4 Forecast loss measure

The forecasting loss (L in Algorithm 2) was adapted from the Overall Weighted Average (OWA) error described in the M4 competitor's guide [3], which adds together the Mean Absolute Scaled Error (MASE) and the symmetric Mean Absolute Percentage Error (sMAPE).

$$MASE = \frac{1}{h} \frac{\sum_{t=1}^{h} |Y_t - \hat{Y}_t|}{\frac{1}{n-m}\sum_{t=m+1}^{n} |Y_t - Y_{t-m}|}$$

$$sMAPE = \frac{1}{h}\sum_{t=1}^{h} \frac{2|Y_t - \hat{Y}_t|}{|Y_t| + |\hat{Y}_y|}$$

With Y_t the post sample value of the time series at point t, $\hat{Y}_t$ the estimated forecast, h the forecasting horizon and m the frequency of the time series (i.e., 12 for monthly series).

To compute the OWA of a method of a given series, its MASE and sMAPE are calculated and then are divided by the respective error of the Naïve 2 method *over all series in the dataset* (i.e., MASE by the average MASE of Naïve 2), and then added.

$$OWA = \frac{MASE}{\text{average Naïve 2 } MASE \text{ in the dataset}} + \frac{sMAPE}{\text{average Naïve 2 } sMAPE \text{ in the dataset}} \tag{5.1}$$

Naïve 2 is an extension of the Naïve method, adjusted for seasonality. To check whether a series should be adjusted for seasonality a 90% autocorrelation test is performed.

5.3.5 Meta-learning model implementation

We used the gradient decision tree boosting model of `xgboost` as the underlying implementation of the learning model [29]. This is a state-of-the-art model that is computationally efficient and has shown good performance in structure based problems. The great advantage of its application here is that we are able to customise the model with our specific objective function.

The basic `xgboost` algorithm produces numeric values from the features, one for each forecasting method in our pool. We applied the softmax transform to these values prior to computing the objective function. This was implemented as a *custom objective function* in the `xgboost` framework.

`xgboost` requires a gradient and hessian of the objective function to fit the model. The *correctly derived* hessian is prone to numerical problems that need to be addressed for the boosting to converge. This is a relatively common problem and one simple fix is to use an upper bound of the hessian by clamping its small values to a larger one. We computed a different upper bound of the hessian by removing some terms from the correct hessian. Although both alternatives converged, the latter required less boosting steps to converge. This not only increased the computational efficiency, it also generalized better due to a less complex set of trees produced in the final solution.

The general parameters of the meta-learning in Algorithm 2 were set as follows.

- $p(\boldsymbol{f}_n)_m$ is the output of the `xgboost` algorithm corresponding to forecasting method m, based on the features extracted from series x_n.
- $w(\boldsymbol{f}_n)_m = \frac{\exp(p(\boldsymbol{f}_n)_m)}{\sum_m \exp(p(\boldsymbol{f}_n)_m)}$ is the transformation to weights of the `xgboost` output by applying the softmax transform.
- L_{nm} is the contribution to the OWA error measure of method m for the series n.
- $\bar{L}_n = \sum_{m=1}^{M} w(\boldsymbol{f}_n)_m L_{nm}$ is the weighted average loss function.
- $G_{nm} = \frac{\partial \bar{L}_n}{\partial p(\boldsymbol{f}_n)_m} = w_{nm}(L_{nm} - \bar{L}_n)$ is the gradient of the loss function.

- The hessian H_{nm} was approximated by our upper bound $\hat{H}_n$:

$$H_n = \frac{\partial G_n}{\partial p(\boldsymbol{f}_n)_m} \approx \hat{H}_n = w_n(L_n(1-w_n) - G_n)$$

The functions G and $\hat{H}$ were passed to `xgboost` to minimize the objective function $\bar{L}$.

The results of `xgboost` are particularly dependent on its hyper-parameters such as learning rate, number of boosting steps, maximum complexity allowed for the trees or sub-sampling sizes. We limited the hyper-parameter search space based on some initial results and rules-of-thumb and explored it using Bayesian optimization implemented in the R package `rBayesianOptimization` [144] measuring performance on a 10% holdout version of the reference set. We picked the simplest hyper-parameter set from the top solutions of the exploration.

The full process is schematized in the flowchart of Figure 5.1.

5.3.6 Prediction intervals

For each new series we want to forecast, x_{new}, we used as the centre of the interval the point forecast produced by our meta-learner. Then the 95% bounds of the interval were generated by a linear combination of the bounds of three forecasting methods: naïve, theta and seasonal naïve. The coefficients for the linear combination were calculated in a data-driven way over the M4 database. The complete procedure was as follows:

1. We randomly divided the M4 dataset into two parts: A and B. We trained the FFORMA learner using the training periods of the series in part A and produced point forecasts over the test periods of the series of part B, and vice versa. Only one replication of this process was applied.

2. We computed the 95% *prediction radius* for the naïve, theta, and seasonal naïve methods. This is the difference between the 95% upper bound and the point forecast for each forecast horizon.

3. For each forecast horizon we found the coefficients that minimized the Mean Scaled Interval Score (MSIS) of the interval, defined in the M4 Competitor's guide [3] the following way:

$$MSIS = \frac{1}{h} \frac{\sum_{t=1}^{h} + \frac{2}{a}(U_t - L_t) + \frac{2}{a}(L_t - Y_t)\mathbf{1}\{Y_t < L_t\} + \frac{2}{a}(Y_t - U_t)\mathbf{1}\{Y_t > U_t\}}{\frac{1}{n-m}\sum_{t=m+1}^{n} |Y_t - Y_{t-m}|}$$

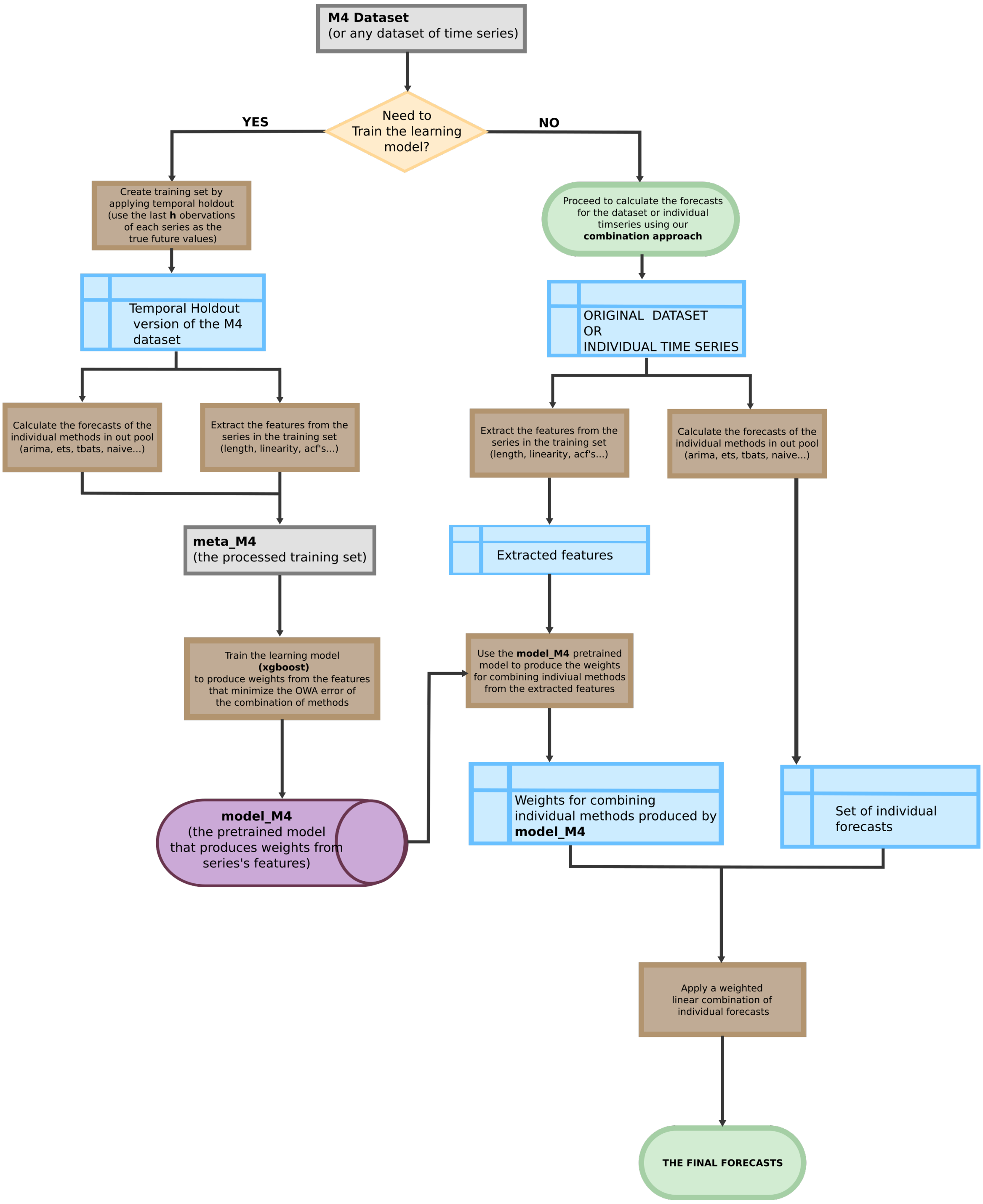

M4 Dataset
(or any dataset of time series)
Need to Train the learning model?
YES
NO
Create training set by applying temporal holdout (use the last h observations of each series as the true future values)
Temporal Holdout version of the M4 dataset
Calculate the forecasts of the individual methods in out pool (arima, ets, tbats, naive...)
Extract the features from the series in the training set (length, linearity, acf's...)
meta_M4
(the processed training set)
Train the learning model (xgboost) to produce weights from the features that minimize the OWA error of the combination of methods
model_M4
(the pretrained model that produces weights from series's features)
Proceed to calculate the forecasts for the dataset or individual timseries using our combination approach
ORIGINAL DATASET OR INDIVIDUAL TIME SERIES
Extract the features from the series in the training set (length, linearity, acf's...)
Calculate the forecasts of the individual methods in out pool (arima, ets, tbats, naive...)
Extracted features
Use the model_M4 pretrained model to produce the weights for combining indiviual methods from the extracted features
Weights for combining individual methods produced by model_M4
Set of individual forecasts
Apply a weighted linear combination of individual forecasts
THE FINAL FORECASTS

Fig. 5.1 Flowchart of the meta-learning process applied to the M4 dataset.

Where L and U are the lower and upper bounds of the prediction intervals, Y the future observations of the series, $a = 0.05$ and $\mathbf{1}$ the indicator function. We minimized the MSIS error produced by the linear combination of the radii of naïve, theta, seasonal naïve forecasts as the radius of the interval, using the point forecasts of FFORMA as the center. The minimization was done by gradient descent (Conjugate Gradient implemented in R `optim` function) over the test period of the series.

This method produced a set of three coefficients for each prediction horizon in the M4 dataset and these coefficients were the same independently of the series we want to forecast. Unlike the point forecasts, these coefficients were not restricted to be probabilities to produce averages.

5.3.7 Results in the M4 Competition

We show the results of the M4 Competition in terms of forecasting error and also present some analysis of the FFORMA model fitted to this dataset.

Table 5.2 shows the top three performing methods of the competition, according to the OWA error measure, and their respective sMAPE and MASE errors. The FFORMA Method achieved second position in average OWA error.

Method	OWA	sMAPE	MASE
Slawek Smyl	0.821	11.37	1.54
FFORMA	0.838	11.72	1.55
Pawlikoski et al.	0.841	11.84	1.55
Benchmark	0.898	12.55	1.56

Table 5.2 M4 Competition: Point forecast error results. Benchmark is a combination of methods of the exponential smoothing family, simple, Holts and damped.

The results of the prediction intervals part of the competition are shown in Table 5.3. FFORMA ranked second according to the MSIS measure used in the competition, and achieved a 95.86% coverage. This is a remarkable difference compared to the rest of the methods that participated in the competition. The third best method by MSIS, which achieved a coverage of 90.70%, is far from the target 95% set by the competition.

Method	MSIS	Coverage
Slawek Smyl	12.23	94.7%
FFORMA	14.33	95.96%
Doornik et al.	15.18	90.70%
ets (Benchmark)	15.68	91.27%

Table 5.3 M4 Competition: Prediction Interval Results.

When calculating rankings of the methods per series according to OWA, as opposed to the overall average OWA, FFORMA achieved the best average rank, followed by the method proposed by Fioruchi et al. (5th according to average OWA) and the method by Pawlikowski at al. reached third position, coinciding with its position according to average OWA.

An extended analysis of the competition results may be found in [87].

It is also interesting to analyze the xgboost model that we use as meta-learning model, fitted to the reference set of the M4 Competition. Figure 5.2 shows the relative importance of the ten most important features in the fitted model. This importance is measured by the *Gain* (see [2]) introduced by the feature in the model, considering all the trees. We see that many of the most important features come from the STL decomposition of the series, measuring the linearity and curvature of the trend, its strength relative the the remainder, etc. The series length and several kinds autocorrelations are also included. Note that given that xgboost uses bagging, results of importance vary may each time the model is fitted due to the randomness of the process.

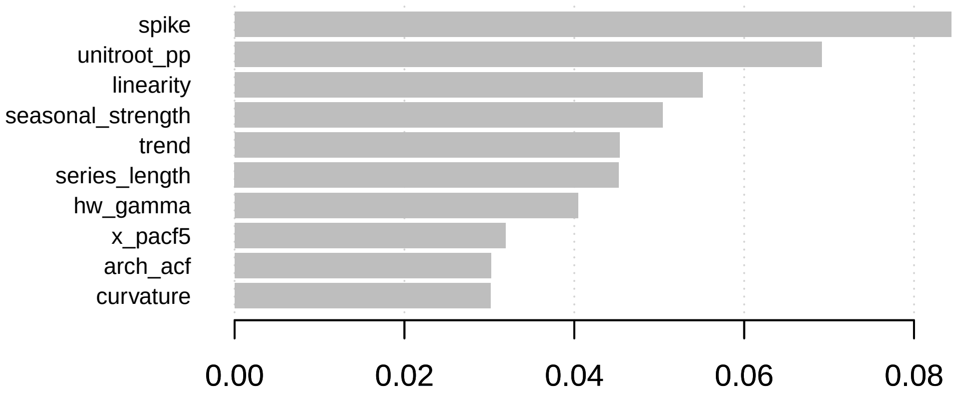


Fig. 5.2 Importance of the features of the xgboost model fitted to the M4 dataset.

The weights that the model assigns to each of the inidividual forecasting methods in the pool also gives a hint of the contribution of each methods to the overall performance. Table 5.4 shows the average weight that each of the methods receives. ARIMA is the method receiving the most weigth on average, and most of the weight is divided among the top four methods: ARIMA, exponential smoothing, TBATS and random walk. Both the autoregressive neural network and seasonal naive methods received very little weight.

arima	ets	nnetar	tbats	stlm	rwf	thetaf	naive	snaive
0.22	0.15	0.03	0.18	0.07	0.16	0.09	0.07	0.03

Table 5.4 Average weights received by each indidividual forecast method under FFORMA in the M4 Reference Dataset.

The whole training and forecasting process for FFORMA on the M4 dataset took approximately 96 hours in a 8-core desktop computer.

5.4 Discussion and conclusions

We have presented an algorithm for forecasting using weighted averaging of a set of models. The objective function of the learning model assigns weights to forecasting methods in order to minimize the forecasting error that would be produced if we picked the methods at random using these weights as probabilities. This contrasts with how the final forecasts are produced, which is a weighted average, not a selection.

These weights can however be used as part of a model selection algorithm, if one picks the method receiving the largest weight. This can be useful for interpretability or computational reasons, at the cost of forecasting performance.

In order to evaluate the impact of our contribution, we compared the average forecast error produced by FFORMA with a model selection approach. All implementation details were kept the same as in FFORMA; specifically we used the same set of features, the same pool of forecasting methods, the same underlying implementation (`xgboost`) but with a standard cross-entropy loss, and the same hyper-parameter search. This enabled us to measure the impact of the FFORMA loss function against a model selection approach, all other things being equal. We applied both FFORMA and the model selection approach to the M4 dataset and compared their overall point forecast errors. The average OWA error of the FFORMA approach was 10% smaller than that for model selection. This improvement is entirely due to the proposed model averaging rather than model selection.

One advantage of our approach is that its form is independent of the forecasting loss measure. Forecast errors enter the model as additional pre-calculated values. This allows FFORMA to adapt to arbitrary loss functions when models that directly minimize them would be restricted. For example, our approach can be applied to non-differentiable errors.

The source code for FFORMA is available at `github.com/robjhyndman/M4metalearning`.

Chapter 6

Distributed classification based on distances between probability distributions in feature space

6.1 Introduction

Data is growing at an unprecedented pace. With the variety, speed and volume of data flowing through networks and databases, it has become more and more difficult to find patterns that lead to meaningful conclusions. At the same time, organizations need to find ways to obtain some value from all of this data. Unlocking the most value from large, varied sets of information requires new approaches based on machine learning. However, traditional machine learning techniques and, more specifically, data mining algorithms, have been designed to run in a centralized computing environment, where all data could fit in a single machine. But nowadays, in the current scenario, where data size increases beyond capacity, these algorithms do not scale well —memory demands and impracticable runtimes—, damaging performance and efficiency. Thus, distributed learning has become essential.

The motivation for distributed learning is at least twofold. The most obvious reason is the volume of data available nowadays. Data generation, which has been estimated at 2.5 exabytes of data per day [14], comes from everywhere: genomics, astronomy, CERN experiments, transaction records, posts to social media sites (Twitter generates 500 million tweets/day, each about 3 kilobytes including metadata) or digital pictures and videos (YouTube currently has 300 hours of video being uploaded every minute). Second, data is often shared across geographical and organizational boundaries, and it is not economic or legal to gather it in a single location. For example, several datasets concerning business information might be owned by separate organizations that have competitive reasons for keeping the data private. In addition, this data may be physically dispersed over many different geographic locations. However, business organizations may be interested in enhancing their own models by exchanging useful information about the data.

The machine learning community has been essentially focused on the design of distributed or parallel algorithms to deal with massive datasets [99]. Different from the traditional centralized algorithms where a single learner has access to the full dataset, distributed learning algorithms have their foundations in ensemble learning [39]. Ensemble learning consists of a hierarchy of multiple local learners operating on subsets of the full dataset, and one or more ensemble learners combining the outputs of all the local learners. Thus, the ensemble approach is almost directly applicable to a distributed scenario since a classifier can be trained at each site, using the subset of data stored in it, and then the classifiers can be eventually combined using ensemble strategies. To combine the predictions of a set of classifiers, one of the simplest ways consists of using decision rules [72]. These decision/fixed rules are defined as functions that receive as inputs the outputs of the set of learned classifiers and combine them to produce a unique output. Chan and Stolfo [24] proposed several meta-learning strategies for integrating independently learned classifiers by the same learner in a parallel and distributed computing environment. Breiman [21] presented a procedure to build ensembles of classifiers from small subsets of data, growing a predictor on each subset and then pasting these predictors together. Lazarevic *e*t al. [75], using the Mahalanobis distance, developed a general framework for distributed boosting to integrate efficiently specialized classifiers learned over very large and distributed homogeneous datasets that cannot be merged at a single location. Tsoumakas *e*t al. [134] presented a framework for distributed stacking of multiple classifiers. Their method was based on local learning and model stacking using the average probability distribution of the local classifiers' output according to the class as input to the second level classifier. Similar to distributed learning, another common approach for scaling up learning algorithms is parallel machine learning [136], which includes GPU architecture and map reduce techniques.

Data can be distributed either horizontally or vertically. In horizontal partitioning, the dataset is divided into several nodes that have the same features as the original dataset, each containing a subset of the original samples. In vertical partitioning, the original dataset is divided into several nodes that have the same number of instances as the original dataset, each containing a subset of the original set of features. This chapter is focused on horizontal partitioning, since it constitutes the most suitable and natural approach for most applications. In addition to the common learning scenario assumptions, we assume the availability of a test set large enough to obtain distributional information. In this distributed scenario, class probabilities can be shown to be a weighted average of the individual class probabilities within each node. These weights depend on the marginal probabilities of the instance over each node and over the entire data set. This result motivates the study of the use of distances between distribution functions for improving classification performance. In this chapter, two different approaches to approximate these weights are proposed. The first one is based on estimating the distance between feature distributions between each node and the test set, while the second one controls the contribution of each instance of the test set in order to minimize these distributional distances. The resulting learning models exhibit interesting properties including that they work with any local classifier and do not require retraining the classifiers or sharing information between individual nodes. The experimental results on several real and synthetic data sets report benefits in terms of classification accuracy, particularly when the second approach is considered. Besides, we assess a common problem in many real world problems, the "class imbalance change" [40] in classification. In a non-distributed framework, this problem appears when the class balance changes between training and test data sets, due to sample selection bias or non-stationarity of the environment [104]. In our case, unbalancedness happens when the feature distributions differ between nodes. Distributed real-world data sets are usually not symmetric, i.e. the distributions of data for different locations may not be the same. Imagine a group of epidemiologists studying the spread of hepatitis-C in Europe. They are interested in detecting any underlying relation of the emergence of hepatitis-C in Europe with the weather and social patterns. They have access to some large hepatitis-C country-specific databases and an environmental database at EEA (European Environment Agency). These patterns could change considerably from one country to another (e.g. from Denmark to Italy). Besides, analyzing the data from these distributed datasets using a traditional data mining algorithm will require combining the databases at a single location, which is quite impractical, or perhaps not possible due to memory reasons or to privacy issues. It will be shown that our model works particularly well in these scenarios.

The remainder of this chapter is organized as follows. Our distributed learning model is introduced and analyzed in detail in Section 6.2. Specifically, the considered framework and the rationale of the proposal are discussed in Sections 6.2.1 and 6.2.2, respectively. A scheme of the methodology is outlined in Section 6.2.3, and the key issues of the procedure involving the choice of the distributional distance, the weighting criteria and the way of combining the single classifier outputs are discussed in Sections 6.2.4, 6.2.5 and 6.2.6, respectively. The results from an experimental study involving simulated and real data sets are presented in Section 6.3, which is structured as follows. The experimental design is described in Section 6.3.1. Different classification algorithms, combination strategies, and partition sizes have been considered in both balanced and unbalanced scenarios. Some results from a pair of specific experiments conducted to motivate our approach are shown in Section 6.3.2, and the bulk of experimental results are analyzed in Section 6.3.3. Additional results measuring Precision and Recall from our experiments are also provided in Appendix B. Lastly, computational complexity is addressed in Section 6.4 and some concluding remarks and proposals for future research are given in Section 6.5.

6.2 A novel distributed learning model

This section is devoted to formally establish the distributed classification problem and describes in detail the learning model we propose.

6.2.1 Background

Consider a population Ξ characterized by pairs of measurements $(x,C) \in \mathcal{X} \times \mathcal{C}$, where $\mathcal{X}$ denotes a domain of d-dimensional feature vectors and $\mathcal{C} = \{C_1,\dots,C_m\}$ is a set of m class labels. Denote by $P(x,C)$ the joint probability function over $\mathcal{X} \times \mathcal{C}$. In a standard classification context, a classifier based on a training sample of n labeled objects $\mathcal{Z} = \{(x_1,C_1),\dots,(x_n,C_n)\}$ is used to predict the class of unlabeled features. In a horizontally distributed framework, the population Ξ spreads across p disjoint nodes, let us say $\mathcal{P} \equiv \{\mathcal{N}_1,\dots,\mathcal{N}_P\}$, and the training data set brings together instances from the different nodes, i.e. $\mathcal{Z} = \cup_{i=1}^{p} \mathcal{Z}_i$, with $\mathcal{Z}_i$ formed by n_i instances belonging to $\mathcal{N}_i$, with $\sum_{i=1}^{p} n_i = n$.

In this chapter we focus on a distributed environment where a set $\mathcal{T}$ of t unlabeled instances is available and our target is to estimate their labels. The availability of a whole set of unlabeled instances $\mathcal{T}$ enables us to gain knowledge about the underlying distribution of X and to assess how well this distribution is represented at each node. As we will see later, our learning model relies on these distributional distances so that the availability of $\mathcal{T}$ is a basic requirement.

In addition, our learning model is constructed under the standard assumption that the training set $\mathcal{Z}$ and the test set $\mathcal{T}$ are independent and identically distributed samples drawn from the population in study, and therefore following the probability model given by $P(x,C)$. Note that this stationarity assumption is the default assumption in many learning scenarios. No distributional or other assumptions are required on the data or how they are distributed across nodes. In fact, data within each node could follow different distributions since no constraints on the fragmentation scheme are imposed. In particular, our framework encompasses scenarios with unbalanced nodes [109] or with data-driven partitions on the basis of heuristic rules stated to obtain better classification rates [38].

We also impose the restriction that no communication between nodes is required. Intelligent interaction between nodes, e.g. taking advantage of the most informative data at each node, can improve the classification accuracy [35]. Nevertheless, exchanging information between nodes is frequently unfeasible in real problems dealing with distributed data for different reasons such as storage cost, communication cost or private and sensitive data, among others [135].

6.2.2 An overview of our approach

A common approach in distributed learning [13] consists of building classifiers trained at each node $\mathcal{N}_i$ using $\mathcal{Z}_i$, $i = 1,\ldots,p$, and then combining the classifier outputs by means of a proper ensemble learning strategy [39]. In our approach, we intend to take advantage of the availability of $\mathcal{T}$ to gain insight into the marginal probability distribution of the feature vectors, and using this knowledge to modulate the importance of each individual classifier in the combination rule. Specifically, we wish to estimate the posterior probability that the j-th instance in $\mathcal{T}$, with observed feature vector x_j, belongs to the class C_k, for $k = 1,\ldots,m$ and $j = 1,\ldots,t$. Under the stationarity assumption and given that $\mathcal{P}$ is a partition of Ξ, we have

$$P(C_k|x_j)P(x_j) = \sum_{i=1}^{p} P(C_k|x_j,\mathcal{N}_i)P(x_j|\mathcal{N}_i)P(\mathcal{N}_i), \tag{6.1}$$

where $P(x_j)$ denotes the marginal density of x_j, $P(x_j|\mathcal{N}_i)$ the density of x_j conditional on the i-th node, and $P(\mathcal{N}_i)$ the prior probability of an instance is allocated to $\mathcal{N}_i$.

Let ω_{ji} be the ratio defined by $\omega_{ji} = \dfrac{P(x_j|\mathcal{N}_i)}{P(x_j)}$, for $i = 1,\ldots,p$. Then, from (6.1) follows that

$$P(C_k|x_j) = \sum_{i=1}^{p} P(C_k|x_j,\mathcal{N}_i)\,\omega_{ji} P(\mathcal{N}_i). \tag{6.2}$$

Equation (6.2) establishes that the posteriori probability of the class C_k given an observed feature vector x_j is a weighted average of the posteriori probabilities within each node, with weights depending on the node size and the ratios ω_{ji}. By definition, ω_{ji} measures how well represented is the observed feature vector x_j in the i-th node. Whether the partition $\mathcal{P}$ has been set evenly and uniformly at random, the feature vectors within each node follow similar distributions and ω_{ji} will take values close to one for all j and i. Otherwise, markedly unbalanced nodes will produce very different ω_{ji}.

The value of $P(C_k|x_j)$ can be directly estimated from (6.2) as long as the remaining involved probabilities are previously approximated. The posteriori probability within each node, $P(C_k|x_j,\mathcal{N}_i)$, is estimated using the classifier trained at $\mathcal{N}_i$ and whose output consists of a vector of m belief values. The proportion of training data belonging to the i-th node can be taken as an estimate of $P(\mathcal{N}_i)$. For the sake of simplicity and computational efficiency, we will assume nodes of equal size so that the weight of a single prediction is not affected by the nodes' sizes. Lastly, the behavior in probability of the feature vectors over Ξ and over each node $\mathcal{N}_i$ can be modeled with nonparametric kernel densities based on the features forming $\mathcal{T}$ and $\mathcal{Z}_i$, respectively. Nevertheless, this involves several difficulties. First, we could face the "curse of dimensionality" problem since the dimension of the feature space may be arbitrarily large. Moreover, we look for a learning model able to manage different types of features, including mixtures of discrete and continuous variables. But even assuming an affordable dimension and continuous features, $(p+1)$ kernel densities should be obtained, which substantially increases the likelihood of estimation errors. In particular, small errors estimating $P(x_j|\mathcal{N}_i)$ or $P(x_j)$ might produce arbitrarily large or small coefficients ω_{ji}, thus leading to overweight or underweight the predictions in specific nodes.

To overcome these drawbacks, the computation of the $(p+1)$ kernel densities is circumvented by directly estimating the coefficients ω_{ij}. Two different approaches are proposed. In both cases, the aim is to measure the dissimilarity d_i between the feature distributions on the i-th node and the global population, hereafter denoted by $F_{\mathcal{N}_i}$ and $F_{\mathcal{X}}$, respectively. A suitable distance between high-dimensional distributions is considered, and then specific values for d_i, $i = 1,\ldots,p$, are obtained using the empirical distributions based on $\mathcal{Z}_i$ and $\mathcal{T}$. The first proposal consists in taking $\omega_{ji} = K \cdot d_i^{-1}$, for all j and i, and K being a constant. This way, all the instances in $\mathcal{T}$ receive the same weight at each node, which decreases with the distance between $F_{\mathcal{N}_i}$ and $F_{\mathcal{X}}$. The second proposed approach is not simply based on the global distance between empirical distributions. The weights ω_{ji} are determined in order to maximize the matching between $F_{\mathcal{N}_i}(x_j)$ and $F_{\mathcal{X}}(x_j)$, for all $x_j \in \mathcal{T}$. Unlike the prior approach, the test instances receive different weights ω_{ji} at the same node. The effective computation of the weights is formalized throughout an optimization problem. A detailed description of the two proposed weighting criteria is provided in Section 6.2.5.

6.2.3 Outline of the methodology

We propose a distributed learning methodology consisting of the following four stages.

Step 1 Assess the distance between the probability distributions of X on the i-th training node $\mathcal{N}_i$ and the global population Ξ using a suitable statistic to measure dissimilarity between high-dimensional distributions. Denote by d_i the normalized distance obtained for the i-th training node, $i = 1,\ldots,p$.

Step 2 Based on a pre-selected classifier, obtain for each feature vector x_j of the test sample the classifier outputs $\mathbf{Y}_j = \{\mathbf{y}_{j1},\ldots,\mathbf{y}_{jp}\}$, where $\mathbf{y}_{ji}$ denotes the response generated by the classifier trained at the node $\mathcal{N}_i$, for $i = 1,\ldots,p$ and $j = 1,\ldots,t$.

It is assumed that each classifier output consists of a vector of m membership or belief values, i.e. $\mathbf{y}_{ji} = (y_{ji1},\ldots,y_{jim})$, where y_{jik} can be interpreted as the amount of confidence or evidence in the assignment of the feature x_j to the k-th class, C_k, for $k = 1,\ldots,m$.

Step 3 Obtain weighted versions of the belief values $\mathbf{Y}_j^{\omega} = \left\{\mathbf{y}_{j1}^{\omega},\ldots,\mathbf{y}_{jp}^{\omega}\right\}$, with $\mathbf{y}_{ji}^{\omega} = \omega_{ji}\mathbf{y}_{ji}$, where the weights ω_{ji} take into consideration the distributional distances d_i. Two different criteria are proposed to determine how the weights are constructed.

Step 4 Generate a unique decision for classifying the j-th instance in $\mathcal{T}$, with observed feature x_j, by combining the corresponding weighted belief sets $\{\omega_{j1}\mathbf{y}_{j1},\ldots,\omega_{jp}\mathbf{y}_{jp})$. Following Kittler *et al.* [73], several fixed rules (*decision rules*) involving functions of the elements of $\mathbf{Y}_j^{\omega}$ are considered to produce the required unique output.

The key points of the proposed methodology involve: (i) the choice of the distributional distance d_i, (ii) the weighting criteria on the belief values regarding the distances d_i, and (iii) the selection of a decision rule. Each of these issues is properly discussed below.

6.2.4 Measuring dissimilarity between high-dimensional distributions

To assess the distance between the probability distributions of X over an arbitrary node $\mathcal{N}_i$ and the population Ξ, we propose to use the so-called *energy statistic* [127, 128]. Consider two independent samples $\mathcal{X}$ and $\mathcal{X}'$ generated from multivariate distributions $F_{\mathcal{X}}$ and $F_{\mathcal{X}'}$, respectively. The energy distance between $\mathcal{X}$ and $\mathcal{X}'$ is defined by

$$E\left(\mathcal{X},\mathcal{X}'\right) = 2d_{\mathcal{X},\mathcal{X}'} - d_{\mathcal{X},\mathcal{X}} - d_{\mathcal{X}',\mathcal{X}'}, \tag{6.3}$$

with

$$d_{\mathcal{A},\mathcal{B}} = \frac{1}{rs}\sum_{u=1}^{r}\sum_{v=1}^{s}\|a_u - b_v\|,$$

where $\|\cdot\|$ denotes the Euclidean norm and $\mathcal{A} \equiv (a_1,\ldots,a_r)$ and $\mathcal{B} \equiv (b_1,\ldots,b_s)$ denote arbitrary data sets. Note that this is in essence the same energy statistic mentioned in Equations 2.6 and 4.1.

Under mild regularity conditions on the generating patterns, Székely and Rizzo [127] established the consistency of the statistic (6.3) to check the equality of the generating distributions $F_{\mathcal{X}}$ and $F_{\mathcal{X}'}$. Hence $E(\mathcal{X},\mathcal{X}')$ can be seen as a measure of the distance between $F_{\mathcal{X}}$ and $F_{\mathcal{X}'}$ in such a way that the larger value of the statistic, the more distant are the distributions. By construction, $E(\mathcal{X},\mathcal{X}')$ is based on comparing averages of interpoint distances evaluated within and between samples, which means to move the multidimensional problem to dimension one. Thus, the energy distance is particularly attractive to be applied in arbitrarily high dimension. It is also worth remarking that different types of interpoint distances could be used to construct $E(\cdot,\cdot)$, thus providing versatility to deal with features taking nominal, categorical, continuous and also mixed values. Also, the good analytical properties of the energy distance will allow us to formalize in the next section a suitable optimization problem designed to provide useful weights for the belief values. Supported by these nice properties, we decided to evaluate the distributional distance between each $\mathcal{N}_i$ and Ξ by means of the energy distance between the i-th training sample and the test sample, i.e. by $d_i = E(\mathcal{Z}_i,\mathcal{T})$, for $i = 1,\ldots,p$.

6.2.5 Weighting the belief values generated by the single classifiers

From Step 2 of the proposed methodology, the outputs of the single classifiers $\mathbf{Y}_j = \{\mathbf{y}_{j1}, \ldots, \mathbf{y}_{jp}\}$ are available for each feature vector x_j of the test sample. As mentioned, we assume that $\mathbf{y}_{ji}$ is a vector of levels of belief in the assignment of x_j to each of the classes. Working with belief levels enables us to analyze the performance of a range of efficient classifier combination rules [73] in Step 4 of the proposed methodology. Step 3 consists in correcting these belief values by introducing the distributional distances d_i. Two different criteria are proposed.

One approach consists in assigning weights in inverse proportion to the energy distance for the corresponding node, i.e. $\omega_{ji} = K \cdot d_i^{-1}$ for all j, where $K = \left(\sum_{i=1}^{p} d_i^{-1}\right)^{-1}$, is a normalizing constant used to make the sum of weights equal to one. This way, the belief values generated from each local classifier receive a common weight for all instances in the test set, resulting $\mathbf{y}_j^{\omega} = K\, d_i^{-1} \mathbf{y}_j$, for all $j = 1, \ldots, t$. Hereafter this weighting approach will be referred to **per-Node Weighting** and denoted by **pNW**.

In order to provide a finer grain approach where the belief degrees per instance in the test set receive different weights, an alternative weighting approach is proposed. The aim is to assign weights in order to minimize the energy distance between each training sample and the test set. The procedure can be understood as if, for each node $\mathcal{N}_i$, a weighted resampling scheme of the test set is carried out to overweight belief values associated to instances better represented at the node than in the test sample. Features x_j allocated in low probability zones in the test set but belonging to high probability zones in a specific node will receive high weights, and conversely instances with low probability in the test set but high probability in the node will be downweighted (see Figure 6.1). By assigning high weights to instances with low representation in the test set but well-represented at the node, we ensure an efficient use of the training samples. Notice that equation (6.2) in Section 6.2.2 leads to theoretical weights $\omega_{ji} = P(x_j | \mathcal{N}_i) / P(x_j)$, thus accounting for the rationale of this approach. Unlike the per-node belief approach, under this new weighting criterion each node produces a weight for each instance in the test set. For this reason, this weighting approach will be referred as **per-Instance Weighting** and denoted by **pIW**.

According to the definition of the energy distance in (6.3), the per-instance weights for the set of test instances at the i-th node, $\omega_i = (\omega_{1i}, \ldots, \omega_{ti})$, are obtained by minimizing the objective function $E(\omega_i)$ given by

$$E(\omega_i) = 2D_{\mathcal{Z}_i,\mathcal{T}}\omega_i^T - D_{\mathcal{Z}_i,\mathcal{Z}_i} - \omega_i D_{\mathcal{T},\mathcal{T}}\omega_i^T, \tag{6.4}$$

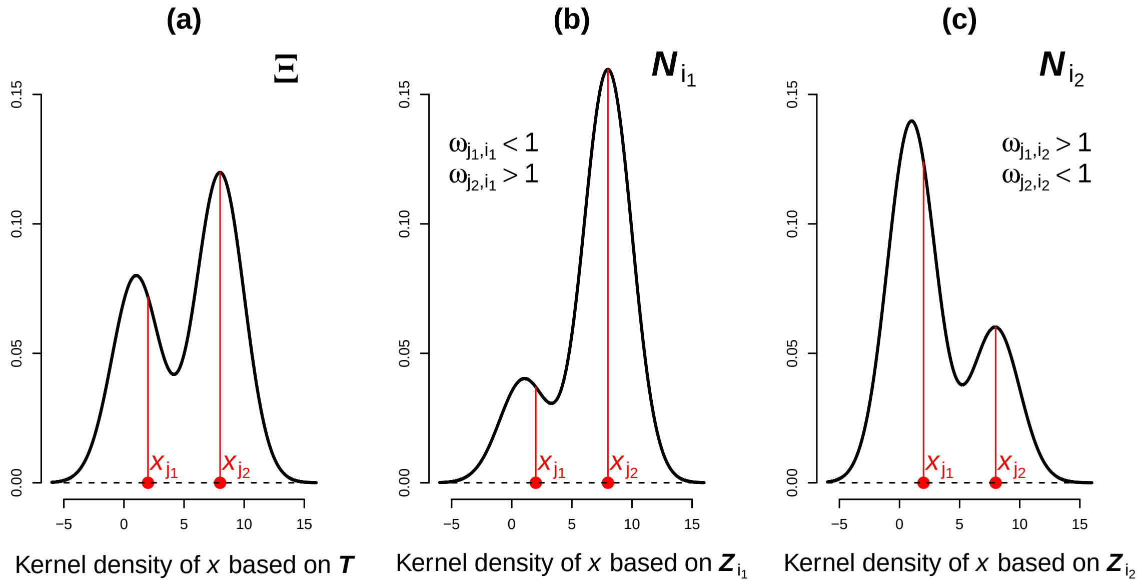


Fig. 6.1 Graphical illustration of the per-Instance Weighting (pIW) criterion.

where $D_{\mathcal{A},\mathcal{B}}$ is the matrix whose (u,v)-element is $D_{\mathcal{A},\mathcal{B}}(u,v) = \|a_u - b_v\|$, for arbitrary data sets $\mathcal{A} \equiv (a_1,\ldots,a_r)$ and $\mathcal{B} \equiv (b_1,\ldots,b_s)$.

In practice, the minimization of $E(\omega_i)$ is posed by means of the optimization problem, solved by the interior-point method:

$$\begin{array}{ll} \underset{\omega_i}{\text{minimize}} & \frac{1}{t} D_{\mathcal{N}_i,\mathcal{T}} \omega_i^T - \omega_i D_{\mathcal{T},\mathcal{T}} \omega_i^T \\ \text{subject to} & \sum_{i=1}^{p} \omega_i = 1, \omega_i \succeq 0. \end{array}$$

6.2.6 Combining the belief values generated by the single classifiers

Last step in the proposed methodology consists in combining the weighted outputs of the single classifiers trained at each of the nodes, namely the vectors of belief degrees $\mathbf{y}_{ji}^{\omega} = \omega_{ji}\mathbf{y}_{ji} = (\omega_{ji}y_{ji1},\ldots,\omega_{ji}y_{jim})$, whose k-th element $y_{jik}^{\omega} = \omega_{ji}y_{jik}$ provides an estimate of the posteriori probability $P(C_k|x_j,\mathcal{N}_i)$, for $k = 1,\ldots,m$. Having available continuous outputs in form of belief values allows us to consider different functions of these values, so-called *decision rules* [99], to get a unique output. Kittler *et* al. [73] argue that the decision rules provide a useful approach to circumvent the complex problem of inferring the posteriori probability function

$$P(x_j \text{ is assigned to the class } C_k \,|\mathbf{y}_{j1},\ldots,\mathbf{y}_{jp}),$$

which would allow us to determine the most likely class using the Bayesian theory. Some alternative classifier combination approaches include techniques such as Stacked Generalization, Meta-Learning, Knowledge Probing and Effective Stacking [99]. Nevertheless, these methods work training a new classifier based on single outputs produced by each node, which requires access to a common training set or sharing of private training information among nodes, thus limiting their applicability and violating the condition of no communication between nodes stated in Section 6.2.1. Supported by these arguments, we propose to use some of the most popular decision rules to generate the final assignment.

Following Kittler *e*t al. [73], where a common theoretical framework for different decision rules is provided, we have considered in our experiments the set of rules presented below. In all cases, we assume that the belief values have been normalized so that $P(C_k|x_j,\mathcal{N}_i) = y^{\omega}_{jik}/\sum_{l=1}^{m} y^{\omega}_{jil}$, for all j and i.

- *Product rule.* The instance with observed feature vector x_j is assigned to the class C_k if
$$\prod_{i=1}^{p} y^{\omega}_{jik} = \max_{1\leq l\leq m} \prod_{i=1}^{p} y^{\omega}_{jil}.$$
Note that, under this rule, a class with a zero or very small belief value from only one node will receive a zero or very small combined belief degree, even if the rest of nodes provide high belief degrees to the mentioned class. Hence, this rule will exhibit a bad performance if for example a class is not represented in a particular node.

- *Sum rule.* The instance with observed feature x_j is assigned to the class C_k if
$$\sum_{i=1}^{p} y^{\omega}_{jik} = \max_{1\leq l\leq m} \sum_{i=1}^{p} y^{\omega}_{jil}.$$
The theoretical support for the sum rule lies on assuming that the posteriori probabilities do not deviate greatly from the prior probabilities [73], that is $P(C_k|x_j,\mathcal{N}_i) = P(C_k) + \varepsilon_{jk}$, with ε_{jk} taking very small values for all k and j. Kittler *e*t al. [73] have shown that the sum rule is less sensitive to the estimate errors than the product rule.

- *Max rule.* The instance with observed feature x_j is assigned to the class C_k if
$$\max_{1\leq i\leq p} y^{\omega}_{jik} = \max_{1\leq l\leq m}\max_{1\leq i\leq p} y^{\omega}_{jil}.$$
The class obtaining the highest belief degree over all the nodes is selected as combined output. It can be shown that this rule approximates the sum rule under the assumption of equal prior probabilities for the classes.

- *Min rule.* The instance with observed feature x_j is assigned to the class C_k if
$$\min_{1\leq i\leq p} y^{\omega}_{jik} = \max_{1\leq l\leq m}\min_{i=1}^{p} y^{\omega}_{jil}.$$
Assuming as before that classes are a priori equiprobable, the min rule approximates the product rule.

- *Majority vote rule.* The instance with observed feature x_j is assigned to the class C_k if
$$\sum_{i=1}^{p}\Delta_{jik} = \max_{1\leq l\leq m}\sum_{i=1}^{p}\Delta_{jil},$$
where $\Delta_{jil} = 1$ if $y^{\omega}_{jil} = \max_{1\leq u\leq m} y^{\omega}_{jiu}$ and $\Delta_{jil} = 0$ otherwise. Therefore, the combined output consists in selecting the class receiving the largest number of votes from the single classifiers. Under the equiprobability assumption for the prior probabilities, this rule matches the sum rule when the belief values are discretized by using the Δ_{jil} values.

6.2.7 Some remarks

Some remarks concerning the proposed methodology are highlighted below.

Remark 1. The estimated distributional distance d_i between a particular node $\mathcal{N}_i$ and Ξ could be small (large) even though a few test instances are bad (well) represented at $\mathcal{N}_i$. In any case, all the classifier outputs obtained at $\mathcal{N}_i$ will receive the same weight when the pNW criterion is used. This is an unsuitable consequence of taking weights based on the global distance d_i such as pNW does. On the contrary, the pIW criterion checks the point-to-point distribution matching, thus being sensitive to local deviations. Note that if a feature vector x_j is badly represented at a specific node, then it must be well represented at another node because $\mathcal{P}$ is a partition of the feature domain. In sum, the pIW criterion is expected to outperform the pNW one, and the improvement would be more substantial with unbalanced nodes. In our experimental evaluation in

Section 6.3.3, both weighting criteria are examined and compared with a standard approach without weighting the single belief values (an unweighted approach denoted by **UW**). A scheme of the three distributed approaches is shown in Figure 6.2.

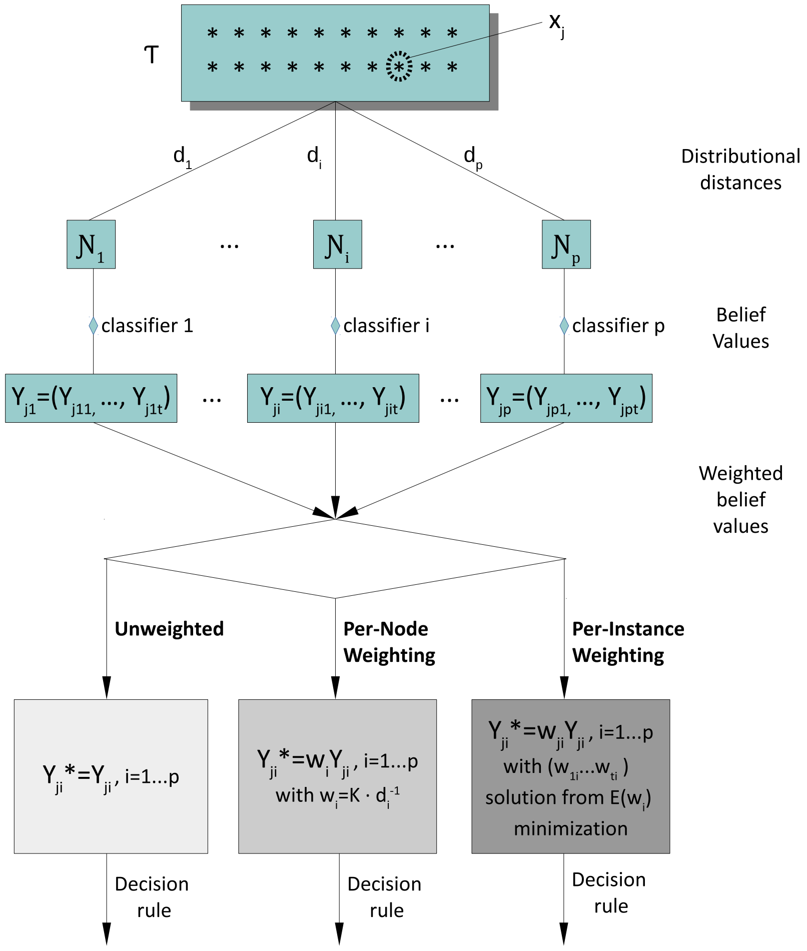

T
x_j
d_1
d_i
d_p
Distributional distances
N_1
...
N_i
...
N_p
classifier 1
classifier i
classifier p
Belief Values
$Y_{j1}=(Y_{j11,} ..., Y_{j1t})$
...
$Y_{ji}=(Y_{ji1,} ..., Y_{jit})$
...
$Y_{jp}=(Y_{jp1,} ..., Y_{jpt})$
Weighted belief values
Unweighted
Per-Node Weighting
Per-Instance Weighting
$Y_{ji}*=Y_{ji}$, i=1...p
$Y_{ji}*=w_iY_{ji}$, i=1...p with $w_i=K \cdot d_i^{-1}$
$Y_{ji}*=w_{ji}Y_{ji}$, i=1...p with $(w_{1i}...w_{ti})$ solution from $E(w_i)$ minimization
Decision rule
Decision rule
Decision rule

Fig. 6.2 Distributed approaches schemes.

Remark 2. In a non-distributed classification context with different distributions for the training and test sets (*sample selection bias* problem), Huang *et* al. [61] proposed to use the unlabeled data to reweight the training data in such a way that the means of the training and test features in a reproducing kernel Hilbert space are close. Although in a different context, this is a similar idea to the pIW approach and it is worth emphasizing the main differences. In our work, the reweighting process is applied to the test data because the nodes cannot be retrained in our distributed scenario. On the other hand, a key assumption in [61] is that the conditional probability of $C|x$ is the same for the training and test populations so that the bias is only exhibited by the feature distributions. In our framework, stationarity is assumed and therefore the bias can only be present between nodes. Nevertheless, it is not necessary to require that the conditional probabilities of $C|x$ remain unchanged across the nodes, which would be a very restrictive constraint.

Remark 3. As already mentioned, Kittler *e*t al.[73] pointed out some nice properties of the sum rule to combine the single classifier outputs. Beyond these properties, equation (6.2) provides theoretical support to use this criterion since the posteriori probabilities are expressed as a weighted sum of the single classifier outputs within each node.

Remark 4. The proposed learning model is not restricted to the use of a particular classification model at each node. The unique requirement is that the classifier outcome consists of a vector of belief values or posteriori probabilities of the classes for a given feature vector. Thus, artificial neural network, logistic regression, support vector machines, Bayesian classifiers, and Random Forest could be used among others.

6.3 Experiments

An empirical study addressed to motivate and evaluate the performance of the proposed learning models has been carried out. A description of the experimental procedure and an overview and discussion of the main results are presented in this section.

6.3.1 Experimental setup

The main characteristics of the experiments are detailed below.

Classifiers. To study the interaction between the distributed learning models and the classifier type, five classification algorithms are considered, namely Random Forest (RF), a support vector machine with RBF Kernel (SVM), the Fisher's linear discriminant (LDA), the classifier based on multinomial logistic regression (Mult), and the XGBoost (eXtreme Gradient Boosting) algorithm (XGB), a fast implementation of the gradient boosting using decision trees. All of them were executed by using different R packages, `randomForest` [78] for RF, `e1071` [93] for SVM, `xgboost`[29] for XGB, and `MASS` and `nnet` [138] for LDA and Mult, respectively. The default parameters are taken in all cases since our concern is not to determine the most efficient inputs but comparing the models under homogeneous conditions. All classifiers provide the options to output belief values in addition to classes

Data sets. Seven data sets are used to analyze the coupling between the proposed method and the underlying classification problem. Five databases (Spambase, KDD

Cup 99, Connect-4, Covertype, and Higgs) contain real data and are available from the UCI Machine Learning Repository [79]. The other two databases (Simul-C2 and Simul-C8) consist of synthetic data generated from simulated classification scenarios. The main characteristics of these data sets are summarized in Table 6.1, including the total number of instances, the dimension d of the feature space, and the number m of classes forming $\mathcal{C}$.

Table 6.1 Data sets characteristics

Dataset	# Instances	# Features	# Classes
Spambase	4,601	57	2
KDD Cup 99	825,050	41	5
Connect-4	67,557	42	3
Covertype	581,012	54	7
Higgs	100,000	28	2
Simul-C2		5	2
Simul-C8		3	8

To get a quick understanding on the nature of these data sets, a very brief description of each one is provided below.

- Spambase. Data set based on the properties of diverse "spam" concept. It includes 4,601 instances corresponding to e-mail messages, 1,813 of which are spam. From the original e-mail messages, 57 attributes were computed, most of them indicating whether a particular word or character frequently occurred in the e-mail.

- KDD Cup 99. Benchmark data set in the intrusion detection field, which contains 5 million instances featured by 41 attributes and 39 types of distinct attacks, grouped into four classes of attack (DoS, Probe, R2L and U2R) and one class of non-attack (normal pattern) [1]. In our study, a smaller subset with 494,021 instances is used as training sample (10% of the original training set). For the test set, we used a subset of 331,029 patterns including new attacks that are not present in the training set. Around 20% of the two datasets are normal patterns (no attacks). The percentages of class labels for the training and test sets are shown in Table 6.2. As can be seen, the percentage of attacks in both data sets is very high, overcoming 80%, where most of the attacks belong to type DoS. Furthermore, it is a very unbalanced data set, with some classes (such as

U2R and R2L) formed by very few instances. Due to these characteristics, KDD Cup 99 becomes a real challenge for the classification task.

Table 6.2 Distribution (in percentage) of normal activities and kinds of attacks in KDD Cup 99 data set.

Type	Training set	Test set
Normal	19.69	19.48
DoS	79.24	73.90
Probe	0.83	1.34
R2L	0.23	5.21
U2R	0.01	0.07

- Connect-4. This data set contains all legal 8-ply positions in the game of Connect-4 in which neither player has won yet, and where the next move is not forced. The dataset contains 67,557 instances represented by 42 attributes indicating whether a particular board position is occupied by the first player, the second player or it is black. The outcome class indicates if the attributes lead to a win, loss, or draw for the first player.

- Covertype. It contains the forest cover type for 30 x 30 meter cells obtained from US Forest Service (USFS) Region 2 Resource Information System (RIS). The database has 581,012 instances. The feature vectors are measurements of 54 cartographic variables used to predict the forest cover type (seven types are available).

- Higgs. The Higgs boson data set was generated using Monte Carlo simulations of physics events. The feature vectors include: 21 attributes with kinematic properties measured by the particle detectors in the accelerator, and 7 attributes with high-level features derived by physicists from the first 21 attributes. The target is to determine whether or not an event corresponds to the Higgs boson. In our study, a subset of 100,000 instances of the original database is considered.

- Simul-C2. Consider a square grid of size 3 in dimension 5 and, centered at each grid node, a 5-dimensional Gaussian distribution with uncorrelated components of equal variance 0.05^2. Each Gaussian is assigned to one of $m = 2$ possible classes at random. In this scenario, an identical number of data are drawn out from each Gaussian to form our first synthetic data set. Figure 6.3 provides an intuition on the structure of Simul-C2 in dimension 2. This scenario lets us have an exact knowledge of the complexity of the classification task in order to derive some insight into the results.

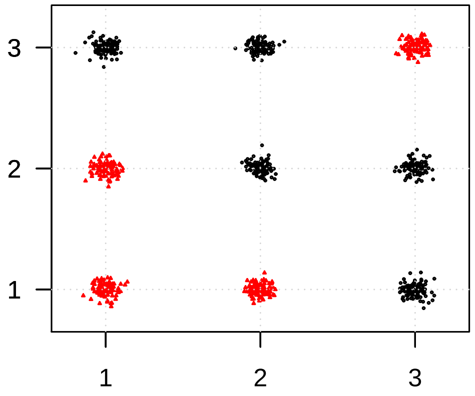


Fig. 6.3 Plot of a simulated trial from a 2-dimensional version of Simul-C2 scenario. Color identifies the class.

- SIMUL-C8. Synthetic data set generated in a similar way as Simul-C2, but now with 3-dimensional Gaussian distributions randomly assigned to $m = 8$ classes.

Sample size. At each experimental trial, the sizes of both the training sample $\mathcal{Z}$ and the test sample $\mathcal{T}$ are fixed to 500, i.e. $n = t = 500$. The training sample is then equidistributed between the nodes so that $n_i = 500/p$, for all $i = 1, \dots, p$.

Balancedness. Since no constraints on the fragmentation scheme are imposed, it is interesting to check the behavior of our learning model with balanced and unbalanced nodes, i.e. nodes exhibiting similar or different distributions, respectively. The balanced scenarios are recreated by allocating instances to each node at random and without replacement while maintaining the class proportions. The unbalanced scenarios are set up as follows. First, one node with the same class proportions as the entire training set is formed. For the rest of nodes, the class proportions are perturbed by multiplying each one of them by a random number uniformly generated between 0.3 and 1.7, and then normalizing. In consecutive nodes, the overall class proportions are updated on the basis of the number of remaining training instances, and sampling without replacement is always carried out.

Partition size. To assess the classification accuracy as data fragmentation increases, the training set was randomly split into 2, 4, 7, 11 and 15 nodes. The unique randomization restrictions are imposed by the class proportions at each node, which depend on whether a balanced or unbalanced scenario is considered.

Decision Rules. The belief values generated by the classifiers at each node are combined according to the five decision rules enumerated in Section 6.2.6, namely the Product (PROD), Sum (SUM), Max (MAX), Min (MIN) and Majority (MAJ) rules.

6.3.2 Some motivating experiments

By construction, the proposed learning models take into account the distances between the probability distributions of the features in the population and within each node. The heuristic is that, in general, smaller distributional distances between training and test sets tend to produce better classification results. Indeed, the key issue is how these distances should be jointly used to attain this improvement. Beyond this issue, a pair of motivating experiments designed to provide empirical support for this heuristic have been carried out. The first experiment consisted in checking for the existence of negative correlation between distributional distance and classification accuracy. In the second one, a distributed scenario is considered, and then the proportion of times that the node with the smallest distributional distance produces the best classification accuracy is measured.

For these specific experiments, the Spambase data set is used and the within-node distributions are generated according to the unbalancing approach described in Section 6.3.1. The number of nodes is set to $p = 5$ and a training sample of size $n_i = 200$ is used at each node to train a Random Forest classifier.

Considering test samples with the same size, $t = 200$, the first experiment consisted in measuring the distributional distances between training and test samples using the energy distance d_i introduced in Section 6.2.4, and simultaneously computing the proportion of test data correctly classified at each node. This process was performed for a large number of trials, and the outputs are plotted in Figure 6.4. A clear negative correlation between distributional distance and classification accuracy is observed, thus supporting the argument that less distant nodes tend to produce better classification accuracy.

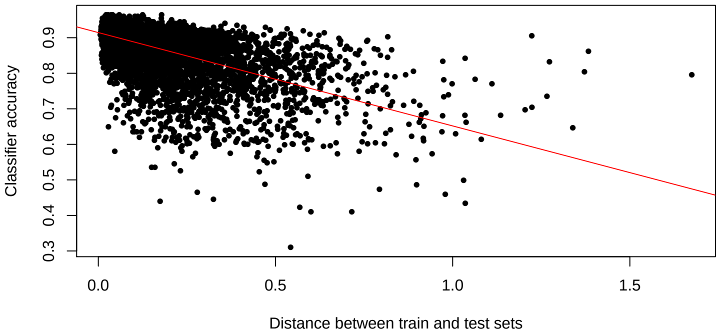


Fig. 6.4 Correlation between a distributional distance (the energy statistic) and classifier accuracy.

In the second experiment, the test sample size is not constant at all trials, taking values moving from 200 to 3200. Notice that the distributional distance becomes more accurately approximated as test sample size increases, and therefore the classification accuracy should be also higher. In a five node distributed scenario, the expected proportion of times that a node picked at random produces the best classifier is 0.2. Table 6.3 shows the proportion of times that the node with the smallest distributional distance produced the highest classification accuracy in our experiment, denoted by $p_{\min(d_i)}$. It is observed that the node with the smallest distributional distance becomes the best one in an increasing proportion with the test sample size, always above the baseline proportion 0.2, until it is approximately doubled.

Table 6.3 Proportion of times that the smallest distributional distance leads to the best classifier node ($p_{\min(d_i)}$) against the test sample size (t).

t	200	1200	2200	3200
$p_{\min(d_i)}$	0.28	0.35	0.37	0.37

In sum, these first experiments empirically illustrate the interest in distributed learning models regarding distributional distances between training nodes and test samples. We propose models taking into consideration this principle, but in addition, they take advantage of combining efficiently the classifiers produced by each node instead of simply selecting one of them.

6.3.3 Results

The accuracy of the two weighting criteria (pNW and pIW) described in Section 6.2.5 was checked on all the combinations of parameters involved in our experimental setup, namely classifiers, data sets, partition sizes, decision rules, and balanced and unbalanced scenarios (Section 6.3.1). For comparison purposes, accuracy results based on a standard unweighted distributed model (UW) and a non-distributed model (ND) were also obtained. The ND model uses the entire data set $\mathcal{Z}$ to train a unique classifier. Hence, ND is expected to achieve the highest classification accuracy, and its results can be taken as an upper reference level. Besides the classification accuracy, values of precision and recall were also evaluated. Since very similar conclusions are obtained, for the sake of clarity in the presentation, the results based on precision and recall are provided in the Appendix B.

For each combination of parameters, the experiment was replicated $N = 300$ times and average classification accuracy values were obtained for each learning model. In order to examine how the learning models interact with the different parameters, the average results were aggregated in different ways. For example, Table 6.4 shows the average accuracy attained with each classifier. It is observed that the weighted models interact better with SVM, LDA and Mult than with Random Forest and XGBoost in the unbalanced setting (see Figure 6.5). In particular, the most significant improvement rates due to the pIW model in the unbalanced setup are observed for Mult and SVM. In the latter case, this may be connected with the fact that SVM with Gaussian kernel and energy distance are based on Euclidean inter-point distances.

The average results aggregated by decision rule are reported in Table 6.5 and graphically represented using bar charts in Figure 6.6. Regardless of whether the partitioning is balanced or not, the SUM rule produces the best average results with the three distributed models. This result is consistent with the experimental findings in Kittler *et al.* [72] and with our theoretical arguments in Section 6.2 (Remark 3 in Section 6.2.7).

Table 6.5 and Figure 6.6 also allow to compare the average accuracy attained with the different models. Except for the MIN and PROD rules, the highest accuracy values are obtained with the per-Instance Weighting. Nevertheless, the MIN rule fairly produces the worst results and no differences between weighting approaches are observed with the PROD rule. Therefore, it is concluded that the per-Instance Weighting approach fairly leads to the best results.

Overall, pNW performs worse than UW on average. This behavior is somewhat surprising in the light of the results showed in the motivating experiments of Section 6.3.2.

Table 6.4 Average classification accuracy values conditional on classifier type. Rows under the last sub-table (MEAN) show the averages over all trials, including balanced and unbalanced scenarios. The lack of results for ND in the first two sub-tables is due to the ND model assumes non-distributed data, i.e. no partitions (balanced or unbalanced) are considered.

	Classifier				
Model	RF	SVM	XGB	LDA	Mult
BALANCED					
pNW	0.7087	0.5852	0.6923	0.5768	0.6084
pIW	0.7173	0.5908	0.7016	0.5826	0.6168
UW	0.7131	0.5881	0.6964	0.5769	0.6066
UNBALANCED					
pNW	0.6935	0.5804	0.6843	0.5727	0.6035
pIW	0.7059	0.5921	0.6977	0.5863	0.6186
UW	0.6996	0.5784	0.6909	0.5749	0.6022
MEAN					
pNW	0.7011	0.5828	0.6883	0.5747	0.6060
pIW	0.7116	0.5915	0.6997	0.5845	0.6177
UW	0.7063	0.5832	0.6937	0.5759	0.6044
ND	0.7566	0.6719	0.7398	0.6315	0.6423

We guess that this may be caused by the joint effect of two circumstances, namely the global character of the per-Node weights (see Remark 1 in Section 6.2.7) and the noise increase generated by the variability of these weights (Figure 6.4 illustrates this variability).

The average accuracies aggregated by partition size are shown in Figure 6.7. Significant degradation of accuracy with the number of nodes is evident for all combinations of decision rule and learning model in balanced and unbalanced scenarios. For all the models, the MIN rule degrades faster with fragmentation, although it is very competitive with UW and pNW for a small number of nodes. As the best-performing pIW approach is considered, the MIN rule is substantial and uniformly the worst decision rule. SUM, MAJ and MAX exhibit similar performance, with SUM having a slight edge. The good behavior of pIW deserves particular attention. Note that, except for the MIN rule, the pIW approach always produces the highest percentages of correct classification for all the levels of fragmentation regardless of the used rule.

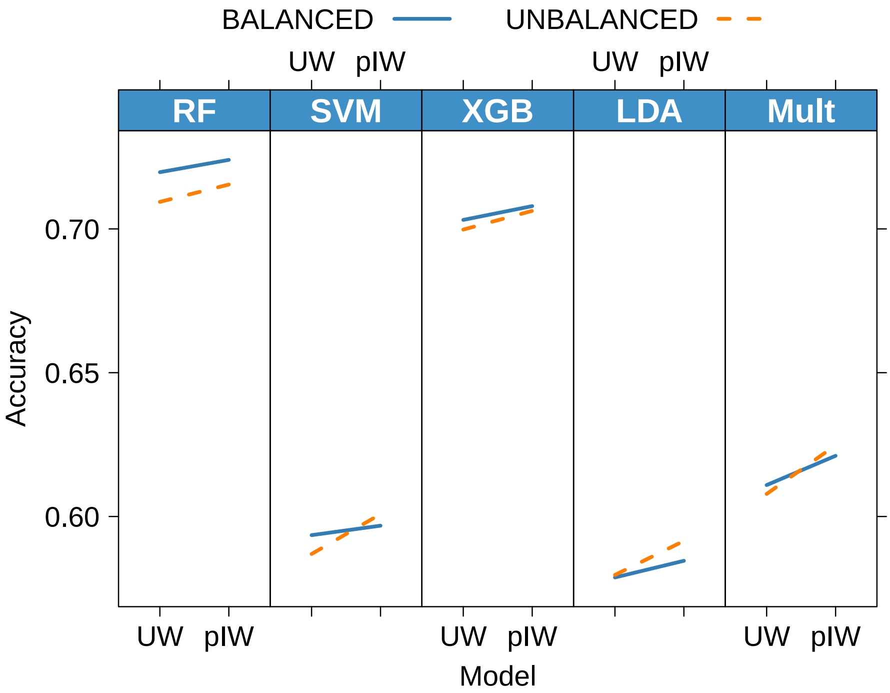


Fig. 6.5 Accuracy-based interaction plot to check the joint effect of classifier, learning model and scenario.

Table 6.5 Average classification accuracy values conditional on the decision rules.

	Decision rule					
Model	MAJ	MAX	MIN	PROD	SUM	Mean
BALANCED						
pNW	0.6442	0.6252	0.6184	0.6347	0.6491	0.6343
pIW	0.6582	0.6461	0.6066	0.6345	0.6639	0.6418
UW	0.6456	0.6283	0.6205	0.6347	0.6519	0.6362
Mean	0.6493	0.6332	0.6152	0.6346	0.6549	0.6374
UNBALANCED						
pNW	0.6407	0.6191	0.5977	0.6276	0.6493	0.6269
pIW	0.6652	0.6513	0.5823	0.6275	0.6743	0.6401
UW	0.6454	0.6189	0.6008	0.6277	0.6531	0.6292
Mean	0.6504	0.6297	0.5936	0.6276	0.6589	0.6321
ND	—	—	—	—	—	0.6884

All our results allow to conclude that the favorable effects of the weighting approaches are more important in unbalanced scenarios. In particular, the amount

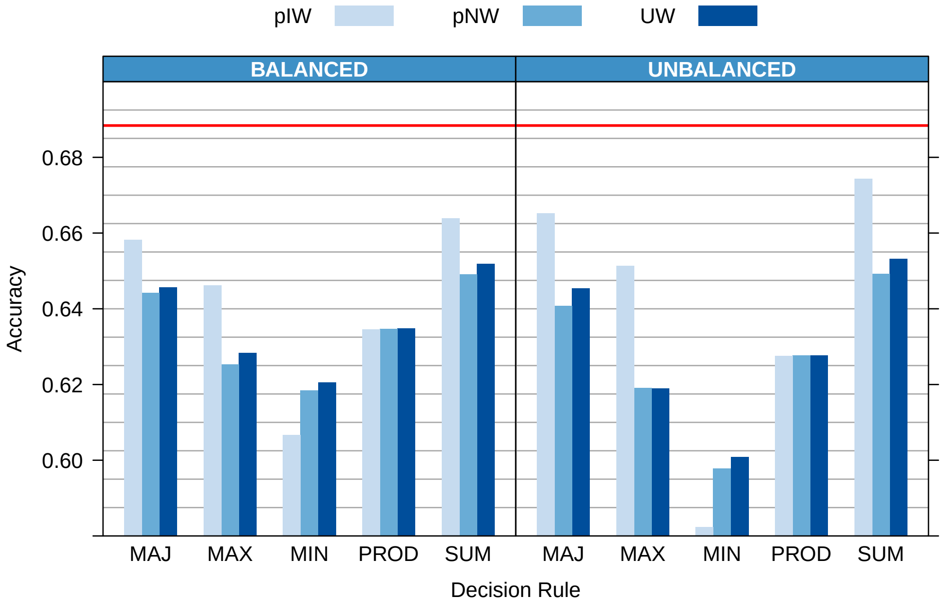


Fig. 6.6 Average classification accuracy values aggregated by decision rules. The horizontal red line indicates the average accuracy for the ND model.

of accuracy improvement produced by the per-Instance Weighting model is clearly stronger when unbalancing.

Figure 6.8 shows separately the average results for each data set and provides additional insight into the behavior of the proposed models. Concerning the KDD Cup 99 data set, it is noticeable the good performance showed by the per-Node Weighting approach in unbalanced scenarios. In fact, pNW and pIW behave very similarly and fairly outperform the Unweighted model. Since KKD Cup 99 exhibits concept drift, this result illustrates that our distributed approaches work well for various types of differences in distribution between nodes and population, no matter how these differences occur. In other words, the effectiveness of our proposal is not restricted to the case of distributional differences caused by a non-uniform partition of the data.

The pIW model reports lower accuracy with Spambase, while it performs slightly better than the other distributed models in Connect-4, Covertype and Higgs, the most complex scenarios in terms of classification. In these cases, the non-distributed model only reaches an accuracy around 0.6, and the distributed approaches are reasonably close to this proportion for all partition sizes. An atypical behavior is observed for Connect-4 since the classification accuracy does not monotonically decrease with the number of partitions. So, in this particular case, it looks more likely that the local scenarios generated by splitting the data lead to an easier classification task.

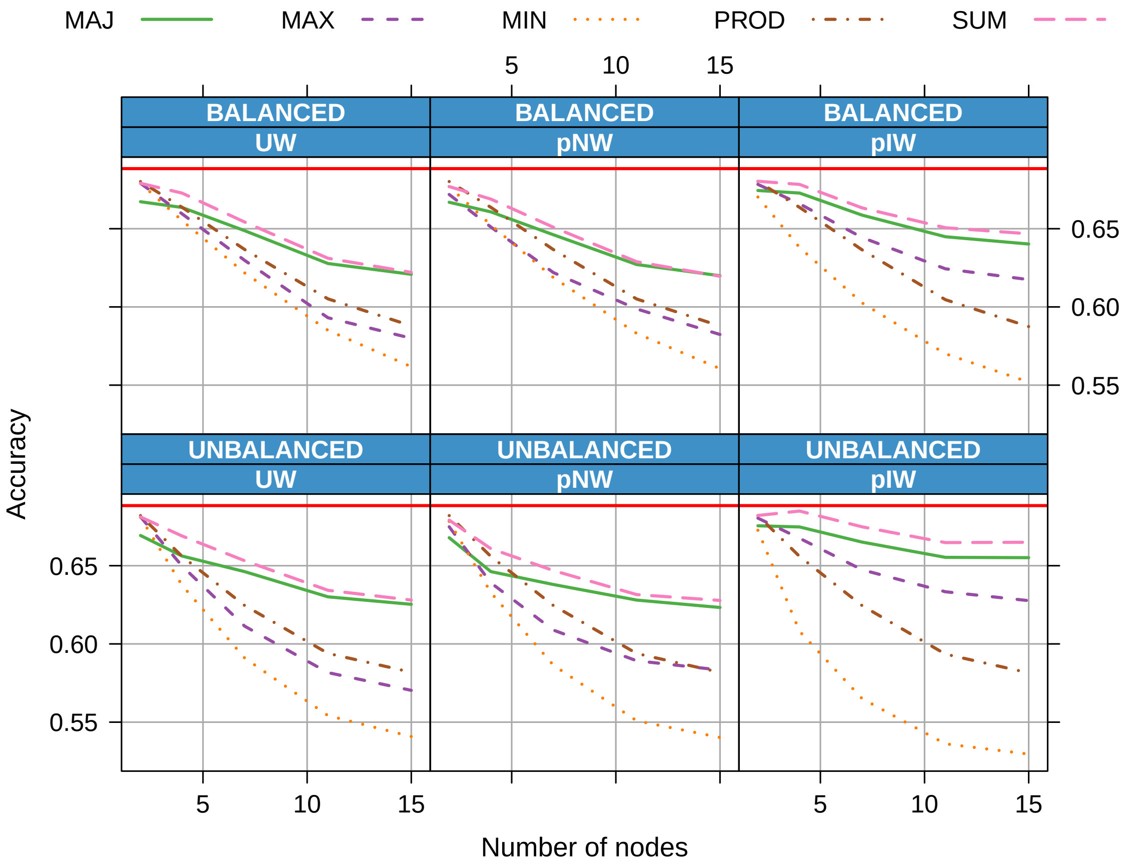


Fig. 6.7 Average accuracy as function of the partition size. The horizontal red lines indicate the average accuracy for the ND model.

In addition to knowing the exact underlying distributions, the analysis of simulated data is free of limitations to generate nodes maintaining the required regularity and provides insight into the level of difficulty. The results for our simulated data sets, Simul-C2 and Simul-C8, are particularly interesting. In both cases, pIW clearly draws the best results, and the differences are more substantial in the unbalanced setting. In the scenario with only two classes, degradation with the number of nodes is almost prevented in the balanced setting, and the results end up surpassing the non-distributed approach in Simul-C2. In the non-distributed approach, a linear classifier will have bad performance in this scenario, since the classification frontier is non linear. When distributed, different local models work better. In Simul-C8, degradation of pIW with the fragmentation is more marked, but still less severe than in the case of pNW and UW.

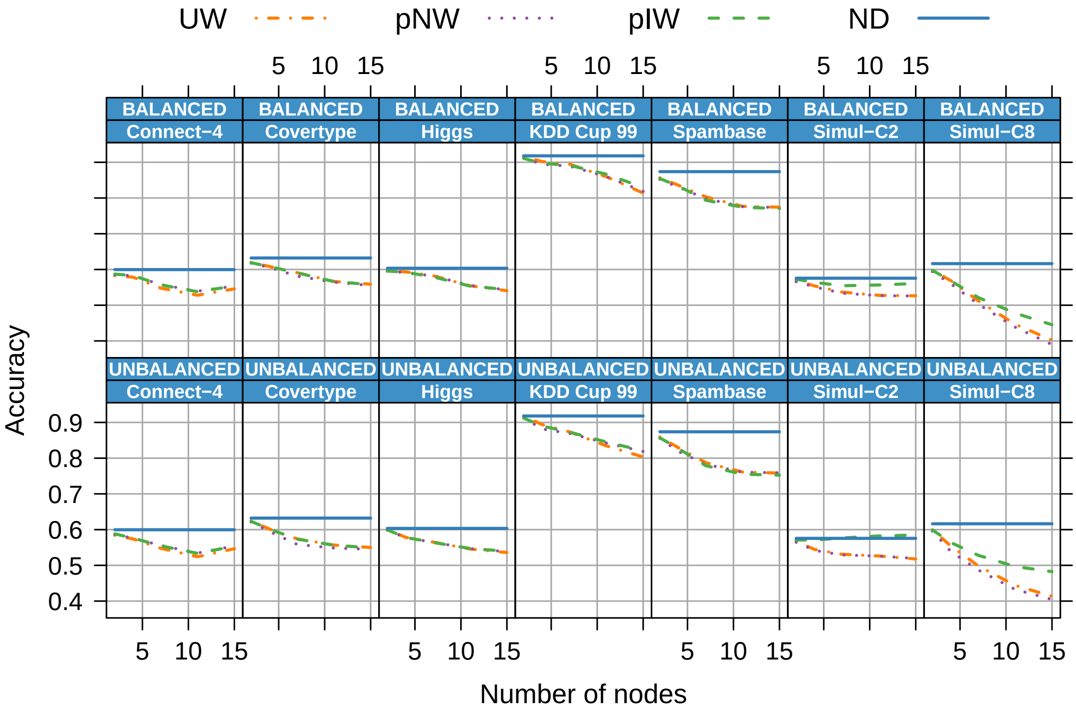


Fig. 6.8 Average accuracy as function of the partition size for each data set. The horizontal blue lines indicate the average accuracy for the ND model.

6.4 Computational cost

In addition to the cost of training each classifier, our approach requires calculating the distributional distance between each node and the test set, and the pIW model requires each node to calculate the weights that minimize this distance.

The cost of training the classifiers depends on the chosen method, but the improvement is equivalent to reducing the training sample by the number of nodes, e.g. if a given classifier trains in $\mathcal{O}(n^3)$ then the complexity will be reduced to $\mathcal{O}(n^3/p^3)$. Similarly, calculating the energy statistic has computational complexity $\mathcal{O}((n/p)^2 + t^2 + (n/p)t)$, given n the training size, p the number of nodes that the training set is fragmented into and t the test size. Even though it can be considered a constant factor, we introduce the number of nodes p to highlight the strong computational benefits of this distributed approach.

Finding the individual weights for the pIW approach has the complexity of solving the related quadratic programming problem, and is the usually dominating complexity: Experimentally, this quadratic programming complexity is between $\mathcal{O}(t^2)$ and $\mathcal{O}(t^3)$.

The influence of the training test size on the quadratic programming complexity is linear: $\mathcal{O}(n/p)$.

Assuming the underlying classifiers test in constant time, the complexity of classifying a given test set goes from $\mathcal{O}(t)$ to a worst case of $\mathcal{O}(t^3+n)$ for the pIW version.

6.5 Conclusions and future work

In a distributed classification framework, we have proposed two weighted approaches that combine local classifiers trained at each node to improve overall classification accuracy. The two approaches assume the availability of a test set and are based on the distance between the distributions of the feature vectors of each node and the test set. The first approach, per-Node Weighting, assigns the same weight at each node to all test instances, while the per-Instance Weighting approach achieves finer granularity by allowing distinct weights for each test instance at each node.

Under the general assumption that both the test set and the entire training set are i.i.d. samples from the population in study, we have motivated the proposed weighting criteria and provided theoretical support for the optimality of combining the classifier outcomes using a weighted sum. Our framework makes no assumptions about the structure or distribution of the data across the nodes. In fact, by construction, our classification models are particularly useful to deal with heterogeneity of data among the nodes, which usually happens in real-world distributed data sets. In addition, our technique requires no communication between nodes, preserving data privacy, allowing combination of different classifier models and maximizing computational efficiency.

Our experimental study involving synthetic and real data sets has illustrated the good performance of the proposed models compared to standard classifier combination rules. Overall, the per-Instance Weighting approach achieves the best results. As expected, the improvement is more substantial when treating with unbalanced nodes, under all tested classifiers an partition sizes. Our experiments also illustrate that the sum rule outperforms other alternative decision rules. The per-Node Weighting approach does not achieve improvement over the standard approaches but on the most extreme cases when the individual nodes training sets differ the most.

There are several topics related to our approach to be considered further. It is interesting to check by the usefulness of our approach to select one or a small subset of nodes to perform the classification and evaluating whether a significant degradation in accuracy is observed. Other future research direction involves the study of a linear-time approximation of the pIW approach.

Chapter 7

Conclusions

We have presented tools for the statistical analysis of complex objects, focusing on time series and shapes. We have identified the use of distances as a useful method of analysis when dealing with these kinds of data. A distance-based method for multivariate two-sample homogeneity test has been introduced in Chapter 2. This method works by comparing the distributions of pairwise distances between observations in the samples, transforming a multivariate problem into a univariate one. The proposed test compares favorably in the range of dissimilarities that can be used with it, both in the theoretical and empirical sense. In the theoretical domain, our method retains its consistency under very few restrictive assumptions of the kinds of distances, inherits this property from a celebrated result. The restrictions on the distances are virtually none when data are discrete; when data are continuous, restrictions are stronger. Besides theoretical consistency, the flexibility in the range of distances also appears in the empirical sense, i.e. in the range of alternatives (e.g. scale or location differences) in which the test achieves good statistical power. If a test has theoretical guarantees but does not discriminate well under a realistic sample size because it is too focused in one kind of alternative, its usefulnes will be limited. The proposed test achieves good performance under many alternatives, at the cost of not being the best in each scenario. In fact, it is the only test in our study that gets good power under both location and scale alternatives, which is a problem that has been targeted in recent works in the area. The empirical performance claims are supported by an extensive empirical study, with scenarios that have been already proposed in other tests, not crafted for this new proposal.

In Chapter 3, we apply this test to the domain of time series. We highlight two conclusions: 1) The statistical power of several distance-based tests is greatly improved when proper distances in the time series domain are introduced. We show that dissimilarities originally for clustering and classification of time series translate their good performance to the two-sample problem. The difference between using a traditional standard multivariate test and a distance for time series can go beyond a merely quantitative improvement, we have shown that statistical power can go from zero discriminatory power to full discriminatory power with the proper distance. This means a qualitative change, some scenarios can now be analyzed when previously it was unfeasible to do so. 2) Among the tests we study, the one we proposed in Chapter 2 shows better empirical results most scenarios. This result highlights the importance of the flexibility of a testing procedure, in the sense of sensitivity against different kinds of alternatives. It is difficul to predict in which way a dissimilarity that works well may differenciate the two samples. We explore several distances for time series and observe this effect. These results also strengthen the claims of Chapter 2 with new empirical evidence.

Two sample test statistics are also useful as dissimilarities between sets of multivariate observations or empirical distribution functions. We leverage these tools to define a distance for time series in Chapter 4. We unify two very good performing approaches in the literature of time series classification and clustering, and view them both as methods which compare autoregressive distribution functions of the time series. From this view point, we apply the two-sample statistics to compare time series, removing some limitations and assumptions that were being made in the methods we are using as base. The resulting distance achieves best average results compared to other distances in the literature in the standard database of time series classification, which features domains as electocardiograms, spectrometry, motion sensors, electricity demands, etc. It also achieves good perfomance under a simulation scenario of autoregressive stochastic time series. The proposed distance compares time series in a very parsimonous way, its only assumption is the existence of some autoregressive structure in the series, and this structure is captured in a nonparametric way, without assuming linearity, a degree of smoothness, etc. The relevant time lags in the autoregressive structure can be chosen by the practictioner when they have some meaning it the context of application, and even results with different lags can be compared, without forcing the practitioner to make additional assumptions that other distances introduce by construction. A data driven scheme for selecting the relevant time lags is also provided in the chapter. This makes the distance parameter-free, and coupled with its light assumptions, may render it superior to other more assumption-heavy distances at larger sample sizes.

We apply the same concept of distance between distributions in the distributed classification setting in Chapter 6. We study the scenario of a large dataset that has been partitioned, and each of the partitions must be processed independently from the others, data cannot be shared or merged. This scenario occurs when parallelism is needed but communication between computing nodes is unfeasible, or for privacy reasons, such as hospitals not being allowed to share information about patients but wanting to the best common statistical model of a problem. We highlight two main contributions from this chapter. 1) We formalized the distributed setting from a probabilistic point of view, establishing a theoretical framework for studying the problem. This allowed us to motivate the approach we presented, and gave and explanation for the good performance of a family of ensemble methods based on the sum of the output probabilities of the individual classifiers. The theoretical framework can also serve as base or starting point for further studies. We are using it to motivate a new method in a similar setting, metioned in Section 7.2. 2) We proposed and studied several methods for improving classification accuracy in the distributed setting. By weighting the contributions of each classifier using a distance between their train data and the test data distributions, the error is reduced. The weights of each of the individual classifiers are generated by the minimization of the distance between distributions, so each classifier has a different weight per obervation, and only weights and target class probabilities are shared among the different processing nodes. Distributed classification under our assumptions is still an open problem, our method improved the accuracy results of the reference approach.

We propose FFORMA, a method for times series forecasting in Chapter 5. We assume that a large database of time series is available for training, generated from the same process that the series we are going to forecast. We pose the problem in a similar way as a time series classification problem. Initially, we consider a pool of time series forecasting algorithms and try to predict which method will be the best for a time series. We relax the assumptions common in classification and consider minimizing the forecasting error of the chosen method, instead of minimizing the traditional classification error. FFORMA produces weights and the individual forecasting methods are combined by a weighting average. The input to the model are features extracted from the series. FFORMA is general in the sense on the pool of methods and set of features, forecasting error measures, etc. can be changed to suit any specific situation. We applied the approach to the M4 Time Series Forecasting Competition, the main competition in forecasting, achieving second best results in point forecasting and prediction interval accuracy.

As part of this thesis, several contributions to free open source software were made:

- TSclust: A package for time series clustering by the author of this thesis, added new procedures and bug fixes.
- M4comp2018: the database of time series used for the M4 competition as an easily accesible R package for the research community.
- M4metalearning: The implementation of the FFORMA method for time series forecasting.
- tsfeatures: a package for extracting time series features, contributed with the implementation of two features.
- xgboost: tree-based classifier, we contributed introducing custom objective functions for multiclass classification.

7.1 Contributions by topic

- Two Sample Homogeneity Tests:
  - New distance-based test, general against multiple alternatives.
  - Used two-sample statistics as a measures of divergence between distributions to time series and distributed classification.
- Time Series:
  - Two sample test applied to time series.
  - A new distance between TS based on lagged distributions
  - A forecasting method based on meta learning.
- Big Data:
  - Distributed classification: We propose a classification scheme able to tackle large dataset that is partitioned, when processing the whole dataset would be computationally unfeasible.
  - A time series forecasting method based on meta learning. We effectively exploit the availability of a large dataset of time series to improve the forecasing of individual time series.
- Clustering:

  - A distance between time series for clustering.

- Classification:

  - A distance between time series for classification.
  - A method for distributed classification.

7.2 Future work

In addition to the future work mentioned in each chapter, there are some lines of research we would like to highlight. The first is a data-driven way for the lag selection in the distance proposed in Chapter 4 that is not greedy, considering all possible lags simultaneously through multiple kernel learning. The second is the study of different measures of divergence between distributions for the methods in Chapters 4 and 6. Also from the starting point of the theoretical framework of distributed learning of Chapter 6, we intend to explore a less restrictive scenario when data is not previously distributed among nodes, but can be partitioned. When a large quantity of unlabeled data is available at the moment of partitioning, this information and can be used to influence the parittion process (i.e. how the data is divided among the computing nodes) with the objective of improving the accuracy of the joint classifier. Regarding our forecasting method FFORMA, we would like to explore a model that directly minimizes the error of a linear combination of single methods, using techniques different from gradient tree boosting.

References

[1] (2017). Kdd cup 99 dataset. http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html. [Online; accessed April-2017].

[2] (2018). Introduction to xgboost. https://xgboost.readthedocs.io/en/latest/tutorials/model.html. Accessed: 2018-11-16.

[3] (2018). M4 competitor's guide. https://www.m4.unic.ac.cy/wp-content/uploads/2018/03/M4-Competitors-Guide.pdf. Accessed: 2018-09-26.

[4] Acuna, E. and the CASTLE research group at The University of Puerto Rico-Mayaguez (2015). *dprep: Data Pre-Processing and Visualization Functions for Classification*. R package version 3.0.2.

[5] Aghabozorgi, S., Shirkhorshidi, A. S., and Wah, T. Y. (2015). Time-series clustering–a decade review. *Information Systems*, 53:16–38.

[6] Alba-Fernández, V., Jiménez-Gamero, M., and Muñoz-García, J. (2008). A test for the two-sample problem based on empirical characteristic functions. *Computational Statistics & Data Analysis*, 52(7):3730 – 3748.

[7] Anderson, M. J. (2006). Distance-based tests for homogeneity of multivariate dispersions. *Biometrics*, 62(1):245–253.

[8] Anderson, T. W. (1962). On the distribution of the two-sample cramér-von mises criterion. *The Annals of Mathematical Statistics*, 33(3):1148–1159.

[9] Bagnall, A., Lines, J., Bostrom, A., Large, J., and Keogh, E. (2016). The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. *Data Mining and Knowledge Discovery*, Online First.

[10] Baringhaus, L. and Franz, C. (2004). On a new multivariate two-sample test. *Journal of Multivariate Analysis*, 88(1):190–206.

[11] Bates, J. M. and Granger, C. W. J. (1969). The combination of forecasts. *Journal of the Operational Research Society*, 20(4):451–468.

[12] Batista, G. E., Keogh, E. J., Tataw, O. M., and de Souza, V. M. (2014). Cid: an efficient complexity-invariant distance for time series. *Data Mining and Knowledge Discovery*, 28(3):634–669.

[13] Bekkerman, R., Bilenko, M., and Langford, J. (2011). Scaling up machine learning: Parallel and distributed approaches. In *Proceedings of the 17th ACM SIGKDD International Conference Tutorials*, KDD '11 Tutorials, pages 4:1–4:1, New York, NY, USA. ACM.

[14] Bello-Orgaz, G., Jung, J. J., and Camacho, D. (2016). Social big data: Recent achievements and new challenges. *Information Fusion*, 28:45–59.

[15] Berndt, D. J. and Clifford, J. (1994). Using dynamic time warping to find patterns in time series. In *KDD workshop*, volume 10, pages 359–370. Seattle, WA.

[16] Berrendero, J. R., Cuevas, A., and Pateiro-López, B. (2016). Shape classification based on interpoint distance distributions. *Journal of Multivariate Analysis*, 146:237–247.

[17] Bickel, P. J. and Breiman, L. (1983). Sums of functions of nearest neighbor distances, moment bounds, limit theorems and a goodness of fit test. *Ann. Probab.*, 11(1):185–214.

[18] Biswas, M., Mukhopadhyay, M., and Ghosh, A. K. (2014). A distribution-free two-sample run test applicable to high-dimensional data. *Biometrika*, pages 913–926.

[19] Blum, A. L. and Langley, P. (1997). Selection of relevant features and examples in machine learning. *Artificial intelligence*, 97(1-2):245–271.

[20] Brandmaier, A. M. (2015). pdc: An r package for complexity-based clustering of time series. *Journal of Statistical Software*, 67(5):1–23.

[21] Breiman, L. (1999). Pasting small votes for classification in large databases and on-line. *Machine learning*, 36(1):85–103.

[22] Caiado, J., Crato, N., and Peña, D. (2006). A periodogram-based metric for time series classification. *Computational Statistics & Data Analysis*, 50(10):2668–2684.

[23] Casado de Lucas, D. (2010). *Classification techniques for time series and functional data*. PhD thesis, Universidad Carlos III de Madrid.

[24] Chan, P. K., Stolfo, S. J., et al. (1993). Toward parallel and distributed learning by meta-learning. In *AAAI workshop in Knowledge Discovery in Databases*, pages 227–240.

[25] Chandola, V., Cheboli, D., and Kumar, V. (2009). Detecting anomalies in a time series database. *Computer Science Department, University of Minnesota, Tech. Rep.*

[26] Chen, B., Zhou, H., and Chen, X. (2018). E-embed: A time series visualization framework based on earth mover's distance. *Journal of Visual Languages & Computing*, 48:110 – 122.

[27] Chen, H. and Friedman, J. H. (2016). A new graph-based two-sample test for multivariate and object data. *Journal of the American Statistical Association*, (just-accepted):1–41.

[28] Chen, L. and Ng, R. (2004). On the marriage of lp-norms and edit distance. In *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*, pages 792–803. VLDB Endowment.

[29] Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794. ACM.

[30] Chen, Y., Keogh, E., Hu, B., Begum, N., Bagnall, A., Mueen, A., and Batista, G. (2015). *The UCR Time Series Classification Archive.*

[31] Choulakian, V., Lockhart, R., and Stephens, M. (1994). Cramér-von mises statistics for discrete distributions. *Canadian Journal of Statistics*, 22(1):125–137.

[32] Chwialkowski, K., Ramdas, A., Sejdinovic, D., and Gretton, A. (2015). Fast two-sample testing with analytic representations of probability measures. In *Proceedings of the 28th International Conference on Neural Information Processing Systems*, NIPS'15, pages 1981–1989, Cambridge, MA, USA. MIT Press.

[33] Clemen, R. (1989). Combining forecasts: a review and annotated bibliography with discussion. *International Journal of Forecasting*, 5:559–608.

[34] Cuadras, C. (1989). Distance analysis in discrimination and classification using both continuous and categorical variables. *Statistical data analysis and inference*, pages 459–473.

[35] Daumé, H., Phillips, J. M., Saha, A., and Venkatasubramanian, S. (2012). Efficient protocols for distributed classification and optimization. In *International Conference on Algorithmic Learning Theory*, pages 154–168. Springer.

[36] Di Gesu, V. and Starovoitov, V. (1999). Distance-based functions for image comparison. *Pattern Recognition Letters*, 20(2):207–214.

[37] Díaz, S. P. and Vilar, J. A. (2010). Comparing several parametric and nonparametric approaches to time series clustering: a simulation study. *Journal of Classification*, 27(3):333–362.

[38] Dick, T., Li, M., Pillutla, V. K., White, C., Balcan, M., and Smola, A. J. (2015). Data driven resource allocation for distributed learning. *CoRR*, abs/1512.04848.

[39] Dietterich, T. G. (2000). Ensemble methods in machine learning. In *International workshop on multiple classifier systems*, pages 1–15. Springer.

[40] Du Plessis, M. C. and Sugiyama, M. (2014). Semi-supervised learning of class balance under class-prior change by distribution matching. *Neural Networks*, 50:110–119.

[41] Engle, R. F. (1982). Autoregressive conditional heteroscedasticity with estimates of the variance of united kingdom inflation. *Econometrica: Journal of the Econometric Society*, pages 987–1007.

[42] Ferraty, F. and Vieu, P. (2006). *Nonparametric Functional Data Analysis: Theory and Practice (Springer Series in Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

[43] Foster, P., Dixon, S., and Klapuri, A. (2015). Identifying cover songs using information-theoretic measures of similarity. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, 23(6):993–1005.

[44] Friedman, J. H. (2003). On multivariate goodness of fit and two sample testing. *eConf*, C030908(SLAC-PUB-10325, PHYSTAT-2003-THPD002):311–313.

[45] Friedman, J. H. and Rafsky, L. C. (1979). Multivariate generalizations of the wald-wolfowitz and smirnov two-sample tests. *The Annals of Statistics*, pages 697–717.

[46] g, A., Estévez, M. G., Varela, M., and Vilar, J. A. (2015). Annual trend patterns of phytoplankton species abundance belie homogeneous taxonomical group responses to climate in the ne atlantic upwelling. *Marine Environmental Research*, 110:81 – 91.

[47] Ghosh, A. K. and Biswas, M. (2016). Distribution-free high-dimensional two-sample tests based on discriminating hyperplanes. *TEST*, 25(3):525–547.

[48] Good, P. (2002). Extensions of the concept of exchangeability and their applications. *Journal of Modern Applied Statistical Methods*, 1(2):243–247.

[49] Górecki, T. and Łuczak, M. (2013). Using derivatives in time series classification. *Data Mining and Knowledge Discovery*, 26(2):310–331.

[50] Gretton, A., Borgwardt, K. M., Rasch, M., Schölkopf, B., and Smola, A. J. (2007). A kernel method for the two-sample-problem. In *Advances in neural information processing systems*, pages 513–520.

[51] Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., and Smola, A. (2012a). A kernel two-sample test. *Journal of Machine Learning Research*, 13(Mar):723–773.

[52] Gretton, A., Sejdinovic, D., Strathmann, H., Balakrishnan, S., Pontil, M., Fukumizu, K., and Sriperumbudur, B. K. (2012b). Optimal kernel choice for large-scale two-sample tests. In Pereira, F., Burges, C. J. C., Bottou, L., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 25*, pages 1205–1213. Curran Associates, Inc.

[53] Hall, P. and Tajvidi, N. (2002). Permutation tests for equality of distributions in high-dimensional settings. *Biometrika*, 89(2):359–374.

[54] Hall, P. and Van Keilegom, I. (2007). Two-sample tests in functional data analysis starting from discrete data. *Statistica Sinica*, pages 1511–1531.

[55] Harchaoui, Z., Vallet, F., Lung-Yut-Fong, A., and Cappé, O. (2009). A regularized kernel-based approach to unsupervised audio segmentation. In *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1665–1668. IEEE.

[56] Hennig, C. and Liao, T. F. (2013). How to find an appropriate clustering for mixed-type variables with application to socio-economic stratification. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 62(3):309–369.

[57] Henning, W. and Srivastava, A. (2016). A two-sample test for statistical comparisons of shape populations. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–9.

[58] Henze, N. (1984). On the number of random points with nearest neighbor of the same type and a multivariate two-sample test. *Metrika*, 31:259–273.

[59] Henze, N. (1988). A multivariate two-sample test based on the number of nearest neighbor type coincidences. *The Annals of Statistics*, pages 772–783.

[60] Hjellvik, V. and Tjøstheim, D. (1995). Nonparametric tests of linearity for time series. *Biometrika*, 82(2):351–368.

[61] Huang, J., Smola, A. J., Gretton, A., Borgwardt, K. M., and Scholkopf, B. (2006). Correcting sample selection bias by unlabeled data. In *Proceedings of the 19th International Conference on Neural Information Processing Systems*, NIPS'06, pages 601–608, Cambridge, MA, USA. MIT Press.

[62] Hušková, M. and Meintanis, S. G. (2008). Tests for the multivariate k-sample problem based on the empirical characteristic function. *Journal of Nonparametric Statistics*, 20(3):263–277.

[63] Hyndman, R., Athanasopoulos, G., Bergmeir, C., Caceres, G., Chhay, L., O'Hara-Wild, M., Petropoulos, F., Razbash, S., Wang, E., and Yasmeen, F. (2018a). *forecast: Forecasting functions for time series and linear models*. R package version 8.3.

[64] Hyndman, R., Wang, E., Kang, Y., and Talagala, T. (2018b). *tsfeatures: Time Series Feature Extraction*. R package version 0.1.

[65] Hyndman, R. J., Wang, E., and Laptev, N. (2015). Large-scale unusual time series detection. In *Data Mining Workshop (ICDMW), 2015 IEEE International Conference on*, pages 1616–1619. IEEE.

[66] Jaccard, P. (1901). Étude comparative de la distribution florale dans une portion des alpes et des jura. *Bull Soc Vaudoise Sci Nat*, 37:547–579.

[67] Jeong, Y.-S., Jeong, M. K., and Omitaomu, O. A. (2011). Weighted dynamic time warping for time series classification. *Pattern Recognition*, 44(9):2231–2240.

[68] Jones, M. C. and Pewsey, A. (2009). Sinh-arcsinh distributions. *Biometrika*, 96(4):761–780.

[69] Kalpakis, K., Gada, D., and Puttagunta, V. (2001). Distance measures for effective clustering of arima time-series. In *Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on*, pages 273–280. IEEE.

[70] Kang, Y., Hyndman, R. J., and Smith-Miles, K. (2017). Visualising forecasting algorithm performance using time series instance spaces. *International journal of forecasting*, 33(2):345–358.

[71] Kiefer, J. (1959). K-sample analogues of the kolmogorov-smirnov and cramer-v. mises tests. *Ann. Math. Statist.*, 30(2):420–447.

[72] Kittler, J., Hatef, M., Duin, R. P., and Matas, J. (1998a). On combining classifiers. *IEEE transactions on pattern analysis and machine intelligence*, 20(3):226–239.

[73] Kittler, J., Hatef, M., Duin, R. P. W., and Matas, J. (1998b). On combining classifiers. *IEEE Trans. Pattern Anal. Mach. Intell.*, 20(3):226–239.

[74] Kück, M., Crone, S. F., and Freitag, M. (2016). Meta-Learning with Neural Networks and Landmarking for Forecasting Model Selection - An Empirical Evaluation of Different Feature Sets Applied to Industry Data Meta-Learning with Neural Networks and Landmarking for Forecasting Model Selection. *International Joint Conference on Neural Networks*, pages 1499–1506.

[75] Lazarevic, A. and Obradovic, Z. (2001). The distributed boosting algorithm. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 311–316. ACM.

[76] Lemke, C. and Gabrys, B. (2010). Meta-learning for time series forecasting and forecast combination. *Neurocomputing*, 73(10):2006 – 2016. Subspace Learning / Selected papers from the European Symposium on Time Series Prediction.

[77] Leucht, A. and Neumann, M. H. (2013). Dependent wild bootstrap for degenerate u- and v-statistics. *Journal of Multivariate Analysis*, 117:257 – 280.

[78] Liaw, A. and Wiener, M. (2002). Classification and regression by randomforest. *R News*, 2(3):18–22.

[79] Lichman, M. (2013). UCI machine learning repository. http://archive.ics.uci.edu/ml.

[80] Lines, J. and Bagnall, A. (2015a). Time series classification with ensembles of elastic distance measures. *Data Mining and Knowledge Discovery*, 29(3):565–592.

[81] Lines, J. and Bagnall, A. (2015b). Time series classification with ensembles of elastic distance measures. *Data Mining and Knowledge Discovery*, 29(3):565–592.

[82] Lines, J., Taylor, S., and Bagnall, A. (2018). Time series classification with hive-cote: The hierarchical vote collective of transformation-based ensembles. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 12(5):52.

[83] Lockhart, R. A., Spinelli, J. J., and Stephens, M. A. (2007). Cramér-von mises statistics for discrete distributions with unknown parameters. *Canadian Journal of Statistics*, 35(1):125–133.

[84] Lopes, M., Jacob, L., and Wainwright, M. J. (2011). A more powerful two-sample test in high dimensions using random projection. In *Advances in Neural Information Processing Systems*, pages 1206–1214.

[85] Maa, J.-F., Pearl, D. K., and Bartoszyński, R. (1996). Reducing multidimensional two-sample data to one-dimensional interpoint comparisons. *Ann. Statist.*, 24(3):1069–1074.

[86] Maharaj, E. A. (2000). Cluster of time series. *Journal of Classification*, 17(2):297–314.

[87] Makridakis, S., Spiliotis, E., and Assimakopoulos, V. (2018). The m4 competition: Results, findings, conclusion and way forward. *International Journal of Forecasting*.

[88] Marco, B. and Marcello, P. (2005). The interpoint distance distribution as a descriptor of point patterns, with an application to spatial disease clustering. *Statistics in Medicine*, 24(5):753–773.

[89] Marron, J. S., Todd, M. J., and Ahn, J. (2007). Distance-weighted discrimination. *Journal of the American Statistical Association*, 102(480):1267–1271.

[90] Marteau, P.-F. (2009). Time warp edit distance with stiffness adjustment for time series matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(2):306–318.

[91] Martínez-Camblor, P. and de Uña-Álvarez, J. (2009). Non-parametric k-sample tests: Density functions vs distribution functions. *Computational Statistics & Data Analysis*, 53(9):3344 – 3357.

[92] McCallum, A., Nigam, K., et al. (1998). A comparison of event models for naive bayes text classification. In *AAAI-98 workshop on learning for text categorization*, volume 752, pages 41–48. Citeseer.

[93] Meyer, D., Dimitriadou, E., Hornik, K., Weingessel, A., and Leisch, F. (2015). *e1071: Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien.* R package version 1.6-7.

[94] Montero, P. and Vilar, J. A. (2015). Tsclust: An r package for time series clustering. *Journal of Statistical Software*, 62(1):1–43.

[95] Morel, M., Achard, C., Kulpa, R., and Dubuisson, S. (2018). Time-series averaging using constrained dynamic time warping with tolerance. *Pattern Recognition*, 74:77 – 89.

[96] Mori, U., Mendiburu, A., and Lozano, J. A. (2016). Distance measures for time series in r: The tsdist package. *R Journal*, 8(2):451–459.

[97] Olivetti, E., Benozzo, D., Kia, S. M., Ellero, M., and Hartmann, T. (2013). The kernel two-sample test vs. brain decoding. In *Pattern Recognition in Neuroimaging (PRNI), 2013 International Workshop on*, pages 128–131. IEEE.

[98] Osada, R., Funkhouser, T., Chazelle, B., and Dobkin, D. (2002). Shape distributions. *ACM Transactions on Graphics (TOG)*, 21(4):807–832.

[99] Peteiro-Barral, D. and Guijarro-Berdiñas, B. (2013). A survey of methods for distributed machine learning. *Progress in Artificial Intelligence*, 2(1):1–11.

[100] Petitjean, F., Ketterlin, A., and Gançarski, P. (2011). A global averaging method for dynamic time warping, with applications to clustering. *Pattern Recognition*, 44(3):678–693.

[101] Piccolo, D. (1990). A distance measure for classifying arima models. *Journal of Time Series Analysis*, 11(2):153–164.

[102] Prudêncio, R. and Ludermir, T. (2004). Using machine learning techniques to combine forecasting methods. In *Australasian Joint Conference on Artificial Intelligence*, pages 1122–1127. Springer.

[103] Qin, J. and Liang, K.-Y. (2011). Hypothesis Testing in a Mixture Case–Control Model. *Biometrics*, 67(1):182–193.

[104] Quionero-Candela, J., Sugiyama, M., Schwaighofer, A., and Lawrence, N. D. (2009). *Dataset shift in machine learning.* The MIT Press.

[105] Ramdas, A., Reddi, S. J., Póczos, B., Singh, A., and Wasserman, L. (2015a). Adaptivity and computation-statistics tradeoffs for kernel and distance based high dimensional two sample testing. *arXiv preprint arXiv:1508.00655.*

[106] Ramdas, A., Reddi, S. J., Póczos, B., Singh, A., and Wasserman, L. (2015b). On the decreasing power of kernel and distance based nonparametric hypothesis tests in high dimensions. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, pages 3571–3577. AAAI Press.

[107] Rhodes, C. and Morari, M. (1997). The false nearest neighbors algorithm: An overview. *Computers & Chemical Engineering*, 21:S1149–S1154.

[108] Rizzo, M. L. and Székely, G. J. (2010). Disco analysis: A nonparametric extension of analysis of variance. *The Annals of Applied Statistics*, 4(2):1034–1055.

[109] Rodríguez, M. Á., Fernández, A., Peregrín, A., and Herrera, F. (2017). A review of distributed data models for learning. *Hybrid Artificial Intelligent Systems*, page 88.

[110] Ross, G. (2015). Parametric and nonparametric sequential change detection in r: The cpm package. *Journal of Statistical Software*, 66(1):1–20.

[111] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. (2015). Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252.

[112] Sakoe, H. and Chiba, S. (1978). Dynamic programming algorithm optimization for spoken word recognition. *IEEE transactions on acoustics, speech, and signal processing*, 26(1):43–49.

[113] Schäfer, P. (2015). The boss is concerned with time series classification in the presence of noise. *Data Mining and Knowledge Discovery*, 29(6):1505–1530.

[114] Schäfer, P. and Leser, U. (2017). Fast and accurate time series classification with weasel. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 637–646. ACM.

[115] Schilling, M. F. (1986). Multivariate two-sample tests based on nearest neighbors. *Journal of the American Statistical Association*, 81(395):799–806.

[116] Sejdinovic, D., Sriperumbudur, B., Gretton, A., Fukumizu, K., et al. (2013). Equivalence of distance-based and rkhs-based statistics in hypothesis testing. *The Annals of Statistics*, 41(5):2263–2291.

[117] Sheather, S. and Jones, C. (1991). A reliable data-based bandwidth selection method for kernel density estimation. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 53(3):683–690.

[118] Silverman, B. (1976). Limit theorems for dissociated random variables. *Advances in Applied Probability*, 8(4):806–819.

[119] Silversides, K. L. and Melkumyan, A. (2016). A dynamic time warping based covariance function for gaussian processes signature identification. *Computers & Geosciences*, 96:69–76.

[120] Small, M. (2003). Optimal time delay embedding for nonlinear time series modeling. *arXiv preprint nlin/0312011*.

[121] Smith, J. and Wallis, K. F. (2009). A simple explanation of the forecast combination puzzle. *Oxford bulletin of economics and statistics*, 71(3):331–355.

[122] Sokolova, M. and Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, 45(4):427–437.

[123] Srivastava, R., Li, P., and Ruppert, D. (2016). Raptt: An exact two-sample test in high dimensions using random projections. *Journal of Computational and Graphical Statistics*, 25(3):954–970.

[124] Stasinopoulos, D. and Rigby, R. (2007). Generalized additive models for location scale and shape (gamlss) in r. *Journal of Statistical Software*, 23(1):1–46.

[125] Stefan, A., Athitsos, V., and Das, G. (2013). The move-split-merge metric for time series. *IEEE transactions on Knowledge and Data Engineering*, 25(6):1425–1438.

[126] Stegle, O., Denby, K. J., Cooke, E. J., Wild, D. L., Ghahramani, Z., and Borgwardt, K. M. (2010). A robust bayesian two-sample test for detecting intervals of differential gene expression in microarray time series. *Journal of Computational Biology*, 17(3):355–367.

[127] Székely, G. J. and Rizzo, M. L. (2004). Testing for equal distributions in high dimension. *InterStat*, 5:1–6.

[128] Székely, G. J. and Rizzo, M. L. (2013). Energy statistics: A class of statistics based on distances. *Journal of statistical planning and inference*, 143(8):1249–1272.

[129] Székely, G. J. and Rizzo, M. L. (2013). Energy statistics: A class of statistics based on distances. *Journal of Statistical Planning and Inference*, 143(8):1249 – 1272.

[130] Talagala, T. S., Hyndman, R. J., and Athanasopoulos, G. (2018). Meta-learning how to forecast time series. *Technical Report 6/18, Monash University.*

[131] Tamma, A. and Khubchandani, B. L. (2016). Accurate determination of time delay and embedding dimension for state space reconstruction from a scalar time series. *arXiv preprint arXiv:1605.01571.*

[132] Thulin, M. (2014). A high-dimensional two-sample test for the mean using random subspaces. *Computational Statistics & Data Analysis*, 74:26 – 38.

[133] Timmermann, A. (2006). Forecast combinations. In Graham Elliot and Clive W. J. Granger and Allan Timmermann, editor, *Handbook of economic forecasting*, chapter 4, pages 135–196. Amsterdam, North-Holland.

[134] Tsoumakas, G. and Vlahavas, I. (2002). Effective stacking of distributed classifiers. In *Proceedings of the 15th European conference on artificial intelligence*, pages 340–344. IOS Press.

[135] Tsoumakas, G. and Vlahavas, I. (2009). Distributed data mining. *Encyclopedia of Data Warehousing and Mining.*

[136] Upadhyaya, S. R. (2013). Parallel approaches to machine learning—A comprehensive survey. *Journal of Parallel and Distributed Computing*, 73(3):284–292.

[137] Vapnik, V. (1998). *Statistical learning theory. 1998*, volume 3. Wiley, New York.

[138] Venables, W. N. and Ripley, B. D. (2002). *Modern Applied Statistics with S.* Springer, New York, fourth edition. ISBN 0-387-95457-0.

[139] Vilar, J. A., Alonso, A. M., and Vilar, J. M. (2010). Non-linear time series clustering based on non-parametric forecast densities. *Computational Statistics & Data Analysis*, 54(11):2850–2865.

[140] Vilar, J. A. and Pértega, S. (2004). Discriminant and cluster analysis for gaussian stationary processes: Local linear fitting approach. *J. Nonparametr. Stat.*, 16(3-4):443–462.

[141] Vlachos, M., Kollios, G., and Gunopulos, D. (2002). Discovering similar multidimensional trajectories. In *Data Engineering, 2002. Proceedings. 18th International Conference on*, pages 673–684. IEEE.

[142] Wang, J., Zhang, T., Sebe, N., Shen, H. T., et al. (2018). A survey on learning to hash. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):769–790.

[143] Wei, S., Lee, C., Wichers, L., and Marron, J. (2016). Direction-projection-permutation for high-dimensional hypothesis tests. *Journal of Computational and Graphical Statistics*, 25(2):549–569.

[144] Yan, Y. (2016). *rBayesianOptimization: Bayesian Optimization of Hyperparameters.* R package version 1.1.0.

[145] Zhang, J. and Chen, H. (2017). Graph-based two-sample tests for discrete data. *arXiv preprint arXiv:1711.04349.*

Appendix A

Proofs of theoretical results in Chapter 2

The proofs of Lemma 1 and Theorem 1 are provided in this Appendix.

Lemma 1. *Let* $\mathbf{X}$ *and* $\mathbf{Y}$ *be independent d-dimensional random vectors with continuous distribution functions* F_X *and* F_Y*, respectively. Assume that the corresponding density functions* f_X *and* f_Y *satisfy*

(A1) $\int_{\mathbb{R}^d} f_X^2(\mathbf{u})\, d\mathbf{u} < \infty$ *and* $\int_{\mathbb{R}^d} f_Y^2(\mathbf{v})\, d\mathbf{v} < \infty$.

(A2) The function $m(\mathbf{v}) = \int_{\mathbb{R}^d} f_Y(\mathbf{u}+\mathbf{v}) f_X(\mathbf{u})\, d(\mathbf{u})$ *has a Lebesgue point in* $\mathbf{0}$.

Let $D : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ *be a nonnegative continuous function such that*

(A3) $D(\mathbf{u},\mathbf{v}) = 0$ *if and only if* $\mathbf{u} = \mathbf{v}$.

(A4) $D(a\mathbf{u}+\mathbf{w}, a\mathbf{v}+\mathbf{w}) = |a|\, D(\mathbf{u},\mathbf{v})$*, for all* $a \in \mathbb{R}$ *and* $\mathbf{w} \in \mathbb{R}^d$.

Then, if $\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3$ *are iid random vectors with distribution* F_X*,* $\mathbf{Y}_1, \mathbf{Y}_2, \mathbf{Y}_3$ *are iid random vectors with distribution* F_Y*, and the* $\mathbf{X}$*'s and* $\mathbf{Y}$*'s are independent, it holds that*

$$F_X = F_Y \quad \text{if and only if} \quad F_{X_1Y_1} = \frac{1}{2}\left(F_{X_2X_3} + F_{Y_2Y_3}\right),$$

where $F_{X_1Y_1}$*,* $F_{X_2X_3}$ *and* $F_{Y_2Y_3}$ *denote the distribution functions of the univariate random variables* $D(\mathbf{X}_1,\mathbf{Y}_1)$*,* $D(\mathbf{X}_2,\mathbf{X}_3)$*, and* $D(\mathbf{Y}_2,\mathbf{Y}_3)$*, respectively.*

Proof. Under assumptions (A1)-(A4), Theorem 2 in [85] establishes that

$$F_X = F_Y \Leftrightarrow F_{X_1Y_1} = F_{X_2X_3} = F_{Y_2Y_3}, \tag{A.1}$$

which directly leads to state that $F_X = F_Y \Rightarrow F_{X_1Y_1} = \frac{1}{2}\left(F_{X_2X_3} + F_{Y_2Y_3}\right)$. Thus we focus on proving the converse.

According to Lemma 1 in [85], under (A1)-(A4), it holds that

$$\lim_{t\to 0} \frac{P\left(D\left(\mathbf{X},\mathbf{Y}\right) < t\right)}{t^d} = \alpha \int_{\mathbb{R}^d} f_X\left(\mathbf{u}\right) f_Y\left(\mathbf{u}\right) d\left(\mathbf{u}\right), \tag{A.2}$$

where $\alpha = \int_{\{D(\mathbf{v},\mathbf{0})<1\}} d\left(\mathbf{v}\right)$.

From $F_{X_1Y_1} = \frac{1}{2}\left(F_{X_2X_3} + F_{Y_2Y_3}\right)$ follows that

$$\begin{aligned} \lim_{t\to 0} \frac{P\left(D\left(\mathbf{X}_1,\mathbf{Y}_1\right) < t\right)}{t^d} &= \\ &= \frac{1}{2}\left(\lim_{t\to 0} \frac{P\left(D\left(\mathbf{X}_2,\mathbf{X}_3\right) < t\right)}{t^d} + \lim_{t\to 0} \frac{P\left(D\left(\mathbf{Y}_2,\mathbf{Y}_3\right) < t\right)}{t^d}\right), \end{aligned}$$

for any $t \geq 0$. Now, (A.2) leads to

$$\alpha \int f_X^2\left(\mathbf{u}\right) d\mathbf{u} + \alpha \int f_Y^2\left(\mathbf{u}\right) d\mathbf{u} = 2\alpha \int f_X\left(\mathbf{u}\right) f_Y\left(\mathbf{u}\right) d\mathbf{u}$$

Therefore, since $0 < \alpha < \infty$, it follows that

$$\int \left(f_X\left(\mathbf{u}\right) - f_Y\left(\mathbf{u}\right)\right)^2 d\mathbf{u} = 0.$$

Thus concluding that f_X and f_Y are equal almost everywhere. □

As mentioned in Chapter 2, conditions (A1)-(A4) are mild regularity conditions on the underlying probability densities f_X and f_Y and the distance $D(\cdot,\cdot)$, which allow us to state that this lemma holds for a wider class of situations. most complex requirement concerns to the function

Theorem 1 establishes the consistency under the null hypothesis of the proposed statistic based on the samples of interpoint distances. The key point in the proof of Theorem 1 is that the centered empirical cumulative distribution function of the interpoint distances is asymptotically distributed as a Gaussian process [118]. It is a well known result that if $\mathbf{Z}_1,\ldots,\mathbf{Z}_r$ are i.i.d. observations from a variable with distribution F, then the empirical process $r^{1/2}\left(\widehat{F}_r(u) - F(u)\right)$ can be almost surely represented by a Gaussian process of zero mean and covariance function given by $C(u,v) = F(u \wedge v) - F(u)F(v)$, with $u \wedge v$ denoting the minimum of u and v. Nevertheless, the central limit theorem established by Silverman [118] focuses on exchangeable dissociated random variables. Consider an m-tuple J of m ordered natural numbers $(j_1,\ldots,j_m)$ and denote by $\mathcal{P}(m)$ a collection of m-tuples. A set of random variables $\{Z_J, J \in \mathcal{P}(m)\}$

is said to be *dissociated* if $\{Z_k, k \in \mathcal{I} \subset \mathcal{P}(m)\}$ is independent of $\{Z_{k'}, k' \in \mathcal{H} \subset \mathcal{P}(m)\}$ whenever $\mathcal{I} \cap \mathcal{H} = \emptyset$. In addition, a dissociated set of variables $\{Z_J, J \in \mathcal{P}(m)\}$ is said to be *exchangeably dissociated* set of variables if and only if for any finite subsequences of m-tuples, $J = (j_1, \ldots, j_m), \ldots, K = (k_1, \ldots, k_m)$, the random vectors $(\mathbf{Z}_J, \ldots, \mathbf{Z}_K)$ and $\left(\mathbf{Z}_{\pi(J)}, \ldots, \mathbf{Z}_{\pi(K)}\right)$ have the same distribution for any permutation π. In particular, a set of distances $D(\mathbf{Z}_i, \mathbf{Z}_j)$ between pairs of i.i.d. observations are a specific example of a family of exchangeable dissociated variables, and for this reason the central limit theorem proved by Silverman [118] applies in our case.

Theorem 1. *Let $\mathbf{X}_1, \ldots, \mathbf{X}_n$ and $\mathbf{Y}_1, \ldots, \mathbf{Y}_m$ be independent random samples of d-dimensional random vectors $\mathbf{X}$ and $\mathbf{Y}$ with respective continuous distribution functions F_X and F_Y. Let $D : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ be a symmetric function having bounded support $[0, D_{Max}]$. If conditions (A1)-(A4) in Lemma 1 hold and H_0: $F_X = F_Y$ $(= F)$ is true, then, as $\min(n,m) \to \infty$, the statistic $T_{n,m}$ converges weakly to the distribution of*

$$T = \int \left(\mathcal{G}_{\mathcal{W}}(u) - \mathcal{G}_{\mathcal{B}}(u)\right)^2 dF_{\mathcal{D}}(u), \tag{2.5}$$

where $F_{\mathcal{D}}$ denotes the common distribution of the interpoint distances within ($F_{\mathcal{W}}$) and between samples ($F_{\mathcal{B}}$), and $\mathcal{G}_{\mathcal{W}}$ and $\mathcal{G}_{\mathcal{B}}$ are Gaussian processes with zero mean and covariance function $C_{\mathcal{D}}(\cdot,\cdot)$ given by

$$C_{\mathcal{D}}(u,v) = 4\left[P\left(D(\mathbf{Z}_i, \mathbf{Z}_j) \le u, D(\mathbf{Z}_s, \mathbf{Z}_t) \le v\right) - F_{\mathcal{D}}(u) F_{\mathcal{D}}(v)\right],$$

where $\mathbf{Z}_i$, $\mathbf{Z}_j$, $\mathbf{Z}_s$ and $\mathbf{Z}_t$ are distributed according to F, for pairs of indexes (i,j) and (s,t) having only one element in common.

Proof. According to the definition of the test statistic in (2.3), we have

$$T_{n,m} = \int S^2_{n,m}(u)\, d\hat{H}_{N+M}(u), \tag{A.3}$$

with

$$S_{n,m}(u) = \left(\frac{NM}{N+M}\right)^{1/2} \left(\widehat{F}_{N,\mathcal{W}}(u) - \widehat{F}_{M,\mathcal{B}}(u)\right), \text{ for } u \in \mathbb{R}. \tag{A.4}$$

Under the null hypothesis H_0, Lemma 1 leads to $F_{\mathcal{W}} = F_{\mathcal{B}}$ $(= F_{\mathcal{D}})$. Thus, by adding and subtracting $F_{\mathcal{D}}(u)$ in the term in brackets in (A.4), we have

$$S_{n,m}(u) = \left(\frac{M}{N+M}\right)^{1/2} S^{\mathcal{W}}_{n,m}(u) - \left(\frac{N}{N+M}\right)^{1/2} S^{\mathcal{B}}_{n,m}(u) \tag{A.5}$$

where $S^{\mathcal{W}}_{n,m}(u) = N^{1/2}\left(\widehat{F}_{N,\mathcal{W}}(u) - F_{\mathcal{D}}(u)\right)$ and $S^{\mathcal{B}}_{n,m}(u) = M^{1/2}\left(\widehat{F}_{M,\mathcal{B}}(u) - F_{\mathcal{D}}(u)\right)$, respectively.

The family of indicator functions $\mathcal{F} = \{I\left(D(\mathbf{Z}_i,\mathbf{Z}_j) \leq u\right), u \in [0, D_{Max}] \subset \mathbb{R}\}$ is a Vapnik-Červonenkis set of measurable symmetric functions. Then, following [118], it can be concluded that the empirical processes $S^{\mathcal{W}}_{n,m}(u)$ and $S^{\mathcal{B}}_{n,m}(u)$ converge to the respective zero mean Gaussian processes $\mathcal{G}_{\mathcal{W}}$ and $\mathcal{G}_{\mathcal{B}}$, with covariance functions $C_{\mathcal{W}}(\cdot,\cdot)$ and $C_{\mathcal{B}}(\cdot,\cdot)$ given by

$$C_{\mathcal{W}}(u,v) = 4\left[P\left(\omega_{ij} \leq u, \omega_{st} \leq v\right) - F_{\mathcal{D}}(u)F_{\mathcal{D}}(v)\right],$$

for arbitrary pairs of indexes (i,j) and (s,t) having only one element in common, and $C_{\mathcal{B}}(u,v)$ analogously defined for the variables b_{ij} in $\mathcal{B}$.

This way, $S_{n,m}(u)$ converges in distribution to $\mathcal{G}_{\mathcal{W}}(u) - \mathcal{G}_{\mathcal{B}}(u)$, and then the convergence established in (2.5) follows from the Lemma of page 424 in Kiefer [71]. □

The requirement of compact support $[0, D_{Max}]$ for the distribution $F_{\mathcal{D}}$ ensures that the conditions of Theorem B in [118] to establish the weak convergence to a Gaussian process in the Skorohod topology are satisfied on the metric space $D[0, D_{Max}]$ of right-continuous functions and having limits to the left. In the case of non-compact support for $F_{\mathcal{D}}$, Silverman claims that a suitable metric space depending on the distance must be chosen to ensure measurability and attain the convergence in a weaker topology than the uniform one (see comments on page 819 in [118]).

Appendix B

Additional results for distributed classification

The same numerical analysis performed in Section 6.3.3 with the accuracy values has been carried out for recall and precision values, two alternative performance measures. The attained results are shown in this Appendix. In the case of binary classification, recall measures the effectiveness of a classifier to identify correctly classified positive instances (sensitivity), while precision evaluates the class agreement of the instance labels with the positive labels given by the classifier. One possible way to extend these concepts to the multi-class classification task is to obtain the averages of these measures calculated over all the classes $\{C_1, \ldots, C_m\}$. This generalization approach is known by *macro-averaging* [122]. This way, we have

$$\begin{aligned} Precision &= \frac{1}{m}\sum_{i=1}^{m}\frac{tp_i}{tp_i + fp_i}, \\ Recall &= \frac{1}{m}\sum_{i=1}^{m}\frac{tp_i}{tp_i + fn_i}, \end{aligned}$$

with m the number of classes in the dataset, and tp_i denoting the number of true positive for C_i, and fp_i and fn_i the false positive and false negative counts, respectively.

The results attained for these alternative criteria are displayed below, using the same scheme of tables and figures as in Section 6.3.3 for accuracy. It can be seen that very similar results are also obtained, thus supporting the main conclusions of our work.

Table B.1 Average recall values conditional on classifier type.

	Classifier				
Model	RF	SVM	XGB	LDA	Mult
BALANCED					
pNW	0.7154	0.5912	0.6991	0.5788	0.6128
pIW	0.7240	0.5968	0.7079	0.5846	0.6211
UW	0.7197	0.5935	0.7031	0.5788	0.6109
UNBALANCED					
pNW	0.7029	0.5890	0.6926	0.5777	0.6093
pIW	0.7154	0.6008	0.7063	0.5915	0.6247
UW	0.7094	0.5869	0.6998	0.5797	0.6078
MEAN					
pNW	0.7091	0.5901	0.6959	0.5783	0.6110
pIW	0.7197	0.5988	0.7071	0.5880	0.6229
UW	0.7146	0.5902	0.7015	0.5792	0.6094
ND	0.7676	0.6794	0.7477	0.6418	0.6515

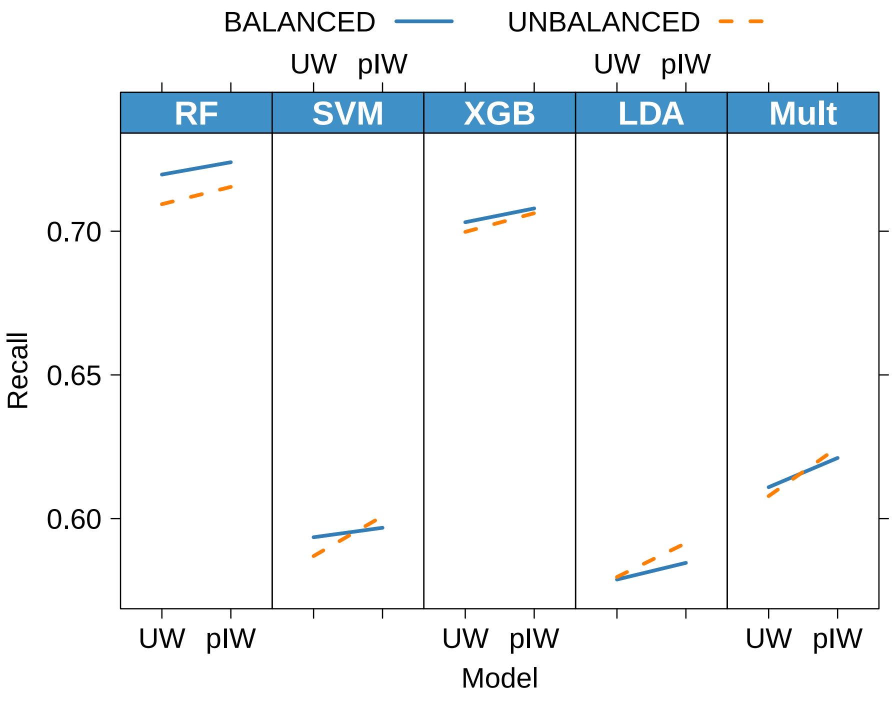


Fig. B.1 Recall-based interaction plot to check the joint effect of classifier, learning model and scenario.

Table B.2 Average recall values conditional on the decision rules.

	Decision rule					
Model	MAJ	MAX	MIN	PROD	SUM	Mean
BALANCED						
pNW	0.6505	0.6291	0.6237	0.6401	0.6539	0.6395
pIW	0.6633	0.6506	0.6119	0.6399	0.6687	0.6469
UW	0.6507	0.6323	0.6262	0.6402	0.6567	0.6412
Mean	0.6548	0.6374	0.6206	0.6401	0.6598	0.6425
UNBALANCED						
pNW	0.6493	0.6245	0.6060	0.6353	0.6564	0.6343
pIW	0.6733	0.6582	0.5895	0.6351	0.6826	0.6478
UW	0.6527	0.6257	0.6086	0.6353	0.6613	0.6367
Mean	0.6584	0.6362	0.6014	0.6353	0.6668	0.6396
ND	—	—	—	—	—	0.6976

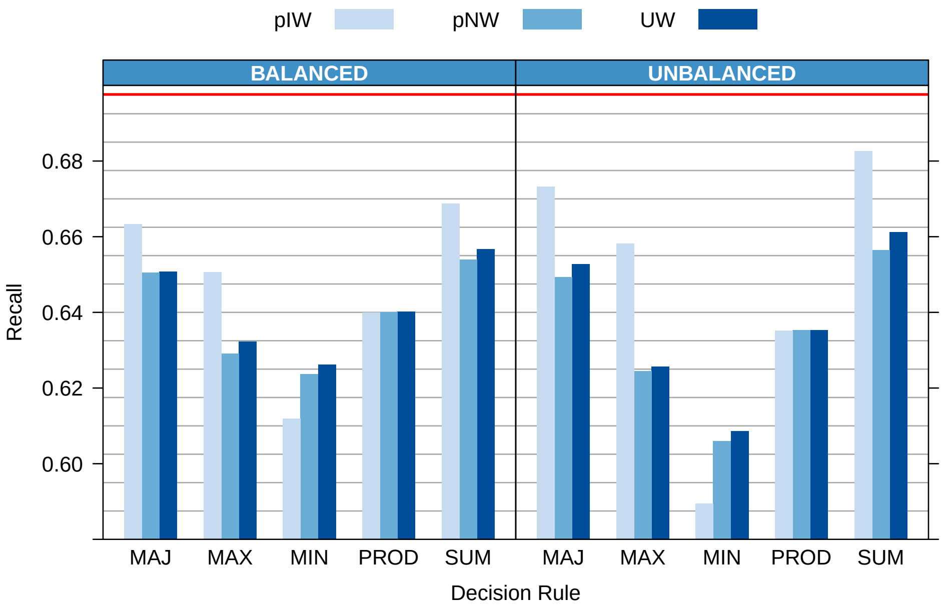


Fig. B.2 Average recall values aggregated by decision rules. The horizontal red line indicates the average recall for the ND model.

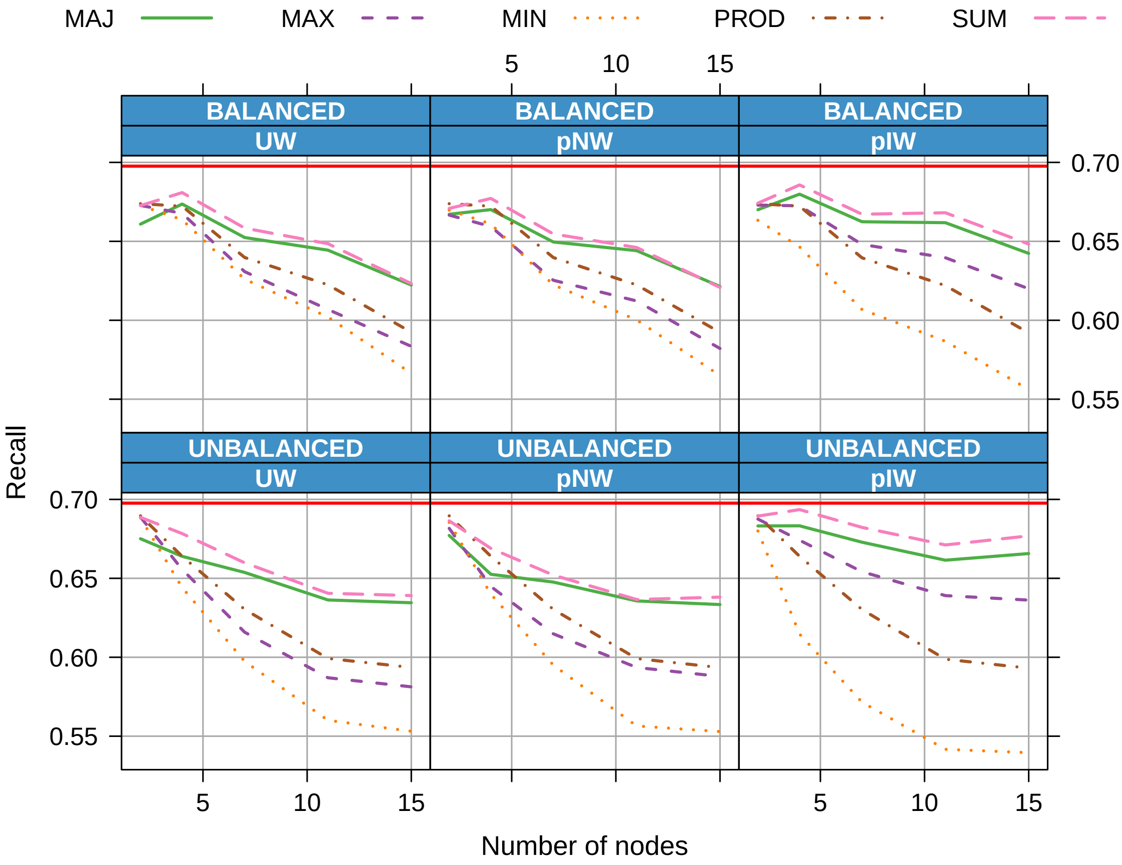


Fig. B.3 Average recall as function of the partition size. The horizontal red lines indicate the average recall for the ND model.

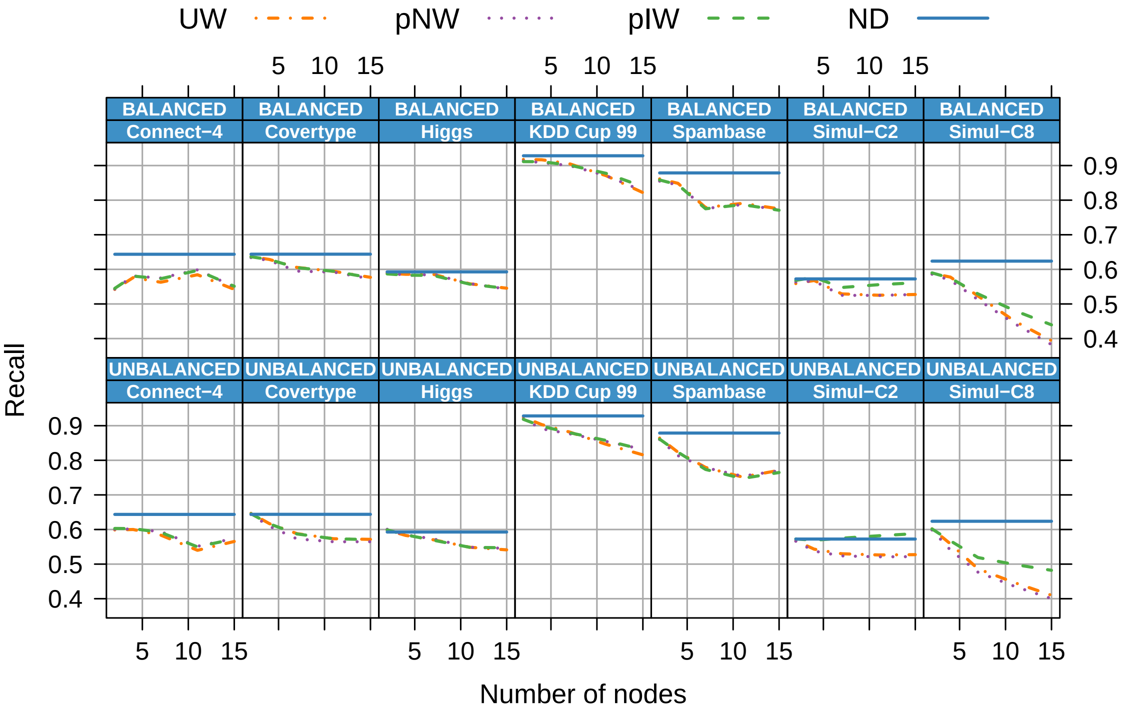


Fig. B.4 Average recall as function of the partition size for each data set. The horizontal blue lines indicate the average recall for the ND model.

Table B.3 Average precision values conditional on classifier type.

	Classifier				
Model	RF	SVM	XGB	LDA	Mult
BALANCED					
pNW	0.7016	0.5631	0.6896	0.5833	0.6118
pIW	0.7113	0.5723	0.6989	0.5895	0.6200
UW	0.7060	0.5665	0.6935	0.5818	0.6103
UNBALANCED					
pNW	0.6875	0.5577	0.6827	0.5815	0.6098
pIW	0.7001	0.5760	0.6956	0.5948	0.6249
UW	0.6929	0.5569	0.6879	0.5819	0.6090
MEAN					
pNW	0.6945	0.5604	0.6862	0.5824	0.6108
pIW	0.7057	0.5741	0.6973	0.5922	0.6224
UW	0.6994	0.5617	0.6907	0.5819	0.6097
ND	0.7516	0.6586	0.7392	0.6405	0.6496

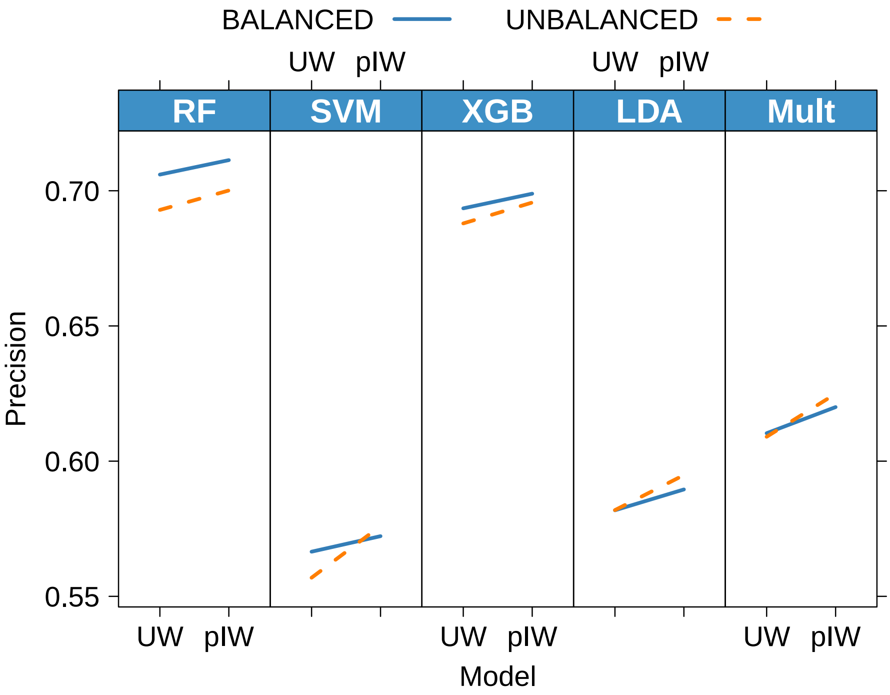


Fig. B.5 Precision-based interaction plot to check the joint effect of classifier, learning model and scenario.

Table B.4 Average precision values conditional on the decision rules.

	Decision rule					
Model	MAJ	MAX	MIN	PROD	SUM	Mean
BALANCED						
pNW	0.6392	0.6239	0.6145	0.6282	0.6436	0.6299
pIW	0.6550	0.6459	0.6030	0.6280	0.6601	0.6384
UW	0.6408	0.6266	0.6166	0.6283	0.6459	0.6316
Mean	0.6450	0.6321	0.6113	0.6282	0.6499	0.6333
UNBALANCED						
pNW	0.6378	0.6193	0.5945	0.6226	0.6451	0.6238
pIW	0.6637	0.6531	0.5799	0.6224	0.6724	0.6383
UW	0.6420	0.6178	0.5985	0.6226	0.6477	0.6257
Mean	0.6478	0.6300	0.5910	0.6225	0.6551	0.6293
ND	—	—	—	—	—	0.6879

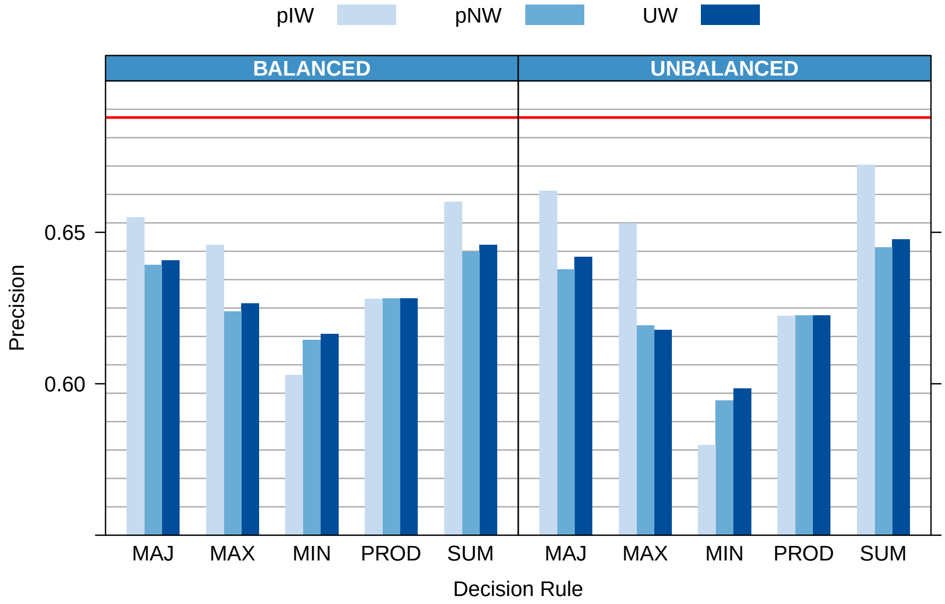


Fig. B.6 Average precision values aggregated by decision rules. The horizontal red line indicates the average precision for the ND model.

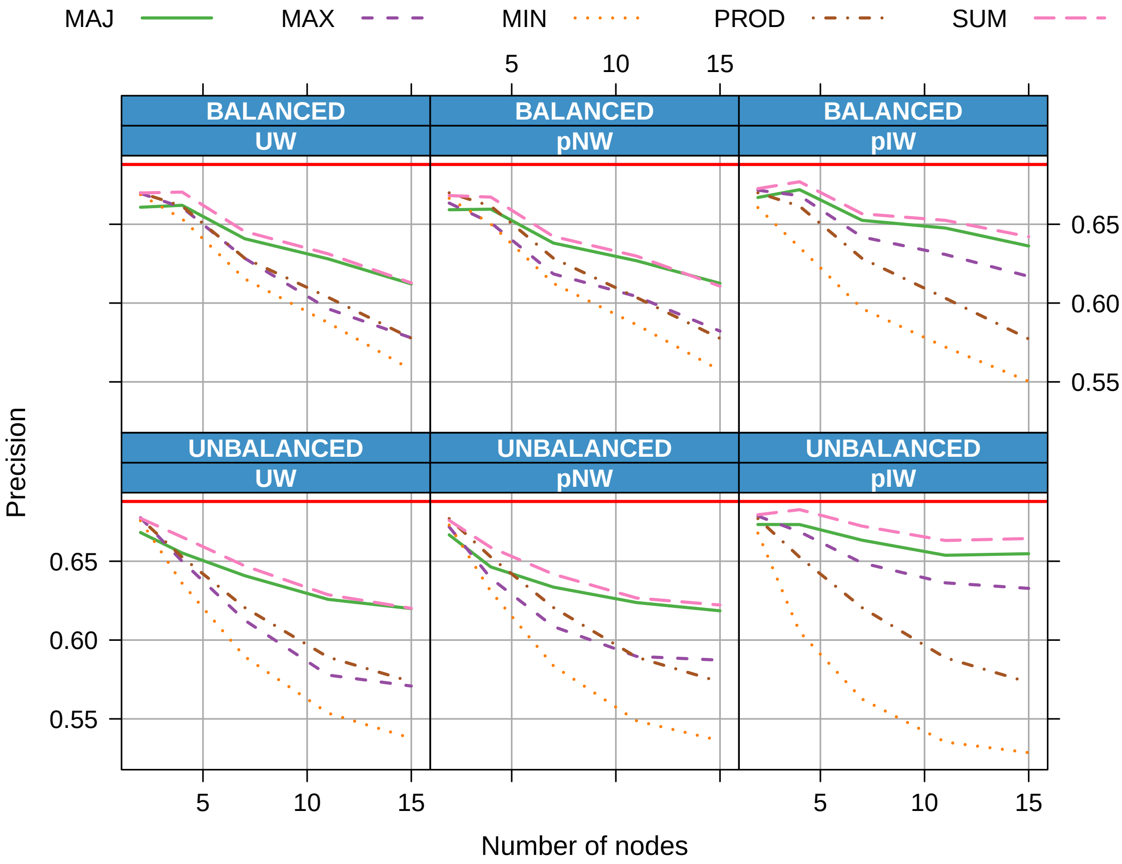


Fig. B.7 Average precision as function of the partition size. The horizontal red lines indicate the average precision for the ND model.

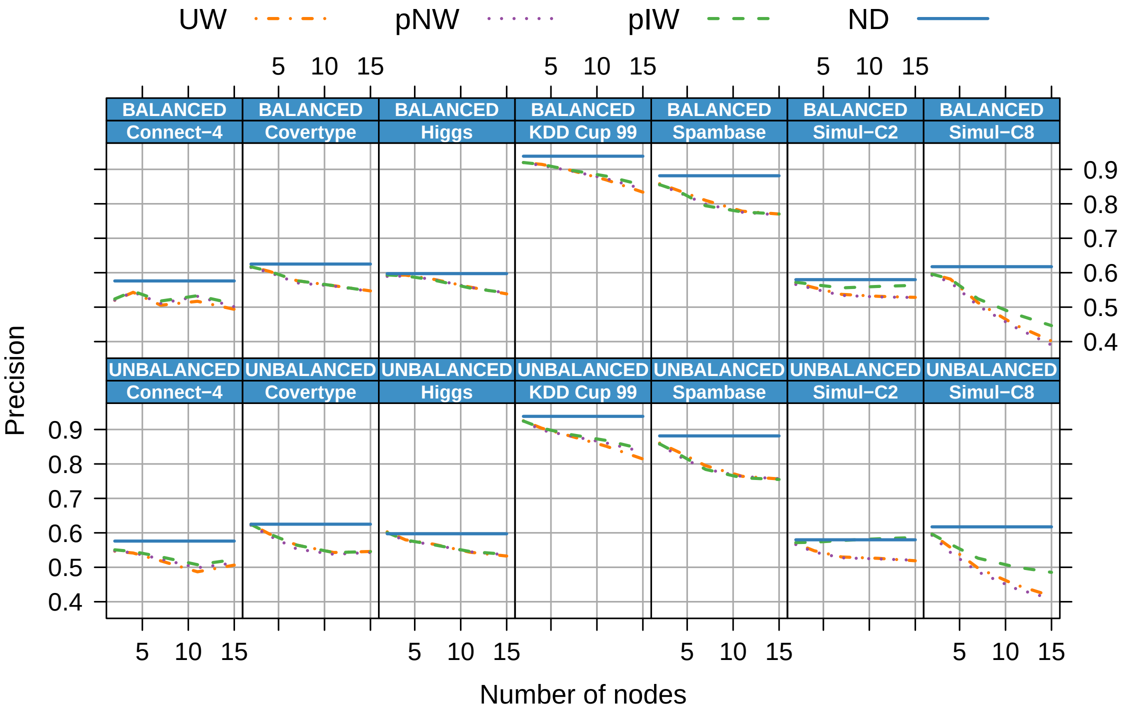


Fig. B.8 Average precision as function of the partition size for each data set. The horizontal blue lines indicate the average precision for the ND model.

Appendix C

Resumen extendido en castellano

Esta tesis trata sobre aprendizaje estadístico, tanto en objetos complejos, como pueden ser series temporales o formas, entre otros. Las tareas que se estudian son contrastes de dos muestras, clasificación, clustering y predicción.

Como objetos complejos entendemos los tipos de datos sobre los que al aplicar técnicas estadísticas generales de una manera directa, sin ningún tipo de preprocesado, se obtienen resultados no satisfactorios.

La manera de abordar datos complejos es mediante la introducción de conocimiento del dominio del fenómeno subyacente que da lugar a los datos complejos al ser medido. Un ejemplo de conocimiento del dominio puede ser, en el campo de series temporales, el uso de los coeficientes de autocorrelación extraídos de las series en lugar de las series en bruto para su análisis.

Ejemplos concretos de tareas de aprendizaje de datos complejos son:

- Test de dos muestras: Detectar si hay diferencias entre los patrones de ondas cerebrales de sujetos alcohólicos y sujetos de control al presentarles un determinado estímulo visual.

- Clasificación: Asignar un hueso a una especie de animal concreto a partir de su silueta digitalizada.

- Clustering: Encontrar grupos de genes de una bacteria en función de cómo se expresan temporalmente ante una alteración, como añadir un antibiótico a su sustrato.

Nuestro objetivo es el desarrollo de técnicas para abordar este tipo de problemas.

Los dos métodos principales para introducir conocimiento del dominio son la extracción de características y el uso de distancias. Las características, también llamadas variables o atributos, son valores que se calculan a partir de los datos, como pueden ser la media, varianza, o coeficientes de autocorrelación de una serie temporal. Cuando usamos extracción de características, el siguiente paso consiste en aplicar una técnica estadística estándar sobre ellas, en lugar de sobre los datos en bruto. Debido a su simpleza y su facilidad de interpretación, el camino de las características suele ser el preferido para analizar datos complejos. Computacionalmente suele ser el más ventajoso. Su limitación viene dada por el hecho de que en muchas situaciones no sabemos qué características extraer de los datos y encontrarlas es un proceso que puede ser inabordable.

El otro camino para analizar datos complejos es mediante el uso de distancias. Como distancia se entiende una función que compara dos objetos, asignando un valor numérico a esta comparación, que decrece conforme más se asemeja el par de objetos comparado. Una vez se han calculado estas distancias entre pares de objetos, se aplican métodos generales de análisis basados en distancias, que trabajan sobre las comparaciones, no sobre los objetos originales.

Los métodos basados en distancias pueden usarse para solucionar la limitación principal de los métodos basados en características, esto es, existen dominios en los que podemos definir una distancia que funcione bien, pero no un conjunto de características.

Aunque los métodos basados en distancias tienen una estrecha relación con los métodos basados en características, siendo equivalentes en muchos casos, los primeros pueden verse como una extensión no paramétrica de los segundos. Podemos encontrar distancias que no tengan un conjunto finito de características equivalente.

La familia de distancias “elásticas”, de uso común en el análisis de series temporales, contiene ejemplos de distancias sin conjunto de características equivalentes e ilustran el hecho de que para analizar un tipo de datos podemos definir una distancia pero no características. Las distancias elásticas se originaron para analizar un tipo de series temporales concreto: la voz humana. La primera distancia elástica, llamada Dynamic Time Warping (DTW), es similar a la distancia Euclídea pero permite aceleraciones y deceleraciones en el tiempo de las series a comparar. Estas aceleraciones se calculan de manera que se minimice la diferencia entre las dos series una vez se han aplicado los cambios. DTW se define algorítmicamente como un problema de programación dinámica, y es fácil ver que no es posible encontrar un conjunto de características que obtenga los mismos resultados. DTW junto con el método de vecinos más próximos fue considerado el estado del arte en clasificación de series temporales hasta muy recientemente, y se ha usado con éxito para otros tipos de datos como formas.

Las distancias también se usan en otras tareas de análisis o minería de datos, como son la detección de datos atípicos o la recuperación de información o la visualización de datos. Por lo tanto, al definir una nueva distancia podemos contribuir a mejorar las herramientas de análisis de múltiples tareas y dominios de aplicación.

Métodos basados en distancias

Existe una relación bidireccional entre las distancias y los métodos que las utilizan: Queremos definir nuevas distancias que puedan ser usadas por los métodos existentes, pero también nos interesa definir nuevos métodos que puedan incorporar un amplio rango de distancias.

Por ejemplo, el algoritmo de clustering K-medias está definido para la distancia Euclídea. Si intentamos emplear K-medias con otra distancia de una manera directa, es posible que alcancemos resultados irrelevantes o incluso engañosos. Para poder usar K-medias con una distancia arbitraria, tenemos que proporcionar una manera de calcular los centros de los grupos con respecto a la distancia, que en general no van a ser la media aritmética. El cálculo de centros para una distancia arbitraria es un problema abierto, y la creación de métodos para obtener centros para distancias populares como DTW es un problema sobre el que se sigue investigando. Por el contrario, otros métodos de clustering como cluster jerárquico o particionado alrededor de medoides (PAM) no necesitan calcular los centros de los grupos, y por lo tanto el abanico de distancias con los que pueden ser usados es mucho mayor que K-medias.

Es por lo tanto muy deseable que los métodos basados en distancias funcionen con una gran abanico de ellas.

Contribuciones

Las contribuciones de esta tesis tienen como hilo conductor dos ideas que además están interrelacionadas:

- Las distancias son una manera muy útil para introducir conocimiento del dominio ante problemas complejos. Es interesante definir nuevas distancias como nuevos métodos que funcionen bien con una gran variedad de distancias.

- Algunos estadísticos (basados en distancias) para contrastes de dos muestras pueden ser a su vez considerados como distancias entre conjuntos de datos multivariantes o funciones de distribución empíricas. Utilizamos esta idea para comparar series temporales o conjuntos de datos.

En el Capítulo 2 se propone un test de dos muestras multivariante basado en distancias. La idea detrás del estadístico es la de comparar las distribuciones de las distancias entrepuntos. Dadas dos muestras de observaciones multivariantes, por ejemplo X e Y y una distancia, se calculan todos los posibles pares de distancias entrepunto. Estas distancias se dividen en dos grupos, las distancias de pares pertenece a la misma muestra, llamadas “intramuestra” esto es, las distancias entre dos observaciones de X y las distancias entre observaciones de Y, y el segundo grupo lo forman las distancias entre muestras, llamadas “entremuestra”. Las distancias entremuestra salen de pares de observaciones en los que un miembro del par pertence al grupo X y otro al grupo Y. Una vez separados estos dos grupos, las distancias pasan a ser consideradas como observaciones univariantes normales, y las distribuciones de los grupos intramuestra y entremuestra se comparan mediante un estadístico tipo Cramér-von Mises.

El estadístico propuesto, llamado DD debido a que compara distribuciones de distancias, pertenece a un grupo más grande de estadísticos de tests de dos muestras basados en distancias entrepuntos. Dentro de esta familia destaca el estadístico llamado “energy”. El estadístico energy funciona de una manera similar a DD, pero compara únicamente las medias de las distancias intragrupo y entregrupo, en lugar de las distribuciones. Aunque es un estadístico consistente y sencillo tanto conceptual como computacionalmente, su principal limitación es su falta de potencia ante algún tipo de alternativa. Por ejemplo, ante cambios de escala o correlación en un escenario de Gaussianas multivariantes, suele obtener peores resultados que sus competidores. Otros tests de dos muestras basados en distancias, por ejemplo todos los basados en el grafo que crea la matriz de distancias dos a dos, tienen limitaciones similares: o bien son potentes ante cambios de escala o antes cambios de localización, pero no ante ambos. La capacidad de detectar diferencias distribucionales de varios tipos es uno de los objetivos que persigue el estadístico DD.

Para evaluar el funcionamiento del estadístico se realizaron experimentos de tanto de simulación como de datos reales. Todos los escenarios estudiados han sido propuestos con anterioridad en la literatura de tests de dos muestras. En concreto se estudia la evolución de la potencia al aumentar la dimensión de Gaussianas multivariantes, considerando cambios de escala y localización. En este escenario, se comprueba que todos los métodos estudiados, incluyendo energy y estadísticos basados en grafos, son potentes ante cambios de escala o localización, pero no ante ambos simultáneamente. La única excepción es el estadístico DD, que se sitúa entre los más potentes en ambos escenarios, si bien no es el que mejor en ninguno. En los escenarios de datos reales, se observa que no existe un método superior a otros, todos los métodos obtienen baja potencia en algún escenario. Como conclusión general del estudio empírico, el estadístico DD obtiene resultados que podrían ser calificados de robustos o flexibles, alcanzando niveles de potencia razonables en la mayoría de situaciones, sin encontrar casos problemáticos particulares donde no obtenga potencia estadística pero sus competidores sí.

Capítulo 3

En el Capítulo 3 se estudia en mayor profundidad el funcionamiento de los tests de dos muestras en el contexto de series temporales, incluyendo el estadístico DD. Además se plantea verificar si las distancias diseñadas para clasificación y clustering de series temporales proporcionarían un incremento de potencia al ser introducidas en los tests basados en distancias. En series temporales nos encontramos con dificultades que aparecen comúnmente en estadística de alta dimensión, como que la dimensión de los datos es superior al tamaño muestral, pero también aparecen dificultades propias del ámbito temporal. Entre éstas está la complejidad debido a la dependencia, la necesidad de analizar series temporales de distinta longitud, o la necesidad de "no considerar" maneras en las que series puede diferir. Como ejemplo del último caso, es posible que a los investigadores no les interese detectar si dos grupos de series temporales difieren en cuanto a escala, porque en su dominio de aplicación la interpretación de la escala es irrelevante. En el caso de ondas de sonido medidas directamene la escala sería el volumen, y es posible que los investigadores deseen detectar si hay diferencias entre dos grupos sin tener en cuenta el volumen. La propiedad de un método de ignorar cambios de un determinado tipo recibe el nombre de invarianza, y las distancias son una buena manera de introducir invarianzas complejas.

Se plantean escenarios de simulación de modelos autorregresivos clásicos, incluyendo modelos heterocedásticos, y de datos reales como electroencefalogramas o consumo eléctrico. Se observa que las versiones estándar de los tests de dos muestras basados en distancias, que usan la distancia Euclídea, obtienen baja potencia en muchos escenarios, y el estadístico DD no alcanza los mejores resultados en la comparativa. Sin embargo, al introducir distancias específicas de series temporales, la potencia de los tests aumenta, llegado incluso a potencia 1 cuando con la Euclídea solo se alcanzaban los valores nominales. Es también destacable que al introducir estas nuevas distancias, el estadístico DD se compara de manera favorable al resto de métodos, debido a su flexibilidad: mediante la introducción de una distancia, por ejemplo un escenario que planteaba una diferencia de distribución en escala, podría pasar a una diferencia en distribución en localización o a una combinación de ambas. Como es posible que haya distancias que discriminen muy bien los dos grupos, pero de una manera no conocida (por ejemplo sabiendo de antemano que generará una diferencia en localización), es deseable que es estadístico empleado se capaz de detectar una gran variedad de alternativas, no centrarse demasiado en alguna en concreto.

Capítulo 4

Como se ha visto con anterioridad, las distancias son un buen vehículo para introducir conocimiento del dominio que mejore los análisis estadísticos. En este capítulo se propone una nueva distancia para series temporales principalmente para ser usada en problemas de clasificación y clustering. Se identifica que métodos con muy buen funcionamiento empírico en clasificación, conocidos como "bolsa de patrones" (Bag of patterns) y otros como distancias basadas en modelos, usadas principalmente en clustering, tienen algunos puntos en común. En concreto, se basan en comparar las distribuciones lageadas de las series temporales, introduciendo la suposición de que una serie puede identificarse por su comportamiento al observarse conjuntamente observaciones próximas en el tiempo. Tanto los métodos Bag of Patterns como las distancias basadas en modelos aplican, sin embargo, simplificaciones muy fuertes para comparar las distribuciones, los primeros cuantizan los datos para pasar de distribuciones multivariantes a histogramas univariantes, y los segundos asumen modelos restrictivos como los lineales. Para comparar las distribuciones de una manera más parsimoniosa, se introduce la idea de usar un estadístico para test de dos muestras como distancia entre distribuciones, en concreto se usa el estadístico energy por su simpleza. Mediante esta distancia, en cierto modo unificamos y extendemos las dos familias de métodos Bag of Patterns y distancias basadas en modelos. Además se propone una manera automática de seleccionar los lags relevantes, que se diferencia de otros en que se centra en discriminar las series, no en modelar cada una por separado de una manera precisa. En un amplio estudio de datos reales (incluyendo problemas con series de distinta naturaleza, como electrocardiogramas, datos de detección de movimiento o consumo eléctrico) y de simulación basada en modelos autorregresivos, comparando su funcionamiento tanto en clasificación como en clustering, la distancia propuesta alcanza mejores resultados que otras alternativas comúnmente usadas en series temporales.

Capítulo 5

En este capítulo se propone un nuevo método para hacer predicción de series temporales (forecasting), que llamamos FFORMA (Feature-based FORecast Model Averaging).

Este método funciona mediante la combinación de las predicciones de algoritmos de forecast individuales. La combinación de métodos obtiene buenos resultados empíricos y puede verse como una manera de protegerse ante el riesgo de seleccionar un modelo érroneo para la serie que queremos predecir. El método que proponemos se encuadra dentro del paradigma del meta-aprendizaje, donde un modelo se utiliza para combinar otros modelos predictivos. En lugar de aprender a seleccionar qué modelo sería el mejor para cada serie, en nuestro caso se aprende a generar pesos que los combinen.

El procedimiento FFORMA se compone de tres elementos básicos:

- Conjunto de referencia: Para ajustar el modelo de meta-aprendizaje es necesario un conjunto de series temporales generado por un procedimiento que debería ser similar al de las series sobre las que queremos hacer forecast. En el caso de necesitar predecir conjuntos de datos muy grandes, estos podrían utilizarse a su vez como conjuntos de referencia. Las series pertenecientes al conjunto de referencia se supone que tienen valores futuros conocidos, en el caso de no ser así, mediante temporal holdout se dividiría cada serie en el período de entrenamiento y período de prueba, extrayendo las últimas observaciones de la serie.

- Reserva de métodos: Son los algoritmos de forecast que van a ser combinados. Normalmente son algoritmos clásicos basados en modelos, como ARIMA o exponetial smoothing. Cada uno de estos métodos es aplicado a cada serie, ajustando sus parámetros, y produce la predicción deseada que luego será combinada con la de los otros métodos.

- Conjunto de características: FFORMA usa características extraídas de las series, en lugar de las series en bruto como entrada para el algoritmo de meta-aprendizaje. Muchas de características se han utilizado previamente en comparación de series, como las autocorrelaciones, fuerza de la tendencia, etc.

La idea detrás de FFORMA está relacionada con la clasificación de series temporales. Partiendo de un método de meta-aprendizaje que selecciona el método más apropiado para predecir una serie, puede verse que al asignar etiquetas a los métodos y asignando a cada serie temporal la etiqueta del método que mejor la predice, obtenemos un problema de clasificación. Sin embargo, esta aproximación tiene el inconveniente de descartar información: Por ejemplo cuando algunos de los métodos en nuestra reserva podría obtener errores muy similares al mejor, de manera que se exagera la relevancia de este último como clase objetivo. También podrían existir series más difíciles que otras (en cuanto al error que cometen todos los métodos) y esto no se tiene en cuenta si se plantea el meta-aprendizaje como un problema de clasificación. Por este motivo, lo que se hace es buscar una función que asigna pesos a los métodos de la reserva, con la sencilla interpretación de minimizar el error de predicción que se produciría si los métodos fuesen asignados al azar a cada serie siguiendo estos pesos.

FFORMA funciona en dos fases:

1. Fase Offline: Entrenar el modelo de meta-aprendizaje. Las series en el conjunto de referencia se dividen en período de prueba y período de aprendizaje, los métodos de la reserva generan sus predicciones y se mide el error de las predicciones sobre el período de prueba. Usando este error y las características, el modelo de meta-aprendizaje se ajusta para generar los pesos que asignados a las predicciones individuales minimizarían el error de predicción.

2. Fase Online: Usar el modelo entrenado en la fase 1 para generar pesos y combinar las predicciones de los métodos en la reserva usando estos pesos para producir la predicción de una nueva serie temporal.

FFORMA se usó para la competición internacional de forecasting M4, sobre un conjunto de 100000 series temporales, alcanzando el segundo mejor resultado en cuanto a error de predicción.

Capítulo 6

En el Capítulo 6 se presenta un método para clasificación distribuida. En el contexto que trabajamos, los datos están repartidos en distintos nodos de cómputo y no se permite que los nodos compartan sus datos. El objetivo es combinar los modelos de clasificación ajustados con los datos de cada nodo de manera que se minimice la pérdida de exactidud de clasificación relativa a si se pudiese ajustar un modelo sobre la totalidad de los datos, al unir los de todos los nodos (llamado conjunto global). Esta restricción de no comunicación aparece en escenarios realistas por motivos éticos o de negocio, como hospitales que no pueden compartir registros de sus pacientes, y también por motivos computacionales, por ejemplo el conjunto al completo sería demasiado grande para ser procesado en un único nodo, o la comunicación es muy costosa.

Se plantea un marco probabilista que permite analizar el mencionado contexto distribuido, llegando a un resultado que relaciona el clasificador que saldría del conjunto global con una suma ponderada de los clasificadores entrenados en cada uno de los nodos. Estos pesos dependen de un cociente de densidades que deben ser estimadas, pero debido a la dificultad de esta estimación, proponemos estimar los pesos directamente. Pare ello se emplea una heurística basada en considerar una distancia entre la distribución de los datos de un nodo y la de los datos del conjunto de prueba. La distancia considerada es el estadístico energy, similar al Capítulo 4. Los pesos calculados por la heurística minimizan esta distancia energy entre distribución de nodo y prueba. Una interpretación del procedimiento sería que al hacer un remuestreo ponderado de la muestra del conjunto de prueba, la distribución de esta remuestra se asemejaría a la del nodo en cuanto a que se minimiza la distancia energy entre ambas. Las observaciones del conjunto de prueba mejor representadas en un nodo (con mayor densidad en su entorno), reciben mayor peso.

Este método consigue mejorar la aproximación de referencia de una manera clara.