Feature selection with limited bit depth mutual information for portable embedded systems

Laura Morán-Fernández[a,*], Konstantinos Sechidis[b], Verónica Bolón-Canedo[a], Amparo Alonso-Betanzos[a], Gavin Brown[b]

[a] *CITIC, Universidade da Coruña, A Coruña, Spain*
[b] *School of Computer Science, University of Manchester, Manchester, UK*

Abstract

Since wearable computing systems have grown in importance in the last years, there is an increased interest in implementing machine learning algorithms with reduced precision parameters/computations. Not only learning, also feature selection, most of the times a mandatory preprocessing step in machine learning, is often constrained by the available computational resources. This work considers mutual information —one of the most common measures of dependence used in feature selection algorithms— with a limited number of bits. In order to test the procedure designed, we have implemented it in several well-known feature selection algorithms. Experimental results over several synthetic and real datasets demonstrate that low bit representations are sufficient to achieve performances close to that of double precision parameters and thus open the door for the use of feature selection in embedded platforms that minimize the energy consumption and carbon emissions.

Keywords: reduced precision, mutual information, feature selection, portable embedded systems, internet of things, edge computing

[*] Corresponding author.

Email addresses: laura.moranf@udc.es (Laura Morán-Fernández), konstantinos.sechidis@manchester.ac.uk (Konstantinos Sechidis), vbolon@udc.es (Verónica Bolón-Canedo), ciamparo@udc.es (Amparo Alonso-Betanzos), gavin.brown@manchester.ac.uk (Gavin Brown)

1. Introduction

With the advent and standardization of wireless connectivity paradigms and the cost reduction of electronic components, the number and diversity of Internet of Things (IoT) devices has exploded over the last decade (Ray et al., 2016). Wearable computing has made successful and significant forays in fitness domains, health care, fashion and entertainment, among other application areas. These devices are usually employed as local systems, and their fundamental requirements are to work with little computing power and small memories. However, these requirements become challenging since emerging computing devices are not just sensor devices: they must perform sophisticated computation, collect and aggregate data for propagation to the cloud, and respond in real time to user requests. This data must be fed on a machine learning (ML) system to analyze information and make decisions. Unfortunately, limitations in the computational capabilities of resource-scarce devices inhibit the implementation of the most current ML algorithms on them. Then, the data must be sent to a remote computational infrastructure. However, an interest in a different paradigm based on Edge Computing has emerged. Edge computing refers to computations being performed as close to data sources as possible, instead of remote locations.

Imagine a health wearable (Figure 1) which measures a high number of body parameters such as vital signs (electrocardiography, pulse, blood oxygen saturation, respiration, skin temperature, CO_2), body kinematics as well as sensorial, emotional and cognitive reactivity such as electrocardiography, posture, fall, movement, speed, acceleration or pressure. It is common that a large number of these features (i.e. body parameters) is not informative because they are either irrelevant or redundant with respect to a specific disease or health condition. Therefore, selecting the most relevant features could significantly improve disease prevention, diagnosis, treatment, disease management and rehabilitation, and help to discover personal patterns of interest. Feature selection arises from the need of determining the "best" subset of variables for a given problem. The use of an adequate feature selection method can avoid over-fitting and improve model performance, providing faster and more cost-effective learning models and a deeper insight into the underlying processes that generate the data (Saeys et al., 2007). Features can be categorized in three ways: relevant, irrelevant and redundant (Huang, 2015). As a result, selecting the relevant features and ignoring the irrelevant and redundant ones is advisable. The process of feature selection is typically performed on a machine using high numerical representation, i.e. double-precision floating point calculations (64 bits). Using a more powerful general purpose processor provides significant benefits in terms of speed and capability to solve more complex problems. But this capability does not come without cost; a conventional microprocessor can require a substantial amount of off-chip support hardware, memory, and often a complex operating system (Koopman, 1990). In contrast to up-to-date computers, these requirements are often not met by embedded systems, low energy computers or integrated solutions that need to optimize the used hardware resources. However, to the best of our knowledge, reduced-precision approaches have not been implemented yet in the area of feature selection. And portable embedded systems, though, call for new feature strategies and methods that are able to deal with big dimensionality.

Figure 1: Health wearable (Commons, 2001).

The majority of the existing approaches available investigated the effect of reduced precision in neural networks (Murshed et al., 2019). Han et al. (2016) presented an energy-efficient engine that performed inference on compressed deep neural networks and accelerated the resulting sparse matrix-vector multiplication with weight sharing. Hubara et al. (2017) introduced a method to train Quantized Neural Networks (QNNs), i.e. neural networks with extremely low precision weights and activations at run-time. They found that QNNs achieved prediction accuracy comparable to their 32-bit counterparts. Jacob et al. (2018) proposed a quantization scheme that relied only on integer arithmetic to approximate the floating-point computations in a neural network. The authors were inspired by the work of Gupta et al. (2015), which leverages low-precision fixed-point arithmetic to accelerate the training speed of convolutional neural networks. In the area of Bayesian networks, Tschiatschek and Pernkopf (2015) considered online learning of these classifiers with reduced precision parameters in order to facilitate their utilization in computationally constrained platforms. All above mentioned authors demonstrated that their proposed reduced-precision algorithms achieved classification performances close to that of Bayesian networks classifiers with parameters learned by traditional algorithms using double-precision floating point representation.

In this work, we investigate feature selection by considering the information theoretic measure of mutual information with reduced precision parameters described in Morán-Fernández et al. (2018). The mutual information measure is used due to its computational efficiency and simple interpretation. Therefore, we are able to provide a limited bit depth mutual information, and —through different feature selection methods based on this measure— experimentally achieve classification performances close to that of 64-bit representations for several real and synthetic datasets. Our reduced precision approach is designed to analyze user level data, i.e. on-board analysis for close-loop feedback. It performs the preprocessing step over private "small" data. Then, this anonymized data could become available in the cloud, by aggregation of personal data from different users, to obtain "big" data that can be processed by more powerful processors and/or distributed to experts for further analysis.

The remainder of this paper is organized as follows. Section 2 provides the background of mutual information in feature selection. Section 3 presents our limited bit depth mutual information approach. Section 4 provides and discusses an experimental study over several real and synthetic datasets in terms of the ranking similarity and the classification accuracy, as well as a case study to analyze the robustness against noise of our proposed method. Finally, Section 5 contains our concluding remarks and proposals for future research.

2. Mutual information in feature selection

Mutual Information (MI) comes from the field of Information Theory and it is widely used in both machine learning and statistics. One of its main uses is feature selection methods, and in fully supervised data, the features X are ranked using this measure, and the ones finally selected are those having the highest mutual information with the class label Y. The mutual information is defined as the expected logarithm of a ratio:

$$I(X;Y) = \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x,y) \ln \frac{p(x,y)}{p(x)p(y)} \tag{1}$$

where $p(x,y) = Pr\{X = x, Y = y\}$ is the probability mass function of the joint distribution when the random variable X takes on the value x from its alphabet $\mathcal{X}$ and Y takes on $y \in \mathcal{Y}$, while $p(x) = Pr\{X = x\}$ and $p(y) = Pr\{Y = y\}$ are the probability mass functions of the marginal distributions. In this work, the function is calculated in natural logarithm, so returned units are "nats". In practice we have to estimate this from data. This can be done by using the sample (maximum-likelihood) estimates of the probabilities $\hat{p}$ and plug them in the Equation 1. This maximum likelihood estimator for the mutual information is consistent (Paninski, 2003), and as a result we have:

$$I(X;Y) \approx \hat{I}(X;Y) = \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} \hat{p}(x,y) \ln \frac{\hat{p}(x,y)}{\hat{p}(x)\hat{p}(y)} \tag{2}$$

In order to calculate this we need the estimated distributions $\hat{p}(x,y), \hat{p}(x)$, and $\hat{p}(y)$. The probability of any particular event $p(X = x)$ is estimated by maximum likelihood, the frequency of occurrence of an event $X = x$ divided by the total number of events.

An example. Let us consider a vector Y with 651 observations, in which the number of occurrences of an event $Y = y$ is 3. The probability $\hat{p}(y)$ will be:

$$\hat{p}(y) = \frac{3}{651} = 0.00460829493$$

which it is approximately zero. For real applications, it is not necessary to store all the decimal digits, which makes mutual information an interesting measure to explore reduced precision. Besides, as the embedded systems market matures, we will likely see a movement away from full mutual information (i.e. 64 bit-representation) to limited approaches using a lower number of bits.

Mutual information definition is useful within the context of feature selection because it gives a way to quantify the output vector. Thus, there exist in the literature several feature selection methods based on mutual information measures (Battiti, 1994; Tesmer and Estévez, 2004; Peng et al., 2005; Guo and Nixon, 2009). Most methods define heuristic functionals to assess feature subsets combining definitions of relevant and redundant features. Among the different feature selection methods based on mutual information, we have chosen three to evaluate our limited bit depth mutual information approach: MIM (Mutual Information Maximisation) Lewis (1992) due to its simplicity, JMI (Joint Mutual Information) (Yang and Moody, 2000) and mRMR (minimum Redundancy Maximum Relevance) multivariate filter (Peng et al., 2005) since they showed the best overall trade-off for accuracy/stability (Brown et al., 2012). In any case, our reduced precision approach could be easily implemented in any other MI-based feature selection algorithms.

- MIM ranks the features by their MI score, and selects the top k features, where k is decided by some predefined need for a certain number of features or some other stopping criterion.

  $$MIM(X_k) = I(X_k; Y) \tag{3}$$

  An important limitation is that this assumes that each feature is independent of all other features and effectively ranks the features in descending order of their MI content. Thus, this approach does not take into account the redundancy between the features.

- JMI is focused on increasing complementary information between features. The JMI score for feature X_k is

  $$JMI(X_k) = \sum_{X_j \in S} I(X_k X_j; Y) \tag{4}$$

  This is the information between the targets and a joint random variable $X_k X_j$, defined by pairing the candidate X_k with each feature previously selected. The idea is if the candidate feature is "complementary" with existing features, we should include it.

- The mRMR feature selection method selects features that have the highest relevance with the target class and are also minimally redundant, i.e. it selects features that are maximally dissimilar to each other. Both optimization criteria (maximum-relevance and minimum-redundancy) are based on mutual information. Let S denote the subset of features we are seeking:

$$mRMR(X_k) = I(X_k; Y) - \sum_{X_j \in S} I(X_k X_j; Y) \tag{5}$$

The mRMR criterion, like JMI, has a strong belief in the pairwise independence assumptions as the feature set S grows.

Table 1 shows the theoretical complexity of the three methods described above (Sechidis et al., 2019). Let us assume that we have a dataset of m samples and n features and we want to select the top-k.

Table 1: Theoretical complexity of the three feature selection methods focus of this work.

Method	Complexity
MIM	$\mathcal{O}(k \cdot m \cdot n)$
JMI	$\mathcal{O}(k^2 \cdot m \cdot n)$
mRMR	$\mathcal{O}(k^2 \cdot m \cdot n)$

3. Limited bit depth mutual information

In information theoretic feature selection, the main challenge is to estimate the mutual information, one of the most common measures of dependence used in machine learning. As said above, to calculate mutual information we need to estimate the probability distributions. Internally, it counts the occurrences of values within a particular group (i.e. its frequency). Thus, based on Tschiatschek and Pernkopf (2015)'s work for approximately computing probabilities, we investigate mutual information with limited number of bits by considering this measure with reduced precision counters. To perform the reduced precision approach, we target a fixed-point representation instead of the 64-bit resolution used typically by the standard hardware platforms. Fixed-point numbers are essentially integers scaled by a constant factor, i.e. the fractional part has a fixed number of digits. We characterize fixed-point numbers by the number of integer bits bi and the number of fractional bits bf. The motivation to move to fixed-point arithmetic is two-fold. The first reason is that these bit representation compute units are typically faster and consume far less hardware resources and power than the conventional floating-point computations. And, second, low-precision data representation reduces the memory footprint, enabling larger models to fit within the given memory capacity and lowering the bandwidth requirements.

Mutual Information parameters are typically represented in the logarithm domain. For the reduced precision parameters, we compute the number of occurrences of an event and use a lookup table to determine the logarithm of the probability of a particular event. The lookup table is indexed in terms of number of occurrences of an event (individual counters) and the total number of events (total counter) and stores values for the logarithms in the desired reduced precision representation. To limit the maximum size of the lookup table and the bit-width required for the counters, we assumed some maximum integer number M. The lookup table L is pre-computed such that:

$$L(i,j) = \left[\frac{ln(i/j)}{q} \right]_R \cdot q \tag{6}$$

where $[\cdot]_R$ denotes rounding to the closest integer, q is the quantization interval of the desired fixed-point representation (2^{-bf}), $ln(\cdot)$ denotes the natural logarithm, and where the counters i and j are in the range $\{0, ..., M - 1\}$.

Given certain specific data, the individual counters c_j^i and the population C are computed according to Algorithm 1. Following the fixed-point representation, we assumed some maximum integer number M, where $M = 2^{(bf+bi)} - 1$ in terms of number of fractional bits bf and number of integer bits bi. After calculating the cumulative count C, we ensure that it is in range. Different from Tschiatschek's algorithm, we also divide by two the individual counters c_i when C reaches its maximum value (lines 9–12 in Algorithm 1). The problem we encountered with the original algorithm was that sometimes the total counter could be lower than the individual counter. And in order to estimate the mutual information, it gave us poor approximations of the logarithmic probabilities.

3.1. Empirical study

Below we empirically evaluate our limited bit depth mutual information in terms of accuracy —using bias and variance measures— and ranking similarity over synthetic data.

Algorithm 1 Our reduced precision algorithm for MI

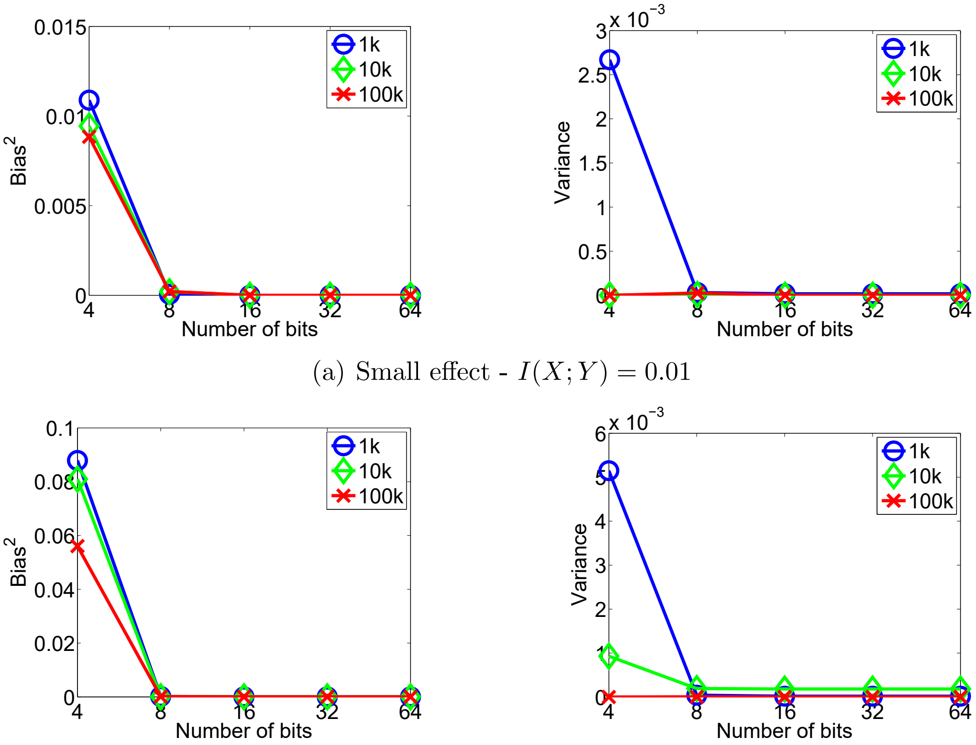
```
Require: Individual counters $c_j^i$ and total counter $C$; lookup table $L$
for $i, j$ do
    if $c_j^i = M$ then                                  ▷ maximum value reached?
        $c_j^i \leftarrow c_j^i/2 \; \forall i, j$                    ▷ half counters (round down)
    end if
end for
$C = \sum (c_j^i)$                                        ▷ sum of the individual counters
while $C \leq M$ do                                      ▷ ensure that $C$ is in range
    $C \leftarrow C/2$
    $c_j^i \leftarrow c_j^i/2 \; \forall i, j$                          ▷ revise index correction
end while
$l_j^i \leftarrow L(c_j^i, C) \; \forall i, j$                ▷ get the log-probability from lookup table
return $l_j^i$
```

3.1.1. Accuracy in terms of bias/variance

To evaluate the performance of the reduced precision mutual information against the full version using a 64-bit representation, we generated synthetic data with two different degrees of dependency with the target class Y. To create the data, firstly we generate the values of Y, by taking n samples from a Bernoulli distribution with $p(y = 1) = 0.5$. Then, we choose the parameters $p(x|y)$ that guarantee the desired degrees of dependency in terms of $I(X; Y)$ and we use these parameters to sample the values of X. All criteria need an estimate of the mutual information between a feature or a feature set and the class variable, which is derived from a finite dataset. For that reason, the accuracy of the estimator plays a crucial role in the ranking of features. The bias and variance are used to measure the accuracy, which can be defined as:

$$\begin{aligned} bias\left(\hat{I}(X;Y)\right) &= \mathbb{E}\left[\hat{I}(X;Y)\right] - I(X;Y) \\ var\left(\hat{I}(X;Y)\right) &= \mathbb{E}\left[\left(\hat{I}(X;Y) - \mathbb{E}\left[\hat{I}(X;Y)\right]\right)^2\right] \end{aligned}$$

Figure 2 shows the results of an experimental study considering two different degrees of dependency, $I(X; Y) = 0.01$ and $I(X; Y) = 0.1$, and three sample sizes, 1k, 10k and 100k. Bias/variance is obtained for the different limited bit depth mutual information versions (4, 8, 16 and 32 bits) and the full mutual information (64 bits), which will be the baseline method for comparison. As can be observed, the bias for 8, 16 and 32 bits converges to the 64-bit representation. Besides, the reduced precision MI using 4 bits does not converge but it is consistent, since both bias and variance decrease as the sample size increases.



(a) Small effect - $I(X;Y) = 0.01$

(b) Large effect - $I(X;Y) = 0.1$

Figure 2: Comparing the performance of our bit limited depth mutual information (4, 8, 16 and 32 bits) with the full mutual information (64 bits) in terms of bias^2/variance. To estimate bias/variance we average over 5000 runs. Please note different axes for the different variables Bias and Variance.

3.1.2. Similarity rankings

Our limited bit depth mutual information described above will be used within a feature selection procedure. The output of a feature selection algorithm might be: a scoring over the features, a ranking of the features or a feature subset. In this section, we aim at illustrating the performance of our limited bit mutual information in terms of feature ranking variability. Let us assume there are d features in total. A ranking r can be formed as a vector of d distinct natural numbers taken from 1 to d. To measure the similarity of the feature rankings obtained by the reduced precision mutual information with different number of bits, we use the Spearman rank-order correlation coefficient (Best and Roberts, 1975), also commonly called Spearman's ρ. This coefficient takes values in the range $[-1, 1]$, where 1 means that the two rankings are identical, -1 means that there is no correlation between them. To be able to do this, we need to know the "true" ranking (Sechidis and Brown, 2018). For this task, we generated various synthetic datasets consisting of $d = 10$ and $d = 20$ features with different degrees of dependency with the target class Y in terms of mutual information. The mutual information $I(X, Y)$ population values for each feature are:

- "Easy" scenario with 10 features: [2 4 6 8 10 12 14 16 18 20] $\times 10^{-2}$.
- "Difficult" scenario with 20 features: [2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21] $\times 10^{-2}$.

where a high mutual information translates into a high rank of the feature. The arity of features is chosen randomly between the following values $|\mathcal{X}| = 2, 5, 10$ and 20. The experiment was repeated taking different sample sizes from 1000 to 100,000 samples to observe the performance when the sample size increases. To estimate the Spearman's ρ we average over 100 runs.

Table 2 shows the Spearman's ρ obtained for the different limited bit depth mutual information versions (4, 8, 16 and 32 bits) and the full mutual information (64 bits). The lower values of the reduced precision approach using 4 bits shows that the correlation between its ranking and the "true" ranking is quite poor in both scenarios. However, from 8 bits all the approaches achieved a Spearman's ρ coefficient close to 1, which means that the rankings obtained by these approaches are similar to the "true" rankings. Moreover, we can observe that by increasing the sample size all of the reduced precision approaches improve their rankings, and they are closer to the "true" ranking in both scenarios. However, differences between both scenarios can be seen when 16 and 32 bits are used. In the difficult scenario, these limited bit depth MI versions do not get the "true ranking". This could be because there is a smaller distance between the population values of the mutual information, and thus the ranking will change.

In light of the results obtained, we proceed to use our limited bit depth mutual information approach within a more sophisticated method. In this

Table 2: Spearman's ρ coefficient

#Features	#Samples	#Bits				
		4	8	16	32	64
10	1000	0.215	0.963	0.983	0.983	0.983
	10,000	0.326	0.963	1.000	1.000	1.000
	100,000	0.429	0.974	1.000	1.000	1.000
20	1000	0.179	0.975	0.973	0.973	0.973
	10,000	0.320	0.973	0.995	0.995	0.995
	100,000	0.472	0.984	0.996	0.996	0.996

work, we have chosen to apply this to feature selection. Despite the poor results using 4 bits, we have kept this approach in order to see how it affects to the accuracy of feature selection methods.

4. Application in feature selection

Our bit limited depth mutual information described above can be applicable to any method that uses internally the mutual information measure. In this work, we have chosen to do so within feature selection since this process has a key role to play in helping reduce high-dimensionality in machine learning problems (Bolón-Canedo et al., 2015), and it is lately specially relevant with the advent of Big Data. There is a large number of feature selection methods that use mutual information as a measure, thus their performance depending on the accuracy obtained by the mutual information step. As mentioned before, we have chosen to implement our reduced precision approach in the MIM, JMI and mRMR filters methods due to their popularity and good results in the machine learning area, but analogous implementations could be derived for any other FS method based on mutual information. We have considered several synthetic —of which the relevant features are already known— and real datasets. Table 3 details the main characteristics of the chosen datasets: for each dataset, the number of features, the number of samples and the number of classes. Experiments were executed in the Matlab2018a and Weka environments.

- **UCI** datasets (Lichman, 2013). This is a collection of datasets of which we have selected Arcene, Congress, Connect-4, Splice and Waveform,

Table 3: Characteristics of the datasets.

Dataset	Type	#Features	#Samples	#Classes
Arcene	Real	10,000	200	2
Congress	Real	16	435	2
Connect-4	Real	42	45,038	3
CorrAL-100	Synthetic	100	100,000	2
GISETTE	Synthetic	5000	6000	2
Led-500	Synthetic	500	200,000	10
Splice	Real	60	3175	3
Waveform	Real	40	5000	3

with small to medium number of samples. The features within each dataset have a variety of characteristics: some are binary/discrete, and some are continuous. Continuous features were discretized, using an equal-width strategy in 5 bins, while features already with a categorical range were left untouched.

- **GISETTE** is a handwritten digit recognition problem from the NIPS 2003 Feature Selection Challenge (Guyon et al.). Features were discretized independently into 10 equal width bins.

- **CorrAL-100**. The CorrAL dataset (John et al., 1994) has six binary features (f_1, f_2, f_3, f_4, f_5, f_6), and its class value is $(f_1 \wedge f_2) \vee (f_3 \wedge f_4)$. Feature f_5 is irrelevant and f_6 is correlated to the class label by 75%. CorrAL-100 was constructed by adding 93 features irrelevant binary features to the previous CorrAL dataset. The data for the added features was generated randomly. The correct behavior for a given feature selection method is to select the four relevant features and to discard the irrelevant and correlated ones. The correlated feature is redundant if the four relevant features are selected and, besides, it is correlated to the class label by 75%, so if one applies a classifier using only this feature, a 25% of error should be obtained.

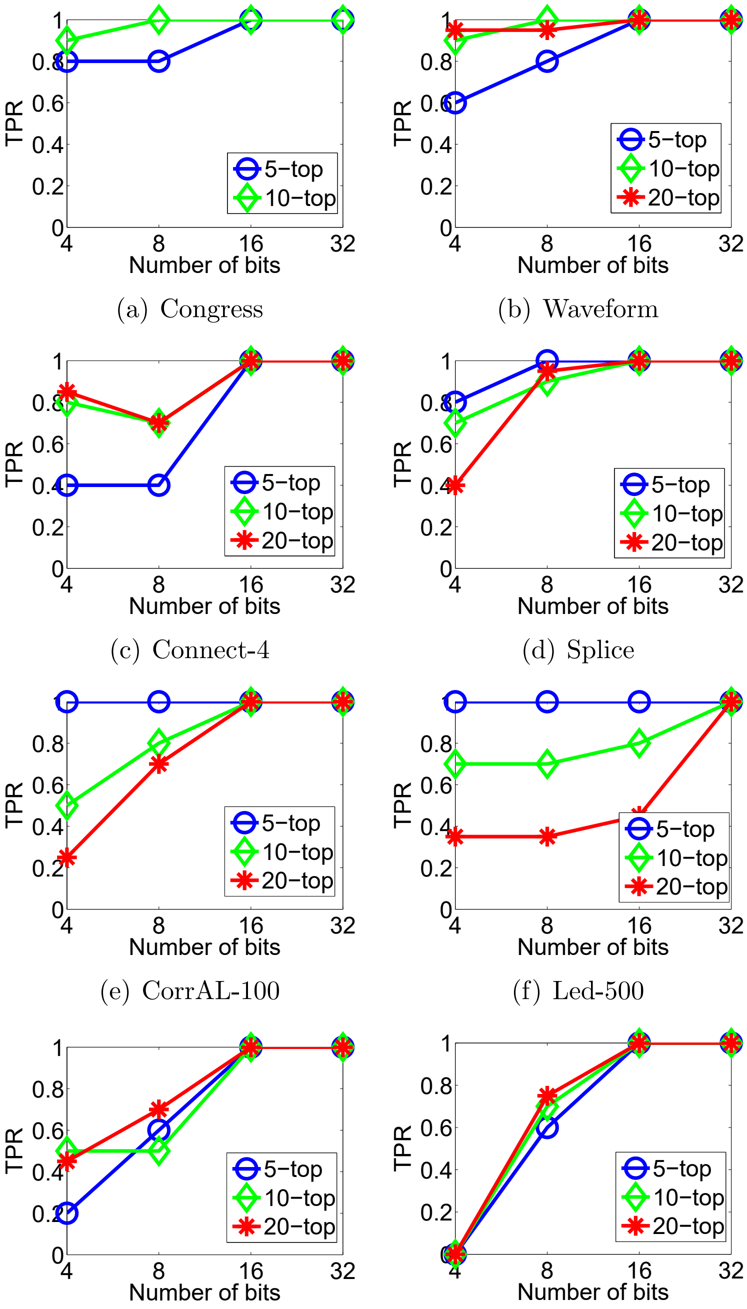
- **LED-500**. The LED problem (Breiman et al., 1984) is a simple classification task that consists of, given the active LEDs on a seven segments display, identifying the digit that the display is representing. Thus, the classification task to be solved is described by seven binary attributes and ten possible classes available. A 1 in a attribute indicates that the LED is active, and a 0 indicates it is not active. Led-500 was constructed by adding 493 irrelevant binary features.

In the following sections we present and discuss the experimental results in terms of the quality of the selected features and the classification accuracy.

4.1. Quality of the selected features

To evaluate the similarity between the rankings obtained by the reduced precision versions and the 64-bit mutual information after performing the MIM, JMI and mRMR methods, we show the true positive rate for each dataset. The true positive rate measures the proportion of features that are correctly identified as such, using the full mutual information version (64 bits) as the ideal ranking. In high dimensional datasets, it is common to focus only on the top features, so in these experiments we compare only the k top features, with $k = 5, 10$ and 20 for all datasets except Congress, for which only the results with the 5 and 10 top features are shown as it has only 16 features.

Figures 3, 4 and 5 show the true positive rate (TPR) over the eight datasets presented in Table 3. The datasets are sorted in ascending order by their number of total features. As can be seen, for the datasets with less than 100 features —Congress, Waveform, Connect-4 and Splice—, our reduced precision approach using only 16 bits selected the same $5, 10$ and 20 features that the full version. Moreover, for the smaller datasets in terms of sample size, Congress and Splice, the reduced precision approach was able to achieve a 100% true positive rate even using 8 bits. When the number of features of the dataset increases, the performance of our reduced precision version using 16 bits started to decrease, and the same effect appears for datasets with high number of samples. The challenge with high dimensionality can be clearly seen in the Arcene dataset, where the limited bit depth MI using 4 bits does not select correctly any feature. For CorrAL-100 dataset, even the reduced precision version using only 4 bits was able to return the same 5 top features of the full version using 64 bits. It might be happening because this dataset has four relevant features (f_1, f_2, f_3 and f_4) and another feature that is correlated to the class label by 75%. This means that there is a slight difference between the mutual information values of these features and the



(a) Congress

(b) Waveform

(c) Connect-4

(d) Splice

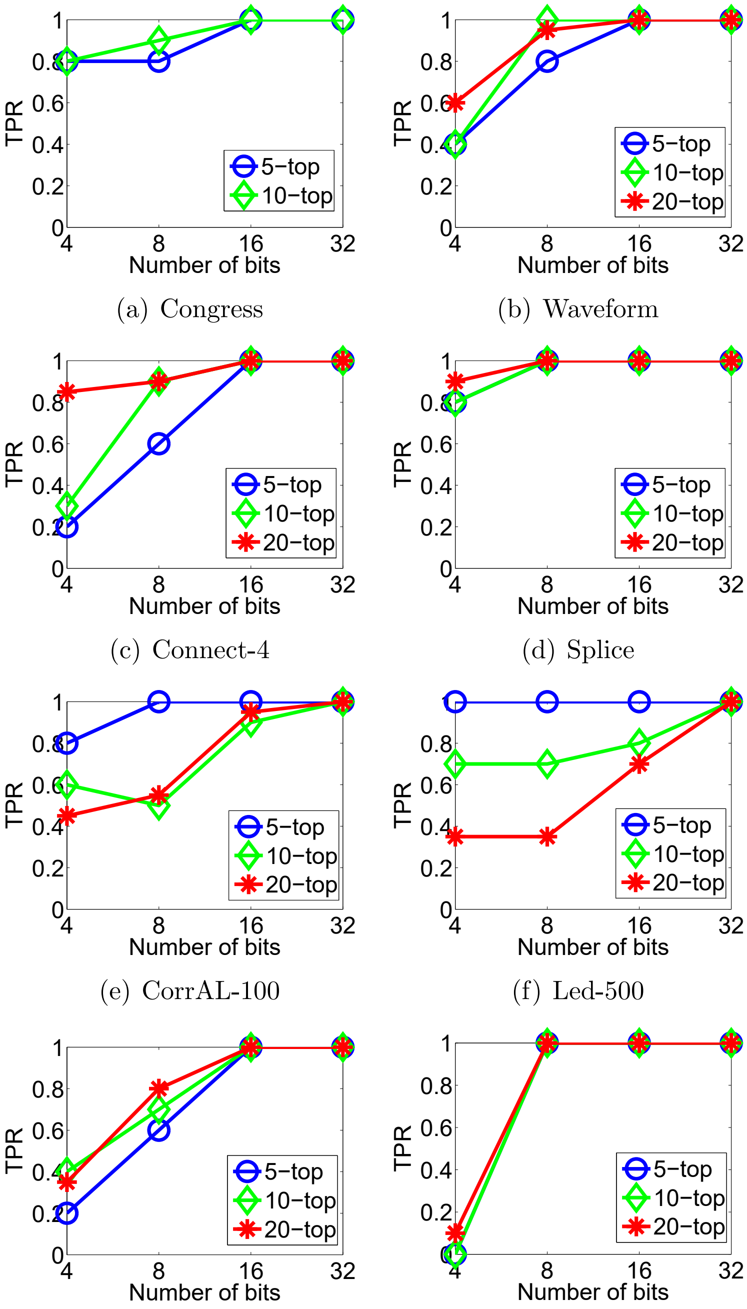
(e) CorrAL-100

(f) Led-500

(g) GISETTE

(h) Arcene

Figure 3: True positive rate of the different reduced precision approaches using MIM.



(a) Congress

(b) Waveform

(c) Connect-4

(d) Splice

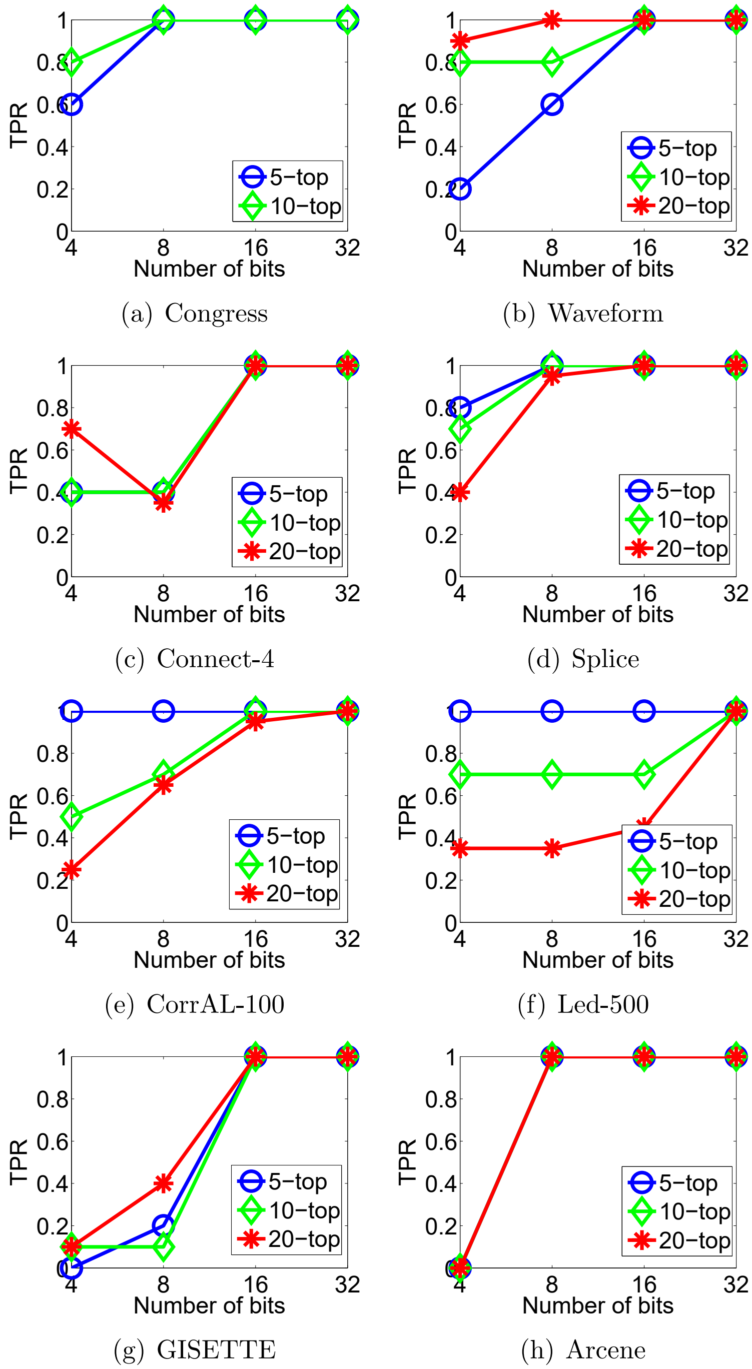
(e) CorrAL-100

(f) Led-500

(g) GISETTE

(h) Arcene

Figure 4: True positive rate of the different reduced precision approaches using JMI.



(a) Congress

(b) Waveform

(c) Connect-4

(d) Splice

(e) CorrAL-100

(f) Led-500

(g) GISETTE

(h) Arcene

Figure 5: True positive rate of the different reduced precision approaches using mRMR.

rest of features. Therefore, we can say that in general, 16 bits are sufficient to select the same features that the full version using 64 bits.

Comparing the results between the different feature selection methods, we can see that JMI performs better —in some cases— than MIM and mRMR when 8 bits are used. This could be because JMI criterion has the best trade-off in terms of stability and flexibility over other feature selection methods based on Information Theory due to its nature (it balances the relevancy and redundancy terms and includes the conditional redundancy) (Brown et al., 2012).

4.2. Classification accuracy

After the feature selection process, and in order to estimate whether the reduced precision in the feature selection process might affect classification, a study using classifiers was carried out. At this point, it is necessary to clarify that including classifiers in our experiments is likely to obscure the experimental observations related to feature selection performance using a limited number of bits, since they include their own assumptions and particularities. Therefore, in these experiments, we used a simple nearest neighbor algorithm (with number of neighbors $k = 3$) (Aha et al., 1991) as classifier since it makes few assumptions about the data, and we avoid the need for parameter tuning. To estimate the error rate we computed a 5-fold cross validation. For evaluating the performance of the reduced precision approaches, we compared the results obtained when using the ranking built with 4, 8, 16, 32 and 64 bits. Due to the large number of results, some tables have been moved to Appendix A.

To explore the statistical significance of our classification results, we analyzed the ranks of the reduced precision approaches by using a Friedman test with the Nemenyi post-hoc test. Figures 6, 7 and 8 present the critical difference diagrams, introduced by Demšar (2006), where groups of methods that are not significantly different (at $\alpha = 0.10$) are connected. As can be seen for the three different k top selected features and JMI and mRMR methods, 64, 32 and 16 bits perform better on average but with no statistical significance over the reduced precisions approaches using only 4 and 8 bits, with the exception of mRMR (Figure 8). In the case of MIM, although there is no statistical significance over the reduced precisions approaches, the best performance is not always achieved through versions with 16, 32 or 64 bits. This could be because this last method assumes that each feature is independent of all other features. However, where features may be interdependent, this is known to be suboptimal. In general, it is widely accepted that a useful and parsimonious set of features should not only be individually relevant, but also should not be redundant with respect to each other—features should not be highly correlated (Brown et al., 2012).

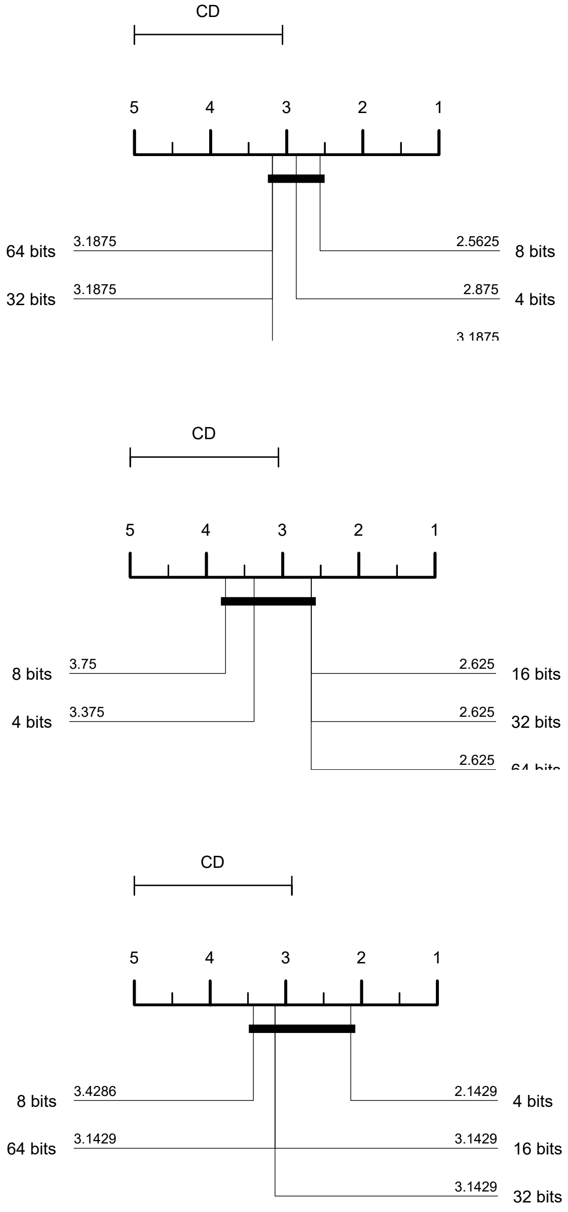
In summary, these experiments demonstrate that with a small number of bits the rankings change, but this variation does not affect significantly the classification accuracy, since this measure is the ultimate form of evaluation of the goodness of a feature ranking method.

4.3. Case study: Dealing with noise in the inputs: LED

The LED dataset consists of correctly identifying seven LEDs that represent numbers between 0 and 9. Some irrelevant features were added forming the Led-500 dataset (493 irrelevant features). In order to make this dataset more complex, different levels of noise in the inputs (6%, 10% and 20%) were added (de Amorim and Hennig, 2015). In this manner, the tolerance to different levels of noise of the bit limited depth MI tested will be checked. Note that, as the attributes take binary values, adding noise means assigning to the relevant features an incorrect value. Besides, and unlike the Led-500 dataset used above, the number of samples was reduced to 10,000 so that its volume does not affect the study of noise.

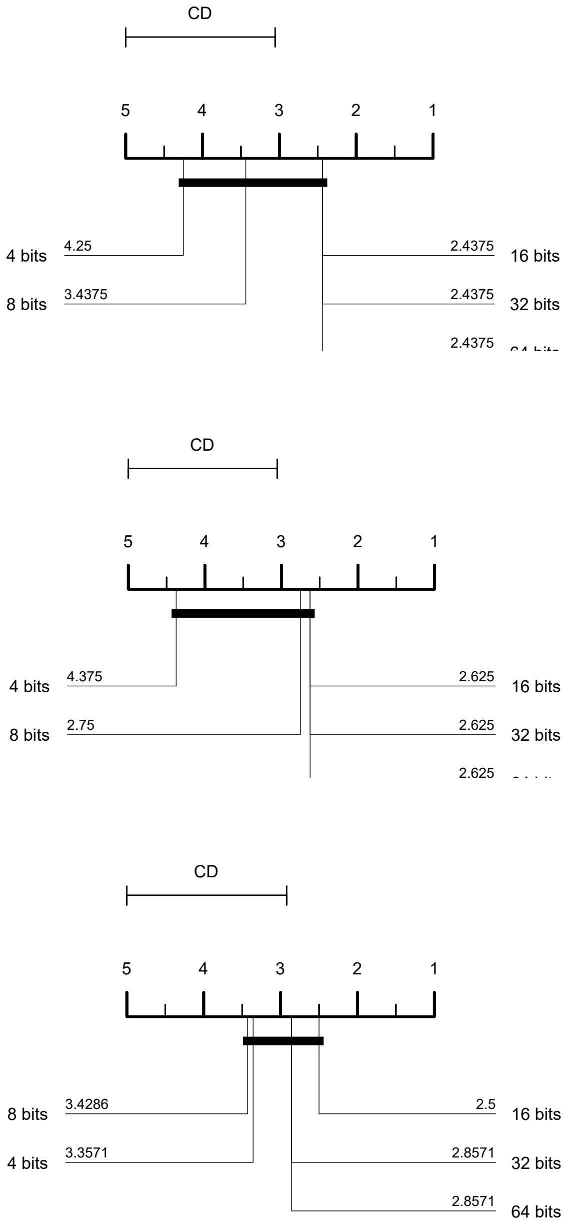
In this case study, we consider JMI as the feature selection method due to the results obtained in Section 4.1. Figure 9 depicts detailed results of these experiments. It is interesting to note that the presence of noise does not seem to influence our limited bit depth MI, except in the case of 20% of noise and 5-top features. The reduced precision approach was not able to achieve a 100% true positive rate using just 8 bits.

With regard to the classification accuracy, it decreases as the level of noise increases, as expected (Table 4). However, the results using 4 bits are somewhat misleading. The 4-bit reduced precision version achieved better results in terms of classification accuracy than other versions with a larger number of bits. This is happening because —due to the high number of features of LED dataset and the low values of mutual information (99% under 0.006, see Figure 10)— the reduced precision approach does not get to sort the features by following the JMI criterion and it returns the feature ranking following its order in the original dataset. And, in this particular synthetic dataset, the classification task to be solved is described by the first seven binary attributes.



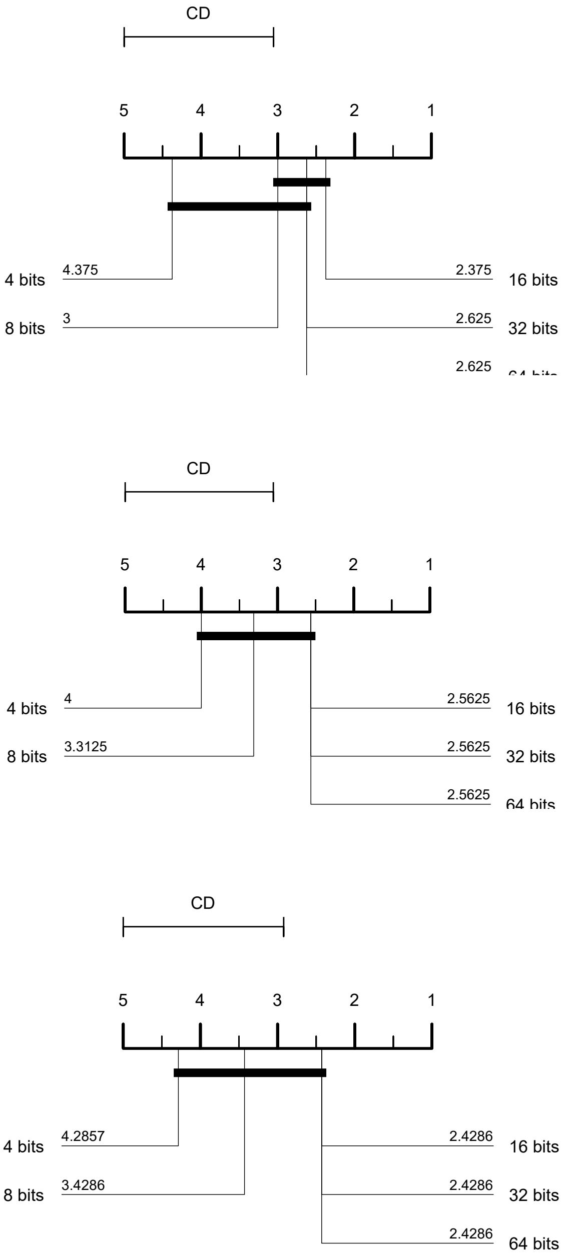
(c) 20-top features

Figure 6: Critical difference diagrams showing the average ranks after applying MIM on the four reduced precision approaches (4, 8, 16 and 32 bits) and the full version (64 bits) for three different k-top selected features.



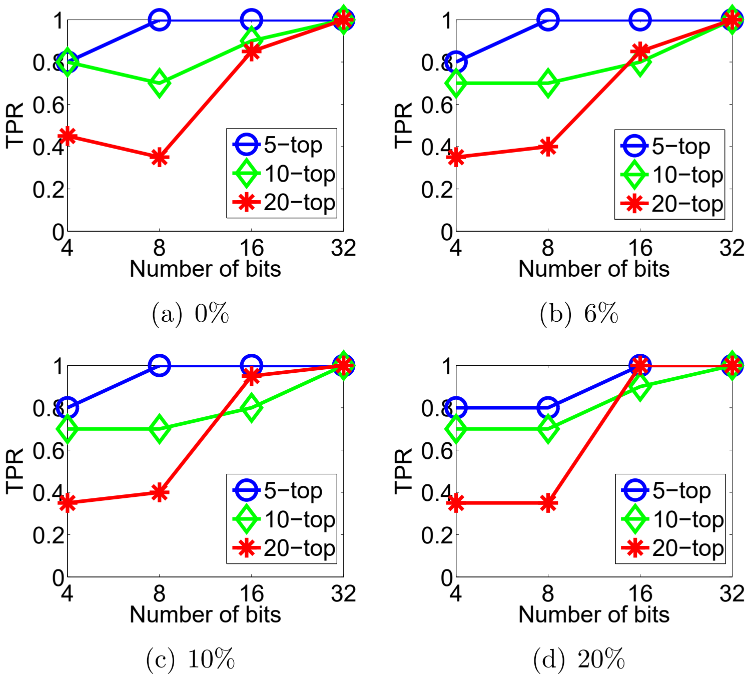
(c) 20-top features

Figure 7: Critical difference diagrams showing the average ranks after applying JMI on the four reduced precision approaches (4, 8, 16 and 32 bits) and the full version (64 bits) for three different *k*-top selected features.



(c) 20-top features

Figure 8: Critical difference diagrams showing the average ranks after applying mRMR on the four reduced precision approaches (4, 8, 16 and 32 bits) and the full version (64 bits) for three different *k*-top selected features.



(a) 0%

(b) 6%

(c) 10%

(d) 20%

Figure 9: True positive rate of the different reduced precision approaches using JMI over LED dataset with different levels of noise (6%, 10% and 20%).

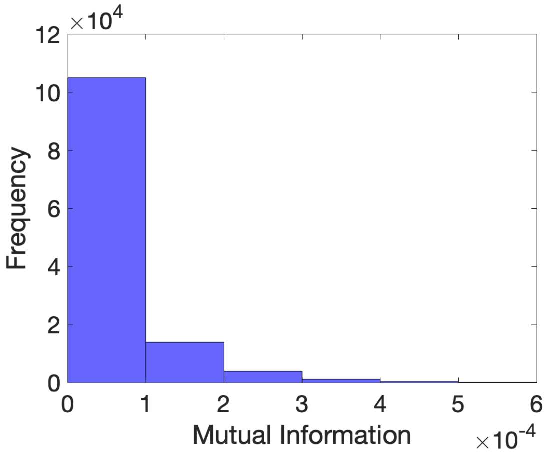


Figure 10: Histogram of frequency distribution values of mutual information of LED dataset.

Table 4: Classification accuracy (%) and standard deviation for LED dataset with different levels of noise (6%, 10% and 20%).

Top features	Noise (%)	#Bits				
		4	8	16	32	64
5	0	100.00 ± 0.00	89.67 ± 0.00	89.67 ± 0.00	89.67 ± 0.00	89.67 ± 0.00
	6	94.00 ± 0.00	84.16 ± 0.00	84.16 ± 0.00	84.16 ± 0.00	84.16 ± 0.00
	10	90.00 ± 0.01	81.10 ± 0.01	81.10 ± 0.01	81.10 ± 0.01	81.10 ± 0.01
	20	80.00 ± 0.01	64.52 ± 0.01	72.27 ± 0.01	72.27 ± 0.01	72.27 ± 0.01
10	0	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00
	6	94.00 ± 0.00	94.00 ± 0.00	94.00 ± 0.00	94.00 ± 0.00	94.00 ± 0.00
	10	90.00 ± 0.00	90.00 ± 0.00	90.00 ± 0.00	90.00 ± 0.00	90.00 ± 0.00
	20	80.00 ± 0.01	80.00 ± 0.01	80.00 ± 0.01	80.00 ± 0.01	80.00 ± 0.01
20	0	98.71 ± 0.00	98.93 ± 0.00	98.75 ± 0.00	98.79 ± 0.00	98.79 ± 0.00
	6	92.52 ± 0.00	92.18 ± 0.01	92.33 ± 0.01	92.23 ± 0.01	92.23 ± 0.01
	10	87.90 ± 0.00	87.85 ± 0.00	87.99 ± 0.01	87.84 ± 0.01	87.84 ± 0.01
	20	76.65 ± 0.00	76.48 ± 0.00	76.39 ± 0.00	76.39 ± 0.00	76.39 ± 0.00

4.4. Comparison with baseline method

As we mentioned above, (Tschiatschek and Pernkopf, 2015) proposed Bayesian Network classifiers when reducing the precision of the probability parameters. Since mutual information also needs to estimate probabilities, our work was built upon this idea. In order to analyze Tschiatschek's algorithm on the mutual information measure, we generated synthetic data in the same way as in Section 3. The degree of dependence with the target class in terms of mutual information was fixed to 0.1 and the number of samples to 10,000. All criteria need an estimate of the mutual information between a feature or a feature set and the class variable, which is derived from finite dataset. For that reason, the accuracy of the estimator plays a crucial role in the ranking of features. To measure the accuracy we use the *Mean Square Error* (MSE), which is calculated from the ground truth we know from artificial data. To estimate MSE we averaged over 5000 runs. Both look up tables are calculated in natural logarithm, so the returned units are "nats".

Figure 11 compares Tschiatschek's algorithm with our proposal in terms of MSE. As can be seen, for the reduced precision approaches using only 4 and 8 bits, Tschiatschek's algorithm obtained high values of MSE while our proposed limited bit mutual information method achieved values close to zero. Besides, we can observe that with 16 and 32 bits both algorithms converge. It is necessary to clarify that Tschiatschek and Pernkopf (2015) performed their experiments using 10 bits. As we aimed to explore the effect of the reduced precision with a small amount of bits, we redefined this algorithm with the aim of achieving better performance for our limited bit mutual information version.

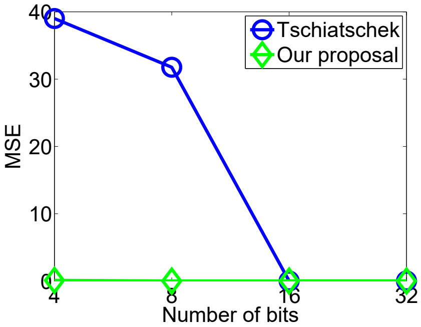


Figure 11: Comparing the performance of Tschiatschek's algorithm with our proposed reduced precision mutual information in terms of Mean Square Error, $I(X;Y) = 0.1$.

5. Conclusions

Since the development and commercialization of wearable technology is growing expansively, we have seen an opportunity to develop machine learning methods, specifically feature selection algorithms, in computationally constrained platforms. In this work we have proposed mutual information using reduced precision parameters within a feature selection procedure. Experimental results over several synthetic and real datasets have shown that 16 bits are sufficient to return the same feature ranking than that of 64-bit representation. As a result, meaningful benefits will be provided when implementing mutual information in embedded systems for on-device analysis. Having on device machine learning has some tremendous profits regarding privacy, reliability, efficient use of network bandwidth and power saving.

From the experiments carried out, we can draw some conclusions and make some recommendations to the users:

- Our reduced precision approach will not be adequate if there is a small distance between the population values of the mutual information. Besides, the ranking will be more unstable in the bottom of the list, where the features contain less and less information.

- When the number of features of the dataset increases, we will need more bits. Nevertheless, it is important to note that our reduced precision approach was designed to analyze user level data. If we are working in a big data scenario, data is probably collected from different users, and so it would be processed either by more powerful central processors or distributed in different nodes by further analysis.

- Regarding the three feature selection methods used to test our limited bit depth mutual information, we have found that JMI was the most stable. However, if we take into account the computational cost of these methods, MIM seems to the most appropriate for this scenario.

- With respect to the presence of noise, it does not seem to influence appreciably our limited bit depth MI . In terms of classification accuracy, and as expected, it decreases as the level of noise increases.

The results obtained by the proposed approach open the door to its use in other feature selection algorithms based on MI and as preprocessing step of some low precision classifiers (Tschiatschek and Pernkopf, 2015). Thus, while initial finding are promising, further research is necessary. As future research, we plan to use our limited bit depth mutual information in a Markov Blanket context.

Acknowledgment

This research has been financially supported in part by the Spanish Ministerio de Economía y Competitividad (research project TIN2015-65069-C2-1-R), by European Union FEDER funds and by the Consellería de Industria of the Xunta de Galicia (research project GRC2014 /035). Financial support from the Xunta de Galicia (Centro singular de investigación de Galicia accreditation 2016-2019) and the European Union (European Regional Development Fund - ERDF), is gratefully acknowledged (research project ED431G/01). Project supported by a 2018 Leonardo Grant for Researchers and Cultural Creators, BBVA Foundation. Laura Morán-Fernández acknowledges predoctoral stay grant by INDITEX-UDC 2015.

References

Aha, D.W., Kibler, D., Albert, M.K., 1991. Instance-based learning algorithms. Machine learning 6, 37–66.

de Amorim, R.C., Hennig, C., 2015. Recovering the number of clusters in data sets with noise features using feature rescaling factors. Information Sciences 324, 126–145.

Battiti, R., 1994. Using mutual information for selecting features in supervised neural net learning. IEEE Transactions on neural networks 5, 537–550.

Best, D., Roberts, D., 1975. Algorithm as 89: the upper tail probabilities of spearman's rho. Journal of the Royal Statistical Society. Series C (Applied Statistics) 24, 377–379.

Bolón-Canedo, V., Sánchez-Maroño, N., Alonso-Betanzos, A., 2015. Recent advances and emerging challenges of feature selection in the context of big data. Knowledge-Based Systems 86, 33–45.

Breiman, L., Friedman, J., Stone, C.J., Olshen, R.A., 1984. Classification and regression trees. CRC press.

Brown, G., Pocock, A., Zhao, M.J., Luján, M., 2012. Conditional likelihood maximisation: a unifying framework for information theoretic feature selection. Journal of machine learning research 13, 27–66.

Commons, C., 2001. Creative commons. url-https://search.creativecommons.org/.

Demšar, J., 2006. Statistical comparisons of classifiers over multiple data sets. Journal of Machine learning research 7, 1–30.

Guo, B., Nixon, M.S., 2009. Gait feature subset selection by mutual information. IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans 39, 36–46.

Gupta, S., Agrawal, A., Gopalakrishnan, K., Narayanan, P., 2015. Deep learning with limited numerical precision, in: Proceedings of the 32nd International Conference on Machine Learning (ICML-15), pp. 1737–1746.

Guyon, I., Gunn, S., Ben-Hur, A., Dror, G., . Nips 2003 workshop on feature extraction. http://clopinet.com/isabelle/Projects/NIPS2003/.

Han, S., Liu, X., Mao, H., Pu, J., Pedram, A., Horowitz, M.A., Dally, W.J., 2016. Eie: efficient inference engine on compressed deep neural network, in: 2016 ACM/IEEE 43rd Annual International Symposium on Computer Architecture (ISCA), IEEE. pp. 243–254.

Huang, S.H., 2015. Supervised feature selection: A tutorial. Artif. Intell. Research 4, 22–37.

Hubara, I., Courbariaux, M., Soudry, D., El-Yaniv, R., Bengio, Y., 2017. Quantized neural networks: Training neural networks with low precision weights and activations. The Journal of Machine Learning Research 18, 6869–6898.

Jacob, B., Kligys, S., Chen, B., Zhu, M., Tang, M., Howard, A., Adam, H., Kalenichenko, D., 2018. Quantization and training of neural networks for efficient integer-arithmetic-only inference, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2704–2713.

John, G.H., Kohavi, R., Pfleger, K., et al., 1994. Irrelevant features and the subset selection problem, in: Machine learning: proceedings of the eleventh international conference, pp. 121–129.

Koopman, P., 1990. Design constraints on embedded real time control systems .

Lewis, D.D., 1992. Feature selection and feature extraction for text categorization, in: Proceedings of the workshop on Speech and Natural Language, Association for Computational Linguistics. pp. 212–217.

Lichman, M., 2013. UCI machine learning repository. URL: `http://archive.ics.uci.edu/ml`.

Morán-Fernández, L., Bolón-Canedo, V., Alonso-Betanzos, A., 2018. Feature selection with limited bit depth mutual information for embedded systems. Multidisciplinary Digital Publishing Institute Proceedings 2, 1187.

Murshed, M., Murphy, C., Hou, D., Khan, N., Ananthanarayanan, G., Hussain, F., 2019. Machine learning at the network edge: A survey. arXiv preprint arXiv:1908.00080 .

Paninski, L., 2003. Estimation of entropy and mutual information. Neural computation 15, 1191–1253.

Peng, H., Long, F., Ding, C., 2005. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. IEEE Transactions on pattern analysis and machine intelligence 27, 1226–1238.

Ray, S., Park, J., Bhunia, S., 2016. Wearables, implants, and internet of things: the technology needs in the evolving landscape. IEEE Transactions on Multi-Scale Computing Systems 2, 123–128.

Saeys, Y., Inza, I., Larrañaga, P., 2007. A review of feature selection techniques in bioinformatics. bioinformatics 23, 2507–2517.

Sechidis, K., Azzimonti, L., Pocock, A., Corani, G., Weatherall, J., Brown, G., 2019. Efficient feature selection using shrinkage estimators. Machine Learning 108, 1261–1286.

Sechidis, K., Brown, G., 2018. Simple strategies for semi-supervised feature selection. Machine Learning 107, 357–395.

Tesmer, M., Estévez, P.A., 2004. Amifs: Adaptive feature selection by using mutual information, in: Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on, IEEE. pp. 303–308.

Tschiatschek, S., Pernkopf, F., 2015. Parameter learning of bayesian network classifiers under computational constraints, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer. pp. 86–101.

Yang, H.H., Moody, J., 2000. Data visualization and feature selection: New algorithms for nongaussian data, in: Advances in neural information processing systems, pp. 687–693.

Appendix A. Tables for the experimental study

This appendix reports the experimental results achieved in this work. Tables A.1, A.2 and A.3 depict the classification accuracy (%) and standard deviation for the MIM, JMI and mRMR feature selection methods, respectively.

Table A.1: Classification accuracy (%) and standard deviation for MIM method.

Top features	Dataset	#Bits				
		4	8	16	32	64
5	Congress	92.64 ± 0.03	94.25 ± 0.01	94.02 ± 0.02	94.02 ± 0.02	94.02 ± 0.02
	Waveform	71.54 ± 0.01	69.46 ± 0.01	68.30 ± 0.01	68.30 ± 0.01	68.30 ± 0.01
	Connect-4	71.81 ± 0.00	69.42 ± 0.00	71.97 ± 0.00	71.97 ± 0.00	71.97 ± 0.00
	Splice	89.32 ± 0.01	88.25 ± 0.01	88.2 ± 0.01	88.25 ± 0.01	88.25 ± 0.01
	CorrAL-100	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00
	Led-500	79.83 ± 0.00	79.83 ± 0.00	79.83 ± 0.00	79.83 ± 0.00	79.83 ± 0.00
	GISETTE	89.65 ± 0.01	86.75 ± 0.01	84.10 ± 0.01	84.10 ± 0.01	84.10 ± 0.01
	Arcene	72.50 ± 0.04	81.00 ± 0.07	76.00 ± 0.04	76.00 ± 0.04	76.00 ± 0.04
10	Congress	91.95 ± 0.02	93.10 ± 0.02	93.10 ± 0.02	93.10 ± 0.02	93.10 ± 0.02
	Waveform	79.80 ± 0.02	78.66 ± 0.02	78.66 ± 0.02	78.66 ± 0.02	78.66 ± 0.02
	Connect-4	74.72 ± 0.01	74.32 ± 0.01	76.55 ± 0.01	76.55 ± 0.01	76.55 ± 0.01
	Splice	84.38 ± 0.01	86.30 ± 0.01	86.55 ± 0.01	86.55 ± 0.01	86.55 ± 0.01
	CorrAL-100	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00
	Led-500	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00
	GISETTE	92.17 ± 0.00	90.17 ± 0.00	90.68 ± 0.00	90.68 ± 0.00	90.68 ± 0.00
	Arcene	73.50 ± 0.05	80.00 ± 0.05	81.00 ± 0.02	81.00 ± 0.02	81.00 ± 0.02
20	Waveform	80.12 ± 0.01	79.32 ± 0.01	80.06 ± 0.01	80.06 ± 0.01	80.06 ± 0.01
	Connect-4	78.51 ± 0.01	76.56 ± 0.00	77.76 ± 0.00	77.76 ± 0.00	77.76 ± 0.00
	Splice	79.05 ± 0.02	79.09 ± 0.01	78.99 ± 0.01	78.99 ± 0.01	78.99 ± 0.01
	CorrAL-100	97.75 ± 0.00	97.81 ± 0.00	97.74 ± 0.00	97.74 ± 0.00	97.74 ± 0.00
	Led-500	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00
	GISETTE	93.42 ± 0.01	91.40 ± 0.01	91.47 ± 0.00	91.47 ± 0.00	91.47 ± 0.00
	Arcene	78.50 ± 0.05	81.50 ± 0.06	83.50 ± 0.04	83.50 ± 0.04	83.50 ± 0.04

Table A.2: Classification accuracy (%) and standard deviation for JMI method.

Top features	Dataset	#Bits				
		4	8	16	32	64
5	Congress	90.80 ± 0.00	93.79 ± 0.00	95.86 ± 0.00	95.86 ± 0.00	95.86 ± 0.00
	Waveform	60.74 ± 0.00	75.08 ± 0.00	75.26 ± 0.00	75.26 ± 0.00	75.26 ± 0.00
	Connect-4	89.10 ± 0.00	88.44 ± 0.00	88.44 ± 0.00	88.44 ± 0.00	88.44 ± 0.00
	Splice	89.10 ± 0.00	88.44 ± 0.00	88.44 ± 0.00	88.44 ± 0.00	88.44 ± 0.00
	CorrAL-100	90.62 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00
	Led-500	89.95 ± 0.00	89.95 ± 0.00	89.95 ± 0.00	89.95 ± 0.00	89.95 ± 0.00
	GISETTE	85.88 ± 0.00	89.23 ± 0.00	91.38 ± 0.00	91.38 ± 0.00	91.38 ± 0.00
	Arcene	78.50 ± 0.05	83.00 ± 0.05	83.00 ± 0.05	83.00 ± 0.05	83.00 ± 0.05
10	Congress	92.41 ± 0.02	95.17 ± 0.02	94.25 ± 0.02	94.25 ± 0.02	94.25 ± 0.02
	Waveform	66.04 ± 0.01	79.94 ± 0.00	79.94 ± 0.00	79.94 ± 0.00	79.94 ± 0.00
	Connect-4	71.37 ± 0.01	75.96 ± 0.01	76.49 ± 0.01	76.49 ± 0.01	76.49 ± 0.01
	Splice	85.38 ± 0.01	87.05 ± 0.01	87.02 ± 0.01	87.02 ± 0.01	87.02 ± 0.01
	CorrAL-100	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00
	Led-500	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00
	GISETTE	91.25 ± 0.00	90.28 ± 0.01	92.67 ± 0.01	92.67 ± 0.01	92.67 ± 0.01
	Arcene	79.00 ± 0.08	80.50 ± 0.11	80.50 ± 0.11	80.50 ± 0.11	80.50 ± 0.11
20	Waveform	72.84 ± 0.01	80.04 ± 0.01	79.92 ± 0.01	79.92 ± 0.01	79.92 ± 0.01
	Connect-4	77.90 ± 0.00	77.66 ± 0.00	78.41 ± 0.00	78.41 ± 0.00	78.41 ± 0.00
	Splice	80.03 ± 0.01	79.05 ± 0.01	79.34 ± 0.01	79.34 ± 0.01	79.34 ± 0.01
	CorrAL-100	97.80 ± 0.00	97.68 ± 0.00	97.76 ± 0.00	97.68 ± 0.00	97.68 ± 0.00
	Led-500	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00
	GISETTE	93.57 ± 0.00	93.57 ± 0.00	93.72 ± 0.00	93.72 ± 0.00	93.72 ± 0.00
	Arcene	77.50 ± 0.07	83.00 ± 0.09	83.00 ± 0.09	83.00 ± 0.09	83.00 ± 0.09

Table A.3: Classification accuracy (%) and standard deviation for mRMR method.

Top features	Dataset	#Bits				
		4	8	16	32	64
5	Congress	94.02 ± 0.02	94.71 ± 0.02	94.71 ± 0.02	94.71 ± 0.02	94.71 ± 0.02
	Waveform	71.24 ± 0.01	73.44 ± 0.02	76.26 ± 0.01	76.26 ± 0.01	76.26 ± 0.01
	Connect-4	69.39 ± 0.00	69.51 ± 0.00	70.74 ± 0.00	70.74 ± 0.00	70.74 ± 0.00
	Splice	87.97 ± 0.01	89.20 ± 0.01	87.99 ± 0.01	87.97 ± 0.01	87.97 ± 0.01
	CorrAL-100	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00
	Led-500	89.95 ± 0.00	89.95 ± 0.00	89.95 ± 0.00	89.95 ± 0.00	89.95 ± 0.00
	GISETTE	84.82 ± 1.21	87.87 ± 0.97	89.63 ± 0.72	89.63 ± 0.72	89.63 ± 0.72
	Arcene	70.50 ± 0.04	76.50 ± 0.06	76.50 ± 0.06	76.50 ± 0.06	76.50 ± 0.06
10	Congress	99.07 ± 0.03	99.05 ± 0.02	99.05 ± 0.02	99.05 ± 0.02	99.05 ± 0.02
	Waveform	77.78 ± 0.01	78.88 ± 0.016	79.74 ± 0.01	79.74 ± 0.01	79.74 ± 0.01
	Connect-4	70.84 ± 0.00	71.35 ± 0.00	72.61 ± 0.00	72.61 ± 0.00	72.61 ± 0.00
	Splice	84.50 ± 0.01	86.80 ± 0.01	86.80 ± 0.01	86.80 ± 0.015	86.80 ± 0.01
	CorrAL-100	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00
	Led-500	100.00 ± 0.00	100.000 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00
	GISETTE	89.42 ± 0.65	90.47 ± 0.52	91.65 ± 0.70	91.65 ± 0.70	91.65 ± 0.70
	Arcene	69.00 ± 0.04	75.00 ± 0.08	75.00 ± 0.08	75.00 ± 0.08	75.00 ± 0.08
20	Waveform	78.16 ± 0.01	80.08 ± 0.01	80.08 ± 0.01	80.08 ± 0.01	80.08 ± 0.01
	Connect-4	73.16 ± 0.00	71.98 ± 0.00	74.18 ± 0.00	74.18 ± 0.00	74.18 ± 0.00
	Splice	77.51 ± 0.02	79.31 ± 0.01	79.18 ± 0.01	79.18 ± 0.01	79.18 ± 0.01
	CorrAL-100	97.72 ± 0.00	97.68 ± 0.00	97.77 ± 0.00	97.77 ± 0.00	97.77 ± 0.00
	Led-500	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00
	GISETTE	92.38 ± 0.64	91.50 ± 0.56	93.57 ± 0.27	93.57 ± 0.27	93.57 ± 0.27
	Arcene	73.00 ± 0.05	80.00 ± 0.06	80.00 ± 0.06	80.00 ± 0.06	80.00 ± 0.05