7-2020

Probabilistic Value Selection for Space Efficient Model

Gunarto Sindoro NGOO

Baihua ZHENG
Singapore Management University, bhzheng@smu.edu.sg

Kuo-Wei HSU

Wen-Chih PENG

Probabilistic Value Selection for Space Efficient Model

Gunarto Sindoro Njoo, Baihua Zheng
Living Analytics Research Centre
Singapore Management University
Singapore
gunarton@smu.edu.sg, bhzheng@smu.edu.sg

Kuo-Wei Hsu
kuowei.hsu@gmail.com

Wen-Chih Peng
Department of Computer Science
National Chiao Tung University
Hsinchu, Taiwan
wcpeng@cs.nctu.edu.tw

Abstract—An alternative to current mainstream preprocessing methods is proposed: Value Selection (VS). Unlike the existing methods such as feature selection that removes features and instance selection that eliminates instances, value selection eliminates the values (with respect to each feature) in the dataset with two purposes: reducing the model size and preserving its accuracy. Two probabilistic methods based on information theory's metric are proposed: PVS and P$^+$VS. Extensive experiments on the benchmark datasets with various sizes are elaborated. Those results are compared with the existing preprocessing methods such as feature selection, feature transformation, and instance selection methods. Experiment results show that value selection can achieve the balance between accuracy and model size reduction.

Index Terms—preprocessing, data mining, value selection, model size reduction, entropy, information theory

I. INTRODUCTION

Machine Learning is revolutionizing the Mobile App industry. For example, only 2% of iPhone users have never used Siri and just 4% of Android users have never used Google Assistant, as reported by Creative Strategies [1]. In other words, 97% of mobile users are using AI-powered voice assistants. Apple has launched Siri SDK and Core ML and now all developers can incorporate this feature into their apps. Similarly, Google has launched TensorFlow for Mobile. Many signals from the major mobile device manufacturers have also confirmed this [2]. Lenovo is working on its new smartphone that also performs without an internet connection and executes indoor geolocation and augmented reality; many mobile chip makers (including Apple, Huawei, Qualcomm, and Samsung) are working on hardware dedicated to accelerating machine learning on mobile devices.

There are many desirable advantages of enabling mobile devices to perform machine learning tasks without connecting to the servers, including but not limited to increased security and privacy, no internet connection required, lower latency and so on. However, many machine learning tasks, originally designed to run in computers, require a lot of computational power; and some model files could be huge, which incur an expensive space overhead. On the other hand, mobile devices have significantly less powerful computation capability and smaller storage space, as compared with servers or computers.

How to adapt the machine learning algorithms/models to resource-limited mobile devices is a challenging issue.

In this paper, we focus on *classification*, one of the most common machine learning tasks. To be more specific, we study compact classification methods such as Decision Tree [3], rule-based method [4], and Naive Bayesian [5] but not complex classification models such as support vector machine (SVM) [6] or neural network [7]. This is because storage spaces and computational power are still considered as limited resources in many low-end smartphones. In addition, compact classification methods are able to achieve certain accuracy and are easy to implement/interpret. For example, decision trees have been extensively used for bank loan approvals owing to their extreme transparency of rule-based decision-making. In summary, we dedicate this paper to the study of ways that could reduce the complexity of compact classification methods to cut down the storage overhead without losing much accuracy.

Reducing the storage overhead is a common requirement of many tasks. For example, in data mining, *dimension reduction* and *instance selection* are two preprocessing methods that are generally used to reduce the complexity of the data. The former is to reduce the dimensional complexity of the data/model by removing unproductive features/dimensions via *feature selection* or projecting the original feature space into a different feature space with lower dimensionality via *feature transformation*. The latter is to reduce the number of training instances to speed-up the training process, under the assumption that the whole dataset can be represented by a number of core instances without suffering much information loss.

Motivated by the strengths of both feature selection and instance selection, we explore a new preprocessing method called "value selection" in this paper. For better understanding, let's assume the data are recorded by a two-dimensional table, with the columns referring to different features and each row (or a record) corresponding to the detailed feature values of one instance. Removing a feature is equivalent to delete a complete column from the table, and removing an instance is equivalent to delete a complete row from the table. Instead of removing a complete row or a complete column, value selection adopts a finer granularity, i.e., the intersection between them, which is the values. It evaluates the importance of individual values, but not that of complete rows/columns when selecting
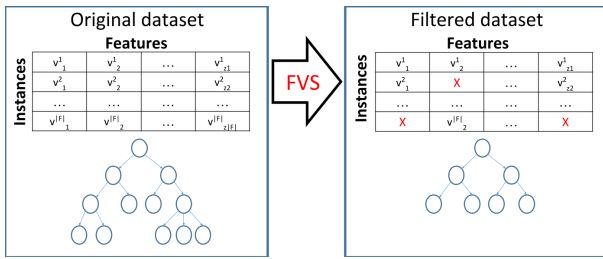
Fig. 1. Illustration of the Value Selection and how it reduces the model size. Left: original data representation. Right: filtered data representation.

data for deletion. We hypothesize that values have different importance in the training process. Value Selection enables the measurement of importance at the finest granularity (i.e., at the value level) and can remove those values that do not improve the classification accuracy. Consequently, it is expected to be able to achieve a better trade-off between the model size and accuracy.

Here, for illustration purpose, value selection is applied to the Decision Tree algorithm [3] since the rules built in the Decision Tree model are conjunctions of values, which makes value selection relevant to improve the model's performance and to reduce the model's complexity. Moreover, Decision Tree is one of the predictive models that is often used in data mining to solve both the classification task and the regression task [8]. Without losing the generality, value selection can also be implemented on other classification methods that share similar properties with Decision Tree, such as rule-based classifier [4]. In conclusion, Figure 1 shows the concept of the proposed Value Selection and illustrates how the model size could be reduced.

The contributions of this work is summarized as follows.

- A new preprocessing method called *Value Selection (VS)* is proposed. Its primary goal is to reduce the model's size without sacrificing the accuracy of the classification model.
- Two value selection methods, namely *PVS* and *P⁺VS*, are formulated based on the information theory's metrics.
- A comprehensive experimental study has been performed to compare the proposed methods with nine baselines using 10 benchmark datasets [9], [10]. The experimental results show that our methods can reduce the model size substantially and maintain acceptable accuracy.

II. RELATED WORKS

In this section, we review existing works related to *instance selection* in Section II-A and those related to *dimension reduction* in Section II-B respectively.

A. Instance Selection

Instance selection is often elaborated to minimize the number of training instances when the training computation cost is high, especially with the usage of computationally expensive classifiers such as nearest neighbor and SVM. Prior research on the instance selection mostly focuses on the classifier-specific instance selection, while the latest instance selection methods

deal with more advanced topics (e.g., unsupervised learning, online learning, and active learning).

Classifier-specific instance selection. Often, instance selection methods are tailored toward a specific classification model that requires high computational costs such as nearest neighbor, SVM, and neural network. Whereas, despite their high computational costs, those classifiers often produce satisfying results in terms of the prediction power. For the nearest neighbor method, the authors in [11] proposed a generalized CNN (Condensed Nearest Neighbor) method to shrink the training set. The generalization of CNN is performed by adding different absorption criteria: using a threshold multiplied by the minimum norm between arbitrary points. For SVM, the authors in [12] proposed two instance selection methods based on the firefly algorithm [13], [14] and edge detection method in image processing, respectively. Another work related to SVM [15] used a geometry-based approach to perform instance selection on SVM. It assumes a spherical class distribution and distributes a decision plane between spheres. Accordingly, it removes the vector points that are not distributed in the adjacent of the two hemispheres, as they are non-support vectors. For the neural network, the authors in [16] proposed an entropy-based method to perform an unsupervised instance selection. They use entropy to evaluate the information contained in each instance, i.e., instances with higher entropy tend to have more information compared to others.

General-purpose instance selection. Various general purpose instance selection methods have been proposed in the past, including randomized approach, clustering the instances, etc. The authors in [17] reported a fast randomized approach for instance selection, by using the reservoir concept. A reservoir with a predefined size is used to store the instances. The first batch of data is stored immediately on the reservoir and others are processed using a probabilistic approach. However, this method makes a strong assumption, that is each instance contains the same amount of information, and the more important instances occupy a larger part of the dataset. The authors in [18] presented three instance selection methods based on the reverse nearest neighbor (RNN) concept. The authors in [19] proposed an instance selection method using hyper-rectangle clustering [20]. Hyper-rectangles are generated for each class, and the mean of its interior instances is used as the representative instance. Recently, the authors in [21] proposed an instance selection with linear time complexity for larger dataset based on the concept of locality-sensitive hashing [22] to quickly identify the similarity between instances.

Instance selection for specific topics. Most recent efforts on the instance selection methods are more related to specific problems related to unsupervised learning, online learning, and active learning. The authors in [23] proposed a data structure called *data sphere* to summarize the data and to speed up hierarchical clustering methods. The proposed method, namely *summarized single-link method*, extends the previous work called *single-link* [24] to scale it for larger datasets and outperforms the original *single-link* by two orders of magnitude. The authors in [25] studied the problem of online

active learning, by using a binary classification to perform selective labeling in the data streams.

Discussion. The difference between existing instance selection methods and the proposed value selection methods lies in their goals: instance selection methods reduce the number of the training instances to ease the training process, whereas value selection methods reduce the classification model size. Undoubtedly, a value selection method can also be generalized as an instance selection method if all the values in an instance are removed.

B. Dimension Reduction

As stated in Section I, we can perform either feature selection or feature transformation to reduce the dimensionality of the feature space.

Feature selection. Selecting relevant features could help classification models to be more accurate and concise. Feature selection methods remove unnecessary features by using various selection criteria. In general, feature selection can be categorized into filter and wrapper approaches based on the inclusion of the classification model in the process. In the light of filter approaches, the authors in [26] built a probabilistic model according to the inconsistency criterion to eliminate the features. Differently, correlation feature selection (CFS) [27] performs feature selection on the dataset by evaluating the correlation between features and observing the predictive power of each feature. Using entropy in mind, the authors in [28] devised a feature selection method on the text data. A term distribution on different categories determines the discriminative power of the term; a term that appears in multiple categories is less discriminative than a term that appears only in one category. In the light of natural computing, the authors in [29] utilize particle swarm optimization which is often applied in a continuous search space problem to find the optimal feature subset. Wrapper feature selection is often used to increase the performance of feature selection, even though it requires longer running time (e.g., [30], [31]). The authors in [31] presented a wrapper approach with two objective functions: minimizing the number of selected features and minimizing the root mean squared error (RMSE) of the model learned by Random Forest (RF). A ten-fold cross-validation is used to minimize the risk of overfitting to certain features. Finally, a hybrid approach is presented in [30] by adapting an incremental wrapper where the features are ranked first using a filter measure and further evaluated using the wrapper approach.

Feature transformation. Feature transformation could reduce the feature complexity too. Instead of removing the feature directly as feature selection does, it projects the dataset into a different data space to reduce the dimensional complexity. Principal component analysis (PCA) [32] is one of the most famous data analysis tools that could reduce the complexity of the dataset. PCA is calculated by eigenvalue decomposition of a data covariance (or correlation) matrix or singular value decomposition of a data matrix. Random projection [33] provides an alternate solution to PCA, using a random matrix to transform the original dataset into another data space. The concept is

TABLE I
SUMMARY OF NOTATIONS

Notation	Meaning
D	Dataset matrix of $|F| \times |I|$
F	Feature set with $|F|$ indicating the number of features
I	Instance set with $|I|$ indicating the number of instances
f^x	The x-th feature / column / dimension
i_y	The y-th instance / row / record
V^x	Collection of all values of x-th feature
v_z^x	The z-th value of x-th feature
$|V^x|$	Number of possible values of x-th feature
V'^x	Filtered value set of x-th feature
L	Class label set
l	A class label $\in L$
$|M|$	Classification's model size
$|M_o|$	Original model's size
$|M_p|$	Preprocessed model's size
Acc_o	Original model's accuracy
Acc_p	Preprocessed model's accuracy
MR	Model size reduction
AR	Accuracy ratio
$H(D)$	Entropy of dataset D
$H(D|v_z^x)$	Conditional entropy of dataset D given value v_z^x
$IG(D, v_z^x)$	Information gain in the dataset given the value v_z^x
$\overline{X}$	Harmonic mean of accuracy ratio and model size reduction

built on top of Johnson-Lindenstrauss lemma [34]. That is, the distance between the projected points, which are projected onto a randomly selected subspace, is an approximation of the distance between the points in the original space. In terms of the computation time, random projection is significantly less expensive than PCA, while random projection could yield results comparable to PCA [33].

Discussion. While most feature selection methods aim to address the curse of dimensionality, they do not consider reducing the model size. Consequently, the classification model built on top of feature selection is only expected to have higher accuracy [35]. On the other hand, feature transformation methods can reduce the model size and maintain acceptable accuracy. However, they produce obscure rules and in general require a long time to compute due to the complex computation.

III. PRELIMINARIES

In this section, we first define in Section III-A the terms and notations that are frequently used in this paper, and then formulate the value selection problem in Section III-B.

A. Terms and Notations

Table I summarizes the notations used throughout this paper. Each instance $i_y \in I$ is a list of values and has a class label $l \in L$. It is important to note that an instance might have values corresponding to certain features missing. Value v_z^x is a distinct value that corresponds to a feature $f^x \in F$, which might appear once or multiple times in any instance $i_y \in I$. The values v_z^xs in each feature f^x are non-overlapping and independent to each other. Finally, to simplify the value removal process and to reduce the search space, we discretize all the continuous values into discretized values.

As our main objective is to reduce the model size without losing the accuracy, we adopt the *model size reduction* and the

accuracy ratio as the major performance metrics. For the sake of simplicity and consistency, the model size $|M|$ is measured by the total number of leaves/internal nodes throughout the paper. In other words, $|M|$ refers to the total number of rules that are used to represent the model. Without losing the generality, a rule is a path from the root to a leaf in a decision tree. The reduction of the tree size in the model represents the reduction of the model's complexity, which ultimately reduces the size of the classification model. Subsequently, model size reduction *MR* is the normalized difference between the original model size $|M_o|$ and the preprocessed model size $|M_p|$, as shown in Equation (1).

$$MR = \frac{|M_o| - |M_p|}{|M_o|} \qquad (1)$$

The range of model size reduction is $-\infty < MR < 1$ but the typical value range is between 0 and 1. A positive *MR* value indicates that the preprocessing has successfully reduced the size of the model; a zero *MR* value indicates that the preprocessing does not reduce the size of the model; and a negative *MR* value reflects that the preprocessing actually enlarges the model size, which is not desirable. In general, a larger *MR* value is more preferable.

Similarly, accuracy of a classification model might be changed when the data is preprocessed, whether it is an improvement or a deterioration. To evaluate the effectiveness of the preprocessing methods, one could evaluate the difference between the accuracy of the model built on the original data (denoted by Acc_o) and the accuracy of the model built on the preprocessed data (denoted by Acc_p). However, the absolute accuracy difference mentioned above cannot reflect the gain or the loss relatively to the original model's accuracy. Thus, a metric called *accuracy ratio (AR)* is introduced to quantify the ratio between the preprocessed model's and the original model's accuracy, as explained in Equation (2).

$$AR = \frac{Acc_p}{Acc_o} \qquad (2)$$

The range of accuracy ratio is $0 \le AR < \infty$, in which $AR = 1$ means no accuracy changes, $AR < 1$ reflects accuracy deterioration, and $AR > 1$ expresses accuracy improvement. Again, a higher *AR* is more desirable than a lower *AR*.

B. Problem Definition

Figure 2 explains how the value selection is performed in a data processing pipeline. The original dataset D needs to be discretized before the value selection stage. Two value selection methods are proposed in this paper: PVS and P$^+$VS. Both methods take advantage of the information metric of each value and apply a probabilistic approach based on the information metric's value. Details of both methods are explained in Sections IV-B and IV-C, respectively.

IV. METHODOLOGY

In this section, we first introduce an approach to evaluate the important of values for the task of classification; we then present the two proposed value selection approaches, namely
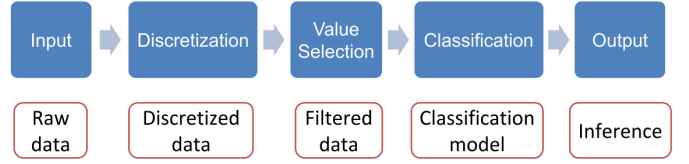

Fig. 2. Value selection in the general classification process.

PVS and P$^+$VS. Finally, we explain the effectiveness proof and intuition behind the proposed solution.

A. The Importance of Values

As explained in Section III-A, a value is the intersection of a feature and an instance. For a given feature, it might have values that are relevant to the classification and values that are irrelevant. Thus, simply removing a feature results in losing all the good values that might hurt the classification accuracy. Therefore, the proposed value selection methods aim to preserve those good values in each feature. By doing so, value selection ultimately can maintain accuracy and reduce the model size (and hence the model complexity). Naively, to perform value selection, one can simply remove values randomly. However, removal of values does not necessarily induce model size reduction in all cases, as it might introduce the overfitting problem. Thus, we introduce new metrics based on information theory to quantify the importance of each value.

The importance of each value is determined by its predictive power to deduce the class label. However, each value might possess different class distribution, which would complicate the model building process. Accordingly, we propose two types of information metrics, i.e., *entropy* and *information gain*, to discern the predictive power of each value. To ease the explanation, information metric ι is denoted to be either entropy or information gain for the values' goodness metric.

Firstly, Shannon entropy [36] is adapted to measure the values' goodness; values with lower entropy (i.e., less confusion) tend to be more useful than values that have higher entropy. The adaptation of the Shannon entropy is stated in Equation (3).

$$H(D|v_z^x) = - \sum_{\forall l \in L} p_{l,z}^x \log_{|L|} p_{l,z}^x \qquad (3)$$

Here, probability $p_{l,z}^x$ expresses the probability of class label l, given the value v_z^x observed corresponding to the feature f^x. To be more specific, given a feature f^x, there are in total $|V^x|$ different values. For each value v_z^x observed in this feature, we can count the probability of this value observed for a given class $l \in L$; and the sum of $p_{l,z}^x$ corresponding to different v_z^x values in feature f^x is one, i.e., $\sum_{v_z^x \in V_z^x} p_{l,z}^x = 1$. The entropy of any value $H(D|v_z^x)$ is in the range of 0 and 1.

Take the sample instance set listed in Table II as an example. We assume there are in total two classes, i.e., $L = \{0, 1\}$. For feature f^3, there are in total three distinct feature values, with $v_1^3 = 2$, $v_2^3 = 1$, and $v_3^3 = -1$ (i.e., $V^3 = \{2, 1, -1\}$). Now, let's derive the entropy of three different values. $H(D|v_1^3) = -p_{l=0,1}^3 \log_2 p_{l=0,1}^3 - p_{l=1,1}^3 \log_2 p_{l=1,1}^3$. As there is no instance i_y in the class $l = 0$ having its feature value in the feature f^3 being 2, $p_{l=0,1}^3 = 0$; similarly, one

instances	class label	f^1	f^2	f^3	f^4
i_1	1	-	1	2	1
i_2	1	1	-	1	1
i_3	0	-1	-2	-1	-1
i_4	0	-1	-	1	-2
i_5	1	1	1	-1	-

instance among three instances belonging to class $l = 1$ has its value corresponding to feature f^3 being 2, i.e., $p^3_{l=1,1} = \frac{1}{3}$. Accordingly, $H(D|v^3_1) = -\frac{1}{3} \log_2 \frac{1}{3} = \frac{log_2 3}{3}$. Following the same logic, we have $H(D|v^3_2) = H(D|v^3_3) = \frac{1}{2} + \frac{log_2 3}{3}$. In other words, v^3_1 is more useful in predicting the classes of instances in feature f^3, as compared with other values.

In addition to entropy, information gain [37], which is widely used in the decision tree, is applied to serve as the other goodness metric for the values. The adaptation of information gain for value selection is presented in Equation (4). Note that $H(D) = \sum_{v^x_z \in D} H(D|v^x_z)$.

$$IG(D, v^x_z) = H(D) - H(D|v^x_z) \tag{4}$$

In order to ensure the values of information gain are also in the range of 0 and 1, we introduce the normalized information gain in Equation (5). Different from the entropy, a larger information gain indicates a value with a stronger predictive power and hence is expected to be more important than a value with smaller information gain, for the task of classification.

$$IG_N(D, v^x_z) = \frac{IG(D, v^x_z)}{\max_{\forall v^x_z \in V^x} IG(D, v^x_z)} \tag{5}$$

B. Probabilistic Value Selection

The most straightforward way to perform value selection using an information metric ι (i.e., either the entropy or the information gain) is to discriminate less useful values from the dataset by using a user-defined threshold τ, i.e., removing all the values v^x_z with entropy $H(D|v^x_z) > \tau$ or with information gain $IG_N(D, v^x_z) < \tau$. However, determining a threshold for an information metric ι is not a trivial task [38]. Therefore, we adopt a probabilistic approach, instead of the threshold-based removal approach, to select the values by using ι as the probability for a value removal, e.g., values with larger entropy (smaller information gain) are more likely than those with smaller entropy (larger information gain) to be removed. This avoids the threshold selection process and eases the application of value selection in other domains. In addition, an amplifier hyperparameter ε (with $0 < \varepsilon \leq 1$) is used to intensify the value removal probability, where a small ε amplifies value selection's impact. In summary, Equation (6) defines the probability of a value v^x_z to be removed from the value set V^x. It is worth highlighting that Equation (6) unifies the two types of metrics, and a higher probability indicates a higher chance to be removed

as the underlying value has either larger entropy or smaller information gain.

$$P(V^x \setminus v^x_z) = \begin{cases} \frac{H(D|v^x_z)}{\varepsilon} & \text{if } \iota \text{ is entropy} \\ \frac{1 - IG(D, v^x_z)}{\varepsilon} & \text{if } \iota \text{ is information gain} \end{cases} \tag{6}$$

The first algorithm, PVS, is guided by the above defined removal probability. We visualize the complete process of PVS in Figure 3 to ease the understanding. We represent the original dataset using a table format, with columns corresponding to features, rows representing instances, and symbol "v" standing for a value (which could be missing). We then extract distinct values v^x_z for each feature f^x to form the respective value set V^x and then derive their probability $P(V^x \setminus v^x_z)$ corresponding to the given information metric ι. Next, we use the probability to guide the value selection process. To be more specific, for each unique value v^x_z corresponding to a feature, we utilize a random generator to produce a score r' between 0 and ε^{-1}. If r is smaller than the probability of v^x_z, value v^x_z will be removed from the respective value set V^x. As shown in the third process in Figure 3 (refer this as "Filtered Values"), the number of values in each feature might be reduced due to the value removal. Finally, we represent the dataset again using a table format, as shown in the last process of Figure 3. Each symbol "x" indicates that the value originally located at this field has been removed.

Please note that the proposed method can mimic the "feature selection" or "instance selection" process, as shown in the last step in Figure 3. To generalize value selection as an instance selection method, the number of missing value in an instance is evaluated. Note a missing value in an instance could be originally missing or removed via value selection. It is assumed that instances with more missing values are more irrelevant toward the training process, and thus, can be eliminated. Ultimately, an instance without any value is meaningless (i.e., all of the values are missing values) and can be removed without any reduction on the model's performance. Similarly, value selection can act as a feature selection when the values in a feature are completely removed, as the predictive power of that feature becomes null and thus, the feature can be safely removed. Finally, the time complexity of the PVS method is bounded by $O(n + x)$, where n and x are the number of instances and the number of values in the dataset, respectively.

C. Extension of the Probabilistic Value Selection

P^+VS is an extension of PVS by performing the value selection per instance (locally), instead of applying the selection on all instances (globally). It is motivated by the following observation. Given a value v^x_z with non-zero information gain, it is expected to have a positive impact on the model's prediction power in some cases. If it is removed, we also lose its positive impact on those cases. That is to say if we select values at feature level, removing an actually useful value has an negative impact. In order to reduce these types of negative impact, we want to enable the value selection at the instance level but not the feature level.
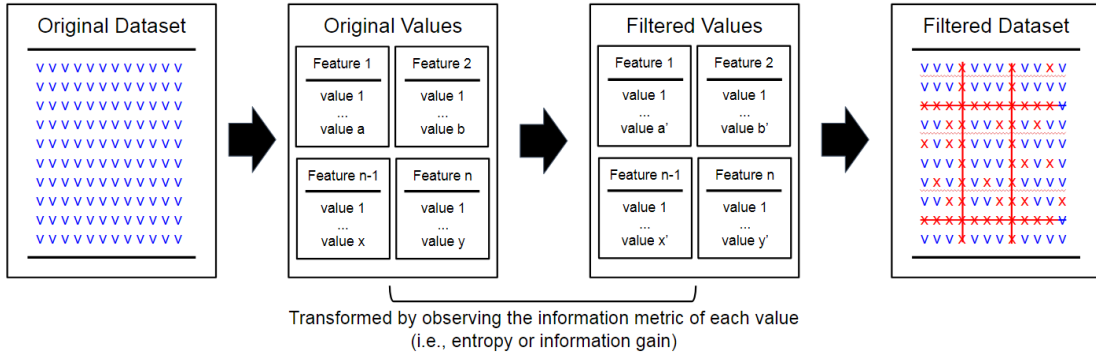
Fig. 3. Flowchart of the Probabilistic Value Selection process. The blue 'v' characters in the dataset represent the original values and the red 'x' characters denote the removed values. The horizontal red line implies that the entire row is removed (i.e., instance selection) and the vertical red line removes the column (i.e., feature selection).

The complete process of P$^+$VS is presented in Algorithm 1. It scans through all the instances in the database. For each instance i'_y, it checks all the non-missing values v^x_zs and decides whether to remove the value. The decision is purely dependent on the importance of this value (via either entropy or information gain metric) and a random score r' generated by a random process (Line 4). We simply compare the importance of this value with the generated random number, which is the same as what PVS does. However, the real removal action is different. P$^+$VS only removes the value from the current instance, and this removal will *NOT* affect other instances (Lines 5 - 8).

In addition to removing values at instance-level, P$^+$VS also introduces the action of eliminating instances. For a given instance i'_y, it derives the ratio of the number of features that instance i'_y does not have values (either removed by the previous value selection process or originally missing) to the total number of features, namely *missRate* in Line 9. Take instance i_1 in Table II as an example. Assume its values corresponding to both features f^2 and f^4 are removed; its value corresponding to feature f^1 is originally missing. Accordingly, its *missRate* is $3/4 = 0.75$. P$^+$VS generates a new random number r' and removes the whole instance i'_y if its *missRate* is larger than the random number (Lines 10 - 12). When many values of an instance is missing, it is more likely that this instance has many high-uncertain/less-informative values. In other words, this instance has a higher chance to be noisy. We expect this instance-level removal could improve the capability of value selection to remove unproductive instances and hence contributes to a more accurate classification performance.

D. Value Selection effectiveness proof

Hypothesis. Value selection will reduce, but will not increase, the confusion of the dataset by removing confusing values. We assume the proportion of each value v^x_z before and after the value selection in feature f^x is represented by w^x_z and $\widetilde{w}^x_z$, respectively, and the entropy of each respective value is described by $H(D|v^x_z)$.

Proposition. Total confusion (i.e., weighted sum of the entropy) post-value selection is no greater than that of the original data. In other words, we need to prove the following inequality is

---

Algorithm 1: P$^+$VS method.

Input: Original instances: I, value set: V, amplifier ε, information metric ι

Output: Filtered instances: I'

1   Define $I' \leftarrow I$;
2   **foreach** i'_y in I' **do**
3   **foreach** v^x_z in i'_y **do**
4       $r' \leftarrow random(0,1)$;
5   **if** $\iota{=}infoGain$ and $IG_N(D,v^x_z) < r' * \varepsilon$ **then**
6   i'_y.remove(v);
7   **else if** $\iota{=}entropy$ and $H(D|v^x_z) > r' * \varepsilon$ **then**
8   i'_y.remove(v^x_z);
9     $missRate \leftarrow countMiss(i'_y)$;
10    $r' \leftarrow random(0,1)$;
11   **if** $missRate > r'$ **then**
12   I'.remove(i'_y);
13   **return** I'

---

correct.

$$\sum_{x=1}^{|F|} \sum_{v=1}^{|V^x|} \widetilde{w}^x_z H(D|v^x_z) \leq \sum_{x=1}^{|F|} \sum_{v=1}^{|V^x|} w^x_z H(D|v^x_z) \qquad (7)$$

Proof.

$$\widetilde{w}^x_z = \begin{cases} 0 & \text{if } H(D|v^x_z) = 1 \\ w^x_z \times (1 - H(D|v^x_z)) & \text{otherwise} \end{cases}$$

Since, $w^x_z \times (1 - H(D|v^x_z))$ is always larger than 0, we have

$$\sum_{x=1}^{|F|} \sum_{v=1}^{|V^x|} \widetilde{w}^x_z H(D|v^x_z) = \sum_{x=1}^{|F|} \sum_{v=1}^{|V^x|} w^x_z (1 - H(D|v^x_z)) H(D|v^x_z)$$

$$= \sum_{x=1}^{|F|} \sum_{v=1}^{|V^x|} w^x_z H(D|v^x_z) - \sum_{x=1}^{|F|} \sum_{v=1}^{|V^x|} w^x_z H^2(D|v^x_z)$$

Accordingly, we can restate Inequality (7) as follows.

$$\sum_{x=1}^{|F|} \sum_{v=1}^{|V^x|} w^x_z H(D|v^x_z) - \sum_{x=1}^{|F|} \sum_{v=1}^{|V^x|} \widetilde{w}^x_z H(D|v^x_z)$$

$$= \sum_{x=1}^{|F|} \sum_{v=1}^{|V^x|} w^x_z H(D|v^x_z) - \sum_{x=1}^{|F|} \sum_{v=1}^{|V^x|} w^x_z H(D|v^x_z) + \sum_{x=1}^{|F|} \sum_{v=1}^{|V^x|} w^x_z H^2(D|v^x_z))$$

$$= \sum_{x=1}^{|F|} \sum_{v=1}^{|V^x|} w^x_z H^2(D|v^x_z) \geq 0$$

Our proof completes.

∎

V. Experiments and Discussions

In order to evaluate the performance of proposed approaches, we have conducted a comprehensive experimental study. In the following, we present the experiment setups, study the impact of hyper-parameters before we determine the exact setups, and then report our major findings.

A. Experiment Settings

Datasets. In our study, we use 10 benchmark datasets from various domains [9], [10] with varying size in terms of the number of features, the number of instances, the possibility of missing values in the data, and the number of class labels. Table III reports the basic statistics of the datasets.

Algorithms. In order to evaluate the performance and effectiveness of proposed PVS and P+VS, we implement in total nine representatives of *feature selection (FS)*, *feature transformation (FT)*, and *instance selection (IS)* as competitors/baselines. They are i) FS_CFS [27], a feature selection method that picks a set of useful features based on an evaluation formula with an appropriate correlation measure and a heuristic search strategy; ii) FS_Consistency [26], a filter solution for feature selection using inconsistency metric and a probabilistic approach; iii) FS_IWSS [30], an incremental wrapper feature selection that first sorts features using a filter approach and evaluates them using a wrapper approach; iv) FS_MOEA [31], a wrapper feature selection that employ genetic algorithm with two objectives: minimizing number of features and RMSE of Random Forests; v) FS_PSO [29], a feature selection that uses particle swarm optimization to find the optimal subset; vi) FT_RandomProjection or FT_RP for short [33], a method that reduces the dimensionality of data by using a random projection from the original data space; vii) FT_PCA [32], a data transformation method that projects the original dataset into a set of values of linearly uncorrelated label called principal components; viii) IS_Misclassified [39], an instance selection method implemented in Weka[1] that filters out instances that often mis-classify the class labels; and ix) IS_Reservoir [17], a fast instance selection method that selects the instances using random sampling without replacement method (i.e., each instance in the dataset is stored into a set with a limited size using a probabilistic approach). Note the prefix (i.e., FS or FT or IS) indicates the category of the baseline

Parameters. The parameter settings of different algorithms are explained as follows. The number of principal components in the *FT_PCA* is 1/2 of the number of features in the dataset and 95% of variance is ensured to be in the original data. Similarly, the number of projected dimensions in the *FT_RP* is 1/2 of the number of features in the dataset. It is assumed that by reducing the number of features into a half of the original number of features, both the model size reduction and accuracy can be simultaneously high. The size of reservoir in *IS_Reservoir* is set to be 1/20 of the number of instances in the dataset. By sampling more instances from the dataset, the accuracy ratio is not necessarily higher than the case with
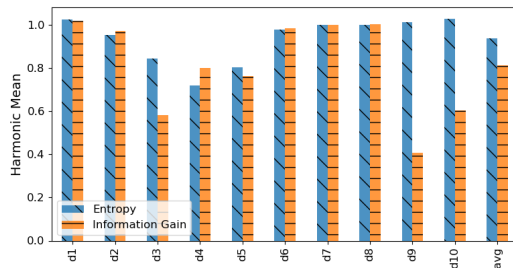
[1]http://www.cs.waikato.ac.nz/ml/weka/

Fig. 4. Value selection (P+VS) performances under different information metric ι settings. Each point represents the performance on each dataset.

fewer instances. Additionally, the original instances set in Weka consists of only 100 instances, which does not favor bigger datasets. Therefore, proportionally sampling 5% of the dataset is more effective than fixing the size of the reservoir as it is difficult to determine a *magic number* that works for all datasets. The other preprocessing methods are implemented using the default settings in Weka.

Setups. All the experiments are conducted on a server with Intel Core i7-4790 running at 3.60GHz, 28 GB RAM, Windows 7, Java 9.0.1, and Weka 3.8.1. Value selection is implemented as a Weka's Filter class and Weka's classification methods are utilized to run the experiments. Our experiments can be divided into three stages namely *discretization*, *value selection*, and *classification*, as shown in Figure 2. We evaluate three different discretization methods throughout our experiments to observe the robustness of the proposed methods under both unsupervised and supervised discretization methods, including *equal-width binning (Binning)*, *equal-frequency binning (Frequency)*, and *minimum description length (MDL)* discretization [40]. For the sake of consistency, we implement all three discretization methods in Weka. Subsequently, we produce filtered data from the preprocessing of the discretized data. To cope with the randomness of produced results in value selection, we repeat the experiments at random for five times. Finally, we construct the classification model using the filtered data and apply the 10-fold cross-validation scheme to evaluate the model.

B. Parameter Analysis

The performance of both PVS and P+VS is dependent on several hyper-parameters, including the selected information metric ι, the discretization method, and an amplifier ε. In our last set of experiments, we study their impact on the performance of newly proposed algorithms. Table IV reports the settings of these three hyper-parameters, with values in bold indicating the default values. When we evaluate the impact of one hyper-parameter, we set the other two hyper-parameters to their defaults. Because of the space limitation, we only report their impacts on P+VS, as those hyper-parameters have similar impacts on PVS. Also, borrowing the idea of F1 score that summarizes precision and recall, we define a *harmonic mean* $\overline{X}$ of MR and AR as $\overline{X} = \frac{2}{\frac{1}{AR} + \frac{1}{MR}} = \frac{2 \times AR \times MR}{AR + MR}$.

First, the impact of the information metric on P+VS is reported in Figure 4. While both entropy and information

TABLE III
THE BASIC STATISTICS OF THE 10 BENCHMARK DATASETS USED IN THIS PAPER.

ID	Dataset name	#Instances	#Features	#Numerical	#Categorical	#Class	Missing values
d1	credit	1,000	20	7	13	2	NO
d2	hypothyroid	3,772	29	7	22	4	YES
d3	mfeat-zernike	2,000	47	47	0	10	NO
d4	segment-challenge	1,500	19	19	0	7	NO
d5	letter	20,000	16	16	0	26	NO
d6	adult	48,855	14	6	8	2	NO
d7	census-income	199,504	41	13	28	2	NO
d8	dota2	92,641	116	0	116	2	YES
d9	cifar10-small	10,000	1,024	1,024	0	10	NO
d10	cifar10-big	60,000	1,024	1,024	0	10	NO

TABLE IV
HYPERPARAMETERS

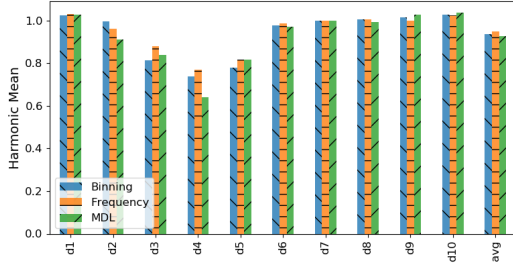| information metric ι	**entropy**, information gain
discretization method	Binning, **Frequency**, MDL
amplifier ε	0.1, 0.2, 0.3, 0.4, **0.5**, 0.6, 0.7, 0.8, 0.9, 1.0

Fig. 5. Value selection (P$^+$VS) performances under different discretization settings. Each point represents the performance on each dataset.

Fig. 6. The impact of parameter ε on the value selection methods (average performance on 10 datasets).

gain are commonly employed as measurements to evaluate the importance of data in the field of information theory, it seems that entropy is more useful to depict the usefulness of each value v_z^x than information gain in our study. Second, we evaluate the impact of discretization methods and report the results in Figure 5. There are no significant differences among three discretization methods, which demonstrates the robustness of the proposed value selection methods on particular discretization methods. Last but not least, we report the impact of amplifier ε in Figure 6. As observed, it has opposite impacts on MR and AR. A small ε favors MR but not AR, while the increase of ε improves AR but not MR. The best trade-off between MR and AR is achieved when $\varepsilon = 0.5$ (i.e., highest harmonic mean $\overline{X}$).

C. Overall Results

Our algorithms are designed to reduce the model sizes without hurting the accuracy achieved by the models. Consequently, we care both model size reduction (MR) and accuracy ratio (AR). The results in terms of $\overline{X}$ of all the algorithms are reported in Figure 7. Table V and Table VI report the detailed accuracy ratio and model size reduction of different algorithms on each dataset, respectively. To ease the understanding of the data, we underline the best performers corresponding to each dataset. Note that for both PVS and P$^+$VS, we report their performance with hyper-parameters set to their default (as listed in Table IV).
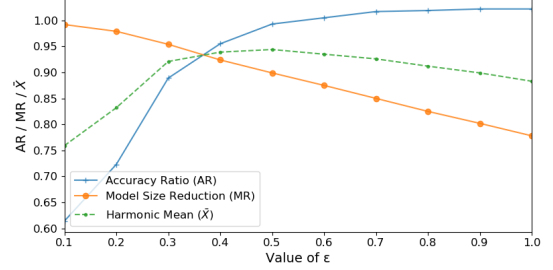
We observe that our algorithms achieve good performance. To be more specific, P$^+$VS achieves the best trade-off between AR and MR. If we take a closer look at its performance in different datasets, P$^+$VS achieves the best AR performance in 2 out of 10 datasets. In the other 8 datasets, its AR performance is also comparable with the best performer, on average 6% lower than the best performer. Its MR performance is even better. It tops in 6 out of 10 datasets.

In terms of the comparison among nine competitors, we have several observations. Feature selection algorithms (i.e., FS_CFS, FS_Consistency, FS_IWSS, FS_MOEA, FS_PSO) are designed to reduce the dimensionality of data and meanwhile to improve the model's performance. They indeed maintain impressive accuracy ratio. They are the best performer in terms of accuracy in 8 out of 10 datasets. However, they are not good in overall model size reduction, especially when the number of features is extremely large (e.g., d9, d10). On the other hand, IS_Reservoir reduces the model size most but it suffers from unstable accuracy. It is worth noting that IS_Reservoir drops its accuracy significantly in several datasets (e.g., d3, d4, d5, and d9). Meanwhile, we also observe that IS_Misclassified, although being a representative of instance selection, performs more similarly to feature selection, because its priority is to remove mis-classified instances and hence focuses more on model's classification accuracy. FT_PCA, as a feature transformation algorithm, tries to achieve a balance between accuracy ratio and model size reduction through the dataset transformation. Although it has not been the best performer in terms of AR or MR in any of the datasets, it does outperform many of the competitors in terms of the harmonic mean $\overline{X}$. Note we do not report the performance of FT_PCA
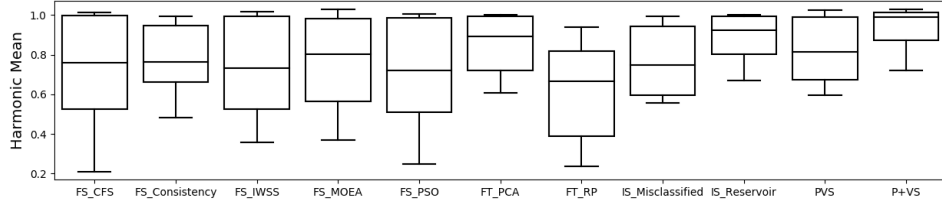
Fig. 7. Harmonic mean of accuracy ratio and model size reduction of the proposed methods along with the baselines.

TABLE V
SUMMARY OF ACCURACY RATIO ON EACH DATASET

Algorithms	d1	d2	d3	d4	d5	d6	d7	d8	d9	d10	Average
FS_CFS	1.07	0.99	1.02	<u>1.00</u>	1.00	1.01	<u>1.01</u>	<u>1.07</u>	1.01	1.01	1.02
FS_Consistency	1.07	<u>1.00</u>	0.96	<u>1.00</u>	0.99	<u>1.02</u>	<u>1.01</u>	1.02	1.03	1.03	1.01
FS_IWSS	<u>1.10</u>	<u>1.00</u>	<u>1.03</u>	0.99	1.00	1.01	<u>1.01</u>	<u>1.07</u>	1.03	1.03	1.03
FS_MOEA	1.07	<u>1.00</u>	<u>1.03</u>	<u>1.00</u>	1.00	<u>1.02</u>	<u>1.01</u>	<u>1.07</u>	1.03	N/A	1.03
FS_PSO	1.09	<u>1.00</u>	1.01	<u>1.00</u>	<u>1.01</u>	<u>1.02</u>	<u>1.01</u>	<u>1.07</u>	1.00	1.02	1.02
FT_PCA	1.04	0.98	0.98	0.91	0.71	1.00	1.00	1.02	N/A	N/A	0.95
FT_RandomProjection	0.98	0.98	0.42	0.72	0.39	0.96	0.99	0.98	0.77	0.70	0.79
IS_Misclassified	1.07	<u>1.00</u>	1.02	<u>1.00</u>	0.99	<u>1.02</u>	<u>1.01</u>	1.04	<u>1.07</u>	1.07	1.03
IS_Reservoir	1.02	0.97	0.52	0.73	0.68	1.01	1.00	1.01	0.79	0.83	0.86
PVS	1.06	0.97	0.96	0.94	0.84	0.95	<u>1.01</u>	1.02	0.96	0.97	0.97
P$^+$VS	1.06	0.96	0.90	0.92	0.91	0.97	1.00	1.00	<u>1.07</u>	<u>1.08</u>	0.99

TABLE VI
SUMMARY OF MODEL SIZE REDUCTION ON EACH DATASET

Algorithms	d1	d2	d3	d4	d5	d6	d7	d8	d9	d10	Average
FS_CFS	0.96	0.81	0.4	0.46	0.34	<u>0.99</u>	<u>0.99</u>	0.96	0.12	0.14	0.62
FS_Consistency	0.93	0.71	0.5	0.39	0.32	0.96	0.98	0.64	0.51	0.58	0.65
FS_IWSS	0.94	0.75	0.42	0.43	0.34	0.95	<u>0.99</u>	0.96	0.28	0.22	0.63
FS_MOEA	0.89	0.7	0.48	0.39	0.32	0.95	0.97	<u>0.99</u>	0.23	N/A	0.66
FS_PSO	0.9	0.7	0.44	0.4	0.32	0.94	0.98	0.95	0.14	0.15	0.59
FT_PCA	0.97	0.44	0.67	0.65	0.53	0.98	0.98	<u>0.99</u>	N/A	N/A	0.78
FT_RandomProjection	0.75	0.24	0.36	0.14	0.28	0.9	0.89	0.59	0.59	0.64	0.54
IS_Misclassified	0.86	0.71	0.47	0.38	0.4	0.95	0.98	0.81	0.48	0.4	0.64
IS_Reservoir	0.98	<u>0.96</u>	<u>0.93</u>	<u>0.85</u>	<u>0.92</u>	<u>0.99</u>	<u>0.99</u>	0.97	0.94	0.95	0.95
PVS	<u>0.99</u>	0.92	0.53	0.47	0.46	<u>0.99</u>	<u>0.99</u>	0.98	0.52	0.52	0.74
P$^+$VS	<u>0.99</u>	0.95	0.8	0.59	0.72	<u>0.99</u>	<u>0.99</u>	<u>0.99</u>	<u>0.97</u>	<u>0.99</u>	0.90

under datasets d9 and d10 (and FS_MOEA on d10). This is because, under those two datasets, the memory usage of FT_PCA (and FS_MOEA) exceeds the capacity and its running time is extremely long. Finally, another feature transformation algorithm, FT_RandomProjection, is faster than FT_PCA and presents acceptable results in certain datasets but it fails to achieve overall good results because unlike FT_PCA, it is not optimized and solely relies on randomization.

VI. CONCLUSIONS AND FUTURE WORKS

A new alternative to the preprocessing methods in data mining is proposed by removing irrelevant values on the dataset. The purpose of the proposed method, value selection, is reducing the model size and maintaining an acceptable accuracy ratio. Two probabilistic methods are presented to solve the value selection problem: PVS and P$^+$VS. The former removes the values globally over all instances while the latter applies the removal locally on each instance. Moreover, P$^+$VS can act as an instance selection by using the ratio of missing values in an instance. Experiment results show that the proposed methods are effective in reducing the model size (59%-99% reduction) and simultaneously maintaining the accuracy ratio above 90% on average. Furthermore, the proposed methods perform in a linear time complexity.

The direction of the future works is given as follows. Firstly, different heuristic metrics could be explored to generate the value subset. In addition, combining value selection with other data reduction methods could also be a good direction for inventing a better space efficient model. Tackling on more sophisticated problems, value selection methods designed for online learning and semi-supervised learning are also an interesting direction.

REFERENCES

[1] C. Milanesi, "Voice assistant anyone? yes please, but not in public!" 2016. [Online]. Available: https://creativestrategies.com/voice-assistant-anyone-yes-please-but-not-in-public

[2] S. Relevant!, "Why use machine learning on mobile devices?" 2019. [Online]. Available: https://www.codementor.io/packt/why-use-machine-learning-on-mobile-devices-tiopoukck

[3] J. R. Quinlan, *C4.5: Programs for Machine Learning.* San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993.

[4] W. W. Cohen, "Fast effective rule induction," in *Proceedings of the Twelfth International Conference on International Conference on Machine Learning*, ser. ICML'95. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1995, pp. 115–123. [Online]. Available: http://dl.acm.org/citation.cfm?id=3091622.3091637

[5] D. J. Hand and K. Yu, "Idiot's bayes—not so stupid after all?" *International statistical review*, vol. 69, no. 3, pp. 385–398, 2001.

[6] J. C. Platt, "Advances in kernel methods," B. Schölkopf, C. J. C. Burges, and A. J. Smola, Eds. Cambridge, MA, USA: MIT Press, 1999, ch. Fast Training of Support Vector Machines Using Sequential Minimal Optimization, pp. 185–208. [Online]. Available: http://dl.acm.org/citation.cfm?id=299094.299105

[7] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Parallel distributed processing: Explorations in the microstructure of cognition, vol. 1," D. E. Rumelhart, J. L. McClelland, and C. PDP Research Group, Eds. Cambridge, MA, USA: MIT Press, 1986, ch. Learning Internal Representations by Error Propagation, pp. 318–362. [Online]. Available: http://dl.acm.org/citation.cfm?id=104279.104293

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

[9] D. Dheeru and E. Karra Taniskidou, "UCI machine learning repository," 2017. [Online]. Available: http://archive.ics.uci.edu/ml

[10] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," *Master's thesis, Department of Computer Science, University of Toronto*, 2009.

[11] C.-H. Chou, B.-H. Kuo, and F. Chang, "The generalized condensed nearest neighbor rule as a data reduction method," in *Proceedings of the 18th International Conference on Pattern Recognition - Volume 02*, ser. ICPR '06. Washington, DC, USA: IEEE Computer Society, 2006, pp. 556–559. [Online]. Available: http://dx.doi.org/10.1109/ICPR.2006.1119

[12] A. A. Akinyelu and A. O. Adewumi, "Improved instance selection methods for support vector machine speed optimization," *Security and Communication Networks*, vol. 2017, 2017.

[13] X.-S. Yang, *Nature-Inspired Metaheuristic Algorithms.* Luniver Press, 2008.

[14] X.-S. Yang and X. He, "Firefly algorithm: recent advances and applications," *International Journal of Swarm Intelligence*, vol. 1, no. 1, pp. 36–50, 2013.

[15] C. Liu, W. Wang, M. Wang, F. Lv, and M. Konan, "An efficient instance selection algorithm to reconstruct training set for support vector machine," *Knowledge-Based Systems*, vol. 116, pp. 58 – 73, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0950705116304257

[16] P. M. Ferreira, "Unsupervised entropy-based selection of data sets for improved model fitting," in *2016 International Joint Conference on Neural Networks (IJCNN)*, July 2016, pp. 3330–3337.

[17] J. S. Vitter, "Random sampling with a reservoir," *ACM Trans. Math. Softw.*, vol. 11, no. 1, pp. 37–57, Mar. 1985. [Online]. Available: http://doi.acm.org/10.1145/3147.3165

[18] B.-R. Dai and S.-M. Hsu, "An instance selection algorithm based on reverse nearest neighbor," *Advances in Knowledge Discovery and Data Mining*, pp. 1–12, 2011.

[19] J. Hamidzadeh, R. Monsefi, and H. S. Yazdi, "Irahc: Instance reduction algorithm using hyperrectangle clustering," *Pattern Recognition*, vol. 48, no. 5, pp. 1878 – 1889, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0031320314004555

[20] S. Salzberg, "A nearest hyperrectangle learning method," *Mach. Learn.*, vol. 6, no. 3, pp. 251–276, May 1991. [Online]. Available: http://dx.doi.org/10.1023/A:1022661727670

[21] Álvar Arnaiz-González, J.-F. Díez-Pastor, J. J. Rodríguez, and C. García-Osorio, "Instance selection of linear complexity for big data," *Knowledge-Based Systems*, vol. 107, pp. 83 – 95, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0950705116301617

[22] L. Paulevé, H. Jégou, and L. Amsaleg, "Locality sensitive hashing: A comparison of hash function types and querying mechanisms," *Pattern Recogn. Lett.*, vol. 31, no. 11, pp. 1348–1358, Aug. 2010. [Online]. Available: http://dx.doi.org/10.1016/j.patrec.2010.04.004

[23] B. K. Patra and S. Nandi, "Effective data summarization for hierarchical clustering in large datasets," *Knowledge and Information Systems*, vol. 42, no. 1, pp. 1–20, 2015. [Online]. Available: http://dx.doi.org/10.1007/s10115-013-0709-8

[24] P. H. Sneath, R. R. Sokal *et al.*, *Numerical taxonomy. The principles and practice of numerical classification.*, 1973.

[25] W. Chu, M. Zinkevich, L. Li, A. Thomas, and B. Tseng, "Unbiased online active learning in data streams," in *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '11. New York, NY, USA: ACM, 2011, pp. 195–203. [Online]. Available: http://doi.acm.org/10.1145/2020408.2020444

[26] H. Liu and R. Setiono, "A probabilistic approach to feature selection - a filter solution," in *13th International Conference on Machine Learning*, 1996, pp. 319–327.

[27] M. A. Hall, "Correlation-based feature subset selection for machine learning," Ph.D. dissertation, University of Waikato, Hamilton, New Zealand, 1998.

[28] C. Largeron, C. Moulin, and M. Géry, "Entropy based feature selection for text categorization," in *Proceedings of the 2011 ACM Symposium on Applied Computing*, ser. SAC '11. New York, NY, USA: ACM, 2011, pp. 924–928. [Online]. Available: http://doi.acm.org/10.1145/1982185.1982389

[29] A. Moraglio, C. Di Chio, and R. Poli, "Geometric particle swarm optimisation," in *Genetic Programming*, M. Ebner, M. O'Neill, A. Ekárt, L. Vanneschi, and A. I. Esparcia-Alcázar, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 125–136.

[30] P. Bermejo, J. A. Gamez, and J. M. Puerta, "Incremental wrapper-based subset selection with replacement: An advantageous alternative to sequential forward selection," in *2009 IEEE Symposium on Computational Intelligence and Data Mining*, March 2009, pp. 367–374.

[31] F. Jiménez, G. Sánchez, J. García, G. Sciavicco, and L. Miralles, "Multi-objective evolutionary feature selection for online sales forecasting," *Neurocomputing*, vol. 234, pp. 75 – 92, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0925231216315612

[32] K. Pearson, "Liii. on lines and planes of closest fit to systems of points in space," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 2, no. 11, pp. 559–572, 1901.

[33] E. Bingham and H. Mannila, "Random projection in dimensionality reduction: Applications to image and text data," in *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '01. New York, NY, USA: ACM, 2001, pp. 245–250. [Online]. Available: http://doi.acm.org/10.1145/502512.502546

[34] W. B. Johnson and J. Lindenstrauss, "Extensions of lipschitz mappings into a hilbert space," *Contemporary mathematics*, vol. 26, no. 189-206, p. 1, 1984.

[35] M. Dash and H. Liu, "Feature selection for classification," *Intelligent Data Analysis*, vol. 1, no. 1, pp. 131 – 156, 1997. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1088467X97000085

[36] C. Shannon, "A mathematical theory of communication," *Bell System Technical Journal, The*, vol. 27, no. 3, pp. 379–423, July 1948.

[37] J. R. Quinlan, "Induction of decision trees," *Mach. Learn.*, vol. 1, no. 1, pp. 81–106, Mar. 1986. [Online]. Available: http://dx.doi.org/10.1023/A:1022643204877

[38] G. S. Njoo, Y. Peng, K. Hsu, and W. Peng, "On selecting feature-value pairs on smart phones for activity inferences," in *International Conference on Data Science and Advanced Analytics, DSAA 2014, Shanghai, China, October 30 - November 1, 2014*, 2014, pp. 319–325.

[39] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: an update," *ACM SIGKDD explorations newsletter*, vol. 11, no. 1, pp. 10–18, 2009.

[40] U. M. Fayyad and K. B. Irani, "Multi-interval discretization of continuous-valued attributes for classification learning," in *Thirteenth International Joint Conference on Artical Intelligence*, vol. 2. Morgan Kaufmann Publishers, 1993, pp. 1022–1027.