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].

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.

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.
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 are 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 [25] built a probabilistic model according to the inconsistency criterion to eliminate the features. Differently, correlation feature selection (CFS) [26] 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 [27] 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 [28] 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 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 $i \in I$ 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 $v_i$ is a distinct value that corresponds to a feature $f_i \in F$, which might appear once or multiple times in any instance $i \in I$. The values $v_i$s in each feature $f_i$ 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

| 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 |
| $f^i$    | The $x$-th feature / column / dimension |
| $i^o$    | The $y$-th instance / row / record |
| $V^i$    | Collection of all values of $x$-th feature |
| $v^i_z$  | The $z$-th value of $x$-th feature |
| $|V^i|$   | Number of possible values of $x$-th feature |
| $V^i_{\text{fs}}$ | Filtered value set of $x$-th feature |
| $L$      | Class label set |
| $l$      | A class label $l \in L$ |
| $|M|$     | Classification’s model size |
| $M_0$    | Original model’s size |
| $M_{\text{ps}}$ | Preprocessed model’s size |
| $Acc_o$  | Original model’s accuracy |
| $Acc_{\text{ps}}$ | Preprocessed model’s accuracy |
| $MR$     | Model size reduction |
| $AR$     | Accuracy ratio |
| $H(D)$   | Entropy of dataset $D$ |
| $H(D|v^i_z)$ | Conditional entropy of dataset $D$ given value $v^i_z$ |
| $IG(D, v^i_z)$ | Information gain in the dataset given the value $v^i_z$ |
| $\bar{X}$ | Harmonic mean of accuracy ratio and model size reduction |
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).

$$MR = \frac{|M_o| - |M_p|}{|M_o|}$$

(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).

$$AR = \frac{Acc_p}{Acc_o}$$

(2)

The range of accuracy ratio is $0 \leq AR < \infty$, 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 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 $I$ 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).

$$H(D|v^+_l) = -\sum_{v \in L} p^+_l \log_2 p^+_l$$

(3)

Here, probability $p^+_l$ expresses the probability of class label $l$ given the value $v^+_l$ 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 $v^+_l$ observed in this feature, we can count the probability of this value observed for a given class $l \in L$; and the sum of $p^+_l$ corresponding to different $v^+_l$ values in feature $f^x$ is one, i.e., $\sum_{v \in V^x} p^+_l = 1$. The entropy of any value $H(D|v^+_l)$ is in the range of 0 and 1.

Take the sample instance set listed in Table III 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 $v^+_2 = 2$, $v^+_3 = 1$, and $v^+_3 = -1$ (i.e., $V^3 = \{2, 1, -1\}$). Now, let’s derive the entropy of three different values. $H(D|v^+_2) = -p^2_{l=0,1} \log_2 p^2_{l=0} - p^2_{l=1,1} \log_2 p^2_{l=1}$, as there is no instance $i^+_2$, in the class $l = 0$ having its feature value in the feature $f^3$ being 2, $p^3_{l=0,1} = 0$ and $p^3_{l=1,1} = 1$ because all the instances (i.e.,

![Fig. 2. Value selection in the general classification process.](image)
one instance) belong to class $l = 1$. Accordingly, $H(D|v^3_1) = -0 \log_2 0 - 1 \log_2 1 = 0$. Following the same logic, we have $H(D|v^3_2) = H(D|v^3_3) = -\frac{1}{3} \log_2 \frac{1}{3} - \frac{2}{3} \log_2 \frac{2}{3}$ and $H(D|v^3_3) = 0$.

In other words, $v^3_1$ and $v^3_3$ are more useful in predicting the classes of instances in feature $f^3$, as compared to $v^3_2$.

In addition to entropy, information gain (\cite{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 $H(D) = \sum_{v \in D} H(D|v^x)$.

\[
IG(D, v^x) = H(D) - H(D|v^x)
\]

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.

\[
IG_N(D, v^x) = \frac{IG(D, v^x)}{\max_{v \in D} IG(D, v^x)}
\]

### B. Probabilistic Value Selection

The most straightforward way to perform value selection using an information metric $t$ (i.e., either the entropy or the information gain) is to discriminate less useful values from the dataset by using a user-defined threshold $\tau$, i.e., removing all the values $v^x$ with entropy $H(D|v^x) > \tau$ or with information gain $IG_N(D, v^x) < \tau$. However, determining a threshold for an information metric $t$ is not a trivial task (\cite{38}). Therefore, we adopt a probabilistic approach, instead of the threshold-based removal approach, to select the values by using $t$ 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 $\varepsilon$ (with $0 < \varepsilon \leq 1$) is used to intensify the value removal probability, where a small $\varepsilon$ amplifies value selection’s impact. In summary, Equation (6) defines the probability of a value $v^x$ 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.

\[
P(V^x \setminus v^x_t) = \begin{cases} 
\frac{H(D|v^x_t)}{1 - IG_N(D, v^x)} & \text{if } t \text{ is entropy} \\
\frac{1}{\varepsilon} & \text{if } t \text{ is information gain}
\end{cases}
\]

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 “\text{v}” standing for a value (which could be missing). We then extract distinct values $v^x_t$ for each feature $f^x$ to form the respective value set $V^x$ and then derive their probability $P(V^x \setminus v^x_t)$ corresponding to the given information metric $t$. Next, we use the probability to guide the value selection process. To be more specific, for each unique value $v^x_t$ corresponding to a feature, we utilize a random generator to produce a score $r'$ between 0 and $\varepsilon^{-1}$. If $r$ is smaller than the probability of $v^x_t$, value $v^x_t$ 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 “\text{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

$P^+\text{VS}$ 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 $v^x_t$ 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.
We assume the proportion of each value will NOT (Line 4). We simply compare the importance of this value with (i.e., feature selection).

Post-value selection is no greater than that of the original data. H described by the value selection in feature f the confusion of the dataset by removing confusing values.

D. Value Selection effectiveness proof

Contributes to a more accurate classification performance. Of value selection to remove unproductive instances and hence instance has many high-uncertain/less-informative values. In number its corresponding to feature f instance introduces the action of eliminating instances. For a given number + 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 – 9).

In addition to removing values at instance-level, P+VS also introduces the action of eliminating instances. For a given instance i, it derives the ratio of the number of features that instance i 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 2. Take instance i in Table 1 as an example. Assume its values corresponding to both features f1 and f4 are removed; its value corresponding to feature f2 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, 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 before and after the value selection in feature f is represented by \( \tilde{w}_v \) and \( w_v \), respectively, and the entropy of each respective value is described by \( H(D|v) \).

Proposition. Total confusion (i.e., weighted sum of the entropy) post-value selection is no greater than that of the original data.

\[
\sum_{x=1}^{[F]} \sum_{v=1}^{[V]} \tilde{w}_v H(D|v) \leq \sum_{x=1}^{[F]} \sum_{v=1}^{[V]} w_v H(D|v) \tag{7}
\]

Proof.\[
\tilde{w}_v = \begin{cases} 0 & \text{if } H(D|v) = 1 \\ w_v \times (1 - H(D|v)) & \text{otherwise} \end{cases}
\]

Since, \( w_v \times (1 - H(D|v)) \) is always larger than 0, we have

\[
\sum_{x=1}^{[F]} \sum_{v=1}^{[V]} \tilde{w}_v H(D|v) = \sum_{x=1}^{[F]} \sum_{v=1}^{[V]} w_v (1 - H(D|v)) H(D|v) = \sum_{x=1}^{[F]} \sum_{v=1}^{[V]} w_v H(D|v) - \sum_{x=1}^{[F]} \sum_{v=1}^{[V]} w_v H^2(D|v)
\]

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

\[
\sum_{x=1}^{[F]} \sum_{v=1}^{[V]} w_v H(D|v) - \sum_{x=1}^{[F]} \sum_{v=1}^{[V]} w_v H^2(D|v)
\]

\[
= \sum_{x=1}^{[F]} \sum_{v=1}^{[V]} w_v H(D|v) - \sum_{x=1}^{[F]} \sum_{v=1}^{[V]} w_v H(D|v) + \sum_{x=1}^{[F]} \sum_{v=1}^{[V]} w_v H^2(D|v)
\]

\[
= \sum_{x=1}^{[F]} \sum_{v=1}^{[V]} w_v H^2(D|v) \geq 0
\]

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 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 \(\tau\), the discretization method, and an amplifier \(\varepsilon\). 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 \(X\) of MR and AR as
\[
X = \frac{2 \times AR \times MR}{AR + MR}.
\]

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            | 25     | 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 | Entropy, Information gain |
|--------------------|---------------------------|
| Discretization method | Binning, Frequency, MDL |
| Amplifier $\epsilon$ | 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.

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 $v_i^k$ 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 $\epsilon$ in Figure 6. As observed, it has opposite impacts on MR and AR. A small $\epsilon$ favors MR but not AR, while the increase of $\epsilon$ improves AR but not MR. The best trade-off between MR and AR is achieved when $\epsilon = 0.5$ (i.e., highest harmonic mean $X$).

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 $X$ 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_Constistency, FS_IWSS, FS_MOEA, FS_PSO) are designed to reduce the dimensionality of data and meanwhile 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 $X$. Note we do not report the performance of FT_PCA.
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.

ACKNOWLEDGEMENTS

This research / project is supported by the National Research Foundation, Singapore under its International Research Centres in Singapore Funding Initiative. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore.
