Fast Bi-Objective Feature Selection Using Entropy Measures and Bayesian Inference

Yi Mei
School of Engineering and CS
Victoria University of Wellington
Wellington, New Zealand
yi.mei@ecs.vuw.ac.nz

Bing Xue
School of Engineering and CS
Victoria University of Wellington
Wellington, New Zealand
bing.xue@ecs.vuw.ac.nz

Mengjie Zhang
School of Engineering and CS
Victoria University of Wellington
Wellington, New Zealand
mengjie.zhang@ecs.vuw.ac.nz

ABSTRACT

The entropy measures have been used in feature selection for decades, and showed competitive performance. In general, the problem aims at minimizing the conditional entropy of the class label on the selected features. However, the generalization of the entropy measures has been neglected in literature. Specifically, the use of conditional entropy has two critical issues. First, the empirical conditional distribution of the class label may have a low confidence and thus is unreliable. Second, there may not be enough training instances for the selected features, and it is highly likely to encounter new examples in the test set. To address these issues, a bi-objective optimization model with a modified entropy measure called the *Bayesian entropy* is proposed. This model considers the *confidence* of the optimized conditional entropy value as well as the conditional entropy value itself. As a result, it produces multiple feature subsets with different trade-offs between the entropy value and its confidence. The experimental results demonstrate that by solving the proposed optimization model with the new entropy measure, the number of features can be dramatically reduced within a much shorter time than the existing algorithms. Furthermore, similar or even better classification accuracy was achieved for most test problems.

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning—*Machine Learning*; G.1.6 [**Numerical Analysis**]: Optimization—*Global Optimization*

Keywords

Feature Selection; Multi-Objective Computation; Generalization

1. INTRODUCTION

In data mining and machine learning tasks, feature selection [9, 25] can improve the quality of the data space by selecting only a small subset of relevant features and removing irrelevant and/or redundant features. By doing so, feature selection can reduce the dimensionality of the data, maintain or even improve the learning performance, reduce the training speed of a learning algorithm, and simplify the learnt models [5, 9]. Feature selection has been successfully used to improve the data space for many tasks, such as classification and clustering [16], and the focus of this paper is feature selection for classification.

Existing feature selection methods can be generally grouped into two categories [5]: filter and wrapper approaches. The main difference between them lies in the evaluation criteria. Wrapper approaches [13] "wrapper" a classification algorithm into the feature selection process to evaluate candidate feature subsets. Filter approaches are independent of any classification algorithm, where the selected features are evaluated based on the characteristics of the data. Wrapper approaches can often achieve better classification performance than filter approaches, but filter approaches are usually faster and the selected features are general to different classification algorithms, i.e. can maintain or even increase the classification accuracy of different classifiers [5]. While the classification algorithm is used as a "black box" in wrappers, filter approaches focus on the data itself, aiming to improve the overall quality of the data space, which is particularly important to avoid overfitting and perform data analysis for real-world complex problems.

There have been different filter measures proposed, such as the measures based on information theory [18], distance measures [2], and rough-set-based measures [22]. Among these measures, those based on information theory are the most popular ones [18, 20, 8]. They have been used in many different ways [27, 23, 17, 28, 21, 7, 20, 3]. For example, they can be used as evaluation criteria to guide the search direction [4, 10]. Most of these approaches are based on entropy measures, e.g. the conditional entropy between the feature and class labels, and the joint distribution of the selected features. These measures become very popular since entropy measures have strong theoretical background to provide reliable solutions. However, there are still some limitations in the existing approaches, mainly in how to calculate the entropies to maintain the generalizability from the training set to the test set (details in Section 2.2).

Furthermore, searching for the optimal feature subsets is challenging due to the large search space, which grows exponentially along with the number of available features in the dataset (2^m, where m is the number of features). Exhaustively searching for all the possible feature subsets is practically impossible in most situations. Therefore, heuristic search techniques are often popular in solving feature selection problems. However, since feature selection has a complex search space, many existing methods, such greedy search, still suffer from a variety of problems, e.g. stag-

GECCO'16, July 20-24, 2016, Denver, Colorado, USA.

DOI: 10.1145/1235

nation into local optima [23, 15]. Evolutionary Computation (EC) includes a group of powerful search techniques, which have been widely applied in many areas, including feature selection [26, 11, 28, 12, 14].

In this paper, we focus on the investigation of the filter approaches, aiming to design faster feature selection approaches and gain a deeper understanding of the relationship between the selected features and the resultant classification accuracy. More specifically, we will focus on the entropy-based feature selection approaches which minimizes the conditional entropy (maximizes the information gain).

1.1 Goals

The overall goal of this paper is to investigate the limitations of the existing entropy-based approaches, and propose new algorithms to address these issues. More specifically,

1. Two limitations that affect the generalization are identified: (1) the calculation of the entropy is based on empirical probabilities, which might be quite different between the training set and the test set; (2) the training set may not contain enough instances to cover all the occurrences of the selected features.

2. To address these two issues and to improve the tradeoff between the empirical samples and the potential outliers, a bi-objective optimization model is proposed. In addition, the empirical probability is modified to a posterior one, which is called the *Bayesian entropy*.

3. The efficacy of the proposed algorithms is verified on a number of classification datasets by comparing the test performance of different classifiers when using the selected features and using all the features.

1.2 Organization

The rest of the paper is organized as follows: First, the background of using entropy measures for feature selection and its limitations are introduced in Section 2. Then, the proposed approach is described in Section 3. Experimental studies and analysis are conducted in Section 5. Finally, conclusions and future work are given in Section 6.

2. ENTROPY MEASURES FOR FEATURE SELECTION

2.1 Entropy Measures

When using entropy measures for feature selection, the problem aims to select a subset of features $\mathcal{X}^s$ out of the entire feature set $\mathcal{X}$, so that the conditional entropy [19] $H(Y|\mathcal{X}^s)$ of the class label Y on $\mathcal{X}^s$ is minimized. The problem can be simply stated as follows:

$$\min_{\mathcal{X}^s \subseteq \mathcal{X}} H(Y|\mathcal{X}^s). \tag{1}$$

For any X and Y, the entropy measures have the following useful properties:

$$H(X) > 0,\ H(Y) > 0, \tag{2}$$

$$H(Y|X) \leq H(Y), \tag{3}$$

$$H(X,Y) \geq \max\{H(X), H(Y)\}. \tag{4}$$

Based on the above properties, we can obtain the lower and upper bounds of the objective function as follows:

$$H(Y|\mathcal{X}) \leq H(Y|\mathcal{X}^s) \leq H(Y). \tag{5}$$

Obviously, the optimal value of Eq. (1) is zero, in which case the class label Y is completely determined by the value of $\mathcal{X}^s$.

2.2 Limitations in Entropy for Feature Selection

Although the entropy-based feature selection method has strong theoretical background and is classifier-independent, there are two major drawbacks that hinder it from selecting the truly important features. First, the conditional entropy totally depends on the empirical probabilities (density or frequency) of the occurrences in the training set. In this case, the empirical probabilities may be highly uncertain, and vary a lot in the test data. A simple example of this is the index feature I. It is obvious that the class label Y is completely determined by the index, since each index corresponds to a unique instance, and thus a unique class label. Hence, $H(Y|I) = 0$. However, there is only one sample for each index value i, which is far less enough for making a reliable prediction about Y. This is essentially the same as the situation when we flip a coin once and obtain a result of head, we are still highly uncertain about which side will be on top in the next flip.

Second, the number of possible (joint) occurrences increases with the increase of the number of selected features. Therefore, there may not be enough training instances to cover all the occurrences of the selected features, especially when the number of selected features is relatively large relative to the given number of training instances. In this case, one will still be unable to predict the label of a test instance whose feature values were never seen before.

3. THE PROPOSED BI-OBJECTIVE OPTIMIZATION FOR FEATURE SELECTION

In order to address the issues discussed above, the joint entropy of the selected features $H(\mathcal{X}^s)$ is included in the features selection as well as the original objective $H(Y|\mathcal{X}^s)$. While $H(Y|\mathcal{X}^s)$ indicates how well Y depends on $\mathcal{X}^s$, $H(\mathcal{X}^s)$ implies how much we can trust the dependency, i.e., how likely the same pattern will occur in the test set as it is in the training set. For example, given a feature X, $H(X) = 0$ indicates that $\Pr(X = a) = 1$, where a is the sole value of X occurring in the training set. In this case, it is reasonable to believe that X has a stable distribution, and will be highly likely to take the value of a in any unseen test instance. On the other hand, a large $H(X)$ corresponds to a more random distribution, which makes the occurrences in the training set unlikely to be repeated in the test set. For instance, the entropy of the index feature I is the largest possible value. That is, $H(I) = \log_2 n$, where n is the number of training instances. This implies that I has a highly random distribution, and each occurrence in the training set is unique. As expected, they are never repeated in the test instances. Therefore, given a test instance whose feature value has not appeared in the training set, its class label can hardly be predicted, since there is no reference training instance. Based on the above considerations, we propose a bi-objective optimization problem for feature selection, which is stated as follows:

$$\min_{\mathcal{X}^s \subseteq \mathcal{X}} H(Y|\mathcal{X}^s), \tag{6}$$

$$\min_{\mathcal{X}^s \subseteq \mathcal{X}} H(\mathcal{X}^s). \tag{7}$$

The above bi-objective optimization problem can be directly solved by any evolutionary multi-objective algorithms. Here, we adopt the well known evolutionary multi-objective algorithm named non-dominated sorting genetic algorithm II (NSGA-II) [6]. The pseudo code of NSGA-II for solving the above problem is described in Algorithm 1. In the algorithm, each individual is represented as an m-dimensional bit string, where m is the total number of features. If the i^{th} bit takes 1, then the i^{th} feature is selected. Otherwise, the feature is not selected. In lines 15 and 16, the normalization of the objective functions are conducted as follows:

$$\bar{H}(Y|\mathcal{X}^s) = \frac{H(Y|\mathcal{X}^s) - H(Y|\mathcal{X})}{H(Y) - H(Y|\mathcal{X})}, \quad (8)$$

$$\bar{H}(\mathcal{X}^s) = \frac{H(\mathcal{X}^s) - \min_{X\in\mathcal{X}}[H(X)]}{H(\mathcal{X}) - \min_{X\in\mathcal{X}}[H(X)]}. \quad (9)$$

Finally, a set of non-dominated solutions (feature subsets) in terms of Eqs. (6) and (7) are obtained by Algorithm 1.

ALGORITHM 1: NSGA-II for minimizing $H(Y|\mathcal{X}^s)$ and $H(\mathcal{X}^s)$.

```
   // Initialization
1  pop ← ∅, arch ← ∅;                    // arch is the
   non-dominated archive
2  for i = 1 → popsize do
3  |  Randomly generate an individual indi;
4  |  pop ← pop ∪ indi;
5  end
6  Update arch with pop;
   // Search process
7  while Stopping criteria are not met do
8  |  tpop ← pop;
9  |  for i = 1 → popsize/2 do
10 |  |  Select two parents par_1 and par_2 from pop by
   |  |  tournament selection;
11 |  |  Generate ox_1 and ox_2 by applying the single-point
   |  |  crossover operator to par_1 and par_2;
12 |  |  Generate om_1 and om_2 by applying the single flip
   |  |  mutation operator to ox_1 and ox_2;
13 |  |  Calculate H(Y|om_1) and H(om_1);
14 |  |  Calculate H(Y|om_2) and H(om_2);
15 |  |  Calculate the normalized H̄(Y|om_1) and H̄(om_1);
16 |  |  Calculate the normalized H̄(Y|om_2) and H̄(om_2);
17 |  |  tpop ← tpop ∪ {om_1, om_2};
18 |  end
19 |  Sort tpop by fast non-dominated sorting;
20 |  pop ← tpop[1 : popsize];
21 end
22 Update arch with pop;
23 return arch;
```

3.1 Bayesian Entropy

The proposed bi-objective optimization problem still has a drawback that may reduce the generalizability of the selected features. The drawback is the use of the empirical entropy, which is calculated directly based on the empirical density, which uses the empirical density of the occurrences. Table 1 gives an example when the empirical entropy fails. In the table, both the training and test sets have five features X_1 to X_5. The class label Y is obtained by either checking whether there are odd number of bits in (X_1, X_2, X_3) (3-bit parity problem), or applying the AND operator to X_4 and X_5. Therefore, one can either choose the feature set $\mathcal{X}_1^s = \{X_1, X_2, X_3\}$ or $\mathcal{X}_2^s = \{X_4, X_5\}$. When calculating the two objectives (6) and (7) from the training set, one can easily get

$$H(Y|\mathcal{X}_1^s) = 0, H(X_1^s) = 1.50,$$
$$H(Y|\mathcal{X}_2^s) = 0, H(X_2^s) = 1.56.$$

Therefore, $\mathcal{X}_1^s$ is considered to be better than $\mathcal{X}_2^s$. However, when being applied to the test set, $\mathcal{X}_2^s$ is much better than $\mathcal{X}_1^s$ in the sense that all the occurrences of $\mathcal{X}_1^s$ are unseen in the training set. Intuitively, if there are more features selected, then the joint variable space of the features will become larger (unless the features are strongly correlated). Thus, given the same number of training instances, it is less likely to cover all the occurrences in the joint variable space. For example, there are three occurrences in the training set for both $\mathcal{X}_1^s$ ($(0,0,1)$, $(1,1,0)$ and $(1,0,1)$) and $\mathcal{X}_2^s$ ($(1,1)$, $(0,1)$ and $(1,0)$). However, in the test set, there will be five outliers for X_1^s and one outlier for X_2^s ($(0,0)$). As a result, one has to make a random guess for all the five test instances with X_1^s, but for only one test instance with X_2^s.

Table 1: An example of the training and test sets.

	Training							Test						
ID	X_1	X_2	X_3	X_4	X_5	Y		ID	X_1	X_2	X_3	X_4	X_5	Y
1	0	0	1	1	1	1		1	1	0	0	1	1	1
2	1	1	0	0	1	0		2	1	1	1	1	1	1
3	1	1	0	0	1	0		3	0	1	1	0	1	0
4	1	0	1	0	1	0		4	0	0	0	0	0	0
5	1	0	1	1	0	0		5	0	1	0	1	1	1
6	1	1	0	1	0	0								
7	0	0	1	1	1	1								
8	1	1	0	1	0	0								

From the above example, one can see two issues of the empirical entropy, which adopts the empirical density. First, the set of occurrences in the training set may not be able to cover all the possible occurrences of the selected features, especially when the number of features is large relative to the number of training instances. Second, the empirical entropy tends to put more emphasis on the more biased distributions such as the one in Table 1. This can increase the confidence of the majority occurrence (e.g., $(X_1, X_2, X_3) = (1, 1, 0)$ occurred 4 times in the training set of Table 1). However, the confidence of the minority occurrences (e.g., $(X_1, X_2, X_3) = (0, 0, 1)$ occurred twice in the training set of Table 1) will be low due to the lack of training instances with the same feature values, not to mention the outliers which failed to occur in the training set at all.

To address the above issues, we proposed to replace the empirical density with the posterior probability, which is estimated by Bayesian inference. The resultant entropies are then called the *Bayesian entropies*. They are defined as follows:

$$H_B(\mathcal{X}^s) = -\sum_{\mathbf{x}^s\in\Omega(\mathcal{X}^s)} p_B(\mathbf{x}^s)\log_2 p_B(\mathbf{x}^s), \quad (10)$$

$$H_B(Y|\mathcal{X}^s) = \sum_{(\mathbf{x}^s,y)\in\Omega(\mathcal{X}^s,Y)} p_B(\mathbf{x}^s, y)\log_2 \frac{p_B(\mathbf{x}^s)}{p_B(\mathbf{x}^s, y)}, \quad (11)$$

where $p_B(\mathbf{x})$ of the vector $\mathbf{x}$ stands for the posterior probability estimated based on the assumed prior distribution and the training instances as samples. $\Omega(\mathcal{X})$ indicates the *domain* of the variables $\mathcal{X}$, which is defined as the set of all the possible values that $\mathcal{X}$ can take. For example, if X is a binary variable, then $\Omega(X) = \{0, 1\}$.

As a starting point, the prior distribution is simply assumed to be the binomial distribution here. For a random variable X and an occurrence $x \in \Omega(X)$, suppose that the prior probability of x

is θ, i.e., $p(x) = \theta$. If x appears m times among n observations ($m \leq n$), then the posterior probability of θ can be calculated based on Bayes' theorem as:

$$p(\theta|m,n) = \frac{p(m,n|\theta)p(\theta)}{\int_\theta p(m,n|\theta)p(\theta)}, \tag{12}$$

where $p(m,n|\theta)p(\theta) = \binom{n}{m}\theta^m(1-\theta)^{n-m}$. In Eq. (12), a conjugate prior distribution of beta function is commonly assumed for θ. That is, $\theta \sim \mathbf{B}(\alpha,\beta)$, where α and β are the parameters of the beta function. Then, the posterior probability still follows a beta function as follows:

$$\theta|m,n \sim \mathbf{B}(\alpha+m, \beta+n-m). \tag{13}$$

The expectation of the posterior probability is

$$E[\theta|m,n] = \frac{\alpha+m}{\alpha+\beta+n}. \tag{14}$$

When, $n = m = 0$, the posterior probability is reduced to the prior probability, whose expectation is

$$E[\theta] = E[\theta|0,0] = \frac{\alpha}{\alpha+\beta}. \tag{15}$$

Given a domain Ω, with no prior knowledge, it is reasonable to assume that each occurrence $occ \in \Omega$ has the same probability. In this case, $\alpha + \beta = \alpha \cdot |\Omega|$.

When determining Ω, there are two different assumptions called *independent* and *dependent* assumptions, which are described as follows:

- *Independent* assumption: The features are independent, and Ω is defined as the Cartesian product of their own domains;
- *Dependent* assumption: The features are dependent, and Ω only consists of the occurrences happened in the training instances.

For example, given two binary features X_4 and X_5 in Table 1, it is known that $\Omega(X_4) = \Omega(X_5) = \{0,1\}$. Then, under the independent assumption, $\Omega(X_1,X_2) = \Omega(X_1) \times \Omega(X_2) = \{(0,0),(0,1),(1,0),(1,1)\}$. However, under the dependent assumption with the training instances given in Table 1, $\Omega(X_1,X_2) = \{(1,1),(0,1),(1,0)\}$.

Algorithm 2 describes the calculation of the proposed Bayesian entropy measures under the independent and dependent assumptions. In lines 13 and 17, n stands for the number of training instances. Different assumptions lead to different size of domains (lines 5 to 10), and thus different $H_B(Y|\mathcal{X}^s)$ and $H_B(\mathcal{X}^s)$ values.

The parameter α controls the degree of impact of the observations on the posterior probabilities. Specifically, the impact decreases with the increase of α. Extremely, when $\alpha = 0$, the posterior probability becomes the empirical density. When $\alpha \to \infty$, the posterior probability converges to the prior probability.

According to Algorithm 2, the computational complexity of fitness evaluation is $O(n + |\Omega(\mathcal{X}^s)| + |\Omega(\mathcal{X}^s,Y)|)$, where n is the number of training instances. In practice, it is normal to assume that $|\Omega(\mathcal{X}^s)| \ll n$ and $|\Omega(\mathcal{X}^s,Y)| \ll n$. In this case, the complexity becomes $O(n)$.

4. HOW IT WORKS: A DEMONSTRATION

To help understand how the proposed approach works, we provide a demonstration based on the example given in Table 1. In the table, suppose that we have already found three feature sets

ALGORITHM 2: Calculation of $H_B(Y|\mathcal{X}^s)$ and $H_B(\mathcal{X}^s)$.

1 **foreach** $X^s \in \mathcal{X}^s$ **do**
2 $\quad \Omega(X^s) \leftarrow$ all the distinct occurrences of X^s in the training set;
3 **end**
4 $\Omega(Y) \leftarrow$ all the distinct occurrences of Y in the training set;
5 **if** *independent assumption* **then**
6 $\quad \Omega(\mathcal{X}^s) \leftarrow \Omega(X_1^s) \times \cdots \times \Omega(X_{|\mathcal{X}^s|}^s)$, $\Omega(\mathcal{X}^s,Y) \leftarrow \Omega(\mathcal{X}^s) \times \Omega(Y)$;
7 **else**
8 $\quad \Omega(\mathcal{X}^s) \leftarrow$ all the distinct occurrences of $\mathcal{X}^s$ in the training set;
9 $\quad \Omega(\mathcal{X}^s,Y) \leftarrow$ all the distinct occurrences of $(\mathcal{X}^s,Y)$ in the training set;
10 **end**
11 **foreach** $\mathbf{x}^s \in \Omega(\mathcal{X}^s)$ **do**
12 $\quad count(\mathbf{x}^s) \leftarrow$ number of training instances where $\mathbf{x}^s$ occurred;
13 $\quad p_B(\mathbf{x}^s) \leftarrow \frac{\alpha+count(\mathbf{x}^s)}{|\Omega(\mathcal{X}^s)|\cdot\alpha+n}$;
14 **end**
15 **foreach** $(\mathbf{x}^s,y) \in \Omega(\mathcal{X}^s,Y)$ **do**
16 $\quad count(\mathbf{x}^s,y) \leftarrow$ number of training instances where $(\mathbf{x}^s,y)$ occurred;
17 $\quad p_B(\mathbf{x}^s,y) \leftarrow \frac{\alpha+count(\mathbf{x}^s,y)}{|\Omega(\mathcal{X}^s,Y)|\cdot\alpha+n}$;
18 **end**
19 Calculate $H_B(\mathcal{X}^s)$ and $H_B(\mathcal{X}^s,Y)$ by Eqs. (10) and (11);
20 **return** $H_B(\mathcal{X}^s)$ and $H(Y|\mathcal{X}^s)$;

$\mathcal{X}_1^s = \{X_1,X_2,X_3\}$, $\mathcal{X}_2^s = \{X_4,X_5\}$ and $\mathcal{X}_3^s = \{\text{ID}\}$, all are able to classify the label Y without confusion, i.e.

$$H(Y|\mathcal{X}_1^s) = H(Y|\mathcal{X}_2^s) = H(Y|\mathcal{X}_3^s) = 0.$$

First, as we introduce the second objective, we have

$$H(\mathcal{X}_1^s) = 1,500,\ H(\mathcal{X}_2^s) = 1.561,\ H(\mathcal{X}_3^s) = 3.00.$$

Obviously, the index feature $\mathcal{X}_1^s$ is much worse than the other two feature sets, as its entropy is much larger. However, $\mathcal{X}_1^s$ is considered to be the best, although it is more likely to have outliers in the test set.

Then, after modifying the original entropy to the Bayesian entropy, we have two parameters to set, one is α and the other is the assumption of the domain (dependent or independent). If we set $\alpha = 10$, then under the dependent assumption, we have

$$H_B(Y|\mathcal{X}_1^s) = H_B(Y|\mathcal{X}_2^s) = H_B(Y|\mathcal{X}_3^s) = 0,$$
$$H_B(\mathcal{X}_1^s) = 1.581,\ H_B(X_2^s) = 1.584,\ H_B(\mathcal{X}_2^s) = 3.00.$$

It can be seen that the difference between $H_B(X_1^s)$ and $H_B(X_2^s)$ is reduced. Under the independent assumption, on the other hand, we have

$$H_B(Y|\mathcal{X}_1^s) = 1.004,\ H_B(Y|\mathcal{X}_2^s) = 0.998,\ H_B(Y|\mathcal{X}_3^s) = 0.998,$$
$$H_B(\mathcal{X}_1^s) = 2.989,\ H_B(\mathcal{X}_2^s) = 1.992,\ H_B(\mathcal{X}_2^s) = 3.00.$$

One can see that $\mathcal{X}_2^s$ becomes better than $\mathcal{X}_1^s$ in terms of both objectives, and thus will be selected instead.

In summary, the above example shows that

- With the help of the proposed second objective, one can identify the features with much higher confidence (e.g. $\mathcal{X}_1^s$ and $\mathcal{X}_1^s$ against $\mathcal{X}_3^s$);
- With proper parameter setting of the Bayesian entropy, one can shift the bias to the empirical samples, and thus change the selected feature set (e.g. from $\mathcal{X}_1^s$ to $\mathcal{X}_2^s$ by setting $\alpha = 10$ and adopting the independent assumption).

5. EXPERIMENTAL STUDIES

5.1 Experimental Settings

Eight discrete datasets were selected from UCI machine learning repository [1], whose details are given in Table 2. In each dataset, 70% of the instances are used as training set and the other 30% are used as test set. To evaluate the selected features, four different types of classification algorithms were applied, and their test accuracies were compared with the test accuracy obtained on the entire feature set. The classification algorithms include k-nearest neighbour algorithm with $k = 1$ (1-NN) based on distance, Random Forest (RF) based on ensemble learning, J48 decision tree based on information gain, and Naïve Bayes (NB) based on probabilities. The parameter $\alpha \in \{0, 0.01, 0.1, 1, 10, 100\}$ were tested and compared to each other. Note that $\alpha = 0$ indicates the use of the original entropy measures.

Table 2: Properties of the datasets.

Dataset	#Features	#Classes	#Instances
Lymphography (Lymph)	18	4	148
Mushroom (Mush.)	22	2	5644
Spect	22	2	267
Leddisplay(Led.)	24	10	1000
Ionosphere (Iono.)	34	2	351
Chess	36	2	3196
Lung Cancer (Lung)	56	3	32
Splice	61	3	3190

The total number of evaluations was set to $1000 \cdot m$, where m is the number of features in the dataset. For each α value on each dataset, 30 independent runs were conducted. For each run, a set of non-dominated feature subsets were obtained. Then, for each classifier, the feature subset yielding the best training accuracy was selected and used in the test phase.

5.2 Results and Discussions

Table 3 shows the results of the 1-NN, RF, J48 and NB classifiers on the features selected by different α values as well as the entire feature set under the *dependent* assumption. For each dataset and classifier, t-test with significance level of 0.05 was also conducted between the accuracies of the best α value(s) and all the features. The significantly better one is marked in bold. From the table, one can see that the proposed algorithm considerably reduced the number of features for all the datasets. For the lung dataset, the average number of selected features is even less than 10% of the total number of features.

In most cases, using the selected features only, the classification performance of the four algorithms can be maintained or even increased. This shows the proposed filter approach can obtain features that are general to different types of classification algorithms. Specifically,

- On 6 out of the 8 datasets, the 1-NN classifier achieved significantly better test accuracy when using the selected features than using all the features. In the Led. dataset, 1-NN classifier managed to achieve 100% accuracy with much fewer features (7 versus 22). On the Iono. dataset, statistically comparable test performance was achieved with less than 18% of the features (6 versus 34).
- For the RF classifier, the advantage of the selected features is not so obvious. Using the selected features led to significantly better accuracy on the Spect dataset, and was defeated by the entire set on the Lymph and Splice dataset. On the remaining 5 datasets, the difference between the test accuracies was insignificant, including the Led. and Mush. datasets, on which a 100% accuracy was consistently achieved.
- In 5 out of the 8 cases, the J48 classifier using the selected features obtained by the best α value(s) achieved significantly better accuracy than using all the features. On the remaining 3 datasets, the difference was insignificant.
- For the NB classifier, the results are mixed. Using the selected features led to significantly better accuracy on 4 datasets, but significantly worse accuracy on 3 datasets. On the remaining Led. dataset, all the feature sets led to a 100% accuracy.
- Overall, the advantage of the selected features over the entire feature set is the most obvious for the 1-NN classifier. This is consistent with the rationale of the entropy optimization, which assumes that given a test instance, if there exists a training instance with the same values of the selected features (zero distance), then they tend to have the same class label. In addition, since the 1-NN classifier is generally the simplest classifier and thus usually perform worse than other classifiers, the significant improvement for the 1-NN classifier reduces the difference between the test performance of different classifiers, and makes the classification more robust.

When comparing among different α values, it can be seen that the relative performance of α depends on classifier. For example, on the Spect dataset, the test accuracy tends to improve with the increase of α for the 1-NN and RF classifiers, but with the decrease of α for the J48 and NB. In most cases, the results of different α values are not much different from each other, i.e. the test accuracy is not sensitive to the α value. From the table, a value from 0.1 to 1 seems to be a reasonable setting in most cases, unless the size of the training set is too small, where a much larger α may be needed.

Table 4 shows the corresponding results under the *independent* assumption. The table shows a similar pattern as Table 3 in terms of the relative performance of the selected feature to that of all the features. For the 1-NN classifier, the selected features led to significantly better accuracy on 6 datasets. For the RF classifier, the selected features achieved significantly better accuracy on 2 datasets, but significantly worse accuracy on 4 datasets. For the J48 classifier, the selected features obtained significantly better accuracy on 5 datasets, and worse on 1 dataset. For the NB classifier, feature selection significantly improved the accuracy on 5 datasets, while outperformed by the original feature set on 2 datasets.

Overall, the dependent assumption led to better results than the independent assumption on most of the datasets, except the Lung dataset. From Table 2, we can see that the number of instances in all the datasets are relatively large except the Lung dataset, which only contains 32 instances in total. As a result, there are only 22 training instances ($70\% \times 32$). Such a small number of training instances can hardly cover all the possible occurrences of the features most of the time. In this case, the independent assumption is better than the dependent assumption, as it takes the outliers into account. Table 4 shows that under the independent assumption, the test accuracy of all the four classifiers have been dramatically improved, and the number of selected features is reduced from 4 to 2. For the other datasets, the probability of missing occurrences in the training set is lower due to the relatively sufficient number of instances. Therefore, the dependent assumption outperformed the independent assumption. In summary, if there are a large number of training instances, then the dependent assumption tends to be better. In contrast, if the training instances are not enough or many

Table 3: The results of 1-NN, Random Forest (RF), J48, and Naïve Bayes (NB) classifiers under the *dependent* assumption.

Dataset	α	1-NN		RF		J48		NB		Time
		Accuracy	$\lvert\mathcal{X}^s\rvert$	Accuracy	$\lvert\mathcal{X}^s\rvert$	Accuracy	$\lvert\mathcal{X}^s\rvert$	Accuracy	$\lvert\mathcal{X}^s\rvert$	(s)
Chess	all	95.40	36	99.06	36	99.34	36	83.76	36	-
	0	97.02 ± 0.29	26.4	98.95 ± 0.18	26.4	99.28 ± 0.23	25.3	93.67 ± 0.24	6.6	55.7
	0.01	**97.04 ± 0.28**	26.6	99.02 ± 0.16	26.6	99.30 ± 0.18	25.6	93.71 ± 0.03	5.5	66.2
	0.1	96.90 ± 0.27	26.7	99.01 ± 0.14	26.7	99.32 ± 0.20	25.6	93.71 ± 0.13	6.5	60.2
	1	97.00 ± 0.34	26.7	99.05 ± 0.15	26.7	99.26 ± 0.15	26.1	93.71 ± 0.05	6.4	61.7
	10	97.04 ± 0.35	26.9	99.09 ± 0.14	26.9	99.27 ± 0.13	26.7	**93.72 ± 0.03**	6.0	58.9
	100	96.13 ± 1.06	25.6	97.96 ± 1.89	25.6	98.31 ± 1.70	25.7	93.22 ± 0.69	11.7	50.4
Lung	all	63.64	56	81.82	56	72.73	56	**72.73**	56	-
	0	45.45 ± 0.00	4.0	45.45 ± 0.00	4.0	36.36 ± 0.00	4.0	45.45 ± 0.00	4.0	0.5
	0.01	45.45 ± 0.00	4.0	45.45 ± 0.00	4.0	36.36 ± 0.00	4.0	45.45 ± 0.00	4.0	0.6
	0.1	45.45 ± 0.00	4.0	45.45 ± 0.00	4.0	36.36 ± 0.00	4.0	45.45 ± 0.00	4.0	0.6
	1	45.45 ± 0.00	4.4	45.45 ± 0.00	4.4	36.36 ± 0.00	4.4	45.45 ± 0.00	4.4	0.6
	10	**81.82 ± 0.00**	3.5	81.82 ± 0.00	3.5	**81.82 ± 0.00**	3.5	45.45 ± 0.00	4.4	0.5
	100	**81.82 ± 0.00**	4.0	81.82 ± 0.00	4.0	**81.82 ± 0.00**	4.0	45.45 ± 0.00	4.8	0.5
Lymph	all	63.27	18	**85.71**	18	75.51	18	**85.71**	18	-
	0	72.14 ± 5.59	9.4	76.79 ± 3.51	9.4	77.86 ± 1.35	8.6	80.05 ± 3.08	6.4	0.8
	0.01	73.06 ± 5.70	9.1	77.04 ± 3.39	9.1	77.50 ± 1.88	8.6	79.69 ± 3.55	6.3	0.8
	0.1	**73.52 ± 5.42**	9.4	76.43 ± 3.33	9.4	**78.01 ± 1.57**	9.0	79.54 ± 3.35	6.6	0.8
	1	72.24 ± 6.02	8.4	75.26 ± 4.60	8.4	75.05 ± 4.30	7.5	79.54 ± 3.05	6.3	0.8
	10	73.47 ± 3.75	8.3	74.08 ± 2.93	8.3	73.47 ± 4.00	8.1	79.95 ± 4.00	7.5	0.8
	100	73.16 ± 5.26	8.7	73.78 ± 4.07	8.7	73.47 ± 4.21	8.2	77.55 ± 3.36	8.3	0.8
Led.	all	79.88	24	100.00	24	100.00	24	100.00	24	-
	0	**100.00 ± 0.00**	6.0	100.00 ± 0.00	6.0	100.00 ± 0.00	6.0	100.00 ± 0.00	6.0	6.4
	0.01	**100.00 ± 0.00**	6.0	100.00 ± 0.00	6.0	100.00 ± 0.00	6.0	100.00 ± 0.00	6.0	7.4
	0.1	**100.00 ± 0.00**	6.0	100.00 ± 0.00	6.0	100.00 ± 0.00	6.0	100.00 ± 0.00	6.0	6.7
	1	**100.00 ± 0.00**	5.4	100.00 ± 0.00	5.4	100.00 ± 0.00	5.4	100.00 ± 0.00	5.4	6.9
	10	**100.00 ± 0.00**	6.2	100.00 ± 0.00	6.2	100.00 ± 0.00	6.2	100.00 ± 0.00	6.2	6.8
	100	**100.00 ± 0.00**	5.7	100.00 ± 0.00	5.7	100.00 ± 0.00	5.7	100.00 ± 0.00	5.7	6.4
Mush.	all	100.00	22	100.00	22	100.00	22	95.75	22	-
	0	100.00 ± 0.00	6.7	100.00 ± 0.00	6.7	99.89 ± 0.04	5.8	**97.50 ± 0.00**	1.8	32.9
	0.01	100.00 ± 0.00	6.3	100.00 ± 0.00	6.3	99.87 ± 0.06	5.7	**97.50 ± 0.00**	1.6	39.0
	0.1	100.00 ± 0.00	6.1	100.00 ± 0.00	6.1	99.88 ± 0.05	6.0	**97.50 ± 0.00**	1.0	35.4
	1	100.00 ± 0.00	6.0	100.00 ± 0.00	6.0	99.87 ± 0.06	5.9	**97.50 ± 0.00**	1.9	36.8
	10	100.00 ± 0.00	6.6	100.00 ± 0.00	6.6	100.00 ± 0.00	6.6	**97.50 ± 0.00**	2.0	35.9
	100	100.00 ± 0.00	6.3	100.00 ± 0.00	6.3	100.00 ± 0.00	6.3	**97.50 ± 0.00**	1.5	34.0
Spect	all	64.04	22	64.04	22	68.54	22	67.42	22	-
	0	62.39 ± 0.95	16.9	63.82 ± 0.46	16.9	**70.25 ± 1.78**	14.5	70.03 ± 2.64	8.3	2.1
	0.01	64.19 ± 1.89	20.2	63.23 ± 1.42	20.2	68.31 ± 1.42	20.1	70.14 ± 2.27	8.4	2.4
	0.1	63.29 ± 2.62	19.4	64.27 ± 1.75	19.4	68.37 ± 1.07	19.2	**70.48 ± 2.29**	8.2	2.2
	1	63.96 ± 2.34	20.2	64.33 ± 1.83	20.2	68.03 ± 2.25	20.1	69.86 ± 2.34	8.1	2.4
	10	65.81 ± 2.67	19.1	64.38 ± 1.73	19.1	67.75 ± 1.73	19.2	69.58 ± 2.77	7.2	2.2
	100	**66.07 ± 3.02**	18.7	**65.25 ± 1.93**	18.7	67.33 ± 1.96	18.9	67.81 ± 3.40	6.2	2.0
Splice	all	63.97	61	**94.83**	61	92.94	61	**92.10**	61	-
	0	76.81 ± 2.56	9.6	90.06 ± 3.56	9.6	93.73 ± 0.86	8.2	89.38 ± 0.74	7.0	71.0
	0.01	75.31 ± 3.42	9.6	88.23 ± 4.90	9.6	93.71 ± 0.95	8.4	89.26 ± 0.70	7.0	82.8
	0.1	74.99 ± 3.54	9.7	88.26 ± 5.03	9.7	93.32 ± 1.49	8.2	89.47 ± 0.73	7.1	75.3
	1	76.32 ± 3.32	9.8	90.43 ± 3.82	9.8	**94.13 ± 1.19**	8.4	88.93 ± 0.50	6.6	78.1
	10	76.50 ± 2.50	9.7	90.16 ± 3.68	9.7	93.64 ± 1.19	8.6	89.36 ± 0.15	7.3	74.2
	100	**77.70 ± 3.50**	9.9	91.34 ± 4.04	9.9	94.10 ± 0.75	8.8	89.41 ± 0.25	7.4	70.8
Iono.	all	87.18	34	94.87	34	87.18	34	81.20	34	-
	0	86.39 ± 2.13	5.2	92.71 ± 2.08	5.2	87.78 ± 1.12	4.4	**89.08 ± 1.12**	3.6	3.0
	0.01	86.69 ± 2.14	5.2	92.69 ± 1.90	5.2	88.12 ± 0.74	4.6	**89.08 ± 0.83**	3.8	3.3
	0.1	86.00 ± 2.03	5.3	92.88 ± 1.95	5.3	89.19 ± 1.39	4.7	88.93 ± 0.84	3.7	3.1
	1	87.63 ± 1.90	5.2	93.89 ± 1.38	5.2	88.76 ± 2.26	4.4	87.54 ± 1.98	4.0	3.2
	10	85.49 ± 1.67	4.8	92.37 ± 2.74	4.8	**89.29 ± 1.78**	4.1	88.80 ± 1.24	3.7	3.1
	100	86.00 ± 2.03	5.8	94.72 ± 1.79	5.8	88.70 ± 2.24	4.4	86.65 ± 2.57	4.2	3.0

· "Accuracy" and "$\lvert\mathcal{X}^s\rvert$ stand for the test accuracy and number of selected features. "Time (s)" is the computational time for feature selection in seconds. "all" means using all the features.
· For each problem and classifier, t-test with the significance level of 0.05 was conducted between the accuracies of the best α value(s) and all the features. The significantly better one is marked in bold.

features are selected, then the independent assumption is recommended to be adopted.

It is worth noting that the independent and dependent assumptions are two extremes of the occurrence coverage in the training set. The actual case can be in between, i.e. the domain is larger than the empirical occurrence set, but smaller than the Cartesian product of the domains of the selected features. Further investigations can be conducted on making a more proper assumption on the domain, which should depend on the number of selected features.

Table 5 shows the average computational time of the proposed algorithm along with that of some representative algorithms (in seconds). F-MI and F-E [4] use the mutual information and entropy based measures, respectively. F-RS and F-PRS [24] adopts the rough set theory and the probabilistic rough set theory, respectively. W-SVM, W-5NN, W-DT and W-NB stand for the PSO-based wrapper approaches whose fitness function is calculated by the corresponding classification methods. It can be seen that the proposed algorithm is much faster than almost all the compared algorithms. The computational time is independent of the α value. It is only slightly slower than only F-MI, but it can improve the classification performance over using all features more frequently than F-MI [4].

6. CONCLUSIONS AND FUTURE WORK

Table 4: The results of 1-NN, Random Forest (RF), J48, and Naïve Bayes (NB) classifiers under the *independent* assumption.

Dataset	α	1-NN		RF		J48		NB		Time
		Accuracy	$\lvert\mathcal{X}^s\rvert$	Accuracy	$\lvert\mathcal{X}^s\rvert$	Accuracy	$\lvert\mathcal{X}^s\rvert$	Accuracy	$\lvert\mathcal{X}^s\rvert$	(s)
Chess	all	95.40	36	**99.06**	36	99.34	36	83.76	36	-
	0	97.02 ± 0.29	26.4	98.95 ± 0.18	26.4	99.28 ± 0.23	25.3	93.67 ± 0.24	6.6	55.7
	0.01	**97.35 ± 0.06**	9.0	97.56 ± 0.01	9.0	97.56 ± 0.01	9.0	93.74 ± 0.15	7.7	37.4
	0.1	94.37 ± 0.00	6.0	94.37 ± 0.00	6.0	94.37 ± 0.00	6.0	**93.90 ± 0.00**	6.0	32.4
	1	93.98 ± 0.18	4.2	93.98 ± 0.18	4.2	93.98 ± 0.18	4.2	93.82 ± 0.25	4.2	28.5
	10	90.33 ± 0.00	3.0	90.33 ± 0.00	3.0	90.33 ± 0.00	3.0	90.33 ± 0.00	3.0	27.5
	100	77.46 ± 0.00	2.0	77.46 ± 0.00	2.0	77.46 ± 0.00	2.0	77.46 ± 0.00	2.0	24.3
Lung	all	63.64	56	81.82	56	72.73	56	72.73	56	-
	0	45.45 ± 0.00	4.0	45.45 ± 0.00	4.0	36.36 ± 0.00	4.0	45.45 ± 0.00	4.0	0.5
	0.01	81.82 ± 0.00	2.3	81.82 ± 0.00	2.3	**81.82 ± 0.00**	2.3	45.45 ± 0.00	4.6	0.5
	0.1	81.82 ± 0.00	3.0	81.82 ± 0.00	3.0	**81.82 ± 0.00**	3.0	**90.91 ± 0.00**	3.0	0.5
	1	**90.91 ± 0.00**	2.0	**90.91 ± 0.00**	2.0	72.73 ± 0.00	2.0	**90.91 ± 0.00**	2.0	0.5
	10	72.73 ± 0.00	1.0	72.73 ± 0.00	1.0	72.73 ± 0.00	1.0	72.73 ± 0.00	1.0	0.5
	100	72.73 ± 0.00	1.0	72.73 ± 0.00	1.0	72.73 ± 0.00	1.0	72.73 ± 0.00	1.0	0.6
Lymph	all	63.27	18	**85.71**	18	75.51	18	**85.71**	18	-
	0	72.14 ± 5.59	9.4	76.79 ± 3.51	9.4	77.86 ± 1.35	8.6	80.05 ± 3.08	6.4	0.8
	0.01	**77.55 ± 0.00**	4.0	75.51 ± 0.00	4.0	73.47 ± 0.00	4.0	79.59 ± 0.00	4.0	0.7
	0.1	75.51 ± 0.00	2.0	73.47 ± 0.00	2.0	**81.63 ± 0.00**	2.0	79.59 ± 0.00	2.0	0.6
	1	73.47 ± 0.00	2.0	73.47 ± 0.00	2.0	73.47 ± 0.00	2.0	73.47 ± 0.00	2.0	0.6
	10	69.39 ± 0.00	1.0	69.39 ± 0.00	1.0	69.39 ± 0.00	1.0	69.39 ± 0.00	1.0	0.6
	100	69.39 ± 0.00	1.0	69.39 ± 0.00	1.0	69.39 ± 0.00	1.0	69.39 ± 0.00	1.0	0.6
Led.	all	79.88	24	100.00	24	100.00	24	100.00	24	-
	0	**100.00 ± 0.00**	6.0	100.00 ± 0.00	6.0	100.00 ± 0.00	6.0	100.00 ± 0.00	6.0	6.4
	0.01	**100.00 ± 0.00**	5.0	100.00 ± 0.00	5.0	100.00 ± 0.00	5.0	100.00 ± 0.00	5.0	6.1
	0.1	**100.00 ± 0.00**	5.0	100.00 ± 0.00	5.0	100.00 ± 0.00	5.0	100.00 ± 0.00	5.0	5.9
	1	64.35 ± 0.32	3.0	64.35 ± 0.32	3.0	64.35 ± 0.32	3.0	64.35 ± 0.32	3.0	5.3
	10	45.65 ± 0.00	2.0	45.65 ± 0.00	2.0	45.65 ± 0.00	2.0	45.65 ± 0.00	2.0	5.4
	100	23.42 ± 0.00	1.0	23.12 ± 0.00	1.0	23.42 ± 0.00	1.0	23.42 ± 0.00	1.0	4.8
Mush.	all	100.00	22	100.00	22	**100.00**	22	95.75	22	-
	0	100.00 ± 0.00	6.7	100.00 ± 0.00	6.7	99.89 ± 0.04	5.8	**97.50 ± 0.00**	1.8	32.9
	0.01	99.73 ± 0.00	4.1	99.73 ± 0.00	4.1	99.73 ± 0.00	4.1	**97.50 ± 0.00**	2.0	31.9
	0.1	99.63 ± 0.00	5.0	99.63 ± 0.00	5.0	99.63 ± 0.00	5.0	**97.50 ± 0.00**	1.0	30.5
	1	98.20 ± 0.78	3.2	98.20 ± 0.78	3.2	98.20 ± 0.78	3.2	**97.50 ± 0.00**	2.3	30.1
	10	97.50 ± 0.00	1.0	97.50 ± 0.00	1.0	97.50 ± 0.00	1.0	**97.50 ± 0.00**	1.0	32.6
	100	97.50 ± 0.00	2.0	97.50 ± 0.00	2.0	97.50 ± 0.00	2.0	**97.50 ± 0.00**	2.0	28.3
Spect	all	64.04	22	64.04	22	68.54	22	67.42	22	-
	0	62.39 ± 0.95	16.9	63.82 ± 0.46	16.9	70.25 ± 1.78	14.5	70.03 ± 2.64	8.3	2.1
	0.01	68.40 ± 0.73	8.0	63.51 ± 2.53	8.0	69.35 ± 1.11	7.2	71.54 ± 1.31	6.3	1.8
	0.1	67.84 ± 1.29	5.9	66.04 ± 1.77	5.9	**73.90 ± 1.33**	5.9	71.97 ± 2.14	4.7	1.5
	1	61.80 ± 0.00	3.0	61.80 ± 0.00	3.0	67.42 ± 0.00	3.0	67.42 ± 0.00	3.0	1.3
	10	**73.03 ± 0.00**	1.0	**73.03 ± 0.00**	1.0	73.03 ± 0.00	1.0	**73.03 ± 0.00**	1.0	1.2
	100	**73.03 ± 0.00**	1.0	**73.03 ± 0.00**	1.0	73.03 ± 0.00	1.0	**73.03 ± 0.00**	1.0	1.2
Splice	all	63.97	60	**94.83**	60	92.94	60	**92.10**	60	-
	0	76.81 ± 2.56	9.6	90.06 ± 3.56	9.6	**93.73 ± 0.86**	8.2	89.38 ± 0.74	7.0	71.0
	0.01	**88.90 ± 0.00**	4.0	89.28 ± 0.00	4.0	89.09 ± 0.00	4.0	88.90 ± 0.00	4.0	63.8
	0.1	80.27 ± 0.16	3.0	80.27 ± 0.16	3.0	80.36 ± 0.15	3.0	80.39 ± 0.30	3.0	60.6
	1	74.79 ± 0.00	2.0	74.79 ± 0.00	2.0	74.79 ± 0.00	2.0	71.50 ± 0.00	2.0	53.9
	10	62.94 ± 0.00	1.0	62.94 ± 0.00	1.0	62.94 ± 0.00	1.0	62.94 ± 0.00	1.0	49.9
	100	62.94 ± 0.00	1.0	62.94 ± 0.00	1.0	62.94 ± 0.00	1.0	62.94 ± 0.00	1.0	45.2
Iono.	all	87.18	34	**94.87**	34	87.18	34	81.20	34	-
	0	86.39 ± 2.13	5.2	92.71 ± 2.08	5.2	87.78 ± 1.12	4.4	**89.08 ± 1.12**	3.6	3.0
	0.01	**91.45 ± 0.00**	2.2	89.94 ± 0.36	2.2	**90.60 ± 0.00**	2.2	87.18 ± 0.00	2.0	2.8
	0.1	85.09 ± 1.37	2.9	89.04 ± 2.51	2.9	89.62 ± 0.46	2.9	87.18 ± 0.00	2.9	2.7
	1	76.07 ± 0.00	2.0	76.07 ± 0.00	2.0	83.76 ± 0.00	2.0	82.05 ± 0.00	2.0	2.6
	10	72.65 ± 0.00	1.1	72.65 ± 0.00	1.1	72.65 ± 0.00	1.1	72.65 ± 0.00	1.1	2.6
	100	64.10 ± 0.00	1.0	64.10 ± 0.00	1.0	64.10 ± 0.00	1.0	64.10 ± 0.00	1.0	2.5

· "Accuracy" and "$\lvert\mathcal{X}^s\rvert$ stand for the test accuracy and number of selected features. "Time (s)" is the computational time for feature selection in seconds. "all" means using all the features.
· For each problem and classifier, t-test with the significance level of 0.05 was conducted between the accuracies of the best α value(s) and all the features. The significantly better one is marked in bold.

In this paper, we investigate the issues of using entropy measures in feature selection, and address the significant generalizability issues by designing a bi-objective optimization model with a novel *Bayesian entropy* measure, which is defined based on Bayesian inference. The bi-objective optimization model is then solved by NSGA-II. The experimental studies show that the generalizability issues were addressed well, and the proposed algorithm managed to select a small subset of features and led to a better test classification accuracy than using the entire feature set on most of the selected discrete datasets within a much shorter time.

There are two main future directions that will be done based on this work. First, due to the nature of entropy, the proposed algorithm cannot classify the test instances whose feature values are unseen before. This issue will be addressed by introducing the distance between instances in the entropy computation, so that the nearby instances can contribute to the entropy as well. Second, the proposed algorithm is to be extended from discrete dataset to continuous ones. This can be done by proper discretization or definition of continuous entropy measures.

7. REFERENCES

[1] K. Bache and M. Lichman. Uci machine learning repository, 2013.

Table 5: Average computational time of the compared algorithms (in seconds).

	Lymph	Mush.	Spect	Led.	Iono.	Chess	Lung	Splice
$\alpha = 0$	0.78	32.88	2.12	6.40	3.00	55.75	0.55	70.97
$\alpha = 0.01$	0.84	39.03	2.42	7.42	3.25	66.21	0.58	82.76
$\alpha = 0.1$	0.80	35.41	2.25	6.67	3.10	60.18	0.57	75.31
$\alpha = 1$	0.82	36.82	2.36	6.90	3.25	61.67	0.56	78.08
$\alpha = 10$	0.78	35.88	2.16	6.76	3.05	58.86	0.54	74.24
$\alpha = 100$	0.76	34.03	2.03	6.38	3.01	50.45	0.55	70.82
F-MI	0.05	0.05	0.05	0.06	0.07	0.09	0.15	0.18
F-E	2.88	97.7	8.64	27.95	9.85	256.57	2.96	236.42
F-RS	2.07	2485.61	8.21	55.3	14.81	1372.93	0.69	928.25
F-PRS	2.86	2766.29	8.28	38.36	9.95	1827.06	0.68	911.3
W-SVM	24.41	5143.18	53.28	270.64	118.37	2441.21	5.4	10937.87
W-5NN	6.12	9311.59	18.89	264.51	72.72	4095.07	1.68	1936.67
W-DT	5.19	189.43	10.53	43.15	47.87	244.55	3.82	529.7
W-NB	13.46	304.08	15.89	150.37	19.42	377.24	4.13	706.23

[2] L. Bobrowski. Feature selection based on some homogeneity coefficient. In *9th International Conference on Pattern Recognition*, pages 544–546. IEEE, 1988.

[3] L. Cervante, B. Xue, L. Shang, and M. Zhang. A multi-objective feature selection approach based on binary PSO and rough set theory. In *13th European Conference on Evolutionary Computation in Combinatorial Optimization (EvoCOP)*, volume 7832 of *Lecture Notes in Computer Science*, pages 25–36. Springer, 2013.

[4] L. Cervante, B. Xue, M. Zhang, and L. Shang. Binary particle swarm optimisation for feature selection: A filter based approach. In *IEEE Congress on Evolutionary Computation (CEC'12)*, pages 881–888, 2012.

[5] M. Dash and H. Liu. Feature selection for classification. *Intelligent Data Analysis*, 1(4):131–156, 1997.

[6] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan. A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.

[7] R. Diao and Q. Shen. Feature selection with harmony search. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 42(6):1509–1523, 2012.

[8] P. Estévez, M. Tesmer, C. Perez, J. M. Zurada, et al. Normalized mutual information feature selection. *IEEE Transactions on Neural Networks*, 20(2):189–201, 2009.

[9] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *The Journal of Machine Learning Research*, 3:1157–1182, 2003.

[10] E. Hancer, B. Xue, D. Karaboga, and M. Zhang. A binary ABC algorithm based on advanced similarity scheme for feature selection. *Applied Soft Computing*, 36:334 – 348, 2015.

[11] M. M. Kabir, M. Shahjahan, and K. Murase. A new hybrid ant colony optimization algorithm for feature selection. *Expert Systems with Applications*, 39(3):3747–3763, 2012.

[12] S. S. Kannan and N. Ramaraj. A novel hybrid feature selection via symmetrical uncertainty ranking based local memetic search algorithm. *Knowledge-Based Systems*, 23(6):580–585, 2010.

[13] R. Kohavi and G. H. John. Wrappers for feature subset selection. *Artificial intelligence*, 97(1):273–324, 1997.

[14] P. P. Kundu and S. Mitra. Multi-objective optimization of shared nearest neighbor similarity for feature selection. *Applied Soft Computing*, 37:751 – 762, 2015.

[15] H. Liu, H. Motoda, R. Setiono, and Z. Zhao. Feature selection: An ever evolving frontier in data mining. In *Feature Selection for Data Mining*, volume 10 of *JMLR Proceedings*, pages 4–13. JMLR.org, 2010.

[16] H. Liu and L. Yu. Toward integrating feature selection algorithms for classification and clustering. *IEEE Transactions on Knowledge and Data Engineering*, 17(4):491–502, 2005.

[17] H. B. Nguyen, B. Xue, I. Liu, and M. Zhang. Filter based backward elimination in wrapper based PSO for feature selection in classification. In *IEEE Congress on Evolutionary Computation (CEC)*, pages 3111 – 3118, 2014.

[18] H. Peng, F. Long, and C. Ding. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1226–1238, 2005.

[19] C. Shannon and W. Weaver. *The Mathematical Theory of Communication*. Urbana:The University of Illinois Press, 1949.

[20] S. S. Shreem, S. Abdullah, and M. Z. A. Nazri. Hybrid feature selection algorithm using symmetrical uncertainty and a harmony search algorithm. *International Journal of Systems Science*, (0):1–18, 2014.

[21] N. Spolaôr, A. Lorena, and H. Lee. Multi-objective genetic algorithm evaluation in feature selection. In *Evolutionary Multi-Criterion Optimization*, volume 6576 of *Lecture Notes in Computer Science*, pages 462–476. Springer Berlin Heidelberg, 2011.

[22] R. W. Swiniarski and A. Skowron. Rough set methods in feature selection and recognition. *Pattern recognition letters*, 24(6):833–849, 2003.

[23] A. Unler and R. B. C. Alper Murat. mr2PSO: A maximum relevance minimum redundancy feature selection method based on swarm intelligence for support vector machine classification. *Information Sciences*, 20:4625–4641, 2011.

[24] B. Xue, L. Cervante, L. Shang, W. N. Browne, and M. Zhang. Binary PSO and rough set theory for feature selection: A multi-objective filter based approach. *International Journal of Computational Intelligence and Applications*, 13(02):1450009:1–34, 2014.

[25] B. Xue, M. Zhang, W. Browne, and X. Yao. A survey on evolutionary computation approaches to feature selection. *IEEE Transactions on Evolutionary Computation*, (99):1–1, doi:10.1109/TEVC.2015.2504420, 2015.

[26] B. Xue, M. Zhang, and W. N. Browne. Particle swarm optimization for feature selection in classification: A multi-objective approach. *IEEE Transactions on Cybernetics*, 43(6):1656–1671, 2013.

[27] L. Yu and H. Liu. Feature selection for high-dimensional data: A fast correlation-based filter solution. In *ICML*, volume 3, pages 856–863, 2003.

[28] Z. Zhu, S. Jia, and Z. Ji. Towards a memetic feature selection paradigm [application notes]. *IEEE Computational Intelligence Magazine*, 5(2):41–53, 2010.