# Mutual Information Estimation for Filter Based Feature Selection Using Particle Swarm Optimization

Hoai Bach Nguyen *, Bing Xue, and Peter Andreae

School of Engineering and Computer Science
Victoria University of Wellington
{Hoai.Bach.Nguyen, Bing.Xue, Peter.Andreae}@ecs.vuw.ac.nz

**Abstract.** Feature selection is a pre-processing step in classification, which selects a small set of important features to improve the classification performance and efficiency. Mutual information is very popular in feature selection because it is able to detect non-linear relationship between features. However the existing mutual information approaches only consider two-way interaction between features. In addition, in most methods, mutual information is calculated by a counting approach, which may lead to an inaccurate results. This paper proposes a filter feature selection algorithm based on particle swarm optimization (PSO) named PSOMIE, which employs a novel fitness function using nearest neighbor mutual information estimation (NNE) to measure the quality of a feature set. PSOMIE is compared with using all features and two traditional feature selection approaches. The experiment results show that the mutual information estimation successfully guides PSO to search for a small number of features while maintaining or improving the classification performance over using all features and the traditional feature selection methods. In addition, PSOMIE provides a strong consistency between training and test results, which may be used to avoid overfitting problem.

**Keywords:** Feature selection, mutual information estimation, particle swarm optimization

## 1 Introduction

A feature refers to a property of an object. In classification problems, each instance in a dataset is a set of values, which are assigned to the instance's features. These values will be used by a classification algorithm to determine which category or class the instance belongs to. A set of instances is used to train the classification algorithm, which is called a training set. However, in many classification problems, a large number of features are used to describe the instances. Due to "the curse of dimensionality", the larger a set of features is, the more difficult the training is and the longer the training time may take. In addition, not

* Corresponding Author

all features provide useful information. Some features have no or little relevance to the class labels, which blur useful information from other features [1]. Such features may lead to classification performance reduction. Also, some features may provide the same information as other features, and therefore do not improve the classification performance but result in a longer training time. In order to reduce the number of features, two feature reduction approaches, including feature selection and feature construction, are proposed. Feature construction constructs a small number of new high-level features while feature selection [2] reduces the size of the feature set by removing irrelevant and redundant features, which hopefully maintains or even increases the classification performance compared with using all features. This paper focuses mainly on feature selection in classification.

Feature selection is a difficult task due to the complex interaction between features. For example, a weakly relevant feature, which may not individually provide useful information to determine the class label, can significantly improve the classification performance when used with other features. Furthermore, an individually relevant feature may become redundant when working with others. Another reason, which makes feature selection become a challenge task, is the large search space, where the search space's size grows exponentially with respect to the number of features. Suppose there are $n$ original features, then the total number of possible subsets is $2^n$. Therefore, the exhaustive search is too slow to perform over the large search space in most situations. In order to reduce the searching time, some greedy algorithms such as sequential forward selection [3] and sequential backward selection [4] are developed. However, these methods usually do not guarantee to find optimal solutions due to getting stuck at local optima. Evolutionary computation (EC) algorithms such as genetic programming (GP) [5], genetic algorithms (GAs) [6] or particle swarm optimization (PSO) [7] are considered global optimization methods, which are suitable for a problem with large search space like feature selection. Therefore, EC have been widely applied to solve feature selection problems in recent years. PSO is chosen as the search technique for this work because it has a natural representation for feature selection, in which each original feature is represented by an entry of a particle's position. In addition, PSO is also simple and converges more quickly than other EC algorithms. In [8],it has been shown that, to achieve the same effectiveness, PSO is more efficient than GAs.

According to the evaluation criterion, existing feature selection methods can fall into two categories: wrapper and filter approaches [9, 10]. In a wrapper approach, a learning algorithm is used to calculate the fitness value of the selected features. Meanwhile, a filter approach is done in an independent way of learning algorithms. Therefore, wrapper methods usually can achieve better classification accuracy than filter ones. However, wrappers may produce a feature subset with poor generality, which is only good for the wrapped classification algorithm. In addition, in comparison with wrappers, filter methods are usually less expensive in terms of the computation complexity. Nowadays, there are many filter measures for feature selection problems, for example fisher score [11], consistency

measure [12], correlation measure [13] and mutual information [14]. Among these measures, mutual information measure gains more attraction. The reason is that mutual information is fast and able to analyze the complex interaction between multiple features or between the class label and a set of features while most other filter measures like correlation coefficients mainly evaluate a pair of features or the class label and an individual feature. However, most existing mutual information based feature selection approaches consider two-way interactions between features and simply calculate probability distributions by counting instances, which results in an inaccurate mutual information. A solution for the above problem is using mutual information estimation [15], which is able to compute the mutual information between multiple features by an accurate estimation approach. However, mutual information estimation has never been used with any EC algorithm to solve feature selection problems. Therefore, this work will propose a new feature selection approach, which bases on PSO algorithm and mutual information estimation.

### 1.1 Goals

The overall goal of this paper is to propose a PSO based filter feature selection approach to evolve a small set of features, which achieves similar or better classification performance than using all features. To achieve this goal, a new fitness function is proposed, which is inspired by the nearest neighbor estimation for mutual information [16]. Specifically, we will investigate:

- whether the proposed feature selection approach (named PSOMIE) can select a small number of features and maintain or even improve the classification performance over using all features.
- whether PSOMIE can maintain or improve the classification accuracy than two traditional feature selection approaches, filter sequential forward and backward feature selection [4, 3].

## 2 Background

### 2.1 Particle Swarm optimization (PSO)

In 1995, Kennedy and Eberhart [17] proposed an EC technique, named PSO. Like other swarm intelligence algorithms, PSO maintains a set of particles, in which each particle represents a candidate solution for an optimization problem. The behaviour of the swarm in PSO originates from social behaviours such as bird flocking and fish schooling. In particular, each particle is guided by its own best experience, called *pbest* and its neighbors best position so far, called *gbest*, to explore the search space. The current position of a particle $i$ is encoded as a vector $x_i = (x_{i1}, x_{i2}, \ldots, x_{iD})$, where $D$ is the dimensionality of the search space. Particle $i$ moves in the search space by using a velocity, which is defined by a vector $v_i = (v_{i1}, v_{i2}, \ldots, v_{iD})$. In PSO, each velocity component is limited by a predefined maximum velocity, called $v_{max}$, and $v_{id} \in [-v_{max}, v_{max}]$. The position and velocity of particle $i$ are updated according to the following equations:

$$v_{id}^{t+1} = w * v_{id}^{t} + c_1 * r_{i1} * (p_{id} - x_{id}^{t}) + c_2 * r_{i2} * (p_{gd} - x_{id}^{t}) \quad (1)$$

$$x_{id}^{t+1} = x_{id}^{t} + v_{id}^{t+1} \quad (2)$$

where $t$ denotes the $t^{th}$ iteration in the search process, $d$ is the $d^{th}$ dimension in the search space, $w$ is inertia weight, $c_1$ and $c_2$ are acceleration constants, $r_{i1}$ and $r_{i2}$ are random values uniformly distributed in [0,1], $p_{id}$ and $p_{gd}$ represent the position entry of *pbest* and *gbest* in the $d^{th}$ dimension, respectively.

### 2.2 Mutual Information

**Basic Concepts:** Entropy and mutual information are two well-known concepts in information theory [18], which are used to measure the information provided by random variables. Let $X$ be a discrete variable, then its uncertainty can be measured by entropy $H(X)$ defined as:

$$H(X) = -\sum_{x \in X} P(X = x) * \log_2 P(X = x) \quad (3)$$

Joint entropy is used to measure the uncertainty of a joint variable, which consists of two random variables $X$ and $Y$. Joint entropy $H(X, Y)$ is defined as:

$$H(X, Y) = -\sum_{x \in X, y \in Y} p(x, y) * \log_2 p(x, y) \quad (4)$$

where $p(x, y) = P(X = x, Y = y)$

When a variable is known and the other is unknown, the remaining uncertainty is measured by the conditional entropy as below

$$H(X|Y) = -\sum_{x \in X, y \in Y} p(x, y) * \log_2 p(x|y) \quad (5)$$

where $p(x|y) = P(X = x|Y = y)$.

Mutual information is a measure of shared information between two random variables. Mutual information between two random variables $X$ and $Y$ can be defined as

$$\begin{aligned} MI(X;Y) &= H(X) + H(Y) - H(X, Y) \\ &= H(X) - H(X|Y) \\ &= H(Y) - H(Y|X) \\ &= -\sum_{x \in X, y \in Y} p(x, y) * \log_2 \frac{p(x, y)}{p(x)p(y)} \end{aligned} \quad (6)$$

Mutual information is very popular in feature selection problems because it is able to detect non-linear relationship between features. According to Eq. (6), if two variables $X$ and $Y$ have a strong relationship, their mutual information $MI(X;Y)$ will be large. In contrast, if $X$ and $Y$ are totally independent then $MI(X;Y) = 0$. Mutual information is also extended to measure the common

information between more than two random variables, which is called multi-information, defined as

$$MI(X_1; X_2; \ldots; X_n) = \sum_{i=1}^{n} H(X_i) - H(X_1, X_2, \ldots, X_n) \tag{7}$$

**Limitations of Current Work on Mutual Information for Feature Selection:** Most of feature selection approaches aim to select a set of features which are most relevant to the class labels and do not contain any redundant features. These goals can be achieved by using mutual information criteria. In particular, the most relevant set of features will share the most information with the class label. A non-redundant set of features will have the least mutual information between the features. These conditions are expressed in the Eq. 8.

$$F_{mi} = Red - Rel \tag{8}$$

where

$$Rel = MI(S; C)$$
$$Red = MI(s_1; s_2; \ldots; s_m)$$

where $S$ is a set of features and its size $|S| = m$, $s_i \in S$ and $C$ is the class label. *Rel* and *Red* measure the relevance of feature set $S$ and the redundancy between features in $S$.

A feature selection approach often aims to find a set of features which minimizes the fitness $F_{mi}$ shown in Eq. (8). However the existing mutual information criteria for feature selection just consider the relationship between a single feature with the class label or between a pair of features. In other words, these approaches only consider two-way interactions between features. This does not ensure an optimal feature set being evolved because the interactions between features are more complex than two-way interactions.

Another limitation of the current mutual information based feature selection approaches is how the mutual information is calculated. To induce the mutual information, the joint probability of multi-variables needs to be known. In most current approaches, the probability distribution is achieved by counting the number of instances in the training set. However, counting approaches can only be applied for discrete variables not continuous variables. Furthermore, even for discrete variables, when the number of variables is large, the value distribution in the training set will be sparse, which leads to an inaccurate probability and mutual information. This is one of the reasons why only two-way interactions between single features are normally considered. To overcome these limitations, mutual information estimator has been developed. Currently, there are many mutual information estimators, such as basic histogram [19], kernel estimator [20] or nearest neighbors-based estimators [16]. Among them, the nearest neighbors-based estimators (NNE) has only one parameter and achieves the most accurate and consistent results with an independent hypothesis [21]. Therefore this work will use NNE incorporated with PSO to achieve feature selection.

### 2.3 Existing Feature Selection Approaches

**Traditional Feature Selection Methods:** A heuristic search, named Sequential Forward Selection (SFS) is proposed by Whitney [3], which starts with an empty set of features. At each step, a single feature, which gives the best fitness value with current selected features, will be added permanently to the feature subset. This process will stop when there is no single feature which is able to improve the current fitness. Another heuristic search is proposed by Maril and Green [4], which is called Sequential Backward Selection (SBS). The search starts with a full set of features. At each step, a single feature, whose removal results in the best score, is permanently removed from current feature set. SBS terminates when removing any feature from current feature set does not lead to any fitness improvement. Although SBS and SFS achieve better performance than feature ranking methods, they still suffer from the "nesting" problem, in which once a feature is added (or removed) from the feature set, it cannot be removed (or added) later. More works can be seen from [9, 22].

**EC Approaches (Non-PSO) for Feature Selection:** EC algorithms have been applied to feature selection problems, such as GAs [23], GP [24, 25]. Sousa, et al. [26] proposed two ensemble GA-based feature selection approaches, where a set of classifiers are used together to evolve better solutions than a single classifier. The first algorithm is a simple filter approach, which uses Pearson correlation measure as the main criterion. In the second algorithm, the filter and wrapper measures are combined into a single fitness function. However, due to the complexity and time consuming process, only a proportion of population are evaluated by wrapper evaluation. This proportion is defined dynamically based on the similarity between two ranked lists by filter and wrapper measures. The experiments show that the proposed algorithms achieve better classification performance than the original GAs.

Two GP-based approaches are proposed by Bhowan, et al. [27] to evolve a set of features, which is used directly in the Watson system, an intelligent open-domain question answering system. The first approach extracts all features, which are used in the evolved best-of-run GP tree. The second approach considers all evolved trees. Particularly, from the set of GP trees, the top $T$ features with the most frequency are chosen as extracting features. The experiment results show that, the set of features selecting from the best GP tree can only work well when the number of selected features is small. Meanwhile, selecting top $T$ features from the whole set of trees produces good results on both small and large feature sets.

**PSO-based Feature Selection Methods:** Xue et al. [28] propose three new initialisation mechanisms, which mimic the sequential feature selection approach. While the small initialisation use about 10% of original features to initialize the particles, particles in the large initialisation are constructed based on 50% of original features. These two initialisation mechanisms are combined in the mixed initialisation, which use the small initialisation for most of particles and the large initialisation for the rest. The experimental results show that the new

initialisation and updating mechanisms led to smaller feature subsets with better classification performance than the standard PSO. Two PSO based filter feature selection algorithms are proposed in [29], where mutual information and entropy are used in the fitness function to evaluate the relevance and redundancy of the selected feature subset. The experiments show that the proposed methods significantly reduce the number of features whilst achieve similar or better classification than using all features. Butler-Yeoman et al. [30] proposed a hybrid filter-wrapper approach named FastPSO, which mainly uses two-way mutual information as a fitness measure. In addition, a wrapper evaluation is used to determine whether or not a *pbest* need to be updated. The experiment results show that FastPSO outperformed not only using all features but also achieved better classification performance than PSO with mutual information as a fitness measure. A comprehensive EC-based feature selection survey can be seen in [31].

However most existing mutual information based feature selection approaches only consider two-way interaction between features and use the counting approach to calculate the mutual information. Therefore, the investigation of using mutual information estimation with a EC technique is still an open issue and the work conducted in this paper is the first effort in this area.

## 3 Proposed Feature Selection Approach

The key ideas of our proposed approach are to use mutual information as the measure of solution quality, calculated using NNE and to use PSO to search for an optimal set of features. This approach not only deals with numeric datasets but also considers the interaction between multiple features. The details of NNE and how we use NNE and PSO to solve feature selection problems are shown in the following sections.

### 3.1 Nearest Neighbors-Based Mutual Information Estimation

In order to estimate the mutual information between variable sets, it is usually necessary to estimate the underlying probability densities, which is a hard task. In addition, because the underlying probability densities will then be used together to induce mutual information, an inaccurate probability distribution is likely to result in a more inaccurate estimation of mutual information. To overcome this problem, Kraskov's Nearest Neighbors-based mutual information Estimation (NNE) [16] directly estimates the mutual information by using nearest neighbors statistics instead of estimating probability densities. The main idea of NNE is that if the neighbors of an instance in $X$ space are similar to the neighbors of that instance in $Y$ space, then there must be a strong relationship between $X$ and $Y$, i.e. the mutual information between $X$ and $Y$ is high. This is true when $X$ and $Y$ are single variables or sets of variables. Therefore, this estimation can be applied for multi-variate mutual information.

The mutual information is calculated via an estimation of entropy. The NNE based entropy estimation of a single variable $X$ ($\hat{H}(X)$) is given by Eq. (9).

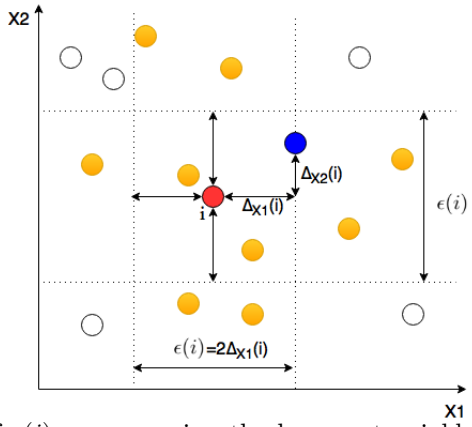


Fig. 1: Example of $\epsilon(i), n_{i1}, n_{i2}$ using the k-nearest neighbors distances, where $k$=3 and for the $i^{th}$ instance.

$$\hat{H}(X) = -\psi(k) + \psi(N) + \log c_d + \frac{d}{N} * \sum_{i=1}^{N} \log \epsilon_X(i) \tag{9}$$

where $\psi$ is the digamma function, $N$ is the total number of instances in the training set, $k$ is the number of nearest neighbors, $d$ is the dimensionality of variable $X$, $c_d$ is the volume of the $d$-dimensional unit ball, $\epsilon_X(i)$ is twice the distance from the $i^{th}$ instance to its $k^{th}$ nearest neighbor.

Given this entropy estimation, a multi-variate mutual information estimation ($\hat{MI}$) of a feature set $S = \{X_1, X_2, \ldots, X_m\}$ can be derived from Eq. (7), where the multi-variate mutual information is defined by Eq. (10).

$$\hat{MI}(X_1; X_2; \ldots; X_m) = \psi(k) - \frac{m-1}{k} + (m-1) * \psi(N) - \frac{1}{N} * \sum_{i=1}^{N} \sum_{j=1}^{m} n_{ij} \tag{10}$$

where $m$ is the number of single variables (features) in the variable (feature) set, $n_{ij}$ is the number of neighbors whose distance from the $i^{th}$ instance in the space specified by $X_j$ is not greater than $0.5 * \epsilon(i) = 0.5 * \max(\epsilon_{X_1}(i), \ldots, \epsilon_{X_m}(i))$.

An example of computing the $n_{ij}$ of the NNE is given in Fig. 1, where there are only 2 variables ($m = 2$) $X_1$ and $X_2$, the number of neighbors is set to $k = 3$ and the $i^{th}$ instance is marked by a red point. Firstly, the $3^{rd}$ nearest neighbor of the $i^{th}$ instance is found, which is marked by a blue point. The distances between the $i^{th}$ instance and its $3^{rd}$ nearest neighbor in each dimensions $X_1$ and $X_2$ are calculated, respectively, $\Delta_{X_1}(i)$ and $\Delta_{X_2}(i)$. In this case, $\Delta_{X_1}(i) > \Delta_{X_2}(i)$ so $\epsilon(i) = \epsilon_{X_1}(i) = 2 * \Delta_{X_1}(i)$. After that, for each dimension $X_1$ and $X_2$, the total number of neighbors whose distance in that dimension from the $i^{th}$ instance is not greater than $0.5 * \epsilon(i)$ are counted. In this case, $n_{i1} = 7$ and $n_{i2} = 6$. Given the $n_{ij}$, $\hat{MI}$ can be calculated using Eq. (10).

### 3.2 Mutual Information Estimation for Feature Selection

By using NNE, mutual information can be used to evaluate the relevance between a set of features and the class label, and the redundancy within a set of features even when the data is too sparse to give good estimates of the probability of density. The aims are to improve the classification performance via maximising the relevance between the set of features and the class label, and to reduce the number of selected features by minimising the redundancy within the set of features. To achieve the above objectives in a PSO search, a new fitness function for PSO is proposed, which is shown by the Eq. (11).

$$Fitness = (1 - \alpha) * Red - \alpha * Rel \tag{11}$$

where

$$Red = \hat{MI}(X_1; X_2; \ldots; X_m)$$
$$Rel = \hat{MI}(S; C)$$

where $S = \{X_1, X_2, \ldots, X_m\}$ is the set of selected features, $C$ is the class labels and $\alpha$ is a weight determining the contribution of *Red* and *Rel* in the fitness measure. Notice that in *Red*, the fitness function considers the multi-way interaction between features in $S$ and in *Rel*, the whole set of features is considered a single variable using NNE. It is easy to calculate redundancy measure by directly applying the mutual information estimation given in Eq. (10). However, the relevance measure is harder to compute because all features in $S$ are numeric variables while the class label is usually a categorical variable. In order to solve this problem, it is necessary to decompose $\hat{MI}(S; C)$, which is achieved by the Eq. (12).

$$\begin{aligned} \hat{MI}(S; C) &= \hat{H}(S) - \hat{H}(S|C) \\ &= \hat{H}(S) - \sum_{l=1}^{L} P(C = C_l) * \hat{H}(S|C_l) \end{aligned} \tag{12}$$

where $L$ is the number of classes and $C_l$ is the $l^{th}$ class.

In the above formula, $\hat{H}(S)$ is easily calculated by using Eq. (9). In order to calculate $\hat{H}(S|C_l)$, Eq. (9) is also applied but only for the instances which belong to class $C_l$.

### 3.3 The New Algorithm: PSOMIE

The algorithm based on our approach is called PSOMIE ("PSO with Mutual Information Estimation"). The representation of a particle in PSOMIE is a vector of $n$ real numbers, where $n$ is the total number of features. Each position entry $x_{id}$ falls in the range [0,1] and corresponds to the $d^{th}$ feature in the original feature set. A threshold $\theta$ is used to determine whether or not a feature is selected: if $x_{id} > \theta$ then the $d^{th}$ feature is selected, otherwise the $d^{th}$ feature is not selected.

Table 1: Datasets.

| Dataset | #features | #classes | #instances |
|---|---|---|---|
| Wine | 13 | 3 | 178 |
| Australian | 14 | 2 | 690 |
| Image Segmentation | 19 | 7 | 210 |
| Wall Robot | 24 | 4 | 5456 |
| Ionosphere | 34 | 2 | 351 |
| Lung | 56 | 2 | 32 |
| Musk1 | 166 | 2 | 476 |
| LSVT | 310 | 2 | 126 |
| Isolet5 | 617 | 5 | 1559 |
| Multiple Features | 649 | 10 | 2000 |

# 4 Experiment Design

## 4.1 Datasets

Ten datasets (Table 1) chosen from the UCI machine learning repository [32] are used in the experiments. These datasets have different numbers of features, classes and instances. Since currently mutual information estimations are mainly used for continuous variables, the data in the selected datasets are continuous values.

For each dataset, all instances are randomly divided into a training set and a test set, which contain 70% and 30% of the instances respectively. The algorithm firstly run on the training set to evolve a subset of features. After that the classification performance of the selected features will be calculated on both training and test set by K-nearest neighbors (KNN) classification algorithm where $K = 5$.

## 4.2 Parameter Settings

In the experiments, the parameters of PSO are set as follows [33]: $w = 0.7298, c_1 = c_2 = 1.49618, v_{max} = 6.0$, population size is 30, the maximum number of iterations is 100. The fully connected topology is used. There are three $\alpha$ values being tested in this work, which are $\alpha = 0.9, \alpha = 0.95, \alpha = 1.0$. For each $\alpha$ value, PSOMIE is ran 40 independent times on each dataset. The threshold $\theta$ in the continuous PSO is set to 0.6. The number of neighbours used in nearest neighbour mutual estimation $k$ is set to 4 [16]. A statistical significance test, Wilcoxon signed-rank test, is performed to compare between PSOMIE and using all features as well as two traditional feature selection approaches (SBS [4], SFS [3]). The significance level of the Wilcoxon test was set as 0.05.

# 5 Experiment Results

Experiment results are shown in Tables 3 and 4. Table 3 shows the comparison between PSOMIE and using all features, while Table 4 shows the comparison between PSOMIE and the two traditional methods. In these tables, "Full" means that all the original features are used for classification, "Ave-Size" stands for the

Table 2: Results of PSOMIE with different $\alpha$ values

| Dataset | Method | Ave-Size | Training Set | | | Test Set | | |
|---|---|---|---|---|---|---|---|---|
| | | | Best | Ave ± Std | Test | Best | Ave ± Std | Test |
| Wine | Full | 13.0 | 83.87 | | | 82.72 | | |
| | $\alpha$ =0.9 | 3.68 | 96.77 | 93.95±3.95 | + | 96.30 | 89.46±3.79 | + |
| | $\alpha$ =0.95 | 4.5 | 98.39 | 89.36±6.47 | + | 97.53 | 86.96±5.85 | + |
| | $\alpha$ =1.0 | 6.12 | 96.77 | 85.61±4.18 | + | 97.53 | 83.7±4.86 | = |
| Australian | Full | 14 | 77.85 | | | 67.15 | | |
| | $\alpha$ =0.9 | 4.98 | 88.82 | 79.92±2.4 | + | 80.68 | 71.28±3.32 | + |
| | $\alpha$ =0.95 | 4.92 | 83.64 | 79.71±1.77 | + | 75.36 | 70.13±2.37 | + |
| | $\alpha$ =1.0 | 5.48 | 83.02 | 79.73±1.82 | + | 75.36 | 70.87±2.0 | + |
| Image Segmentation | Full | 19 | 94.75 | | | 93.65 | | |
| | $\alpha$ =0.9 | 2.86 | 96.89 | 95.23±1.0 | + | 98.19 | 95.06±1.65 | + |
| | $\alpha$ =0.95 | 3.5 | 97.28 | 96.05±0.56 | + | 98.64 | 94.89±2.14 | = |
| | $\alpha$ =1.0 | 7.54 | 97.86 | 97.41±0.33 | + | 98.64 | 98.11±0.37 | + |
| Wall Robot | Full | 24 | 94.87 | | | 92.33 | | |
| | $\alpha$ =0.9 | 2.06 | 97.6 | 95.25±1.33 | = | 96.18 | 92.85±2.12 | = |
| | $\alpha$ =0.95 | 2.84 | 97.97 | 97.33±0.99 | + | 96.67 | 95.76±1.36 | + |
| | $\alpha$ =1.0 | 3.76 | 98.28 | 97.93±0.22 | + | 97.43 | 96.6±0.42 | + |
| Ionosphere | Full | 34 | 89.02 | | | 81.9 | | |
| | $\alpha$ =0.9 | 3.8 | 93.09 | 90.04±1.73 | + | 95.24 | 87.83±3.4 | + |
| | $\alpha$ =0.95 | 4.3 | 93.9 | 91.27±1.4 | + | 94.29 | 87.9±2.51 | + |
| | $\alpha$ =1.0 | 13.12 | 92.68 | 89.93±0.98 | + | 89.52 | 83.9±2.39 | + |

average number of selected features over the 40 runs. "Ave" and "Std" represents the average and standard deviation of the training or test accuracies over the 40 runs. "Test" represents the significant test comparing between PSOMIE and other approaches. "+", "=" or "-" mean that PSOMIE is respectively significantly better, similar or significantly worse than using all features or the traditional methods.

### 5.1 Comparison with All Features

According to the results shown in Table 3, in terms of test results, in almost all datasets, the number of features selected by PSOMIE is at least 70% lower than the total number of original features. In seven datasets, PSOMIE always achieves significantly better classification accuracy than using all features. For example, in the Ionosphere dataset, by selecting around 4 features from 34 original features, PSOMIE significantly increases the classification accuracy about 6% over using all features. In addition, in all datasets, the best solutions evolved by PSOMIE always outperform the set of original features in both classification performance as well as the number of selected features.

The results suggest that PSO with NNE for mutual information can significantly reduce the dimensionality of datasets, while maintains or even improves the classification accuracy over using all features.

### 5.2 Results of PSOMIE with Different $\alpha$

As can be seen from Table 3, in almost all datasets, with at least one of $\alpha$ value, despite of selecting a small number of features, PSOMIE is still able to achieve better classification accuracy than using all features. The only exception

Table 3: Results of PSOMIE with different $\alpha$ values

| Dataset | Method | Ave-Size | Training Set | | | Test Set | | |
|---|---|---|---|---|---|---|---|---|
| | | | Best | Ave ± Std | Test | Best | Ave ± Std | Test |
| Lung | Full | 56 | 86.36 | | | 90.0 | | |
| | $\alpha$ =0.9 | 15.96 | 90.91 | 82.91±7.53 | = | 100.0 | 95.0±8.06 | + |
| | $\alpha$ =0.95 | 16.54 | 90.91 | 83.18±5.77 | - | 100.0 | 93.2±9.47 | + |
| | $\alpha$ =1.0 | 17.64 | 90.91 | 81.55±6.19 | - | 100.0 | 93.6±12.77 | + |
| Musk1 | Full | 166 | 90.99 | | | 81.12 | | |
| | $\alpha$ =0.9 | 18.52 | 90.69 | 87.27±1.66 | - | 84.62 | 77.69±3.56 | - |
| | $\alpha$ =0.95 | 18.98 | 90.69 | 87.79±1.89 | - | 88.11 | 78.29±3.54 | - |
| | $\alpha$ =1.0 | 60.68 | 94.59 | 91.2±1.36 | = | 89.51 | 84.15±2.46 | + |
| LSVT | Full | 310 | 78.41 | | | 52.63 | | |
| | $\alpha$ =0.9 | 55.74 | 81.82 | 76.14±2.18 | - | 76.32 | 59.16±5.4 | + |
| | $\alpha$ =0.95 | 54.56 | 82.95 | 76.36±2.22 | - | 65.79 | 58.32±3.74 | + |
| | $\alpha$ =1.0 | 111.26 | 84.09 | 76.93±1.97 | - | 65.79 | 54.1±3.54 | + |
| Isolet5 | Full | 617 | 99.16 | | | 98.37 | | |
| | $\alpha$ =0.9 | 107.28 | 99.15 | 98.87±0.1 | - | 98.5 | 98.09±0.2 | - |
| | $\alpha$ =0.95 | 108.52 | 99.06 | 98.94±0.0 | - | 98.47 | 98.19±0.14 | - |
| | $\alpha$ =1.0 | 210.14 | 99.28 | 99.14±0.0 | = | 98.8 | 98.45±0.14 | + |
| Multiple Features | Full | 649 | 99.1 | | | 99.0 | | |
| | $\alpha$ =0.9 | 121.66 | 99.37 | 98.87±0.4 | - | 99.1 | 98.43±0.51 | - |
| | $\alpha$ =0.95 | 120.0 | 99.39 | 98.88±0.39 | - | 99.13 | 98.45±0.57 | - |
| | $\alpha$ =1.0 | 256.52 | 99.51 | 99.13±0.26 | = | 99.37 | 98.89±0.33 | - |

is the largest dataset, Multiple Features, which is not too surprising since filter approaches often can not scale well [34] and the classification accuracy is already very high (99.0%) when using all features. In Multiple Features, although the feature set selected by PSOMIE is not better than all features according to the significant test, PSOMIE can select around only 18.5% of the features and still achieve aroud 98.45% classification accuracy, which is only 0.55% lower than using all features. In addition, the best accuracy is better than using all features.

In these experiments, $\alpha$ is given 3 values, which corresponds to the contribution of relevant measure to the fitness function. When $\alpha$ is larger than 0.5, relevant measure will contribute more than redundant measure, which means that the searching process will focus more on improving classification accuracy than reducing the number of features. As can be seen from Table 3, on most datasets, when $\alpha$ is increased, PSOMIE tends to select more features and achieves better classification performance. In seven of the ten datasets, with $\alpha = 0.95$, PSOMIE only selects 16.7% of the original feature set and still achieves better classification than using all features. On the other hand, when $\alpha$ is assigned to 1.0, although the feature sets evolved by PSOMIE are larger than when $\alpha = 0.95$, these sets achieve better classification accuracy than using all features in nine of the ten cases.

The results suggest that different weights for the two fitness components, i.e. the relevant and redundant measures, have significant effect on the searching process. In other words, by setting a proper value for $\alpha$, PSOMIE is able to evolve a set of small number of features and achieve better classification accuracy than using all features.

Table 4: Comparison between PSOMIE and SFS, SBS

| Dataset | Method | Ave-Size | Ave | Std | Test | Time (ms) |
|---|---|---|---|---|---|---|
| Wine | PSOMIE | 4.5 | 86.96 | 5.85 | | 528 |
| | SFS | 5.0 | 81.48 | | + | 1 |
| | SBS | 5.0 | 81.48 | | + | 5 |
| Australian | PSOMIE | 4.9 | 70.13 | 2.37 | | 1883 |
| | SFS | 4.0 | 67.63 | | + | 1 |
| | SBS | 5.0 | 69.57 | | + | 4 |
| Image Segmentation | PSOMIE | 3.5 | 94.89 | 2.14 | | 685 |
| | SFS | 8.0 | 89.57 | | + | 1 |
| | SBS | 9.0 | 89.57 | | + | 5 |
| Wall Robot | PSOMIE | 2.8 | 95.76 | 2.12 | | 142605 |
| | SFS | 4.0 | 94.77 | | + | 1 |
| | SBS | 4.0 | 95.20 | | = | 3 |
| Ionosphere | PSOMIE | 4.3 | 87.90 | 2.51 | | 2427 |
| | SFS | 5.0 | 78.10 | | + | 1 |
| | SBS | 5.0 | 85.71 | | + | 4 |
| Lung | PSOMIE | 16.5 | 93.20 | 9.47 | | 952 |
| | SFS | 18.0 | 80.00 | | + | 8 |
| | SBS | 18.0 | 80.00 | | + | 5 |
| Musk1 | PSOMIE | 19.0 | 78.29 | 3.54 | | 61789 |
| | SFS | 2.0 | 62.94 | | + | 8 |
| | SBS | 3.0 | 62.24 | | + | 213 |
| LSVT | PSOMIE | 54.56 | 58.32 | 3.74 | | 19313 |
| | SFS | 3.0 | 76.32 | | - | 9 |
| | SBS | 3.0 | 60.53 | | - | 1599 |
| Isolet5 | PSOMIE | 108.5 | 98.19 | 0.14 | | 2981564 |
| | SFS | 13 | 95.83 | | + | 43 |
| | SBS | 28 | 95.33 | | + | 26419 |
| Multiple Features | PSOMIE | 120.0 | 98.45 | 0.57 | | 5063384 |
| | SFS | 15 | 96.93 | | + | 56 |
| | SBS | 91 | 99.17 | | - | 28466 |

### 5.3 Analysis between Training and Test Results

In classification tasks, overfitting is a very common problem. After a long training time, the training accuracy still increases while the test accuracy becomes worse. This is because the classifier may remember all specific properties of training instances. As can be seen from Table 3, in almost all datasets, the results of training and test sets are very consistent. In particular, a small increase/decrease of the classification accuracy on the training set usually corresponds to a small increase/decrease of the accuracy on the test set. This consistency suggests that mutual information estimation is able to extract a consistent pattern, which is effective in both training and test (unseen data) set.

The results show that, by using NNE for mutual information in PSO, the overfitting problem in feature selection might be avoided. With this property, the testing accuracy can be improved by advancing the accuracy on the training set, which does not happen in many feature selection algorithms.

### 5.4 Comparison with SFS and SBS

The comparison between PSOMIE and two traditional methods, SFS [3] and SBC [4] are shown in Table 4. These traditional methods use standard mutual information with the counting approach to calculate the redundancy and relevance, which are combined into a fitness function as shown in Eq. (11). However,

in the traditional methods, the counting approach is applied to calculate the mutual information. To ensure a fair comparison, the $\alpha$ value in the fitness function shown in Eq. (11) is set to 0.95 in both PSOMIE and traditional methods. Since SFS and SBS are deterministic methods, they produce a single solution on each dataset.

As can be seen from this table, in the first six datasets where the number of features is relatively small, although PSOMIE always selects a smaller number of features than both SFS and SBS, it still achieves better classification performance. For example, in Image Segmentation, the subset evolved by PSOMIE is three times smaller than the traditional methods, but the PSOMIE's subset is about 5% better than the sequential selection in terms of classification performance. In the other four large datasets, PSOMIE selects more features than the traditional methods and obtains higher classification accuracy in three of the four cases. The results suggest that PSOMIE is able to adapt with different datasets to balance between the number of selected features and the classification performance.

However, in terms of efficiency, PSOMIE is much more expensive than sequential approaches, but the performance of SBS and SFS can not be improved by giving longer time since they are deterministic methods and stop when the fitness value is not further improved. For PSOMIE, although the Ionosphere dataset has a larger number of features than Wall Robot, the selection time of Wall Robot is still longer. Similarly, despite of having a smaller number of features than the LSVT dataset, the Musk1 dataset needs longer training time. This is caused by the large number of instances. Particularly Musk1 and Wall Robot datasets have more instances than Ionosphere and LSVT datasets, respectively. Because for each instance, PSOMIE needs to calculate a distance between the current instance to another instances, the total cost for finding distance for all instances is $O(N^2)$ where $N$ is the total number of instances in the training set. After that, for each instance, there is another process which counts all neighbors which fall in a range around that instance in each dimension. Therefore, in order to estimate one mutual information, the worst cost will be $n * O(N^3)$ where $n$ is the total number of features. So the computation cost of mutual information estimation increases with respect to the number of instances in the training set.

## 6 Conclusions and Future Work

The goal of this paper is to investigate a PSO based filter feature selection approach, which uses nearest neighbor mutual information estimation to evolve a set of small number of features while maintaining or improving the classification performance over using all features. The experiment results show that PSOMIE substantially reduces the dimensionality of the datasets and achieves the similar or better classification performance than using all features. PSOMIE also produces a strong consistency between training and test accuracies. In addition, by using mutual information estimation, PSOMIE can better balance between

a smaller number of features in comparison with sequential feature selection approaches.

However, besides the strengths, PSOMIE still has several limitations, which will be addressed in our future work. Firstly, PSOMIE could not be applied to categorical datasets because NNE requires a numeric dataset. In order to solve this problem, a good distance measure needs to be developed, which can perform well on both numeric and categorical datasets. In addition, PSOMIE will have an expensive computation cost if there are a large number of instances in the training set. Therefore, developing an instance selection algorithm, which reduces the number of instances while maintaining most of important information, will lower the computation cost. In addition, scalability is a common issue in filter feature selection. So developing novel feature selection methods to solve feature selection tasks with thousands of features is also needed in future work. This work is investigates the use of mutual information estimation in standard PSO for feature selection. In the future, other advanced PSO algorithms, such as constriction factor version of PSO [35], and other methods will be investigated to improve the performance of filter feature selection on different problems.

## References


1. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. The Journal of Machine Learning Research **3** (2003) 1157–1182
2. Liu, H., Motoda, H., Setiono, R., Zhao, Z.: Feature selection: An ever evolving frontier in data mining. FSDM **10** (2010) 4–13
3. Whitney, A.W.: A direct method of nonparametric measurement selection. Computers, IEEE Transactions on **100**(9) (1971) 1100–1103
4. Marill, T., Green, D.M.: On the effectiveness of receptors in recognition systems. Information Theory, IEEE Transactions on **9**(1) (1963) 11–17
5. Nag, K., Pal, N.R.: A multiobjective genetic programming-based ensemble for simultaneous feature selection and classification. (2015)
6. Lin, F., Liang, D., Yeh, C.C., Huang, J.C.: Novel feature selection methods to financial distress prediction. Expert Systems with Applications **41**(5) (2014) 2472–2483
7. Chuang, L.Y., Chang, H.W., Tu, C.J., Yang, C.H.: Improved binary pso for feature selection using gene expression data. Computational Biology and Chemistry **32**(1) (2008) 29–38
8. Hassan, R., Cohanim, B., De Weck, O., Venter, G.: A comparison of particle swarm optimization and the genetic algorithm. In: Proceedings of the 1st AIAA multidisciplinary design optimization specialist conference. (2005) 1–13
9. Dash, M., Liu, H.: Feature selection for classification. Intelligent data analysis **1**(3) (1997) 131–156
10. Kohavi, R., John, G.H.: Wrappers for feature subset selection. Artificial intelligence **97**(1) (1997) 273–324
11. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern classification. John Wiley & Sons (2012)

12. Dash, M., Liu, H., Motoda, H.: Consistency based feature selection. In: Knowledge Discovery and Data Mining. Current Issues and New Applications. Springer (2000) 98–109
13. Hall, M.: Correlation-based feature selection for discrete and numeric class machine learning, proceedings of 7th intentional conference on machine learning, stanford university (2000)
14. Kononenko, I.: On biases in estimating multi-valued attributes. In: IJCAI. Volume 95., Citeseer (1995) 1034–1040
15. Walters-Williams, J., Li, Y.: Estimation of mutual information: A survey. In: Rough Sets and Knowledge Technology. Springer (2009) 389–396
16. Kraskov, A., Stögbauer, H., Grassberger, P.: Estimating mutual information. Physical review E **69**(6) (2004) 066138
17. Kennedy, J., Eberhart, R., et al.: Particle swarm optimization. In: Proceedings of IEEE international conference on neural networks. Volume 4., Perth, Australia (1995) 1942–1948
18. Jaynes, E.T.: Information theory and statistical mechanics. Physical review **106**(4) (1957) 620
19. Sturges, H.A.: The choice of a class interval. Journal of the American Statistical Association **21**(153) (1926) 65–66
20. Parzen, E.: On estimation of a probability density function and mode. The annals of mathematical statistics (1962) 1065–1076
21. Doquire, G., Verleysen, M.: A performance evaluation of mutual information estimators for multivariate feature selection. In: Pattern Recognition-Applications and Methods. Springer (2013) 51–63
22. Stearns, S.D.: On selecting features for pattern classifiers. In: Proceedings of the 3rd International Conference on Pattern Recognition (ICPR 1976), Coronado, CA (1976) 71–75
23. Zhu, Z., Ong, Y.S., Dash, M.: Wrapper–filter feature selection algorithm using a memetic framework. Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on **37**(1) (2007) 70–76
24. Neshatian, K., Zhang, M.: Genetic programming for feature subset ranking in binary classification problems. In: Genetic programming. Springer (2009) 121–132
25. Hunt, R., Neshatian, K., Zhang, M.: A genetic programming approach to hyper-heuristic feature selection. In: Simulated Evolution and Learning. Springer (2012) 320–330
26. Sousa, P., Cortez, P., Vaz, R., Rocha, M., Rio, M.: Email spam detection: A symbiotic feature selection approach fostered by evolutionary computation. International Journal of Information Technology & Decision Making **12**(04) (2013) 863–884
27. Bhowan, U., McCloskey, D.: Genetic programming for feature selection and question-answer ranking in ibm watson. In: Genetic Programming. Springer (2015) 153–166
28. Xue, B., Zhang, M., Browne, W.N.: Particle swarm optimisation for feature selection in classification: Novel initialisation and updating mechanisms. Applied Soft Computing **18** (2014) 261–276
29. Cervante, L., Xue, B., Zhang, M., Shang, L.: Binary particle swarm optimisation for feature selection: A filter based approach. In: Evolutionary Computation (CEC), 2012 IEEE Congress on, IEEE (2012) 1–8
30. Butler-Yeoman, T., Xue, B., Zhang, M.: Particle swarm optimisation for feature selection: A hybrid filter-wrapper approach. In: Evolutionary Computation (CEC), 2015 IEEE Congress on, IEEE (2015) 2428–2435

31. Xue, B., Zhang, M., Browne, W., Yao, X.: A survey on evolutionary computation approaches to feature selection
32. Asuncion, A., Newman, D.: Uci machine learning repository (2007)
33. Van Den Bergh, F.: An analysis of particle swarm optimizers. PhD thesis, University of Pretoria (2006)
34. Zhai, Y., Ong, Y.S., Tsang, I.W.: The emerging" big dimensionality". Computational Intelligence Magazine, IEEE **9**(3) (2014) 14–26
35. Eberhart, R.C., Shi, Y.: Comparing inertia weights and constriction factors in particle swarm optimization. In: Evolutionary Computation, 2000. Proceedings of the 2000 Congress on. Volume 1., IEEE (2000) 84–88