Information Sciences 318 (2015) 14–27

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Feature selection for clustering using instance-based learning by exploring the nearest and farthest neighbors

Chien-Hsing Chen

Department of Information Management, Ling Tung University, Taiwan

ARTICLE INFO

Article history:
Received 21 September 2011
Received in revised form 23 March 2015
Accepted 8 May 2015
Available online 14 May 2015

Keywords:
Feature selection
Instance-based learning
Neighbor
Mutual information
Clustering

ABSTRACT

Feature selection for clustering is an active research topic and is used to identify salient features that are helpful for data clustering. While partitioning a dataset into clusters, a data instance and its nearest neighbors will belong to the same cluster, and this instance and its farthest neighbors will belong to different clusters. We propose a new Feature Selection method to identify salient features that are useful for maintaining the instance's Nearest neighbors and Farthest neighbors (referred to here as FSNF). In particular, FSNF uses the mutual information criterion to estimate feature salience by considering maintainability. Experiments on benchmark datasets demonstrate the effectiveness of FSNF within the context of cluster analysis.

1. Introduction

The selection of salient features is an important issue in cluster analysis and is relevant for topics such as image representation [24], time-series prediction [45], machine learning [41] and natural language processing [5]. Typically, the use of a large number of features to represent a pattern is highly informative for a learning algorithm. However, in a high-dimensional dataset, some features are noisy; thus, learning algorithms are often biased by noisy features that affect the learning process. The goal of feature selection for clustering is usually to identify a subset of salient features from the original representation space; the identified salient features are helpful for data clustering that aims to maximize the between-cluster scatter and minimize the within-cluster scatter. A previous semi-supervised learning method [4] utilized the pairwise constraints between instances with class labels to identify salient features. Because the class labels would usually be unavailable in a real-world dataset, we consider an unsupervised learning method for the selection of salient features applied for data clustering.

Feature selection algorithms are classified into two primary categories: those based on the filter model and those based on the wrapper model [1,27]. The filter model requires an evaluator to measure the intrinsic characteristics of each feature. One of the most well-known criterion functions for evaluating features uses a relevance metric [37,41]; a feature that is either dependent (relevant, consistent, reliable, important or informative) on the target class label or conditionally independent of the other features is usually defined as a relevant (salient) feature. However, a feature subset selected by the filter model is problematic with respect to achieving the goal of feature selection for clustering because the number of clusters or the clustered structure cannot be effectively predicted in advance. This problem can be easily overcome when we have prior information implying that instances having the same class labels will belong to the same clusters; however, such information is unlikely to be available for a real-world dataset.

E-mail address: ktfive@gmail.com

http://dx.doi.org/10.1016/j.ins.2015.05.019
0020-0255/

The wrapper model requires a pre-specified classification (or clustering) learning algorithm trained on data instances and an evaluator to quantify the performance in terms of a generalization criterion (e.g., accuracy or between-cluster scatter). Using heuristic algorithms (e.g., forward or backward selection), the goal is to effectively visit the space of all possible feature subsets to identify the best feature subset, which is usually a local-optimal solution. However, the selected feature subset suffers from two biases. First, the feature subset depends strongly on the selection of the learning algorithms because different clustering algorithms produce differently shaped clusters [19] and favor different prior preferences (e.g., the specification of the number of clusters in *K*-means, visual surface inspection in a self-organizing map (SOM) and a hierarchy of clusters represented as a dendrogram in hierarchical clustering). We would be less likely to trust a feature subset selected by biased clusters, as calculated by an arbitrary clustering algorithm, when we lack prior information (such as natural shape, number of clusters and clustered structure), as is the case with a real dataset. Second, many studies based on the wrapper model for unsupervised learning partition a dataset into clusters in advance, using all of the features. Consequently, the selected feature subset identifies biased clusters due to their noisy features. Such shortcomings directly affect the flexibility of wrapper-based feature selection methods because the salient feature subset that results from a particular clustering algorithm may not be appropriate for use with another clustering algorithm.

In this paper, we present a new method to achieve the goal of feature selection for clustering without the need to explore the exact clustered information. Specifically, FSNF uses the mutual information criterion to identify salient features due to its robustness and popularity. The nearest and farthest neighbors help select the salient features; FSNF uses the mutual information criterion to assess features while considering these neighbors based on distinguishability and redundancy toward a robust statistical evaluator. FSNF then identifies a real-valued salience vector instead of heuristically visiting the space of all possible feature subsets. Once the salience vector, which is usually a local-optimal solution, is obtained, the salient features are used to perform clustering using learning algorithms.

The rest of this paper is organized as follows. Section 2 reviews work related to feature selection. Section 3 describes FSNF, which attempts to identify the best feature salience vector. Experimental results are presented in Section 4. Section 5 presents a discussion of the results and concluding remarks.

2. Related work

Feature selection for clustering is an active topic in the data-mining field. In general, feature selection algorithms are classified into two categories: those based on the filter model [8] and those based on the wrapper model [27]. A number of feature selection methods based on the wrapper model for classification [30,38] and clustering [52] have been proposed in the literature. In supervised learning, one can generally use the wrapper model to construct a classifier and use the criterion (e.g., accuracy) to observe how the features can positively predict class labels via this classifier [10,35,36,40]. Maldonado and Weber [31] introduced a new wrapper-based method in which the classifier was built using support vector machines with kernel functions. Su and Hsiao [42] developed a system for simultaneous multiclass classification and feature selection. Wang et al. [49] found the smallest set of genes that can ensure the highly accurate classification of cancers from microarray data using supervised machine learning algorithms. Additionally, various studies have identified salient features that are class-dependent [29,33,37] or class-separable [48,54] features. Nicolas et al. [11] proposed a new method for instance and feature selection based on supervised learning.

Class labels may be absent in a real-world dataset. Consequently, the wrapper model is integral to the field of unsupervised learning, and nonparametric techniques can be applied to many interesting problems [44]. Yang et al. [52] presented a feature selection method for selecting features by minimizing the Bayes error rate estimated with a nonparametric estimator. Lin et al. [25] also introduced a nonparametric technique for feature screening. Chen [4] developed a nonparametric technique using pairwise instance constraints to supervise the feature selection process.

Another option is to develop a clustering algorithm to partition a dataset into clusters and use internal or relative cluster validity as the criterion to observe whether a feature is informative with respect to the clustered information [23]. Chow et al. [7] proposed a selection method that considered the compactness and separation of clusters. Huang et al. [17] proposed a feature co-selection method for Web document clustering in which the clustering results in one type of feature space helped identify salient features in other types of feature spaces. Li et al. [22] proposed a new text-clustering method with feature selection and extended the chi-square term-category independence test to measure whether the dependency between a term and a cluster was positive or negative. Sanguinetti [39] presented a latent variable model to perform dimensionality reduction on a dataset that contained clusters; specifically, a variable was considered salient when it preserved clustered information by mapping an original representation space to a latent space. Mitra [32] proposed using structural similarity between clusters for feature selection, where the topological neighborhood information about pairs of instances was considered to assess the similarity. Furthermore, a feature selection method based on the wrapper model in semi-supervised feature selection, considering labeled and unlabeled examples, has also been described [50,53].

The filter model does not require a clustering (or classification) learning algorithm to provide cluster information in terms of cluster shape and number of clusters. It does require an evaluator to measure the intrinsic characteristics of each feature. Han et al. [14] presented a new criterion function that identifies features pertinent to the classification task at a very low computational cost. Chen [3] selected salient features using compactness and separability. Peng et al. [37] developed criteria based on mutual information for feature selection. Recently, a hybrid model that captures the advantages of the filter and wrapper models was developed to address the above-mentioned computational issue [27,55].

In summary, FSNF has the potential to overcome the above-mentioned shortcomings of filter- and wrapper-based feature selection for clustering. The filter-based model cannot predict cluster information effectively regardless of the learning algorithm used. The wrapper-based model must explore the exactly clustered information using a predetermined clustering algorithm. Therefore, FSNF may better achieve the goal of feature selection for clustering.

3. The proposed method

3.1. Observations concerning unsupervised feature selection

We present an example in which we have a set of instances. Each instance has two dimensions, "Feature 1" and "Feature 2" (Fig. 1), and all of the instances can be clustered into two assumptive clusters, "Cluster 1" and "Cluster 2." For this example, we discuss two methods (FSNF and a benchmark wrapper-based method) for extracting salient features for clustering.

First, we briefly introduce FSNF for determining a salient feature. This method begins with the instance "$\mathbf{x}_1$," which has specific nearest and farthest neighbors (Fig. 1). We define feature separability as the average distance from an instance to its farthest neighbors (its magnitude is represented by a dotted line) and feature compactness as the average distance from this instance to its nearest neighbors (its magnitude is represented by a solid line), where both distances are measured with respect to a particular feature. We assume that a feature is more salient if it yields a higher value of the following measure:

$$\frac{\#\{\text{average length of the part of the line corresponding to the separability}\}}{\#\{\text{average length of the part of the line corresponding to the compactness}\}}$$

Therefore, based on this assumption, "Feature 1" is more salient. When partitioning the dataset into clusters, we would be much more likely to believe that the contribution of "Feature 1" is greater than that of "Feature 2."

Second, we consider another model that is typically a basic wrapper model for unsupervised learning. In particular, we apply a clustering algorithm (e.g., *K*-means, where *K* is set to 2) to this dataset and practically obtain the two assumptive clusters. Then, we use a criterion function (e.g., entropy, a well-known measure in information theory) to evaluate which feature better informs the obtained clusters. We observe that "Feature 1" would be much more salient than "Feature 2." Similar studies can be found in the literature [7,17,22,50].

These two methods are similar in that they can each correctly recognize the salient feature. Importantly, however, FSNF can find salient features without the need for a clustering algorithm, and it does not need to precisely know the actual clusters.

For the sake of implementing FSNF, two major problems require solutions. First, the informal formula mentioned above appears to be an appropriate and straightforward evaluator; however, the formula cannot effectively measure salient or redundant features for scenarios with high data dimensionality. Second, the salient features are dynamic; not every instance will result in the same salient features. We should therefore search for the best salient features, i.e., those features that can effectively achieve the goal of feature selection for clustering.

3.2. FSNF method

In this paper, we present a new unsupervised feature selection method that is based on a basic characteristic of clustering: an instance usually belongs to the same cluster as its nearest neighbors and belongs to a different cluster than its farthest neighbors. We construct three major components for feature assessment. The first is neighbor definition. Once the number of nearest and farthest neighbors is empirically determined, the second step is to generate a new data representation for these neighbors. Finally, we develop a method for evaluating feature salience using the mutual information criterion.

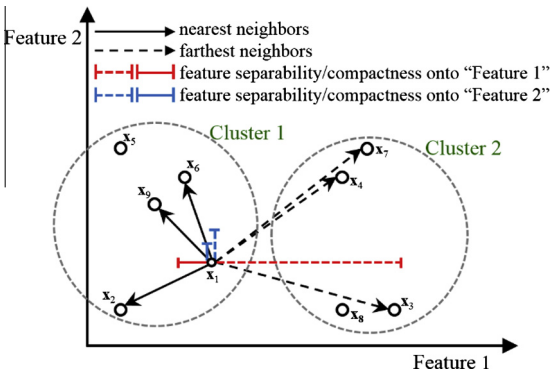

Fig. 1. A schematic example: an instance "$\mathbf{x}_1$" has its nearest neighbors {"$\mathbf{x}_9$", "$\mathbf{x}_6$", "$\mathbf{x}_2$"} and its farthest neighbors {"$\mathbf{x}_3$", "$\mathbf{x}_7$", "$\mathbf{x}_4$"}. For instance "$\mathbf{x}_1$," "Feature 1" should be more salient than "Feature 2."

3.2.1. Neighbor definition

Assume that we have a dataset $\boldsymbol{X} \in \mathbb{R}^d$ consisting of $\mathbf{x}_1, \ldots, \mathbf{x}_i, \ldots, \mathbf{x}_n$ data instances, where $\mathbf{x}_i = [x_{1,i}, \ldots, x_{j,i}, \ldots, x_{d,i}]^T$ represents the ith instance in $\boldsymbol{X}$. In addition, assume a non-zero feature vector $\mathbf{w}(t) = [w_1(t), \ldots, w_j(t), \ldots, w_d(t)]^T$, where element $w_j(t)$ is a real-valued quantity at the tth iteration. We first consider $\mathbf{w}(t)$ to obtain the nearest and farthest neighbors for a given instance. Searching for these neighbors is quite simple and consists of two major steps [3]: calculating the distance between instances and identifying the shortest and longest distances from a set of distances [3]. The kth nearest neighbor $\mathbf{x}^{\Theta}_{i\to k,\mathbf{w}(t)}$ and the lth farthest neighbor $\mathbf{x}^{\phi}_{i\to l,\mathbf{w}(t)}$ for an instance $\mathbf{x}_i$ are defined subject to

$$\pi(k) = \sum_{r=1, r\neq i}^{n} I\Big(dist(\mathbf{x}_i, \mathbf{x}_r|\mathbf{w}(t)) \leqslant dist\Big(\mathbf{x}_i, \mathbf{x}^{\Theta}_{i\to k;\mathbf{w}(t)}|\mathbf{w}(t)\Big)\Big) \tag{1}$$

$$\psi(l) = \sum_{r=1, r\neq i}^{n} I\Big(dist(\mathbf{x}_i, \mathbf{x}_r|\mathbf{w}(t)) \geqslant dist\Big(\mathbf{x}_i, \mathbf{x}^{\phi}_{i\to l;\mathbf{w}(t)}|\mathbf{w}(t)\Big)\Big) \tag{2}$$

where $\pi(k)$ and $\psi(l)$ are applied to define the number of nearest neighbors k and the number of farthest neighbors l, respectively. The indicator Θ represents a nearest neighbor, and ϕ represents a farthest neighbor. The function $I()$ outputs 1 when the condition is satisfied and is zero otherwise. We use the weighted Euclidean metric to compute and obtain the neighbors; the Euclidean metric is another option. Here, $dist(\mathbf{x}_i, \mathbf{x}_r|\mathbf{w}(t))$ is the distance function applied to measure the distance between $\mathbf{x}_i$ and $\mathbf{x}_r$ using $\mathbf{w}(t)$. The distance function is expressed as

$$dist(\mathbf{x}_i, \mathbf{x}_r|\mathbf{w}(t)) = \left(\sum_{j=1}^{d} w_j(t) \times \left(x_{j,i} - x_{j,r}\right)^2\right)^{0.5} \tag{3}$$

3.2.2. Neighbor encoding

We generate a new data representation for the nearest and farthest neighbors. When the number of neighbors (e.g., K and L) is assigned, we have K nearest neighbors and L farthest neighbors. We then define two sets of distances, $\mathbf{NS}^{\mathbf{w}(t)}_{i,K}$ and $\mathbf{FS}^{\mathbf{w}(t)}_{i,L}$, to record the distances from $\mathbf{x}_i$ to $\mathbf{x}^{\Theta}_{i\to k,\mathbf{w}(t)}$ and $\mathbf{x}^{\phi}_{i\to l,\mathbf{w}(t)}$, respectively. These two sets are represented as

$$\mathbf{NS}^{\mathbf{w}(t)}_{i,K} = \left[d\left(\mathbf{x}_i, \mathbf{x}^{\Theta}_{i\to 1;\mathbf{w}(t)}\right), \ldots, d\left(\mathbf{x}_i, \mathbf{x}^{\Theta}_{i\to k;\mathbf{w}(t)}\right), \ldots, d\left(\mathbf{x}_i, \mathbf{x}^{\Theta}_{i\to K;\mathbf{w}(t)}\right)\right]^{\mathrm{T}} \tag{4}$$

$$\mathbf{FS}^{\mathbf{w}(t)}_{i,L} = \left[d\left(\mathbf{x}_i, \mathbf{x}^{\phi}_{i\to 1;\mathbf{w}(t)}\right), \ldots, d\left(\mathbf{x}_i, \mathbf{x}^{\phi}_{i\to l;\mathbf{w}(t)}\right), \ldots, d\left(\mathbf{x}_i, \mathbf{x}^{\phi}_{i\to L;\mathbf{w}(t)}\right)\right]^{\mathrm{T}} \tag{5}$$

where $d(.,.)$ is an element-wise absolute operator, yielding a d-dimensional data instance (e.g., $d\left([0.3, 0.5]^{\mathrm{T}}, [0.1, 0.9]^{\mathrm{T}}\right) = [0.2, 0.4]^{\mathrm{T}}$). Both $\mathbf{NS}^{\mathbf{w}(t)}_{i,K}$ and $\mathbf{FS}^{\mathbf{w}(t)}_{i,L}$ are vectors, and their sizes are $K \times d$ and $L \times d$, respectively.

To measure the salience of each feature, FSNF aims to observe which feature primarily results in these two sets of distances. Let us assume that the two sets of distances are generalized by salient features. To implement this generalization, we examine the two sets as a new data fraction $\mathbf{S}_i(\mathbf{w}(t))$ expressed as

$$\mathbf{S}_i(\mathbf{w}(t)) = \left[\mathbf{NS}^{\mathbf{w}(t)}_{i,K}, \mathbf{FS}^{\mathbf{w}(t)}_{i,L}\right]^{\mathrm{T}} \tag{6}$$

This fraction consists of $K + L$ data instances with d dimensions (i.e., $\mathbf{S}_i(\mathbf{w}(t)) \in \mathrm{R}^d$). We then assign a categorical variable c to represent labels for these instances: the label -1 is assigned for the nearest neighbors, and the label +1 is assigned for the farthest neighbors. The size of $\mathbf{S}_i(\mathbf{w}(t))$ thus becomes $(K + L)\times (d + 1)$. The label c_r for an instance $\mathbf{x}_r \in \mathbf{S}_i(\mathbf{w}(t))$ is assigned as follows:

$$c_r = \begin{cases} -1, & \text{if } \mathbf{x}_r \in \mathbf{NS}^{\mathbf{w}(t)}_{i,K} \\ +1, & \text{if } \mathbf{x}_r \in \mathbf{FS}^{\mathbf{w}(t)}_{i,L} \end{cases} \tag{7}$$

FSNF is an unsupervised feature selection method; therefore, the class labels are not considered during the feature selection process. Fortunately, while new labels are assigned using Eq. (7), the filter- and wrapper-based feature selection methods in supervised learning can be used to help FSNF to identify salient features. For example, we can use a filter-based feature selection method (e.g., mutual information) to evaluate how an individual feature informs the target variable [6,20,37]. Alternatively, we can apply a wrapper-based feature selection method (e.g., using SVM to construct a classifier) to observe how a feature better distinguishes between the nearest and farthest neighbors' instances using the generalized classifier [20,35,36,51].

3.2.3. Feature assessment

In this paper, we use the filter-based feature selection method to evaluate features because of its efficiency. The basis of the method for evaluating feature salience is to determine whether a feature is able to distinguish the nearest and farthest neighbors of a data point. In particular, a feature that is more dependent on the target variable and is less redundant with other features is more salient [37]. Consider $\mathbf{S}_i(\mathbf{w}(t))$. We define $\mathbf{S}_i(\mathbf{w}(t))$ as consisting of $s_1,\ldots,s_j,\ldots,s_d$ variables, where s_j is the jth variable. We then denote a salience vector $\mathbf{u}_i(t)=[u_{1,i}(t),\ldots,u_{j,i}(t),\ldots,u_{d,i}(t)]^T$ for an instance $\mathbf{x}_i$, where $u_{j,i}(t)$ represents feature salience for the jth feature. With this data fraction, $\mathbf{S}_i(\mathbf{w}(t))$, we measure the salience of the variable s_j, which represented as Φ_j:

$$\Phi_j = D(s_j,c) - R(s_j), \tag{8}$$

in which we use the $D()$ and $R()$ criteria to evaluate dependency and redundancy, respectively. The functions $D()$ and $R()$ are expressed as

$$D(s_j,c) = MI(s_j;c) \tag{9}$$

$$R(s_j) = \frac{1}{d-1}\sum_{h=1,h\neq j}^{d} MI(s_h;s_{h\neq j}) \tag{10}$$

where $MI()$ is a mutual information criterion. The criterion for measuring $MI()$ is quite simple. Assume that we want to measure the mutual information of two discrete random variables X and Y. $MI(X,Y)$ is thus defined as

$$MI(X;Y) = \sum_{y\in Y}\sum_{x\in X} p(x,y)\times\log\left(\frac{p(x,y)}{p(x)\times p(y)}\right) \tag{11}$$

where x and y are the domains in X and Y, respectively. The functions $p(x)$ and $p(y)$ represent the marginal probabilities of X and Y, respectively, and $p(x,y)$ is the joint probability density function of X and Y.

Each data fraction $\mathbf{S}_i(\mathbf{w}(t))$ then produces $\mathbf{u}_i(t)$, in which each element is measured using Eq. (8). The follow-up step is to determine $\mathbf{w}(t)$ and search for the final $\mathbf{w}(T)$, which is the local-optimal solution.

3.2.4. Convergence analysis

We discuss the characteristics of $\mathbf{w}(t)$ and $\mathbf{u}_i(t)$ and further provide a local optimal solution to search for $\mathbf{w}(T)$. For an assigned $\mathbf{w}(t)$, we have $\mathbf{u}_1(t),\ldots,\mathbf{u}_i(t),\ldots,\mathbf{u}_n(t)$, the values of which are all different. This difference indicates that different instances would favor features differently. If the values of $\mathbf{u}_1(t),\ldots,\mathbf{u}_i(t),\ldots,\mathbf{u}_n(t)$ are the same, the goal of selecting a subset of features is achieved (i.e., the salient features are selected according to the use of $\mathbf{u}_i(t)$ or any $\mathbf{u}_{r\neq i}(t)$ because these selected features are preferred for all instances). However, assigning a $\mathbf{w}(t)$ that produces equal values for $\mathbf{u}_1(t),\ldots,\mathbf{u}_i(t),\ldots,\mathbf{u}_n(t)$ is very difficult.

To overcome the above difficulty, we use an instance-based learning to produce $\mathbf{w}(T)$, which is a local-optimal solution, and the product $\mathbf{u}(T)$ is further applied for the feature selection. In particular, each iteration t accepts only one data instance $\mathbf{x}_t$ to participate in the resulting $\mathbf{u}_t(t)=[u_{1,t}(t),\ldots,u_{j,t}(t),\ldots,u_{d,t}(t)]^T$. $\mathbf{u}_t(t)$ is simply rewritten as $\mathbf{u}(t)$, and $\mathbf{x}_t$ is randomly sampled from $\boldsymbol{X}$. The given $\mathbf{w}(t)$ thus produces $\mathbf{u}(t)$. We then expect to search for $\mathbf{w}(T)$, producing $\mathbf{u}(T)$ such that the loss value, $||\mathbf{u}(T)-\mathbf{w}(T)||^2$, is minimized. We follow an update function [4,3,15,28] to measure this loss and search the $\mathbf{w}(T)$. The update function is written as

$$\mathbf{w}(t+1) = \mathbf{w}(t) + \alpha(t)\times[\mathbf{u}(t)-\mathbf{w}(t)] \tag{12}$$

where $\alpha(t)$ is a monotonically decreasing learning coefficient. We empirically set the initial learning rate $\alpha(1)$ to 0.8 and decrease the learning rate $\alpha(t)$ at the tth iteration. This equation to measure the loss is commonly applied to achieve the local-optimal solution and is useful in many real-world applications, such as neural networks and self-organizing map algorithms. This equation is stopped when T is satisfied. If a value $w_j(t)$ is larger than $w_{h\neq j\in\{1,\ldots,d\}}(t)$ and $u_{j,t}(t)$ is larger than $u_{h\neq j\in\{1,\ldots,d\},t}(t)$, we say that $w_j(t)$ is positively used to obtain $u_{j,t}(t)$. The jth feature should then increase the influence at the $t+1$th iteration (i.e., the value of $w_j(t+1)$ should be larger). Conversely, when the obtained $u_{j,t}(t)$ is smaller than $u_{h\neq j\in\{1,\ldots,d\},t}(t)$, the jth feature is not salient (the value of $w_j(t+1)$ will be discounted). Of course, if $w_j(t)$ is smaller than $w_{h\neq j\in\{1,\ldots,d\}}(t)$ but $u_{j,t}(t)$ is larger than $u_{h\neq j\in\{1,\ldots,d\},t}(t)$, the influence of $w_j(t+1)$ must increase.

We briefly illustrate the example mentioned in Fig. 1 to show the balance of a mutually strengthening $\mathbf{u}(t)$ and $\mathbf{w}(t)$ pair. Assume that we have an instance $\mathbf{x}_t=[0.4,0.4]^T$. We wish to find a data fraction $\mathbf{S}_t(\mathbf{w}(t))$ from eight instances, as shown in Table 1. The parameters K and L are each set to 3. In this example, we observe which feature will be more appropriate for $\mathbf{x}_t$ to contribute force at $\mathbf{w}(t+1)$. $\mathbf{w}(t)$ is set to $[0.5,0.5]^T$ and $[0.1,0.9]^T$ and is used to compare the resulting $\mathbf{w}(t+1)$.

First, we use $\mathbf{w}(t)=[0.5,0.5]^T$ to find a data fraction $\mathbf{S}_t(\mathbf{w}(t))$ consisting of $\mathbf{NS}_{t,3}^{\mathbf{w}(t)}$ and $\mathbf{FS}_{t,3}^{\mathbf{w}(t)}$, as shown in Table 2. Intuitively, we would believe "Feature 1" to be more salient because it is more dependent on the label variable. We can use Eq. (9) to validate this dependence; therefore, $u_{1,t}(t)$ must be larger than $u_{2,t}(t)$. After performing the update function (i.e., Eq. (12)), $w_1(t+1)$ should be larger than $w_2(t+1)$.

Table 1
Six instances with two features.

Instance	Feature 1	Feature 2
$\mathbf{x}_2$	0.1	0.1
$\mathbf{x}_3$	0.9	0.1
$\mathbf{x}_4$	0.7	0.7
$\mathbf{x}_5$	0.1	0.8
$\mathbf{x}_6$	0.3	0.7
$\mathbf{x}_7$	0.8	0.8
$\mathbf{x}_8$	0.7	0.1
$\mathbf{x}_9$	0.2	0.6

Second, we set $\mathbf{w}(t) = [0.1, 0.9]^T$ while leaving the other parameters unchanged. A new data fraction $\mathbf{S}_i(\mathbf{w}(t))$ is shown in Table 3. We can again use Eq. (9) to measure the dependence. Although we assign "Feature 2" a larger weight to find $\mathbf{S}_i(\mathbf{w}(t))$, "Feature 1" remains more salient than "Feature 2" (i.e., $u_{1,t}(t)$ is larger than $u_{2,t}(t)$). At iteration $t+1$, the influence of "Feature 1" should increase, $w_1(t+1)$ must be larger than $w_1(t)$, and $w_2(t+1)$ must be smaller than $w_2(t)$.

4. Experiment

This section presents evaluations of FSNF for feature selection in clustering problems. We apply FSNF and other existing filter- and wrapper-based feature selection methods to select features for several real-world datasets. To demonstrate the effectiveness of the selected feature subsets, the subsets are applied to discover clusters whose qualities are evaluated using relative cluster validations. Because different feature subsets have different underlying numbers of natural clusters [26], we measure the experimental results by determining which method obtains clusters that best maximize the between-cluster scatter and minimize the within-cluster scatter.

4.1. Parameter and dataset description

We now set the parameters for FSNF. Assume that we have a dataset consisting of a total of N instances. We set the learning rate $\alpha(1)$ to 0.8, and $\alpha(t) = \alpha(1) \times ((T-t)/T)$ at the tth iteration. The initial feature salience vector $\mathbf{w}(1) = [0.5, 0.5, \ldots, 0.5]^T$, where we set the weight of each element to the same value. The parameters K and L are empirically set to 30 based on the findings of the study presented in [4]. The number of iterations T should be large, such that most instances can be randomly selected for training on the algorithm. We thus set T to $10N$, where N is the size of the training dataset.

We used six real-world datasets for the experiments. These datasets have been widely used to investigate the feature selection problem. The arrhythmia dataset was tested in a previous study [37], and we followed that study's procedures to ignore noisy instances. We tested several benchmark datasets found in the literature, including bench, muskv1, wdbc and wine [21,47]. The OT dataset [2,34] has eight categories: coasts, forests, mountains, open country, highways, inside cities, tall buildings and streets. The dataset consists of 2688 RGB images. The size of each image is 256 $\subseteq$ 256 pixels. Following the procedures of study [2], we randomly selected 100 images for each category and captured features for the used images. The datasets are summarized in Table 4.

4.2. Preparation of other feature selection methods and clustering algorithms

We implemented other filter- and wrapper-based methods and compared them with FSNF for the selection of salient features. Often, the purpose of the filter- and wrapper-based methods for selecting a feature subset is to maximize the objective criterion. However, the selected feature subset, especially for clustering, suffers from two biases: (1) a given feature subset has its own clusters, which may be formed from a different feature subset, and (2) the optimal clusters evaluated by an objective function depend strongly on the dimensionality of the selected feature subset. Therefore, the search for the best

Table 2
A data fraction $\mathbf{S}_t(\mathbf{w}(t))$ with labels is obtained under the settings $K = 3$, $L = 3$ and $\mathbf{w}(t) = [0.5, 0.5]^T$.

$\mathbf{S}_t(\mathbf{w}(t))$	Instance	Feature 1	Feature 2	Label
$\mathbf{NS}_{t,3}^{\mathbf{w}(t)}$	$\mathbf{x}_9$	0.2	0.2	−1
	$\mathbf{x}_6$	0.1	0.3	−1
	$\mathbf{x}_2$	0.3	0.3	−1
$\mathbf{FS}_{t,3}^{\mathbf{w}(t)}$	$\mathbf{x}_3$	0.5	0.3	1
	$\mathbf{x}_7$	0.4	0.4	1
	$\mathbf{x}_4$	0.3	0.3	1

Table 3
A data fraction $\mathbf{S}_t(\mathbf{w}(t))$ with labels is obtained under the settings $K = 3$, $L = 3$ and $\mathbf{w}(t) = [0.1, 0.9]^T$.

$\mathbf{S}_t(\mathbf{w}(t))$	Instance	Feature 1	Feature 2	Label
$\mathbf{NS}_{t,3}^{\mathbf{w}(t)}$	$\mathbf{x}_9$	0.2	0.2	−1
	$\mathbf{x}_6$	0.1	0.3	−1
	$\mathbf{x}_4$	0.3	0.3	−1
$\mathbf{FS}_{t,3}^{\mathbf{w}(t)}$	$\mathbf{x}_7$	0.4	0.4	1
	$\mathbf{x}_3$	0.5	0.3	1
	$\mathbf{x}_2$	0.3	0.3	1

Table 4
Six real-world datasets.

Dataset	Full name	#instances	#attributes	#classes
arrhythmia	Arrhythmia	420	278	16
bench	Connectionist Bench (Sonar, Mines vs. Rocks)	208	60	2
muskv1	Musk (Version 1)	476	168	2
OT	Oliva and Torralba	800	512	8
wdbc	Wisconsin diagnostic breast cancer	569	30	2
wine	Wine recognition	178	13	3

feature subset for clustering has expensive computational costs. Although some studies aiming to improve computational efficiency have been proposed, the computational cost is great when the number of dimensions is large.

To address this computational issue, we used a ranking strategy [27] for every feature, without searching the space of all possible feature subsets. As such, each feature was ordered by its measured score using an evaluation criterion. The number of features was then chosen by manually setting a threshold for selecting a feature subset, using the filter- and wrapper-based methods.

After the selected feature subset was chosen, the features were used to perform clustering using existing clustering learning algorithms, including *K*-means and SOM, which are very popular and have inspired many applications in the field of cluster analysis. We then tested whether FSNF can perform better than others when selecting a subset of features for the cluster analysis.

4.3. Comparison to filter-based methods

We compared FSNF to three filter-based methods. The Max-Relevance (Max-Rel) method was implemented in study [37] and is also based on mutual information. The method selects features according to the maximal statistical dependency criterion. The variance metric (Var.) is commonly used to evaluate the variance for each individual feature; therefore, a variable that has a variance that is large would be selected. IBNF [3] uses instance-based learning to quantify how a feature contributes to compactness and separability. These three methods and FSNF were applied to select salient features for discovering clusters.

4.3.1. Number of features

We first observed the sensitivity of the number of selected features to the clusters on the six real-world datasets. After selecting a subset of features, we used a *K*-means algorithm to partition each dataset into clusters. The sensitivity of *K* is discussed below. To evaluate the quality of the clusters, we used the R-squared validity [13], which has been used to measure the interpretation capability of statistical models. Three symbols are associated with the R-squared validity [13], SS_w, SS_b and SS_t, where SS_w represents the within-group sum of squares, SS_b represents the between-group sum of squares, and SS_t measures the total sum of squares in the entire dataset. The R-squared value is the ratio of SS_b to SS_t; thus, a high R-squared value implies that SS_b is large and that SS_w is small, indicating that the clusters maximize between-cluster scatter and minimize within-cluster scatter, thereby producing high-quality clusters.

The core of this experiment was the number of selected features. As we observed, R_{tr} represents the rate at which the top-ranked features are selected. For example, $R_{tr} = 1$ indicates that all of the features are selected. Fig. 2 shows the R-squared values for various R_{tr} in FSNF compared to those calculated for the three filter-based methods.

In Fig. 2, the experimental results indicate that the selected features are informative to the clusters. The squared value usually increases as R_{tr} decreases. As R_{tr} increases, the squared value decreases. Therefore, we conclude that a few features positively contribute to achieving both maximal between-cluster scatter and minimal within-cluster scatter. Based on the overall performance, we see observed FSNF performs better than the others do when selecting a subset of features. This subset is thus applied to perform clustering using the *K*-means algorithm.

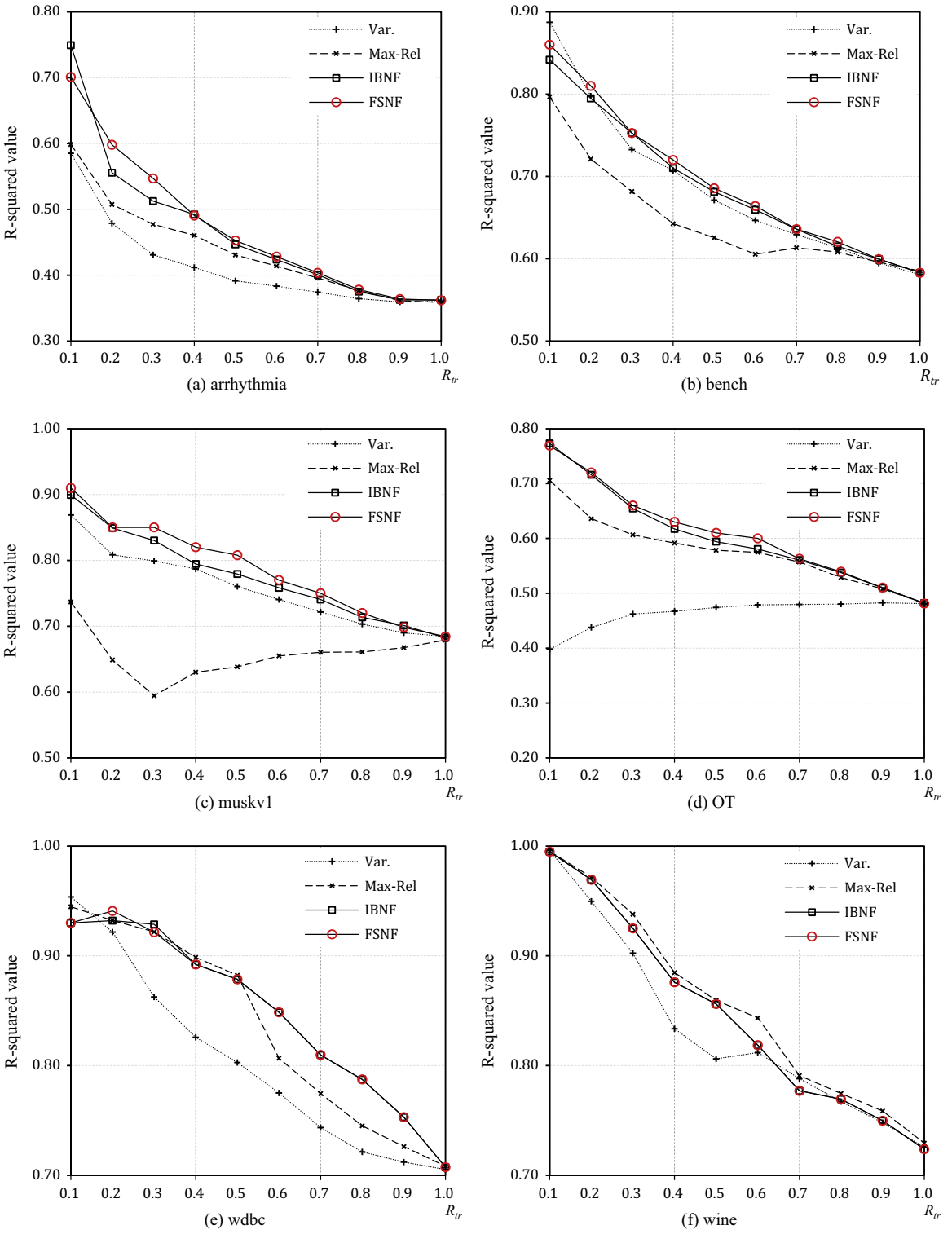

Fig. 2. Comparison of R-squared values for various R_{tr} in the datasets: (a) arrhythmia, (b) bench, (c) muskv1, (d) OT, (e) wdbc and (f) wine. R_{tr} was set to select a feature subset for clustering using the *K*-means algorithm.

4.3.2. Cluster quantity

Previously, we provided an analysis explaining the selection of a salient subset for discovering various numbers of clusters. The selected features were applied to partition each dataset into clusters; then, the quality of those clusters was evaluated using relative cluster validity. Two clustering algorithms, *K*-means and SOM, were utilized to perform clustering. For each algorithm, the number of clusters was set to $M \in \{2, 3, \ldots, 30\}$ to partition a dataset into M clusters. We used the Davies–Bouldin index (DBI) [9] to measure the performance because of its popularity for cluster evaluation. Low DBI values indicate high clustering performance.

Fig. 3 shows the performance obtained when using the selected features to partition each dataset into M clusters using the *K*-means algorithm. Fig. 3(a) shows that the features selected by FSNF produces a better cluster quality than do features selected by other methods. In Fig. 3(b), for some cases, such as $M = \{4, 8, 9, 10, 17 \text{ and } 21\}$, FSNF does not select the feature subset with the best performance. Fig. 3(c) and (e) show that FSNF does not provide the best performance in some cases. Fig. 3(f) shows that FSNF does not perform better than others when setting M to 5. In general, however, the overall performance indicates that FSNF selects a better subset of features for performing clustering, leading to a lower DBI value. For these four methods, the features selected using Var. usually resulted in the lowest cluster quality.

We then used SOM to partition the datasets with the selected features. With SOM, similar data points might be projected to a neuron or a neighboring neuron on the map. However, the clusters and the number of clusters are not available in the SOM map. To obtain those clusters, we followed the methods of the study described in [46] to obtain a user-defined number of clusters for the SOM neurons. Specifically, the major steps for obtaining the clusters included the following:

- Data instances were projected to the map neurons using SOM.
- Map neurons were clustered into M clusters using the *K*-means algorithm, using the weights of the neurons as inputs.
- Data instances were partitioned into $M \in \{2, 3, \ldots, 30\}$ clusters. If two neurons were clustered together, the instances projected to these two neurons were referred to as a cluster.

Fig. 4 reports the performance of partitioning each dataset into M clusters, where the DBI metric was applied a second time to measure the quality of the clusters. Fig. 4 shows that the performance supports the claim that FSNF is better than other methods in selecting features that are effective for performing clustering. We also observe that the other methods have the potential to perform better. For example, IBNF performs better with the setting $M = \{6, 9, 11, 15, 29\}$, as shown in Fig. 4(a), or the setting $M = \{4, 6, 13, 24\}$, as shown in Fig. 4(d). In Fig. 4(e), it is interesting to observe that Max-Rel and FSNF often produce the same DBI values because FSNF and Max-Rel select very similar features.

4.4. Comparison to wrapper-based methods

We implemented a wrapper-based method (abbreviated hereafter as "wrapper method") to select features and compared the selected features to those selected by FSNF when performing clustering. The wrapper method must consist of a learning algorithm trained for feature assessment. We used the *K*-means algorithm to train the wrapper method due to the algorithm's simplicity, and K was set to be equivalent to the number of class labels because the optimal number of clusters in a real-world dataset is usually unavailable and would be difficult to obtain. The steps of the implemented wrapper method for selecting features were as follows:

Step 1: Access a dataset for the selection of features.
Step 2: Partition the dataset into clusters using the retained features.
Step 3: Measure the dependencies between the clusters and the retained features.
Step 4: Ignore a feature (with a minimal dependency score) from the set of retained features.
Step 5: Go to Step 2 until the number of retained features is satisfied.

Step 1 involved accessing a dataset to train the wrapper method, and Step 2 involved obtaining clusters for the dataset. Once the clusters were obtained, the feature assessment involved measuring a feature's dependency when obtaining these clusters. The follow-up step (Step 3) involved measuring the dependence with the option of using the mutual information criterion or statistical regression analysis. Specifically, we used the proposed criterion [37] to evaluate how informative a feature was to the labeled clusters. We then studied the greedy backward selection algorithm [16] while considering clusters to select a subset of features; only one feature at a time was ignored according to its minimal dependence (Step 4). Step 5 involved an iterative mechanism to identify a user-defined number of features.

To validate the performance of the selected features for clustering, we applied a 10-fold cross-validation to separate each prepared dataset (shown in Table 4) into a training dataset and a testing dataset. In this particular case, the training dataset was applied to the selected half of the salient features using the above-mentioned steps. The testing dataset, which retained the selected features, was partitioned into $M \in \{10, 20, 30\}$ clusters. The cluster quality was measured using DBI validity; there were 10 DBI values over the 10-fold cross-validation procedure. We used the box-plot to graphically display the variation range of the DBI values. Fig. 5 shows the DBI values for various M values in FSNF and compared them to those from the wrapper method.

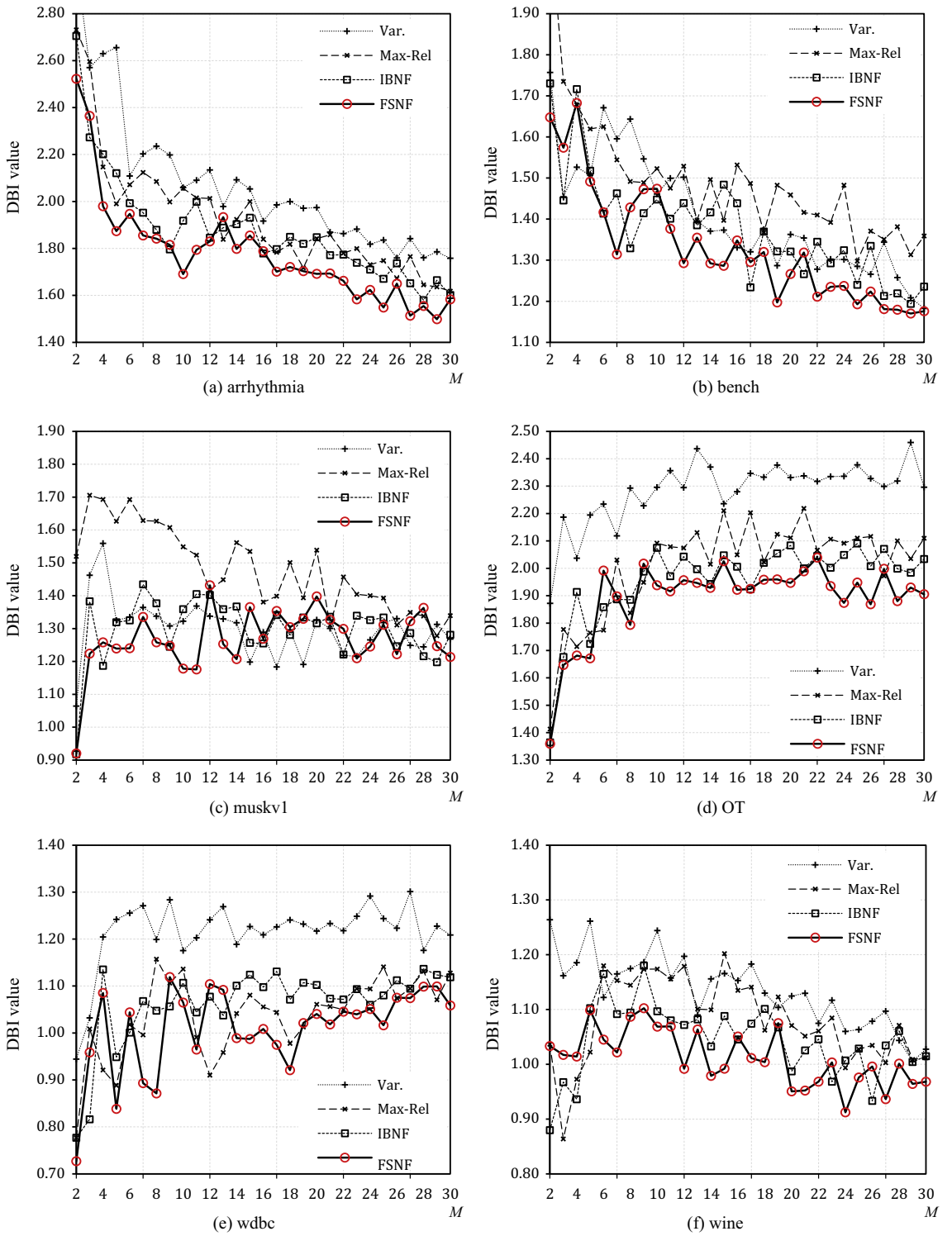

Fig. 3. The DBI value with M clusters using the K-means method to perform clustering on the following datasets: (a) arrhythmia, (b) bench, (c) muskv1, (d) OT, (e) wdbc and (f) wine.

Fig. 5 shows that the overall performance of FSNF is better than that of the wrapper method based on the observed DBI values. Fig. 5(a), (d), (e) and (f) show that FSNF results in lower DBI values, aiming to maximize the between-cluster scatter and minimize the within-cluster scatter. Fig. 5(b) and (c) show that the wrapper might be comparable to FSNF. Moreover, it

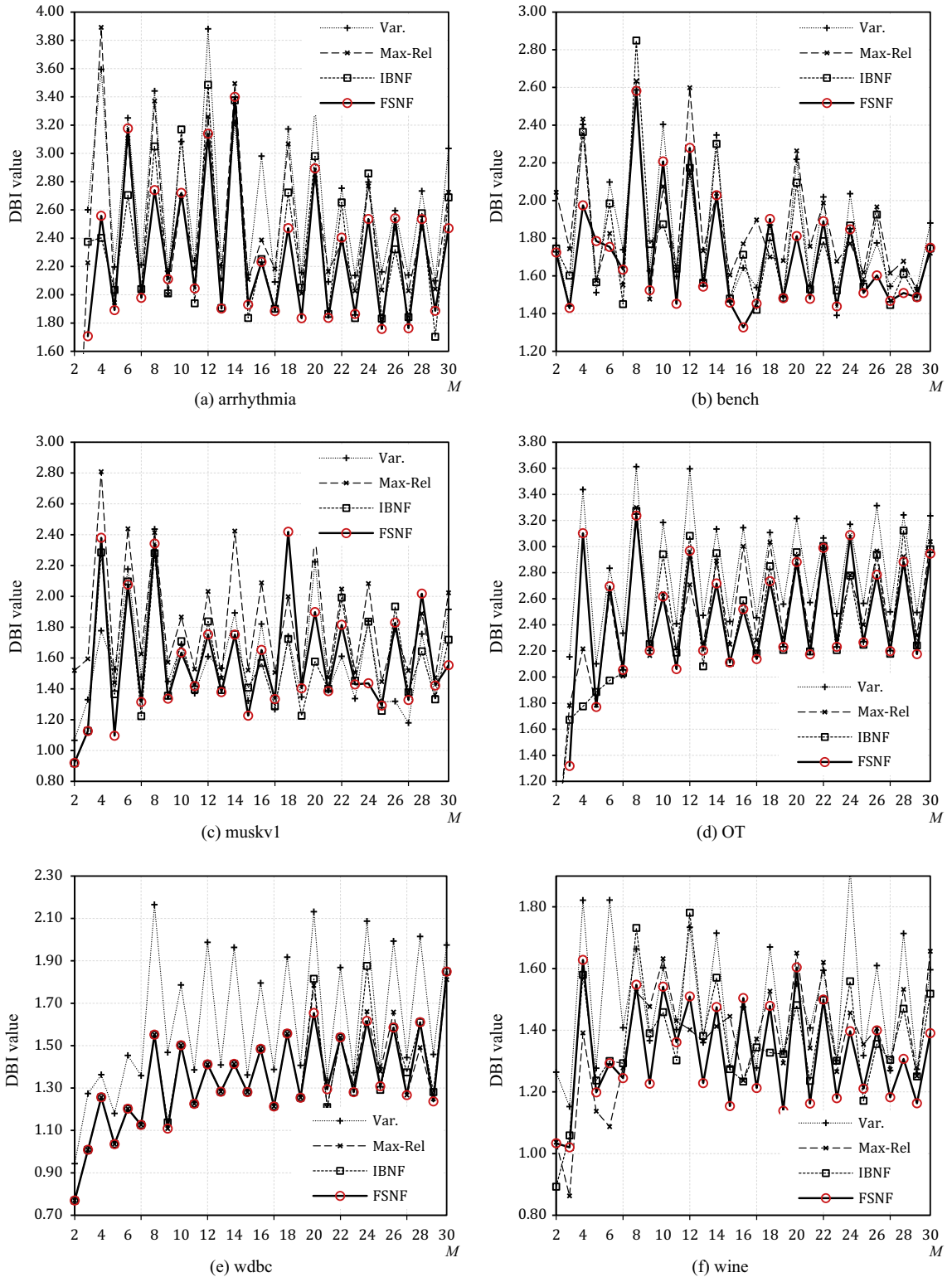

Fig. 4. The DBI value according to the number of clusters M using the SOM algorithm to perform clustering in the following datasets: (a) arrhythmia, (b) bench, (c) muskv1, (d) OT, (e) wdbc and (f) wine.

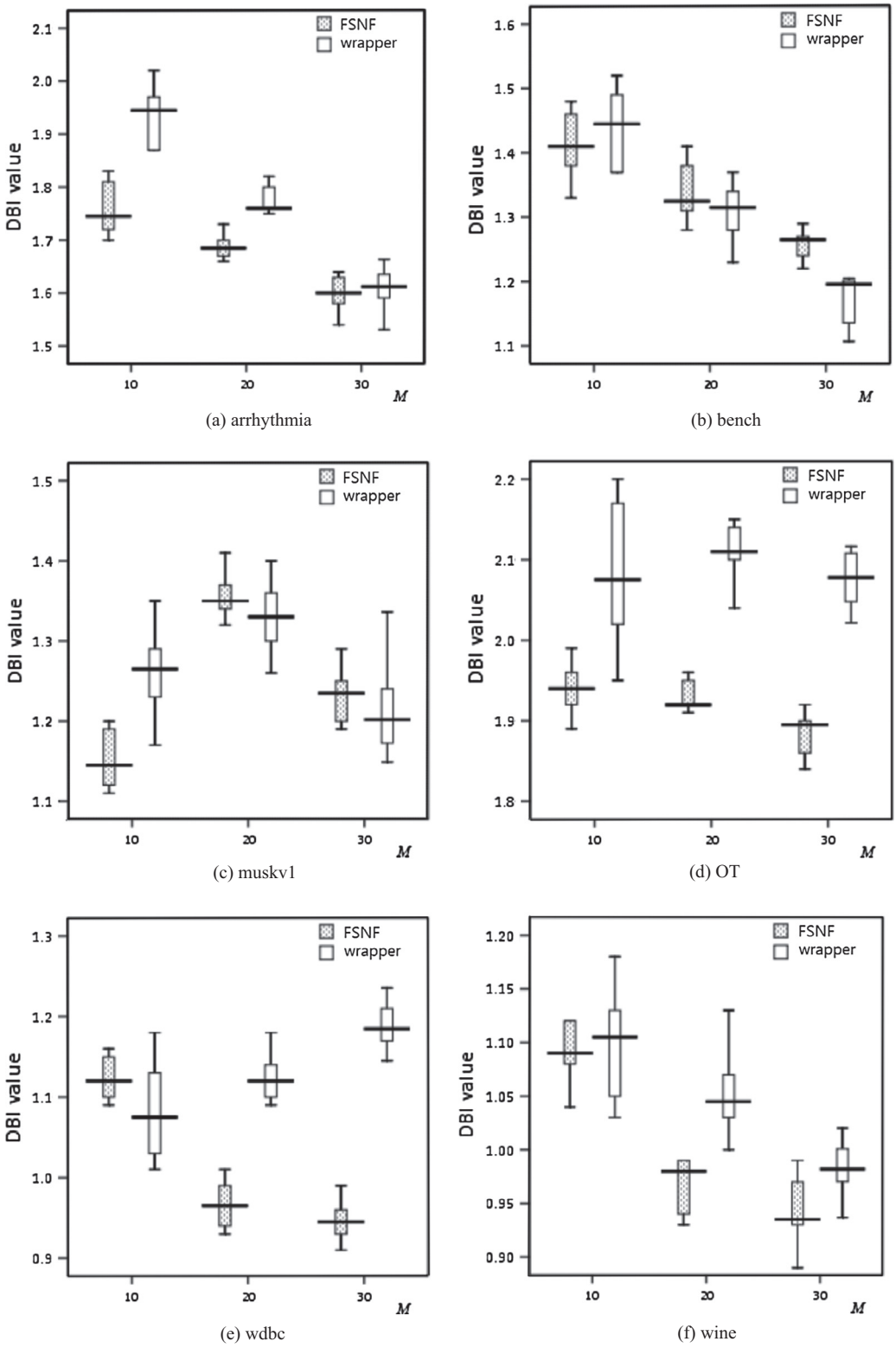

Fig. 5. The box-plot result for graphically displaying DBI values according to the number of clusters *M* on the following datasets: (a) arrhythmia, (b) bench, (c) muskv1, (d) OT, (e) wdbc and (f) wine.

is interesting to see that FSNF is more stable than the wrapper method; the variation of the DBI values from FSNF is often less than that from the wrapper method. Recall that the wrapper method measures features based on how informative they are to those clusters, i.e., the selected features are further applied to perform clustering. Thus, we see that the wrapper method is not specifically designed to maximize the between-cluster scatter criterion and minimize the within-cluster scatter criterion leading to a satisfactory cluster quality.

5. Discussion and conclusions

In this paper, we present a new feature selection method for selecting a subset of salient features for clustering. We note that the selected features are effectively used for cluster analysis because FSNF is motivated by a basic characteristic of clustering: an instance usually belongs to the same cluster as its geometrically closest neighbors and to a different cluster than its geometrically farthest clusters. To demonstrate the effectiveness of FSNF, we carefully implement several existing filter- and wrapper-based feature selection methods for comparison with FSNF. We use the methods to select a subset of features, which are then applied to perform clustering using the *K*-means and SOM algorithms.

The experiments show that FSNF usually outperforms the filter-based methods for selecting features to perform clustering with. The filter-based methods do not involve a clustering algorithm; therefore, their favored features are usually not effective for cluster discovery because the number of clusters or the clustered structure cannot be effectively predicted with or without the help of a clustering algorithm.

It is also interesting to observe that FSNF, which does not require a clustering algorithm to learn the salient features, exhibits better performance than the wrapper method, which captures salient features learned from a predetermined clustering algorithm. Although the wrapper method is more intuitive for the use of selecting feature subsets for clustering, it might suffer from two biases: (1) the selection of the clustering algorithms and (2) the participation of noisy features. The first issue implies that different clustering algorithms select different clusters. The favored features might thus be different. The second issue implies that noisy features are not excluded; the obtained clusters are based on those noisy features.

FSNF attempts to identify the best weight vector by considering the nearest and farthest neighbors of every instance. In the literature, some supervised local-learning-based feature selection methods have considered the characteristics of the nearest neighbors for every instance. For example, the RFE method [12] and RELIEF [18] identified relevant features that were effective for distinguishing neighbors with different class labels. Sun et al. [43] used the support vector machine method to handle this distinguishability. In contrast, FSNF focuses on unsupervised feature selection for clustering, as their methods are limited when the data does not provide class labels.

FSNF may provide a general framework for selecting salient features for clustering and supporting comprehensive filter-based and wrapper-based methods for implementation schemes. Recall that we produce a data fraction with assigned labels; unsupervised learning is thus reduced to supervised learning for this fraction (mentioned in Section 3.2.2). Therefore, many filter- and wrapper-based feature selection methods in supervised learning can be used to help FSNF identify salient features. Because cluster analysis for very high data dimensionality has been in high demand, we expect that our work will generate broad interest in the data-mining field.

Acknowledgment

This research was supported by the National Science Council, Taiwan, under Grants NSC 102-2410-H- 275-009-MY2.

References

[1] P. Bermejo, L. Ossa, J.A. Gamez, J.M. Puerta, Fast wrapper feature subset selection in high-dimensional datasets by means of filter re-ranking, Knowl.-Based Syst. 25 (2012) 35–44.
[2] S. Boutemedjet, N. Bouguila, D. Ziou, A hybrid feature extraction selection approach for high-dimensional non-Gaussian data clustering, IEEE Trans. Pattern Anal. Machine Intell. 3 (8) (2009) 1429–1443.
[3] C.H. Chen, Feature selection based on compactness and separability: comparison with filter-based methods, Comput. Intell. 30 (3) (2014) 636–656.
[4] C.H. Chen, A semi-supervised feature selection method using a non-parametric technique with side information, J. Inform. Sci. 39 (3) (2013) 359–371.
[5] Y. Chen, H. Sampathkumar, B. Luo, X.-w. Chen, iLike: bridging the semantic gap in vertical image search by integrating text and visual features, IEEE Trans. Knowl. Data Eng. 25 (10) (2013) 2257–2270.
[6] T.W.S. Chow, D. Huang, Estimating optimal feature subsets using efficient estimation of high-dimensional mutual information, IEEE Trans. Neural Netw. 16 (1) (2005) 213–224.
[7] T.W.S. Chow, P. Wang, E.W.M. Ma, A new feature selection scheme using a data distribution factor for unsupervised nominal data, IEEE Trans. Syst. Man Cybernet.—PART B: Cybernet. 38 (2) (2008) 499–509.
[8] A. Coletta, A. Nowe, C. Molter, C. Lazar, D. Steenhoff, H. Bersini, J. Taminau, R. Duque, S. Meganck, V.d. Schaetzen, A survey on filter techniques for feature selection in gene expression microarray analysis, IEEE/ACM Trans. Comput. Biol. Bioinform. 9 (4) (2012) 1106–1119.
[9] D.L. Davies, D.W. Bouldin, A cluster separation measure, IEEE Trans. Pattern Anal. Machine Intell. 1 (4) (1979) 224–227.
[10] J. Derrac, C. Cornelis, S. Garcia, F. Herrera, Enhancing evolutionary instance selection algorithms by means of fuzzy rough set based feature selection, Inform. Sci. 186 (2012) 73–92.
[11] N. Garcia-Pedrajas, A. Haro-Garcia, J. Perez-Rodriguez, A scalable approach to simultaneous evolutionary instance and feature selection, Inform. Sci. 228 (2014) 150–174.
[12] I. Guyon, J. Weston, S. Barnhill, V. Vapnik, Gene selection for cancer classification using support vector machines, Machine Learn. 46 (1–3) (2002) 389–422.
[13] M. Halkidi, Y. Batistakis, M. Vazirgiannis, On clustering validation techniques, J. Intell. Inform. Syst. 17 (2001) 107–145.
[14] J.S. Han, S.W. Lee, Z. Bien, Feature subset selection using separability index matrix, Inform. Sci. 223 (2013) 102–118.

[15] S.S. Haykin, B. Widrow, Least-Mean-Square Adaptive Filters, Wiley, 2003.
[16] C.N. Hsu, H.J. Huang, D. Schuschel, The ANNIGMA-wrapper approach to fast feature selection for neural nets, IEEE Trans. Syst. Man Cybernet.—PART B: Cybernet. 32 (2) (2002) 207–212.
[17] S. Huang, Z. Chen, Y. Yu, W.Y. Ma, Multitype features coselection for Web document clustering, IEEE Trans. Knowl. Data Eng. 18 (4) (2006) 448–459.
[18] K. Kira, L.A. Rendell, A practical approach to feature selection, in: Proceedings of the 9th Conference on Machine Learning, 1992, pp. 249–256.
[19] L.I. Kuncheva, D.P. Vetrov, Evaluation of stability of k-means cluster ensembles with respect to random initialization, IEEE Trans. Pattern Anal. Machine Intell. 28 (11) (2006) 1798–1807.
[20] N. Kwak, C.H. Choi, Input feature selection for classification problems, IEEE Trans. Neural Netw. 13 (1) (2002) 143–159.
[21] M.H.C. Law, M.A.T. Figueiredo, A.K. Jain, Simultaneous feature selection and clustering using mixture models, IEEE Trans. Pattern Anal. Machine Intell. 26 (9) (2004) 1154–1166.
[22] Y. Li, C. Luo, S.M. Chung, Text clustering with feature selection by using statistical data, IEEE Trans. Knowl. Data Eng. 20 (5) (2008) 641–652.
[23] Z. Li, J. Liu, Y. Yang, X. Zhou, H. Lu, Clustering-guided sparse structural learning for unsupervised feature selection, IEEE Trans. Knowl. Data Eng. 26 (9) (2014) 2138–2150.
[24] Z. Liao, H. Hoppe, D. Forsyth, Y. Yu, A subdivision-based representation for vector image editing, IEEE Trans. Visual. Comput. Graphics 18 (11) (2012) 1858–1867.
[25] L. Lin, J. Sun, L. Zhu, Nonparametric feature screening, Comput. Stat. Data Anal. 67 (2013) 162–174.
[26] H. Liu, E.R. Dougherty, J.G. Dy, K. Torkkola, E. Tuv, H. Peng, C. Ding, F. Long, M. Berens, L. Parsons, Z. Zhao, L. Yu, G. Forman, Evolving feature selection, IEEE Intell. Syst. 20 (6) (2005) 64–76.
[27] H. Liu, L. Yu, Toward integrating feature selection algorithms for classification and clustering, IEEE Trans. Knowl. Data Eng. 17 (4) (2005) 491–502.
[28] O. Macchi, Adaptive Processing: The LMS Approach with Applications in Transmission, Wiley, New York, 1995.
[29] P. Maji, A rough hypercuboid approach for feature selection in approximation spaces, IEEE Trans. Knowl. Data Eng. 26 (1) (2014) 16–29.
[30] S. Maldonado, J. Perez, R. Weber, M. Labbe, Feature selection for support vector machines via mixed integer linear programming, Inform. Sci. 279 (2014) 163–175.
[31] S. Maldonado, R. Weber, A wrapper method for feature selection using support vector machines, Inform. Sci. 179 (2009) 2208–2217.
[32] S. Mitra, P.P. Kundu, W. Pedrycz, Feature selection using structural similarity, Inform. Sci. 198 (2012) 48–61.
[33] Y. Motai, H. Yoshida, Principal composite kernel feature analysis: data-dependent kernel approach, IEEE Trans. Knowl. Data Eng. 25 (8) (2013) 1863–1875.
[34] A. Oliva, A. Torralba, Modeling the shape of the scene: a holistic representation of the spatial envelope, Int. J. Comput. Vis. 42 (2001) 145–175.
[35] M. Pal, G.M. Foody, Feature selection for classification of hyperspectral data by SVM, IEEE Trans. Geosci. Remote Sensing 48 (5) (2010) 2297–2307.
[36] J.M. Pena, R. Nilsson, On the complexity of discrete feature selection for optimal classification, IEEE Trans. Pattern Anal. Machine Intell. 32 (8) (2010) 1517–1522.
[37] H. Peng, F. Long, C. Ding, Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy, IEEE Trans. Pattern Anal. Machine Intell. 27 (8) (2005) 1226–1238.
[38] B. Peralta, A. Soto, Embedded local feature selection within mixture of experts, Inform. Sci. 269 (2014) 176–187.
[39] G. Sanguinetti, Dimensionality reduction of clustered data sets, IEEE Trans. Pattern Anal. Machine Intell. 30 (3) (2008) 535–540.
[40] C. Shang, M. Li, S. Feng, Q. Jiang, J. Fan, Feature selection via maximizing global information gain for text classification, Knowl.-Based Syst. 54 (2013) 298–309.
[41] Q. Song, J. Ni, G. Wang, A fast clustering-based feature subset selection algorithm for high-dimensional data, IEEE Trans. Knowl. Data Eng. 25 (1) (2013) 1–14.
[42] C.T. Su, Y.H. Hsiao, Multiclass MTS for simultaneous feature selection and classification, IEEE Trans. Knowl. Data Eng. 21 (2) (2009) 192–205.
[43] Y. Sun, S. Todorovic, S. Goodison, Local-learning-based feature selection for high-dimensional data analysis, IEEE Trans. Pattern Anal. Machine Intell. 32 (9) (2010) 1610–1626.
[44] J. Tang, H. Liu, Unsupervised feature selection for linked social media data, in: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, USA, 2012, pp. 904–912.
[45] P.K. Vemulapalli, V. Monga, S.N. Brennan, Robust extrema features for time-series data analysis, IEEE Trans. Pattern Anal. Machine Intell. 35 (6) (2013) 1464–1479.
[46] J. Vesanto, E. Alhoniemi, Clustering of the self-organizing map, IEEE Trans. Neural Netw. 11 (3) (2000) 586–600.
[47] M. Vlachos, C. Domeniconi, D. Gunopulos, G. Kollios, N. Koudas, Non-linear dimensionality reduction techniques for classification and visualization, in: Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Edmonton, Canada, 2002, pp. 645–651.
[48] L. Wang, Feature selection with kernel class separability, IEEE Trans. Pattern Anal. Machine Intell. 30 (9) (2008) 1534–1546.
[49] L. Wang, F. Chu, W. Xie, Accurate cancer classification using expressions of very few genes, IEEE/ACM Trans. Comput. Biol. Bioinform. 4 (1) (2007) 40–53.
[50] Z. Xu, I. King, M.R.T. Lyu, R. Jin, Discriminative semi-supervised feature selection via manifold regularization, IEEE Trans. Neural Netw. 21 (7) (2010) 1033–1047.
[51] J.B. Yang, K.Q. Shen, C.J. Ong, X.P. Li, Feature selection for MLP neural network: the use of random permutation of probabilistic outputs, IEEE Trans. Neural Netw. 20 (12) (2009) 1911–1922.
[52] S.H. Yang, B.-G. Hu, Discriminative feature selection by nonparametric bayes error minimization, IEEE Trans. Knowl. Data Eng. 24 (8) (2012) 1422–1434.
[53] Z. Yu, H. Chen, J. You, H.S. Wong, J. Liu, L. Li, G. Han, Double selection based semi-supervised clustering ensemble for tumor clustering from gene expression profiles, IEEE/ACM Trans. Comput. Biol. Bioinform. 11 (4) (2014) 727–740.
[54] L. Zhou, L. Wang, C. Shen, Feature selection with redundancy-constrained class separability, IEEE Trans. Neural Netw. 21 (5) (2010) 853–858.
[55] Z. Zhu, Y.S. Ong, M. Dash, Wrapper-filter feature selection algorithm using a memetic framework, IEEE Trans. Syst. Man Cybernet.—PART B: Cybernet. 37 (1) (2007) 70–76.