Big Data Mining and Analytics

Volume 2 | Issue 2 Article 1

2019

Feature Selection with Graph Mining Technology

Thosini Bamunu Mudiyanselage
the Department of Computer Science, Georgia State University, Atlanta, GA 30302, USA.

Yanqing Zhang
the Department of Computer Science, Georgia State University, Atlanta, GA 30302, USA.

BIG DATA MINING AND ANALYTICS
ISSN 2096-0654 01/05 pp73–82
Volume 2, Number 2, June 2019
DOI: 10.26599/BDMA.2018.9020032

Feature Selection with Graph Mining Technology

Thosini Bamunu Mudiyanselage* and Yanqing Zhang

Abstract: Many real world applications have problems with high dimensionality, which existing algorithms cannot overcome. A critical data preprocessing problem is feature selection, whereby its non-scalability negatively influences both the efficiency and performance of big data applications. In this research, we developed a new algorithm to reduce the dimensionality of a problem using graph-based analysis, which retains the physical meaning of the original high-dimensional feature space. Most existing feature-selection methods are based on a strong assumption that features are independent of each other. However, if the feature-selection algorithm does not take into consideration the interdependencies of the feature space, the selected data fail to correctly represent the original data. We developed a new feature-selection method to address this challenge. Our aim in this research was to examine the dependencies between features and select the optimal feature set with respect to the original data structure. Another important factor in our proposed method is that it can perform even in the absence of class labels. This is a more difficult problem that many feature-selection algorithms fail to address. In this case, they only use wrapper techniques that require a learning algorithm to select features. It is important to note that our experimental results indicates, this proposed simple ranking method performs better than other methods, independent of any particular learning algorithm used.

Key words: graph mining; network embedding; big data analysis; feature selection; high-dimensional data

1 Introduction

Data processing and decision-making in today's world have become more complex with the continuously expanding volume of data. As a result, big data applications are much bigger and more complex than traditional data processing applications can handle. Due to the presence of such large-scale data, it has become a challenge to know how to effectively apply existing algorithms that were originally designed for low-dimensional space. As a result, we are forced to revisit a feature-selection method that provides an effective strategy for preparing high-dimensional data for existing algorithms.

Most existing feature-selection algorithms are based on a strong assumption that the features are independent of each other and are identically distributed. As such, since these algorithms neglect the structure or intrinsic dependencies among features, the selected feature set may not effectively represent the data. For instance, many problem domains contain feature spaces that have pairwise dependencies. In natural language processing or text mining applications, each feature is considered as a word or term, and those that have similarity with other words are called synonyms[1]. Furthermore, in biological applications, some genes work in groups in which there are interdependencies between genes. In our experiments, we took into consideration this dependency information to reduce the feature space while still ensuring the readability

• Thosini Bamunu Mudiyanselage and Yanqing Zhang are with the Department of Computer Science, Georgia State University, Atlanta, GA 30302, USA. E-mail: tbamunumudiyanselag1@student.gsu.edu; yzhang@gsu.edu.
∗ To whom correspondence should be addressed.
Manuscript received: 2018-06-20; accepted: 2018-08-02

and interpretability of the original data space. We used a graph $G(V, E)$ to encode these dependencies, where V is the set of all features and E is the set of all pairwise dependencies among features. If there are m nodes $V = \{V_1, V_2, \ldots, V_m\}$ and a set of n edges $E = \{E_1, E_2, \ldots, E_n\}$ in $G(V, E)$, node V_i corresponds to the i-th feature and E_j represents a pairwise dependency. Learning the representations of nodes in a network or graph with preserving certain properties of the network is beneficial for various analysis tasks and has attracted significant attention in recent years[2].

Filter feature-selection methods have attracted more attention than wrapper methods due to their computational simplicity. However, there remains a problem of whether they can find the optimal feature subset. To explain, the more popular method calculates the similarity between each feature and the class output, and then selects the top k features with high similarity values, where k is an integer. Recent studies show, however, that to identify the optimal feature subset, we must also consider the inter-feature similarity rather than only the similarity of features with the class output. For example, assume that we have four features A, B, C, and D and have found that B, D, C, and A is the descending order of similarity with the output variable E. Now, we might select only the two most similar features B and D to further reduce the number of features. But the fact that A and B together can discriminate the label of the output class better than B and D illustrates the drawback of the above feature-selection method.

Therefore, we must also consider the similarity (mutual information) between new features and those already selected. In other words, we must consider the discriminative power of the class label of a new feature given that some features have already been selected. To do so, we apply the Markov chain, which is a stochastic process that enables us to predict the future given the current state. Here the current state is a subset of already selected features and the future is the newly selected feature. The Markov chain is a random process, in which a system transitions from one node to another in discrete time steps and this visitation process has a strong connection with the pairwise dependencies of each pair of nodes. We assign each node a value, depending on the similarity of that feature node with the output label, and each edge represents a similarity value for a pair of feature nodes. Then, we allow random transitions between feature nodes and at a specific time we find that the vector of probabilities for visiting each node is fixed. We can then rank each node or feature with these fixed probability values.

Highly ranked feature nodes indicate the most highly connected optimal feature subset and we evaluate the performance of this set using classification algorithms to predict its output. In our experiments, we used the widely applied machine learning package scikit-learn and two publicly available datasets.

In this paper, we describe related work in Section 2 and present our proposed method in Section 3. Section 3 also contains a detailed description of our new feature score algorithm and a proof of the concept. In Section 4, we discuss our experimental results and we draw our research conclusions in Section 5.

2 Related Work

We are interested in filter methods as they mainly rely on ranking features and then select a more highly ranked set of features. These methods are a good solution for many applications due to their simplicity (light computation) and the fact that they avoid overfitting in that they do not rely on learning algorithms. Recently, many researchers have proposed filter feature-selection methods that use various ways to measure the relevance of variables to differentiate between classes. One such application is gene microarray analysis and in Ref. [3], the authors discussed two strategies: ranking a score value based on the significance of variables and space search methods that use optimization functions. The authors in Ref. [4] proposed a method that first divides genes into subsets, then selects informative smaller subsets of genes to address the problem in which weakly ranked genes could perform well in classification. The authors in Refs. [5, 6] suggested the use of optimization functions such as binary swarm optimization and genetic algorithms to find the most relevant subset of features in generic (normal) and sensed-image data, respectively. Text classification is another application that is greatly affected by a high-dimensional feature space, and the authors in Ref. [7] suggested a new supervised feature-selection approach that defines a similarity value between a term and a class as a score for ranking. The authors in Ref. [8] introduced a conventional combinatorial optimization formulation for similarity-preserving feature selection, then extended it with a sparse multiple-output regression formulation to improve its efficiency and effectiveness. A group incremental

rough feature-selection algorithm based on information entropy was introduced in Ref. [9], and a novel 3-D segmentation technique within a random forest classification framework is presented by the authors in Ref. [10]. One of the main drawbacks of filter methods is that the less important features on their own can be more informative and have a good relation with the output when combined with other features. Also, filter methods always rely on class labels, which are not always available.

The challenge of this problem is that data is always linked, which invalidates assumptions of independence and identical distributions. Recently developed unsupervised feature-selection algorithms try to find the best set of features by finding this hidden structures in data. The primary technique used by these methods is clustering. Most of the works which use graphs for their analysis focus on representing the data space as a graph and cluster data to find relevant features. The authors in Ref. [11] proposed such an algorithm in which data are first divided into clusters using graph-theoretic clustering and then the most representative feature that is strongly related to the target is selected from each cluster to form a feature subset. Among all such similar work, only the authors in Ref. [12] used graph representation of features where each node is a feature. But they also first clustered the feature vectors based on similarity and rank features in each cluster to find best feature set for the classification. The authors in Ref. [13] used a hybrid of particle swarm optimization algorithm with genetic operators for the unsupervised feature selection and K-means clustering is used to evaluate the effectiveness of obtained feature subsets. It is a known fact that, high-dimensional space K-means is very inefficient and always fails to address complex data clusters. Also, we still have the problem of whether the feature subset comprising features from each cluster can really differentiate between all the data instances in different classes.

When we have high dimensional feature space with less number of samples, learning algorithms always tend to remember those training data and then badly perform on the testing data which is called over-fitting problem. To tackle this issue, there is another feature selection method which is called embedded method. These methods try to embed feature selection phase into learning algorithm construction where both phase complement each other. Among embedded methods, sparse learning methods are popular in recent research work. Sparse learning based methods aim to minimize the fitting errors in learning algorithm by using sparse regularization terms. The sparse regularizer forces some feature coefficient to be small and exactly zero and corresponding features can be eliminated. In Lasso, $l1 - norm$ regularization term is added to achieve feature selection. But still these methods suffer in an environment where the class labels are not available for the data set. As a result unsupervised sparse learning based feature selection has received increasing attention in recent years. MCFS[14] is one of the first unsupervised sparse learning based method and it proposes to select features which cover cluster structure where clusters are made using spectral analysis. Very recent works use the same process with small changes to previous work in unsupervised sparse learning. The work in Ref. [15] suggests to select a sample subset which includes the most important samples to build an initial feature selection model first and then improved by generalization involving other important samples in contrast to feature selection methods equivalently consider the samples to select important features. The main disadvantage of the above methods is they do not consider data dependencies or feature relations well as feature-representation coefficient matrix is fixed and cannot be fine-tuned according to the structure of data.

To solve the above problems, the authors in Ref. [16] suggested an unsupervised feature selection algorithm that combines sparse feature selection and manifold structure learning. Specifically, they first utilize feature representation to construct the model. After that, the feature-representation coefficient matrix is dynamically adjusted based on the data similarity matrix and combine with sparse learning. Adding a graph regularization term into main objective function which is the combination of feature-level representation loss function plus a regularization term is done in Ref. [17]. The authors in Ref. [18] used another spectral feature selection model by embedding a graph regularize into the framework of joint sparse regression for preserving the local structure of data. Their objective is to take subspace learning and joint sparse regression together into account. Meanwhile some research work proposes to learn and update these graph similarity matrices iteratively to avoid high dimensional and noisy data problem. Reference [19] proposes a Low-rank Sparse Subspace (LSS) clustering method via dynamically learning the affinity matrix from low-

dimensional space of the original data. The authors in Ref. [20] also proposed to learn the graph matrix measuring the similarity of samples for preserving the local structure among samples. Furthermore, they proposed to iteratively update the graph matrix. To achieve a desirable feature subset, the authors in Ref. [21] learned the reconstruction graph and selective matrix simultaneously, instead of using a predetermined graph. The authors in Ref. [22] proposed to learn clustering labels of the training samples by a subspace clustering method, and features that can well preserve the cluster labels are selected. In all above mentioned methods, we can see that the performance of feature selection directly depends on the clustering performance of the original data. In some work, we see that the graph matrix is created iteratively or on a low dimensional space and the created graph sometimes can lose important features which already have strong dependencies with remaining features and have higher discriminative power. Also we are not sure that the feature dependencies existing on original data space will remain the same in low dimensional data space with iterative learning of the similarity matrix, and hence it will learn the data distribution and subspace structure of the data correctly. Moreover, all of the above mentioned methods are embedded feature selection methods and in each learning phase, subset of features will be considered. In some approaches, subspace learning and clustering are completely separate from feature selection. Finally, aggregation of all the steps is done based on the performance of each iteration or learning phase using methods such as weighted aggregation. Thus, these methods completely lose the dependency information or important meaning of the original feature space. Most of the works we discussed still use data manifold structure learning where only few works consider making feature space into graph. But still their objective of embedding graph regularization term into main objective function of the learning model is to give the same coefficient to similar features and thus remove them as redundant features in final step. But in our approach, we have considered feature dependencies or similar features together and have identified as best features which have the discriminative power together between different class labels.

Further above mentioned methods are still based on learning algorithms and still we cannot completely eliminate the drawbacks known from wrapper methods, such as overfitting issue and being bias towards only one specific algorithm such as KNN or Support Vector Machine. Another main issue is optimizing the hyper-parameters in above mentioned embedded methods. Most of the above methods are sparse feature selection on embedded learning along with subspace learning. So the combination of loss function and different regularization terms is used as the main objective function and hyper-parameters are used to balance the contribution of these regularization terms. Some researchers have taken steps on tuning these parameters while some are not. This is also critical as if the value of parameter (penalty) is high, less number of features will be selected, while more features are selected with low penalty. One such example is in Ref. [23] where they combined a hidden Markov model, a localized feature saliency measure, and two t-Student distributions to describe the relevant and non-relevant features, to accurately model emission parameters for each hidden state. The increase of variables to be estimated from data becomes a problem as this method needs many parameters to estimate as saliency parameters and others required by the model.

Data variance is the simplest unsupervised feature-selection method, in which features with high variance are identified as having the highest data representation power. However, we still do not know that these features have the best discriminative power between different classes. The Laplacian score[24], the most famous unsupervised feature-selection method, finds the subset of features that preserves the local structure of the data more than the global structure. In contrast to all above mentioned unsupervised or graph based methods, our new approach is to use a graph to represent feature space. Also we focus more on the physical meaning of the whole feature space for which highly connected features always have discriminative power between different classes of a main classification subject.

3 Proposed Method

In our proposed method, the novelty is we build the graph based on features in contrast to majority of work which consider graph representation of data instances. Then the score of each feature is calculated based on the Markov chain process where random transitions are entered on a graph, which contains feature dependency information. The goal is to identify a highly connected feature set that contains the physical meaning of the real problem and to prove that this set of features has the best

discriminative power. We detail the steps of the new algorithm and validate this new method in the following subsections.

3.1 Algorithm

Input: All the features f_i, where $i = 1, \ldots, m$.
Output: Ranking values (visiting probabilities) for each feature f_i.

(1) Find the similarity values (such as Euclidean distance) between all feature pairs f_i and f_j and create an edge between those two features if their similarity value is greater than a certain threshold.

(2) Create a similarity graph $G = (V, E)$, where V is the set of features and E is the edge set that indicates the similarity between each pair of features.

As we can see in Fig. 1, features which have more relation are connected in the graph. But some feature nodes such as node 2 and node 4 are not connected as they do not have significant relationship between them.

(3) (Optional) Calculate the feature relevance value r_i (such as correlation or mutual information) with an output label for each node v_i, only when class label information is available.

(4) Define a row normalized adjacency (transition) matrix P where each element $p_{i,j}$ corresponds to the transition probability from node i to node j.

$$p_{i,j} = r_i / (\textit{number of connected nodes to node } i).$$

Now we can calculate the transition probability from one node to another according to the above equation on the feature graph. In the given example, Fig 2, as node 1 is connected with three nodes 2, 3 and 4 the transition probabilities from node 1 to each three nodes are equal to 1/3. Thus we can calculate transition probability

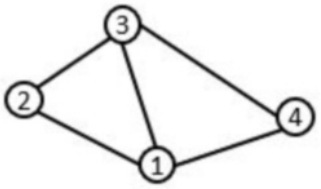


Fig. 1 An example of feature similarity graph.

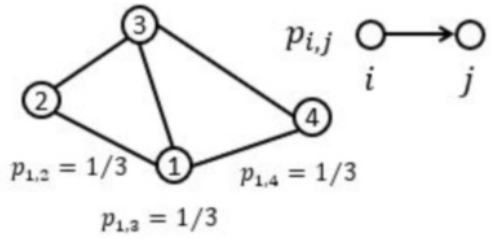


Fig. 2 Transition probabilities on the graph.

matrix P for the given graph and each element of the matrix P corresponds to $p_{i,j}$ (transition probability value from node i to node j).

Calculated transition probability matrix for the above sample graph is given below.

$$P = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{pmatrix} 0 & 1/2 & 1/3 & 1/2 \\ 1/3 & 0 & 1/3 & 0 \\ 1/3 & 1/2 & 0 & 1/2 \\ 1/3 & 0 & 1/3 & 0 \end{pmatrix} \end{array}.$$

After we calculate the transition probability matrix, next step is to calculate the visiting probability of each node. Let's take x_1^t as the probability of visiting node 1 at time t. So, the visiting probabilities of each node at any given time can be calculated as follows.

$$\begin{aligned} x_1^t &= x_2^{t-1} \cdot p_{2,1} + x_3^{t-1} \cdot p_{3,1} + x_4^{t-1} \cdot p_{4,1}, \\ x_2^t &= x_1^{t-1} \cdot p_{1,2} + x_3^{t-1} \cdot p_{3,2}, \\ x_3^t &= x_1^{t-1} \cdot p_{1,3} + x_3^{t-1} \cdot p_{2,3} + x_4^{t-1} \cdot p_{4,3}, \\ x_4^t &= x_1^{t-1} \cdot p_{1,4} + x_3^{t-1} \cdot p_{3,4}. \end{aligned}$$

Next, all the above equations can be simplified into a matrix notation as $x^t = P^{\mathrm{T}} \cdot x^{t-1}$, where P^{T} is the transpose of the transition probability matrix.

(5) Let x^0 be the starting vector of the visiting probability of each node whose values are randomly initialized.

The next step is to allow random transitions for many iterations until the visiting probabilities of nodes are converged. It is known that if the matrix P is a primitive matrix which means that the related graph is well connected, the visiting probabilities will be converged with any randomly initialized vector of visiting probabilities.

(6) While (x^t has not converged)

$$x^t = P^{\mathrm{T}} x^{t-1},$$

where t is time.

(7) Output x, which is the converged visiting probability vector where each entry is a fixed value.

Then, features relevant to nodes with high visiting probability can be grouped as a highly ranked feature subset and be used to measure the accuracy of various classification methods.

3.2 Proof of concept

3.2.1 Objective function of feature selection

Recall that the objective of many feature-selection methods is to find a feature set that preserves the local structure of the data space. Given a dataset $X \in \mathbf{R}^{n \times d}$ with n instances and d features, the pairwise

similarity among instances is encoded in an affinity matrix $S \in \mathbf{R}^{n\times n}$. The affinity matrix S is symmetric and its (i, j)-th entry indicates the similarity between the i-th instance x_i and the j-th instance x_j, where the larger is the value of S_{ij}, the more similar are x_i and x_j. Suppose that we want to select the k most relevant features from the all feature set F. One way to do so is to maximize their utility as follows:

$$\max \sum_{f\in F} \mathrm{SC}(f) = \max \sum_{f\in F} f' \overline{S} f \tag{1}$$

where $\mathrm{SC}(f)$ is a utility function for feature f. $\overline{S}$ denotes the transformation (e.g., scaling, normalization, etc.) result of the original feature vector f. $\overline{S}$ is a refined affinity matrix obtained from the affinity matrix S. The maximization problem in Eq. (1) shows that we would select a subset of features from F that preserves the data similarity structures defined in $\overline{S}$. This problem is usually solved by the greedy selection of the top k features that maximizes their individual utility $f' \overline{S} f$[13].

3.2.2 Connection to the objective function

Here, we present our theoretical analysis of the connection between proposed algorithm and another method known as the Laplacian score, which is an unsupervised feature-selection method in which the aim is related to the objective function described above. The importance of a feature can be thought of as the degree to which it respects the graph structure. A good feature is one in which two data points are close to each other if and only if there is an edge between these two points. To choose good features, we must minimize the Laplacian score of each feature r as follows:

$$L_r = \sum_{ij} (f_{r_i} - f_{r_j})^2 S_{ij} = f' \overline{S} f \tag{2}$$

Features that respect the pre-defined graph structure have a smaller $(f_{r_i} - f_{r_j})$ value and a bigger S_{ij} value, i.e., $f_{r_i} \approx f_{r_j}$. In other words, two data instances i and j with respect to feature r are very close and the difference between them is a very small value. Having a small value means this specific feature r respects the pre-defined graph structure.

Let's assume that data points in S are ordered according to the class to which they belong, so that $x_1, \ldots, x_n$ are in the first class, and $x_{n_1+1}, \ldots, x_{n_1+n_2}$ are in the second class. Thus, we can write S as follows:

$$S = \begin{pmatrix} S_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & S_c \end{pmatrix} \tag{3}$$

where data belong to c number of classes. By taking each data similarity matrix in each class from $S_1, \ldots, S_c$, we have a set of features relevant to each matrix that preserves the local structure of the data in each of the $S_1, \ldots, S_c$ similarity matrices.

Assume we have selected two features f_1 and f_2 for the data in class 1 (i.e., S_1). Both features have smaller values for $(f_{1_i} - f_{1_j})$ and $(f_{2_i} - f_{2_j})$ with respect to two data instances x_i and x_j, which belong to the same class (class 1).

$$(f_{1_i} - f_{1_j}) \approx (f_{2_i} - f_{2_j}) \approx 0 \Rightarrow f_{1_i} - f_{1_j} = f_{2_i} - f_{2_j} \Rightarrow$$
$$f_{1_i} - f_{2_i} = f_{1_j} - f_{2_j} \Rightarrow f_1 \propto f_2.$$

We can see that f_1 and f_2 are also similar based on the fact that they are the best features with which to discriminate any one instance from the instances belonging to class 1. Taking the similarity between features also results in a common objective function of feature selection. Thus, we can conclude that our approach is connected with the above objective function.

4 Experimental Results

We conducted experiments using the machine learning package scikit-learn and publicly available datasets. In the subsections below, we provide specifics about the data we used and our experimental results.

4.1 Data

In the experiments, we mainly used two face image datasets called COIL20 and ORL, which we obtained from the scikit-feature open-source feature selection repository at Arizona State University[25]. Both data sets have a 1024-dimensional feature space in which all the features have numerical and continuous values. The first data set has 20 different objects and the second has 40. Table 1 lists the details of these two datasets.

Later, to demonstrate that the performance of a method is greatly influenced by how we calculate the similarity between two features, we compared our proposed feature-selection method with two other datasets, BASEHOCK and GLIOMA, which are text and bio data sets, respectively.

4.2 Results

We compared our new feature-selection algorithm

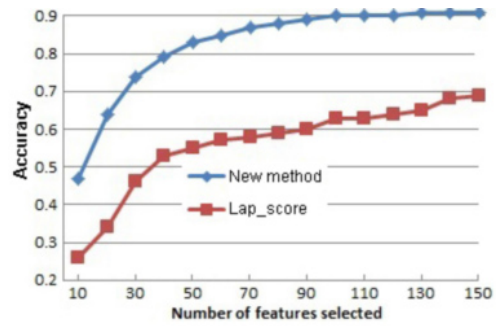
Table 1 Two image datasets used for performance comparison.

Dataset	Instances	Features	Classes
COIL20	1440	1024	20
ORL	400	1024	40

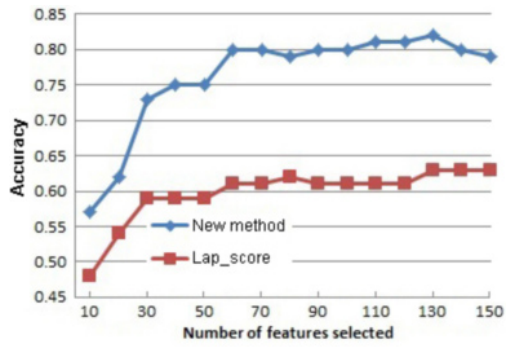
with various other feature-selection methods on the above-mentioned datasets. To do so, we used each feature-selection method to find the 10, 20, 30, and up to 150 best features and then determined their classification accuracies using three different classification algorithms: linear Support Vector Machine (SVM), Naive Bayes, and decision tree learning.

First, we compared our new feature-selection method with the Laplacian score, which selects features that can best preserve the data manifold structure. We determined the accuracy for different sets of features given by both methods on the COIL20 face image data set.

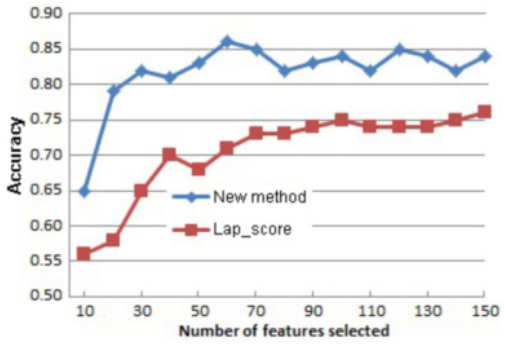
Figure 3 shows the experimental results for the unsupervised Feature-Selection (FS) method Laplacian score and our new method. As we can see, for all three classification methods, the best feature sets from 10 to 150 were generated by our new feature-selection method, which showed greater classification accuracy than the Laplacian score feature-selection method. We also tested the Laplacian score algorithm on the ORL image data set and found it to have a very low accuracy of about 3%, whereas the new algorithm achieved 62% accuracy on the 10 best selected features using the SVM classification algorithm.



(a) SVM classification



(b) Naive Bayes classification



(c) Decision tree classification

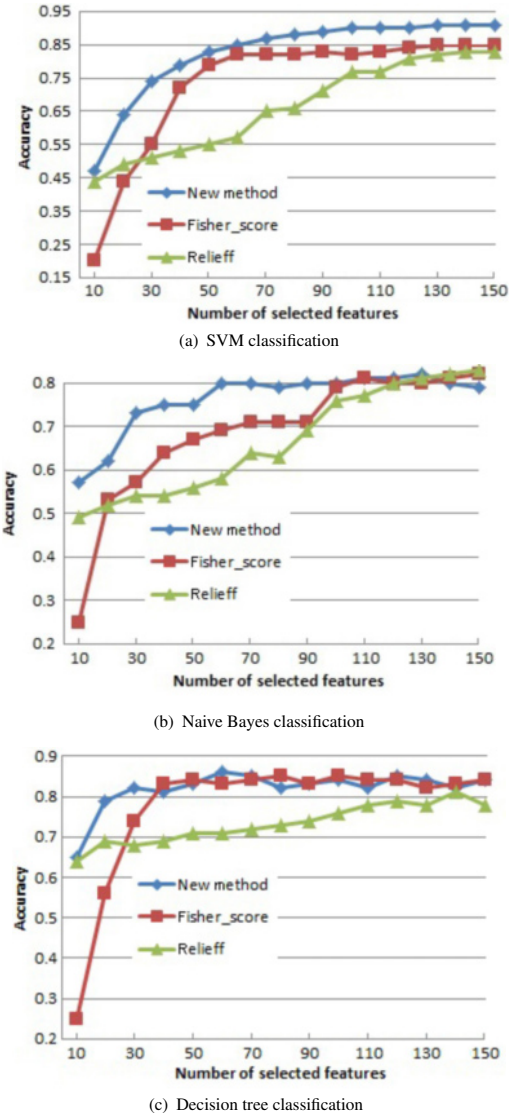
Fig. 3 Performance comparison of new FS with unsupervised FS.

Next, we compared two well-known supervised feature-selection methods known as Fisher score[26] and Relieff[27] with our new unsupervised feature-selection method. The Fisher score selects features based on feature values of samples within the same class that are small, and those from different classes that are large. Relieff selects features of separate instances from different classes. In our experiment, to determine the strength of the new feature-selection method, we compared its performance in supervised feature selection without the use of any class labels. Figure 4 shows comparisons of the performances of the Fisher score, Relieff, and the new feature-selection method on the COIL20 image dataset, which has 20 different classes.

By a careful examination of the performances in Fig. 4, we can clearly see that new feature-selection method almost outperforms Relieff in almost all cases and is very competitive with the Fisher score in a few cases. We have to keep in mind that the new feature-selection method used no label information whereas the others did.

Also, we can see that for all three classification methods, the highest accuracy for the best 10 features was realized by the newly proposed unsupervised feature-selection method. Also, the best selected features in the range from 10 to 150 are from the new feature-selection method which achieves the highest accuracy. Specifically, the values realized were 91% accuracy for 130 features using the SVM classification algorithm, 82% accuracy for 130 features by the Naive Bayes classification algorithm, and 86% accuracy for 60 features by the decision tree classification algorithm. Thus, we find that the new selection method demonstrated the best performance with the smallest number of features, clearly outperforming the other methods.

Figure 5 shows a comparison of the performances of the Fisher score, Relieff, and new feature-selection

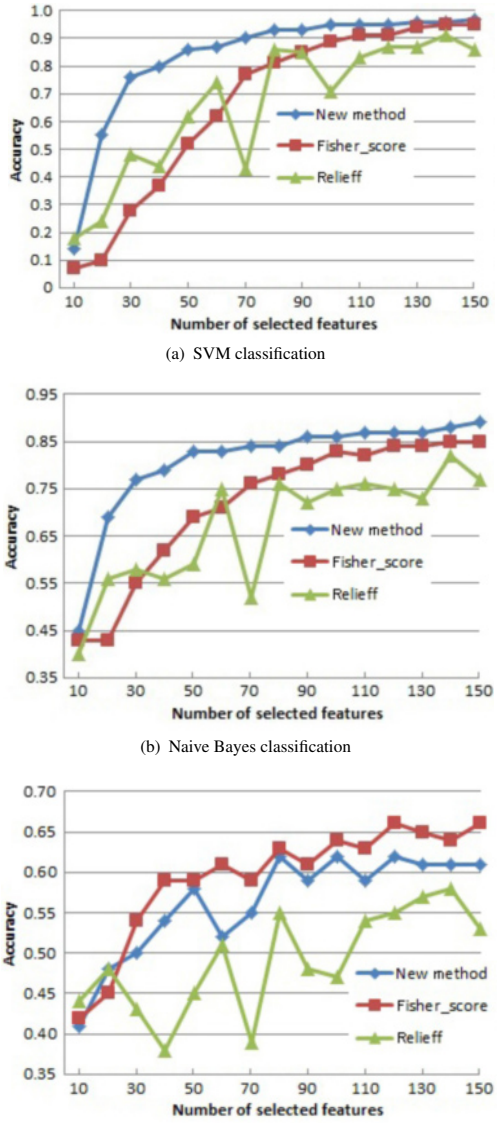


(a) SVM classification

(b) Naive Bayes classification

(c) Decision tree classification

Fig. 4 Performance comparison of new FS with supervised FS on COIL20 image dataset.



(a) SVM classification

(b) Naive Bayes classification

(c) Decision tree classification

Fig. 5 Performance comparison of new FS with supervised FS on ORL image dataset.

method on the ORL image dataset, which has 40 different classes. Again, we see that without the benefit of any label data, the new feature-selection method achieved the highest accuracy in almost all cases, being outperformed only in a few cases by the supervised feature-selection method. However, we can see that the new feature-selection method outperformed the Fisher score when using the first two classification algorithms SVM and Naive Bayes, and performed slightly worse using the decision tree classification algorithm. However, in all three cases, the new feature-selection method outperformed supervised Relieff feature selection. In addition, the fluctuations in accuracy for the new feature-selection method are significantly reduced after it has gained some considerable experience, which means that it can guarantee to yield the best features. From Fig. 3, we can see that the accuracy change of the new feature-selection method is $\pm 1\%$ for two consecutive points for the 60–150 range of best features, whereas that of Relieff is greater than $\pm 25\%$ for all three classification algorithms.

We also conducted experiments using the new feature-selection method on heterogeneous datasets. We found that there is a relationship between the calculation method used to determine the similarity between features and the classification accuracy realized on each type of dataset. This again confirms the importance of taking feature dependencies into

consideration in feature-selection processes.

Table 2 shows the classification accuracies achieved using the SVM algortihm with the new feature-selection method and Fisher score on three different datasets, i.e., text, microarray, and image. We can clearly see that new feature-selection method, which took no label information into consideration, outperformed the Fisher score supervised feature-selection method only for the image data set. The reason for this is that to determine the similarity between two features, we calculate the Euclidian distance between them. This gives a strong sense of an image dataset because we try to learn important features by representing the feature space using closely related features or pixels on the image as they appear. Thus, the original structure of the feature space is preserved, which supports the motivation for developing this new feature-selection method.

In the same way, we must develop new calculation methods that suit other datasets and expand this new feature-selection method such that it can better represent the original feature space and find the best feature set.

5 Conclusion

Most real life problems are inherently unsupervised. The expectation that there will always be label data is not realistic and labeling is expensive. Yet most existing filter feature-selection methods are supervised and rely on labeled data. In the same way, existing unsupervised feature-selection methods do not perform well in high-dimensional feature space. Our newly proposed feature-selection method offers a powerful solution and it can be used in both supervised and unsupervised environments and does not depend on labeled data. In our experiments, it also outperformed existing methods in a high-dimensional feature space. Comparing its accuracy when using different classification algorithms, we found the performance of the new feature-selection method to be independent of any specific learning algorithm. Also, the structure and dependencies of a feature space are carefully considered in the calculations of this new method, whereas most existing filter feature-selection methods assume that the features are independent of each other and identically distributed. Based on our performance comparisons, we conclude that our newly developed feature-selection method performs well on image data sets as it considers existing feature dependencies, and also performs better than both supervised and unsupervised feature-selection methods.

Table 2 Comparison of accuracies on heterogeneous datasets.

Number of features	Text		Microarray		Image	
	New	Fisher	New	Fisher	New	Fisher
10	0.51	0.80	0.62	0.61	**0.47**	0.20
20	0.52	0.85	0.56	0.73	**0.64**	0.44
30	0.53	0.89	0.56	0.67	**0.74**	0.55
40	0.54	0.93	0.52	0.72	**0.79**	0.72
50	0.54	0.93	0.69	0.72	**0.83**	0.79
60	0.54	0.95	0.60	0.74	**0.85**	0.82
70	0.54	0.95	0.53	0.71	**0.87**	0.82
80	0.55	0.95	0.58	0.74	**0.88**	0.82
90	0.55	0.95	0.58	0.79	**0.89**	0.83
100	0.56	0.95	0.53	0.76	**0.90**	0.82

References

[1] C. Fellbaum, *WordNet*. Wiley Online Library, 1998.

[2] Y. Wang, Y. Yao, H. Tong, F. Xu, and J. Lu, A brief review of network embedding, *Big Data Mining and Analytics*, vol. 2, no. 1, pp. 35–47, 2019.

[3] C. Lazar, J. Taminau, S. Meganck, D. Steenhoff, A. Coletta, C. Molter, V. de Schaetzen, R. Duque, H. Bersini, and A. Nowe, A survey on filter techniques for feature selection in gene expression microarray analysis, *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 9, no. 4, pp. 1106–1119, 2012.

[4] A. Sharma, S. Imoto, and S. Miyano, A top-r feature selection algorithm for microarray gene expression data, *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 9, no. 3, pp. 754–764, 2012.

[5] L. Cervante, B. Xue, M. Zhang, and L. Shang, Binary particle swarm optimization for feature selection: A filter based approach, in *IEEE Congress on Evolutionary Computation*, Brisbane, Australia, 2012, pp. 1–8.

[6] M. Pedergnana, P. R. Marpu, M. D. Mura, J. A. Benediktsson, and L. Bruzzone, A novel technique for optimal feature selection in attribute profiles based on genetic algorithms, *IEEE Transactions on Geoscience and Remote Sensing*, vol. 51, no. 6, pp. 3514–3528, 2013.

[7] T. Basu and C. A. Murthy, Effective text classification by a supervised feature selection approach, in *IEEE* 12^{th} *International Conference on Data Mining Workshops*, Brussels, Belgium, 2012, pp. 918–925.

[8] Z. Zhao, L. Wang, H. Liu, and J. Ye, On similarity preserving feature selection, *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 3, pp. 619–632, 2013.

[9] J. Liang, F. Wang, C. Dang, and Y. Qian, A group incremental approach to feature selection applying rough set technique, *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 2, pp. 294–308, 2014.

[10] M. Yaqub, M. K. Javaid, C. Cooper, and J. A. Noble, Investigation of the role of feature selection and weighted voting in random forests for 3-D volumetric segmentation,

IEEE Transactions on Medical Imaging, vol. 33, no. 2, pp. 258–271, 2014.

[11] Q. Song, J. Ni, and G. Wang, A fast clustering-based feature subset selection algorithm for high-dimensional data, *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 1, pp. 1–14, 2013.

[12] Z. Zhang and E. R. Hancock, A graph-based approach to feature selection, in *Graph-Based Representations in Pattern Recognition*. Springer, 2011, pp. 205–214.

[13] L. M. Abualigah and A. T. Khader, Unsupervised text feature selection technique based on hybrid particle swarm optimization algorithm with genetic operators for the text clustering, *The Journal of Supercomputing*, vol. 73, no. 11, pp. 4773–4795, 2017.

[14] D. Cai, C. Zhang, and X. He, Unsupervised feature selection for multi-cluster data, in *16^{th} ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Washington, DC, USA, 2010, pp. 333–342.

[15] W. Zheng, X. Zhu, G. Wen, Y. Zhu, H. Yu, and J. Gan, Unsupervised feature selection by self-paced learning regularization, *Pattern Recognition Letters*, https://doi.org/10.1016/j.patrec.2018.06.029.

[16] C. Lei and X. Zhu, Unsupervised feature selection via local structure learning and sparse learning, *Multimedia Tools and Applications*, vol. 77, no. 22, pp. 29605–29622, 2018.

[17] R. Hu, X. Zhu, D. Cheng, W. He, Y. Yan, J. Song, and S. Zhang, Graph self-representation method for unsupervised feature selection, *Neurocomputing*, vol. 220, pp. 130–137, 2017.

[18] X. Zhu, X. Li, S. Zhang, C. Ju, and X. Wu, Robust joint graph sparse coding for unsupervised spectral feature selection, *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 6, pp. 1263–1275, 2017.

[19] X. Zhu, S. Zhang, Y. Li, J. Zhang, L. Yang, and Y. Fang, Low-rank sparse subspace for spectral clustering, *IEEE Transactions on Knowledge and Data Engineering*, doi:10.1109/TKDE.2018.2858782.

[20] X. Zhu, S. Zhang, R. Hu, Y. Zhu, and J. Song, Local and global structure preservation for robust unsupervised spectral feature selection, *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 3, pp. 517–529, 2018.

[21] M. Luo, F. Nie, X. Chang, Y. Yang, A. G. Hauptmann, and Q. Zheng, Adaptive unsupervised feature selection with structure regularization, *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 4, pp. 944–956, 2018.

[22] P. Zhu, W. Zhu, Q. Hu, C. Zhang, and W. Zuo, Subspace clustering guided unsupervised feature selection, *Pattern Recognition*, vol. 66, pp. 364–374, 2017.

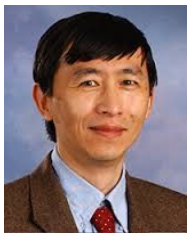
[23] Y. Zheng, B. Jeon, L. Sun, J. Zhang, and H. Zhang, Students t-Hidden Markov model for unsupervised learning using localized feature selection, *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 10, pp. 2586–2598, 2018.

[24] X. He, D. Cai, and P. Niyogi, Laplacian score for feature selection, in *Advances in Neural Information Processing Systems 18*. Cambridge, MA, USA: MIT Press, 2005.

[25] J. Li, K. Cheng, S. Wang, F. Morstatter, R. P. Trevino, J. Tang, and H. Liu, Feature selection: A data perspective, arXiv preprint arXiv:1601.07996, 2016.

[26] Q. Gu, Z. Li, and J. Han, Generalized fisher score for feature selection, in *Proc. 27^{th} Uncertainty in Artificial Intelligence Conf.*, Barcelona, Spain, 2011, pp. 266–273.

[27] M. Robnik-Sikonja and I. Kononenko, Theoretical and empirical analysis of ReliefF and RReliefF, *Machine Learning*, vol. 53, nos. 1&2, pp. 23–69, 2003.



Yanqing Zhang is a full professor of the Computer Science Department at Georgia State University, Atlanta, USA. He received the PhD degree in computer science from the University of South Florida in 1997. His research interests include hybrid intelligent systems, computational intelligence, machine learning, data mining, deep learning, fuzzy logic, granular neural networks, bioinformatics, brain informatics, health informatics, computational web intelligence, granular computing, parallel computing, green computing, Yin-Yang computation, nature-inspired computing, and security. He received Outstanding Academic Service Award at IEEE 7th International Conference on Bioinformatics & Bioengineering (IEEE BIBE 2007), Achievement Award of the 2007 World Congress in Computer Science, Computer Engineering and Applied Computing, and 2005 IEEE-Granular Computing Outstanding Service Award at 2005 IEEE International Conference on Granular Computing.

Thosini Bamunu Mudiyanselage is a PhD student of the Computer Science Department at Georgia State University, Atlanta, USA. She received the bachelor of science with First Class Honors from University of Kelaniya, SriLanka. She was awarded the Gold Medal for the best performance in computer science in 2014 from the University of Kelaniya. In 2016, she moved to USA to pursue her PhD in computer science at Georgia State University. She worked as a lecturer in computer science at University of Kelaniya before joining the PhD program. Her current research interests include big data mining, semantic networks, fuzzy logic, machine learning, and deep learning.