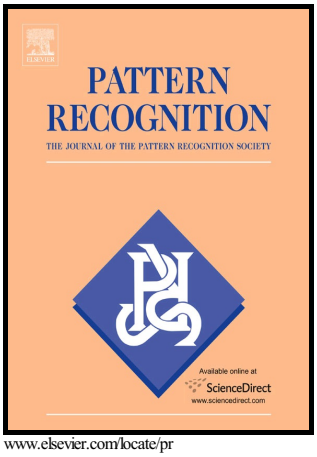
A Survey on semi-supervised feature selection methods

Razieh Sheikhpour, Mehdi Agha Sarram, Sajjad Gharaghani, Mohammd Ali Zare Chahooki



www.elsevier.com/locate/pr

PII: S0031-3203(16)30354-5
DOI: http://dx.doi.org/10.1016/j.patcog.2016.11.003
Reference: PR5946

To appear in: *Pattern Recognition*

Received date: 16 February 2016
Revised date: 3 November 2016
Accepted date: 5 November 2016

Cite this article as: Razieh Sheikhpour, Mehdi Agha Sarram, Sajjad Gharaghan and Mohammd Ali Zare Chahooki, A Survey on semi-supervised feature selection methods, *Pattern Recognition* http://dx.doi.org/10.1016/j.patcog.2016.11.003

A Survey on semi-supervised feature selection methods

Razieh Sheikhpour[1], Mehdi Agha Sarram[1*], Sajjad Gharaghani[2], Mohammd Ali Zare Chahooki[1]

[1]Department of Computer Engineering, Yazd University, Yazd, Iran

[2]Laboratory of Bioinformatics and Drug Design (LBD), Institute of Biochemistry and Biophysics, University of Tehran, Tehran, Iran

r_sheikhpour@stu.yazd.ac.ir

mehdi.sarram@yazd.ac.ir

s.gharaghani@ut.ac.ir

chahooki@yazd.ac.ir

[*]Corresponding author: Department of Computer Engineering, Yazd University, Yazd, Iran. Tel: +989133510376

Abstract

Feature selection is a significant task in data mining and machine learning applications which eliminates irrelevant and redundant features and improves learning performance. In many real-world applications, collecting labeled data is difficult, while abundant unlabeled data are easily accessible. This motivates researchers to develop semi-supervised feature selection methods which use both labeled and unlabeled data to evaluate feature relevance. However, till-to-date, there is no comprehensive survey covering the semi-supervised feature selection methods. In this paper, semi-supervised feature selection methods are fully investigated and two taxonomies of these methods are presented based on two different perspectives which represent the hierarchical structure of semi-supervised feature selection methods. The first perspective is based on the basic taxonomy of feature selection methods and the second one is based on the taxonomy of semi-supervised learning methods. This survey can be helpful for a researcher to

obtain a deep background in semi-supervised feature selection methods and choose a proper semi-supervised feature selection method based on the hierarchical structure of them.

Keywords: Semi-supervised learning, Feature selection, Survey

1. Introduction

Dealing with high-dimensional data such as digital images, financial time series and gene expression microarrays has been a main problem in data mining and machine learning applications which requires high computational time and storage capability and leads to poor performances [1–4].

Feature selection is a significant task to confront the problem of high-dimensional data. It can be applied to reduce the dimensionality of data, remove irrelevant and redundant features, shorten the training time and improve learning performance [5–8]. A number of feature selection methods have been successfully used in a wide range of applications [4,9–39]. Feature selection methods can be categorized into three types: filter, wrapper and embedded [9,10,40–46]. Filter methods select the subset of features as a pre-processing step independently of the selected classifier [9,40,47]. Wrapper methods use a single learner as a black box to evaluate the subsets of features according to their predictive power [9,10,41]. Embedded methods perform feature selection in the process of training and are usually specific to a single learner [9,10,41]. Moreover, feature selection methods can be categorized into three types according to the class label information: supervised feature selection, unsupervised feature selection and semi-supervised feature selection [40].

Supervised feature selection methods use labeled data for feature selection and evaluate feature relevance by measuring the feature correlation with the class label [40,48–50]. Unsupervised feature selection methods evaluate feature relevance by the capability of keeping particular properties of the data, such as the variance or the locality preserving ability [1,6,40]. Because of the utilization of labeled information, supervised feature selection methods usually have better performance than unsupervised feature selection methods. However, supervised feature selection methods require sufficient labeled data which need extensive expertise and are expensive to obtain [1,6,48]. In many real-world applications,

there are abundant unlabeled data and small labeled data. To deal with the 'small labeled-sample problem', semi-supervised feature selection methods have been developed by Zhao and Liu [51] which use both labeled and unlabeled data for feature selection. Semi-supervised feature selection methods use the label information of labeled data and data distribution or local structure of both labeled and unlabeled data to evaluate feature relevance [49].

Recently, increasing attention has been directed to the study of semi-supervised feature selection, and hence, many semi-supervised feature selection methods have been proposed in the literature. However, there is no comprehensive survey on semi-supervised feature selection methods. This paper presents a comprehensive survey on semi-supervised feature selection methods, categorizes the methods from two different perspectives, summarizes them with specific details and describes advantage and disadvantage of them. To the best knowledge of the authors, this is the first comprehensive survey on semi-supervised feature selection methods which categorizes them from two different perspectives.

The survey presented by Chin et al. [42] has a broader scope, provides the basic taxonomy of feature selection and reviews supervised, unsupervised and semi-supervised feature selection methods in gene expression microarray data. It [42] only reviews some supervised, unsupervised and semi-supervised feature selection methods which have been applied for gen selection in microarray data. The survey [42] has not comprehensively reviewed semi-supervised feature selection methods and has not described characteristics, specific details, advantage and disadvantage of each semi-supervised feature selection method.

Two taxonomies presented in this survey are hierarchically such that, initially semi-supervised feature selection methods are divided into several categories. Then, each category is divided into smaller categories. The perspective of the first taxonomy is based on the basic taxonomy of feature selection methods which classifies semi-supervised feature selection methods into filter, wrapper and embedded. Some previous literature [42,52,55] has consider this taxonomy for semi-supervised feature selection methods. Since this taxonomy is very general, in this survey, each of the categories (filter, wrapper and embedded) is divided into smaller categories based on the procedures used for semi-supervised feature

selection in the literature, and characteristics of them are described. There is no such division in previous literature [42,52,55]. The perspective of the second taxonomy is based on the taxonomy of semi-supervised learning methods which divides semi-supervised feature selection methods into five categories: graph-based semi-supervised feature selection, self-training based semi-supervised feature selection, co-training based semi-supervised feature selection, support vector machine based semi-supervised feature selection, and other semi-supervised feature selection methods. These categories are also divided into smaller categories.

The rest of this paper is organized as follows: Section 2 provides an overview of semi-supervised learning. Semi-supervised feature selection is fully investigated in section 3 and the hierarchical structure of semi-supervised feature selection methods is provided in this section. Comparison of different semi-supervised feature selection methods is presented in section 4. In this section, advantage and disadvantage of semi-supervised feature selection methods are summarized and performance evaluation of them is provided. Finally, conclusion and future work are provided in section 5.

2. Semi-supervised learning

Semi-supervised learning learns from a small number of labeled data and a large number of unlabeled data [56–58]. In semi-supervised learning, certain smoothness assumptions such as cluster assumption [5] and manifold assumption must be met [56,59]. Cluster assumption states that if samples are in the same cluster, they are likely to be of the same class [59]. Manifold assumption assumes that the high-dimensional data lie on a low-dimensional manifold formed by the data [57,59–61]. A variety of semi-supervised learning methods exist in the literature. These methods can be categorized into generative models, self-training, co-training, semi-supervised support vector machines (S^3VM) and graph-based methods [58,59,62–65].

2.1 Generative models

Generative semi-supervised models assume a model, $p(x,y) = p(y)\ p(x|y)$ where $p(x|y)$ is an identifiable mixture distribution. With a large amount of unlabeled data, the mixture components can be

identified; then only one labeled instance per component is needed to fully determine the mixture distribution [58,64,65].

2.2 Self-training

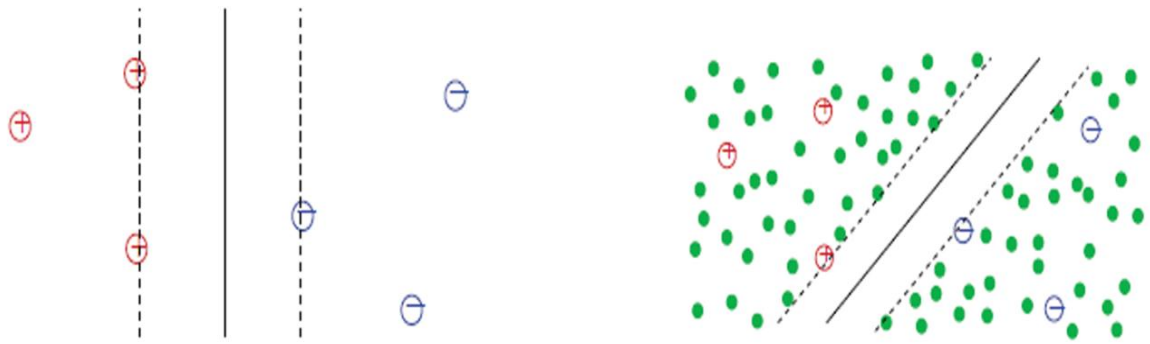
The main idea of self-training is to first train a classifier with labeled data. The classifier is then applied to predict the labels of unlabeled data. A subset of the most confident unlabeled data, along with their predicted labels, are then selected and added to the training set. The classifier is re-trained on the new training set, and the procedure repeated. Self-training is characterized by the fact that the classifier uses its own predictions to teach itself [59,63,64].

2.3 Co-training

Co-training is a semi-supervised learning method with two different classifiers which need two different sets of features on labeled data. Each classifier is trained with a set of features and used to classify the unlabeled data. The most confident predictions of each classifier on the unlabeled data are iteratively used by other classifier as the labeled training data [58,59,64,65].

2.4 Semi-supervised support vector machines

Semi-Supervised Support Vector Machines (S^3VMs) were originally called Transductive Support Vector Machines (TSVMs). The goal of S^3VMs is to use both labeled and unlabeled data to find a linear boundary which has the maximum margin. Figure 1(a) and 1(b) show the decision boundary found by SVM on labeled data and S^3VM on both labeled and unlabeled data, respectively [63,64].



(a) SVM decision boundary (b) S3VM decision boundary

Figure1: (a) With only labeled data, the linear decision boundary found by SVM. (b) With both labeled and unlabeled data, the linear decision boundary found by S^3VM [63].

2.5 Graph-based methods

During the recent years, the most active area of research in semi-supervised learning has been graph-based methods which start by constructing a graph from the training samples. The vertices of the graph are the labeled and unlabeled training samples, and the undirected edge between two vertices x_i, x_j represents the similarity of the two samples. Graph-based methods usually assume label smoothness over the graph [59,63,64].

3. Semi-supervised feature selection

In this section, various semi-supervised feature selection methods are categorized based on two different perspectives and summarized with specific details. The first perspective is based on the basic taxonomy of feature selection methods which categorizes semi-supervised feature selection methods into several categories, depending on how they interact with the learning algorithm. The second perspective is based on the taxonomy of semi-supervised learning methods which categorizes semi-supervised feature selection methods into several categories, depending on what semi-supervised learning algorithm corresponds to the procedure used in the semi-supervised feature selection method.

According to the perspective of the first taxonomy, semi-supervised feature selection methods can be categorized into filter [1,11,40,49,51,66–75], wrapper [52–55] and embedded [15,48,76–81]. Then, each category is divided into smaller categories based on the procedures used for semi-supervised feature selection in the literature. Figure 2 shows the hierarchical structure of semi-supervised feature selection methods based on the basic taxonomy of feature selection methods.

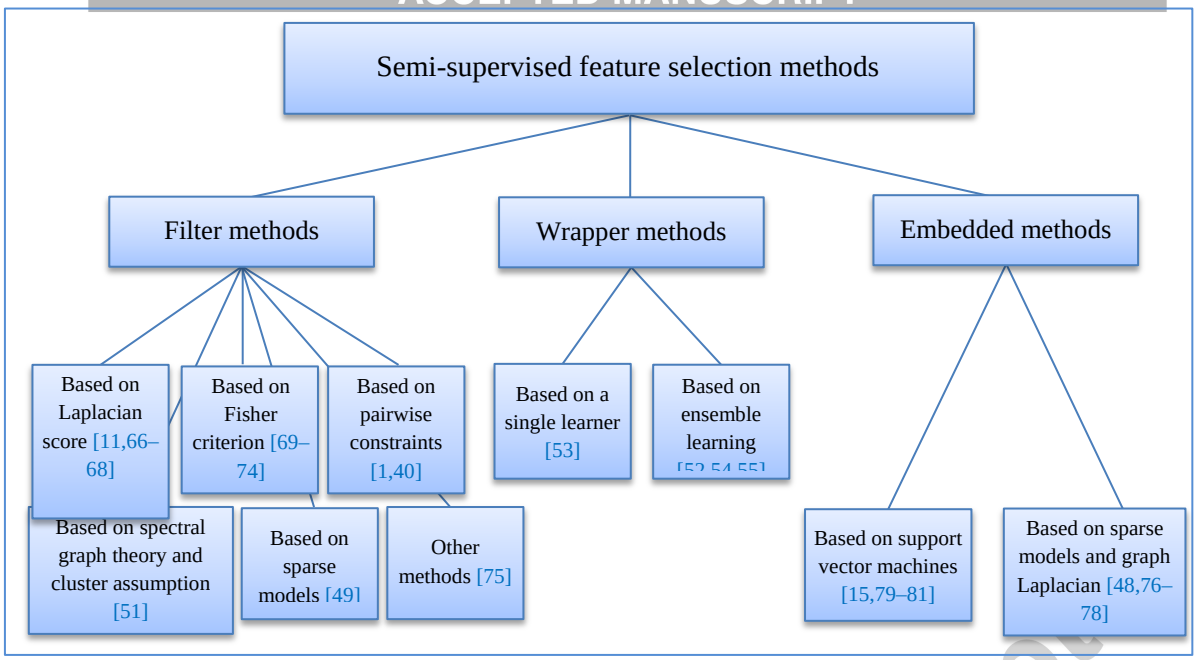


Figure 2: The hierarchical structure of semi-supervised feature selection methods based on the basic taxonomy of feature selection methods

As discussed in Section 2, semi-supervised learning methods are categorized into generative models, self-training, co-training, semi-supervised support vector machine (S^3VM) and graph-based methods. According to this taxonomy and literature review, semi-supervised feature selection methods can be categorized into five categories: graph-based semi-supervised feature selection [1,11,40,48,51,66–71,73,74,76–78], self-training based semi-supervised feature selection [52–54], co-training based semi-supervised feature selection [55], support vector machine (SVM) based semi-supervised feature selection [15,79–81] and other semi-supervised feature selection methods [49]. Figure 3 shows the hierarchical structure of semi-supervised feature selection methods based on the taxonomy of semi-supervised learning methods.

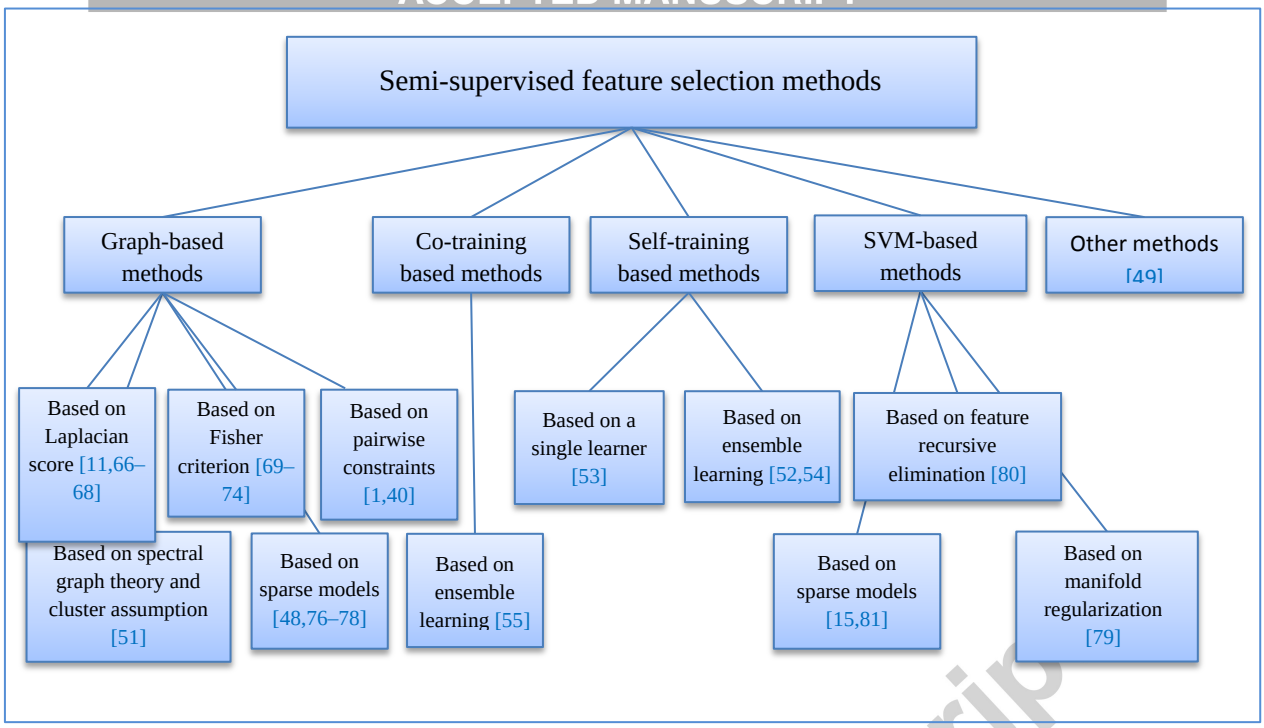


Figure 3: The hierarchical structure of semi-supervised feature selection methods based on the taxonomy of semi-supervised learning methods

Most of the semi-supervised feature selection methods use a procedure for semi-supervised feature selection that corresponds to one of the semi-supervised learning methods. Majority of semi-supervised feature selection methods construct a graph using the training samples which correspond to graph-based semi-supervised learning methods. Some semi-supervised feature selection methods use a single learner or an ensemble learning model to predict the labels of unlabeled data. They select a subset of unlabeled data along with the predicted labels and extend the initial labeled training set. The idea of these methods corresponds to that of semi-supervised learning methods based on self-training or co-training. Some semi-supervised feature selection methods perform feature selection based on semi-supervised support vector machines. The procedure used in these methods corresponds to semi-supervised SVM learning methods.

In general, according to the procedures used in most of the semi-supervised feature selection methods, they are placed in the categories of their corresponding semi-supervised learning methods. Therefore, we can categorize most of the semi-supervised feature selection methods based on the taxonomy of semi-supervised learning methods. Then, each of the categories is divided into smaller categories based on the

procedures used in literature. If the procedure used in a semi-supervised feature selection method does not correspond to each of the semi-supervised leaning methods, the method is placed in the category of other methods. In the following, the notations used in this survey are provided, and the details of semi-supervised feature selection methods are described.

Let us denote $X = [x_1, \dots, x_l, x_{l+1}, \dots, x_n] \in R^{d \times n}$ as the matrix of training data, where l is the number of labeled samples, n is the total number of the training samples and d is the dimensions of each sample. $x_i \in R^d$ $(1 \leq i \leq n)$ is the ith training sample, f_r is the rth feature and f_{ri} is the rth feature of the ith sample. Let $X_L = [x_1, \dots, x_l] \in R^{d \times l}$ indicate the labeled training samples in X, $Y_L = [y_1, \dots, y_l] \in \{1, \dots, c\}^l$ denote the label vector of training samples and c indicate the number of classes.

3.1 Semi-supervised filter feature selection methods

Semi-supervised filter feature selection methods examine intrinsic properties of the labeled and unlabeled data to evaluate the features prior to the learning tasks. Most of the semi-supervised filter feature selection methods correspond to graph-based semi-supervised feature selection methods from the second taxonomy. The graph-based methods construct the neighborhood graph and identify relevant features through measuring their capability of preserving geometrical structure of graph [11].

Semi-supervised filter feature selection methods can be divided into six categories: semi-supervised feature selection methods based on spectral graph theory and cluster assumption, semi-supervised feature selection methods based on Laplacian score, semi-supervised feature selection methods based on pairwise constraints, semi-supervised feature selection methods based on Fisher criterion, semi-supervised feature selection methods based on sparse models, and other methods.

3.1.1 Scoring in filter methods

In this subsection, we briefly introduce several supervised and unsupervised score functions used in semi-supervised feature selection methods, including variance score [82], Laplacian score [83], Fisher score [82] and constraint score [6]. Among them, variance and Laplacian scores are unsupervised, while Fisher and constraint scores are supervised.

Variance score

Variance score [82] is an unsupervised method for evaluation of the features. It uses the variance along a dimension to reflect its representative power and selects the features with the maximum variance. The variance score of the *r*th feature V_r, is defined as follows:

$$\boldsymbol{V_r = \frac{1}{n}\sum_{i=1}^{n}(f_{ri} - \mu_r)^2} \quad (1)$$

where μ_r is defined as Eq. (2):

$$\mu_r = \frac{1}{n}\sum_{i} f_{ri} \quad (2)$$

The features are sorted according to the decreasing order of V_r to select the most relevant ones.

Laplacian score

Laplacian score [83] is an unsupervised feature selection method which evaluates the features according to their locality preserving power. In other words, a feature is considered as ‘‘good’’ if two close data points in the original space are also close to each at this dimension. So, a good feature preserves the local geometrical structure of the data. Laplacian score is computed as follows:

According to the spectral graph theory, a nearest neighbor graph *G* is constructed with *n* nodes (one for each data sample), which contains an edge between two nodes *i* and *j* if the corresponding samples x_i and x_j are close, i.e. x_i is among the *k* nearest neighbors of x_j or x_j is among the *k* nearest neighbors of x_i. The edge between two connected nodes *i* and *j* is weighted as follows:

$$\boldsymbol{S_{ij} = \begin{cases} e^{-\frac{\|x_i - x_j\|^2}{t}} & \textit{if } x_i \in KNN(x_j) \textit{ or } x_j \in KNN(x_i) \\ 0 & \textit{otherwise} \end{cases}} \quad (3)$$

where *t* is a constant to be tuned. The weight matrix *S* can be redefined as follows for reducing the number of parameters.

$$S_{ij} = \begin{cases} 1 & if\ x_i \in KNN(x_j)\ or\ x_j \in KNN(x_i) \\ 0 & otherwise \end{cases} \quad (4)$$

Here *S* in Eq. (4) is a special case of *S* in Eq. (3), provided that t approaches to ∞. The weight matrix S of the graph models the local structure of the data space. For the rth feature, the Laplacian score L_r, is computed as follows:

$$\mathbf{L}_r = \frac{\tilde{\mathbf{f}}_r^T \boldsymbol{L} \tilde{\mathbf{f}}_r}{\tilde{\mathbf{f}}_r^T \boldsymbol{D} \tilde{\mathbf{f}}_r} \tag{5}$$

Where *D* is a diagonal matrix, $D_{ii} = \sum_j S_{ij}$, *L* is the Laplacian matrix defined as *L* = *D*-*S* and $\tilde{\mathrm{f}}_r$is defined as follows:

$$\tilde{\mathbf{f}}_r = \mathbf{f}_r - \frac{\mathbf{f}_r^T \boldsymbol{D} \mathbf{1}}{\mathbf{1}^T \boldsymbol{D} \mathbf{1}} \tag{6}$$

where 1=[1,…,1]^T. After calculating the Laplacian score for each feature, they are sorted in the ascending order of L_r to select the relevant features.

Fisher score

Fisher score [82] is a supervised feature selection method which seeks the features with the best discriminant ability. The basic idea of Fisher score is to maximize distances between data samples in the different classes and minimizes the distances between data samples in the same class. Given a set of data samples with label, $\{x_i, y_i\}$, $y_i \in \{1,\ldots,c\}$, $i=1,\ldots,n$, where *c* is the number of classes and n_i denotes the number of data samples in class *i*. Let μ_r denote the mean of all data samples on the *r*th feature, μ_r^i and $(\sigma_r^i)^2$ be the mean and variance of class *i* corresponding to the *r*th feature, respectively. The Fisher Score of the *r*th feature F_r is defined as:

$$\boldsymbol{F}_r = \frac{\sum_{i=1}^{c} \boldsymbol{n}_i (\boldsymbol{\mu}_r^i - \boldsymbol{\mu}_r)^2}{\sum_{i=1}^{c} \boldsymbol{n}_i (\boldsymbol{\sigma}_r^i)^2} \tag{7}$$

where $\sum_{i=1}^{c} n_i (\mu_r^i - \mu_r)^2$ is the between-class variance at the *r*th feature, and $\sum_{i=1}^{c} n_i (\sigma_r^i)^2$ is the within-class variance at the *r*th feature. After calculating the score for each feature, the features with the highest scores can be selected.

Fisher criterion [84] is an equivalent formulation of Fisher score which can be used to compute Fisher score. The criterion is as follows:

$$\boldsymbol{W = \arg\max_{W} \frac{W^T S_b W}{W^T S_w W}} \quad (8)$$

where $W \in R^{d \times c}$ (c<d) is the transformation matrix which can be used for feature selection. S_b is the between-class scatter matrix, and S_w is the within-class scatter matrix, which are defined as:

$$S_b = \sum_{i=1}^{c} n_i (\mu^i - \mu)(\mu^i - \mu)^T \quad (9)$$

$$S_w = \sum_{i=1}^{c} \left(\sum_{j=1}^{n_i} (x_j^i - \mu^i)(x_j^i - \mu^i)^T \right) \quad (10)$$

where μ denotes the total sample mean vector, μ^i the mean vector of class i, and x_j^i the jth sample in the ith class.

Constraint score

The supervision information can be expressed in two different forms: class labels and pairwise constraints. The pairwise constraints specify whether a pair of data samples belong to the same class (must-link constraints) or different classes (cannot-link constraints). Given a set of data samples $X = \{x_1, x_2, \dots, x_n\}$, the user must define the must-link constraints (M) and the cannot-link constraints (C) as [6]:

- M={(x_i , x_j) |x_i and x_j belong to the same class}
- C={(x_i , x_j) |x_i and x_j belong to the different classes}

Constraint score [6] is a supervised method for feature selection which uses pairwise constraints to evaluate the features. In this method, features with the best constraint preserving ability are selected. If there is a must-link constraint between a pair of data samples, the samples should be close to each other on a 'good' feature. If there is a cannot-link constraint between a pair of data samples, the samples should be far away from each other on a 'good' feature. Let f_{ri} denote the rth feature of the ith sample x_i. The constraint score of the rth feature is computed by minimizing C_r^1 or C_r^2:

$$C_r^1 = \frac{\sum_{(x_i,x_j)\in M}(f_{ri} - f_{rj})^2}{\sum_{(x_i,x_j)\in C}(f_{ri} - f_{rj})^2} \tag{11}$$

$$C_r^2 = \sum_{(x_i,x_j)\in M}(f_{ri} - f_{rj})^2 - \lambda \sum_{(x_i,x_j)\in C}(f_{ri} - f_{rj})^2 \tag{12}$$

where λ is a regularization parameter which balances the contributions of the two terms in Eq. (12).

Constraint score functions in Eqs. (11) and (12) can be also computed using the spectral graph theory. First, two graphs G^M and G^C are constructed both with n nodes, using the pairwise constraints in M and C, respectively. In both graphs, the ith node corresponds to the ith sample x_i. In graph G^M, an edge is placed between two nodes i and j if (x_i ,x_j) or (x_j ,x_i) belong to the subset M of must-link constraints. Similarly, in graph G^C, an edge is placed between two node i and j if (x_i ,x_j) or (x_j ,x_i) belong to the subset C of cannot-link constraints. The weight matrices of the graphs G^M and G^C, denoted by S^M and S^C, respectively, are defined as:

$$S_{ij}^M = \begin{cases} 1 & if\ (x_i, x_j) \in M\ or\ (x_j, x_i) \in M \\ 0 & otherwise \end{cases} \tag{13}$$

$$S_{ij}^C = \begin{cases} 1 & if\ (x_i, x_j) \in C\ or\ (x_j, x_i) \in C \\ 0 & otherwise \end{cases} \tag{14}$$

Then, the Laplacian matrices are computed as $L^M = D^M - S^M$ and $L^C = D^C - S^C$, so that $D_{ii}^M = \sum_j S_{ij}^M$ and $D_{ii}^C = \sum_j S_{ij}^C$. According to the Eqs. (13) and (14), the following equations are obtained:

$$\sum_{(\mathbf{x}_i,\mathbf{x}_j)\in M}(f_{ri} - f_{rj})^2 = \sum_{i,j}(f_{ri} - f_{rj})^2 S_{ij}^M = 2\mathbf{f}_r^T L^M \mathbf{f}_r \tag{15}$$

$$\sum_{(\mathbf{x}_i,\mathbf{x}_j)\in C}(f_{ri} - f_{rj})^2 = \sum_{i,j}(f_{ri} - f_{rj})^2 S_{ij}^C = 2\mathbf{f}_r^T L^C \mathbf{f}_r \tag{16}$$

According to the Eqs. (15) and (16), neglecting the constant 2, constraint scores defined in Eqs. (11) and (12) are changed into:

$$C_r^1 = \frac{\mathbf{f}_r^T L^M \mathbf{f}_r}{\mathbf{f}_r^T L^C \mathbf{f}_r} \tag{17}$$

$$C_r^2 = \mathbf{f}_r^T L^M \mathbf{f}_r - \lambda \mathbf{f}_r^T L^C \mathbf{f}_r \tag{18}$$

Once the constraint score is computed for each feature, the features with the lowest scores are selected. Unfortunately, the constraint scores strongly depend on the constraint sets (the must-link constraints (M) and the cannot-link constraints (C)) defined by the user. So, when the constraint sets are modified by the user, the feature scores are also changed [1].

3.1.2 Semi-supervised feature selection methods based on spectral graph theory and cluster assumption

Zhao and Liu presented a semi-supervised feature selection method [51] which uses the spectral graph theory and cluster assumption. The method seeks the cluster which maximizes the separability and consistency with the label information performed by cluster indicator. The ideal case is all labeled data in each cluster coming from the same class. Consider two feature vectors f and f', and their corresponding cluster indicators g and g'. The cluster structures formed by these cluster indicators are shown in Figure 4. In this Figure, the labeled data are indicated with "+" and "-", and unlabeled data are shown with "Δ".

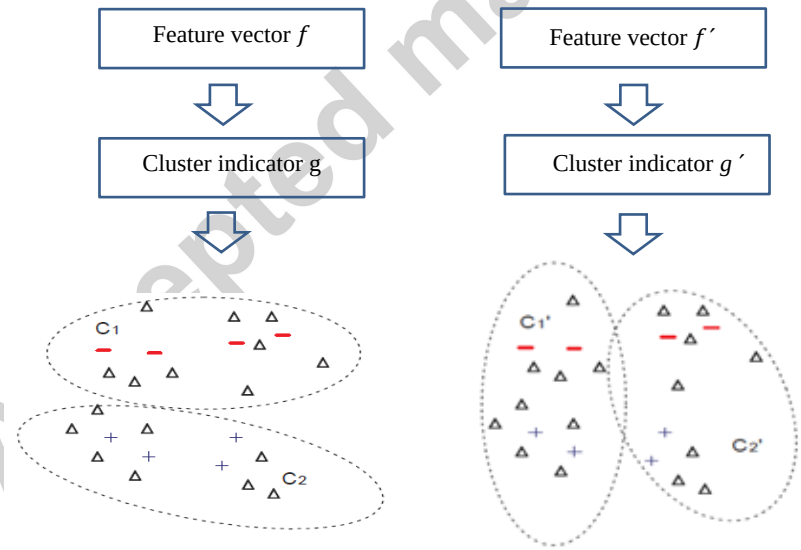


Figure 4: The cluster structures formed by the cluster indicators g and g'[51]

Both cluster indicators g and g' form clearly separable cluster structures. However, the cluster structure formed by g is preferred to that formed by g' due to the fact that all labeled data in a cluster are

of the same class. So, the feature corresponding to the feature vector *f* is more relevant with target concept than the feature corresponding to feature vector *f′*.

This method initially constructs a neighborhood graph with *n* nodes as the graph constructed by Laplacian score in subsection 3.1.1; the weight matrix *S* of the graph, the diagonal matrix *D* and graph Laplacian matrix *L* are computed as those in Laplacian score in subsection 3.1.1. Then, for each feature vector *f*, its corresponding cluster indicator *g* is computed. After computing the cluster indicator for each feature vector, its importance is evaluated by: (1) whether the cluster structures formed by the indicator are well, and (2) whether it is consistent with the label information as shown in Figure 4.

The following equation can be used for evaluation of cluster indicator.

$$\mathbf{Score}(\boldsymbol{F_r}) = \boldsymbol{\lambda}\frac{\sum_{v_i \sim v_j} \left(\boldsymbol{g_i} - \boldsymbol{g_j}\right)^2 \times \boldsymbol{S_{ij}}}{\mathbf{2}\sum_{v_i \in V} \boldsymbol{g_i^2} \times \boldsymbol{D_{ii}}} + (\mathbf{1} - \boldsymbol{\lambda})(\mathbf{1} - \boldsymbol{NMI}(\hat{\boldsymbol{g}}, \boldsymbol{y})) \quad (19)$$

where $\hat{g} = sign(g)$, λ is aregularization parameter¸and $NMI(\hat{g}, y)$ is the normalized mutual information between $\hat{g}$ and y. The features are sorted according to the decreasing order of $Score(F_r)$ to select the most relevant ones.

The time complexity of the construction *S*, *D* and *L* matrices is $O(n^2)$, and the time complexity of calculating the score in Eq. (19) for *d* features is $O(dn^2)$. Last, $O(d \log d)$ operations are needed to rank the features. Hence, the overall time complexity of the method is $O(d \max (n^2, \log d))$.

3.1.3 Semi-supervised feature selection methods based on Laplacian score

Semi-supervised feature selection methods based on Laplacian score [11,66–68] combine the concepts of Laplacian criteria and output information for feature selection. These methods are placed in the category of graph-based methods which construct the neighborhood graph and evaluate the features according to their capability of preserving the local structure of the data. In Laplacian score described in subsection 3.1.1, the structure is defined based on the unsupervised part of the data. In semi-supervised feature selection methods based on Laplacian score, the structure is defined based on the supervised and unsupervised part of the data. According to the literature review, semi-supervised feature selection

methods based on Laplacian score can be divided into two categories: Semi-supervised feature selection methods based on Laplacian score for classification problems [11,66] and semi-supervised feature selection methods based on Laplacian score for regression problems [67,68].

Semi-supervised feature selection methods based on Laplacian score for classification problems

In semi-supervised feature selection methods based on Laplacian score for classification problem, a within-class graph G^w and a between-class graph G^b are constructed [11,66]. The within-class graph combines the information from the labeled and unlabeled part of the data. The graph consists of n nodes and connects the samples which have the same label, or the samples which are sufficiently close to each other. The between-class graph consists of n nodes which connects the samples with different labels. This graph only uses the information from the labeled part of the data. Relevant features must preserve the geometrical structure of these graphs.

The weight matrix S^w can be computed as Eqs. (20) or (21) and the weight matrix S^b is computed as Eq. (22).

$$S_{ij}^{W} = \begin{cases} \gamma & \text{if } x_i \text{ and } x_j \text{ share the same label} \\ 1 & \text{if } x_i \text{ or } x_j \text{ is unlabeled but } x_i \in KNN(x_j) \text{ or } x_j \in KNN(x_i) \\ 0 & otherwise \end{cases} \tag{20}$$

where γ is a suitable constant.

$$S_{ij}^{W} = \begin{cases} e^{-\frac{\|x_i - x_j\|^2}{t}} & \text{if } x_i \text{ and } x_j \text{ share the same label or if } x_i \text{ or } x_j \text{ is} \\ & \text{unlabeled but } x_i \in KNN(x_j) \text{ or } x_j \in KNN(x_i) \\ 0 & otherwise \end{cases} \tag{21}$$

where t is a constant to be tuned.

$$S_{ij}^{b} = \begin{cases} 1 & \text{if } x_i \text{ and } x_j \text{ have different labels} \\ 0 & otherwise \end{cases} \tag{22}$$

The weight matrices S^w and S^b evaluate the similarity between samples in G^w and the dissimilarity between samples in G^b, respectively.

After Construction of the graphs, the graph Laplacians of G^w and G^b are computed as follows:

$$L^w = D^w - S^w, D^w = diag(S^w 1), D^w_{ii} = \sum_j S^w_{ij}, 1 = [1, \dots, 1]^T$$

$$L^b = D^b - S^b, D^b = diag(S^b 1), D^b_{ii} = \sum_j S^b_{ij}, 1 = [1, \dots, 1]^T$$

Let f_r denote the rth feature. The following semi-supervised Laplacian score was defined by Zhao et al. [66] for the rth feature:

$$\boldsymbol{L_r^1 = \frac{\mathbf{f}_r^T \boldsymbol{L}^w \mathbf{f}_r}{\mathbf{f}_r^T \boldsymbol{L}^b \mathbf{f}_r}} \quad (23)$$

Cheng et al. [11] defined the following semi-supervised Laplacian score for the rth feature:

$$\boldsymbol{L_r^2 = \frac{\tilde{\mathbf{f}}_r^T \boldsymbol{L}^W \tilde{\mathbf{f}}_r}{\tilde{\mathbf{f}}_r^T \boldsymbol{D}^b \tilde{\mathbf{f}}_r}} \quad (24)$$

where $\tilde{f}_r$ is defined as Eq.(6).

The calculated scores by Eq. (23) or Eq. (24) are sorted in ascending order and the features with minimal scores are selected.

The construction of within-class graph G^w and between-class graph G^b requires $O(n^2)$ operations and the evaluation of d features based on the graphs requires $O(dn^2)$ operations. The ranking of the features in ascending order requires $O(d \log d)$ operations. Thus, the overall time complexity of the method is $O(d \max(n^2, \log d))$.

Semi-supervised feature selection methods based on Laplacian score for regression problems

Doquire and Verleysen [67,68] proposed a semi-supervised feature selection method based on Laplacian score for regression problems. In this method, the distance between two known samples is computed from their outputs, and between unknown samples is computed from the unsupervised part of them. Doquire and Verleysen, initially introduced a supervised feature selection method, which then was extended to achieve semi-supervised feature selection. In regression problems, an output vector $Y =$

$[y_1, \dots, y_l] \in R^l$ is considered. In this problems, it is expected that close samples x_i and x_j to have close output values y_i and y_j. In this case, good features are expected to have close values for data samples whose outputs are close too. The weight matrix S^{sup} of the graph is defined as:

$$S_{i,j}^{sup} = \begin{cases} e^{-(y_i - y_j)^2/t} & if\ x_i\ and\ x_j\ are\ close, \\ o & otherwise \end{cases} \quad (25)$$

where t is a constant to be tuned. For the construction of S^{sup}, two data samples x_i and x_j are considered as close if one of the corresponding outputs (y_i or y_j) is among the k nearest neighbors of the other. After construction of S^{sup}, the following equations are computed:

$$D^{sup} = diag(S^{sup}\mathbf{1}), \mathbf{1} = [\mathbf{1}, \dots, \mathbf{1}]^T, L^{sup} = D^{sup} - S^{sup},$$

$$\tilde{\mathbf{f}}_r = \mathbf{f}_r - \left(\mathbf{f}_r^{\mathrm{T}} D^{sup}\mathbf{1}/\mathbf{1}^{\mathrm{T}} D^{sup}\mathbf{1}\right)\mathbf{1}$$

Supervised Laplacian score (*SLS*) for regression problems can be defined as:

$$SSL_r = \frac{\tilde{\mathbf{f}}_r^T L^{sup} \tilde{\mathbf{f}}_r}{\tilde{\mathbf{f}}_r^T D^{sup} \tilde{\mathbf{f}}_r} \quad (26)$$

Semi-supervised Laplacian score for regression problems is based on the developments of the concepts in supervised and unsupervised Laplacian scores. For the computation of semi-supervised Laplacian score, the matrix dist of pairwise distances between each pair of data samples is defined as:

$$dist_{i,j}^2 = \begin{cases} (y_i - y_j)^2 & if\ y_i\ and\ y_j\ are\ known, \\ \frac{1}{d}\sum_{r=1}^{d}(f_{ri} - f_{rj})^2 & otherwise \end{cases} \quad (27)$$

Based on *dist*, a matrix S^{semi} is defined as follows:

$$S_{i,j}^{semi} = \begin{cases} e^{-dist_{i,j}^2/t} & if\ x_i\ and\ x_j\ are\ close\ and\ y_i\ and/or\ y_j\ is\ unknown \\ C \times e^{-dist_{i,j}^2/t} & if\ x_i\ and\ x_j\ are\ close\ and\ y_i\ and/or\ y_j\ is\ know \\ 0 & otherwise \end{cases} \quad (28)$$

In the above equation, C is a positive constant which gives more importance to the information of supervised part of the data.

After construction of S^{semi} matrix, the following equations are computed:

$$D^{semi} = diag\left(S^{semi}1\right), 1 = [1, \dots, 1]^T, L^{semi} = D^{semi} - S^{semi},$$

$$\tilde{\mathrm{f}}_r = \mathrm{f}_\mathrm{r} - \left(\mathrm{f}_\mathrm{r}^\mathrm{T} D^{semi} 1/1^\mathrm{T} D^{semi} 1\right)1$$

Eventually, semi-supervised Laplacian score (*SSLS*) for *r*th feature is computed as:

$$\mathbf{SSLS}_r = \frac{\tilde{\mathbf{f}}_r^T \boldsymbol{L}^{semi} \tilde{\mathbf{f}}_r}{\tilde{\mathbf{f}}_r^T \boldsymbol{D}^{semi} \tilde{\mathbf{f}}_r} \times \boldsymbol{SSL}_r \tag{29}$$

where SLS_r is the supervised Laplacian score introduced in Eq. (26) and gives more importance to the supervised part of the data.

The features are sorted according to the ascending order of $SSLS_r$ and the features with the lowest scores are selected. The time complexity of this method is the same as that of semi-supervised feature selection methods based on Laplacian score for classification problems.

3.1.4 Semi-supervised feature selection methods based on pairwise constraints

Semi-supervised feature selection methods based on pairwise constraints [1,40] use both pairwise constraints described in section 3.1.1 and the local properties of the labeled and unlabeled data to evaluate the relevance of features according to their constraint and locality preserving power. In other words, relevant features must preserve the local structure of the data and the pairwise constraints created by users. These methods are placed in the category of graph-based methods which construct two graphs using the supervised and unsupervised part of the data. Semi-supervised constraint scores are less sensitive to constraint changes than the supervised constraint scores, while providing satisfying results. In semi-supervised feature selection methods based on pairwise constraints, a nearest neighbor graph G^{kn} is built using the data set X and the subset M of must-link constraints and a graph G^C using the subset C of cannot link constraints. The weight matrix of graph G^{kn} can be computed as Eqs. (30) or (31) and the weight matrix of graph G^C can be computed as Eq. (14).

$$S_{ij}^{kn} = \begin{cases} e^{-\frac{\|x_i - x_j\|^2}{t}} & \textit{if } x_i \textit{ or } x_j \textit{ are close or } (x_i, x_j) \in M \textbf{ or } (x_j, x_i) \in M \\ 0 & \textit{otherwise} \end{cases} \quad (30)$$

where *t* is a constant to be tuned.

$$S_{ij}^{kn} = \begin{cases} \gamma & \textbf{if } (x_i, x_j) \in M \textbf{ or } (x_j, x_i) \in M \\ 1 & \textbf{if } x_i \textit{ or } x_j \textit{ is unlabeled but } x_i \in KNN(x_j) \textit{ or } x_j \in KNN(x_i) \\ 0 & \textit{otherwise} \end{cases} \quad (31)$$

where γ is a suitable constant.

Benabdeslem and Hindawi [40] defined the following semi-supervised constraint score for the *r*th feature:

$$C_r^3 = \frac{\mathbf{f}_r^T L^{kn} \mathbf{f}_r}{\mathbf{f}_r^T L^C D^{kn} \mathbf{f}_r} \quad (32)$$

where f_r denotes the *r*th feature, $L^{kn}=D^{kn}-S^{kn}$, $D_{ii}^{kn} = \sum_j S_{ij}^{kn}$, $L^C=D^C-S^C$, and $D_{ii}^C = \sum_j S_{ij}^C$.

Zhao et al. [66] introduced the semi-supervised constraint score for the *r*th feature as follows:

$$C_r^4 = \frac{\mathbf{f}_r^T L^{kn} \mathbf{f}_r}{\mathbf{f}_r^T L^C \mathbf{f}_r} \quad (33)$$

Semi-supervised constraint score can be defined as follows which is less sensitive to the constraints chosen by the user [1].

$$C_r^5 = \frac{\tilde{\mathbf{f}}_r^T L \tilde{\mathbf{f}}_r}{\tilde{\mathbf{f}}_r^T D \tilde{\mathbf{f}}_r} \cdot \frac{\mathbf{f}_r^T L^M \mathbf{f}_r}{\mathbf{f}_r^T L^C \mathbf{f}_r} \quad (34)$$

The score C_r^5 is the product between the Laplacian score L_r (see Eq. (5)) and the constraint score C_r^1 (see Eq. (17)) defined by Zhang:

$$C_r^5 = L_r . C_r^1 \quad (35)$$

After calculating the semi-supervised constraint score by each of the mention methods, the score is sorted in ascending order and the features with minimal scores are chosen.

The analyzing of time complexity for the semi-supervised constraint score is as follows: First, the number of pairwise constraints used in constraint score is assumed as l, which is bounded by $0<l<O(n^2)$. The construction of graph matrices requires $O(n^2)$ operations. Calculating constraint score for d features needs $O(dn^2)$ operations and ranking d features needs $O(d \log d)$ operations. The overall time complexity of the semi-supervised constraint score is $O(d \max (n^2, \log d))$.

3.1.5 Semi-supervised feature selection methods based on Fisher criterion

Semi-Supervised feature selection methods based on Fisher criterion [69–72] use the properties of Fisher criterion and the local structure and distribution information of both labeled and unlabeled data to select the features with the best discriminant and locality preserving ability. The aim is to maximize the separability between different classes using the labeled data and preserve the local structure of the data using the unlabeled ones. In these methods, Fisher score as Eq. (7) or Fisher criterion as Eq. (8) can be used for computing the semi-supervised Fisher score. In the methods which use Fisher score, a new term based on manifold assumption is added to the objective function of Fisher score to consider the local structure of data [70,71]. In the methods which use Fisher criterion, a new term based on manifold assumption can be added to the objective function of Fisher criterion [72] or the within-class scatter matrix S^w and the between-class scatter matrix S^b can be defined using the labeled and unlabeled data [69]. For this purpose, the within-class graph can be defined using the weight matrix of Eq. (20) or (21) and the between-class graph can be defined using the weight matrix of Eq. (22). Then, graph Laplacians and scatter matrices can be defined using the defined weight matrices. Label propagation methods can also be used to determine the scatter matrices. These methods are used to determine the label of unlabeled data and define the scatter matrices using the determined labels [73,74].

Yang et al. [70] proposed a semi-supervised feature selection method based on Fisher score which combines local structure preserving criterion and variance strategy into the framework of Fisher score. This method uses the local structure information and global distribution information of labeled and unlabeled data. They proposed the following semi-supervised Fisher score which selects the features with the highest scores:

$$SF_r = \frac{\sum_{i=1}^{c} n_i(\mu_r^i - \mu_r)^2 + \delta * V_r}{\sum_{i=1}^{c} n_i(\sigma_r^i)^2 + \lambda * J(\mathrm{f_r})} \tag{36}$$

where c is the number of classes, n_i denotes the number of data samples in class i, μ_r denotes the mean of all data samples on the rth feature, μ_r^i and $(\sigma_r^i)^2$ are the mean and variance of class i corresponding to the rth feature, respectively. δ ($\delta \geq 0$) and λ ($\lambda \geq 0$) are two controlled parameters, V_r is the variance score of the rth feature and $J(\mathrm{f}_r)$ is defined as follows:

$$J(\mathbf{f_r}) = \sum_{i,j} (f_{ri} - f_{rj})^2 S_{ij} = 2\mathbf{f}_r^T(D - S)\mathbf{f}_r = 2\mathbf{f}_r^T L \mathbf{f}_r \tag{37}$$

In the above equation, S denotes the weight matrix can be computed as Eq. (3) or Eq. (4), D is a diagonal matrix defined as $D_{ii} = \sum_j S_{ij}$, f_r is the rth feature and L is the graph Laplacian obtained using the labeled and unlabeled data.

The time complexity of calculating semi-supervised Fisher score by Eq. (36) for d features is O(d max (c, n^2, log d)). In the semi-supervised Fisher score, O(dn^2), O(dn) and O(dc) operations need for calculating $J(\mathrm{f}_r)$, V_r and Fisher score, respectively. The ranking of the features in descending order needs O(d log d) operations.

3.1.6 Semi-supervised feature selection methods based on sparse models

During recent years, sparse feature selection [48,76–78,85,86] has developed rapidly. The aim of sparse feature selection is to use a variety of sparse models to select the most sparse and discriminative features. The most well-known sparse model is l_1-norm (lasso), but sparse feature selection methods based on l_1-norm model can't select features sufficiently sparse sometimes. Much work [87–90] has extended the l_1-norm model to the l_p-norm model ($0 < p < 1$) to obtain more sparse representation of data. Xu et al. [91] have proposed that when p is $\frac{1}{2}$, the l_p-norm, i.e., $l_{1/2}$-norm model has the best performance and sparsity. However, none of the above models consider the useful information of the correlation among different features. Recent researches [49,92,93] have shown that it is beneficial to consider the correlation among different features and select features jointly from all data samples. Some sparse

models such as $l_{2,1}$-norm and $l_{2,p}$-norm consider the correlation among features and select the most relevant ones jointly from the data samples.

In semi-supervised feature selection based on sparse models, the aim is to compute a transformation matrix $W \in R^{d\times c}$ ($c < d$) which optimally preserves discriminative information and data distribution of training data. Each row of W is used to weight each feature, if some rows of W shrink to zero, W can be used for feature selection so that the features associated with the non-zero rows in W are selected [49,78]. To make W suitable for feature selection, a regularization term based on $l_{2,1}$-norm or $l_{2,p}$-norm of W is added to the objective function to ensure that W is sparse in rows which makes the objective function non-smooth. To solve the non-smooth objective function, an efficient iterative algorithm is required. In this paper, semi-supervised feature selection methods based on sparse models are divided into filter and embedded methods. Most studies done in the field of semi-supervised sparse feature selection can be placed in the category of semi-supervised embedded feature selection methods. In the following, a semi-supervised filter feature selection method based on sparse models is described.

Han et al. [49] presented a semi-supervised filter feature selection method based on sparse models which uses a combined semi-supervised scatter matrix to preserve the discriminant information of labeled data and the local structure of both labeled and unlabeled data. Semi-supervised scatter matrix uses the within-class scatter matrix of Fisher discriminant analysis to preserve discriminant information of labeled data and a spline scatter matrix to preserve the local geometry of both labeled and unlabeled data.

Let $W \in R^{d\times c}$ ($c<d$) denote the transformation matrix used for feature selection. The $l_{2,1}$-norm for the matrix W is defined as:

$$\|\boldsymbol{W}\|_{\boldsymbol{2,1}} = \sum_{\boldsymbol{i=1}}^{\boldsymbol{d}} \sqrt{\sum_{\boldsymbol{j=1}}^{\boldsymbol{c}} \boldsymbol{W}_{\boldsymbol{i,j}}^{\boldsymbol{2}}} = \sum_{\boldsymbol{i=1}}^{\boldsymbol{d}} \|\boldsymbol{w}^{\boldsymbol{i}}\|_{\boldsymbol{2}} \tag{38}$$

An objective function for semi-supervised sparse feature selection method can be defined as [49]:

$$\min_{W^TW=1} Tr\left(W^T M W\right) + \lambda \|W\|_{2,1} \quad (39)$$

where the regularization term $\|W\|_{2,1}$ ensures that W is sparse in rows, making it suitable for feature selection, parameter $\lambda > 0$ controls the regularization effect, and $M \in R^{d\times d}$ is a semi-supervised scatter matrix which encodes both label information and local structure of data. The orthogonal constraint $W^TW = I$ is imposed to avoid arbitrary scaling and the trivial solution of all zeros. The semi-supervised scatter matrix M is defined as:

$$M = A + \mu D \quad (40)$$

where matrix $A \in R^{d\times d}$ is a scatter matrix which encodes label information of labeled data, matrix $D \in R^{d\times d}$ is a scatter matrix which encodes the local structural information of labeled and unlabeled data, and parameter μ ($0 \leq \mu \leq 1$) controls the weight of matrix D. The time complexity of computing D, A, and M in Eq. (40) is about $O(n^2)$. An iterative algorithm is used for solving the $l_{2,1}$-norm minimization problem with the orthogonal constraint in Eq. (39) and obtaining optimal W. The time complexity of the iterative algorithm is $O(d^3)$. Once the optimal W is obtained, the features are sorted according to the value of $\|w^i\|_2$ (i = 1,..., d) in descending order and top-ranked features are selected which needs O(d log d) operations.

3.1.7 Other methods

Some semi-supervised filter feature selection methods [75] are not placed in any of the mentioned categories. In general, using unlabeled data, the objective function of many supervised feature selection methods can be turned to the objective function of semi-supervised feature selection methods.

3.2 Semi-supervised wrapper feature selection methods

Semi-supervised wrapper feature selection methods use a single learner or an ensemble learning model to predict the labels of the unlabeled data as well as to evaluate the effectiveness of the chosen feature subset. Semi-supervised wrapper feature selection methods correspond to self-training or co-training based semi-supervised feature selection methods from the second taxonomy. In these methods, after

training the learning model by the labeled data, a subset of unlabeled data along with the predicted labels is selected and added to the labeled data.

Semi-supervised wrapper feature selection methods can be divided into semi-supervised wrapper feature selection methods based on a single learner and semi-supervised wrapper feature selection methods based on ensemble learning.

3.2.1 Semi-supervised wrapper feature selection methods based on a single learner

Semi-supervised wrapper feature selection methods based on a single learner [53] can be placed in the category of self-training based semi-supervised feature selection methods from the second taxonomy. In a semi-supervised wrapper feature selection method based on a single learner, a supervised feature selection method such as forward feature selection and a single learner are used to select a subset of features. Then, the selected features are used to train a classifier; the classifier is used to predict the labels of unlabeled data. Afterward, a subset of the unlabeled data along with the predicted labels is randomly selected and added to the labeled data to form a new training set. The new training set is used for selection of features based on the supervised feature selection method and the learning model. The random selection and feature selection process are repeated several times and the subsets of features are selected. The frequency of every feature is considered in the feature subsets and the feature with the most frequency is added to the selected feature subset to form a new subset of features. This process is repeated until the size of the feature subset reaches a predefined number.

3.2.2 Semi-supervised wrapper feature selection methods based on ensemble learning

Ensemble learning methods train multiple classifiers and then combine their output results [55]. Semi-supervised wrapper feature selection methods based on ensemble learning [52,54,55] are placed in the category of self-training or co-training based semi-supervised feature selection methods from the second taxonomy. These methods [52,54,55] utilize the ability of ensemble learning to identify relevant features in a semi-supervised setting based on self-training or co-training. Semi-supervised feature selection methods based on ensemble learning use a confidence measure to select the predicted unlabeled data. Confidence measure is an important criterion which affects the performance of semi-supervised feature

selection method on ensemble learning. In these methods, there are a number of classifiers which differ on the training sets or feature sets. Different training sets are created by resampling methods such as Bagging and different feature sets are created by random subspace methods (RSM). The combination of resampling with random subspace can also be used for creation of different data set.

In semi-supervised feature selection methods based on ensemble learning, for each classifier h^i, the new data sets $L^i_{bag}$and U^i_{bag} are drawn, from the original labeled data set L and unlabeled data set U, respectively. After creating the different sets for different classifiers, self-training or co-training methods can be used. In one of the self-training semi-supervised feature selection methods based on ensemble learning [52], each classifier h^i is trained by the L^i_{bag}set to predict the label of U^i_{bag} set. Then, the most confident unlabeled data from U^i_{bag} set are added into L^i_{bag}set. The steps are repeated several times.

In another self-training semi-supervised feature selection methods based on ensemble learning [54], each classifier h^i is trained using the L^i_{bag}set. Then, a number of samples are selected from the unlabeled data U to form U' set. For predicting the labels of U' data set, all classifiers are used and the most confident unlabeled data are added into the labeled data set L. These steps are repeated and a new L^i_{bag} set is created for each classifier h^i in each iteration.

In co-training semi-supervised feature selection methods based on ensemble learning [55], each classifier h^i is trained using the L^i_{bag}set. The labels of U^i_{bag} set are predicted using other classifiers and the most confident unlabeled data are added into the L^i_{bag}set. Then, the classifier $h^i$is retrained using L^i_{bag}set. The co-training steps are repeated several times.

After completing the training of ensemble learning model, feature selection can be done using the supervised feature selection methods such as forward feature selection [54], or a permutation-based out-of-bag feature importance measure [52,55].

3.3 Semi-supervised embedded feature selection methods

Semi-supervised embedded feature selection methods perform feature selection in the process of training using the labeled and unlabeled data. In this survey, semi-supervised embedded feature selection methods are divided into semi-supervised embedded feature selection methods based on sparse models and graph Laplacian and semi-supervised embedded feature selection methods based on support vector machines. Semi-supervised embedded feature selection methods based on sparse models and graph Laplacian are placed in the category of graph-based semi-supervised feature selection methods from the second taxonomy. Semi-supervised embedded feature selection methods based on support vector machines correspond to SVM-based semi-supervised feature selection methods from the second taxonomy.

3.3.1 Semi-supervised embedded feature selection methods based on sparse models and graph Laplacian

Semi-supervised embedded feature selection methods based on sparse models and graph Laplacian [76–78] aims to utilize a variety of sparse model for feature selection and use graph-based semi-supervised learning to simultaneously exploit labeled and unlabeled.

Let $W \in R^{d\times c}$ (c<d) denote the transformation matrix used for sparse feature selection. An objective function for semi-supervised embedded sparse feature selection method is defined as:

$$\min_{W} loss(W) + \lambda \|W\|_{2,p}^{p} \tag{41}$$

where $loss(.)$ is the loss function and λ is a regularization parameter. $\|W\|_{2,p}$ is defined as:

$$\|W\|_{2,p} = \left(\sum_{i=1}^{d} \|w^i\|_2^p \right)^{1/p} \qquad p \in (0,1] \tag{42}$$

$l_{2,p}$-norm is reduced to $l_{2,1}$-norm when p = 1.

Denote $X = [x_1, \dots, x_l, x_{l+1}, \dots, x_n]^T$ as the feature matrix of training data, where l is the number of the labeled data and n is the total number of the training data. Given $Y = [y_1, \dots, y_l, y_{l+1}, \dots, y_n]^T \in \{0,1\}^{n\times c}$

be the label matrix of training data, where c is the number of classes and $y_i \in R^c \ (1 \leq i \leq n)$ is the ith label vector. Let Y_{ij} denote the jth sample of y_i, then $Y_{ij} = 1$ if x_i is in the jth class, while $Y_{ij} = 0$ otherwise. If x_i is not labeled, i.e. $i > l$, y_i is set to a vector with all zeros.

In semi-supervised embedded feature selection methods based on sparse models and graph Laplacian, a graph G is constructed with n nodes as the graph constructed by Laplacian score in subsection 3.1.1. Then, the weight matrix S of the graph, the diagonal matrix D and graph Laplacian matrix L are computed as those in Laplacian score in subsection 3.1.1. Manifold Regularization is the most well-known method based on the graph Laplacian which extends many algorithms to semi-supervised way [76,78]. By applying Manifold Regularization to the loss function of Eq. (41), the following objective function is obtained:

$$\underset{\boldsymbol{W},\boldsymbol{b}}{\arg\min}\, \boldsymbol{Tr}\left(\boldsymbol{W}^{\boldsymbol{T}}\boldsymbol{X}\boldsymbol{L}\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{W}\right) + \boldsymbol{\mu}\left\|\boldsymbol{X}_{\boldsymbol{l}}^{\boldsymbol{T}}\boldsymbol{W} + \boldsymbol{1}_{\boldsymbol{n}}\boldsymbol{b}^{\boldsymbol{T}} - \boldsymbol{Y}_{\boldsymbol{l}}\right\|_{\boldsymbol{F}}^{\boldsymbol{2}} + \boldsymbol{\lambda}\|\boldsymbol{W}\|_{\boldsymbol{2,p}}^{\boldsymbol{p}} \tag{43}$$

where $Tr(.)$ denotes the trace operator, X_l and Y_l denote the labeled training data and their ground truth labels, respectively. $b \in R^c$ is the bias term and $1_n \in R^n$ denotes a column vector with all its elements being 1. μ and λ are regularization parameters.

The optimal W obtained from Eq. (43) is affected by the known ground truth labels Y_l. However, inspired by the recent transductive classification algorithm [64,94], in order to exploit both labeled and unlabeled data, a predicted label matrix is defined as $F = [f_1, f_2, \ldots, f_n]^T \in \mathrm{R}^{n \times c}$ for all the training data. $f_i \in R^c \ (1 \leq i \leq n)$ is the predicted label of x_i. According to [95], F should simultaneously satisfy the smoothness on the ground truth labels of the training data and the manifold structure. Hence, F can be defined as follows:

$$\underset{\boldsymbol{F}}{\boldsymbol{arg\,min}}\, \boldsymbol{Tr}\left(\boldsymbol{F}^{\boldsymbol{T}}\boldsymbol{L}\boldsymbol{F}\right) + \boldsymbol{Tr}\left((\boldsymbol{F}-\boldsymbol{Y})^{\boldsymbol{T}}\boldsymbol{U}(\boldsymbol{F}-\boldsymbol{Y})\right) \tag{44}$$

where $U \in R^{n \times n}$ is a diagonal matrix named as a decision rule matrix, whose diagonal elements $Uii=\infty$ if x_i is labeled data and $Uii=1$ otherwise. This decision rule matrix U makes the predicted labels F consistent with the ground truth labels Y.

By incorporating Eq. (44) into Eq. (43) and considering all the training data with their labels (note that X and F are used instead of X_l and Y_l, respectively), the following objective function is obtained:

$$\arg\min_{\boldsymbol{F},\boldsymbol{W},\boldsymbol{b}} \boldsymbol{Tr}\left(\boldsymbol{F}^{\boldsymbol{T}}\boldsymbol{L}\boldsymbol{F}\right) + \boldsymbol{Tr}\left((\boldsymbol{F}-\boldsymbol{Y})^{\boldsymbol{T}}\boldsymbol{U}(\boldsymbol{F}-\boldsymbol{Y})\right) + \boldsymbol{\mu}\left\|\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{W} + \boldsymbol{1}_{\boldsymbol{n}}\boldsymbol{b}^{\boldsymbol{T}} - \boldsymbol{F}\right\|_{\boldsymbol{F}}^{\boldsymbol{2}} + \boldsymbol{\lambda}\|\boldsymbol{W}\|_{\boldsymbol{2},\boldsymbol{p}}^{\boldsymbol{p}} \tag{45}$$

From Eq. (45), F, W and b can be solved at the same time. The objective function in Eq. (45) involves the $l_{2,p}$-norm which makes it non-smooth. It can be converted to the following quadratic problem.

$$\arg\min_{W} Tr(W^T AW) + 2Tr(B^T W) + \lambda\|W\|_{2,p}^{p} \tag{46}$$

A and B are defined as follows:

$$A = XH(\mu I - \mu^2 P)HX^T \tag{47}$$

$$B = \mu XHPUY \tag{48}$$

where I is an identity matrix, $H = I - (1)/(n)1_n 1_n^T$ and $P = (L + U + \mu H)^{-1}$.

Given W = [w^1, ⋯,w^d]^T, then $\|W\|_{2,p}^{p} = \frac{2}{p}Tr(W^T DW)$. Then the objective in Eq. (48) is equivalent to:

$$\arg\min_{W} Tr(W^T AW) + 2Tr(B^T W) + \lambda\frac{2}{p}Tr(W^T DW) \tag{49}$$

An efficient iterative algorithm is required to solve the objective problem in Eq. (49). After obtaining the optimal W, it can be used for feature selection or computation of label prediction matrix. If we want

to use the optimal W for feature selection, the features are sorted according to the value of $\|w^i\|_2$ (i = 1,..., d) in descending order to select the top-ranked features.

The time complexity of the semi-supervised embedded feature selection methods based on sparse models and graph Laplacian is as follows: Computing the graph Laplacian matrix L and the decision rule matrix U requires $O(n^2)$ operations. Learning the optimized transformation matrix W using the iterative algorithm contains calculating the inverse of a few matrices, among which the most complex part is $O(n^3)$. If we want to use the optimal W for computation of label prediction matrix at the test stage, It needs to perform $c \times d \times n_{test}$ times multiplications to predict the label of the testing data, where n_{test} is the number of testing data. If the optimal W is used for feature selection, the time complexity for ranking the features in descending order is $O(d \log d)$.

3.3.2 Semi-supervised embedded feature selection methods based on support vector machines

Semi-supervised embedded feature selection methods based on support vector machines select features through maximizing the classification margin between different classes and simultaneously exploiting the local structure of both labeled and unlabeled data. Different methods such as manifold regularization, recursive feature elimination, combining l_1-norm and l_2-norm and replacement of l_2-norm with l_1-norm can be used for SVM-based semi-supervised feature selection. Therefore, In this survey, semi-supervised feature selection methods based on support vector machines are divided into SVM-based semi-supervised feature selection based on manifold regularization [79], SVM-based semi-supervised feature selection based on feature recursive elimination (RFE) [80] and SVM-based semi-supervised feature selection based on sparse models [15,81].

SVM-based semi-supervised feature selection based on manifold regularization

Xu et al. [79] proposed an SVM-based semi-supervised feature selection based on manifold regularization. In the method, an optimal subset of features is identified by maximizing a performance measure that combines classification margin with manifold regularization. The manifold regularization in the feature selection method guarantees that the decision function is smooth on the manifold built by the selected features of the unlabeled data. The overall complexity of this method is $O(n^{2.5}/\varepsilon^2)$.

SVM-based semi-supervised feature selection based on feature recursive elimination

SVM-RFE [80] is an SVM-based supervised feature selection method which ranks the features using backward feature elimination. In this approach, the features are sorted in decreasing order of their weights, and the lowest ranked features are removed. Support vector machine is recursively trained using remaining features until all the features are ranked.

Semi-supervised SVM-based RFE (S^3VM-RFE) [80] is an improvement of the supervised SVM-RFE, which uses both labeled and unlabeled data in training process. In this approach, supervised SVM is trained with labeled data and unsupervised one-class SVM algorithm is trained with unlabeled data. The feature weight vector created using supervised SVM (*wl*) and unsupervised SVM (*wu*) are summed up as the final ranking criterion (*W*). The feature with the smallest ranking criterion is eliminated. The training process continues using remaining features until all the features are ranked.

SVM-based semi-supervised feature selection based on sparse models

Dai et al. [81] proposed an SVM-based semi-supervised feature selection based on the elastic net penalty (SENFS) which combines l_1-norm and l_2-norm. In this method, highly correlated features can be selected or removed together. SENFS uses the l_1-norm penalty to select the features, and the l_2-norm penalty to help groups of highly correlated features get selected or removed together.

Yang and Wang [15] proposed a feature selection method based on semi-supervised support vector machine (S^3VM) which uses l_1-norm. This method uses the weigh vector *W* for feature selection. If a component of *W* is zero, the corresponding feature of *W* is removed from the problem. The l_1-norm makes the most components of *W* to be zero, which results in reducing feature space dimension.

4. Comparison of semi-supervised feature selection methods

In the previous section, semi-supervised feature selection methods are categorized from two different perspectives. In this section, some characteristics, advantages and disadvantages of semi-supervised feature selection methods are described.

Table 1 presents the category of different semi-supervised feature selection methods described previously from two mentioned perspectives.

Table1: The category of different semi-supervised feature selection methods from two different perspectives

Number	Methods	based on the taxonomy of semi-supervised learning methods										based on the basic taxonomy of feature selection methods											
		Filter methods						Wrapper methods		Embedded methods		Based on graph					Self-training		Co-training	support vector machines			Other methods
		Graph theory & cluster assumption	Laplacian score	Pairwise constraints	Fisher criterion	Sparse models	Other methods	Based on a single learner	Based on ensemble learning	Sparse models and graph Laplacian	Support vector machines	Graph theory & cluster assumption	Laplacian score	Pairwise constraints	Fisher criterion	Sparse models	Based on ensemble learning	Based on a single learner	Based on ensemble learning	Manifold regularization	Feature recursive elimination	Sparse models	Other methods
1	Zhao & Liu(2007) [51]	✓										✓											
2	Yanh & Wang(2007) [15]										✓											✓	
3	Zhao et al.(2008) [66]		✓										✓										
4	Li(2008) [75]						✓																
5	Ren et al.(2008) [53]							✓										✓					
6	Yang et al.(2010) [70]				✓										✓								
7	Chen et al.(2010) [69]				✓										✓								
8	Liu et al.(2010) [74]				✓										✓								
9	Xu et al.(2010) [79]										✓									✓			
10	Benabdeslem & Hindawi (2011) [40]			✓										✓									
11	Cheng et al.(2011) [11]		✓										✓										
12	Doquire & Verleysen (2011) [67]		✓										✓										
13	Kalakech et al.(2011) [1]			✓										✓									
14	Ren et al.(2011) [96]			✓										✓									
15	Yang et al.(2011) [72]				✓										✓								
16	Han et al.(2011) [54]								✓								✓						
17	Barkia et al.(2011) [55]								✓										✓				
18	Ma et al.(2011) [78]									✓						✓							
19	Ma et al.(2012) [76]									✓						✓							
20	Bellal et al.(2012) [52]								✓								✓						
21	Doquire & Verleysen (2013) [68]		✓										✓										
22	LV et al.(2013) [71]				✓										✓								
23	Liu et al.(2013) [73]				✓										✓								
24	Dai et al.(2013) [81]										✓											✓	
25	Shi et al.(2014) [77]									✓						✓							
26	Song et al.(2014) [48]									✓						✓							
27	Han et al.(2015) [49]					✓																	✓

28	Ang et al.(2015) [80]										✓									✓		

As shown in Table 1, from the perspective of the first taxonomy, most of the semi-supervised feature selection methods presented in the literature are filter-based methods and few semi-supervised feature selection methods are wrapper-based methods. From the perspective of the second taxonomy, most of the semi-supervised feature selection methods presented in the literature have been focused on graph-based methods and few semi-supervised feature selection methods have been provided based on self-training and co-training.

The general advantage and disadvantage of semi-supervised feature selection methods from two mentioned perspectives are summarized in Table 2 and Table 3.

Table 2: General advantage and disadvantage of semi-supervised feature selection methods based on the basic taxonomy of feature selection methods

Method	Advantage	Disadvantage
Filter	▪ Computationally much more efficient than wrapper and embedded ▪ Independent of the classifier ▪ Easy and fast to implement ▪ Better generalization ability than wrapper and embedded ▪ Easily scale up to very high-dimensional data	▪ Usually evaluate the features one by one and ignore the correlation among features ▪ Ignore interaction with the classifier
Wrapper	▪ Interact with classifier ▪ Usually evaluate the features jointly and consider the dependency among them	▪ Computationally more expensive than filter and embedded ▪ Require building many models which is very time-consuming ▪ More susceptible to overfitting ▪ Low generalization ability ▪ Classifier-dependent selection
Embedded	▪ Interact with classifier ▪ Evaluate the features jointly and consider the dependency among them ▪ Less susceptible to over fitting than wrapper ▪ Better computational complexity than wrapper	▪ Classifier-dependent selection

Table 3: General advantage and disadvantage of semi-supervised feature selection methods based on the taxonomy of semi-supervised learning methods

Method	Advantage	Disadvantage
Graph-based	▪ Computationally much more efficient than the self-training and co-training based methods ▪ Better generalization ability than the self-training and co-training based methods	▪ Usually evaluate the features one by one ignore the dependency among features ▪ Usually ignore interaction with the classifier
Self-training and co-training based	▪ Interact with classifier ▪ Usually evaluate the features jointly and consider the dependency among them	▪ Need expensive computation ▪ Require building many models which is very time-consuming ▪ More susceptible to overfitting ▪ Low generalization ability ▪ Classifier-dependent selection
SVM-based	▪ Interact with classifier ▪ Evaluate the features jointly and consider the dependency among them ▪ Less susceptible to over fitting than the self-training and co-training based methods ▪ Better computational complexity than the self-training and co-training based methods	▪ Classifier-dependent selection

In addition to the general advantage and disadvantage, each of the semi-supervised feature selection methods mentioned in previous section has its own advantage and disadvantage described in Table 4.

Table 4: Characteristics, advantage and disadvantage of semi-supervised feature selection methods

Method	Characteristics	Advantage	Disadvantage
Graph theory & cluster assumption [51]	▪ Construct a neighborhood graph ▪ Transform each feature vector into a cluster indicator ▪ The cluster indicators are evaluated by separability and consistency ▪ Couple the Laplacian score with normalized mutual information	▪ Have only general advantage of filter methods and graph-based methods	▪ Ignore the correlation among features and evaluate the features one by one
Laplacian score [11,66]	▪ Construct a within-class graph and a between-class graph ▪ Evaluate the features through their degree of preserving the graph structures	▪ Have only general advantage of filter methods and graph-based methods	▪ Ignore the correlation among features and evaluate the features one by one

Constraint score [1,40]	▪ Use a few supervision information in the form of pairwise constraints ▪ Construct two graphs using the pairwise constraints and unlabeled data ▪ Evaluate the features through locality and constraint preserving ability ▪ The number of used pairwise constraints are usually much lower than the number of all possible pairwise constraints	▪ Require only a few supervision information in the form of pairwise constraints ▪ The pairwise constraints are obtained easier by the user than the class labels ▪ With a few pairwise constraints, achieve good performance compared to the semi-supervised scores which use full class labels on the whole training data	▪ Ignore the correlation among features ▪ Evaluate the features one by one ▪ Depend on the subsets of pairwise constraints created by the user ▪ The generated constraints subsets may not have the same importance ▪ Some constraints can be redundant, incoherent or incomplete with respect to the data set ▪ Seeking an appropriate constraints subset is time-consuming
Fisher criterion [69,71,73,74]	▪ Use the Fisher criterion and the local structure of both labeled and unlabeled data ▪ Evaluate the features through the discriminant and locality preserving ability	▪ Have only general advantage of filter methods and graph-based methods	▪ Ignore the correlation among features and evaluate the features one by one
Sparse-based filter methods [49]	▪ Combine two supervised and unsupervised scatter matrices ▪ Preserve the discriminant information of labeled data and the local geometry structure of labeled and unlabeled data ▪ An $l_{2,1}$-norm is added to objective function makes it suitable for feature selection ▪ Use an iterative algorithm to solve the objective function	▪ Evaluate the features jointly and consider correlation among the features	▪ The objective function is non-smooth and difficult to solve ▪ Need an efficient iterative algorithm to solve the objective function
Based on a single learner[53]	▪ Extend the initial labeled training set by using the predicted unlabeled data ▪ Use the mechanism of random selection on unlabeled data to form new training sets ▪ Add the most frequently selected feature to the feature subset during each iteration	▪ Interact with classifier	▪ Do not consider a confidence measure for unlabeled data ▪ Mislabeled data may be added and degraded the prediction ability ▪ Ignore the discriminative power of a combination of features ▪ Require a large amount of computational time
Based on ensemble learning [52,54,55]	▪ Use ensemble learning with self-training or co-training to predict the labels of unlabeled data ▪ Exploit the power of ensemble learning to identify and remove the irrelevant features in a semi-supervised setting	▪ Interact with classifier ▪ Consider the dependency among features ▪ Use a confidence measure to select the predicted unlabeled data ▪ Improve the generalization ability of a single learner by an ensemble classifier ▪ Select a relevant feature subset faster than the methods based on a single learner	▪ Mislabeled data can be added by an inaccurate confidence measure which degraded the classification performance ▪ Require a large amount of computational time

Sparse-based Embedded methods [48,76–78]	▪ Learn the classifiers directly when performing feature selection ▪ Construct a graph using both labeled and unlabeled data ▪ Select the features using both labeled and unlabeled data while simultaneously considering the correlation between them ▪ An $l_{2,p}$-norm is added to objective function makes it suitable for feature selection	▪ Combine the strengths of joint feature selection and semi-supervised learning	▪ The objective function is non-smooth and hard to solve ▪ Require an efficient iterative algorithm to obtain the optimal solution
SVM-based methods [15,79,81]	▪ Select the features through maximizing the margin between different classes and at the same time exploiting the local structure of both labeled and unlabeled data	▪ Have only general advantage of embedded methods and SVM-based methods	▪ The objective function is difficult to solve ▪ Require an efficient optimization method to solve the objective function

4.1 Performance evaluation

To evaluate the performance of some semi-supervised feature selection methods mentioned in pervious section, three UCI data sets were used and several experiments were conducted using K-nearest neighbor (KNN) and SVM classifiers. In the experiments, the original data sets were split into training and testing sets, and 25% of training data in each data set were used as labeled training data and the remaining were considered as unlabeled training data. The labeled and unlabeled training data were used for semi-supervised feature selection, the labeled training data was used for model construction, and the testing set was used for model evaluation. The details of the used data sets in our experiments are summarized in Table 5.

Table5: The details of the used data sets

Data set	Number of features	Number of samples	Number of training data	Number of testing data	Classes
WDBC	30	569	376	193	2
WBCD	9	683	410	273	2
Diabetes	8	786	500	286	2

In the experiments, several parameters should be set and tuned. The number of nearest neighbors (k) used to compute the graph Laplacian was set to 5 and the regularization parameter λ in Eqs. (39) and (41) was set to 1. For semi-supervised feature selection based on Laplacian score, the score in Eq.(23) was

used and the weight matrix S^w of the within-class graph G^w was computed using Eq. (20). For semi-supervised feature selection based on pairwise constraints, the scores in Eqs.(33) and (34) were used which were called constraint score-1 and constraint score-2, respectively. The weight matrix S^{kn} of graph G^{kn} in semi-supervised feature selection methods based on pairwise constraints was computed by Eq. (31). In semi-supervised constraint score, a total of 10 pairwise constraints, including 5 must-link constraints and 5 cannot-link constraints, were used which is very small compared to the total number of possible constraints. During the experiments, the pairwise constraints subsets were generated L times as follows: The pairs of samples were randomly selected from the labeled training data and must-link or cannot-link constraints were generated depending on whether the underlying classes of the two samples were the same or different. Because of the random generation of pairwise constraints, 20 different subsets including 10 subset of must-link constraints and 10 subsets of cannot-link constraints were generated and 10-fold cross validation was done on the labeled training set to obtain the best subsets of must-link and cannot-link constraints.

The optimal number of features in each feature selection method was found using 10-fold cross-validation on the labeled training set by maximizing the classification accuracy. Also, the optimal value of parameter k in k-nearest neighbor (KNN) classifier was obtained by 10-fold cross-validation on labeled training data. Once the optimal number of features was determined, the experiments were performed on the testing set using the selected feature subset.

The experimental results of different semi-supervised feature selection methods using KNN and SVM classifiers on three data sets are shown in Tables 6-8

Table 6: Classification performance of different semi-supervised feature selection methods on WDBC data set

Classifier	Feature selection Method	Type (from perspective-1)	Type (from perspective-2)	Accuracy	Features Number
KNN	Laplacian score [66]	Filter	Graph-based	0.9578	12
	Constraint score-1 [66]	Filter	Graph-based	0.9430	8
	Constraint score-2 [1]	Filter	Graph-based	0.9534	11
	Fisher score [70]	Filter	Graph-based	0.9637	7
	Fisher criterion[73,74]	Filter	Graph-based	0.9741	5
	Sparse-based filter method [49]	Filter	Other methods	0.9482	20
	Based on a single learner [53]	Wrapper	Self-training based	0.9637	7
	Based on sparse models and graph Laplacian [76,78]	Embedded	Graph-based	0.9585	6

	Based on sparse models and graph Laplacian [77]	Embedded	Graph-based	0.9326	9
SVM	Laplacian score [66]	Filter	Graph-based	0.9067	18
	Constraint score-1 [66]	Filter	Graph-based	0.8860	5
	Constraint score-2 [1]	Filter	Graph-based	0.9430	9
	Fisher score [70]	Filter	Graph-based	0.9119	12
	Fisher criterion[73,74]	Filter	Graph-based	0.9119	13
	Sparse-based filter method [49]	Filter	Other methods	0.9482	11
	Based on a single learner [53]	Wrapper	Self-training based	0.9119	5
	Based on sparse models and graph Laplacian [76,78]	Embedded	Graph-based	0.9016	9
	Based on sparse models and graph Laplacian [77]	Embedded	Graph-based	0.8860	11

As can be seen in Table 6, semi-supervised filter feature selection based on Fisher criterion and sparse models have the best performance on WDBC data set using KNN and SVM classifiers, respectively.

Table 7: Classification performance of different semi-supervised feature selection methods on WBCD data set

Classifier	Feature selection Method	Type (from perspective-1)	Type (from perspective-2)	Accuracy	Features Number
KNN	Laplacian score [66]	Filter	Graph-based	0.9817	6
	Constraint score-1 [66]	Filter	Graph-based	0.9853	5
	Constraint score-2 [1]	Filter	Graph-based	0.9780	6
	Fisher score [70]	Filter	Graph-based	0.9780	6
	Fisher criterion[73,74]	Filter	Graph-based	0.9817	5
	Sparse-based filter method [49]	Filter	Other methods	0.9817	7
	Based on a single learner [53]	Wrapper	Self-training based	0.9780	4
	Based on sparse models and graph Laplacian [76,78]	Embedded	Graph-based	0.9707	5
	Based on sparse models and graph Laplacian [77]	Embedded	Graph-based	0.9817	6
SVM	Laplacian score [66]	Filter	Graph-based	0.9780	3
	Constraint score-1 [66]	Filter	Graph-based	0.9707	4
	Constraint score-2 [1]	Filter	Graph-based	0.9853	3
	Fisher score [70]	Filter	Graph-based	0.9817	3
	Fisher criterion[73,74]	Filter	Graph-based	0.9634	5
	Sparse-based filter method [49]	Filter	Other methods	0.9670	7
	Based on a single learner [53]	Wrapper	Self-training based	0.9634	5
	Based on sparse models and graph Laplacian [76,78]	Embedded	Graph-based	0.9780	4
	Based on sparse models and graph Laplacian [77]	Embedded	Graph-based	0.9780	3

As shown in Table 7, the best results on WBCD data set have been obtained by semi-supervised constraint score-1 and constraint score-2 using KNN and SVM classifiers, respectively.

Table 8: Classification performance of different semi-supervised feature selection methods on Diabetes data set

Classifier	Feature selection Method	Type (from perspective-1)	Type (from perspective-2)	Accuracy	Features Number
KNN	Laplacian score [66]	Filter	Graph-based	0.7090	4
	Constraint score-1 [66]	Filter	Graph-based	0.7425	3
	Constraint score-2 [1]	Filter	Graph-based	0.7425	5
	Fisher score [70]	Filter	Graph-based	0.7724	4

	Fisher criterion[73,74]	Filter	Graph-based	0.7724	4
	Sparse-based filter method [49]	Filter	Other methods	0.7724	3
	Based on a single learner [53]	Wrapper	Self-training based	0.6530	2
	Based on sparse models and graph Laplacian [76,78]	Embedded	Graph-based	0.7910	4
	Based on sparse models and graph Laplacian [77]	Embedded	Graph-based	0.7910	4
SVM	Laplacian score [66]	Filter	Graph-based	0.8060	3
	Constraint score-1 [66]	Filter	Graph-based	0.7836	4
	Constraint score-2 [1]	Filter	Graph-based	0.7836	6
	Fisher score [70]	Filter	Graph-based	0.8060	3
	Fisher criterion[73,74]	Filter	Graph-based	0.7873	5
	Sparse-based filter method [49]	Filter	Other methods	0.7948	5
	Based on a single learner [53]	Wrapper	Self-training based	0.8060	5
	Based on sparse models and graph Laplacian [76,78]	Embedded	Graph-based	0.7948	6
	Based on sparse models and graph Laplacian [77]	Embedded	Graph-based	0.7910	5

It is clear from Table 8 that using KNN classifier, semi-supervised feature selection methods based on sparse models and graph Laplacian have better performance than other semi-supervised feature selection methods. Semi-supervised Laplacian score and Fisher score have the best performance on diabetes data set using SVM classifier. Semi-supervised feature selection based on a single learner has the same performance as Laplacian score and Fisher score but it uses more features than these scores. Semi-supervised feature selection based on a single learner has a bad performance on diabetes data set using KNN classifier. This is due to that in the semi-supervised feature selection based on a single learner, the confidence of predicted labels cannot be ensured. When these unconfident data are used in semi-supervised learning, the performance of the classifier may be degraded.

According to the results of Tables 6-8, the performance of SVM and KNN classifiers and semi-supervised feature selection methods depends on the data, and we cannot select the best semi-supervised feature selection method and classifier regardless of the data. The problem is discussed in details in Discussion subsection.

4.2 Discussion

Semi-supervised wrapper feature selection methods from the first taxonomy and semi-supervised feature selection methods based on self-training and co-training from the second taxonomy use the predicted unlabeled data to extend the initial labeled training set. If a classifier is trained with mislabeled

data, its prediction ability will be degraded. Hence, a confidence measure should be used to verify whether the unlabeled data are correctly predicted.

Semi-supervised feature selection based on a single learner uses the mechanism of random selection on unlabeled data and does not consider the confidence about the labeling of unlabeled data. This may lead to adding mislabeled data to the labeled training set which causes performance degradation. This method evaluates the features according to their frequency and ignores the discriminative ability of a combination of features which is the primary advantage of wrapper methods. Semi-supervised feature selection methods based on ensemble learning use a confidence measure for selecting unlabeled data and achieve better performance than the methods based on a single learner. An important criterion that affects the performance of semi-supervised feature selection methods based on ensemble learning is the use of a confidence measure for the selection of predicted unlabeled data. An accurate confidence measure leads to performance improvement during the semi-supervised learning. However, semi-supervised feature selection methods based on ensemble learning needs expensive computation times and may be intractable for large data sets.

Most of the semi-supervised feature selection methods presented in the literature are filter-based methods from the perspective of the first taxonomy and graph-based methods from the perspective of the second taxonomy. When there are very few labeled data, semi-supervised constraint score in the category of filter and graph-based methods can be an appropriate feature selection method to achieve acceptable performance. The pairwise constraints used in this score are easier obtained by the user than the class labels. Since the performance of constraint score is depend on the subset of pairwise constraints created by the user, we should generate different subsets of pairwise constraints and average the results of constraint score over different runs. We also can seek the suitable constraints subset using cross-validation.

In general, semi-supervised filter methods are computationally efficient and have good generalization ability and performance in some applications. Because of the computational efficiency of filter methods, these methods are usually adopted in dealing with data with a large number of features. Most of the filter

methods ignore the dependency among features. When there are several highly correlated features, embedded methods and filter methods based on sparse models which are computationally more efficient than wrapper can be used to obtain good performance. These methods consider the correlations among the features but require an efficient iterative algorithm to solve the non-smooth objective function.

According to the experimental results using KNN and SVM classifiers on a few data sets and literature review, filter methods from the perspective of the first taxonomy and graph-based methods from the perspective of the second taxonomy seem to be good approaches for semi-supervised feature selection in many applications.

5. Conclusion and future work

Semi-supervised feature selection has received considerable attention in the last decade. In this paper, a survey on semi-supervised feature selection methods has been given. In addition, two taxonomies of semi-supervised feature selection methods have been presented from two different perspectives. The first is based on the basic taxonomy of feature selection methods and the latter is based on the taxonomy of semi-supervised learning methods.

According to the first taxonomy, semi-supervised feature selection methods can be categorized into filter, wrapper and embedded methods. Most of the methods presented in the literature have been focused on the filter methods and few methods have been provided based on wrapper methods. Semi-supervised filter methods are fast and computationally efficient, and easily scale up to very high dimensional data sets. Based on the study of literature and experimental results, it seems that these methods are good approaches for feature selection in many applications.

According to the second taxonomy, semi-supervised feature selection methods can be categorized into five types: graph-based semi-supervised feature selection, self-training based semi-supervised feature selection, co-training based semi-supervised feature selection, SVM-based semi-supervised feature selection, and other semi-supervised feature selection methods. From this perspective, most of the

researches done on the semi-supervised feature selection have been focused on the graph-based methods and few semi-supervised feature selection methods have been proposed based on self-training and co-training. Graph-based semi-supervised feature selection methods are computationally efficient and have good generalization ability. These methods seem to be appropriate approaches in selection of features in many applications.

Based on the literature review, it was observed that most of the semi-supervised feature selection methods have been presented for classification problems and few methods have been proposed for regression problem. Semi-supervised feature selection for regression problems is a research direction that must be further explored and new methods must be developed for it. Semi-supervised feature selection based on sparse models for regression problems is a new field which uses both labeled and unlabeled data for selection of features while simultaneously considering the correlation between them. In the future research, we intend to present semi-supervised filter and embedded feature selection methods based on the sparse models for regression problems. For this purpose, the label information of labeled data and the local structure of both labeled and unlabeled data will be used to select the most relevant features jointly for regression problems.

References

[1] M. Kalakech, P. Biela, L. Macaire, D. Hamad, Constraint scores for semi-supervised feature selection: A comparative study, Pattern Recognit. Lett. 32 (2011) 656–665. doi:10.1016/j.patrec.2010.12.014.

[2] M. Zhao, Z. Zhang, T.W.S. Chow, Trace ratio criterion based generalized discriminative learning for semi-supervised dimensionality reduction, Pattern Recognit. 45 (2012) 1482–1499. doi:http://dx.doi.org/10.1016/j.patcog.2011.10.008.

[3] M. Hindawi, K. Allab, K. Benabdeslem, Constraint Selection-Based Semi-supervised Feature Selection., in: ICDM, IEEE, 2011: pp. 1080–1085.

[4] K.-Q. Shen, C.-J. Ong, X.-P. Li, E.P. V. Wilder-Smith, Feature selection via sensitivity analysis of SVM probabilistic outputs, Mach. Learn. 70 (2008) 1–20. doi:10.1007/s10994-007-5025-7.

[5] K. Benabdeslem, M. Hindawi, Efficient Semi-Supervised Feature Selection: Constraint, Relevance, and Redundancy, IEEE Trans. Knowl. Data Eng. 26 (2014) 1131–1143.

doi:10.1109/TKDE.2013.86.

[6] D. Zhang, S. Chen, Z.-H. Zhou, Constraint score: A new filter method for feature selection with pairwise constraints, Pattern Recognit. 41 (2008) 1440–1451. doi:10.1016/j.patcog.2007.10.009.

[7] M. Reif, F. Shafait, Efficient feature size reduction via predictive forward selection, Pattern Recognit. 47 (2014) 1664–1673. doi:10.1016/j.patcog.2013.10.009.

[8] B. Xue, M. Zhang, S. Member, W.N. Browne, Particle swarm optimization for feature selection in classification: A multi-objective approach, IEEE Trans. Cybern. 43 (2013) 1656–1671.

[9] X. Zhang, G. Wu, Z. Dong, C. Crawford, Embedded feature-selection support vector machine for driving pattern recognition, J. Franklin Inst. 352 (2015) 669–685. doi:10.1016/j.jfranklin.2014.04.021.

[10] J.D. Yang, H. Xu, P.F. Jia, Effective search for genetic-based machine learning systems via estimation of distribution algorithms and embedded feature reduction techniques, Neurocomputing. 113 (2013) 105–121. doi:Doi 10.1016/J.Neucom.2013.01.014.

[11] H. Cheng, W. Deng, C. Fu, Y. Wang, Z. Qin, Graph-based semi-supervised feature selection with application to automatic spam image identification, in: Comput. Sci. Environ. Eng. EcoInformatics, Springer, 2011: pp. 259–264.

[12] X. Chen, T. Fang, H. Huo, D. Li, Semisupervised Feature Selection for Unbalanced Sample Sets of VHR Images, IEEE Geosci. Remote Sens. Lett. 7 (2010) 781–785. doi:10.1109/LGRS.2010.2048197.

[13] Y. Sun, G. Wen, Emotion recognition using semi-supervised feature selection with speaker normalization, Int. J. Speech Technol. (2015) 1–15. doi:10.1007/s10772-015-9272-x.

[14] C.-H. Chen, A semi-supervised feature selection method using a non-parametric technique with pairwise instance constraints, J. Inf. Sci. 39 (2013) 359–371. doi:10.1177/0165551512456502.

[15] L. Yang, L. Wang, Simultaneous feature selection and classification via semi-supervised models, Nat. Comput. 2007. ICNC 2007. Third Int. Conf. 1 (2007) 646–650. doi:10.1109/ICNC.2007.666.

[16] P. Mitra, C. a. Murthy, S.K. Pal, Unsupervised feature selection using feature similarity, IEEE Trans. Pattern Anal. Mach. Intell. 24 (2002) 301–312. doi:10.1109/34.990133.

[17] S. Maldonado, R. Weber, J. Basak, Simultaneous feature selection and classification using kernel-penalized support vector machines, Inf. Sci. (Ny). 181 (2011) 115–128. doi:10.1016/j.ins.2010.08.047.

[18] H. Uğuz, A two-stage feature selection method for text categorization by using information gain, principal component analysis and genetic algorithm, Knowledge-Based Syst. 24 (2011) 1024–

1032. doi:10.1016/j.knosys.2011.04.014.

[19] M. Hall, G. Holmes, Benchmarking attribute selection techniques for discrete class data mining, IEEE Trans. Knowl. Data Eng. 15 (2003) 1437–1447.

[20] A. Unler, A. Murat, R.B. Chinnam, Mr2PSO: A maximum relevance minimum redundancy feature selection method based on swarm intelligence for support vector machine classification, Inf. Sci. (Ny). 181 (2011) 4625–4641. doi:10.1016/j.ins.2010.05.037.

[21] C.-H. Chen, A hybrid intelligent model of analyzing clinical breast cancer data using clustering techniques with feature selection, Appl. Soft Comput. 20 (2014) 4–14.

[22] J. Pohjalainen, O. Räsänen, S. Kadioglu, Feature selection methods and their combinations in high-dimensional classification of speaker likability, intelligibility and personality traits, Comput. Speech Lang. 29 (2015) 145–171.

[23] Z. Zhao, G. Fu, S. Liu, K.M. Elokely, R.J. Doerksen, Y. Chen, et al., Drug activity prediction using multiple-instance learning via joint instance and feature selection, BMC Bioinformatics. 14 Suppl 1 (2013) S16. doi:10.1186/1471-2105-14-S14-S16.

[24] B. Xue, M. Zhang, W.N. Browne, Particle swarm optimisation for feature selection in classification: Novel initialisation and updating mechanisms, Appl. Soft Comput. 18 (2014) 261–276. doi:10.1016/j.asoc.2013.09.018.

[25] Y. Peng, Z. Wu, J. Jiang, A novel feature selection approach for biomedical data classification, J. Biomed. Inform. 43 (2010) 15–23. doi:10.1016/j.jbi.2009.07.008.

[26] T. Nowotny, A.Z. Berna, R. Binions, S. Trowell, Optimal feature selection for classifying a large set of chemicals using metal oxide sensors, Sensors Actuators, B Chem. 187 (2013) 471–480. doi:10.1016/j.snb.2013.01.088.

[27] A. Unler, A. Murat, A discrete particle swarm optimization method for feature selection in binary classification problems, Eur. J. Oper. Res. 206 (2010) 528–539. doi:10.1016/j.ejor.2010.02.032.

[28] E. Rashedi, H. Nezamabadi-Pour, S. Saryazdi, A simultaneous feature adaptation and feature selection method for content-based image retrieval systems, Knowledge-Based Syst. 39 (2013) 85–94. doi:10.1016/j.knosys.2012.10.011.

[29] H.-L. Chen, B. Yang, J. Liu, D.-Y. Liu, A support vector machine classifier with rough set-based feature selection for breast cancer diagnosis, Expert Syst. Appl. 38 (2011) 9014–9022. doi:10.1016/j.eswa.2011.01.120.

[30] J. Kersten, Simultaneous feature selection and Gaussian mixture model estimation for supervised classification problems, Pattern Recognit. 47 (2014) 2582–2595.

[31] B. Peralta, A. Soto, Embedded local feature selection within mixture of experts, Inf. Sci. (Ny). 269 (2014) 176–187. doi:10.1016/j.ins.2014.01.008.

[32] S. Wang, D. Li, X. Song, Y. Wei, H. Li, A feature selection method based on improved fisher's discriminant ratio for text sentiment classification, Expert Syst. Appl. 38 (2011) 8696–8702. doi:10.1016/j.eswa.2011.01.077.

[33] M.F. Akay, Support vector machines combined with feature selection for breast cancer diagnosis, Expert Syst. Appl. 36 (2009) 3240–3247. doi:10.1016/j.eswa.2008.01.009.

[34] S.M.H. Bamakan, P. Gholami, A novel feature selection method based on an integrated data envelopment analysis and entropy model, Procedia Comput. Sci. 31 (2014) 632–638. doi:10.1016/j.procs.2014.05.310.

[35] S. Nakariyakul, Suboptimal branch and bound algorithms for feature subset selection: A comparative study, Pattern Recognit. Lett. 45 (2014) 62–70. doi:10.1016/j.ins.2014.03.072.

[36] J. Yang, Y. Liu, Z. Liu, X. Zhu, X. Zhang, A new feature selection algorithm based on binomial hypothesis testing for spam filtering, Knowledge-Based Syst. 24 (2011) 904–914. doi:10.1016/j.knosys.2011.04.006.

[37] G.-Z. Li, H.-H. Meng, W.-C. Lu, J.Y. Yang, M. Yang, Asymmetric bagging and feature selection for activities prediction of drug molecules, BMC Bioinformatics. 9 (2008). doi:10.1186/1471-2105-9-S6-S7.

[38] P. Shi, S. Ray, Q. Zhu, M.A. Kon, Top scoring pairs for feature selection in machine learning and applications to cancer outcome prediction, BMC Bioinformatics. 12 (2011) 375. doi:10.1186/1471-2105-12-375.

[39] W. Zhou, J. a Dickerson, A novel class dependent feature selection method for cancer biomarker discovery., Comput. Biol. Med. 47 (2014) 66–75. doi:10.1016/j.compbiomed.2014.01.014.

[40] K. Benabdeslem, M. Hindawi, Constrained laplacian score for semi-supervised feature selection, in: Mach. Learn. Knowl. Discov. Databases, Springer, 2011: pp. 204–218.

[41] R. Sheikhpour, M.A. Sarram, R. Sheikhpour, Particle swarm optimization for bandwidth determination and feature selection of kernel density estimation based classifiers in diagnosis of breast cancer, Appl. Soft Comput. 40 (2016) 113–131. doi:10.1016/j.asoc.2015.10.005.

[42] A. Chin, A. Mirzal, H. Haron, H. Hamed, Supervised, Unsupervised and Semi-supervised Feature selection: A Review on Gene Selection, IEEE/ACM Trans. Comput. Biol. Bioinforma. PP (2015) 1–1. doi:10.1109/TCBB.2015.2478454.

[43] V. Bolón-Canedo, N. Sánchez-Maroño, A. Alonso-Betanzos, J.M. Benítez, F. Herrera, A review

of microarray datasets and applied feature selection methods, Inf. Sci. (Ny). 282 (2014) 111–135. doi:10.1016/j.ins.2014.05.042.

[44] G. Chandrashekar, F. Sahin, A survey on feature selection methods, Comput. Electr. Eng. 40 (2014) 16–28. doi:10.1016/j.compeleceng.2013.11.024.

[45] Y. Saeys, I. Inza, P. Larrañaga, A review of feature selection techniques in bioinformatics, Bioinformatics. 23 (2007) 2507–2517. doi:10.1093/bioinformatics/btm344.

[46] I. Guyon, a Elisseeff, An introduction to variable and feature selection, J. Mach. Learn. Res. 3 (2003) 1157–1182. doi:10.1162/153244303322753616.

[47] G.-Z. Yang, X.-P. Hu, Feature selection, U.S. Pat. Appl. 12/064,993 (2006).

[48] X. Song, J. Zhang, Y. Han, J. Jiang, Semi-supervised feature selection via hierarchical regression for web image classification, Multimed. Syst. (2014). doi:10.1007/s00530-014-0390-0.

[49] Y. Han, Y. Yang, Y. Yan, Z. Ma, N. Sebe, S. Member, Semisupervised feature selection via spline regression for video semantic recognition, IEEE Trans. NEURAL NETWORKS Learn. Syst. 26 (2015) 252–264.

[50] X. Chang, Y. Yang, Semi-supervised feature analysis by mining correlations among multipe tasks, (2014) 11. http://arxiv.org/abs/1411.6232.

[51] Z. Zhao, H. Liu, Semi-supervised feature selection via spectral snalysis., in: Proc. 7th SIAM Int. Conf. Data Mining, SIAM, 2007: pp. 641–646.

[52] F. Bellal, H. Elghazel, A. Aussem, A semi-supervised feature ranking method with ensemble learning, Pattern Recognit. Lett. 33 (2012) 1426–1433. doi:10.1016/j.patrec.2012.03.001.

[53] J. Ren, Z. Qiu, W. Fan, H. Cheng, P.S. Yu, S.Y. Philip, Forward semi-supervised feature selection, in: Adv. Knowl. Discov. Data Min., Springer, 2008: pp. 970–976.

[54] Y. Han, K. Park, Y.K. Lee, Confident wrapper-type semi-supervised feature selection using an ensemble classifier, 2011 2nd Int. Conf. Artif. Intell. Manag. Sci. Electron. Commer. AIMSEC 2011 - Proc. (2011) 4581–4586. doi:10.1109/AIMSEC.2011.6010202.

[55] H. Barkia, H. Elghazel, A. Aussem, Semi-supervised feature importance evaluation with ensemble learning, Data Min. (ICDM), 2011 IEEE 11th Int. Conf. (2011) 31–40. doi:10.1109/icdm.2011.129.

[56] L. Zuo, L. Li, C. Chen, The graph based semi-supervised algorithm with ℓ1-regularizer, Neurocomputing. 149 (2015) 966–974. doi:10.1016/j.neucom.2014.07.037.

[57] K. Zhang, L. Lan, J.T. Kwok, S. Vucetic, B. Parvin, Scaling Up Graph-Based Semisupervised Learning via Prototype Vector Machines., IEEE Trans. Neural Networks Learn. Syst. 26 (2015)

444–57. doi:10.1109/TNNLS.2014.2315526.

[58] N.N. Pise, P. Kulkarni, A survey of semi-supervised learning methods, in: Comput. Intell. Secur. 2008. CIS'08. Int. Conf., IEEE, 2008: pp. 30–34. doi:10.1109/CIS.2008.204.

[59] O. Chapelle, B. Schölkopf, A. Zien, Semi-supervised learning, MIT press Cambridge, 2006.

[60] M.A.Z. Chahooki, N.M. Charkari, Unsupervised manifold learning based on multiple feature spaces, Mach. Vis. Appl. 25 (2014) 1053–1065. doi:10.1007/s00138-014-0604-7.

[61] M.A.Z. Chahooki, N.M. Charkari, Improvement of supervised shape retrieval by learning the manifold space, in: Mach. Vis. Image Process. (MVIP), 2011 7th Iran., IEEE, 2011: pp. 1–4.

[62] A. Halder, S. Ghosh, A. Ghosh, Aggregation pheromone metaphor for semi-supervised classification, Pattern Recognit. 46 (2013) 2239–2248. doi:10.1016/j.patcog.2013.01.002.

[63] X. Zhu, A.B. Goldberg, Introduction to semi-supervised learning, 2009. doi:10.2200/S00196ED1V01Y200906AIM006.

[64] X. Zhu, Semi-Supervised Learning Literature Survey, 2008. doi:10.1.1.146.2352.

[65] V.J. Prakash, L.M. Nithya, A Survey On Semi-Supervised Learning Techniques, Int. J. Comput. Trends Technol. 8 (2014) 25–29.

[66] J. Zhao, K. Lu, X. He, Locality sensitive semi-supervised feature selection, Neurocomputing. 71 (2008) 1842–1849. doi:10.1016/j.neucom.2007.06.014.

[67] G. Doquire, M. Verleysen, Graph laplacian for semi-supervised feature selection in regression problems, Lect. Notes Comput. Sci. (Including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics). (2011) 248–255. doi:10.1007/978-3-642-21501-8_31.

[68] G. Doquire, M. Verleysen, A graph laplacian based approach to semi-supervised feature selection for regression problems, Neurocomputing. 121 (2013) 5–13. doi:10.1016/j.neucom.2012.10.028.

[69] L.C.L. Chen, R.H.R. Huang, W.H.W. Huang, Graph-based semi-supervised weighted band selection for classification of hyperspectral data, Audio Lang. Image Process. (ICALIP), 2010 Int. Conf. (2010) 1123–1126. doi:10.1109/ICALIP.2010.5685086.

[70] M. Yang, Y. Chen, G. Ji, Semi_fisher score : a semi-supervised method for feature selection, in: Int. Conf. Mach. Learn. Cybern., 2010: pp. 527–532.

[71] S. Lv, H. Jiang, L. Zhao, D. Wang, M. Fan, Manifold based fisher method for semi-supervised feature selection, in: 2013 10th Int. Conf. Fuzzy Syst. Knowl. Discov., 2013: pp. 664–668.

[72] W. Yang, C. Hou, Y. Wu, A semi-supervised method for feature selection, 2011 Int. Conf. Comput. Inf. Sci. (2011) 329–332. doi:10.1109/ICCIS.2011.54.

[73] Y. Liu, F. Nie, J. Wu, L. Chen, Efficient semi-supervised feature selection with noise insensitive trace ratio criterion, Neurocomputing. 105 (2013) 12–18. doi:10.1016/j.neucom.2012.05.031.

[74] Y. Liu, F. Nie, J. Wu, L. Chen, Semi-supervised feature selection based on label propagation and subset selection, in: Comput. Inf. Appl. (ICCIA), 2010 Int. Conf., IEEE, 2010: pp. 293–296.

[75] J. Li, Semi-supervised feature selection under logistic I-RELIEF framework, in: 2008 19th Int. Conf. Pattern Recognit., 2008: pp. 1–4. doi:10.1109/ICPR.2008.4761687.

[76] Z. Ma, F. Nie, Y. Yang, J.R.R. Uijlings, N. Sebe, S. Member, et al., Discriminating joint feature analysis for multimedia data understanding, IEEE Trans. Multimed. 14 (2012) 1662–1672.

[77] C. Shi, Q. Ruan, G. An, Sparse feature selection based on graph Laplacian for web image annotation, Image Vis. Comput. 32 (2014) 189–201. doi:10.1016/j.imavis.2013.12.013.

[78] Z. Ma, Y. Yang, F. Nie, J. Uijlings, N. Sebe, Exploiting the entire feature space with sparsity for automatic image annotation, Proc. 19th ACM Int. Conf. Multimed. - MM '11. (2011) 283. doi:10.1145/2072298.2072336.

[79] Z. Xu, I. King, M.R.T. Lyu, R. Jin, Discriminative semi-supervised feature selection via manifold regularization, IEEE Trans. Neural Networks. 21 (2010) 1033–1047. doi:10.1109/TNN.2010.2047114.

[80] J.C. Ang, H.H. B, H. Nuzly, A. Hamed, H. Haron, H.N.A. Hamed, Semi-supervised SVM-based feature felection for cancer classification using microarray gene expression data, Curr. Approaches Appl. Artif. Intell. (2015) 468–477. doi:10.1007/978-3-319-19066-2.

[81] K. Dai, H.-Y. Yu, Q. Li, A semisupervised feature selection with support vector machine, J. Appl. Math. 2013 (2013).

[82] C.M. Bishop, Neural networks for pattern recognition, Oxford university press, 1995.

[83] X. He, D. Cai, P. Niyogi, Laplacian Score for Feature Selection, Adv. Neural Inf. Process. Syst. 18. (2005) 507–514. doi:http://books.nips.cc/papers/files/nips18/NIPS2005_0149.pdf.

[84] Q. Gu, Z. Li, J. Han, Generalized Fisher Score for Feature Selection, CoRR. abs/1202.3 (2012).

[85] Z. Zeng, X. Wang, J. Zhang, Q. Wu, Semi-supervised feature selection based on local discriminative information, Neurocomputing. (2015). doi:10.1016/j.neucom.2015.05.119.

[86] X. Chang, F. Nie, Y. Yang, H. Huang, A Convex formulation for semi-supervised multi-label feature selection, Twenty-Eighth AAAI Conf. Artif. Intell. (2014) 1171–1177.

[87] S. Foucart, M.-J. Lai, Sparsest solutions of underdetermined linear systems via ℓq-minimization for $0< q<1$, Appl. Comput. Harmon. Anal. 26 (2009) 395–407.

[88] D. Krishnan, R. Fergus, Fast image deconvolution using hyper-Laplacian priors, in: Adv. Neural Inf. Process. Syst., 2009: pp. 1033–1041.

[89] R. Chartrand, Exact reconstruction of sparse signals via nonconvex minimization, Signal Process. Lett. IEEE. 14 (2007) 707–710.

[90] R. Chartrand, Fast algorithms for nonconvex compressive sensing: MRI reconstruction from very few data, in: Biomed. Imaging From Nano to Macro, 2009. ISBI'09. IEEE Int. Symp., IEEE, 2009: pp. 262–265.

[91] X. Zongben, C. Xiangyu, X. Fengmin, Z. Hai, l1/2 regularization: a thresholding representation theory and a fast solver, IEEE Trans. Neural Networks Learn. Syst. 23 (2012) 1013–1027.

[92] F. Nie, H. Huang, X. Cai, C.H. Ding, Efficient and robust feature selection via joint ℓ2, 1-norms minimization, in: Adv. Neural Inf. Process. Syst., 2010: pp. 1813–1821.

[93] Z. Zhao, L. Wang, H. Liu, Efficient spectral feature selection with minimum redundancy., in: AAAI Conf. Artif. Intell, Citeseer, 2010.

[94] X. Zhu, Z. Ghahramani, J. Lafferty, Semi-supervised learning using gaussian fields and harmonic functions, in: ICML, 2003: pp. 912–919.

[95] F. Nie, D. Xu, I.W.-H. Tsang, C. Zhang, Flexible manifold embedding: A framework for semi-supervised and unsupervised dimension reduction, Image Process. IEEE Trans. 19 (2010) 1921–1932.

[96] Y. Ren, G. Zhang, G. Yu, Random subspace based semi-supervised feature selection, in: Proc. 2011 Int. Conf. Mach. Learn. Cybern., 2011: pp. 113–118.

Razieh Sheikhpour currently is a Ph.D candidate at the department of Computer Engineering in Yazd University, Iran. She is member of International Association of Engineers (IAENG). Her research interests include Bioinformatics, Data mining, Machine learning and Semi-supervised learning.

Mehdi Agha Sarram is an associate professor at the department of Computer Engineering in Yazd University, Iran. He received his Ph.D. degree from University of Wales, Cardiff, U.K. in1979. He is Member of Australian Institute of Control and Instrumentation and also Member of Steering Committee on IT standards (ISIRI-ITTC). He has been Casual Lecturer in Australian Universities such as SIBT Macquarie University, University of Western Sydney Macarthur and SWIC University of Western Sydney from 2000- 2003. His research interests include Machine learning, Data mining, Network coding and Wireless sensor networks.

Sajjad Gharaghani received his Ph.D. in Chemoinformatics from Isfahan University of Technology, Iran. Currently, he is an assistant professor at the department of bioinformatics, Institute of Biochemistry and Biophysics (IBB), University of Tehran, Iran. His research interest includes Bioinformatics, Statistical techniqes, Machine Learning, Chemoinformatics, Pharmacoinformatics, Biological networks, Computer-Aided Drug Design (CADD) and QSAR.

Mohammad Ali Zare Chahooki received his BS in computer engineering from Shahid Beheshti University, Tehran, Iran in 2000 and his MS and PhD in software engineering from Tarbiat Modares University, Tehran, Iran in 2004 and 2013, respectively. Currently, he is an assistant professor at the department of Computer Engineering in Yazd University, Yazd, Iran. His research interests include Machine learning, Computer vision, and Software engineering.

Highlights

- A comprehensive survey on semi-supervised feature selection methods is presented.
- Two categories of these methods are presented from two different perspectives.
- The hierarchical structure of semi-supervised feature selection methods is given.
- Advantage and disadvantage of the survey methods are presented.
- Future research directions are presented.