Information Sciences 480 (2019) 144–159

Contents lists available at ScienceDirect

Information Sciences

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

A factor graph model for unsupervised feature selection

Hongjun Wang[a,*], Yinghui Zhang[b], Ji Zhang[a], Tianrui Li[a], Lingxi Peng[c]

[a] School of Information Science and Technology, Southwest Jiaotong University, Chengdu 611756, China
[b] Software Center, Northeastern University, Shenyang 110819, China
[c] School of Mechanical and Electrical Engineering, Guangzhou University, Guangzhou, Guangdong 510006, China

ARTICLE INFO

Article history:
Received 17 July 2018
Revised 17 December 2018
Accepted 20 December 2018
Available online 21 December 2018

Keywords:
Feature selection
Factor graph
Message-passing algorithm
Unsupervised learning

ABSTRACT

In this paper, a factor graph model for unsupervised feature selection (FGUFS) is proposed. FGUFS explicitly measures the similarities between features; these similarities are passed to each other as messages in the graph model. The importance score of each feature is calculated using the message-passing algorithm, and then feature selection is performed based on the final importance scores. Extensive experiments were performed on several datasets, and the results demonstrate that FGUFS outperforms other state-of-art unsupervised feature selection algorithms on several performance measures.

1. Introduction

The feature selection problem has been studied by the statistics and machine learning communities for many years and is one of the most important methods for data preprocessing [10]. The main idea is to select a subset of features by eliminating irrelevant features that have little or no predictive information. This provides an important and frequently employed dimensionality reduction technique for machine learning [9]. In addition, it reduces the number of features, and removes irrelevant or redundant noisy data. The best subset of features can contain the least number of features and contribute to accuracy as much as possible. This is a key method of preprocessing datasets and provides one approach to avoiding the pitfalls of dimensionality. This approach has been employed for many applications, such as images [25], video [15], text [35,36], and gene analysis [7,34]. It is also of interest that wood moisture content prediction is based on feature selection techniques [18].

Feature selection can considerably enhance the interpretability of machine learning models, and models integrated with feature selection often exhibit better generalization [16]. Furthermore, it is a prominent and popular method for finding an appropriate subset of predictive features. Several problems may result from the presence of irrelevant features during the learning process, and one of the motivations of feature selection is to solve these problems:

1. Additional computational costs are always induced by the presence of irrelevant features in linear models, and the computational cost for prediction increases polynomially with the number of features.
2. The presence of irrelevant or redundant features may lead to overfitting and a poorly constructed model.
3. The goal of feature selection is to keep the size of the learning model as small as possible. It is reasonable and important to ignore irrelevant features or those that have little effect.

[*] Corresponding author.
E-mail addresses: wanghongjun@swjtu.edu.cn (H. Wang), JiZhang@my.swjtu.edu.cn (J. Zhang), trli@swjtu.edu.cn (T. Li), scu.peng@gmail.com (L. Peng).

https://doi.org/10.1016/j.ins.2018.12.034

Prominent among the common objectives of feature selection are the aims of improving the prediction performance and providing faster and more effective predictors while enabling a better understanding of the preprocessing of data. In many data analysis tasks, feature selection is commonly regarded as a crucial method for performing data preprocessing, and it is frequently applied for dimensionality reduction. In addition, it is significant that feature selection can make machine learning models comprehensible, and often improves the generalizability of constructed models. Therefore, finding a good subset of features is an important task in many situations.

Feature selection and transformation are typical methods of dimensionality reduction. The restricted Boltzmann machine (RBM) [19] in deep learning is a successful feature transformation method, and there have been at least 1000 papers on improving RBM or applying RBM in connection with feature extraction published in the last three years. Among these, Mocanu et al. [27] provided a topological insight into RBM, and the authors found that RBM is a factor graph, which naturally has a small-world topology. Inspired by the viewpoint of that study, we attempt to modify RBM for feature selection, and so we design a factor graph for feature selection. From the topological viewpoint, the proposed graph is close to RBM, but the meanings of all the variable nodes and factor nodes in the graph are completely different. Moreover, the inference and problem formulation proceed according to different theories. Finally, the corresponding algorithm of our approach is different from RBM. It is worth highlighting the main contributions of this study, as follows:

1. A novel filter-type unsupervised feature selection algorithm is proposed, namely a factor graph model for unsupervised feature selection (FGUFS). Furthermore, an energy function is applied to represent the proposed model.
2. In FGUFS, the maximal information coefficient (MIC) is used to measure the similarities between features, and a message-passing algorithm is developed for the purpose of inferring the factor graph.
3. Extensive experiments were conducted on a variety of datasets, demonstrating that the proposed approach outperforms the state-of-the-art methods in different applications in terms of the most popular performance measures.

The idea of the proposed filter model is to maximize the MIC between the selected feature subset and the whole features set, which means that the feature selected subset can preserve the maximum information of all feature sets. The messages are passed between all features, and finally the importance score of each feature is calculated using the message-passing algorithm. Unlike existing filter-type methods, which compute the feature importance based on their statistical properties, FGUFS explicitly measures the feature similarities to build the factor graph and utilizes them for feature selection. The proposed method is simple and effective.

The remainder of this paper is organized as follows. We describe related work in Section 2. In Section 3, we introduce the feature selection objective function based on feature similarity. In Section 4, we present the factor graph model for feature selection in detail, including the inference method and algorithm description. Experimental results are presented in Section 5. Finally, conclusions and further research topics are given in Section 6.

2. Related work

Existing feature selection algorithms are based on various selection strategies, which can be broadly classified into filter, wrapper, and embedded methods. Filter methods select features based on their intrinsic properties, which are often measured by certain statistical criteria. Brown et al. [2] proposed a unifying framework for feature selection based on information theory, which is a typical filter method for feature selection. The relevance or discriminative powers of the selected features can be determined using predictive labels. ReliefF [31], the Fisher score, correlation-based feature selection [14], and the fast correlation-based filter [43] are among the most representative filter-type feature selection methods. In wrapper methods, feature selection is integrated with a learning algorithm or model. Its performance is directly evaluated by the performance of the learning algorithm or model [49]. In general, wrapper methods can obtain better results than filter methods, although they entail much higher computational costs. Embedded methods select features as a part of the model construction process. Banerjee and Pal [1] designed an interesting unsupervised feature selection scheme that can select features with controlled redundancy and also discard irrelevant features. Wang et al. [37] designed an embedded unsupervised feature selection method, which can embed feature selection into a clustering algorithm.

According to whether the labels or other information are used to train the feature selection model, feature selection methods can also be classified as unsupervised [1,5,17,20,26,28,37,41,44,46], semi-supervised [39,45], and supervised [21,29,32,38,40]. Supervised and semi-supervised feature selection methods make use of discriminative information, which is usually encoded in labels, constraints, or other background information, while unsupervised feature selection involves designing the model or algorithm without labels or background information.

2.1. Unsupervised feature selection

Supervised feature selection has been successfully applied in industries. However, the selection of discriminative features in unsupervised scenarios is a significant and difficult task owing to the lack of information. Cai et al. [3] proposed a multi-cluster feature selection method for unsupervised feature selection. This approach can preserve the multi-cluster structure of the data, but it can solve the sparse eigenproblem or an L_1-regularized least squares problem. A hierarchical graphical model [12] has been proposed that can achieve both global and local feature selection for clustering. This uses a beta-Bernoulli prior and Dirichlet process for mixture models. Gu et al. [11] proposed a locality-preserving feature learning approach. That method finds a subset of features, and then a linear transformation is learned from these features to optimize the locality-preserving criterion. Law et al. [22] designed an expectation-maximization algorithm based on Gaussian mixture-based clustering for feature salience, and they extended Koller and Sahami's mutual-information-based criterion for unsupervised feature selection. Luo et al. [24] designed an adaptive reconstruction graph to characterize the intrinsic local structure, and then employed a multi-cluster structure to impose a rank constraint on the corresponding Laplacian matrix. Here, the optimal reconstruction graph and selective matrix can be learned simultaneously. He et al. [17] proposed a filter method using the Laplacian score for feature selection. It is interesting that this method can be applied in either a supervised or unsupervised fashion. The method is applied in unsupervised learning scenarios in their study. Moreover, He et al. [17] proposed a unified supervised and unsupervised Laplacian score feature selection method, and enabled their joint study under a general framework. Yang et al. Yang et al. [41] incorporated discriminative analysis and $\ell_{2,1}$-norm minimization into a joint framework for unsupervised feature selection, which can preserve the most discriminative feature subset. Nie et al. [28] proposed a feature selection method in which feature selection and local structure learning are achieved simultaneously. Gui et al. [13] explored various structured sparsity-inducing feature selection methods, and conducted a comprehensive study investigating the connections between different methods. Hou et al. [20] designed an unsupervised feature selection method for joint embedding learning and sparse regression. Then, the weights via local linear approximation incorporating $\ell_{2,1}$-norm regularization are used to solve the optimization problem.

2.2. Graphical model for dimensionality reduction

Graphical models have been employed for feature selection, feature transformation, and automatic feature representation learning. A graphical model is used to represent the relationships between labels and features. This is a supervised feature selection model. Law et al. [23] proposed a mixture model for simultaneous feature selection and clustering. A directed graphical model is designed, and this is inferred with conditional probabilities [12] for feature selection. Unsupervised feature selection is considered from the viewpoint of graph-regularized data reconstruction. Graph regularization [48] can be used to preserve the local structure of the original data space, and linear combination is also applied to approximately reconstruct each data point. Sun and Zhou [33] proposed a directed acyclic graph model for unsupervised feature selection, which conforms to all the properties of a Bayesian network.

A factor graph is a popular tool utilized in many situations. For example, RBM [19] is a factor graph, which is used as encoder and decoder for deep learning. A simple neural network can be regarded as two or more factor graphs connected with each other. Affinity propagation (AP) [6] is also a factor graph for clustering, which can be applied for feature selection [44]. Zhao et al. [44] used the maximal information coefficient to obtain the similarity matrix, and applied AP to cluster the features. The representative features of the cluster center are kept for feature selection.

Most of the above studies relate to unsupervised feature selection and the graph model for dimensionality reduction. There are hundreds of algorithms based on all kinds of theories, which have advantages and disadvantages from different perspectives. A few studies have applied graphical models for feature selection or feature transformation. One important difference between our model and existing graphical models is that in this study an undirected graphical model is employed, and an energy function is applied to represent the proposed model. Thus far, no methods using energy functions for feature selection have been proposed.

Furthermore, a higher goal of unsupervised feature selection is that the selected subset of features obtains the best clustering or classification results while having minimum redundancy among selected features. However, there are very few methods to achieve this higher goal, especially when it comes to filter feature selection methods. In general, a filter method involves pursuing the highest accuracy in clustering or classification, but ignores the redundancy among selected features. The characteristics of the factor graph can effectively model feature selection according to the similarities between features. The variable nodes represent the features effectively, and the function nodes indicate whether the corresponding features are selected. The full connection between a variable and functional node indicates that they have fully communicated with each other and have reached the selected subset, preserving the maximum correlation of the feature set and achieving less redundancy at the same time.

3. Feature selection objective function based on feature similarity

In this section, we provide the definition of the MIC [30] for measuring the relationship between features, and we describe the details of the proposed objective function for unsupervised feature selection based on feature similarity. The main notations utilized in the paper are summarized in Table 1.

To measure the similarity between features, the mutual information $I(f_i, f_j)$ between features f_i and f_j is given by

$$\begin{aligned} I(f_i, f_j) &= H(f_i) - H(f_i|f_j) \\ &= \sum_{f_i^r \in f_i} \sum_{f_j^u \in f_j} p\left(f_i^r, f_j^u\right) \log_2 \frac{p(f_i^r, f_j^u)}{p(f_i^r)p(f_j^u)} = E_{p(f_i^r, f_j^u)}\left[\log_2 \frac{p(f_i^r, f_j^u)}{p(f_i^r)p(f_j^u)}\right] \end{aligned}$$

where $p(f_i^r)$ and $p(f_j^u)$ are elements of f. To more accurately determine the relationship between two discrete, two real, or mixed features (one real and one discrete), the MIC [30] is applied to calculate the similarity between the features. Let F_P

Table 1
Main notation.

Symbol	Explanation
f	Random variable of a feature
D	A set of ordered pairs of features
G	A grid constructed from all the feature pairs
w(f$_i$)	The hidden variable in a factor graph
h	The serial number of the selected feature
I	A function that computes mutual information
MIC	Maximal information coefficient [30]
S	A function that computes similarity
M	The number of features
E	The energy function

be a set of ordered pairs of features. Furthermore, let the i- and j-values of F_P be partitioned one-by-one into i and j bins, respectively, and let a pair of partitions define an i-by-j grid G. Given such a grid G, let $F_P|_G$ be the distribution induced by the points in F_P on the cells of G. That is, the distribution on the cells of G obtained by letting the probability mass in each cell be the fraction of points in D falling in that cell. For a fixed F_P, different grids G result in different distributions $F_P|_G$.

For a finite set $F_P \subset \mathbf{R}^2$ and positive integers i, j,

$$I^*(F_P, i, j) = \max I(F_P|_G),$$

where ***max*** is the maximum over all grids G with i columns and j rows, and $I(F_P|_G)$ denotes the mutual information of $F_P|_G$. Then, the characteristic matrix and MIC of F_P can be defined in terms of I^*. The characteristic matrix $M(F_P)$ of a set F_P of two-variable data is an infinite matrix with entries

$$M(F_P)_{(i,j)} = \frac{I^*(F_P, i, j)}{\log \min\{i, j\}}.$$

The MIC of a set F_P of two-variable data with sample size M and a grid size less than $B(M)$ is given by

$$MIC(F_P) = \max_{ij<B(M)} M(F_P)_{(i,j)},$$

where $\omega(1) < B(M) \le o(n^{M^{1-\varepsilon}})$ for some $0 < \varepsilon < 1$. The MIC falls between 0 and 1 and is symmetric, and higher values imply greater relevance between features.

Brown et al. [2] presented a unifying framework for feature selection based on mutual information, which formulates the feature selection task as a conditional likelihood problem. In the proposed algorithm, The MIC is the maximum value of the mutual information matrix and is used to measure the similarity between features. However, all the methods mentioned in [2] use mutual information to measure the relevance between features. The MIC finds the f_i-by-f_j grid with the highest induced mutual information, and the mutual information scores are normalized. Then, the normalized scores form a matrix, and the MIC is the highest score of the matrix. The MIC is calculated according to mutual information. However, it is the highest normalized mutual information, and strengthens the relationship between the two features. It is more capable of reflecting the dependence between two attributes and is used to evaluate the similarity of features, which can better reduce the redundancy among features. However, owing to the higher complexity of the MIC, it requires more time than mutual information to evaluate the similarities among features.

For unsupervised feature selection, the best selected subset should contain the lowest number of features that retain as much of the original information as possible. Given a high dimensional dataset $X = (x_1, \ldots, x_n)$ in the instance space and $F = (f_1, \ldots, f_M)$ in the feature space, where $x_i \in R$ and $f_j \in R$, let K be the number of selected features, and let $\bar{F} = \{\bar{f}_1, \ldots, \bar{f}_K\}$ denote the selected feature subset.

In terms of maintaining the original information (e.g., maximum mutual information), the feature selection objective function can be expressed as follows:

$$\Gamma(\bar{F}) = \arg\max_{\bar{F}} (\sum_{j=1}^{M} \sum_{i=1}^{K} MIC(\bar{f}_i, f_j)) \tag{1}$$

$$s.t \quad \bar{f}_i \neq f_j,$$

where K and M are the numbers of selected and total features, respectively, and $\bar{F}$ is the subset of selected features. Eq. (1) maximizes the MIC between the selected feature subset and whole feature set, which means that the selected feature subset can preserve the maximum information of all feature sets. In other words, to a certain extent maximizing the MIC can remove redundant features.

If an exhaustive search for the objective function in Eq. (1) with M features with N dimensions is employed and K features are selected, then the computation complexity is $O(NM!/K!(M-K)!)$. Because this is fairly complex, in the next section an effective selection approach based on the factor model is presented.

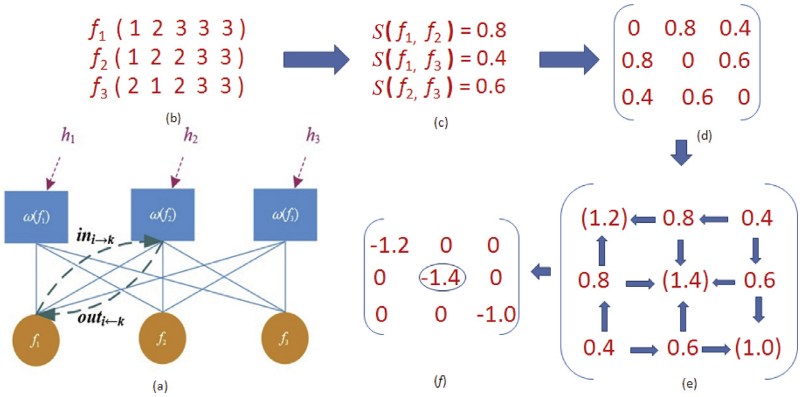

Fig. 1. An illustrative example of a factor graph model for unsupervised feature selection. The constructed factor graph for the example dataset in (b) is shown in (a). By convention, variable nodes in the graph are represented by circles, and factor nodes are represented by squares. The factor nodes are fully connected with variable nodes; whereas, there are no edges between factor nodes or variable nodes. Messages are passed along the edges. (b) shows an example dataset of five instances with three features: $f_1 = (1, 2, 3, 3, 3)^{\dagger}$, $f_2 = (1, 2, 2, 3, 3)^{\dagger}$, $f_3 = (2, 1, 2, 3, 3)^{\dagger}$. Assume that only one feature will be selected. If the MIC [30] is used to calculate $S(f_i, f_j)$, then $S(f_1, f_2) = 0.8, S(f_2, f_3) = 0.6$, and $S(f_1, f_3) = 0.4$. The results are shown in (c). (d) shows the similarity matrix, whose diagonal values are set to 0. (e) shows the process of the message-passing algorithm for the factor graph. There are three situations in this example: (1) $H = (1, 1, 1)$, indicating that the feature f_1 is selected and the other two are dropped, in which case $E(H) = -S(f_2, f_1) - S(f_3, f_1) = -1.2$; (2) $H = (3, 3, 3)$, in which case $E(H) = -S(f_1, f_3) - S(f_2, f_3) = -1.0$; and (3) $H = (2, 2, 2)$, in which case $E(H) = -S(f_1, f_2) - S(f_3, f_2) = -1.4$. In (f), $E(H) = -1.4$ is clearly the lowest value, indicating that f_2 should be the feature selected according to the objective function. This also illustrates how much redundancy is removed among the three features. From another perspective, redundancy is a kind of similarity, and the redundancy of a pair of features can be quantitatively measured by the MIC. If the f_1 feature is selected while the other two are dropped, then a redundancy of 1.2 (the redundancy between f_1 and f_2 is 0.8, and that between f_1 and f_3 is 0.4) is removed between three features. If the feature f_2 is selected while the other two are dropped, then the redundancy of 1.4 is removed between three features. If the feature f_3 is selected while the other two are dropped, then the redundancy of 1.0 is removed between the three features. Finally, the f_2 feature is selected, because the most redundancy is removed in this case.

4. New feature selection method

In this section, we propose a factor graph model for effective unsupervised feature selection, and then we describe how to utilize the message-passing algorithm to perform inferences on the model. There are two reasons that a graph factor model is required to model feature selection. The first is that the factor graph is a popular tool for data preprocessing. The RBM of deep learning is a factor graph for feature extraction, and affinity propagation is another type of factor graph for data clustering, which can be also regarded as data preprocessing. However, there is no factor graph for feature selection. The second reason is that a factor graph can effectively solve the feature selection objective function in Eq. (1). The two variables $\bar{f}_i$ and f_j in Eq. (1) naturally correspond to two kinds of nodes of the factor graph. Furthermore, an edge of the factor graph indicates the relationship between the two variables in Eq. (1). Moreover, the Σ in Eq. (1) is simply modeled as the accumulation of weights.

4.1. Factor graph for unsupervised feature selection

A factor graph is a bipartite graph with two kinds of nodes: factor and variable nodes. There are no edges between variable nodes or factor nodes: Edges only connect variable nodes with factor nodes. The factor graph model is simple but useful, and one designed for feature selection is presented in this section. The idea of FGUFS is that each feature assigns weights to other features as candidates for selected features according to the similarities between them. Then, these weights are passed between all nodes of the factor graph and accumulated. Finally, the features are selected according to the accumulated weights. Fig. 1 illustrates the detailed flow of this idea.

Given high-dimensional data points X or F, the feature selection problem can be modeled using a factor graph $G = (F, \omega, S)$, where f_i $(i = 1, \ldots, M)$ are viewed as variable nodes, and their similarities are edges S_i in the factor graph. Furthermore, $\omega(F)$ are factor nodes, whose weights comprise the feature importance scores. Feature selection on the factor graph is viewed as the problem of searching for the minimum of an energy function with a set of M hidden variables, $H = \{h_1, \ldots, h_M\}$, indicating the selection of the M features. In other words, in the factor graph model each feature f_i will choose a feature to represent itself, and the variable h_i indicates the index of the feature chosen by the feature f_i. We use $f_i^{(h_i)}$ to represent the feature chosen by f_i. In general, a feature will choose its nearest neighbor as its representation. We note that not all configurations of the variables are valid. A configuration H is valid when for a feature f_i, if another feature f_i' chooses f_i as its representation (i.e., $f_i'^{(h_i')} = f_i$), then the feature f_i must have a high similarity with the feature f_i'. The energy of a valid configuration is as follows:

$$E(H) = -\sum_{i=1}^{M} S(f_i, f_i^{(h_i)}) \quad s.t \quad f_i \neq f_i^{(h_i)}, \tag{2}$$

where $S = MIC(f_i, f_j)$ is a function that computes the similarity between features. It is computationally intractable to minimize the energy, because in a special case this is the NP-hard K-median problem [4]. However, the max-sum algorithm in a factor graph can be applied to solve the problem of minimizing a Bethe free energy approximation [6,42].

4.2. Inference on the factor graph

In this section, the inference on the factor graph for feature selection is described in detail. The proposed model is a graph with cycles, and the computation of the marginal probability functions of the model is difficult, because an exponentially large number of terms must be summed. As is well known, the belief propagation (BP) algorithm is always applied to extract the factor graph model without cycles [42], and it can obtain the exact result. To our surprise, we found that this still works well and obtains a good approximate result even when there are cycles in the graph model. Therefore, for the proposed model BP can be utilized for the inference if the cycles in the proposed model are ignored. Moreover, owing to the cycles in FGUFS, the messages are skillfully divided into in- and out-messages for a node, which can solve the message-passing loop problem.

For inference on the factor model for feature selection, we introduce two types of messages passed from variable nodes f_i to their factor nodes $w(f_i)$ and vice versa. A message $in_{\alpha \leftarrow i(f_i)}$ from the feature node i to the factor node α is regarded as the weights of the relative probabilities that the feature node i is in its different states, based on all the information that i has received according to the factor w_α. The message $out_{\alpha \rightarrow i(f_i)}$ from the factor node α to the feature node i is a vector of all the possible states of f_i. This message can be interpreted as the weights from factor node w to feature node i of the relative probabilities that i is in its different states, based on the factor w_α. The messages are updated according to the following rules:

$$in_{\alpha \leftarrow i(f_i)} \equiv \prod_{b \in N(i) \bigcap b \neq \alpha} out_{b \rightarrow i(f_i)} \tag{3}$$

and

$$out_{\alpha \rightarrow i(f_i)} \equiv \sum_{\alpha \neq i} w_\alpha(f_\alpha) \prod_{j \in N(\alpha) \bigcap j \neq i} in_{\alpha \leftarrow j(f_j)}, \tag{4}$$

where $N(\alpha)$ is a set of α neighbors. This is a standard BP algorithm, which can be utilized to infer the proposed feature selection model. It is also a sum-product algorithm with the sum and product. The BP algorithm can be formally defined in terms of belief equations. The beliefs and joint beliefs in feature nodes can be computed through the BP message-update method until they converge (although sometimes they will not).

The factor model of feature selection has cycles, and in our model we apply the Bethe free energy to deal with this situation. According to Yedidia et al. [42], the BP algorithm corresponds to the stationarity conditions for the beliefs of the Bethe free energy, and the BP fixed points converge to the local optima of the Bethe free energy [42]. This fact shows that the BP algorithm can handle the factor graph of feature selection with cycles and that BP is improved upon by the approximation results of the Bethe free energy.

In a simplified form, feature selection can be viewed as searching over valid configurations of the state $H = (h_i, \ldots, h_M)$ to minimize the energy function as follows:

$$E(H) = -\sum_{i=1}^{M} S(f_i, f_i^{(h_i)}), \tag{5}$$

where $f_i^{(h_i)}$ represents the feature selected by f_i. In other words, $E(H)$ is the objective function of feature selection, and the problem can be modeled as follows:

$$H^* = \arg\min_H \left(-\sum_{i=1}^{M} S(f_i, f_i^{(h_i)}\right), \tag{6}$$

where H goes though all possible states and H^* is the optimal one. The maximization of the net similarity of the factor model, which is the negative energy plus a constraint function, makes the configurations available.

$$S(H) = -E(H) + \sum_{k=1}^{M} \omega_k(f_k^{(h)})$$

$$= \sum_{i=1}^{M} S(f_i, f_i^{(h_i)}) + \sum_{k=1}^{M} \omega_k(f_k^{(h)}). \tag{7}$$

It is precisely the min-sum algorithm that is employed for the factor graph of feature selection, and this represents a local message-passing algorithm over the factor graph. Two kinds of messages are passed back and forth between factor and feature nodes during the execution of the min-sum algorithm. If the message-passing algorithm is utilized, then M messages can be reduced to a single message. A message sent from factor nodes $\omega_k(f_k^{(h)})$ to $f_i^{(h_i)}$ consists of M real numbers and can be denoted by $out_{i\leftarrow k}(j)$. At any time, the value of $f_i^{(h_i)}$ can be estimated by summing over all *in* messages and correlation messages.

Because the *in* messages originate from features, they are computed as the element-wise sum of all *in* messages:

$$in_{i\to k}(h_i = k)$$

$$= S(f_i, f_k) - \max_{i,j\neq k}[S(f_i, f_j) + out_{i\leftarrow j}(h_j = j)]. \tag{8}$$

Messages sent from factor nodes to feature nodes are computed by summing over the *in* messages and then maximizing over all feature nodes except the one that the message is being sent to. The message sent from the factor ω_k to feature f_i is as follows:

$$out_{i\leftarrow k}(h_i = k) = \min[0, in_{i\to k}(h_k = k)$$

$$+ \sum_{i'\neq\{i,k\}} \max(0, in_{i'\to k}(h_i = k))], k \neq i, \tag{9}$$

or

$$out_{i\leftarrow k}(h_i = k)$$

$$= \sum_{i'\neq k} \max(0, in_{i'\to k}(h_i = k)), k = i. \tag{10}$$

To estimate the value of a variable $f_i^{(h_i)}$ after any iteration, we sum over all *in* and *out* messages to $f_i^{(h_i)}$ and take the value that minimizes the objective function. To elucidate the proposed model more clearly, we consider a simple example with three features to illustrate how the factor graph model for feature selection works, which is shown in Fig. 1.

In affinity propagation, representative data points are chosen as cluster centers, and the proposed model keeps the representative features as the selected features. We believe that the representative features can preserve considerably more information than other features. Even though the proposed model is inspired by affinity propagation [6], there are two differences between the two:

1. Affinity propagation requires the input of a preference p, whereas the proposed model does not require this parameter: it only requires the input parameter of the similarity matrix.
2. Affinity propagation outputs the exemplar of each data point, whereas the proposed model outputs the rank of each feature, after which we select the top l features according to the requirements.

4.3. Algorithm description

According to the above factor model and inference procedure for feature selection, the proposed algorithm is summarized as follows:

```
FGUFS algorithm:
Input: An M-by-M similarity matrix of features.
Output: The feature ranks.

1. Initialize the in and out messages as $in^0 = 0$ and $out^0 = 0$.
2. Begin the iteration:
   (a) out message function: Compute all the messages from factor nodes $\omega$ to variable nodes $f$ according to Eqs. (9, 10)
       with $in^{(t-1)}$ and $out^{(t-1)}$, where $t$ is the order of iteration, to obtain an $out^t$ matrix.
   (b) in message function: Compute all the messages from variables according to Eq. (8) with $out^t$ and $in^{(t-1)}$ to obtain an
       $in^t$ matrix.
3. End the iteration when Eq. (5) is satisfied or when the in and out messages are stable.
4. Sum over all the messages produced by the out and in functions for each factor node, and order the nodes according to
   the messages.
```

The algorithm alternates between two functions, *out* and *in* message-passing steps, until convergence. Each *out* or *in* message function requires M operations, and each iteration requires $2 \times M$ operations. Thus, the complexity of the algorithm is $O(2TM)$, where M is the number of features and T is the number of iterations. It is well known that a message-passing

Table 2
Sources and numbers of instances, features, and classes in each dataset.

Dataset	Source	Characteristic	Instances	Features	Categories
DriveFace	UCI	real	606	6400	3
secom	UCI	real	1567	590	2
sEMG_sub	UCI	real	400	2500	2
isolet	UCI	real	1560	617	2
apple	Microsoft	image	871	892	3
arcene_valid	NIPS2003	real	100	10000	3
beer	Microsoft	image	870	892	3
beret	Microsoft	image	876	892	3
bible	Microsoft	image	835	892	3
boot	Microsoft	image	845	892	3
brain	Microsoft	image	891	892	3
bugatti	Microsoft	image	882	892	3
ufo	Microsoft	image	881	899	3
video	Microsoft	image	936	899	3
vistawallpaper	Microsoft	image	799	899	3
weddingdress	Microsoft	image	883	899	3

algorithm for factor graphs may not converge in numerous problems. In this study, we use free-energy minimizations to perform inference, to guarantee convergence with a low computational power cost [42]. The proposed algorithm requires considerable memory space on a single computer as the number of features increases, because the input of the algorithm is designed to be an *M*-by-*M* similarity matrix of features. This limitation can be solved using a triple-table representation of the matrix and a corresponding programming implementation, for which the time consumption complexity increases. Therefore, the limitation of our algorithm is that it requires significant memory space when applied to large datasets with a large number (more than 50,000) of features.

5. Empirical study

In this section, we first introduce the datasets, algorithms, and evaluation metrics in the experiment. Then, we discuss the steps performed in the experiment and its results.

5.1. Experimental setup

In the experiment, we compared the performance of our proposed algorithm, FGUFS, with five other widely employed unsupervised feature selection algorithms: LaplacianScore [17], fsSpectrum [46], LquadR21_reg [41], embedded unsupervised feature selection (EUFS) [37], and structured optimal graph feature selection (SOGFS) [28]. LaplacianScore is a filter-type method, which evaluates the importance of a feature based on its power to preserve locality. Furthermore, fsSpectrum determines the feature importance based on a spectrum analysis of feature similarity, LquadR21_reg performs feature selection by incorporating a discriminative analysis and $l_{2,1}$-norm minimization, and SOGFS can perform feature selection and local structure learning simultaneously. In addition, EUFS is an embedded-type unsupervised feature selection, which directly performs feature selection by embedding in a clustering algorithm.

We performed an experimental comparison of the datasets from different applications using different characteristics. These are summarized in Table 2. These datasets were variously obtained from UCI, the NIPS2003 competition, and Microsoft Research Asia Multimedia. A validation strategy can be employed in unsupervised learning to estimate the distributions of parameters in a dataset. This is not required in this experiment, because no assumptions concerning the distributions of datasets are integrated into the proposed algorithm.

To test the quality of the selected features, we performed the following four types of evaluation:

(1) The achieved accuracy by clustering algorithms on the selected features,
(2) The Rand index (RI) achieved by clustering algorithms on the selected features,
(3) The purity achieved by clustering algorithms on the selected features, and
(4) The redundancy contained in the selected features.

The three measures of accuracy, RI, and purity are generally positively correlated, and larger values indicate better clustering results. An ideal feature selection algorithm should select features that result in high accuracy, Rand index, and purity, while containing few redundant features. The performance measures are described in detail below.

1. All of the above datasets come with labels. Viewing these labels as indicative of a reasonable clustering, we used the micro-precision (MP) to evaluate the clustering accuracy. The micro-precision is defined as $MP = \sum_{i=1}^{k} a_i/n$, where k is the number of clusters, n is the number of objects, and a_i denotes the number of objects in cluster i that are correctly assigned to the corresponding class. Note that $0 \le MP \le 1$, with 1 indicating the best possible clustering, which requires full agreement with the class labels.

2. The Rand index was also calculated to evaluate the results of the clustering. RI is usually employed for standard classification problems, and it is a natural extension to use it to compare two clustering results. Thus, RI is defined as $R(C, C') = \frac{2(n_{11}+n_{00})}{n(n-1)}$, where R ranges from 0 to 1. Higher values are better.
3. The purity is a count of the number of data points from the ground truth cluster that each prediction cluster contains. The purity of a clustering result is the sum of the individual cluster purities, or $purity = \sum_{i=1}^{K} \frac{n_i}{n} P(s_i), P(s_i) = \frac{1}{n_i} \max_j (n_i^j)$, where S_i is a particular cluster of size n_i and n_i^j is the number of features from the ith input class assigned to the jth cluster.
4. We used the redundancy rate (RED) [47] to measure the redundancy among the selected features. Assume that F is the set of selected features. Then, the redundancy rate of F is computed as $RED(F) = \sum_{f_i, f_j \in F, i \neq j} I(f_i, f_j)$, where $I(f_i, f_j)$ is the mutual information between the two features f_i and f_j. This measurement enumerates all the mutual information among the feature pairs, and a high value indicates that many selected features are strongly correlated, and thus, that redundancy is expected to exist in F. Lower RED values indicate a better feature selection performance.

5.2. Results

For each dataset, we first applied the six unsupervised learning algorithms FGUFS, EUFS, LaplacianScore, fsSpectrum, LquadR21_reg, and SOGFS to perform feature selection. The number of selected features ranged from three to $M-1$. Then, the K-means algorithm with random initialization was applied to the selected features for clustering. This was repeated five times to obtain an average of the results. The accuracy results were recorded one-by-one. The average accuracies are listed in Table 3, and the detailed accuracies are illustrated in Fig. 2. For example, in Table 3, the accuracy of FGUFS on the dataset apple is given as 0.4623, which is the average from the figure dataset: apple in Fig. 2. The figure dataset: apple contains 598 accuracy values, which were obtained by applying FGUFS on the same dataset, apple, with different numbers of selected features. Fig. 2 shows the detailed results for different numbers of selected features. The x-axis represents the number of selected features, and the y-axis represents the accuracy obtained by the algorithm on the corresponding selected features. We also recorded the clustering accuracies of K-means clustering on the original datasets without feature selection. From Table 3, it is clear that the proposed algorithm, FGUFS, outperforms all the other algorithms, and obtains the best average clustering accuracy results for all datasets. Moreover, we note that FGUFS obtains the 15 best among all results, including the *original* column results. However, Table 3 and Fig. 2 show that FGUFS exhibits a considerably larger variance compared with the baseline approaches. The reason for this is probably that the accuracy is external criteria, but FGUFS is designed according to internal criteria of feature similarities.

The Friedman aligned test [8] is employed to perform a further comparison to show the significance of differences between the algorithms. Table 4 shows the aligned observations with aligned ranks in parentheses for the seven algorithms and 16 data sets. As presented in the table, on average FGUFS ranks first at 10.1875, fsSpectrum ranks second at 45, EUFS ranks third at 62.8750, SOGFS ranks fourth at 63.8125, original ranks fifth at 64.9688, LquadR21_score ranks sixth at 73.25, and LaplacianScore ranks last at 75.4063. The Friedman aligned test can be utilized to check whether the measured sum of aligned ranks is different from the total aligned ranks $\hat{R}_j = 904$ at a high level of significance expected under the null hypothesis:

$$\sum_{j=1}^{n} \hat{R^2_{i,.}} = 346^2 + 332^2 + \cdots + 305^2 + 401^2 = 2,548,602$$

$$\sum_{j=1}^{k} \hat{R^2_{.,j}} = 163^2 + 1021^2 + \cdots + 1006^2 + 1039.5^2 = 6,509,232.5$$

$$T = \frac{(7-1)[6,509,232.5 - (7 \cdot 16^2/4)(7 \cdot 16 + 1)^2]}{7 \times 16(7 \times 16 + 1)(2 \times 7 \times 16 + 1)/6 - 2,548,602/7} = 42.8210$$

For the seven algorithms and 16 data sets, T is distributed according to the chi-squared distribution with $7-1=6$ degrees of freedom. The p-value computed using the $\chi^2(6)$ distribution is 1.27×10^{-7}. Thus, the null hypothesis is significantly rejected. It is clear that the value is significantly lower than 0.05, which shows that the results of the algorithms are significantly different.

For the performance measure RED, Table 5 shows the average redundancies among the selected features, and Fig. 3 illustrates the detailed RED values with different numbers of selected features. Fig. 3 shows that in general, the RED values increase as the number of selected features increases. Among the 16 datasets, FGUFS obtains the best result in 14 cases and the second best result in two. Thus, we can conclude that FGUFS is an effective feature selection algorithm, as it outperforms alternative algorithms in terms of the RED measure on most datasets. We also note that the RED curves in Fig. 3 are generally smooth, whereas the accuracy curves in Fig. 2 exhibit relatively large fluctuations. The reason for this is that RED is an internal criterion, whereas the accuracy is an external criterion. The RED values for FGUFS are considerably lower than those of the baseline approaches. The reason for this is that FGUFS is designed according to the feature similarity, as shown in Eqs. (1) and (2), and RED is calculated by the mutual information, which is a kind offeature similarity.

Table 3
Aggregated clustering accuracies for the selected features obtained by the six feature selection algorithms. These values are computed from the results displayed in Fig. 2. The *original* column shows the clustering accuracies on the original dataset without feature selection.

Dataset	FGUFS	SOGFS	LaplacianScore	fsSpectrum	LquadR21_score	EUFS	Original
DriveFace	**0.6372 ± 0.0011**	0.5045 ± 0.0014	0.5112 ± 0.0007	0.5381 ± 0.0009	0.4924 ± 0.0008	0.5263 ± 0.0013	0.3647 ± 0.0103
sEMG_sub	**0.5149 ± 0.0014**	0.5128 ± 0.0017	0.5010 ± 0.0004	0.5033 ± 0.0008	0.5032 ± 0.0007	0.5020 ± 0.0004	0.5025 ± 0.0001
brain	**0.4441 ± 0.0512**	0.3949 ± 0.0016	0.3716 ± 0.0116	0.4010 ± 0.0082	0.3979 ± 0.0459	0.3797 ± 0.0104	0.3754 ± 0.0122
bugatti	**0.4788 ± 0.0532**	0.4178 ± 0.0009	0.3920 ± 0.0234	0.4321 ± 0.0424	0.3750 ± 0.0257	0.3986 ± 0.0493	0.4086 ± 0.0315
secom	**0.8772 ± 0.1102**	0.7695 ± 0.0010	0.7569 ± 0.0003	0.7804 ± 0.0765	0.7600 ± 0.0208	0.7570 ± 0.0006	0.6439 ± 0.0035
isolet	0.5583 ± 0.0143	0.5141 ± 0.0010	0.5173 ± 0.0095	0.5462 ± 0.0069	0.5342 ± 0.0044	0.5384 ± 0.0261	**0.5654 ± 0.0125**
apple	**0.4623 ± 0.0202**	0.3804 ± 0.0007	0.3826 ± 0.0149	0.3851 ± 0.0367	0.3715 ± 0.0173	0.3883 ± 0.0216	0.3904 ± 0.0224
beer	**0.5001 ± 0.0429**	0.4024 ± 0.0041	0.4372 ± 0.0382	0.4314 ± 0.0439	0.3864 ± 0.0277	0.4523 ± 0.0176	0.4517 ± 0.0211
beret	**0.5695 ± 0.0364**	0.4119 ± 0.0018	0.3834 ± 0.0314	0.4866 ± 0.0757	0.3850 ± 0.0226	0.4131 ± 0.0613	0.3983 ± 0.0326
bible	**0.5172 ± 0.0278**	0.3727 ± 0.0014	0.3841 ± 0.0235	0.4536 ± 0.0525	0.3734 ± 0.0262	0.3833 ± 0.0323	0.3533 ± 0.0419
boot	**0.5073 ± 0.0227**	0.4293 ± 0.0010	0.4494 ± 0.0232	0.4366 ± 0.0188	0.4565 ± 0.0182	0.4673 ± 0.0165	0.4698 ± 0.0124
ufo	**0.4775 ± 0.0344**	0.4020 ± 0.0011	0.4057 ± 0.0118	0.4338 ± 0.0199	0.3994 ± 0.0157	0.4083 ± 0.0211	0.4143 ± 0.0211
video	**0.4126 ± 0.0222**	0.3784 ± 0.0007	0.3521 ± 0.0221	0.3730 ± 0.0247	0.3874 ± 0.0192	0.3743 ± 0.0170	0.3803 ± 0.0205
vistawallpaper	**0.5147 ± 0.0567**	0.4612 ± 0.0026	0.4646 ± 0.0312	0.3850 ± 0.0263	0.4783 ± 0.0418	0.4674 ± 0.0248	0.4731 ± 0.0212
weddingdress	**0.4501 ± 0.0089**	0.4132 ± 0.0008	0.3828 ± 0.0128	0.4178 ± 0.0220	0.3955 ± 0.0127	0.3881 ± 0.0225	0.3964 ± 0.0304
arcene_valid	**0.6509 ± 0.0257**	0.5939 ± 0.0031	0.5600 ± 0.0000	0.5626 ± 0.0082	0.5148 ± 0.0111	0.5283 ± 0.0217	0.5600 ± 0.0206

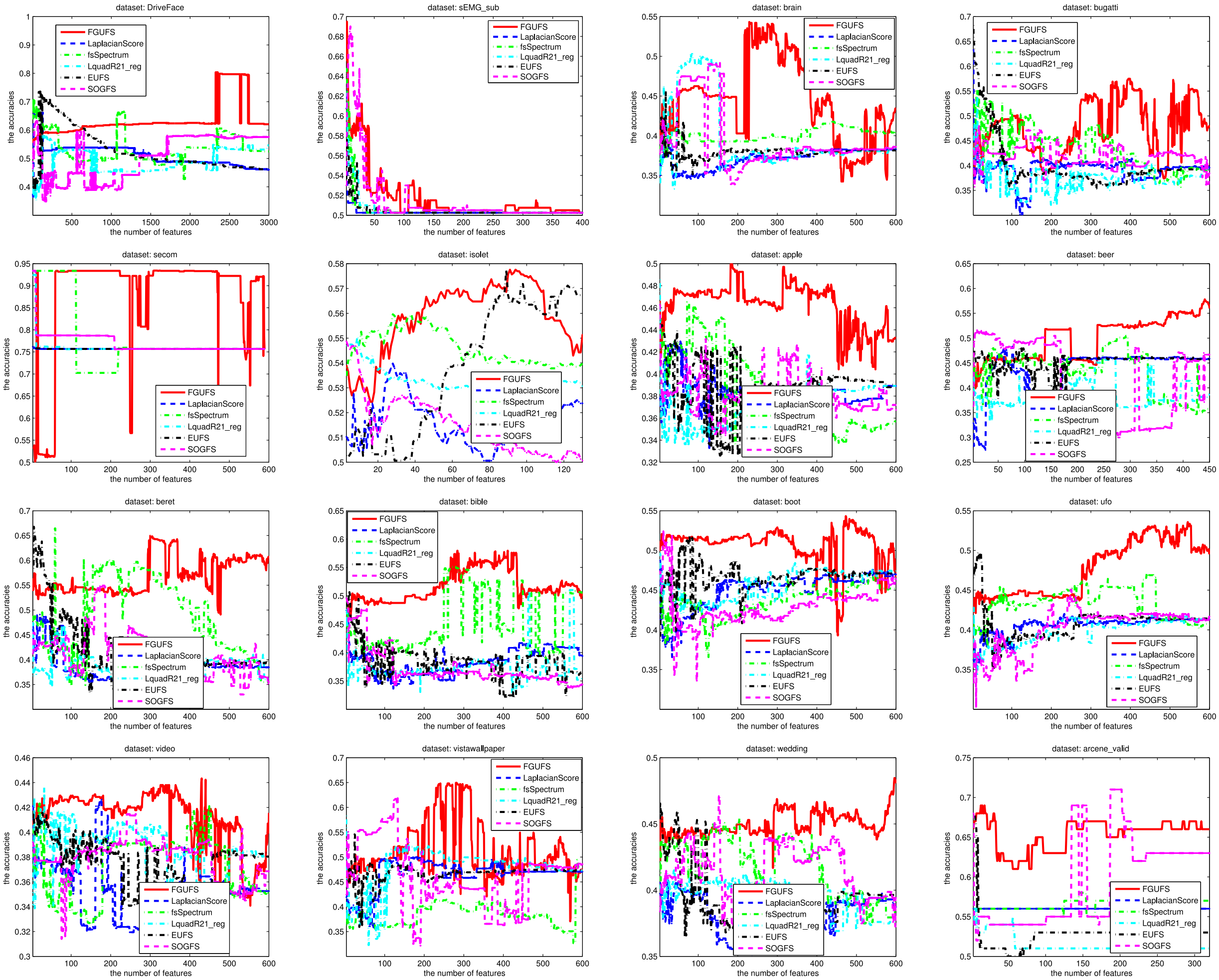

Fig. 2. The clustering accuracies as a function of the number of selected features for different datasets. The *x*-axis represents the number of selected features and the *y*-axis shows the corresponding clustering accuracy for each selected feature subspace. The minimum number of selected features was three.

Table 4
Aligned observations are results of the experimental study, and the ranks in parentheses are used in the computation of the Friedman aligned ranks test.

Dataset	FGUFS	SOGFS	LaplacianScore	fsSpectrum	LquadR21_score	EUFS	Original	Total
DriveFace	1.2657(2)	−0.0613(65)	0.0057(44)	0.2747(17)	−0.1823 (83)	0.1567(23)	−1.4593(112)	346
sEMG_sub	0.0923(31)	0.0713(34)	−0.0467(58)	−0.0237(50)	−0.0247(51)	−0.0367(55)	−0.0317(53)	332
brain	0.4916(12)	−0.0004(45)	−0.2334(93)	0.0606(37)	0.0296(41)	−0.1524(79)	−0.1954(84)	391
bugatti	0.6410(7)	0.0310(40)	−0.2270(90)	0.1740(21)	−0.3970(104)	−0.1610(80)	−0.0610(64)	406
secom	1.1364(3)	0.0594(38)	−0.0666(67)	0.1684(22)	−0.0356(54)	−0.0656(66)	−1.1966(111)	361
isolet	0.1917(20)	−0.2503(96)	−0.2183(87)	0.0707(35)	−0.0493(59)	−0.0073(47)	0.2627(19)	363
apple	0.6793(6)	−0.1397(77)	−0.1177(75)	−0.0927(71)	−0.2287(92)	−0.0607(63)	0.0397(56)	440
beer	0.6274(8)	−0.3496(101)	−0.0016(46)	−0.0596(62)	−0.5096(106)	0.1494(24)	0.1434(26)	373
beret	1.3410(1)	−0.2350(95)	−0.5200(107)	0.5120(11)	−0.5040(105)	−0.2230(89)	−0.3710(102)	510
bible	1.1183(4)	−0.3267(100)	−0.2127(86)	0.4823(13)	−0.3197(99)	−0.2207(88)	−0.5207(108)	498
boot	0.4784(14)	−0.3016(98)	−0.1006(73)	−0.2286(91)	−0.0296(52)	0.0784(32)	0.1034(29)	389
ufo	0.5736(9)	−0.1814(81)	−0.1444(78)	0.1366(27)	−0.2074(85)	−0.1184(76)	−0.0584(61)	417
video	0.3287(16)	−0.0133(48)	−0.2763(97)	−0.0673(68)	0.0767(33)	−0.0543(60)	0.0057(43)	365
vistawallpaper	0.5123(10)	−0.0227(49)	0.0113(42)	−0.7847(110)	0.1483(25)	0.0393(39)	0.0963(30)	305
weddingdress	0.4383(15)	0.0693(36)	−0.2347(94)	0.1153(28)	−0.1077(74)	−0.1817(82)	−0.0987(72)	401
arcene_valid	0.8369(5)	0.2669(18)	−0.0721(69.5)	−0.0461(57)	−0.5241(109)	−0.3891(103)	−0.0721(69.5)	
Total	**163**	1021	1206.5	720	1172	1006	1039.5	
Av.	**10.1875**	63.8125	75.4063	45	73.25	62.8750	64.9688	

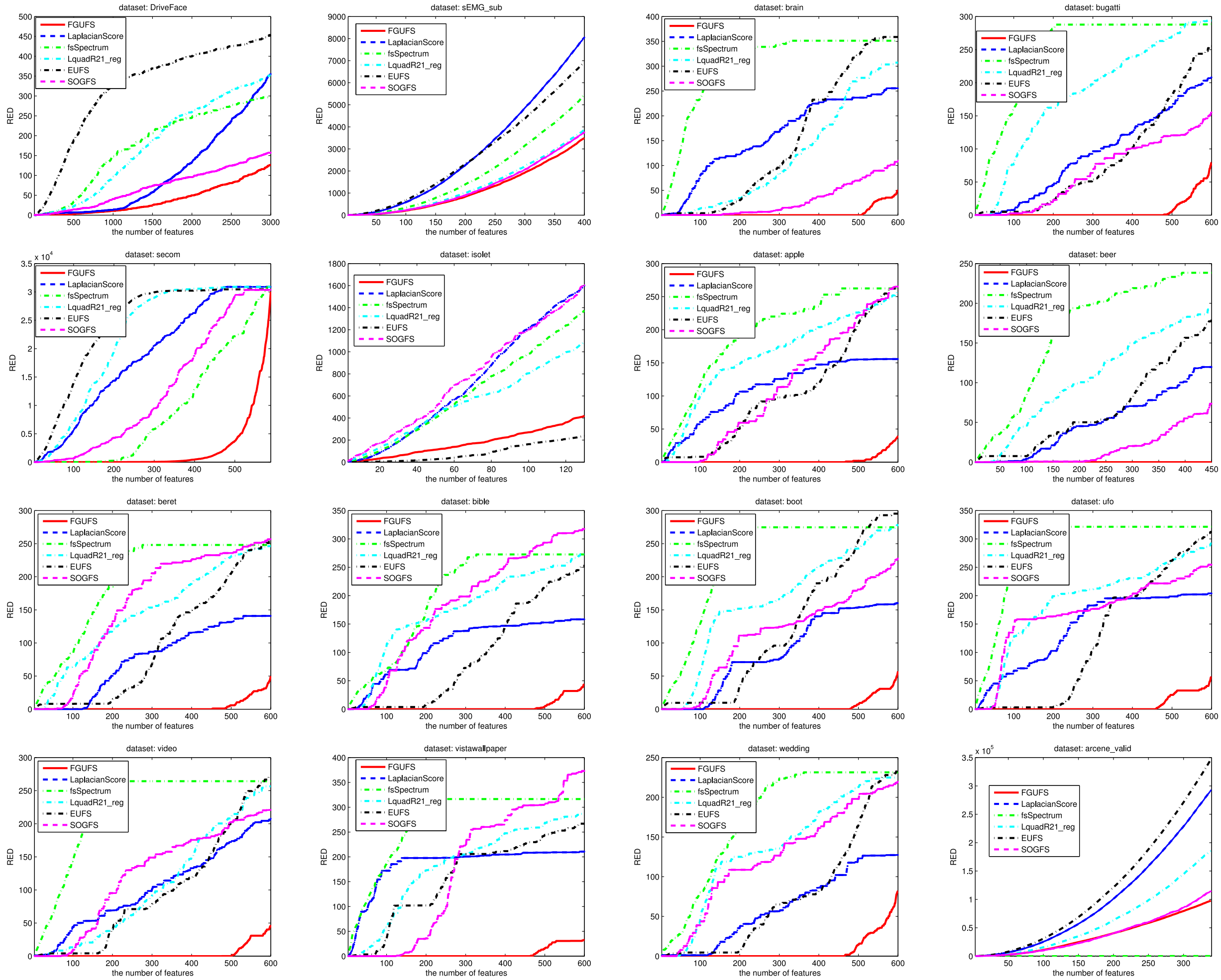

Fig. 3. RED values as a function of the number of selected features for different datasets. The *x*-axis represents the number of selected features, and the *y*-axis shows the corresponding RED value for each selected feature subspace. The minimum number of selected features was three.

Table 5
Average RED values for clustering on the selected features obtained by the six feature selection algorithms. These values are computed from the results displayed in Fig. 3.

Dataset	FGUFS	SOGFS	LaplacianScore	fsSpectrum	LquadR21_score	EUFS
DriveFace	**38.2946**	70.4146	103.0627	176.4106	175.3525	322.3064
sEMG_sub	**1149.3955**	1243.5670	2879.7289	1851.9101	1315.4745	2683.0336
brain	**4.1591**	28.6939	159.8256	303.5970	118.7730	145.5800
bugatti	**6.0556**	64.0142	92.1190	245.8041	183.0733	82.8024
secom	**2050.9982**	26044.2378	18794.6892	8943.6312	22133.2111	24271.3901
isolet	173.0386	829.4883	714.5548	619.1127	533.7387	**83.3866**
apple	**2.5048**	113.4718	111.5940	195.9386	163.1931	103.6921
beer	**4.5187**	16.5310	67.9050	183.8958	130.5867	107.9549
beret	**3.8185**	159.1515	76.3118	194.7961	146.3528	97.0076
bible	**4.2419**	184.3093	112.6645	200.8725	175.0474	94.3906
boot	**4.5523**	112.9620	86.7909	228.7093	164.0930	122.3118
ufo	**6.6835**	170.4752	143.6355	285.5156	191.0317	122.6436
video	**3.9483**	122.9589	102.3063	224.6559	106.7811	97.0585
vistawallpaper	**5.2806**	173.0898	184.7652	273.0364	182.1953	157.8156
weddingdress	**6.8146**	125.3750	61.2869	174.6861	136.7456	75.1762
arcene_valid	36546.3284	38007.5648	98895.1908	521242.4131	61893.2597	**11781.2357**

Table 6
Aggregated RI clustering values for the selected features obtained by the six feature selection algorithms.

Dataset	FGUFS	SOGFS	LaplacianScore	fsSpectrum	LquadR21_score	EUFS
DriveFace	**0.5162** ± **0.0012**	0.4551 ± 0.0073	0.4530 ± 0.0004	0.4767 ± 0.0053	0.4347 ± 0.0042	0.4586 ± 0.0085
sEMG_sub	0.5009 ± 0.0003	**0.5015** ± **0.0005**	0.4989 ± 0.0001	0.4993 ± 0.0002	0.4992 ± 0.0001	0.4989 ± 0.0005
brain	**0.5299** ± **0.0189**	0.5210 ± 0.0014	0.5097 ± 0.0355	0.4619 ± 0.0082	0.5221 ± 0.0361	0.5173 ± 0.0385
bugatti	**0.4984** ± **0.0075**	0.4915 ± 0.0004	0.4920 ± 0.0069	0.4926 ± 0.0093	0.4867 ± 0.0053	0.4922 ± 0.0115
secom	**0.8172** ± **0.1192**	0.6462 ± 0.0013	0.6318 ± 0.0003	0.6772 ± 0.1020	0.6368 ± 0.0285	0.6319 ± 0.0006
isolet	**0.5069** ± **0.0030**	0.5003 ± 0.0001	0.5005 ± 0.0007	0.5040 ± 0.0013	0.5021 ± 0.0007	0.5040 ± 0.0039
apple	0.5308 ± 0.0129	**0.5417** ± **0.0002**	0.5316 ± 0.0143	0.5099 ± 0.0279	0.5234 ± 0.0159	0.5368 ± 0.0195
beer	0.5369 ± 0.0279	**0.5431** ± **0.0010**	0.5211 ± 0.0219	0.4874 ± 0.0166	0.5130 ± 0.0148	0.5205 ± 0.0261
beret	**0.5502** ± **0.0103**	0.4955 ± 0.0003	0.4923 ± 0.0053	0.5212 ± 0.0273	0.4909 ± 0.0044	0.4992 ± 0.0170
bible	**0.5495** ± **0.0226**	0.5309 ± 0.0004	0.5374 ± 0.0068	0.5190 ± 0.0193	0.5349 ± 0.0048	0.5357 ± 0.0100
boot	0.5511 ± 0.0210	0.5541 ± 0.0005	0.5574 ± 0.0046	0.5304 ± 0.0313	**0.5633** ± **0.0046**	0.5572 ± 0.0180
ufo	0.5645 ± 0.0144	0.5618 ± 0.0004	**0.5655** ± **0.0047**	0.5269 ± 0.0233	0.5632 ± 0.0061	0.5645 ± 0.0175
video	0.5350 ± 0.0127	**0.5475** ± **0.0001**	0.5430 ± 0.0073	0.5200 ± 0.0162	0.5353 ± 0.0118	0.5352 ± 0.0127
vistawallpaper	0.5333 ± 0.0371	0.5253 ± 0.0007	0.5298 ± 0.0168	0.4884 ± 0.0101	**0.5336** ± **0.0084**	0.5319 ± 0.0122
weddingdress	0.5151 ± 0.0162	0.5313 ± 0.0004	**0.5362** ± **0.0035**	0.5145 ± 0.0234	0.5349 ± 0.0028	0.5343 ± 0.0098
arcene_valid	**0.5423** ± **0.0133**	0.5197 ± 0.0013	0.5022 ± 0.0000	0.5030 ± 0.0020	0.4956 ± 0.0012	0.4975 ± 0.0077

Table 7
Aggregated clustering purities for the selected features obtained by the six feature selection algorithms.

Dataset	FGUFS	SOGFS	LaplacianScore	fsSpectrum	LquadR21_score	EUFS
DriveFace	**0.9024**	0.9011	0.9010	0.9010	0.9010	0.9011
sEMG_sub	**0.5175**	0.5154	0.5035	0.5058	0.5057	0.5045
brain	**0.5578**	0.5482	0.5477	0.5477	0.5477	0.5477
bugatti	**0.6975**	0.6973	0.6973	0.6973	0.6973	0.6973
secom	**0.9336**	0.9336	0.9336	0.9336	0.9336	0.9336
isolet	0.5353	0.5141	0.5416	0.5473	0.5301	**0.5526**
apple	**0.5190**	0.5166	0.5167	0.5166	0.5166	0.5166
beer	0.5472	**0.5582**	0.5467	0.5437	0.5437	0.5450
beret	0.6896	0.6895	0.6895	0.6895	0.6895	**0.6898**
bible	**0.5529**	0.5270	0.5317	0.5369	0.5247	0.5260
boot	**0.5355**	0.4921	0.4994	0.4936	0.5151	0.4955
ufo	0.4871	0.4721	0.4837	0.4611	0.4782	**0.4894**
video	**0.4472**	0.4471	0.4467	0.4466	0.4468	0.4466
vistawallpaper	0.6467	0.6397	**0.6469**	0.6327	0.6452	0.6427
weddingdress	0.4721	0.4724	0.4715	0.4592	0.4711	**0.4768**
arcene_valid	**0.6512**	0.6036	0.5600	0.5646	0.5600	0.5620

Table 8
Running times in seconds of the six feature selection algorithms on the datasets.

Dataset	FGUFS	SOGFS	LaplacianScore	fsSpectrum	LquadR21_score	EUFS
DriveFace	86.2627	24458.8484	**0.1367**	6.2542	2795.7639	37.5841
sEMG_sub	14.9605	2706.1599	**0.0462**	1.1287	125.4637	40.0862
brain	3.7483	422.8922	**0.0536**	3.9844	25.3108	14.8432
bugatti	3.7753	314.9103	**0.0511**	3.8081	25.7482	15.8075
secom	3.5898	119.3594	**0.0575**	13.2770	10.6062	25.9769
isolet	3.9503	36.4483	**0.0605**	14.5688	12.1631	26.7744
apple	3.7792	817.1589	**0.0596**	3.7095	26.7432	15.1322
beer	3.7627	893.8309	**0.0563**	3.7028	25.6826	19.6022
beret	3.7659	530.6257	**0.0511**	3.6906	26.1828	14.7244
bible	3.7608	486.1239	**0.0490**	3.3790	24.7814	13.2966
boot	4.1047	358.4164	**0.0500**	3.3805	26.2758	13.6209
ufo	3.9850	1559.0823	**0.0533**	3.7857	25.6541	23.8379
video	3.7739	577.7876	**0.0577**	4.4421	25.1477	17.0262
vistawallpaper	3.7939	230.7871	**0.0463**	3.1177	34.6376	17.5579
weddingdress	3.9919	413.1151	**0.0531**	3.8438	26.5721	23.5552
arcene_valid	951.1334	89400.1874	**0.0923**	0.4797	6301.1950	13.0191

RI was also employed to evaluate the clustering results. Table 6 shows the Rand index results obtained by the various feature selection methods. Among the six methods, FGUFS obtains the best result in eight of the datasets, whereas SOGFS achieves the best result in four, and LaplacianScore and LquadR21_score each obtain the best result in two. To some degree, these results also show that the proposed algorithm outperforms the others.

Finally, the purity results for the various feature selection methods are presented in Table 7. Among the six methods, FGUFS obtains the best result for 10 datasets, whereas the other five algorithms collectively achieve the other six best

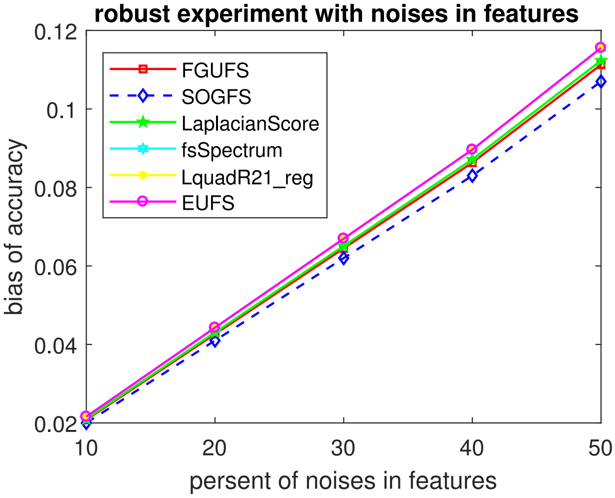

Fig. 4. Clustering accuracies for different percentages of noise in the features.

results. We also note that all the algorithms obtain the same result on the dataset secom. These results show that the proposed algorithm is highly meaningful.

In addition to the four evaluations, the runtimes on the different datasets were recorded for each algorithm, and are listed in Table 7. Among the six methods, LaplacianScore, a filter-type method that evaluates features based on their locality-preserving power, is the most efficient. Our proposed FGUFS algorithm requires time to construct the factor graph and perform the inference. However, its runtime is neutral, occupying a middle rank.

Moreover, noise was randomly added to the existing features for the experiment, and the corresponding biases of the accuracies were recorded. The biases of the accuracies were calculate according to $b = |a_{noise} - a_{nonoise}|$, where a_{noise} is the accuracy with noise in features and $a_{nonoise}$ is the accuracy without noise in features. In Fig. 4, the x axis represents the percentages of noise in features, and the y axis shows the biases of accuracies corresponding to these. We can see that the accuracies of all algorithms oscillate with the increasing percentage of noise. however, the variance is far lower than the noise ratio. FGUFS performs second best in this experiment, and the results show that the proposed algorithm is highly competitive among the six algorithms.

To summarize, the experimental results show that the proposed feature selection algorithm FGUFS is able to select a subset of features resulting in a high clustering accuracy, RI, and purity, while containing few redundant features. Therefore, FGUFS significantly enhances the unsupervised feature selection performance and can deal with noisy features.

6. Conclusions

In this study, a factor graph model has been proposed for unsupervised feature selection. To best of our knowledge, this is the first study to solve feature selection by using a factor graph, which is a filter method utilizing feature similarities and message passing between features. The only input to FGUFS is the feature similarity matrix, and so it is simple but effective. In FGUFS, the MIC is used to measure the similarities between features. Then, a factor graph model for feature selection is utilized, and the message-passing algorithm is employed to perform the inference. Extensive experiments on datasets from UCI, the NIPS2003 competition, and Microsoft were conducted to evaluate the effectiveness of our proposed algorithm. The results demonstrate that FGUFS outperforms other state-of-the-art unsupervised feature selection algorithms, and that it can select a subset of features with high accuracy and low redundancy.

There are several possible avenues for future research. First, the problem of determining the best number of features for unsupervised learning has yet to be explored. Second, it would be natural to study the factor-graph-based framework for supervised feature selection. Furthermore, the MapReduce framework can be used to modify the proposed method, and the runtime can be improved.

Acknowledgment

This work is partially supported by NSFC (Nos. 61573292, 61603313 and 61773324.), the Fundamental Research Funds for the Central Universities (No. 2682017CX097), and the Blockchain Technology Fund of Hi-Tech Information Technology Research Institute of Chengdu (No. 2018H01207).

References

[1] M. Banerjee, N.R. Pal, Unsupervised feature selection with controlled redundancy (UFeSCoR), IEEE Trans. Knowl. Data Eng. 27 (12) (2015) 3390–3403.

[2] G. Brown, A. Pocock, M.-J. Zhao, M. Luján, Conditional likelihood maximisation: a unifying framework for information theoretic feature selection, J. Mach. Learn. Res. 13 (1) (2012) 27–66.

[3] D. Cai, C. Zhang, X. He, Unsupervised feature selection for multi-cluster data, in: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2010, pp. 333–342.

[4] M. Charikar, S. Guha, É. Tardos, D.B. Shmoys, A constant-factor approximation algorithm for the *k*-median problem, in: Proceedings of the Thirty-First Annual ACM Symposium on Theory of Computing, ACM, 1999, pp. 1–10.

[5] J.G. Dy, C.E. Brodley, Feature selection for unsupervised learning, J. Mach. Learn. Res. 5 (4) (2004) 845–889.

[6] B.J. Frey, D. Dueck, Clustering by passing messages between data points, Science 315 (5814) (2007) 972–976.

[7] M.J. Gangeh, H. Zarkoob, A. Ghodsi, Fast and scalable feature selection for gene expression data using hilbert-schmidt independence criterion, IEEE/ACM Trans. Comput. Biol. Bioinf. 14 (1) (2017) 167–181.

[8] S. García, A. Fernández, J. Luengo, F. Herrera, Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: experimental analysis of power, Inf. Sci. (Ny) 180 (10) (2010) 2044–2064.

[9] S. García, J. Luengo, F. Herrera, Tutorial on practical tips of the most influential data preprocessing algorithms in data mining, Knowl. Based Syst. 98 (2016) 1–29.

[10] S. García, J. Luengo, F. Herrera, Data Preprocessing in Data Mining, New York: Springer, 2015.

[11] Q. Gu, M. Danilevsky, Z. Li, J. Han, Laplacian score for feature selection, in: Proceedings of the 15th International Conference on Artificial Intelligence and Statistics, 2012, pp. 477–485.

[12] Y. Guan, J.G. Dy, M.I. Jordan, A unified probabilistic model for global and local unsupervised feature selection, in: Proceedings of the 28th International Conference on Machine Learning, 2011, pp. 1073–1080.

[13] J. Gui, Z. Sun, S. Ji, D. Tao, T. Tan, Feature selection based on structured sparsity: a comprehensive study, IEEE Trans. Neural Netw. Learn Syst. PP (99) (2016) 1–18.

[14] M.A. Hall, Correlation-based feature selection for machine learning, The University of Waikato, 1999 Ph.D. thesis.

[15] Y. Han, Y. Yang, Y. Yan, Z. Ma, N. Sebe, X. Zhou, Semisupervised feature selection via spline regression for video semantic recognition, IEEE Trans. Neural Netw. Learn Syst. 26 (2) (2015) 252.

[16] M. He, Research on feature selection algorithm based on mixed model, in: Proceedings of the IEEE International Conference on Computer and Electrical Engineering, IEEE, 2008, pp. 70–72.

[17] X. He, D. Cai, P. Niyogi, Locality preserving feature learning, in: Proceedings of the Advances in Neural Information Processing Systems, 2005, pp. 507–514.

[18] D.G. Hela, O. Yacine, M. Maria, D. Thierry, G. Emmanuel, H. Stphane, Wood moisture content prediction using feature selection techniques and a kernel method, Neurocomputing 237 (2016) 79–91.

[19] G.E. Hinton, R.R. Salakhutdinov, Reducing the dimensionality of data with neural networks, Science 313 (5786) (2006) 504–507.

[20] C. Hou, F. Nie, X. Li, D. Yi, Y. Wu, Joint embedding learning and sparse regression: a framework for unsupervised feature selection., IEEE Trans. Cybern. 44 (6) (2017) 793–804.

[21] G. Lastra, O. Luaces, J.R. Quevedo, A. Bahamonde, Graphical feature selection for multilabel classiffication tasks, in: Proceedings of the International Symposium on Intelligent Data Analysis, 2011, pp. 246–257.

[22] M.H. Law, A.K. Jain, M.A.T. Figueiredo, Feature selection in mixture-based clustering, in: Proceedings of the Advances in Neural Information Processing Systems, 2002, pp. 625–632.

[23] M.H.C. Law, M.A.T. Figueiredo, A.K. Jain, Simultaneous feature selection and clustering using mixture models, IEEE Trans. Pattern Anal. Mach. Intell. 26 (9) (2004) 1154–1166.

[24] M. Luo, F. Nie, X. Chang, Y. Yang, A.G. Hauptmann, Q. Zheng, Adaptive unsupervised feature selection with structure regularization, IEEE Trans. Neural Netw. Learn Syst. (99) (2018) 1–13.

[25] L. Ma, M. Li, Y. Gao, A novel wrapper approach for feature selection in object-based image classification using polygon-based cross-validation, IEEE Geosci. Remote Sens. Lett. 14 (3) (2017) 409–413.

[26] P. Mitra, C. Murthy, S.K. Pal, Unsupervised feature selection using feature similarity, IEEE Trans. Pattern Anal. Mach. Intell. 24 (3) (2002) 301–312.

[27] D.C. Mocanu, E. Mocanu, P.H. Nguyen, M. Gibescu, A. Liotta, A topological insight into restricted Boltzmann machines, Mach. Learn. 104 (2–3) (2016) 243–270.

[28] F. Nie, W. Zhu, X. Li, Unsupervised feature selection with structured graph optimization, in: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, 2016, pp. 1302–1308.

[29] H. Peng, F. Long, C. Ding, Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy, IEEE Trans. Pattern Anal. Mach. Intell. 27 (8) (2005) 1226–1238.

[30] D.N. Reshef, Y.A. Reshef, H.K. Finucane, S.R. Grossman, G. McVean, P.J. Turnbaugh, E.S. Lander, M. Mitzenmacher, P.C. Sabeti, Detecting novel associations in large data sets, Science 334 (6062) (2011) 1518–1524.

[31] M. Robnik-Šikonja, I. Kononenko, Theoretical and empirical analysis of ReliefF and RReliefF, Mach. Learn. 53 (1) (2003) 23–69.

[32] L. Song, A. Smola, A. Gretton, K.M. Borgwardt, J. Bedo, Supervised feature selection via dependence estimation, in: Proceedings of the 24th International Conference on Machine Learning, ACM, 2007, pp. 823–830.

[33] J. Sun, A. Zhou, Unsupervised robust Bayesian feature selection, in: Proceedings of the International Joint Conference on Neural Networks, 2014, pp. 558–564.

[34] L. Tao, C. Zhang, M. Ogihara, A comparative study of feature selection and multiclass classification methods for tissue classification based on gene expression, Bioinformatics 20 (15) (2004) 2429–2437.

[35] A.K. Uysal, An improved global feature selection scheme for text classification, Expert Syst. Appl. 43 (C) (2016) 82–92.

[36] D. Wang, H. Zhang, R. Liu, Gs-orthogonalization based “basis feature” selection from word co-occurrence matrix, in: Proceedings of the IEEE International Conference on Data Mining, 2016, pp. 1027–1032.

[37] S. Wang, J. Tang, H. Liu, Embedded unsupervised feature selection, in: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, 2015, pp. 470–476.

[38] J. Weston, A. Elisseeff, B. Schölkopf, M. Tipping, Use of the zero norm with linear models and kernel methods, J. Mach. Learn. Res. 3 (2003) 1439–1461.

[39] Z. Xu, I. King, M.-T. Lyu, R. Jin, Discriminative semi-supervised feature selection via manifold regularization, IEEE Trans. Neural Netw. 21 (7) (2010) 1033–1047.

[40] H. Yang, M.R. Lyu, I. King, Efficient online learning for multitask feature selection, ACM Trans. Knowl. Discov. Data 7 (2) (2013) 1–27.

[41] Y. Yang, H.T. Shen, Z. Ma, Z. Huang, X. Zhou, L 2, 1-norm regularized discriminative feature selection for unsupervised learning, in: Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence-Volume Volume Two, AAAI Press, 2011, pp. 1589–1594.

[42] J.S. Yedidia, W.T. Freeman, Y. Weiss, Constructing free-energy approximations and generalized belief propagation algorithms, IEEE Trans. Inf. Theory 51 (7) (2005) 2282–2312.

[43] L. Yu, H. Liu, Efficient feature selection via analysis of relevance and redundancy, J. Mach. Learn. Res. 5 (12) (2004) 1205–1224.

[44] X. Zhao, W. Deng, Y. Shi, Feature selection with attributes clustering by maximal information coefficient, Proced. Comput. Sci. 17 (2) (2013) 70–79.

[45] Z. Zhao, H. Liu, Semi-supervised feature selection via spectral analysis, in: Proceedings of the 7th SIAM International Conference on Data Mining, SIAM, 2007, pp. 641–646.

[46] Z. Zhao, H. Liu, Spectral feature selection for supervised and unsupervised learning, in: Proceedings of the 24th International Conference on Machine Learning, ACM, 2007, pp. 1151–1157.

[47] Z. Zhao, L. Wang, H. Liu, Efficient spectral feature selection with minimum redundancy, in: Proceedings of Twenty-Fourth AAAI Conference on Artificial Intelligence, 2010, pp. 673–678.

[48] Z. Zhao, X. He, D. Cai, L. Zhang, W. Ng, Y. Zhuang, Graph regularized feature selection with data reconstruction, IEEE Trans. Knowl. Data Eng. 28 (3) (2016) 689–700.

[49] S. Zhu, D. Wang, K. Yu, T. Li, Y. Gong, Feature selection for gene expression using model-based entropy, IEEE/ACM Trans. Comput. Biol. Bioinf. 7 (1) (2010) 25–36.