Adaptive Hypergraph Learning for Unsupervised Feature Selection*

Xiaofeng Zhu1,2, Yonghua Zhu1,3, Shichao Zhang1,2, Rongyao Hu1,2, Wei He1,2
1 Guangxi Key Lab of Multi-source Information Mining & Security, China
2 Guangxi Normal University, China
3 Guangxi University, China

Abstract
In this paper, we propose a new unsupervised feature selection method to jointly learn the similarity matrix and conduct both subspace learning (via learning a dynamic hypergraph) and feature selection (via a sparsity constraint). As a result, we reduce the feature dimensions using different methods (i.e., subspace learning and feature selection) from different feature spaces, and thus makes our method select the informative features effectively and robustly. Experimental results show that our proposed method outperforms all the comparison methods in terms of clustering tasks.

1 Introduction
With the rapid growth of contemporary information technology, high-dimensional data becomes very common for representing the data. Due to the challenges such as the curse of dimensionality, storage and computation costs, it is an urgent problem to deal with high-dimensional data in practical applications. Feature selection, which selects the informative features from high-dimensional data, has been becoming a popular solution for solving the problem of high-dimensional data [Chang et al., 2014; Zhu et al., 2013; Gao et al., 2013; Zhu et al., 2014]. In particular, unsupervised feature selection (UFS) without using the label information is attracting a lot of interests since it is difficult to obtain the labels in practical applications [Zhu et al., 2017; Chang et al., 2016].

Current UFS methods include three types, i.e., filter method [He et al., 2005], wrapper method [Chang et al., 2017; Tabakhj et al., 2014], and embedded method [Zhu et al., 2016b]. The embedded method constructs a learning model to output a subset of the features which achieves the best accuracy of the model, and has been shown superior both filter method and wrapper method [Morclhid et al., 2014]. Many embedded methods first construct a similarity matrix measuring the pair-wise relation among the training data via a simple graph to preserve either the local or global structures of the training data, and then use the resulting graph regularizer plus the sparsity constraint (e.g., an $\ell_1$-norm regularizer or an $\ell_{2,1}$-norm regularizer) to select informative features [Zhao et al., 2013; Zhu et al., 2016b].

Current embedded methods still have limitations to be addressed. First, the two-step strategy (i.e., learning similarity matrix and conducting feature selection) of the embedded methods possibly degrades the performance of feature selection as the similarity matrix learning aims at achieving an optimal similarity relation, instead of the feature selection results. Second, current embedded methods construct the similarity matrix from the original data which usually contain redundant and irrelevant features, and thus may select uninformative features. Third, the similarity matrix is constructed via a simple graph, which measures the pair-wise relations of the training data, instead of considering their high-order relations, so that not sufficient to capture the complex structures in the training data.

To address the above issues, in this paper, we propose a new unsupervised embedded feature selection method, namely Adaptively Hypergraph Learning for Feature Selection (AHLFS), involving three components: 1) constructing the similarity matrix from the low-dimensional space of the original training data (i.e., the low-dimensional training data) using a hypergraph to preserve their high-order local structures, 2) penalizing an orthogonal constraint on the covariance matrix of the low-dimensional training data to preserve their global structures, and 3) using an $\ell_{2,1}$-norm sparsity constraint, to reduce the dimensions of the features. We further propose a new alternative optimization method to adaptively adjust each of these components, so that learning the similarity matrix from the low-dimensional training data and outputting reliable and informative features.

Compared with the current feature selection methods, the proposed AHLFS has the following contributions:

- Propose a novel UFS method via jointly conducting subspace learning and feature selection in a framework since our first two components actually conduct subspace learning by preserving the local and global structures of the low-dimensional training data. Our method enables to reduce the feature dimensions via different modes (i.e., subspace learning and features selection) from different spaces, i.e., the low-dimensional feature space of the training data preserves two complementary structures and the original feature space removes the redundant/irrelevant features.
Propose reasonable constraints. We embed subspace learning to the feature selection model for strengthening the discriminative ability of feature selection to remove the redundant/irrelevant features. Moreover, the proposed alternative optimization method adaptively adjusts them to achieve their individual optimizations. Experimental results on benchmark datasets show that our method outperforms the state-of-the-art methods in clustering tasks using the selected features. This further verifies the effectiveness and robustness of the designed constraints in the proposed method.

2. Approach

This paper denotes matrices as boldface uppercase letters, vectors as boldface lowercase letters, and scalars as normal italic letters, also denotes the $i$-th row and $j$-th column of a matrix $X = [x_{ij}]$ as $x^i$ and $x_j$, and its Frobenius norm and $\ell_{2,1}$-norm as $||X||_F = \sqrt{\sum_i \sum_j x_{i,j}^2}$ and $||X||_{2,1} = \sum_i \sqrt{\sum_j x_{i,j}^2}$, and further denotes the transpose, the trace, and the inverse, of a matrix $X$, as $X^T$, $\text{tr}(X)$, and $X^{-1}$.

2.1. Hypergraph Learning

The traditional graph methods use the pair-wise relations among the training data to preserve the geometric structures of the training data. This usually is insufficient to capture the complex relations in the training data. Give an illustration on the author-paper relation in Figure 1. The left subfigure of Figure 1 uses a simple graph to describe the author-paper relations, e.g., $a_1$ vs. $a_2$ (i.e., $a_1$ and $a_2$ are the authors of a paper), $a_2$ vs. $a_3$, and $a_2$ vs. $a_4$, but cannot imply the relations we may really focus on, e.g., the first paper has three authors (i.e., $a_1$, $a_2$, and $a_3$) and the second paper has two authors (i.e., $a_2$ and $a_4$). The right subfigure of Figure 1 easily indicates these two types of relations via constructing a hypergraph. Hence, this paper focuses on using a hypergraph to preserve the local structures of the training data as a hypergraph may capture more complex relations than a simple graph [Zhou et al., 2006; Somu et al., 2016; Gao et al., 2014].

By denoting a hypergraph as $G = (V, E, w)$, where $V = [v_i]$ and $E = [e_i]$, respectively, are the set of the vertices and the hyperedges, and $w = [w_i]$ is the weight of the hyperedges, the construction of a hypergraph includes three sequential steps: 1) the incidence matrix $H$ representing the binary vertex-edge relation, where each element is defined as:

$$H(v_i, e_j) = \begin{cases} 1, & \text{if } v_i \in e_j, \\ 0, & \text{otherwise.} \end{cases}$$

(1)

2) the weight vector $w$ measuring the importance of hyperedges; and 3) the hypergraph Laplacian $L$, i.e., the normalized Laplacian matrix of the resulting hypergraph.

Different from the simple graph where each edge represents the vertex-to-vertex relation, the incidence matrix $H$ of a hypergraph describes the vertex-to-hyperedge relation. To achieve this, first, given the training data $X \in \mathbb{R}^{c \times n}$ where $c$ and $n$, respectively, indicate the numbers of the features and the samples, we regard each sample as one vertex and try to generate a hyperedge for each vertex by following the method in [Zhou et al., 2006]. More specifically, we generate the hyperedge $e_i$ by the following formulation:

$$e_i = \{v_j | \theta(x_i, x_j) \leq 0.1 \sigma_i^2, i, j = 1, \ldots, n \}$$

(2)

where $\theta(x_i, x_j)$ indicates a similarity measurement between $x_i$ and $x_j$ (e.g., Euclidean distance on a Gaussian kernel function in this paper) and $\sigma_i$ is the average similarity between $x_i$ and each of the other samples. Such a threshold method is very popular for the construction of the hyperedges [Somu et al., 2016; Gao et al., 2012; Peng et al., 2016a] and obviously results in that different samples have different numbers of nearest neighbors, instead of the previous methods [Elhamifar et al., 2016; Peng et al., 2016b] which set the same number of nearest neighbors to all the samples.

Second, we use the resulting incidence matrix $H$ and the training data to learn the importance of each hyperedge, i.e., $w$. After this, we further obtain $d(e_i)$ (i.e., the degree of a hyperedge $e_i$ via $d(e_i) = \sum_{v_j \in E} h(v_j, e_i)$) and $d(v_j)$ (i.e., the degree of a vertex $v_j$ via $d(v_j) = \sum_{e_i \in E} \sum_{v_j \in E} w(e_i) h(v_j, e_i)$).

Third, we obtain the hypergraph Laplacian matrix as:

$$L = I - D^{-\frac{1}{2}}HWD^{-\frac{1}{2}}H^T D^{-\frac{1}{2}}$$

(3)

where $I \in \mathbb{R}^{n \times n}$ is an identity matrix, $D_e$, $D_c$, and $W$, respectively, are the diagonal matrices of $\delta = [\delta(e_i)]$, $d = [d(v_j)]$, and $w = [w(e_i)]$.

If we want to use a hypergraph to preserve the local structures of the training data, we follow the literatures [Zhou et al., 2006; Zhang et al., 2016; Peng et al., 2017] to have the following objective function:

$$\min_{s^TXX^TS = I} \sum_{e \in E, x_i, x_j \in V} \Delta \left( \frac{S^T x_i}{d(x_i)} - \frac{S^T x_j}{d(x_j)} \right)^2$$

(4)

where $\Delta = \frac{w(e_i) h(x_i, e) h(x_j, e)}{d(x_i) d(x_j)}$, and $S \in \mathbb{R}^{c \times c}$ is the weight matrix. Obviously, Eq. (4) is equivalent to:

$$\min_{s^TXX^TS = I} \text{tr}(S^T XLX^T S)$$

(5)

It is noteworthy that the rank of the incidence matrix $H$ is no larger than the value of $\min\{V|, |E|\}$, where $|V|$ and $|E|$, respectively, are the number of the vertexes and the hyperedges. Therefore, many previous methods set $|V| = |E|$ for computational efficiency.
where the orthogonal constraint on the covariance matrix of $X$ (i.e., $STXX^T S = I$) can be regarded to implicitly conduct subspace learning, i.e., PCA, which preserves the global structures of the training data [Morchid et al., 2014].

### 2.2 Proposed Method

The three components for the construction of a hypergraph in Section 2.1 are sequential. Thus the quality of either $W$ or $L$ depends on $H$. However, $H$ is learnt from the original training data, which usually contain redundant and irrelevant features. Thus the low-quality $H$ is not able to output the high-quality $L$ so that forbidding to effectively remove the noisy/redundant features via Eq. (4). In this paper, we couple the learning of the incidence matrix $H$ with the learning of the similarity matrix $S$ in a formulation. We expect to iteratively update each of them by fixing the others, so that they are updated adaptively to output the optimal $H$ and $S$. We thus design the final objective function for our AHLFS method as follows:

$$\min_{S,H,D,e,W} \sum_{e \in E, x_i \in S} \Delta \| \frac{S^T x_i}{d(x_i)} - \frac{S^T x_j}{d(x_j)} \|_2^2 + \alpha \| W \|_F^2 + \beta \| S \|_{2,1}^2$$

$$s.t., w^T 1 = 1, w_i > 0, S^T XX^T S = I$$

where $W = \text{diag}(w)$ and $w_i$ is the $i$-th element of the vector $w$. Eq. (6) can directly be changed to:

$$\min_{S,H,D,e,W} tr(S^T XLX^T S) + \alpha \| W \|_F^2 + \beta \| S \|_{2,1}^2$$

$$s.t., w^T 1 = 1, w_i > 0, S^T XX^T S = I$$

where $\alpha$ and $\beta$ are two tuning parameters, $1$ is a vector whose elements are $1$. The $\ell_{2,1}$-norm on $S$ pushes to produce the row sparsity on $S$ to select the informative features, while the constraint $S^T XX^T S = I$ actually conducts subspace learning (i.e., PCA) to make the feature selection discriminative [Ang et al., 2016]. Thus the variable $S$ is used to simultaneously select the informative features (via the sparsity constraint) and conduct subspace learning (i.e., preserving the local structures via the first term of Eq. (7) and the global structures via the orthogonal constraint) in the low-dimensional training data.

### 2.3 Optimization

Eq. (6) is not jointly convex to all five variables ($i.e., W, S, D, e$, and $H$), but is convex for each variable while fixing the others. Thus we employ the alternative optimization strategy to optimize Eq. (6), i.e., iteratively optimizing each variable while fixing the others until the algorithm converges.

#### Update S by fixing other variables

After fixing the other variables, the objective function with respect to $S$ becomes:

$$\min_S tr(S^T XLX^T S) + \beta \| S \|_{2,1}^2$$

$s.t., S^T XX^T S = I$ (8)

Since the $\ell_{2,1}$-norm regularizer is convex and non-smooth, we employ the framework of iteratively reweighted least squares (IRLS) [Wolfe and Schwetlick, 1988] to optimize $S$, via changing Eq. (8) to:

$$\min_{S^T XX^T S = I} tr(S^T XLX^T S + \beta S^T PS)$$

where $P$ is a diagonal matrix, which element is defined as:

$$p_{i,i} = \frac{1}{\sigma_i^2}, i = 1, ..., c.$$ (10)

In Eq. (9), both $P$ and $S$ are unknown. Moreover, $P$ depends on $S$. According to the IRLS framework, we design an iterative algorithm to solve problem Eq. (9) by two sequential steps until the algorithm converges: 1) By fixing $S$, we obtain the $P$ by Eq. (10); 2) By fixing $P$, Eq. (9) is changed to an eigen-decomposition problem with respect to $S$, i.e.,

$$\min_{S^T XX^T S = I} tr(S^T (XLX^T + \beta P) S)$$

The optimal solution $S$ in Eq. (11) is the eigenvectors of $(XLX^T + \epsilon I)^{-1} (XLX^T + \beta P)$ since $XLX^T + \epsilon I$ is invertible, where $\epsilon$ is a very small positive value.

#### Update H and D_e by fixing other variables

According to Eq. (2) in Section 2.1, the hyperedges are generated from the original training data and thus may result in an inaccurate hypergraph. To do this, we design to learn the hyperedges from low-dimensional training data, whose redundant and irrelevant features have been removed as much as possible. Thus we use the following formulation to construct the set of the hyperedges:

$$e_i = \{ v_j | \theta(S^T x_i, S^T x_j) \leq 0.1 \bar{\sigma}_i \}, i, j = 1, ..., n$$ (12)

where $\bar{\sigma}_i$ is the average similarity between $S^T x_i$ and each of the other low-dimensional training data.

Eq. (12) indicates that the proposed method learns: 1) the incidence matrix $H$ from the low-dimensional feature space; 2) different numbers of the neighbors for different samples. By contrast, both the previous simple graph methods [Nie et al., 2016; Ha et al., 2017; Zhang et al., 2017a] and the previous hypergraph methods [Somu et al., 2016; Raman et al., 2016; Zhang et al., 2017b] learn the graphs from the original data as well as assume the same number of neighbors for all the samples. Obviously, our method is more flexible and robust than the previous methods in practical applications. This may be the first work to learn a dynamic hypergraph from the low-dimensional training data for simultaneously conducting subspace learning and feature selection in a formulation.

After yielding the incidence matrix $H$, it is easy to work out $D_e$ via the following formulation:

$$\{ \delta(e_i) = \sum_{v_j \in E} h(v_j, e_i), i, j = 1, ..., n \}$$

$$D_e = \text{diag}(\delta)$$

#### Update W and D_e by fixing other variables

By fixing other variables, we obtain the objective function on the variable $W$ as follows:

$$\min_W tr(S^T X (I - D_e^{-\frac{1}{2}} HWD_e^{-\frac{1}{2}} H^T D_e^{-\frac{1}{2}}) X^T)$$

$$+ \alpha \| W \|_F^2, s.t., W^T 1 = 1, w_i > 0$$ (14)
<table>
<thead>
<tr>
<th>Datasets</th>
<th># (Samples)</th>
<th># (Features)</th>
<th># (Classes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pcmac</td>
<td>1943</td>
<td>3289</td>
<td>2</td>
</tr>
<tr>
<td>Madelon</td>
<td>2000</td>
<td>500</td>
<td>2</td>
</tr>
<tr>
<td>Cll</td>
<td>111</td>
<td>11340</td>
<td>3</td>
</tr>
<tr>
<td>Yeast</td>
<td>1484</td>
<td>1470</td>
<td>10</td>
</tr>
<tr>
<td>Usps</td>
<td>1854</td>
<td>256</td>
<td>10</td>
</tr>
<tr>
<td>Mnist</td>
<td>3495</td>
<td>784</td>
<td>10</td>
</tr>
</tbody>
</table>

By letting $Q = D_v^{-1}H^TD_v^{-1}XSS^TX^TD_v^{-1}H$ and $q = diag(Q)$, according to that $W$ is a diagonal matrix, Eq. (14) is changed to the following formulation:

$$\min_w -qw + \alpha \|w\|^2_2, \ s.t., \ w^T1 = 1, \ w_i > 0 \tag{15}$$

We further change Eq. (15) into the following:

$$\min_w \|w - \frac{1}{2\alpha}q\|^2_2, \ s.t., \ w^T1 = 1, \ w_i > 0 \tag{16}$$

We use the lagrangian function to change Eq. (16) to:

$$\Gamma(w, \eta, \gamma) = \|w - \frac{1}{2\alpha}q\|^2_2 - \eta(w^T1 - 1) - \gamma w \tag{17}$$

where $\eta \geq 0$ and $\gamma \geq 0$ are the lagrangian multipliers. Based on the Karush–Kuhn–Tucker conditions, we can obtain the close-form solution for $w_i (i = 1, ..., n)$, as:

$$w_i = \frac{1}{2\alpha}(\eta_i + \gamma), \ i = 1, ..., n \tag{18}$$

3 Experiment Analysis

In this section, we evaluate our proposed AHLFS with the comparison methods in terms of the clustering accuracy of the clustering tasks, on eight public UCI datasets [Frank et al., 2010], whose detail is listed in Table 1.

The comparison methods include Laplacian Score (LS) [He et al., 2005], Minimize the feature Redundancy for spectral Feature Selection (MRFS) [Zhao et al., 2013], Structured Optimal Graph Feature Selection (SOGFS) [Nie et al., 2016], Coupled Dictionary Learning Feature Selection (CDLFS) [Zhu et al., 2016a], Joint Hypergraph Learning and Sparse Regression (JHLSR) [Zhang et al., 2016], and Baseline which uses all features to conduct k means clustering.

In our experiments, we set the parameters’ range as $\{10^{-3}, 10^{-2}, ..., 10^0\}$ where all the methods can achieve their best results. We first use all the feature selection methods to select the features (i.e., $20\%, 30\%, ..., 80\%$ of all the features) and then conduct k means clustering on the selected features. We repeat k means clustering 20 times to report their average results. Finally, we employ the clustering accuracy to evaluate the clustering performance of all the methods.

3.1 Cluster Accuracy

We list the clustering accuracy of all the methods with different numbers of selected features in Figure 2.

Our proposed AHLFS achieves the best clustering performance, followed by JHLSR, CDLFS, SOGFS, MRFS, LS, and Baseline. For example, our method on average improves by 6.0% and 4.9%, compared to Baseline (the worst comparison method) and JHLSR (the best comparison methods). The reason may be that our method 1) conducts subspace learning and feature selection in a framework, and 2) learns the hypergraph from the low-dimensional training data. Moreover, in the comparison methods, the methods (such as JHLSR and SOGFS) satisfying one of these benefits outperform LS which only conducts feature selection without considering any relation among the data.

In Figure 2, two observations show that there are redundant and irrelevant features in the original training data and it is necessary to conduct dimensionality reduction before conducting cluster tasks. First, all the feature selection methods outperform Baseline. For example, LS (the worst feature selection method) on average improves by 3.7%, than Baseline, on all the datasets in our experiments. Second, the clustering accuracy of all feature selection methods first increases with the increase of the dimensions. After reaching to a peak, the clustering accuracy of these methods begins to decrease or even unstable. This trend indicates that a small number of the features cannot explain the samples well so that outputting bad clustering performance, while the excessively large number of the features may add redundant features to degrade the clustering performance.

3.2 Parameters Sensitivity and Convergence

Our objective function has two tuning parameters, i.e., $\alpha$ and $\beta$. We fix the value of $\alpha$ in section 2.3. In Eq. (7), $\beta$ is designed to adjust the sparsity of weight matrix $S$, the larger the value of $\beta$, the more the sparsity of $S$ (i.e., the less features are selected to conduct the clustering tasks). Figure 3 demonstrates the variation of the clustering accuracy with respect to $\beta$ on four datasets2. From Figure 3, our method achieved the best clustering performance on some values of $\beta$, which produce sparsity, i.e., selecting a subset of the features. This verified our conclusion again, i.e., it is necessary to conduct dimensionality reduction on high-dimensional data. For example, on the dataset Mnist, the best range of the values of $\beta$ is $[10^{-3}, 10^{-1}]$, which corresponds to keep around 30% dimensions of all the features, as in Figure 2.

Figure 4 shows the variation of the objective values in Eq. (7), which shows that our proposed optimization method is very efficient, i.e., converging within about 10 iterations.

4 Conclusion

This paper has proposed a novel unsupervised feature selection method by coupling the hypergraph learning and feature selection in an iteration way. In this way, the hypergraph is constructed to capture the complex structures of the

---

2 Other datasets have similar trends for the variation of $\beta$ and we did not report them due to the limited space.
Figure 2: Clustering accuracy of all the methods on eight different datasets.

Figure 3: The variation of our proposed method on the different parameter setting with respect to $\beta$ on four datasets.
low-dimensional training data without the impact of redundant and irrelevant features. This makes our method reduce the feature dimensions using different methods (i.e., subspace learning and feature selection), and thus resulting in an effective and robust feature selection model. Experiment results on benchmark datasets verified the effectiveness and the robustness of the proposed method, compared to the state-of-the-art feature selection method, in terms of clustering accuracy.

Acknowledgements

This work was supported in part by the Nation Natural Science Foundation of China (Grants No: 61573270, 61363009 and 61672177), the China 973 Program (Grant No: 2013CB329404), the China Key Research Program (Grant No: 2016YFB1000905), the Guangxi Natural Science Foundation (Grant No: 2015GXNSFBC139011), the Innovation Project of Guangxi Graduate Education (YCSW2017039), the Guangxi Bagui Teams for Innovation and Research, the Guangxi High Institutions Program of Introducing 100 High-Level Overseas Talents, the China 1000-Plan National Distinguished Professorship, the Guangxi Collaborative Innovation Center of Multi-Source Information Integration and Intelligent Processing, and the Guangxi High Institutions Program of Introducing 100 High-Level Overseas Talents, the China 1000-Plan National Distinguished Professorship.

References


[Gao et al., 2012] Yue Gao, Meng Wang, Dacheng Tao, Rongrong Ji, and Qionghai Dai. 3d object retrieval and


