Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16)

Consensus Guided Unsupervised Feature Selection

Hongfu Liu[1], **Ming Shao**[1], **Yun Fu**[1,2]
[1]Department of Electrical and Computer Engineering, Northeastern University, Boston
[2]College of Computer and Information Science, Northeastern University, Boston
liu.hongf@husky.neu.edu, mingshao@ece.neu.edu, yunfu@ece.neu.edu

Abstract

Feature selection has been widely recognized as one of the key problems in data mining and machine learning community, especially for high-dimensional data with redundant information, partial noises and outliers. Recently, unsupervised feature selection attracts substantial research attentions since data acquisition is rather cheap today but labeling work is still expensive and time consuming. This is specifically useful for effective feature selection of clustering tasks. Recent works using sparse projection with pre-learned pseudo labels achieve appealing results; however, they generate pseudo labels with all features so that noisy and ineffective features degrade the cluster structure and further harm the performance of feature selection; besides, these methods suffer from complex composition of multiple constraints and computational inefficiency, e.g., eigen-decomposition. Differently, in this work we introduce consensus clustering for pseudo labeling, which gets rid of expensive eigen-decomposition and provides better clustering accuracy with high robustness. In addition, complex constraints such as non-negative are removed due to the crisp indicators of consensus clustering. Specifically, we propose one efficient formulation for our unsupervised feature selection by using the utility function and provide theoretical analysis on optimization rules and model convergence. Extensive experiments on several real-world data sets demonstrate that our methods are superior to the most recent state-of-the-art works in terms of NMI.

Introduction

Nowadays cutting edge technologies for data mining manage to tackle two major problems with the exponential growth in harvest data: high dimensionality (Agrawal et al. 1998), huge data size (Fayyad, Piatetsky-Shapiro, and Smyth 1996). High dimensional data have never been common than today in many areas since the entity itself includes rich contents that can not be easily abstracted by machines via algorithms: text, images, videos, etc. On one hand, researchers attempt to extract as many features as possible to assist learning algorithms for better performance; on the other hand, rich features lead to redundant information, noisy portion, and outlier samples. In the worst case, an expansion of feature size will take exponentially more computing resource but only provide limited performance improvement.

Feature selection manages to tackle the problem above by selecting the pivot portion of feature, which has been widely discussed in machine learning and data mining community (Guyon and Elisseeff 2003; Li and Fu 2015). Clearly, features after selection are easily interpreted, need shorter training time, and most importantly overcome the over fitting problem. A straightforward way is to enumerate all different feature subsets and evaluate each by certain metric or scores. Such exhaust search is usually computational intractable but for small feature sets. When it comes to unsupervised tasks, feature selection becomes more challenging.

Many algorithms have been proposed to effectively solve the feature selection problem, which can be briefly categorized into three groups according to the accessibility to labels: supervised feature selection, semi-supervised feature selection and unsupervised feature selection.

The first category utilizes label information to find the feature relevance or mutual information, and ensures that the discriminative knowledge is correctly encoded by the selected features (Wolf and Shashua 2005; Theodoridis and Koutroumbas 2008; Nie et al. 2010). Such strategy is usually very effective, but needs costly label information, which is not always available in mining task. Therefore, unsupervised feature selection is demanded for the scenario without any label information but the underlying distribution of the data (Wolf and Shashua 2005; Dy and Brodley 2004; Zhao and Liu 2007; Alelyani, Tang, and Liu 2013). However, due to the absence of labels, it is usually hard to determine the quality of features directly. To this end, unsupervised feature selection approaches usually proceed with certain evaluation criteria such as: Laplacian Score (He, Cai, and Niyogi 2005), mutual information (Peng, Long, and Ding 2005), and maximum likelihood (Dy and Brodley 2004), or generate pseudo labels to apply supervised feature selection (Cai, Zhang, and He 2010; Li et al. 2013).

Recently, some achievements have been witnessed in Unsupervised Feature Selection scenario, which use sparse projection with pseudo labels to guide the process of feature selection. However, they generate pseudo labels with all features so that noisy and ineffective features degrade pseudo labels and further harm the performance of feature selection (Liu and Tao 2015); besides, these methods suffer from

complex composition of multiple constraints and computational inefficiency. In light of this, we propose a novel unsupervised feature selection methods based on consensus clustering, called Consensus Guided Unsupervised Feature Selection (CGUFS), which jointly learns pseudo labels and sparse feature projection in an efficient manner via a step-one framework. Three main contributions are highlighted as follows.

- Consensus clustering is used for pseudo label learning and solve it in linear time, which not only improves the clustering accuracy, but also substantially reduces the running time compared those exploiting spectral methods.
- We get rid of complex compositions of different constraints such as non-negative by the built in crisp indicator of our model, which reduces the model complexity. UFS with utility function is implemented within our framework, and theoretical analyses on efficient optimization rules and convergence are given.
- Extensive experiments on several data sets demonstrate that our method is not only superior to the most state-of-the-art works in terms of clustering quality, but also robust to the model parameters.

Related Work

Here we introduce the related work in terms of unsupervised feature selection and consensus clustering.

As mentioned before, the main challenge for unsupervised feature selection is to find appropriate evaluation criteria or pseudo labels. Such evaluations criteria can be utilized to explore feature relevance, or intrinsic data structure, and the learned pseudo labels can guide the feature selection in a supervised fashion. In the following part, we introduce unsupervised feature selection methods in three categories: filter, wrapper, embedded approaches (Guyon and Elisseeff 2003; Liu and Yu 2005).

Filter algorithms take advantage of proxy measure to score the feature set rather than the commonly used error rate (Zhao and Liu 2007; He, Cai, and Niyogi 2005; Peng, Long, and Ding 2005; Mitra, Murthy, and Pal 2002; Cheung and Zeng 2009). However, such measures may easily ignore factors critical to feature selection since the proxy measures usually concentrate on one aspect of the problem. For example, Laplacian Score is able to select features contributing most to the underlying manifold, but does not consider the factors such as mutual information (He, Cai, and Niyogi 2005). Therefore, it is hard to get rid of feature redundancy. On the other hand, methods considering mutual information may not explicitly explore underlying data structure (Peng, Long, and Ding 2005), which is important in clustering tasks. It should be noted that although filter proxy measure may not be related to the target task, it is usually simple and computationally efficient.

Wrapper method works usually associated with learning algorithms, or specifically, predictive model in most cases (Kim, Street, and Menczer 2002). Therefore, such predictive model can guide the feature selection process with certain objective. Since clustering is the core problem in unsupervised learning, a few learning algorithms and criteria such as maximum likelihood, scatter separability, mixture of Gaussian are learned or checked for the purpose of feature selection and clustering. In (Wolf and Shashua 2005; Dy and Brodley 2004; Zeng and Cheung 2011), the selected features firstly train a predictive model and then are evaluated on a fixed validation set. Other criteria, e.g., least square, spectral theory, sparse regularizer, have been recently proposed towards this problem (Wolf and Shashua 2005; Cai, Zhang, and He 2010).

Embedded methods incorporate the unsupervised feature selection as part of the learning objective (Law, Jain, and Figueiredo 2002; Constantinopoulos, Titsias, and Likas 2006; Shao et al. 2015). For example, feature selection is considered as the missing variable of a probabilistic model jointly learning both clusters, and feature sets (Law, Jain, and Figueiredo 2002). Recently, sparsity has drawn considerable attentions in feature selection and classification tasks (Lee et al. 2006; Wright et al. 2009). A highly sparse vector can be approximately seen as a feature selection process, where redundant or irrelevant features are shrunk via l_1 or $l_{2,1}$ norm. In addition, spectral analysis has been widely adopted in unsupervised feature selection for pseudo label/indicator learning due to its global solution to graph partition problems (Li et al. 2013; Yang et al. 2011; Li et al. 2012).

Another related area is consensus clustering, also known as ensemble clustering, which fuses several basic partitions into an integrated one. As a pioneering work, (Strehl and Ghosh 2003) transformed this problem into a hypergraph partition problem. Followed by this topic, some graph-based methods (Fern and Brodley 2004), co-association matrix based (Fred and Jain 2005) and K-means-based methods (Topchy, Jain, and Punch 2003; Wu et al. 2015; Liu et al. 2015a; 2015b) are proposed to fuse these basic partitions via different optimization objectives in an efficient way. To our best knowledge, we are probably the first to apply consensus clustering to obtain a robust and clean pseudo labels for getting rid of the irrelevant and reductant features.

The Proposed Framework

In this section, some notations used throughout this paper are firstly illustrated; then some preliminary knowledge about consensus clustering and sparse learning for feature selection is showcased; finally we demonstrate our framework of Consensus Guided Unsupervised Feature Selection and give the corresponding objective function.

Notations

Bold uppercase and lowercase characters are used to denote matrices and vectors, respectively. For an arbitrary matrix $\mathbf{A} \in \mathcal{R}^{n\times m}$, $\mathbf{a}_i$ denotes the i-th row of $\mathbf{A}$, A_{ij} denotes the (i,j)-th element of $\mathbf{A}$, $||\mathbf{A}||_{\mathrm{F}} = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{m} A_{ij}^2}$ is well-known Frobenius norm of $\mathbf{A}$ and $\mathrm{tr}(\mathbf{A})$ is the trace of a squared matrix, and its $\ell_{2,1}$-norm is defined as $||\mathbf{A}||_{2,1} = \sum_{i=1}^{n}\sqrt{\sum_{j=1}^{m} A_{ij}^2}$. Let $\mathbf{X} \in \mathcal{R}^{n\times m}$ represent the data matrix with n instances and m features. A partition of these instances into K disjointed subsets can be represented as an

indicator matrix $\mathbf{H} \in \{0,1\}^{n\times K}$, where $\mathbf{h}_{ij} = 1$ represents that $\mathbf{x}_i$ belongs to the j-th cluster.

Consensus Clustering

Consensus clustering aims to fuse several existing partitions into the integrated one. Let $\mathcal{H} = \{\mathbf{H}^{(1)}, \mathbf{H}^{(2)}, \cdots, \mathbf{H}^{(r)}\}$ of $\mathbf{X}$ be the set of r basic partitions and each basic partition $\mathbf{H}^{(i)}$ contains K_i clusters, the goal consensus clustering is to obtain a consensus partition $\mathbf{H}^*$ from $\mathcal{H}$. Here we focus on the consensus clustering with utility function, which has the following formulation.

$$\mathbf{H}^* = \arg\max_{\mathbf{H}^*} \sum_{i=1}^{r} U(\mathbf{H}^*, \mathbf{H}^{(i)}), \tag{1}$$

where U is a *utility function* measuring the similarity of two clustering results in the partition-level.

For better understanding the utility function, A *contingency matrix* is often used for calculating the similarity of two partitions. Let $p_{kj}^{(i)}$ denote the joint probability of one instance simultaneously belongs to cluster $C_j^{(i)}$ in $\mathbf{H}^{(i)}$ and cluster C_k in $\mathbf{H}^*$, and $p_{k+}, p_{+j}^{(i)}$ denote the cluster size portion of $\mathbf{H}^*$ and $\mathbf{H}^{(i)}$, respectively. Based on the above notations, we have the computation of the widely used Category Utility Function (Mirkin 2001) as follows:

$$U_c(\mathbf{H}^*, \mathbf{H}^{(i)}) = \sum_{k=1}^{K} p_{k+} \sum_{j=1}^{K_i} (p_{kj}^{(i)}/p_{k+})^2 - \sum_{j=1}^{K_i} (p_{+j}^{(i)})^2. \tag{2}$$

Although consensus clustering outperforms traditional clustering methods in terms of effectiveness and robustness, it suffers from high computational cost. In the following sections, we apply fast optimization methods to update $\mathbf{H}^*$ in the framework of unsupervised feature selection.

Sparse Learning Analysis

In the scenario of unsupervised feature selection, discriminative features are selected according to pseudo labels. Let $\mathbf{Z} \in \mathcal{R}^{n\times K}$ denote the feature selection matrix, which plays a key role in mapping the original features to the cluster indicator. Therefore, we calculate the loss function between selected features and pseudo labels and add a regularization function with sparsity; by this means, the sparse learning is formulated as:

$$||\mathbf{XZ} - \mathbf{H}^*||_{\mathrm{F}}^2 + \beta||\mathbf{Z}||_{2,1}. \tag{3}$$

Here $\ell_{2,1}$-norm regularization is adopted on $\mathbf{Z}$ to guarantee that $\mathbf{Z}$ is sparse in rows and β is the trade-off parameter. The joint minimization of the loss function and $\ell_{2,1}$-norm regularization makes $\mathbf{Z}$ serve as a bridge between the selected features and pseudo labels. Specially speaking, $\mathbf{z}_i$, the i-th row of $\mathbf{Z}$ shrinks to zeros if the i-th feature contributes little to predicting pseudo labels. Once $\mathbf{Z}$ is learned, top p features are selected by sorting $||\mathbf{z}_i||_2$ in descending order.

Framework Formulation

As we know, there exist many noisy or ineffective features especially in high dimensional data. A lot of existing work utilizes pseudo-labels to select the discriminative features in the unsupervised setting, which makes the selected features contain the structural information as much as possible. One drawback of these methods is that they employ all features to generate the pseudo-labels. This means the pseudo-labels might be inconsistent with the true cluster structure due to irrelevant and ineffective features.

Inspired by the huge success of ensemble clustering, we propose the framework of unsupervised feature selection with consensus guidance to overcome the above limitation. Generally speaking, robust consensus labels are derived from multiple basic partitions to guide the feature selection process. By this means, the features which are most related to the pseudo class labels are selected. In this framework, we simultaneously explore the consensus cluster structure and select informative features in a one-step framework. Here we give the objective function of our unsupervised feature selection with consensus guidance.

$$\min_{\mathbf{H}^*,\mathbf{G},\mathbf{Z}} \alpha\mathcal{J}(\mathbf{H}^*, \mathcal{H}) + ||\mathbf{XZ} - \mathbf{H}^*\mathbf{G}||_{\mathrm{F}}^2 + \beta||\mathbf{Z}||_{2,1}, \tag{4}$$

where $\mathcal{J}(\cdot)$ is the cost of consensus clustering, $\mathbf{G}$ is a $K \times K$ alignment matrix between $\mathbf{XZ}$ and $\mathbf{H}^*$; α and β are two trade-off parameters to control the importance of consensus clustering and sparse learning, respectively.

Our objective function consists of three parts, the first term is for consensus clustering, and the second term is for feature selection and the last one is a regularizer. Compared with the existing unsupervised feature selection methods, our framework has two major differences. One is that we make use of consensus clustering to learn the pseudo class labels, which are more effective to capture the true cluster structure than tradition single clustering algorithms. Another is that a crisp indicator matrix $\mathbf{H}^*$ is used to represent the cluster structure instead of a scaled indicator matrix, which might contain the mixed signs and make itself severely deviate from the ideal cluster indicator. Since clustering is an orderless process, an alignment matrix $\mathbf{G}$ is introduced to shuffle the order of $\mathbf{H}^*$.

UFS with Consensus Guidance

Based on the framework of CGUFS, we give the concrete objective function and corresponding solutions in this section. According to Eq. 4 and Eq. 1, we have

$$\min_{\mathbf{H}^*,\mathbf{G},\mathbf{Z}} -\alpha \sum_{i=1}^{r} U_c(\mathbf{H}^*, \mathbf{H}^{(i)}) + ||\mathbf{XZ} - \mathbf{H}^*\mathbf{G}||_{\mathrm{F}}^2 + \beta||\mathbf{Z}||_{2,1}, \tag{5}$$

The optimization problem of Eq. 5 involves non-matrix variables and non-smooth derivative, which is difficult to solve. Consequently, we propose a non-trivial iterative algorithm to update $\mathbf{H}^*$ and $\mathbf{Z}$ in an efficient way.

Update $\mathbf{H}^*$, G As Given Z

We can see that $\mathbf{H}^*$ exists in the first and second term in Eq. 5. Besides $\mathbf{H}^*$ in the first term is not represented in the matrix formulation, which makes it difficult to optimize. To better understand the consensus part, we introduce the following Theorem 1 to give another interpretation for $\mathbf{H}^*$.

Theorem 1. *Let* $\boldsymbol{B} = [\boldsymbol{H}^{(1)}, \boldsymbol{H}^{(2)}, \cdots, \boldsymbol{H}^{(r)}]$ *be a matrix concatenating all basic partitions, we have*

$$\sum_{i=1}^{r} U_c(\boldsymbol{H}^*, \boldsymbol{H}^{(i)}) = -||\boldsymbol{B} - \boldsymbol{H}^*\boldsymbol{C}||_{\mathrm{F}}^2 + constant, \quad (6)$$

where $\boldsymbol{C} = [\boldsymbol{C}^{(1)}, \boldsymbol{C}^{(2)}, \cdots, \boldsymbol{C}^{(r)}]$ *is the centroid of* $\boldsymbol{B}$.

The proof of Theorem 1 can be found in (Wu et al. 2013; 2015)[1].

Remark 1. *Theorem 1 uncovers the equivalent relationship between consensus clustering with Category Utility function and K-means clustering. Recall that only one element in each row of* $\boldsymbol{H}^{(i)}$ *is 1, others are all 0. According to this property, we even do not build the concatenating* $\boldsymbol{B}$ *in memory, instead we use the indexes of positions with 1. As a result, the time complexity drops to* $O(IKnr)$ *and the space complexity drops to* $O(nr)$ *as well.*

Remark 2. *Theorem 1 also gives a new insight into the objective function of Eq. 5, which can be rewritten as:*

$$\min_{\boldsymbol{H}^*, \boldsymbol{C}, \boldsymbol{G}, \boldsymbol{Z}} \alpha||\boldsymbol{B} - \boldsymbol{H}^*\boldsymbol{C}||_{\mathrm{F}}^2 + ||\boldsymbol{X}\boldsymbol{Z} - \boldsymbol{H}^*\boldsymbol{G}||_{\mathrm{F}}^2 + \beta||\boldsymbol{Z}||_{2,1} \quad (7)$$

Other than utility function to measure the similarity between two partitions, we can also employ the distance to calculate the disagreement between them. The benefits lies in two folds: one is that the objective function has a more concise formulation to understand the essence of the consensus clustering, the other is that the optimization problem is rewritten in a matrix formulation.

After the transform of Theorem 1, it is easy to find that the first two terms are very similar to the standard K-means clustering. Therefore, we wonder that if we can update $\mathbf{H}^*$ and $\mathbf{G}$ within a K-mean-like optimization framework as well.

In the following we introduce a concentrating matrix $\mathbf{U} = [\sqrt{\alpha}\mathbf{B}\ \mathbf{X}\mathbf{Z}]$, where $\mathbf{u}_l$ denote l-th row of $\mathbf{U}$, which consists of two parts, one is the first $\sum_{i=1}^{r} K_i$ columns for the basic partition $\mathbf{u}_l^{(1)} = \langle u_{l,1}, \cdots, u_{l,\sum_{i=1}^{r} K_i} \rangle$; the other last K columns $\mathbf{u}_l^{(2)} = \langle u_{l,R+1}, \cdots, u_{l,R+K} \rangle$ denotes the selected features. Then we have the following Theorem for updating $\mathbf{H}^*$, $\mathbf{C}$ and $\mathbf{G}$.

Theorem 2. *Given the concatenating matrix* $\boldsymbol{U}$*, we have*

$$\begin{aligned} &\min_{\boldsymbol{H}^*, \boldsymbol{C}, \boldsymbol{G}} \alpha||\boldsymbol{B} - \boldsymbol{H}^*\boldsymbol{C}||_{\mathrm{F}}^2 + ||\boldsymbol{X}\boldsymbol{Z} - \boldsymbol{H}^*\boldsymbol{G}||_F^2 \\ &\Leftrightarrow \min_{\boldsymbol{H}^*} \sum_{k=1}^{K} \sum_{\boldsymbol{u}_l \in \mathcal{C}_k} f(\boldsymbol{u}_l, \boldsymbol{m}_k), \end{aligned} \quad (8)$$

where f *is the squared Euclidean distance and* $\boldsymbol{m}_k$ *is the* k*-th centroid of the concatenating matrix* $\boldsymbol{U}$.

Remark 3. *Theorem 2 gives a neat mathematical way to update* $\boldsymbol{H}^*, \boldsymbol{C}$ *and* $\boldsymbol{G}$ *in a K-means optimization framework, which can be solved in roughly linear time complexity.* $\boldsymbol{G}$ *is the last* K *columns of the centroid of* $\boldsymbol{U}$.

By Theorem 1 and Theorem 2, we can update $\mathbf{H}^*$ and $\mathbf{G}$ by just running K-means on the the concentrating matrix $\mathbf{D}$.

[1]The proof of other theorems can be found at http://www.northeastern.edu/smilelab/.

Algorithm 1 UFS with Utility Function

Require: $\mathbf{X}$: the data matrix;
$\mathbf{B}$: the matrix concatenating of r basic partitions;
α, β: the trade-off parameters.

1: Initialize $\mathbf{H}^*$, $\mathbf{Z}$ and $\mathbf{F}$;
2: **repeat**
3: Build the matrix $\mathbf{U} = \left[\sqrt{\alpha}\mathbf{B}\ \mathbf{X}\mathbf{Z}\right]$;
4: Run K-means on $\mathbf{U}$ to update $\mathbf{H}^*$ and $\mathbf{G}$;
5: Update $\mathbf{Z} = (\mathbf{X}^\top\mathbf{X} + \beta\mathbf{F})^{-1}\mathbf{X}^\top\mathbf{H}^*\mathbf{G}$;
6: Update $\mathbf{F}$ according to $\mathbf{Z}$;
7: **until** The objective function value of Eq. 5 remains unchanged.

Ensure: Sort all m features according to $||\mathbf{z}_i||_2$ and select the certain number of ranked features with large values.

Update Z As Given H*, G

Let $\mathcal{L} = ||\mathbf{X}\mathbf{Z} - \mathbf{H}^*\mathbf{G}||_{\mathrm{F}}^2 + \beta||\mathbf{Z}||_{2,1}$, which is only related to $\mathbf{Z}$. Next we take the derivative of $\mathcal{L}$ over $\mathbf{Z}$, and have

$$\frac{\partial \mathcal{L}}{\partial \mathbf{Z}} = 2\mathbf{X}^\top(\mathbf{X}\mathbf{Z} - \mathbf{H}^*\mathbf{G}) + \beta\mathbf{F}\mathbf{Z}, \quad (9)$$

where $\mathbf{F}$ is $\mathrm{diag}(\frac{1}{2||\mathbf{z}_1||_2}, \cdots, \frac{1}{2||\mathbf{z}_m||_2})$. When $\frac{\partial \mathcal{L}}{\partial \mathbf{Z}} = 0$, we have the update rule for $\mathbf{Z}$.

$$\mathbf{Z} = (\mathbf{X}^\top\mathbf{X} + \beta\mathbf{F})^{-1}\mathbf{X}^\top\mathbf{H}^*\mathbf{G}. \quad (10)$$

Finally, we summarize the proposed iterative algorithm to solve Eq. 5 in Algorithm 1.

Convergence Analysis

In the above solution, we decompose the optimization problem in Eq. 5 into two sub problems. Here we show that the proposed iterative procedure can converge to the local minimum by the following Theorem 3.

Theorem 3. *The objective function value of Eq. 5 continuously decreases by the alternative updating rules in Algorithm 1.*

Computational Complexity Analysis

Next we analyse the computational complexity of CGUFS. When updating $\mathbf{H}^*$ and $\mathbf{G}$, the time complexity is $\mathcal{O}(In(R+K)K)$, where I is the number of iteration for K-means and $R = \sum_{i=1}^{r} K_i$. Recall that $K \ll n$, therefore the time complexity during this sub problem is roughly linear to n. During the process of updating $\mathbf{Z}$, it takes $\mathcal{O}(m^3)$ for the matrix inverse. Thus, the overall cost is $\mathcal{O}(tn + tm^3)$, where t is the iteration number for the whole algorithm.

Experimental Results

Here we first evaluate the performance of CGUFS on several public data sets compared with several state-of-the-art methods. Then convergence study and parameter analysis are given to show the robustness of our algorithms. Finally we analyse the impact of different strategies of basic partitions to CGUFS.

Table 1: Experimental Data Sets.

Domain	Data set	#instance	#feature	#class
Image	Coil20	1440	1024	20
	MNIST	4000	784	10
	ORL	400	1024	40
	Yale	165	1024	15
Text	tr11	414	6429	9
	tr41	878	7454	10
	oh15	913	3100	10
	re1	1657	3758	25

Experimental Setup

Data sets. Eight public data sets are used to evaluate the performance of CGUFS including 4 image data sets and 4 text data sets. Table 1 summarizes some important characteristics of these 8 benchmark data sets.

Comparative algorithms. In the unsupervised feature selection scenario, six competitive methods are chosen for comparisons including **Baseline**, which means that all features are used for clustering, **MaxVar**, in which features with maximum variance are selected for clustering, **LS** (He, Cai, and Niyogi 2005) in which features are selected with the most consistency with Gaussian Laplacian matrix, **MCFS** (Cai, Zhang, and He 2010) which selects features based on spectral analysis and spare regression, **UDFS** (Yang et al. 2011) which selects features in a joint framework of discriminative analysis and $\ell_{2,1}$-norm regularization and **NDFS** (Li et al. 2012) in which nonnegative spectral analysis and $\ell_{2,1}$-norm regularization are used for selecting features.

Parameter setting. For LS, MCFA and NDFS, the number of neighbors is set to be 5 for the Laplacian graph. In the CGUFS framework, we employ Random Parameter Selection strategy to generate basic partitions. Generally speaking, *k-means* is conducted on all features with different cluster numbers from K to $\sqrt{n}$; for *ORL* and *Yale* data sets, the range of the numbers of clusters varies in $[2, 2K]$, where K is the true cluster number. 100 basic partitions are produced for robustness. Here we set the cluster structural parameter $\alpha = 10^4$ and the sparse regularization parameter $\beta = 1$. The numbers of selected features vary from 50 to 300 with 50 as the interval, then *k-means* is used on the selected features to validate the performance. Each algorithm runs 20 times, and the average result and standard deviation are reported.

Validation metric. Since we evaluate the feature selection performance via the clustering and all data sets provide the true labels, normalized mutual information (NMI) is used as an external metric for cluster validity. Note that NMI is a positive measure between 0 and 1, and the larger the better.

Performance Comparison

Here we empirically evaluate the performance of CGUFS and other comparative algorithms. Table 2 demonstrates the best results of each algorithm on different numbers of selected features from 50 to 300. The best results are highlighted in bold and the second best results are represented in italic. From this table, we have the following observations. (1) Generally speaking, compared with the baseline method, feature selection is effective by getting rid of many noisy, redundant and non-informative features. Even though there exists a little gap between the performance with feature selection or not on *COIL20* and *MNIST*, it is still appealing to achieve the matchable results with only small partial of the features. (2) CGUFS methods achieve better performance on these 8 data sets than the one of others in most scenarios, which indicates the effectiveness of the consensus guide for unsupervised feature selection. In UFS, pseudo labels are used to guide the process of feature selection; thus the quality of pseudo labels have great impact on the final results. For the comparative methods, all features are employed for the pseudo labels; since there might exist noisy or irrelevant features, the pseudo labels generated from all features might have large difference from the true cluster structure. Thanks to the robustness of consensus clustering, more accuracy labels are used to lead a better feature subset. For instance, CGUFS outperforms the best comparative method by almost 10% on *Yale* and *tr11*. Besides, our model gives the high level guide by a crisp partition, rather than the mapping matrix used in (Cai, Zhang, and He 2010; Yang et al. 2011; Li et al. 2012). It indicates we can get rid of complex compositions of different constraints such as non-negative by the built in crisp indicator of our model, which reduces the model complexity. (3) On these 8 data sets, CGUFS obtains the best results 6 times and the second best 1 time; for the rest data set, CGUFS also achieves the competitive performance.

Next we show the concrete performance of these algorithms on different selected numbers in Figure 1. On *Yale*, CGUFS outperforms other methods with all different numbers of selected features by a large margin and UDFS provides unstable results on these three data sets. It is worthy to note that CGUFS enjoys higher performance only with 50 features, which is preferred by the real-world applications to achieve satisfactory results with as few as features.

Convergence Study

The convergence of CGUFS has been proven in the previous section, here we experimentally study the speed of convergence of CGUFS. Figure 2 shows the convergence curves of *COIL20* and *MNIST*. We can see that CGUFS converges fast within 10 iterations, which demonstrates high quality pseudo labels generated from consensus clustering are conductive to the convergence speed of CGUFS.

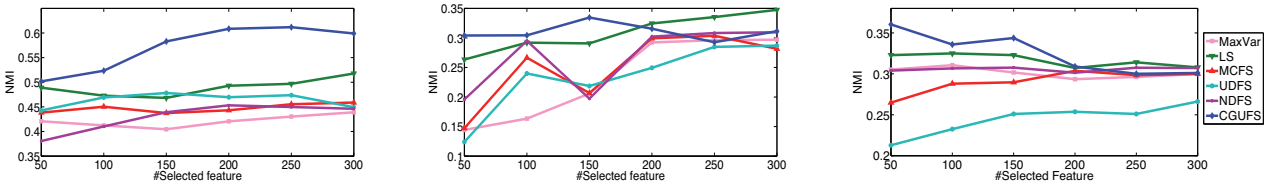
Parameter Sensitivity

In the following we explore the impact of these two parameters on the final performance. Generally speaking, we vary α and β as from $1e-4$ to $1e+4$. The results on *ORL* and *oh15* are shown in Figure 3.

In our CGUFS model, α controls the quality of consensus clustering and β determines the degree of sparseness during the feature selection. Intuitively, we expect a high quality $\mathbf{H}^*$ and a sparse $\mathbf{Z}$. Figure 3 validates our conjecture. In general, the performance of CGUFS has been improved with increase of α and β. A larger α leads to a high quality of $\mathbf{H}^*$, which contributes to the final feature selection; with the increase of

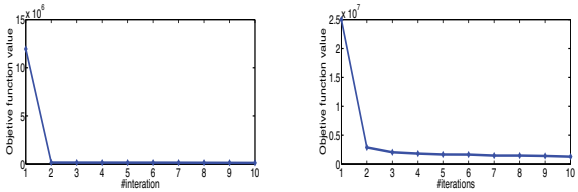
Table 2: Performance of different algorithms on real-world data sets via NMI.

Data set	Baseline	MaxVar	LS	MCFS	UDFS	NUFS	CGUFS
COIL20	0.7675 ± 0.0219	0.7247 ± 0.0146	0.7156 ± 0.0148	*0.7461 ± 0.0181*	0.7328 ± 0.0208	0.7305 ± 0.0213	**0.7555 ± 0.0191**
MNIST	0.4544 ± 0.0143	*0.4683 ± 0.0096*	0.4389 ± 0.0184	0.4677 ± 0.0186	0.4090 ± 0.0105	**0.4686 ± 0.0113**	0.4665 ± 0.0162
ORL	0.4078 ± 0.0150	0.7349 ± 0.0083	0.7344 ± 0.0116	*0.7653 ± 0.0165*	0.7277 ± 0.0121	0.7352 ± 0.0154	**0.7889 ± 0.0176**
Yale	0.5126 ± 0.0338	0.4390 ± 0.0147	*0.5177 ± 0.0173*	0.4588 ± 0.0240	0.4782 ± 0.0161	0.4531 ± 0.0125	**0.6118 ± 0.0399**
tr11	0.0703 ± 0.0141	0.0764 ± 0.0112	0.0803 ± 0.0205	0.1354 ± 0.0332	0.1375 ± 0.0192	*0.1426 ± 0.0301*	**0.2514 ± 0.0312**
tr41	0.2585 ± 0.0903	0.2970 ± 0.0456	**0.3475 ± 0.0519**	30.36 ± 0.0414	0.2870 ± 0.0410	0.3096 ± 0.0162	*0.3345 ± 0.0371*
oh15	0.1770 ± 0.0301	0.2219 ± 0.0275	0.2324 ± 0.0282	0.2312 ± 0.0305	0.2181 ± 0.0204	*0.2423 ± 0.0274*	**0.2911 ± 0.0192**
re1	0.2914 ± 0.0105	0.3106 ± 0.0187	*0.3250 ± 0.0175*	0.3037 ± 0.0177	0.2662 ± 0.0113	0.3076 ± 0.0138	**0.3604 ± 0.0115**



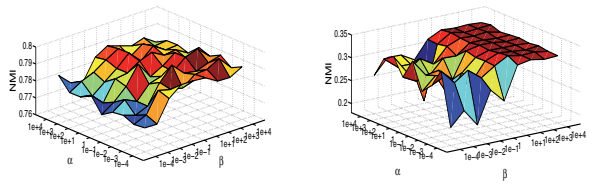
(a) *Yale* (b) *tr41* (c) *re1*

Figure 1: Clustering performance with different selected features.



(a) *COIL20* (b) *MNIST*

Figure 2: Convergence curve of CGUFS.



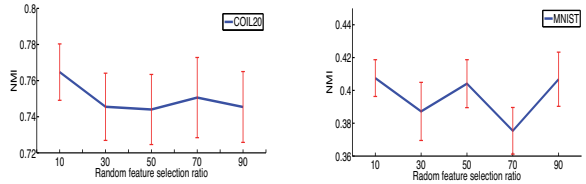
(a) *ORL* (b) *oh15*

Figure 3: Clustering performance with different α and β.

β, we get much better performance on *ORL* and *oh15*, which validates the capability of $\ell_{2,1}$-norm for feature selection. It is worthy to note that when β is larger than 1, the results become stable and insensitive to α.

Impact of Generation Strategy of Basic Partitions

Basic partitions have great impact on the pseudo labels, and they also play important roles to unsupervised feature selection. So far, we use random parameter selection strategy to generation basic partitions. In this subsection, we explore another generation strategy of basic partitions, called random feature selection. Generally speaking, each time we only randomly select partial features to produce the basic



(a) *COIL20* (b) *MNIST*

Figure 4: Random feature selection on *COIL20* and *MNIST*.

partitions, and this process is repeated $r = 100$ times. Here we vary the ratio of random feature selection from 10% to 90% with 20% as the interval. Figure 4 shows the performance with random feature selection on *COIL20* and *MNSIT*.

As can be seen in Figure 4, the consensus guided unsupervised feature selection achieves stable results with the large range of the feature selection ratio. Even with 10% features, CGUFS can also obtain satisfactory results, which demonstrates the robustness of CGUFS. Compared with the results in Table 2, we can find that CGUFS with random feature selection has the comparative results to the one with random parameter selection on *COIL20*; however, on *MNIST* there exists a large gap between these two kinds of generation strategies for basic partitions. We will further explore the impact of basic partitions on feature selection in the future.

Conclusions

In this paper, we employed the consensus pseudo labels to guide the unsupervised feature selection process and proposed the consensus guided unsupervised feature selection framework. Generally speaking, one efficient algorithm by using the utility function was proposed and we provided

theoretical analysis on consensus clustering and model convergence. Extensive experiments on 8 widely used data sets demonstrated that our method has significant advantages over the most recent state-of-the-art works in terms of NMI.

Acknowledgments

This research is supported in part by the NSF CNS award 1314484, ONR award N00014-12-1-1028, ONR Young Investigator Award N00014-14-1-0484, NPS award N00244-15-1-0041, and U.S. Army Research Office Young Investigator Award W911NF-14-1-0218.

References

Agrawal, R.; Gehrke, J.; Gunopulos, D.; and Raghavan, P. 1998. Automatic subspace clustering of high dimensional data for data mining applications. In *SIGMOD*.

Alelyani, S.; Tang, J.; and Liu, H. 2013. Feature selection for clustering: A review. *Data Clustering: Algorithms and Applications*.

Cai, D.; Zhang, C.; and He, X. 2010. Unsupervised feature selection for multi-cluster data. In *KDD*.

Cheung, Y.-m., and Zeng, H. 2009. Local kernel regression score for selecting features of high-dimensional data. *IEEE Transactions on Knowledge and Data Engineering*.

Constantinopoulos, C.; Titsias, M. K.; and Likas, A. 2006. Bayesian feature and model selection for gaussian mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Dy, J., and Brodley, C. 2004. Feature selection for unsupervised learning. *Journal of Machine Learning Research*.

Fayyad, U.; Piatetsky-Shapiro, G.; and Smyth, P. 1996. From data mining to knowledge discovery in databases. *AI magazine*.

Fern, and Brodley, C. 2004. A survey of clustering ensemble algorithms. In *ICML*.

Fred, A., and Jain, A. 2005. Combining multiple clusterings using evidence accumulation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Guyon, I., and Elisseeff, A. 2003. An introduction to variable and feature selection. *Journal of Machine Learning Research*.

He, X.; Cai, D.; and Niyogi, P. 2005. Laplacian score for feature selection. In *NIPS*.

Kim, Y.; Street, W. N.; and Menczer, F. 2002. Evolutionary model selection in unsupervised learning. *Intelligent Data Analysis*.

Law, M. H.; Jain, A. K.; and Figueiredo, M. 2002. Feature selection in mixture-based clustering. In *NIPS*.

Lee, H.; Battle, A.; Raina, R.; and Ng, A. Y. 2006. Efficient sparse coding algorithms. In *NIPS*.

Li, S., and Fu, Y. 2015. Learning balanced and unbalanced graphs via low-rank coding. *IEEE Transactions on Knowledge and Data Engineering*.

Li, Z.; Yang, Y.; Liu, J.; Zhou, X.; and Lu, H. 2012. Unsupervised feature selection using nonnegative spectral analysis. In *AAAI*.

Li, Z.; Liu, J.; Yang, Y.; Zhou, X.; and Lu, H. 2013. Clustering-guided sparse structural learning for unsupervised feature selection. *IEEE Transactions on Knowledge Data Engineering*.

Liu, T., and Tao, D. 2015. Classification with noisy labels by importance reweighting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Liu, H., and Yu, L. 2005. Toward integrating feature selection algorithms for classification and clustering. *IEEE Transactions on Knowledge and Data Engineering*.

Liu, H.; Liu, T.; Wu, J.; Tao, D.; and Fu, Y. 2015a. Spectral ensemble clustering. In *KDD*.

Liu, H.; Wu, J.; Tao, D.; Zhang, Y.; and Fu, Y. 2015b. Dias: A disassemble-assemble framework for highly sparse text clustering. In *SDM*.

Mirkin, B. 2001. Reinterpreting the category utility function. *Machine Learning*.

Mitra, P.; Murthy, C.; and Pal, S. K. 2002. Unsupervised feature selection using feature similarity. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Nie, F.; Huang, H.; Cai, X.; and Ding, C. H. 2010. Efficient and robust feature selection via joint l2, 1-norms minimization. In *NIPS*.

Peng, H.; Long, F.; and Ding, C. 2005. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Shao, M.; Li, S.; Ding, Z.; and Fu, Y. 2015. Deep linear coding for fast graph clustering. In *IJCAI*.

Strehl, A., and Ghosh, J. 2003. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research*.

Theodoridis, S., and Koutroumbas, K. 2008. *Pattern Recognition*.

Topchy, A.; Jain, A.; and Punch, W. 2003. Combining multiple weak clusterings. In *ICDM*.

Wolf, L., and Shashua, A. 2005. Feature selection for unsupervised and supervised inference: The emergence of sparsity in a weight-based approach. *Journal of Machine Learning Research*.

Wright, J.; Yang, A.; Ganesh, A.; Sastry, S.; and Ma, Y. 2009. Robust face recognition via sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Wu, J.; Liu, H.; Xiong, H.; and Cao, J. 2013. A theoretic framework of k-means-based consensus clustering. In *IJCAI*.

Wu, J.; Liu, H.; Xiong, H.; Cao, J.; and Chen, J. 2015. K-means-based consensus clustering: A unified view. *IEEE Transactions on Knowledge and Data Engineering*.

Yang, Y.; Shen, H. T.; Ma, Z.; Huang, Z.; and Zhou, X. 2011. l2, 1-norm regularized discriminative feature selection for unsupervised learning. In *IJCAI*.

Zeng, H., and Cheung, Y.-m. 2011. Feature selection and kernel learning for local learning-based clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Zhao, Z., and Liu, H. 2007. Spectral feature selection for supervised and unsupervised learning. In *ICML*.