Exploiting Hierarchical Structures for Unsupervised Feature Selection

Suhang Wang* Yilin Wang* Jiliang Tang† Charu Aggarwal‡ Suhas Ranganath*
Huan Liu*

Abstract

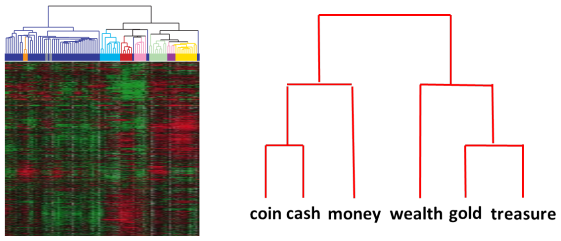
Feature selection has been proven to be effective and efficient in preparing high-dimensional data for many mining and learning tasks. Features of real-world high-dimensional data such as words of documents, pixels of images and genes of microarray data, usually present inherent hierarchical structures. In a hierarchical structure, features could share certain properties. Such information has been exploited to help supervised feature selection but it is rarely investigated for unsupervised feature selection, which is challenging due to the lack of labels. Since real world data is often unlabeled, it is of practical importance to study the problem of feature selection with hierarchical structures in an unsupervised setting. In particular, we provide a principled method to exploit hierarchical structures of features and propose a novel framework HUFS, which utilizes the given hierarchical structures to help select features without labels. Experimental study on real-world datasets is conducted to assess the effectiveness of the proposed framework.

1 Introduction

High-dimensional data is ubiquitous in many data mining and machine learning applications [1, 2]. Data with high dimensionality not only significantly increases the time and memory requirements of many algorithms, but also degenerates algorithms' performance due to the curse of dimensionality and the existence of irrelevant, redundant and noisy dimensions [3]. Feature selection, which aims at reducing the dimensionality by selecting a subset of most relevant features, has been proven to be an effective and efficient way to handle high-dimensional data [4, 3].

In terms of the label availability, feature selection methods can be generally categorized into supervised and unsupervised methods [5, 6]. With class labels, supervised feature selection [7, 8, 9] is able to effectively select discriminative features to distinguish samples from different classes. As most data is unlabeled in many applications and it is very expensive to label data, unsupervised feature selection has attracted increasing attention in recent years [10, 11, 12, 13, 14, 15].

*Arizona State University {suhang.wang, ywang370, srangan8, huan.liu}@asu.edu
†Michigan State University, tangjili@mus.edu
‡IBM T.J. Watson, charu@us.ibm.com



(a) Gene Expression (b) WordNet

Figure 1: Examples of Feature Hierarchical Structures

Features of real-world high-dimensional data, such as words of documents [16], pixels of images [17] and genes of microarray data [18], usually exhibit certain hierarchical structures. For example, Figure 1 gives two examples of hierarchical structures of features. Figure 1(a) illustrates the hierarchical structure of genes from a cancer tumour dataset in [18], where each row is a cell (data instance) and each column is a gene (feature). The genes are hierarchically clustered using their gene expressions and genes in the same subgroup, such as yellow group in Figure 1(a), share similar gene expressions. Figure 1(b) gives an example of the word hierarchical structure constructed by word senses, where words with the same (or close) meanings are grouped together and fine-grained meaning groups are nested under coarse-grained ones. As shown in the Figure 1(b), all the words are in the same coarse group since they are kinds of *asset*. Furthermore, *coin* and *cash*, are further grouped together with *money* as they are kinds of *money*. Obviously, features in the same group share certain properties, functionalities or semantic meanings. These hierarchical structures of features could give important guidance to feature selection. Therefore, recently, there are supervised feature selection algorithms exploiting hierarchical structures to improve performance [17, 19]. However, little work exists for unsupervised feature selection given hierarchical structures of features.

As data is often unlabeled and hierarchical structures can be important, we study the novel problem of utilizing given hierarchical structures for unsupervised feature selection in this paper. Without label information, we need to investigate the following two challenges: (1) how to mathematically model given hierarchical structures? and (2) how to incorporate the hierarchical structures for unsupervised feature selection?

In an attempt to address these two challenges, we propose a novel unsupervised feature selection framework HUFS. It integrates given hierarchical structures of features for feature selection. The major contributions of the paper are :

- A principled approach to exploit hierarchical structures for unsupervised feature selection;
- A novel framework HUFS, which utilizes given hierarchical structures to select features in an unsupervised scenario by modeling HUFS as a non-smooth optimization problem; and
- Extensive experiments on various datasets with demonstration on the effectiveness of HUFS.

2 Related Work

Feature selection is a process of choosing a subset of original features so that the feature space is optimally reduced according to a certain evaluation criterion. It is frequently used as a preprocessing step to machine learning and data mining and has been proven to be an effective and efficient way in reducing dimensionality, removing irrelevant features, increasing learning accuracy, and improving comprehensibility [20]. As unlabeled data is pervasive in many applications and it is very expensive to label data, unsupervised feature selection has attracted increasing attention in recent years [10, 11, 12, 13, 14, 15, 21].

Without label information to define feature relevance, a number of alternative criteria have been proposed for unsupervised feature selection. Similar to supervised feature selection, one commonly used criterion is to select features that can preserve the data similarity or manifold structure. Since no label information are given, data similarity are usually constructed from the whole feature space without label information such as Laplacian Score [11] and Unsupervised SPEC [22]. As the success of sparse learning in supervised feature selection, in recent years, applying sparse learning in unsupervised feature selection has attracted increasing attention. The general idea is to generate pseudo cluster labels via clustering algorithms and then transform unsupervised feature selection into sparse learning based supervised feature selection with these generated cluster labels such as Multi-cluster feature selection (MCFS) [23], Nonnegative Discriminative Feature Selection (NDFS) [13], Robust Unsupervised Feature Selection [14] and EUFS [21].

In real-world, features are usually not independent. For example, in the multi-factor analysis-of-variance (ANOVA) problem, each factor may have several levels and can be denoted as a group of dummy features. Auxiliary information of such feature relations have proven to be effective in improving supervised feature selection performance [24, 17, 19]. For example, Yuan et al [24] studied the non-overlap group lassos to exploit group structures of features. Liu et al [19] proposed weakly hierarchical lasso to study feature interaction. However, little work exists for *unsupervised feature selection* with hierarchical structures. There is a similar effort to impose overlapping group structures on data instances under unsupervised scenario [25], but it is not about hierarchical structures of features; in addition, its purpose is of clustering, while our study is of *unsupervised feature selection*. To the best of our knowledge, we are the first one to study unsupervised feature selection with given hierarchical structures. We propose a novel framework HUFS and an efficient algorithm to solve the non-smooth optimization problem of HUFS.

3 Unsupervised Feature Selection with Hierarchical Structures

In the paper, matrices are written as boldface capital letters and vectors are denoted as boldface lowercase letters. For an arbitrary matrix $\mathbf{M} \in \mathbb{R}^{m\times n}$, $\mathbf{M}_{ij}$ denotes the (i,j)-th entry of $\mathbf{M}$ while $\mathbf{m}_i$ and $\mathbf{m}^j$ mean the i-th row and j-th column of $\mathbf{M}$, respectively. $\|\mathbf{M}\|_F$ is the Frobenius norm of $\mathbf{M}$ and $Tr(\mathbf{M})$ is the trace of a square matrix $\mathbf{M}$. $\langle \mathbf{A}, \mathbf{B}\rangle$ equals to $Tr(\mathbf{A}^T\mathbf{B})$, which is the standard inner product between two matrices. $\mathbf{I}$ is the identity matrix. The $\ell_{2,1}$-norm is defined as $\|\mathbf{M}\|_{2,1} = \sum_{i=1}^{m} \|\mathbf{m}_i\|_2$. Capital letters in the calligraphic font such as $\mathcal{G}$ are used to denote sets.

3.1 A Basic Model for Unsupervised Feature Selection

Sparse learning has been proven to be a very powerful tool for feature selection [23, 26, 14, 27]. One effective approach of applying sparse learning for unsupervised feature selection is to embed feature selection into a clustering algorithm by performing feature selection on the latent features with sparse learning techniques [21]. Following the idea, the basic model for our proposed framework directly embeds feature selection into a low-rank matrix approximation algorithm and performs $\ell_{2,1}$ norm on the latent feature matrix to achieve feature selection. Let $\mathbf{X} \in \mathbb{R}^{N\times m}$ be the data matrix where N is the number of data samples and m is the number of features. The basic model decomposes $\mathbf{X}$ into two low-rank matrices $\mathbf{U} \in \mathbb{R}^{N\times K}$ and $\mathbf{V} \in \mathbb{R}^{m\times K}$, and applies $\ell_{2,1}$ norm on $\mathbf{V}$ as follows:

$$\begin{aligned} &\min_{\mathbf{U},\mathbf{V}} \|\mathbf{X} - \mathbf{U}\mathbf{V}^T\|_{2,1} + \beta\|\mathbf{V}\|_{2,1}, \\ &s.t.\ \mathbf{U}^T\mathbf{U} = \mathbf{I}, \mathbf{U} \geq \mathbf{0} \end{aligned} \tag{3.1}$$

In Eq.(3.1), $\mathbf{U}$ is the cluster affiliation matrix. Nonnegative orthogonal constraints are applied on $\mathbf{U}$ to enforce that each row of $\mathbf{U}$ has one non-zero element as the cluster affiliation. $\mathbf{V}$ is the latent feature matrix. Each feature $\mathbf{x}^i$, i.e., i-th column of $\mathbf{X}$, is reconstructed

as $\mathbf{x}^i \approx \mathbf{U}\mathbf{v}_i^T$. Thus, there's one to one correspondence between the original feature $\mathbf{x}^i$ and the latent feature $\mathbf{v}_i$. A good feature $\mathbf{x}^i$ should be well reconstructed by its latent feature $\mathbf{v}_i$. Therefore, by adding $\ell_{2,1}$ norm on $\mathbf{V}$, it aims to eliminate $\mathbf{v}_i$ that cannot properly reconstruct $\mathbf{x}^i$ [21]. $\ell_{2,1}$-norm on $\mathbf{X}-\mathbf{U}\mathbf{V}^T$ is adopted to avoid reconstructing errors that dominate the objective function. Finally, the importance of the i-th feature is indicated by $\|\mathbf{v}_i\|_2$ – the larger $\|\mathbf{v}_i\|_2$ is, the more important the i-th feature is. This serves as a good basic model for exploiting hierarchical structures because: (1) $\mathbf{v}_i$ and $\mathbf{x}^i$ has one to one correspondence and the quality of $\mathbf{v}_i$ reflects quality of $\mathbf{x}^i$, which allows us to model hierarchical structures on $\mathbf{v}_i$; and (2) the learning of $\mathbf{v}_i$ doesn't need label information.

3.2 Modeling Feature Hierarchical Structures

Features in many real-world applications often present certain inherent hierarchical structures; and features in the same group of the hierarchical structure usually share similar functionalities, properties or semantic meanings, which can provide helpful information for feature selection and have been widely captured under supervised settings [17, 28, 29]. In this subsection, we discuss how to capture hierarchical structures based on the unsupervised basic model shown in Eq.(3.1).

In the basic model, $\mathbf{v}_i$ is used to reconstruct $\mathbf{x}^i$ and they have one-to-one correspondence. Meanwhile, with extra constraints on features, unsupervised feature selection is likely to achieve better performance [30]. Hence, we can model hierarchical structures as constraints on the latent feature matrix $\mathbf{V}$ to guide feature selection. Next we will use the example in Figure 1(b) to demonstrate how to model hierarchical structures. Index tree is a natural way to represent the hierarchical structure of Figure 1(b) [17]. The definition of index tree is given as:

DEFINITION 1. *For an index tree $\mathcal{T}$ of depth d, $\mathcal{G}_t^s, s = 1, \ldots, d, t = 1, \ldots, n_s$, denotes the t-th node in the level s, where n_s is the number of nodes in the s-th level. $\mathcal{G}_1^1 = \{f_1, f_2, \ldots, f_m\}$ is the root node that contains all the features $f_1, \ldots, f_m$. The nodes satisfy the following conditions: 1) the nodes from the same depth level have non-overlapping indices, i.e, $\mathcal{G}_i^s \cap \mathcal{G}_j^s = \emptyset, \forall s = 2, \ldots, d$ and $, i \neq j, 1 \leq i, j \leq n_s$; and (2) let $\mathcal{G}_{j0}^{s-1}$ be the parent node of a non-root node $\mathcal{G}_j^s$, then $\mathcal{G}_j^s \subset \mathcal{G}_{j0}^{s-1}$.*

Figure 2 is an index tree to denote the hierarchical structure in Figure 1(b). In Figure 2, $f_1, \ldots, f_6$ are six features corresponding to the words *coin*, ..., *treasure* in Figure 1(b), respectively. The tree is composed of 5 nodes, i.e., $\mathcal{G}_1^1 = \{f_1, f_2, \ldots, f_6\}$, $\mathcal{G}_1^2 = \{f_1, f_2, f_3\}$, $\mathcal{G}_2^2 = \{f_4, f_5, f_6\}$, $\mathcal{G}_1^3 = \{f_1, f_2\}$ and $\mathcal{G}_2^3 = \{f_5, f_6\}$. Therefore, the hierarchical structure in Figure 1(b) can

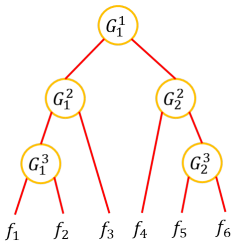


Figure 2: An example of tree guided groups.

be viewed as a tree guided groups where features in same group share similar properties. The reasons why the index tree captures information from hierarchical structure are two-fold. First, each node in the index tree represents a group in the hierarchical structure. For example, $\mathcal{G}_1^2$ represents the group $\{coin, cash, money\}$ and $\mathcal{G}_1^3$ denotes the group $\{coin, cash\}$. Second, in the index tree, the child node is a subset of its parent node because in the hierarchical structure, a small group can nest under a large group.

We note that features in a node of an index tree share similar semantic meanings; while semantic meanings of features in a child node are similar to that of its parent node but is more fine-grained. For example, in $\mathcal{G}_1^2$, the features are $\{coin, cash, money\}$, which share the same semantic meaning *money* and the features in its child node $\mathcal{G}_1^3$ are specific types of money. Thus, each node could guide unsupervised feature selection by providing different semantic meanings. For example, if topics of documents we want to cluster are irrelevant to *money*, we should discard these features in $\mathcal{G}_1^2$ simultaneously , as they are not discriminative; while if one of the topics of the documents is about *money*, all these features in node $\mathcal{G}_1^2$ could be helpful in identifying a cluster of documents about *money*. Note that in the basic model in Eq.(3.1), we cluster the data $\mathbf{X}$ into K clusters. Therefore, we can add constraints to features in each node of the index tree to ensure that all features in the same group (a node of the index tree) are either relevant or irrelevant to one of the K clusters. To achieve this, for a node $\mathcal{G}_t^s$, we use $\mathbf{v}_{\mathcal{G}_t^s}^i$ to denote the sub-vector of $\mathbf{v}^i$ corresponding to the features in $\mathcal{G}_s^t$. For example, if $\mathcal{G}_1^2 = \{f_1, f_2, f_3\}$, then $\|\mathbf{v}_{\mathcal{G}_1^2}^i\|_2 = \|[\mathbf{V}_{1i}, \mathbf{V}_{2i}, \mathbf{V}_{3i}]\|_2$. If features in $\mathcal{G}_t^s$ are irrelevant to the K clusters, we want to make elements in $\mathbf{V}_{\mathcal{G}_t^s} = [\mathbf{v}_{\mathcal{G}_t^s}^1, \ldots, \mathbf{v}_{\mathcal{G}_t^s}^K]$ to be close to zero or exactly zero. In this way, when we select features based the value of $\|\mathbf{v}_p\|_2, p = 1, \ldots, m$, any feature $f_j \in \mathcal{G}_t^s$ with $\|\mathbf{v}_j\|_2 \approx 0$ will be eliminated, which achieves the goal of feature selection with hierarchical structures. To force elements of some $\mathbf{V}_{\mathcal{G}_t^s}$ to be close to zero, we add the constraint as follows:

$$\sum_{i=1}^{K} \|\mathbf{v}_{\mathcal{G}_t^s}^i\|_2. \tag{3.2}$$

The effect of the constraint $\sum_{i=1}^{K} \|\mathbf{v}^i_{\mathcal{G}^s_t}\|_2$ is equivalent to add ℓ_1 norm on the vector $\mathbf{g} = [\|\mathbf{v}^1_{\mathcal{G}^s_t}\|_2, \|\mathbf{v}^2_{\mathcal{G}^s_t}\|_2, \ldots, \|\mathbf{v}^K_{\mathcal{G}^s_t}\|_2]$, i.e., $\|\mathbf{g}\|_1$. It could make the solution of $\mathbf{g}$ sparse; in other words, some elements in $\mathbf{g}$ could be exactly zero. If $\mathbf{g}(i) = 0$ or $\|\mathbf{v}^i_{\mathcal{G}^s_t}\|_2 = 0$, then the effect of all features in $\mathcal{G}^s_t$ on the i-th cluster is eliminated. If all the features in $\mathcal{G}^s_t$ are irrelevant to the K clusters, then all elements in $\mathbf{V}_{G^s_t}$ will be close to 0. Thus, given an index tree $\mathcal{T}$ we propose to minimize the following term to capture the hierarchical structure:

$$\sum_{s=1}^{d}\sum_{t=1}^{n_s}\sum_{i=1}^{K} \|\mathbf{v}^i_{\mathcal{G}^s_t}\|_2 \tag{3.3}$$

3.3 The Proposed Framework – HUFS With the model component to exploit hierarchical structures, the proposed framework HUFS is to solve the following optimization problem:

$$\begin{aligned} &\arg\min_{\mathbf{U},\mathbf{V}} \|\mathbf{X} - \mathbf{U}\mathbf{V}^T\|_{2,1} + \beta\|\mathbf{V}\|_{2,1} + \alpha\sum_{i=1}^{K}\Omega(\mathbf{v}^i) \\ &s.t.\ \ \mathbf{U}^T\mathbf{U} = \mathbf{I}, \mathbf{U} \geq \mathbf{0} \end{aligned} \tag{3.4}$$

In Eq. (3.4), the first and second terms with constraints are from the basic unsupervised feature selection framework in Eq. (3.1); and the third term is to capture hierarchical structures, which is controlled by a parameter α. And $\Omega(\mathbf{v}^i)$ is defined as:

$$\Omega(\mathbf{v}^i) = \sum_{s=1}^{d}\sum_{t=1}^{n_s} \|\mathbf{v}^i_{\mathcal{G}^s_t}\|_2 \tag{3.5}$$

4 An Optimization Method for HUFS

The objective function in Eq.(3.4) is not convex in both $\mathbf{U}$ and $\mathbf{V}$, which makes it difficult to optimize. In addition, the index tree regularizer on $\mathbf{V}$ contains overlapping group structures and is non-smooth, which makes the optimization problem more difficult. Following [31, 32], we use alternating direction method of multiplier (ADMM) [33] to optimize the objective function. ADMM is a popular method for solving non-convex and non-smooth optimization problem. ADMM can break the complicated problem into small sub-problems, each of which is then easier to solve. We next give details of using ADMM. We first introduce three auxiliary variables $\mathbf{E} = \mathbf{X} - \mathbf{U}\mathbf{V}^T$, $\mathbf{Z} = \mathbf{U}$ and $\mathbf{P} = \mathbf{M}\mathbf{V}$, where $\mathbf{M} \in \{0,1\}^{\sum_{s=1}^{d}\sum_{t=1}^{n_s}|\mathcal{G}^s_t| \times m}$ is a sparse matrix whose definition will be given next. With these auxiliary variables, the objective function becomes

$$\begin{aligned} &\arg\min_{\mathbf{U},\mathbf{V},\mathbf{E},\mathbf{Z},\mathbf{P}} \|\mathbf{E}\|_{2,1} + \alpha\sum_{i=1}^{K}\Omega_1(\mathbf{p}^i) + \beta\|\mathbf{V}\|_{2,1} \\ &s.t.\ \ \mathbf{E} = \mathbf{X} - \mathbf{U}\mathbf{V}^T, \mathbf{Z} = \mathbf{U}, \mathbf{P} = \mathbf{M}\mathbf{V}, \mathbf{U}^T\mathbf{U} = \mathbf{I}, \mathbf{Z} \geq \mathbf{0} \end{aligned} \tag{4.6}$$

We will give the definition of $\Omega_1(\mathbf{p}^i)$ shortly. The goal of introducing $\mathbf{M}$ is to ensure that the constraint $\Omega_1(\mathbf{p}^i)$ has the same regularization effects as $\Omega(\mathbf{v}^i)$ but is easier to optimize, i.e., there are no overlapping groups on $\mathbf{p}^i$. To achieve this goal, we allow each row of $\mathbf{M}$ to contain exactly one nonzero element. Specifically, if the k-th element in $\mathcal{G}^p_q \in \mathcal{T}$ is the feature f_i, we set $\mathbf{M}(\sum_{s=1}^{p-1}\sum_{t=1}^{n_s}|\mathcal{G}^s_t| + \sum_{t=1}^{q-1}|\mathcal{G}^p_t| + k, i) = 1$, where $\sum_{s=1}^{p-1}\sum_{t=1}^{n_s}|\mathcal{G}^s_t| + \sum_{t=1}^{q-1}|\mathcal{G}^p_t|$ is the total number of features encoded in nodes from root node $\mathcal{G}^1_1$ to node $\mathcal{G}^p_{q-1}$ and the addition of k is because f_i is the k-th feature in node $\mathcal{G}^p_q$. The effect of $\mathbf{M}\mathbf{v}^i$ is the same as concatenating elements in $\mathbf{v}^i$ by the features encoded in the nodes of the index tree. Let $\mathbf{M}_{|\mathcal{G}^s_t}$ denote the rows of $\mathbf{M}$ corresponding to $\mathcal{G}^s_t$, i.e, the rows that are constructed by features in $\mathcal{G}^s_t$. Then, we have $\mathbf{M}_{|\mathcal{G}^s_t}\mathbf{v}^i = \mathbf{v}^i_{\mathcal{G}^s_t}$. Since we require $\mathbf{p}^i = \mathbf{M}\mathbf{v}^i$, we also have $\mathbf{p}^i_{|\mathcal{G}^s_t} = \mathbf{M}_{|\mathcal{G}^s_t}\mathbf{v}^i$ and $\mathbf{p}^i_{|\mathcal{G}^s_t} = \mathbf{v}^i_{\mathcal{G}^s_t}$. We define $\Omega_1(\mathbf{p}^i)$:

$$\Omega_1(\mathbf{p}^i) = \sum_{s=1}^{d}\sum_{t=1}^{n_s} \|\mathbf{p}^i_{|\mathcal{G}^s_t}\|_2 \tag{4.7}$$

With $\mathbf{M}$ and $\Omega_1(\mathbf{p}^i)$ defined as above, we have

$$\Omega_1(\mathbf{p}^i) = \sum_{s=1}^{d}\sum_{t=1}^{n_s} \|\mathbf{p}^i_{|\mathcal{G}^s_t}\|_2 = \sum_{s=1}^{d}\sum_{t=1}^{n_s} \|\mathbf{v}^i_{\mathcal{G}^s_t}\|_2 = \Omega(\mathbf{v}^i) \tag{4.8}$$

Thus, we have shown that $\Omega_1(\mathbf{p}^i) = \Omega(\mathbf{v}^i)$ and it is obvious that there is no overlapping group structure in $\Omega_1(\mathbf{p}^i)$ because $\mathbf{p}^i_{|\mathcal{G}^s_t}$ and $\mathbf{p}^i_{|\mathcal{G}^p_q}$ do not overlap. With these auxiliary variables, Eq.(4.6) can be solved by the following optimization problem:

$$\begin{aligned} \min_{\mathbf{U},\mathbf{V},\mathbf{E},\mathbf{Z},\mathbf{P}} \ & \|\mathbf{E}\|_{2,1} + \alpha\sum_{i=1}^{K}\Omega_1(\mathbf{p}^i) + \beta\|\mathbf{V}\|_{2,1} \\ & + \langle\mathbf{Y}_1, \mathbf{Z} - \mathbf{U}\rangle + \langle\mathbf{Y}_2, \mathbf{X} - \mathbf{U}\mathbf{V}^T - \mathbf{E}\rangle \\ & + \langle\mathbf{Y}_3, \mathbf{P} - \mathbf{M}\mathbf{V}\rangle + \frac{\mu}{2}\|\mathbf{Z} - \mathbf{U}\|_F^2 \\ & + \frac{\mu}{2}\left(\|\mathbf{X} - \mathbf{U}\mathbf{V}^T - \mathbf{E}\|_F^2 + \|\mathbf{P} - \mathbf{M}\mathbf{V}\|_F^2\right) \\ s.t.\ & \mathbf{U}^T\mathbf{U} = \mathbf{I}, \mathbf{Z} \geq 0 \end{aligned} \tag{4.9}$$

where $\mathbf{Y}_1$, $\mathbf{Y}_2$ and $\mathbf{Y}_3$ are Lagrangian multipliers and μ is to control the penalty for the violation of equality constraints $\mathbf{E} = \mathbf{X} - \mathbf{U}\mathbf{V}^T$, $\mathbf{Z} = \mathbf{U}$ and $\mathbf{P} = \mathbf{V}$.

4.1 Update E To update $\mathbf{E}$, we fix the other variables except $\mathbf{E}$ and remove terms that are irrelevant to $\mathbf{E}$. Then Eq.(4.9) becomes

$$\min_{\mathbf{E}} \frac{1}{2}\|\mathbf{E} - (\mathbf{X} - \mathbf{U}\mathbf{V}^T + \frac{1}{\mu}\mathbf{Y}_2)\|_F^2 + \frac{1}{\mu}\|\mathbf{E}\|_{2,1} \tag{4.10}$$

The equation has a closed form solution by the following Lemma [34]

LEMMA 4.1. *Let* $\mathbf{q}$ *be a given vector and* λ *a positive scalar. If the optimal solution of*

$$\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}-\mathbf{q}\|_2^2 + \lambda\|\mathbf{w}\|_2 \tag{4.11}$$

is $\mathbf{w}^*$*, then* $\mathbf{w}^*$ *is*

$$\mathbf{w}^* = \begin{cases} (1-\frac{\lambda}{\|\mathbf{q}_i\|})\mathbf{q}, & \text{if } \|\mathbf{q}\| > \lambda \\ \mathbf{0}, & \text{otherwise} \end{cases} \tag{4.12}$$

Apparently, if we let $\mathbf{Q} = \mathbf{X} - \mathbf{U}\mathbf{V}^T + \frac{1}{\mu}\mathbf{Y}_2$ and decompose Eq.(4.10) row-wise, then using Lemma 4.1, $\mathbf{E}$ can be updated as

$$\mathbf{e}_i = \begin{cases} (1-\frac{1}{\mu\|\mathbf{q}_i\|})\mathbf{q}_i, & \text{if } \|\mathbf{q}_i\| > \frac{1}{\mu} \\ \mathbf{0}, & \text{otherwise} \end{cases} \tag{4.13}$$

4.2 Update V To update $\mathbf{V}$, we remove terms that are irrelevant to $\mathbf{V}$ and use the fact that $\mathbf{U}^T\mathbf{U} = \mathbf{I}$, Eq.(4.9) becomes

$$\min_{\mathbf{V}} \frac{\mu}{2}\|\mathbf{V}-\mathbf{K}\|_F^2 + \frac{\mu}{2}\|\mathbf{M}\mathbf{V}-\mathbf{H}\|_F^2 + \beta\|\mathbf{V}\|_{2,1} \tag{4.14}$$

where $\mathbf{K} = (\mathbf{X}-\mathbf{E}+\frac{1}{\mu}\mathbf{Y}_2)^T\mathbf{U}$ and $\mathbf{H} = \mathbf{P}+\frac{1}{\mu}\mathbf{Y}_3$. Since each row of $\mathbf{M}$ contains only one nonzero element with value 1, then if $\mathbf{M}(j,i) = 1$, we have $\mathbf{M}(j,:)\mathbf{V} = \mathbf{v}_i$. Thus, let $\mathcal{H}_i = \{j : \mathbf{M}(j,i) = 1\}$, then $\|\mathbf{M}\mathbf{V}-\mathbf{H}\|_F^2$ can be rewritten as

$$\|\mathbf{M}\mathbf{V}-\mathbf{H}\|_F^2 = \sum_{i=1}^{m}\sum_{j\in\mathcal{H}_i}\|\mathbf{v}_i-\mathbf{h}_j\|_2^2 \tag{4.15}$$

With the above trick, Eq.(4.14) can be decomposed into row-wise sub-problem as

$$\min_{\mathbf{v_i}} \frac{\mu(1+|\mathcal{H}_i|)}{2}\|\mathbf{v}_i - \frac{1}{1+|\mathcal{H}_i|}(\mathbf{k}_i + \sum_{j\in\mathcal{H}_i}\mathbf{h}_j)\|_2^2 + \beta\|\mathbf{v}_i\|_2 \tag{4.16}$$

Similarly, using Lemma 4.1, $\mathbf{V}$ can be updated as

$$\mathbf{v}_i = \begin{cases} \frac{1}{(1+|\mathcal{H}_i|)}(1-\frac{\beta}{\mu\|\mathbf{k}_i+\sum_{j\in\mathcal{H}_i}\mathbf{h}_j\|_2})(\mathbf{k}_i+\sum_{j\in\mathcal{H}_i}\mathbf{h}_j), \\ \qquad \text{if } \|\mathbf{k}_i+\sum_{j\in\mathcal{H}_i}\mathbf{h}_j\|_2 > \frac{\beta}{\mu} \\ \mathbf{0}, \qquad \text{otherwise} \end{cases} \tag{4.17}$$

4.3 Update U Optimizing Eq.(4.9) with respect to $\mathbf{U}$ yields the equation

$$\min_{\mathbf{U}^T\mathbf{U}=\mathbf{I}} ||\mathbf{U}-\mathbf{N}||_F^2 \tag{4.18}$$

where $\mathbf{N}$ is defined as $\mathbf{N} = \frac{1}{\mu}\mathbf{Y}_1 + \mathbf{Z} + (\mathbf{X}-\mathbf{E}+\frac{1}{\mu}\mathbf{Y}_2)\mathbf{V}$. Now we have converted the objective function of updating $\mathbf{U}$ to the classical Orthogonal Procrutes problem [35], which can be solved using the following lemma [36]

LEMMA 4.2. *Given the objective in Eq.(4.18), the optimal* $\mathbf{U}$ *is defined as*

$$\mathbf{U} = \mathbf{S}\mathbf{Q}^T \tag{4.19}$$

where $\mathbf{S}$ *and* $\mathbf{Q}$ *are the left and right singular vectors of the economic singular value decomposition (SVD) of* $\mathbf{N}$.

4.4 Update P After removing terms that are irrelevant to $\mathbf{P}$, Eq.(4.9) becomes

$$\min_{\mathbf{P}} \sum_{i=1}^{K}(\frac{\mu}{2}\|\mathbf{p}^i-\mathbf{v}^i+\frac{1}{\mu}\mathbf{y}_3^i\|_2^2 + \alpha\sum_{s=1}^{d}\sum_{t=1}^{n_s}\|\mathbf{p}^i_{|\mathcal{G}_t^s}\|_2) \tag{4.20}$$

Obviously, the above equation can be solved through addressing the following sub-problems:

$$\min_{\mathbf{p}^i_{|\mathcal{G}_t^s}} \frac{1}{2}\|\mathbf{p}^i_{|\mathcal{G}_t^s} - (\mathbf{v}^i_{\mathcal{G}_t^s} - \frac{1}{\mu}\mathbf{y}^i_{3,|\mathcal{G}_t^s})\|_2^2 + \frac{\alpha}{\mu}\|\mathbf{p}^i_{|\mathcal{G}_t^s}\|_2 \tag{4.21}$$

Again, we can apply Lemma 4.1 to solve the above problem and $\mathbf{P}$ is updated as

$$\mathbf{p}^i_{|\mathcal{G}_t^s} = \begin{cases} (1-\frac{\alpha}{\mu\|\mathbf{v}^i_{\mathcal{G}_t^s}-\frac{1}{\mu}\mathbf{y}^i_{3,|\mathcal{G}_t^s}\|})(\mathbf{v}^i_{\mathcal{G}_t^s}-\frac{1}{\mu}\mathbf{y}^i_{3,|\mathcal{G}_t^s}), \\ \qquad \text{if } \|\mathbf{v}^i_{\mathcal{G}_t^s}-\frac{1}{\mu}\mathbf{y}^i_{3,|\mathcal{G}_t^s}\| > \frac{\alpha}{\mu} \\ \mathbf{0}, \qquad \text{otherwise} \end{cases} \tag{4.22}$$

4.5 Update Z Optimizing Eq.(4.9) with respect to $\mathbf{U}$ yields the equation

$$\min_{\mathbf{Z}\geq\mathbf{0}} ||\mathbf{Z}-\mathbf{T}||_F^2 \tag{4.23}$$

where $\mathbf{T}$ is defined as $\mathbf{T} = \mathbf{U} - \frac{1}{\mu}\mathbf{Y}_1$. Clearly, the optimal solution of the above problem is

$$Z_{ij} = \max(T_{ij}, 0) \tag{4.24}$$

4.6 Update $\mathbf{Y}_1$, $\mathbf{Y}_2$, $\mathbf{Y}_3$ and μ After updating the variables, we now need to update the ADMM parameters. According to [33], they are updated as follows:

$$\begin{aligned} \mathbf{Y}_1 &= \mathbf{Y}_1 + \mu(\mathbf{Z}-\mathbf{U}) \\ \mathbf{Y}_2 &= \mathbf{Y}_2 + \mu(\mathbf{X}-\mathbf{U}\mathbf{V}^T-\mathbf{E}) \\ \mathbf{Y}_3 &= \mathbf{Y}_3 + \mu(\mathbf{P}-\mathbf{M}\mathbf{V}) \\ \mu &= \min(\rho\mu, \mu_{max}) \end{aligned} \tag{4.25}$$

Here, $\rho > 1$ is a parameter to control the convergence speed and μ_{max} is a larger number to prevent μ from becoming too large.

With these updating rules, the optimization method for HUFS is summarized in Algorithm 1.

Algorithm 1 The Algorithm for HUFS.

Input: $\mathbf{X} \in \mathbf{R}^{N\times m}, \alpha, \beta, n,$ latent dimension $K, \mathcal{T}$
Output: n selected features

```
Initialize μ = 10^{-3}, ρ = 1.1, μ_max = 10^{10}, U = 0, V = 0 (or use K-means) and M from T
repeat
    Calculate Q = X − UV^T + (1/μ)Y_2
    Update E by Eq.(4.13)
    Update V by Eq.(4.17)
    Calculate N = (1/μ)Y_1 + Z + (X − E + (1/μ)Y_2)V
    Update U by Lemma 4.2
    Update P by Eq.(4.22)
    Calculate T = U − (1/μ)Y_1
    Update Z using Eq.(4.24)
    Update Y_1, Y_2, Y_3 and μ
until convergence
Sort each feature of X according to ||v_i||_2 in descending order and select the top-n ranked ones
```

4.7 Parameter Initialization One way to initialize $\mathbf{U}$ and $\mathbf{V}$ is to set them to be $\mathbf{0}$. As the algorithm runs, the objective function will gradually converge to the optimal value. To accelerate the convergence speed, following the common way of initializing NMF, we use k-means to initialize $\mathbf{U}$ and $\mathbf{V}$. To be specific, we first apply k-means to cluster $\mathbf{X}$, then get the soft cluster indicator to initialize $\mathbf{U}$ and set $\mathbf{V}$ as $\mathbf{X}^T\mathbf{U}$. $\mathbf{Y}_1, \mathbf{Y}_2$ and $\mathbf{Y}_3$ are initialized to be $\mathbf{0}$. μ is typically set in the range of 10^{-6} to 10^{-3} initially depending on the datasets and is updated in each iteration. μ_{max} is often set to be a large value such as 10^{10} to give μ freedom to increase but prevent it from being too large. ρ is empirically set to 1.1 in our algorithm. The larger ρ is , the faster μ becomes larger and the more we penalize the deviation of the equality constraint, which makes it converge faster. However, we may sacrifice some precision of final objective function with very large ρ.

4.8 Convergence Analysis Since our algorithm uses ADMM to optimize the objective function, the convergence of our algorithm adapts from the convergence of ADMM. The detailed convergence proof of ADMM can be found in [37, 33]. Empirically, we find that our algorithm converges within 100 iterations for all the datasets used in evaluation.

4.9 Time Complexity Analysis The computation cost for $\mathbf{E}$ depends on the computation of $\mathbf{Q}$ and the update of $\mathbf{E}$, which are $\mathcal{O}(NmK)$ and $\mathcal{O}(Nm)$, respectively. Similarly, the computation cost for $\mathbf{V}$ involves the computation of $\mathbf{K}, \mathbf{H}$ and the update of $\mathbf{V}$, which are $\mathcal{O}(NmK)$, $\mathcal{O}(K\sum_{s=1}^{d}\sum_{t=1}^{n_s}|\mathcal{G}_t^s|$ and $\mathcal{O}(mK)$. The cost of updating $\mathbf{Z}$ is $\mathcal{O}(NK)$. The main computation cost of $\mathbf{U}$ involves the computation of $\mathbf{N}$ and its SVD decomposition, which are $\mathcal{O}(NmK)$ and $\mathcal{O}(NK^2)$. The cost of updating $\mathcal{P}$ is calculating $\mathcal{P}$ in Eq.(4.22), which is $\mathcal{O}(K\sum_{s=1}^{d}\sum_{t=1}^{n_s}|\mathcal{G}_t^s|)$. Therefore, the overall time complexity in each iteration is $\mathcal{O}(NmK + NK^2 + K\sum_{s=1}^{d}\sum_{t=1}^{n_s}|\mathcal{G}_t^s|)$.

5 Experiments

In this section, we conduct experiments to evaluate the effectiveness of the proposed framework HUFS. After introducing experimental settings, we compare HUFS with the state-of-the-art unsupervised feature selection methods. We conduct experiments on three different categories of datasets, i.e., text, image and biology datasets so as to see how HUFS performs on datasets from different domain. Further experiments are conducted to investigate the effects of important parameters on HUFS.

5.1 Experimental Settings The experiments are conducted on 11 publicly available and widely used benchmark datasets, which can be divided into three different categories as follows: (i) *5 text datasets*: BBC-Sport[1], CNNStory[2], Webkb4[3], Guardian[4] and 20Newsgroups[5]; (ii) *3 image datasets:* COIL20, Yale and warpPIE [6]; and (iii) *3 biology datasets:* Carcinoma [38], B_Cell Chronic Lymphocytic Leukemia (CLL) [7] and Global Cancer Map (GCM) [8].

In practice, datasets that demand feature selection most are those short and fat datasets, i.e., small number of data samples and large number of features. Therefore, our experiments focus on such kind of datasets. The statistics of the aforementioned datasets are summarized in Table 1. In the table, C denotes number of classes. Note that we have label information for these datasets, however, label information is only used as the ground truth for the evaluation purpose; in other words, it is not used by HUFS as well as baseline methods.

Since HUFS require hierarchical structure as input, we briefly describe how we get the hierarchical structures. For text datasets, we use WordNet to get the hierarchical structures based on the semantic meaning of words. For Image datasets, since pixels in images have spatial locality relations. For example, nearby pixels

[1] http://mlg.ucd.ie/datasets/bbc.html
[2] https://sites.google.com/site/qianmingjie/home/datasets/cnntop-and-npr-news
[3] http://www.cs.cmu.edu/afs/cs/project/theo-20/www/data/
[4] http://mlg.ucd.ie/datasets/3sources.html
[5] http://qwone.com/ jason/20Newsgroups/
[6] All three datasets can be downloaded from http://featureselection.asu.edu/datasets.php
[7] Carcinoma and CLL can be downloaded from http://featureselection.asu.edu/datasets.php
[8] http://eps.upo.es/bigs/datasets.html

Table 1: Statistics of the Datasets

Type	Dataset	# Samples	# Features	C
Text	BBCSport	737	4613	5
	CNNStory	142	8682	10
	Webkb4	4168	7770	4
	Guardian	302	3631	6
	20Newsgroup	1000	14675	20
Image	COIL20	1440	1024	20
	Yale	165	1024	15
	warpPIE	210	2420	10
Biology	Carcinoma	174	9182	11
	CLL	111	11340	3
	GCM	190	16063	14

have a high probability to share similar values. We construct the hierarchical structure based on spatial locality. For biology datasets, we use hierarchical clustering on features to get the structure. *Note that constructing hierarchical structures is not part of HUFS. HUFS doesn't produce hierarchical structures but utilize these hierarchical structures for feature selection.*

Following the common way to evaluate unsupervised feature selection, we assess HUFS in terms of clustering performance [22, 13]. Two widely used evaluation metrics, *accuracy* (ACC) and *normalized mutual information* (NMI), are employed to evaluate the quality of clusters. The larger the ACC and NMI are, the better the performance is. In the evaluation, we choose K-means to cluster samples based on the selected features. Since K-means depends on initialization, following previous work [26], we repeat the experiments 20 times and the average results with standard deviation are reported.

5.2 Quality of Selected Features With the hierarchical structures constructed in the above way, we compare HUFS with the following representative state-of-the-art unsupervised feature selection algorithms:

- **LS**: Laplacian Score [11] which evaluates the importance of a feature through its power of locality preservation;
- **MCFS**: Multi-Cluster Feature Selection [23] which selects features using spectral regression with ℓ_1-norm regularization;
- **NDFS**: Nonnegative Discriminative Feature Selection [26] which selects features by a joint framework of nonnegative spectral analysis and $\ell_{2,1}$ regularized regression;
- **RUFS**: Robust Unsupervised Feature Selection [14] which jointly performs robust label learning via local learning regularized robust orthogonal nonnegative matrix factorization and robust feature learning via joint $\ell_{2,1}$-norms minimization; and
- **EUFS**: Embedded Unsupervised Feature Selection [21] which embeds feature selection into the nonnegative matrix factorization based clustering algorithm. Note that EUFS is a special case of HUFS by setting $\alpha = 0$ in HUFS and removing the graph regularizer of EUFS.

There are some parameters to be set. First, all baseline methods need to construct the affinity matrix. Following [14], for the baseline methods, we fix the neighborhood size to be 5 for all the datasets to construct the affinity matrix. Second, to fairly compare different unsupervised feature selection methods, we tune the parameters for all methods by a "grid-search" strategy from $\{10^{-6}, 10^{-4}, \ldots, 10^4, 10^6\}$. More details about parameter analysis on HUFS will be discussed in the following subsection. How to determine the optimal number of selected features is still an open problem [39], we vary the number of selected features as {50, 100, 150, ..., 300} for all datasets. Due to the page limitation, we only report the best performance for each algorithm with the above settings. The comparison results are summarized in Table 2 and Table 3 in terms of ACC and NMI, respectively. Note that numbers in parentheses are the numbers of selected features achieving the best performance. From the two tables, we make the following observations:

- The proposed framework HUFS outperforms EUFS. Compared to EUFS, HUFS also utilizes given hierarchical structures. These results support the importance of hierarchical structures for unsupervised feature selection.
- Most of the time, HUFS achieves the best performance with smaller numbers of selected features, which supports that HUFS is more likely to select discriminative features given the auxiliary information from the hierarchical structure.
- On all 11 datasets, HUFS often obtains better performance than baseline methods. There are two major reasons. First, HUFS directly embeds feature selection into a clustering algorithm and selects features in a batch mode. Second, HUFS exploits hierarchical structures, which provides auxiliary information to guide feature selection.

5.3 Parameter Sensitivity Analysis There are two important parameters for HUFS – (1) α controlling the contribution from hierarchical structures of features and (2) β controlling the row sparsity of $\mathbf{V}$. In this subsection, we perform parameter analysis on these two parameters.

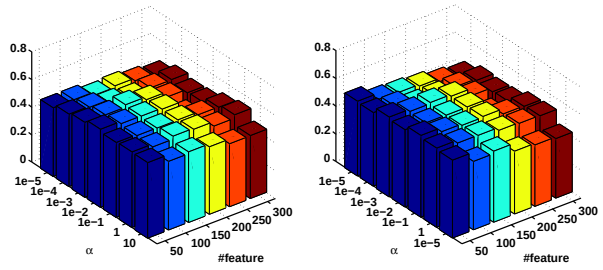
To evaluate the sensitivity of α, we fix $\beta = 0.01$ and vary the value of α as $\{10^{-5}, 10^{-4}, \ldots, 10\}$. The performance variation w.r.t. α and the number of selected fea-

Table 2: Clustering performance(ACC%±std) of feature selection algorithms on the 11 datasets in terms of ACC

Dataset	LS	MCFS	NDFS	RUFS	EUFS	HUFS
BBCSport	73.4±10.9(300)	75.7±12.8(200)	77.0±1.15(250)	76.2±11.9(250)	75.9±13.87(150)	**79.2±6.69**(150)
CNNStory	54.4±4.55(150)	53.5±7.71(50)	49.3±4.97(50)	51.6±6.01(50)	51.9±4.27(200)	**56.3±5.66**(50)
Webkb4	49.5±1.07(300)	48.5±2.50(250)	51.2±0.74(250)	51.1±0.81(200)	50.7±1.67(200)	**52.1±1.05**(200)
Guardian	50.0±6.55(200)	51.1±4.55(150)	51.2±8.96(300)	53.0±6.87(150)	51.1±6.90(250)	**53.9±6.78**(150)
20Newsgroup	17.8±1.33(100)	17.1±0.78(150)	17.2±1.85(250)	17.7±1.35(200)	17.1±1.23(200)	**18.2±1.49**(150)
COIL20	56.2±5.45(250)	60.4±4.52(50)	59.3±3.86(300)	62.02±6.35(250)	61.9±5.43(250)	**63.9±4.25**(250)
Yale	43.9±4.58(250)	42.9±4.19(150)	42.5±2.21(200)	41.5±3.35(250)	42.0±3.52(150)	**44.5±3.12**(150)
warpPIE	33.8±2.54(300)	38.5±3.72(200)	37.4±3.89(250)	39.9±4.10(50)	41.3±4.21(50)	**42.5±3.27**(50)
Carcinoma	69.6±7.95(300)	72.7±6.73(50)	67.9±8.25(200)	72.2±8.16(150)	72.7±7.31(200)	**73.8±6.78**(150)
CLL	55.1±1.44(100)	53.0±4.69(50)	51.8±4.42(200)	49.5±7.39(200)	52.4±5.19(200)	**55.2±1.76**(100)
GCM	41.9±4.25(300)	47.2±4.58(150)	48.5±5.07(300)	47.9±4.36(200)	47.7±3.98(100)	**49.5±3.90**(100)

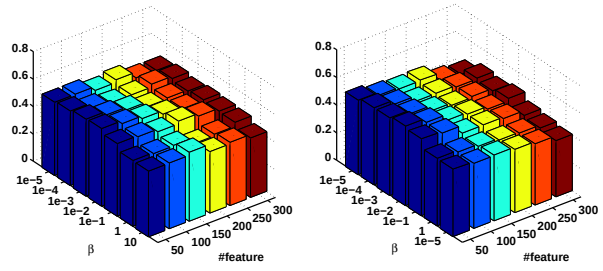
Table 3: Clustering performance(NMI±std) of feature selection algorithms on the 11 datasets in terms of NMI

Dataset	LS	MCFS	NDFS	RUFS	EUFS	HUFS
BBCSport	0.627±0.10(300)	0.655±0.12(300)	0.643±0.02(300)	0.652±0.09(250)	0.628±0.11(200)	**0.660±0.09**(150)
CNNStory	0.562±0.04(200)	0.517±0.08(50)	0.509±0.07(50)	0.538±0.06(50)	0.522±0.05(150)	**0.570±0.05**(50)
Webkb4	0.208±0.02(250)	0.227±0.02(200)	0.235±0.03(250)	0.236±0.02(200)	0.228±0.03(200)	**0.245±0.02**(200)
Guardian	0.369±0.06(200)	0.371±0.07(150)	0.393±0.10(300)	0.413±0.07(300)	0.385±0.07(250)	**0.425±0.08**(200)
20Newsgroup	0.165±0.02(200)	0.153±0.01(150)	0.170±0.02(200)	0.167±0.02(200)	0.166±0.02(200)	**0.174±0.02**(150)
COIL20	0.708±0.03(250)	0.737±0.03(50)	0.727±0.02(300)	0.746±0.02(250)	0.748±0.02(250)	**0.767±0.03**(250)
Yale	0.518±0.02(200)	0.515±0.03(150)	0.501±0.03(250)	0.503±0.02(150)	0.508±0.03(150)	**0.522±0.03**(150)
warpPIE	0.364±0.03(300)	0.457±0.03(200)	0.438±0.03(250)	0.442±0.03(100)	0.447±0.04(50)	**0.467±0.04**(50)
Carcinoma	0.721±0.06(300)	0.785±0.04(50)	0.714±0.06(200)	0.776±0.05(150)	0.774±0.04(200)	**0.787±0.03**(150)
CLL	0.230±0.01(100)	0.195±0.09(50)	0.173±0.04(150)	0.172±0.08(200)	0.236±0.01(200)	**0.318±0.02**(100)
GCM	0.470±0.02(300)	0.542±0.02(150)	0.544±0.03(300)	0.540±0.03(200)	0.536±0.03(100)	**0.556±0.03**(100)



(a) ACC on CNNStory (b) NMI on CNNStory

Figure 3: Performance with different α



(a) ACC on CNNStory (b) NMI on CNNStory

Figure 4: Performance with different β

tures is depicted in Figure 3. Note that due to the page limitation, we only show the results on CNNStory; however, we make similar observations on other datasets. In general, with the increase of α, the performance first increases and then decreases. In particular, when α increases from 10^{-5} to 10^{-4}, the performance increases a lot, which further supports the importance of hierarchical structures. When α is between 10^{-3} and 0.1, the performance is relatively stable, which eases the process to determine the optimal value of α in practice. Similarly, to evaluate the sensitivity of β, we fix $\alpha = 0.01$ and vary the value of β as $\{10^{-5}, 10^{-4}, \ldots, 10\}$. The performance variation w.r.t. β and the number of selected features is demonstrated in Figure 4. We have similar observations about β compared to α.

6 Conclusion

In this paper, we propose a new unsupervised feature selection approach, HUFS, which embeds feature selection into a clustering algorithm and captures hierarchical structures of features. In particular, we use the index tree to represent hierarchical structures and each node of the index tree works as constraints to guide feature selection. We propose an efficient optimization framework based on ADMM to solve the proposed framework. Experimental results on 11 different real-world datasets demonstrate the effectiveness of the proposed framework and the importance of hierarchical structures for unsupervised feature selection. We also give guidances on how to construct hierarchical structures of features in different domains.

There are several directions needing further investigation. Currently, we only explore three different ways of constructing the hierarchical structures, and one fu-

ture work is to explore more methods of constructing hierarchical structures for HUFS. Another direction is to extend the unsupervised feature selection with hierarchical structures with semi-supervised or multitask learning [40] setting.

7 Acknowledgements

This material is based upon work supported by, or in part by, the NSF grants #1614576 and IIS-1217466, and the ONR grant N00014-16-1-2257.

References

[1] A. Jain and D. Zongker, "Feature selection: Evaluation, application, and small sample performance," pp. 153–158, 1997.

[2] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *JMLR*, vol. 3, 2003.

[3] H. Liu and H. Motoda, *Computational methods of feature selection*. CRC Press, 2007.

[4] G. H. John, R. Kohavi *et al.*, "Irrelevant features and the subset selection problem." in *ICML*, 1994.

[5] S. Wang, J. Tang, and H. Liu, "Feature selection," in *Encyclopedia of Machine Learning and Data Mining*, C. Sammut and G. I. Webb, Eds. Springer, 2016.

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

[7] R. O. Duda, P. E. Hart, and D. G. Stork, "Pattern classification. 2nd," *Edition. New York*, 2001.

[8] F. Nie, S. Xiang, Y. Jia, C. Zhang, and S. Yan, "Trace ratio criterion for feature selection." in *AAAI*, 2008.

[9] Z. Zhao, L. Wang, and H. Liu, "Efficient spectral feature selection with minimum redundancy," in *AAAI*.

[10] L. Wolf and A. Shashua, "Feature selection for unsupervised and supervised inference: The emergence of sparsity in a weight-based approach," *JMLR*, 2005.

[11] X. He, D. Cai, and P. Niyogi, "Laplacian score for feature selection," in *NIPS*, 2005, pp. 507–514.

[12] C. Boutsidis, P. Drineas, and M. W. Mahoney, "Unsupervised feature selection for the k-means clustering problem," in *NIPS*, 2009.

[13] Y. Yang, H. T. Shen, Z. Ma, Z. Huang, and X. Zhou, "l2, 1-norm regularized discriminative feature selection for unsupervised learning." in *IJCAI*, 2011.

[14] M. Qian and C. Zhai, "Robust unsupervised feature selection," in *IJCAI*, 2013.

[15] S. Alelyani, J. Tang, and H. Liu, "Feature selection for clustering: A review," in *Data Clustering: Algorithms and Applications*. CRC Press, 2013, pp. 29–60.

[16] G. A. Miller, "Wordnet: a lexical database for english," *Communications of the ACM*, 1995.

[17] J. Liu and J. Ye, "Moreau-yosida regularization for grouped tree structure learning," in *NIPS*, 2010.

[18] A. A. Alizadeh, M. B. Eisen, R. E. Davis *et al.*, "Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling," *Nature*, 2000.

[19] Y. Liu, J. Wang, and J. Ye, "An efficient algorithm for weak hierarchical lasso," ACM, pp. 283–292, 2014.

[20] L. Yu and H. Liu, "Feature selection for high-dimensional data: A fast correlation-based filter solution," in *ICML*, vol. 3, 2003, pp. 856–863.

[21] S. Wang, J. Tang, and H. Liu, "Embedded unsupervised feature selection," in *AAAI*, 2015.

[22] Z. Zhao and H. Liu, "Spectral feature selection for supervised and unsupervised learning," in *ICML*, 2007.

[23] D. Cai, C. Zhang, and X. He, "Unsupervised feature selection for multi-cluster data," in *SIGKDD*, 2010.

[24] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *Journal of the Royal Statistical Society: Series B*, 2006.

[25] M. Shiga and H. Mamitsuka, "Non-negative matrix factorization with auxiliary information on overlapping groups," *TKDE*, no. 1, pp. 1–1, 2015.

[26] Z. Li, Y. Yang, J. Liu, X. Zhou, and H. Lu, "Unsupervised feature selection using nonnegative spectral analysis," in *AAAI*, 2012.

[27] J. Gui, Z. Sun, S. Ji, D. Tao, and T. Tan, "Feature selection based on structured sparsity: A comprehensive study," *TNNLS*, 2016.

[28] R. Jenatton, J. Mairal, G. Obozinski, and F. Bach, "Proximal methods for hierarchical sparse coding," *JMLR*, vol. 12, pp. 2297–2334, 2011.

[29] D. Yogatama, M. Faruqui, C. Dyer, and N. A. Smith, "Learning word representations with hierarchical sparse coding," in *Proc. of ICML*, 2015.

[30] J. Tang and H. Liu, "An unsupervised feature selection framework for social media data," *TKDE*, 2014.

[31] Y. Wang, S. Wang, J. Tang, G. Qi, H. Liu, and B. Li, "Clare: A joint approach to label classification and tag recommendation," in *AAAI*, 2017.

[32] Y. Wang, S. Wang, J. Tang, H. Liu, and B. Li, "PPP: joint pointwise and pairwise image label prediction," in *CVPR*, 2016.

[33] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *FTML*, vol. 3, no. 1, pp. 1–122, 2011.

[34] J. Liu, S. Ji, and J. Ye, "Multi-task feature learning via efficient l 2, 1-norm minimization," in *UAI*, 2009.

[35] P. H. Schönemann, "A generalized solution of the orthogonal procrustes problem," *Psychometrika*, vol. 31, no. 1, pp. 1–10, 1966.

[36] J. Huang, F. Nie, H. Huang, and C. Ding, "Robust manifold nonnegative matrix factorization," *TKDD*, vol. 8, no. 3, p. 11, 2014.

[37] T. Goldstein, B. ODonoghue, and S. Setzer, "Fast alternating direction optimization methods," *CAM report*, pp. 12–35, 2012.

[38] A. I. Su, J. B. Welsh, L. M. Sapinoso *et al.*, "Molecular classification of human carcinomas by use of gene expression signatures," *Cancer research*, 2001.

[39] J. Tang and H. Liu, "Feature selection with linked data in social media," in *SDM.*, 2012, pp. 118–128.

[40] W. Lian, R. Henao, V. Rao, J. Lucas, and L. Carin, "A multitask point process predictive model," in *ICML*, 2015, pp. 2030–2038.