Graph Autoencoder-Based Unsupervised Feature Selection

Siwei Feng
Department of Electrical and Computer Engineering
University of Massachusetts Amherst
Amherst, MA, 01003
siwei@umass.edu

Marco F. Duarte
Department of Electrical and Computer Engineering
University of Massachusetts Amherst
Amherst, MA, 01003
mduarte@ecs.umass.edu

***Abstract*—Feature selection is a dimensionality reduction technique that selects a subset of representative features from high-dimensional data in order to eliminate redundancy. Recently, feature selection methods based on sparse learning have attracted significant attention due to their outstanding performance compared with traditional methods that ignore correlation between features. However, they are restricted by design to linear data transformations, a potential drawback given that the underlying correlation structures of data are often non-linear. To leverage a more sophisticated embedding, we propose an autoencoder-based unsupervised feature selection approach that leverages a single-layer autoencoder for a joint framework of feature selection and manifold learning, with spectral graph analysis on the projected data into the learning process to achieve local data geometry preservation from the original data space to the low-dimensional feature space.**

***Index Terms*—Unsupervised Feature Selection, Autoencoder, Manifold Learning, Spectral Graph Analysis, Column Sparsity**

I. Introduction

In recent years, high-dimensional data can be found in many areas such as computer vision, pattern recognition, data mining, etc. High dimensionality enables data to include more information, but learning from high-dimensional data often suffer from several issues such as Hughes phenomenon [1] and feature redundancy [2], etc. Moreover, several papers in the literature have shown that the intrinsic dimensionality of high-dimensional data is actually small [3–5]. Thus, dimensionality reduction is a popular preprocessing step for high-dimensional data analysis, which decreases time for data processing and also improves generalization of learned models.

Feature selection [6–11] approaches aim at selecting a subset of the features, and have the advantage of preserving the same feature space as that of raw data. In recent years, feature selection algorithms aiming at preserving intrinsic data structure [12–21] have attracted significant attention due to their good performance and interpretability [22]. In these methods, data are linearly projected onto new spaces through a transformation matrix, with fitting errors being minimized along with some sparse regularization terms. Feature importance is usually scored using the norms of corresponding rows/columns in the transformation matrix. In some methods [15–17, 23], the local data geometric structure, which is usually characterized by nearest neighbor graphs, is also preserved in the low-dimensional projection space. One basic assumption of these methods is that the data to be processed lie in or near linear space. However, this is not always true in practice, in particular with more sophisticated data.

In this paper, we propose a novel algorithm for graph and autoencoder-based feature selection (GAFS). The reason we choose an autoencoder as a data model is because of its broader goal of data reconstruction, which is a good match in spirit for an unsupervised feature selection framework: we expect to be able to infer the entire data vector from just a few of its dimensions. In this method, we integrate three objective functions into a single optimization framework: (i) we use a single-layer autoencoder to reconstruct the input data; (ii) we use an $\ell_{2,1}$-norm penalty on the columns of the weight matrix connecting the autoencoder's input layer and hidden layer to provide feature selection; (iii) we preserve the local geometric structure of the data through to the corresponding hidden layer activation space.

The key contribution of this paper is twofold: 1). We propose a novel unsupervised feature selection framework which is based on an autoencoder and graph data regularization. By using this framework, data manifold can be leveraged, which loosens the assumption of a subspace model in many relevant techniques. 2). We present an efficient solver for the optimization problem underlying the proposed unsupervised feature selection scheme.

The rest of this paper is organized as follows. Section II overviews related work. The proposed framework and the corresponding optimization scheme are presented in Section III. Experimental results and the corresponding analysis are provided in Section IV. Section V includes conclusion and future work.

II. Related Work

In this section, we provide a review of literature related to our proposed method and introduce the paper's notation standard. For a matrix $\mathbf{Z}$, $\mathbf{Z}^{(q)}$ denotes the q^{th} column of the matrix, while $\mathbf{Z}^{(p,q)}$ denotes the entry of the matrix at the p^{th} row and q^{th} column.

The $\ell_{r,p}$-norm for a matrix $\mathbf{W} \in \mathbb{R}^{a\times b}$ is denoted as

$$\|\mathbf{W}\|_{r,p} = \left(\sum_{j=1}^{b}\left(\sum_{i=1}^{a}|\mathbf{W}^{(i,j)}|^r\right)^{p/r}\right)^{1/p}. \quad (1)$$

Note that unlike most of the literature, our outer sum is performed over the ℓ_r-norms of the matrix columns instead of its rows; this is done for notation convenience of our subsequent mathematical expressions. Datasets are denoted by $\mathbf{X} = [\mathbf{X}^{(1)}, \mathbf{X}^{(2)}, \cdots, \mathbf{X}^{(n)}] \in \mathbb{R}^{d \times n}$, where $\mathbf{X}^{(i)} \in \mathbb{R}^d$ is the ith sample in $\mathbf{X}$ for $i = 1, 2, \cdots, n$, and where d and n denote data dimensionality and number of data points in $\mathbf{X}$, respectively.

A. Sparse Learning-Based Unsupervised Feature Selection

Many unsupervised feature selection methods based on subspace structure preservation have been proposed in the past decades. The basic idea is to use a transformation matrix to project data to a particular embedding space and guide feature selection based on the sparsity of the transformation matrix [12]. To be more specific, the generic framework of these methods is based on the optimization

$$\min_{\mathbf{W}} \mathcal{L}(\mathbf{Y}, \mathbf{W}\mathbf{X}) + \lambda \mathcal{R}(\mathbf{W}), \tag{2}$$

where $\mathbf{Y} = [\mathbf{Y}^{(1)}, \mathbf{Y}^{(2)}, \cdots, \mathbf{Y}^{(n)}] \in \mathbb{R}^{m \times n}$ $(m < d)$ is an embedding matrix in which $\mathbf{Y}^{(i)} \in \mathbb{R}^m$ for $i = 1, 2, \cdots, n$ denotes the representation of data point $\mathbf{X}^{(i)}$ in the obtained low-dimensional space. $\mathcal{L}(\cdot)$ denotes a loss function, and $\mathcal{R}(\cdot)$ denotes a regularization function on the transformation matrix $\mathbf{W} \in \mathbb{R}^{m \times d}$. The methods differ in their choice of embedding $\mathbf{Y}$ and loss and regularization functions; some examples are presented below.

Multi-cluster feature selection (MCFS) [13] and minimum redundancy spectral feature selection (MRSF) [14] are two long-standing and well-known unsupervised feature selection methods. In MCFS, a graph is first constructed on training data. Then spectral clustering is performed on data points using the top eigenvectors of graph Laplacian. We refer readers to [13] for more details on this spectral clustering procedure. After that, all data points are regressed to the learned embedding through a transformation matrix $\mathbf{W} \in \mathbb{R}^{m \times d}$. The loss function is set to the Frobenius norm of the linear transformation error and the regularization function is set to the $\ell_{1,1}$ norm of the transformation matrix, which promotes sparsity. A score for each feature is measured by the maximum absolute value of the corresponding column of the transformation matrix. MRSF is a variant of MCFS that changes the regularization function from an $\ell_{1,1}$-norm to an $\ell_{2,1}$-norm that enforces column sparsity on the transformation matrix. These two algorithms are equivalent otherwise.

The performance of both MCFS and MRSF is often degraded by the separate nature of linear dimensionality reduction and feature selection [24]. Many approaches on joint linear embedding and feature selection have been proposed to address this problem. For example, in unsupervised discriminative feature selection (UDFS) [16], data instances are assumed to come from c classes. UDFS uses local data geometric structure, which is based on the k-nearest neighbor set of each data point, to incorporate local data discriminative information into a feature selection framework. Like MCFS and MRFS, UDFS also assumes the existence of a transformation matrix $\mathbf{W} \in \mathbb{R}^{m \times c}$ that maps data to a low-dimensional space. One drawback of these discriminative exploitation feature selection methods is that the feature selection performance relies on an accurate estimation of number of classes.

Instead of projecting data onto a low-dimensional subspace, some approaches consider combining unsupervised feature selection methods with self-representation. In these methods, each feature is assumed to be representable as a linear combination of all (other) features, i.e., $\mathbf{X} = \mathbf{W}\mathbf{X} + \mathbf{E}$, where $\mathbf{W} \in \mathbb{R}^{d \times d}$ is a representation matrix and $\mathbf{E} \in \mathbb{R}^{d \times n}$ denotes a reconstruction error. Zhu et. al. [18] proposed a regularized self-representation (RSR) model for unsupervised feature selection that sets both the loss function and the regularization function to $\ell_{2,1}$-norms on the representation error $\mathbf{E}$ (for robustness to outlier samples) and transformation matrix $\mathbf{W}$ (for feature selection), respectively. Extensions of RSR include [19, 20].

B. Single-Layer Autoencoder

A single-layer autoencoder is an artificial neural network that aims to learn a function $h(\mathbf{x}; \mathbf{\Theta}) \approx \mathbf{x}$ with a single hidden layer, where $\mathbf{x} \in \mathbb{R}^d$ is the input data, $h(\cdot)$ is a nonlinear function, and $\mathbf{\Theta}$ is a set of parameters. To be more specific, an autoencoder contains a two-fold workflow, which are encoding and decoding. Encoding aims at mapping the input data $\mathbf{x}$ to a compressed data representation $\mathbf{y} \in \mathbb{R}^m$ using $\mathbf{y} = \sigma(\mathbf{W_1}\mathbf{x} + \mathbf{b}_1)$, where $\mathbf{W}_1 \in \mathbb{R}^{m \times d}$ is a weight matrix, $\mathbf{b}_1 \in \mathbb{R}^m$ is a bias vector, and $\sigma(\cdot)$ is a nonlinear activation function applied element-wise. Decoding aimes at mapping the compressed data representation $\mathbf{y}$ to a vector in the original data space $\bar{\mathbf{x}} \in \mathbb{R}^d$ using $\bar{\mathbf{x}} = \sigma(\mathbf{W_2}\mathbf{y} + \mathbf{b}_2)$, where $\mathbf{W}_2 \in \mathbb{R}^{d \times m}$ and $\mathbf{b}_2 \in \mathbb{R}^d$ are the corresponding weight matrix and bias vector, respectively.

The data reconstruction capability of the autoencoder makes it suitable to capture the essential information of the data while discarding information that is not useful or redundant. Therefore, it is natural to assume that the compressed representation in the hidden layer of a single-layer autoencoder can capture the manifold structure of the input data when such manifold structure exists and is approximated well by the underlying weighting and nonlinearity operations.

The idea of using a single layer autoencoder to do unsupervised feature selection is first introduced in [25], in which an autoencoder feature selector (AEFS) is proposed. The objective function of AEFS includes three parts: a loss function based on a single-layer autoencoder promoting broad data structure preservation; a regularization term promoting feature selection; and a weight decay regularization term. As mentioned above, a single-layer autoencoder aims at minimizing the reconstruction error between output and input data by optimizing a reconstruction error-driven loss function:

$$\mathcal{L}(\mathbf{\Theta}) = \frac{1}{2n} \sum_{i=1}^{n} \|\mathbf{X}^{(i)} - h(\mathbf{X}^{(i)}; \mathbf{\Theta})\|_2^2 = \frac{1}{2n} \|\mathbf{X} - h(\mathbf{X}; \mathbf{\Theta})\|_F^2,$$
$$h(\mathbf{X}; \mathbf{\Theta}) = \sigma\left(\mathbf{W}_2 \cdot \sigma(\mathbf{W}_1\mathbf{X} + \mathbf{b}_1) + \mathbf{b}_2\right)$$

where $\mathbf{\Theta} = [\mathbf{W}_1, \mathbf{W}_2, \mathbf{b}_1, \mathbf{b}_2]$. Since $\mathbf{W}_1$ is a weight matrix applied directly on the input data, each column of $\mathbf{W}_1$ can be used to measure the importance of the corresponding data feature. Therefore, $\mathcal{R}(\Theta) = \|\mathbf{W}_1\|_{2,1}$ can be used as a regularization function to promote feature selection. A weight decay regularization term $\mathcal{Q}(\Theta) = \frac{1}{2}\left(\|\mathbf{W}_1\|_F^2 + \|\mathbf{W}_2\|_F^2\right)$ is also included in AEFS. Therefore, the objective function in AEFS can be formulated as

$$\min_{\mathbf{\Theta}} \frac{1}{2n}\|\mathbf{X} - h(\mathbf{X}; \mathbf{\Theta})\|_F^2 + \lambda\|\mathbf{W}_1\|_{2,1} + \frac{\beta}{2}\left(\|\mathbf{W}_1\|_F^2 + \|\mathbf{W}_2\|_F^2\right)$$

where λ and β are two balance parameters.

One drawback of this method is that they do not preserve local data geometric structure from the raw feature space to the hidden layer/activation space. This means there is no guarantee that the relative similarity of each instance pair is preserved across the embedding, which may be harmful to the performance of the subsequent feature selection.

III. Proposed Method

In this section, we introduce our proposed graph autoencoder-based unsupervised feature selection (GAFS). Our proposed framework performs broad data structure preservation through a single-layer autoencoder and also preserves local data geometric structure through spectral graph analysis.

Local geometric structures of the data often contain discriminative information of neighboring data point pairs [13]. They assume that nearby data points should have similar representations. It is often more efficient to combine both broad and local data information during low-dimensional subspace learning [26]. In order to characterize the local data geometric structure, we construct a k-nearest neighbor (kNN) graph $\mathbb{G}$ on the data space. The edge weight between two connected data points is determined by the similarity between those two points. In this paper, we choose cosine distance as similarity measurement for its simplicity. Therefore the adjacency matrix $\mathbf{A}$ for the graph $\mathbb{G}$ is defined as

$$\mathbf{A}^{(i,j)} = \begin{cases} \dfrac{\mathbf{X}^{(i)^T}\mathbf{X}^{(j)}}{\|\mathbf{X}^{(i)}\|_2\|\mathbf{X}^{(j)}\|_2} & \mathbf{X}^{(i)} \in \mathcal{N}_k(\mathbf{X}^{(j)}) \text{ or } \mathbf{X}^{(j)} \in \mathcal{N}_k(\mathbf{X}^{(i)}), \\ 0 & \text{otherwise,} \end{cases}$$

where $\mathcal{N}_k(\mathbf{X}^{(i)})$ denotes the k-nearest neighborhood set for $\mathbf{X}^{(i)}$, and $\mathbf{X}^{(i)^T}$ refers to the transpose of $\mathbf{X}^{(i)}$. The Laplacian matrix $\mathbf{L}$ of the graph $\mathbb{G}$ is defined as $\mathbf{L} = \mathbf{D} - \mathbf{A}$, where $\mathbf{D}$ is a diagonal matrix whose i^{th} element on the diagonal is defined as $\mathbf{D}^{(i,i)} = \sum_{j=1}^{n} \mathbf{A}^{(i,j)}$.

In order to preserve the local data geometric structure in the learned embedding (i.e., if two data points $\mathbf{X}^{(i)}$ and $\mathbf{X}^{(j)}$ are close in original data space then the corresponding low-dimensional representations $\mathbf{Y}^{(i)}$ and $\mathbf{Y}^{(j)}$ are also close in the low-dimensional embedding space), we set up the following graph regularization cost function:

$$\mathcal{G}(\mathbf{\Theta}) = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\|\mathbf{Y}^{(i)} - \mathbf{Y}^{(j)}\|_2^2\mathbf{A}^{(i,j)} = \text{Tr}(\mathbf{Y}(\mathbf{\Theta})\mathbf{L}\mathbf{Y}(\mathbf{\Theta})^{\text{T}}),$$

where $\text{Tr}(\cdot)$ denotes the trace operator, $\mathbf{Y}^{(i)}(\mathbf{\Theta}) = \sigma(\mathbf{W}_1\mathbf{X}^{(i)} + \mathbf{b}_1)$ for $i = 1, 2, \cdots, n$ (and we often drop the dependence on $\mathbf{\Theta}$ for readability), and $\mathbf{Y}(\mathbf{\Theta}) = [\mathbf{Y}^{(1)}(\mathbf{\Theta}), \mathbf{Y}^{(2)}(\mathbf{\Theta}), \cdots, \mathbf{Y}^{(n)}(\mathbf{\Theta})]$.

The objective function of GAFS can be written in terms of the following minimization with respect to the parameters $\mathbf{\Theta} = [\mathbf{W}_1, \mathbf{W}_2, \mathbf{b}_1, \mathbf{b}_2]$:

$$\begin{aligned} \hat{\mathbf{\Theta}} &= \arg\min_{\mathbf{\Theta}} \mathcal{F}(\mathbf{\Theta}) = \arg\min_{\mathbf{\Theta}} \mathcal{L}(\mathbf{\Theta}) + \mathcal{R}(\mathbf{\Theta}) + \mathcal{G}(\mathbf{\Theta}) \\ &= \arg\min_{\mathbf{\Theta}} \left[\frac{1}{2n}\|\mathbf{X} - h(\mathbf{X}; \mathbf{\Theta})\|_F^2 + \lambda\|\mathbf{W}_1\|_{2,1} + \gamma\text{Tr}(\mathbf{Y}\mathbf{L}\mathbf{Y}^{\text{T}})\right], \end{aligned} \tag{3}$$

where λ and γ are two balance parameters. In this paper, we use the sigmoid function as the nonlinear activation function $h(\cdot)$. Since we do not observe significant influence of the weight decay regularization term $\mathcal{Q}$ on feature selection, we discard this term in our objective function in order to reduce computational load. Filter-based feature selection is then performed using the score function $GAFS(q) = \|\mathbf{W}_1^{(q)}\|_2$ based on the weight matrix $\mathbf{W}_1$ from $\hat{\mathbf{\Theta}}$. Details on the implementation of (3) are provided in [27].

IV. Experiments

In this section, we evaluate the feature selection performance of GAFS as well as other state-of-the-art algorithms on two benchmark datasets. We first select p representative features and then perform clustering on those selected features.

A. Data Description

We perform experiments on two image benchmark datasets: MNIST and COIL20. Datasets are downloaded from http://featureselection.asu.edu/datasets.php

B. Evaluation Metric

We perform unsupervised learning (i.e., clustering) tasks on datasets formulated by the selected features in order to evaluate the effectiveness of feature selection algorithms. We use k-means clustering on the selected features and use clustering accuracy (ACC) to evaluate the clustering performance of all methods, which is defined as $\text{ACC} = \frac{1}{n}\sum_{i=1}^{n}\delta(g_i, \text{map}(c_i))$, where n is the total number of data samples, $\delta(\cdot)$ is defined by $\delta(a, b) = 1$ when $a = b$ and 0 when $a \neq b$, $\text{map}(\cdot)$ is the optimal mapping function between cluster labels and class labels obtained using the Hungarian algorithm [28], and c_i and g_i are the clustering and ground truth labels of a given data sample $\mathbf{x_i}$, respectively. We repeat the clustering process 20 times with random initialization for each case following the setup of [13] and [16], and we report the corresponding mean values.

C. Experimental Setup

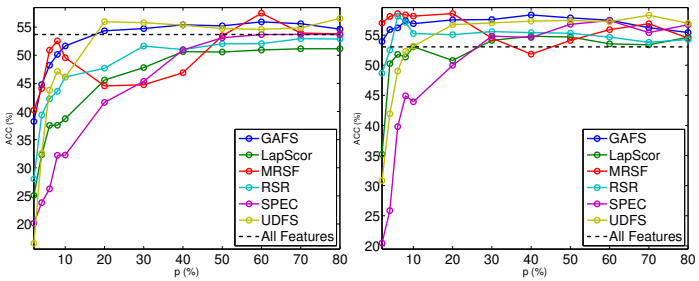
In our last experiment, we compare GAFS with Laplacian Score[1] [29], SPEC[2] [30], MRSF[3] [14], UDFS[4] [16], and

[1]Available at http://www.cad.zju.edu.cn/home/dengcai/Data/code/LaplacianScore.m

[2]Available at https://github.com/matrixlover/LSLS/blob/master/fsSpectrum.m

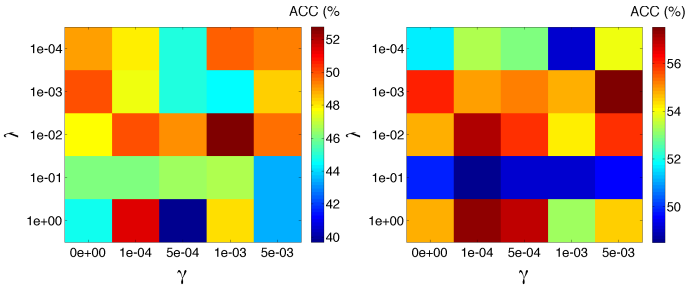
[3]Available at https://sites.google.com/site/alanzhao/Home

[4]Available at http://www.cs.cmu.edu/ yiyang/UDFS.rar



(a) MNIST (b) COIL20

Fig. 1. Clustering accuracy with respect to different unsupervised feature selection algorithms and the percentage of features selection p (%)



(a) MNIST (b) COIL20

Fig. 2. Performance variation of the GAFS w.r.t. balance parameters λ (sparsity penalty) and γ (graph penalty).

RSR[5] [18]. In this experiment, we fix some parameters and tune others according to a "grid-search" strategy. For all algorithms, we select $p \in \{2\%, 4\%, 6\%, 8\%, 10\%, 20\%, 30\%, 40\%, 50\%, 60\%, 70\%, 80\%\}$ of all features for each dataset. For all graph-based algorithms, the number of nearest neighbor in a kNN graph is set to 5. For all algorithms projecting data onto a low-dimensional space, the space dimensionality is set in the range of $m \in \{10, 20, 30, 40\}$. In GAFS, the range for the hidden layer size is set to match that of the subspace dimensionality m[6], while the balance parameters are given ranges $\lambda \in \{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1\}$ and $\gamma \in \{0, 10^{-4}, 5 \times 10^{-4}, 10^{-3}, 5 \times 10^{-3}\}$, respectively. We refer readers to [27] for parameter setup of competing methods.

D. Performance Comparison

We present the ACC results of GAFS and the comparison feature selection algorithms on all datasets in Fig. 1. From these figures, we can find that GAFS is consistently comparable with the best performing methods with respect to different percentages of selected features. Though methods such as MRSF and UDFS outperform GAFS in some cases, they cannot provide stable performance. To be more concrete, MRSF provides similar performance with GAFS when p is large or small, but the clustering curves drop drastically in the middle; UDFS is comparable with GAFS after $p = 20\%$ for both datasets, but for smaller p the competing method provide lower clustering accuracy than GAFS. Comparing the performance of GAFS with that of using all features, which is represented by a black dashed line in each figure, we can find that GAFS can always achieve better performance with far less features. In the meanwhile, with fewer features, the computational load in corresponding clustering tasks can be decreased. These results demonstrate the effectiveness of GAFS in terms of removing irrelevant and redundant features in clustering tasks.

E. Parameter Sensitivity

We study the performance of GAFS on balance parameters λ and γ, with fixed percentage of selected features and hidden layer size. We set $p = 20\%$, as Fig. 1 shows that the performance stabilizes starting at that value of p. For subspace dimensionality, we choose $m = 10$ since we observe that the performance of GAFS is not sensitive to the value of m. The performance results are shown in Fig. 2. For the parameter λ, which controls the column sparsity of $\mathbf{W}_1$, we can find that for both MNIST and COIL20, the overall performance is best when $\lambda = 10^{-2}$ and both smaller and larger values of λ degrade the performance. This is because the diversity among instances of these two datasets is large enough: a large value of λ may remove informative features, while a small value of λ prevents the exclusion of small, irrelevant, or redundant features. For the parameter γ, which controls local data geometric structure preservation, we can find that both large values and small values of γ degrade performance; we also note that $\gamma = 0$ is a case similar to AEFS. On one hand, we can conclude that local data geometric structure preservation does help improve feature selection performance to a certain degree. On the other hand, large weights on local data geometric structure preservation may also harm feature selection performance.

V. Conclusion

In this paper, we proposed a graph and autoencoder-based unsupervised feature selection (GAFS) method which projects the data to a lower-dimensional space using a single-layer autoencoder. We bypass the limitation of existing methods that the dimensionality reduction subspace must be a linear projection of the data space, and simultaneously take local data geometric structure preservation into consideration. Experimental results demonstrate the advantages of GAFS versus methods in the literature for clustering tasks.

In the future, we plan to explore the effectiveness of more elaborate versions of an autoencoder for feature selection purposes. Furthermore, we will also extend our framework to more sophisticated machine learning problems such as transfer learning.

References

[1] M. Pal and G. M. Foody, "Feature selection for classification of hyperspectral data by SVM," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 5, pp. 2297–2307, 2010.

[5]Available at https://github.com/guangmingboy/github_doc

[6]We will alternatively use the terminologies *subspace dimensionality* and *hidden layer size* in descriptions of GAFS.

[2] H. Liu, X. Wu, and S. Zhang, "Feature selection using hierarchical feature clustering," in *Proc. ACM Int. Conf. Info. Knowl. Manag.*, 2011, pp. 979–984.

[3] X. Lu, Y. Wang, and Y. Yuan, "Sparse coding from a Bayesian perspective," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 6, pp. 929–939, 2013.

[4] X. Zhu, S. Zhang, Z. Jin, Z. Zhang, and Z. Xu, "Missing value estimation for mixed-attribute data sets," *IEEE Trans. Knowl. Data Eng.*, vol. 23, no. 1, pp. 110–121, Jan. 2011.

[5] Y. Yang, Z. Ma, F. Nie, X. Chang, and A. G. Hauptmann, "Multi-class active learning by uncertainty sampling with diversity maximization," *Int. J. Comput. Vis.*, vol. 113, no. 2, pp. 113–127, 2015.

[6] L. Song, A. Smola, A. Gretton, K. M. Borgwardt, and J. Bedo, "Supervised feature selection via dependence estimation," in *Int. Conf. Mach. Learn.*, 2007, pp. 823–830.

[7] J. M. Sotoca and F. Pla, "Supervised feature selection by clustering using conditional mutual information-based distances," *Pattern Recognit.*, vol. 43, no. 6, pp. 2068–2081, 2010.

[8] M. Thoma, H. Cheng, A. Gretton, J. Han, H.-P. Kriegel, A. Smola, L. Song, P. S. Yu, X. Yan, and K. Borgwardt, "Near-optimal supervised feature selection among frequent subgraphs," in *Proc. SIAM Int. Conf. Data Mining*, 2009, pp. 1076–1087.

[9] Z. Zhao and H. Liu, "Semi-supervised feature selection via spectral analysis," in *Proc. SIAM Int. Conf. Data Mining*, 2007, pp. 641–646.

[10] Z. Xu, I. King, M. R.-T. Lyu, and R. Jin, "Discriminative semi-supervised feature selection via manifold regularization," *IEEE Trans. Neural Netw.*, vol. 21, no. 7, pp. 1033–1047, 2010.

[11] X. Kong and P. S. Yu, "Semi-supervised feature selection for graph classification," in *Proc. ACM SIGKDD Int. Conf. Knowl. Dis. Data Mining*, 2010, pp. 793–802.

[12] N. Zhou, Y. Xu, H. Cheng, J. Fang, and W. Pedrycz, "Global and local structure preserving sparse subspace learning: An iterative approach to unsupervised feature selection," *Pattern Recognit.*, vol. 53, pp. 87–101, 2016.

[13] D. Cai, C. Zhang, and X. He, "Unsupervised feature selection for multi-cluster data," in *Proc. ACM SIGKDD Int. Conf. Knowl. Dis. Data Mining*, 2010, pp. 333–342.

[14] Z. Zhao, L. Wang, and H. Liu, "Efficient spectral feature selection with minimum redundancy," in *Proc. Assoc. Adv. Artif. Intell.*, Jul. 2010, pp. 673–678.

[15] Q. Gu, Z. Li, and J. Han, "Joint feature selection and subspace learning," in *Proc. Int. Joint Conf. Artif. Intell.*, 2011, pp. 1294–1299.

[16] Y. Yang, H. T. Shen, Z. Ma, Z. Huang, and X. Zhou, "$\ell_{2,1}$-norm regularized discriminative feature selection for unsupervised learning," in *Proc. Int. Joint Conf. Artif. Intell.*, 2011, pp. 1589–1594.

[17] C. Hou, F. Nie, X. Li, D. Yi, and Y. Wu, "Joint embedding learning and sparse regression: A framework for unsupervised feature selection," *IEEE Trans. Cybern.*, vol. 44, no. 6, pp. 793–804, 2014.

[18] P. Zhu, W. Zuo, L. Zhang, Q. Hu, and S. C. Shiu, "Unsupervised feature selection by regularized self-representation," *Pattern Recognit.*, vol. 48, no. 2, pp. 438–446, 2015.

[19] P. Zhu, W. Zhu, W. Wang, W. Zuo, and Q. Hu, "Non-convex regularized self-representation for unsupervised feature selection," *Image Vis. Comput.*, vol. 60, pp. 22–29, 2017.

[20] R. Hu, X. Zhu, D. Cheng, W. He, Y. Yan, J. Song, and S. Zhang, "Graph self-representation method for unsupervised feature selection," *Neurocomputing*, vol. 220, pp. 130–137, 2017.

[21] H. Dadkhahi and M. F. Duarte, "Masking strategies for image manifolds," *IEEE Trans. Image Proc.*, vol. 25, no. 9, pp. 4314–4328, 2016.

[22] J. Li, K. Cheng, S. Wang, F. Morstatter, R. P. Trevino, J. Tang, and H. Liu, "Feature selection: A data perspective," *arXiv preprint arXiv:1601.07996*, 2016.

[23] Z. Li, J. Liu, Y. Yang, X. Zhou, and H. Lu, "Clustering-guided sparse structural learning for unsupervised feature selection," *IEEE Trans. Knowl. Data Eng*, vol. 26, no. 9, pp. 2138–2150, 2014.

[24] R. Shang, Z. Zhang, L. Jiao, C. Liu, and Y. Li, "Self-representation based dual-graph regularized feature selection clustering," *Neurocomputing*, vol. 171, pp. 1241–1253, 2016.

[25] K. Han, Y. Wang, C. Zhang, and C. Xu, "Autoencoder inspired unsupervised feature selection," *arXiv preprint arXiv:1710.08310*, 2018.

[26] L. Bottou and V. Vapnik, "Local learning algorithms," *Neural Comput.*, vol. 4, no. 6, pp. 888–900, 1992.

[27] S. Feng and M. F. Duarte, "Graph autoencoder-based unsupervised feature selection with broad and local data structure preservation," *arXiv preprint arXiv:1801.02251*, 2018.

[28] H. W. Kuhn, "The Hungarian method for the assignment problem," *Nav. Res. Logis.*, vol. 2, no. 1–2, pp. 83–97, 1955.

[29] X. He, D. Cai, and P. Niyogi, "Laplacian score for feature selection," in *Adv. Neural Inf. Proc. Syst.*, 2005, pp. 507–514.

[30] Z. Zhao and H. Liu, "Spectral feature selection for supervised and unsupervised learning," in *Proc. Int. Conf. Mach. Learn.*, 2007, pp. 1151–1157.