MIXED SPARSITY REGULARIZED MULTI-VIEW UNSUPERVISED FEATURE SELECTION

Kennedy W. Wangila, Ke Gao, Pengfei Zhu, Qinghua Hu, Changqing Zhang

School of Computer Science and Technology, Tianjin Univerity, Tianjin 300350, China

ABSTRACT

The traditional learning machines suffer from the curse of dimensionality because of the data explosion in the areas of multi-media, social network, etc. Feature selection is an effective technique to reduce storage burden and time complexity, and improve generalization ability of the learned models. In real-world applications, the data can be collected from different modalities, or described from multi-views as well. Compared with supervised cases, it is more challenging to reduce the feature dimensionality of multi-view data in unsupervised circumstances. The key difficulty with multi-view unsupervised feature selection is how to characterize the multi-view relationships. In this paper, we propose a novel method for multi-view unsupervised feature selection by imposing sparsity on both individual features and views . To exploit the complementary information, we also take the view importance into consideration without introducing explicit view weights. Experiments on benchmark datasets show on the proposed algorithm outperforms other unsupervised feature selection methods

Index Terms— multi-view learning, unsupervised feature selection, group sparsity, parameter-free learning.

1. INTRODUCTION

With advent of information technology, there has been a tremendous increase in data accumulation derived from multiple sources. When users post photographs and other daily events on any social network, it is often accompanied by visual information and textual descriptions [1]. When a document is presented in a language unfamiliar to the reader, there is usually a need for translation into different languages for easy communication. In collection and analysis of complex biological data such as genetic codes, different genetic traits are harnessed and analyzed from different views [2]. For pattern recognition tasks, different features can be extracted from images or videos, and then a multi-view description is generated [3]. Thus, it is evident that in real-world applications, it is unavoidable to encounter data from multiple perspectives, which provides complementary information for the semantically same object [4].

In general, combination of different views provides us with better performances as opposed to single view [1]. However, data explosion leads to curse of dimensionality. Various phenomena that do not occur in low-dimensional settings arises when analyzing and organizing data in high-dimensional spaces [5][6]. Curse of dimensionality leads to huge storage requirements, increase in time complexity and subsequent failure in classic learning machines [7].

Reduction of dimensionality is one of the most effective ways to deal with the aforementioned problem. Feature selection directly selects a subset of relevant and most representative features. As opposed to feature selection, feature extraction projects the original high dimensional feature space to a new feature space with low dimensionality [2]. In comparison with feature extraction, feature selection models have better readability and interpretability, and can maintain physical meanings of original features. Therefore, feature selection is suitable for real world applications such as text mining and genetic analysis. Feature selection can be categorized into three cases, i.e., supervised, unsupervised and semi-supervised case, respectively [2][8] [9][10][11]. Importance of features is usually evaluated through correlation between features and class labels in supervised feature selection methods [12]. In reality, we often encounter plenty of information with unlabeled data. Since labels are quite expensive, unsupervised feature selection algorithms are developed to make use of all data points available in absence of class labels [12].

Without label information, unsupervised feature selection (UFS) is very challenging compared with the supervised case. Generally, there are three types of UFS methods, i.e., filter, wrapper, and embedding methods. For filter methods, indexes independent of any learning machine are proposed to evaluate single feature or subset of features. Laplacian score [8], SPEC [13] and MCFS [12] are the most common filter methods for unsupervised feature selection. They mainly emphasize the locality preserving ability of features from different perspectives. Wrapper methods search a subset of features that can achieve the optimal performance of certain learning machines. Although wrapper methods achieve promising performances, they are time-consuming and have poor generalization abilities. Embedding methods combine feature selection and model construction together without searching strategies. Pseudo labels are generated by spectral analysis [14], matrix factorization, or dictionary learning [11], and UFS is converted to a supervised problem. Features are then ranked according to the learned feature weight vector or matrix. Em-

bedding methods can incorporate different data or model priors into the model, and achieve superior results.

For multi-view tasks, traditional UFS algorithms do not exploit the relations among multiple views, and therefore multi-view unsupervised feature selection methods are developed. The main challenge for multi-view UFS is how to characterize the correlations of multi-views to boost the performance of feature selection. In [15], a distributed coding algorithm is proposed for multi-view unsupervised feature selection to accelerate object recognition and improve the performance. In [3], data similarity in different views are simultaneously considered by graph regularization and pseudo labels are generated by spectral clustering. In [16], features are selected for multi-view data in social media. Rather than select features together in all views, this work selects features in each view and then combines them for clustering or classification. The multi-view correlations are used by forcing all views to own the same cluster structure. In [17], based on prior that the text information is more discriminative than images in web news data, orthogonal nonnegative matrix factorization is used to learn pseudo labels on text data. In [18], subspace learning and graph manifold are combined together to learn the feature selection matrix for multi-view data. In [4], an online algorithm is proposed by extending EUFS [19] to multi-view data.

For multi-view learning, there are redundant and irrelevant features in each view. Some views with noisy information can degrade the overall performance of multi-view learning as well. Additionally, complementarity and individuality should be well modelled for multi-view unsupervised feature selection. Hence, for multi-view unsupervised feature selection, an effective algorithm should select relevant and representative features from multi-views by removing noisy features and views, and simultaneously combining the complementary view-specific information. In this paper, we aim to select features in unsupervised multi-view setting. The main contributions of this paper are summarized as follows:

- Mixed group sparsity regularization is imposed on the feature selection matrix to eliminate the effect of both outlier features and views.
- The view weights are learned by a parameter-free way to combine the complementary information from multi-views.
- Extensive experimental analysis on benchmark databases show that the proposed method outperform the existing UFS algorithms.

The rest of this paper is organized as follows: Section 2 presents the proposed multi-view unsupervised feature selection model. The experimental results are shown in section 3 and Section 4 concludes.

2. MULTI-VIEW UNSUPERVISED FEATURE SELECTION

Let $\mathbf{X}=[\mathbf{X}_1, \ldots, \mathbf{X}_i, \ldots, \mathbf{X}_m] \in \mathbb{R}^{d\times n}$, where each $\mathbf{X}_i \in \mathbb{R}^{d_i\times n}$ is the data matrix of view i, $d = \sum_{i=1}^{m} d_i$, d_i is the feature dimension of $\mathbf{X}_i$, and n is the number of data points. Unsupervised feature selection methods usually first calculate the sample similarity. Then, the multi-view feature selection problem can be formulated as:

$$\min_{\mathbf{P}^T\mathbf{P}=\mathbf{I}} \sum_{i=1}^{m} Tr(\mathbf{P}^T\mathbf{X}\omega_i\mathbf{L}_i\mathbf{X}^T\mathbf{P}) + \lambda R(\mathbf{P}) + \beta R(\mathbf{w}) \quad (1)$$

where $\mathbf{P} \in \mathbb{R}^{d\times r}$ is the feature weight matrix, $\mathbf{L}_i \in \mathbb{R}^{n\times n}$ and ω_i are the Laplacian matrix and weight of view i, respectively. λ and β are two positive scalar constants. $R(\mathbf{P})$ and $R(\mathbf{w})$ are the regularization items imposed on $\mathbf{P}$ and $\mathbf{w}$, respectively.

Parameter-free multi-view learning. To reduce the parameters of the model and avoid the specific design of $\mathbf{w}$, in this paper, we introduce a parameter-free multi-view learning strategy [20] with the following form:

$$\min_{\mathbf{P}^T\mathbf{P}=\mathbf{I}} \sum_{i=1}^{m} \sqrt{Tr(\mathbf{P}^T\mathbf{X}\mathbf{L}_i\mathbf{X}^T\mathbf{P})} + \lambda R(\mathbf{P}) \quad (2)$$

where no weight factors are explicitly defined for each view.

Mixed sparsity regularized learning In real-world applications, there are a large number of redundant, irrelevant, and noisy features. Additionally, if one view contains too much useless and noisy information, it will deteriorate the holistic performance. Hence, how to define $R(\mathbf{P})$ is very important for multi-view unsupervised feature selection. We propose a block-row sparsity regularizer and embed it into Eq. (2) to build a block-row sparse multi-view learning framework. On one hand, we use a block-sparsity regularizer to conduct view selection by keeping the important views and preserving the natural group structures of the data at the same time. On the other hand, we use a $\ell_{2,1}$-norm regularizer to conduct feature selection by selecting the important features from the important views.

Let $\mathbf{P} = [\mathbf{p}_1; ...; \mathbf{p}_j; ...; \mathbf{p}_d]$, where $\mathbf{p}_j$ is j^{th} row of $\mathbf{P}$. $\|\mathbf{p}_j\|_2$ can be used as the feature weight because it reflects the importance of the i^{th} feature in preserving the data structure. Therefore, the $\ell_{2,1}$-norm (i.e. $\|\mathbf{P}\|_{2,1}$) leads to the row-sparsity for all features.

To take the individuality of views into account, a block sparsity regularizer, i.e., $\sum_{i=1}^{m} \left\|\mathbf{P}^i\right\|_F$ is also introduced, where $\mathbf{P}^i$ is the i^{th} block (i.e., view) of matrix $\mathbf{P}$. The F-norm of $\mathbf{P}^i$ as follows:

$$\left\|\mathbf{P}^i\right\|_F = \sqrt{\sum_{j=1}^{d_i} \left\|\mathbf{p}_j^i\right\|_2^2} = \sqrt{\sum_{j=1}^{d_i}\sum_{k=1}^{r} (p_{jk}^i)^2} \quad (3)$$

where $\mathbf{p}_j^i$ is the j^{th} row of block $\mathbf{P}^i$, and p_{jk}^i is the k^{th} element of $\mathbf{p}_j^i$. Similar to l_{21}-norm sparsity, the impact of noisy views can be suppressed by the block-sparsity.

Optimization and algorithm. We model mixed sparsity regularized multi-view unsupervised feature selection (MSMFS) as:

$$\min \sum_{i=1}^{m} \sqrt{Tr(\mathbf{P}^T \mathbf{X} \mathbf{L}_i \mathbf{X}^T \mathbf{P})} + \lambda_1 \|\mathbf{P}\|_{2,1} + \lambda_2 \sum_{j=1}^{m} \left\|\mathbf{P}^j\right\|_F$$
$$s.t.\ \mathbf{P}^T \mathbf{P} = \mathbf{I} \tag{4}$$

where both λ_1 ($\lambda_1 > 0$), λ_2 ($\lambda_2 > 0$) are the parameters which control the sparsity of matrix $\mathbf{P}$. The Lagrange function of problem in Eq. (4) can be written as:

$$\left\{ \begin{array}{l} \sum_{i=1}^{m} \sqrt{Tr(\mathbf{P}^T \mathbf{X} \mathbf{L}_i \mathbf{X}^T \mathbf{P})} + \lambda_1 \|\mathbf{P}\|_{2,1} \\ + \lambda_2 \sum_{j=1}^{m} \left\|\mathbf{P}^j\right\|_F + G(\Lambda, F) \end{array} \right\} \tag{5}$$

where Λ is the Lagrange multiplier, and $\mathcal{G}(\Lambda, F)$ is the formalized term derived from the constraints. Taking the derivative of Eq. (5) w.r.t $\mathbf{P}$, we obtain:

$$\left\{ \begin{array}{l} \sum_{i=1}^{m} \alpha_i \frac{\partial Tr(\mathbf{P}^T \mathbf{X} \mathbf{L}_i \mathbf{X}^T \mathbf{P})}{\partial \mathbf{P}} + \lambda_1 \frac{\partial \|\mathbf{P}\|_{2,1}}{\partial \mathbf{P}} \\ + \lambda_2 \frac{\partial \sum_{j=1}^{m} \left\|\mathbf{P}^j\right\|_F}{\partial \mathbf{P}} + \frac{\partial G(\Lambda, F)}{\partial \mathbf{P}} \end{array} \right\} \tag{6}$$

where

$$\alpha_i = 1 \Big/ \left(2\sqrt{Tr(\mathbf{P}^T \mathbf{X} \mathbf{L}_i \mathbf{X}^T \mathbf{P})} \right) \tag{7}$$

Although we do not introduce a weight vector, α_i can be used to reflect the view-difference. By the derived vector $\mathbf{a}$, a multi-view graph is updated in each iteration, i.e., $\mathbf{L} = \sum_{\mathbf{i=1}}^{\mathbf{m}} \alpha_\mathbf{i} \mathbf{L_i}$. According to Eq. (7), α_i is dependent on the target variable $\mathbf{P}$ and Eq. (6) cannot be directly solved. But if we set α_i to be stationary, Eq. (6) can be considered as the solution to the following problem:

$$\sum_{i=1}^{m} \alpha_i Tr(\mathbf{P}^T \mathbf{X} \mathbf{L}_i \mathbf{X}^T \mathbf{P}) + \lambda_1 Tr(\mathbf{P}^T \mathbf{G} \mathbf{P}) + \lambda_2 Tr(\mathbf{P}^T \mathbf{H} \mathbf{P}) \tag{8}$$

which is easier to be solved. To update α_i and $\mathbf{P}$, we should take an alternating optimization strategy to compute α_i and $\mathbf{P}$ iteratively. $\mathbf{G}$ is a diagonal matrix with the k^{th} diagonal element

$$g_{kk} = \left\{ \begin{array}{ll} 0 & \text{if } \|\mathbf{p}_k\|_2 = 0 \\ \dfrac{1}{2\|\mathbf{p}_k\|_2} & \text{otherwise} \end{array} \right. \tag{9}$$

Algorithm 1 The algorithm of MSMFS

Input:

$\mathbf{X}$=$[\mathbf{X}_1, \ldots, \mathbf{X}_i, \ldots, \mathbf{X}_m] \in \mathbb{R}^{d \times n}$

1: Initialize $\mathbf{a}$, $\mathbf{G}$, and $\mathbf{H}$;
2: Compute the Laplacian matrix $\mathbf{L} = \sum_{\mathbf{i=1}}^{\mathbf{m}} \alpha_\mathbf{i} \mathbf{L_i}$;
3: **repeat**
4: Compute $\mathbf{P}$ by Eq. (11);
5: Update $\mathbf{G}$ by Eq. (9)
6: Update $\mathbf{H}$ by Eq. (10)
7: Update α_i by Eq. (7)
8: **until** Convergence criterion satisfied.

Output:

Feature selection matrix $\mathbf{P} \in \mathbb{R}^{d \times r}$

where $\mathbf{p}_k$ is the k^{th} row of $\mathbf{P}$, $k = 1, ..., d$. $\mathbf{H} = \text{diag}(\mathbf{H}_1, ..., \mathbf{H}_m)$, and each $\mathbf{H}_l$ is also diagonal with the l^{th} diagonal element as

$$h_{ll} = \left\{ \begin{array}{ll} 0 & \text{if } \left\|\mathbf{P}^i\right\|_F = 0 \\ \dfrac{1}{2\|\mathbf{P}^i\|_F} & \text{otherwise} \end{array} \right. \tag{10}$$

where $i = 1, ..., m$, and $l = 1, ...d_i$. Then the objective function can be written as:

$$\min Tr(\mathbf{P}^T (\mathbf{X} \sum_{i=1}^{m} (\alpha_i \mathbf{L}_i) \mathbf{X}^T + \lambda_1 \mathbf{G} + \lambda_2 \mathbf{H}) \mathbf{P})$$
$$s.t.\ \mathbf{P}^T \mathbf{P} = \mathbf{I} \tag{11}$$

To solve the optimization problem in Eq. (11), we can decompose each column $\mathbf{P}_{\cdot}, c$ of the matrix $\mathbf{P}$ into an optimization sub-problems by using the method of Lagrange multipliers. Then we derive that each column $\mathbf{P}_{\cdot}, c$ of the optimal $\mathbf{P}$ should satisfy the following condition:

$$(\mathbf{X} \sum_{i=1}^{m} (\alpha_i \mathbf{L}_i) \mathbf{X}^T + \lambda_1 \mathbf{G} + \lambda_2 \mathbf{H}) \mathbf{P}_{\cdot,c} = \theta_c \mathbf{P}_{\cdot,c} \tag{12}$$

where θ_c is the introduced Lagrange multiplier for the c^{th} column $\mathbf{P}_{\cdot}, c$. It can be seen that the optimization for $\mathbf{P}$ is transformed to a general eigenvalue problem.

We summarize MSMFS in Algorithm 1. We initialize $\alpha_i = \frac{1}{m}$, and $\mathbf{G}, \mathbf{H}$ as identity matrix. Laplacian matrix $\mathbf{L}_i$ is computed for each view. Then we can get $\mathbf{L} = \sum_{\mathbf{i=1}}^{\mathbf{m}} \alpha_\mathbf{i} \mathbf{L_i}$. The iteration will stop until the convergence criterion is satisfied.

Complexity and convergence. For MSMFS, an alternation minimization strategy is adopted and the main computation burden lies in the updating of the feature selection matrix $\mathbf{P}$ by Eq. (11). In each iteration, the time complexity of updating $\mathbf{P}$ is $O(d^3)$. Hence, the time complexity of MSMFS is $O(Td^3)$, where T is the iteration number. According to the convergence analysis in [10][20], it can be easily proved that the optimization problem in Eq. (2) can converge to the local optimum.

3. EXPERIMENTS

In this section, we will perform the proposed method on seven benchmark datasets, compared with five related unsupervised feature selection methods.

3.1. Datasets

Seven diverse publicly available datasets are selected for comparison, including Caltech10, Corel800, flickr, mfeat, PPMI, MSRA, Still DB. The statistics of the seven datasets are shown in Table 1. All the datasets contain at least 3 different views. The sample number of each dataset varies between 210 and 2000, and the feature dimension of each view between 6 and 680. Caltech10 has 800 images of objects belonging to 10 categories. Corel is a content-based dataset that consists of 800 images. The flickr dataset selects photos from Flickr as samples for common material categories. PPMI is a dataset containing 12 classes, i.e., humans interacting with 12 different musical instruments. MSRA is an image dataset contains 5000 images. We use a subset including 210 images. Multiple Features (mfeat) consists of handwritten numbers '0' to '9', with 200 patterns per class. Still DB is a still images dataset for action recognition.

DATA	Smaples	view	Features	Classes
Caltech10	800	4	$200, 512, 59, 680$	10
Corel800	800	4	$200, 512, 59, 680$	10
flickr	1000	4	$200, 512, 59, 680$	10
mfeat	2000	6	$216, 76, 64, 6, 240, 47$	10
PPMI	1400	3	$200, 200, 200$	7
MSRA	210	5	$1302, 512, 256, 210, 100$	7
Still DB	467	3	$200, 200, 200$	6

Table 1: Summary of the benchmark data sets.

3.2. Parameter setting

For all the comparison methods, including Laplacian Score [8], MCFS [13], UDFS [21] and AUMFS [3], the size of neighborhood k is set as 5 on all the datasets. For fair comparison, we tune the parameters using a grid-search strategy from $\{0.001, 0.01, 0.1, 1, 10, 100\}$ and record the best result. For the proposed method, both λ_1 and λ_2 are tuned by the grid-search strategy. For feature dimension, we set the number of features as $\{10, 30, ..., 150\}$ and report the average results over different dimensions. All the experiments are repeated for 20 times, and the average results are reported.

3.3. Comparison methods

We compare the proposed method MSMFS with following representative and state-of-the-art unsupervised feature selection methods:

Laplacian Score [8] **:** select features according to the power of locality preserving .

SPEC [13] **:** Spectral Feature Selection is a feature selection method using spectral clustering.

MCFS [12] **:** MCFS utilizes spectral clustering with ℓ_1-norm regularization.

UDFS [21] **:** Unsupervised Discriminative Feature Selection with $\ell_{2,1}$-norm regularization.

AUMFS [3] **:** Adaptive unsupervised multi-view feature selection uses multi-view information to select features.

3.4. Performance comparison

The clustering accuracy and NMI are shown in Table 2 and Table 3, respectively. From these results we can see that MSMFS achieves the best performance among all the competing methods. The superiority of out method may arise in the twofold: 1) the block sparsity and row sparsity constraints on the feature selection matrix $\mathbf{P}$ are incorporated in the final objective function, to choose the important views and select the most useful features from these important views; 2) MSMFS learns the view weights automatically without introducing explicit parameters. Complementary information from multi-views are then combined by learned weights.

DATA	Laplacian	SPEC	MCFS	UDFS	AUMFS	MSMFS
Caltech10	0.2562	0.2223	0.2873	0.2887	0.3205	**0.3444**
Corel800	0.2986	0.2514	0.2851	0.2702	0.2913	**0.3073**
flickr	0.2146	0.2086	**0.2369**	0.2262	0.2288	0.2360
mfeat	0.5608	0.6416	0.6242	0.6538	0.6129	**0.7105**
PPMI	0.1969	0.2180	0.1987	0.2005	0.1989	**0.2366**
MSRA	0.5099	0.4786	0.5390	0.5155	0.5110	**0.6746**
Still DB	0.3013	0.2857	0.3004	0.3017	**0.3124**	0.3004

Table 2: Clustering accuracy result of all data sets.

DATA	Laplacian	SPEC	MCFS	UDFS	AUMFS	MSMFS
Caltech10	0.1461	0.0962	0.1734	0.1767	0.2059	**0.2199**
Corel800	0.2198	0.1235	0.2255	0.1960	0.2302	**0.2400**
flickr	0.0993	0.1026	**0.1353**	0.1184	0.1309	0.1279
mfeat	0.5699	0.5960	0.6157	0.5983	0.5920	**0.6253**
PPMI	0.0224	0.0310	0.0255	0.0194	0.0238	**0.0461**
MSRA	0.4076	0.3902	0.4467	0.4100	0.4122	**0.5915**
Still DB	0.1019	0.0850	0.0930	0.0951	0.1035	**0.1051**

Table 3: Clustering NMI result of all data sets.

4. CONCLUSIONS

In this paper, we proposed a mixed sparsity regularized multi-view unsupervised feature selection (MSMFS) model. MSMFS incorporates the difference of multi-views into the model learning by automatically learning the view weights. Mixed group sparsity regularization is imposed to alleviate the effect of the outlier features and the views with noisy information. Experiments on benchmark datasets show that MSMFS outperforms the state-of-the-art single view and multi-view unsupervised feature selection algorithms.

5. REFERENCES

[1] Shiliang Sun, "A survey of multi-view machine learning," *Neural Computing and Applications*, vol. 23, no. 7-8, pp. 2031–2038.

[2] Jundong Li, Kewei Cheng, Suhang Wang, Fred Morstatter, Robert P Trevino, Jiliang Tang, and Huan Liu, "Feature selection: A data perspective," *arXiv preprint arXiv:1601.07996*, 2016.

[3] Yinfu Feng, Jun Xiao, Yueting Zhuang, and Xiaoming Liu, "Adaptive unsupervised multi-view feature selection for visual concept recognition," in *Asian Conference on Computer Vision*, 2012, pp. 343–357.

[4] Weixiang Shao, Lifang He, Chun-Ta Lu, Xiaokai Wei, and Philip S. Yu, "Online unsupervised multi-view feature selection," *CoRR*, vol. abs/1609.08286, 2016.

[5] R Bellman, "Dynamic programming: Princeton univ. press," 1957.

[6] Richard E Bellman, *Adaptive control processes: a guided tour*, Princeton university press, 2015.

[7] Lior Wolf and Amnon Shashua, "Feature selection for unsupervised and supervised inference: The emergence of sparsity in a weight-based approach," *Journal of Machine Learning Research*, vol. 6, no. Nov, pp. 1855–1887, 2005.

[8] Xiaofei He, Deng Cai, and Partha Niyogi, "Laplacian score for feature selection," in *Advances in neural information processing systems*, 2005, pp. 507–514.

[9] Khalid Benabdeslem and Mohammed Hindawi, "Efficient semi-supervised feature selection: constraint, relevance, and redundancy," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 5, pp. 1131–1143, 2014.

[10] Feiping Nie, Heng Huang, Xiao Cai, and Chris H Ding, "Efficient and robust feature selection via joint ?2, 1-norms minimization," in *Advances in neural information processing systems*, 2010, pp. 1813–1821.

[11] Pengfei Zhu, Qinghua Hu, Changqing Zhang, and Wangmeng Zuo, "Coupled dictionary learning for unsupervised feature selection," in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[12] Deng Cai, Chiyuan Zhang, and Xiaofei He, "Unsupervised feature selection for multi-cluster data," in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2010, pp. 333–342.

[13] Zheng Zhao and Huan Liu, "Spectral feature selection for supervised and unsupervised learning," in *Proceedings of the 24th international conference on Machine learning*. ACM, 2007, pp. 1151–1157.

[14] Zechao Li, Yi Yang, Jing Liu, Xiaofang Zhou, Hanqing Lu, et al., "Unsupervised feature selection using nonnegative spectral analysis.," in *AAAI*, 2012.

[15] C. M. Christoudias, R. Urtasun, and T. Darrell, "Unsupervised feature selection via distributed coding for multi-view object recognition.," in *IEEE International Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8.

[16] Jiliang Tang and Huan Liu, "An unsupervised feature selection framework for social media data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 12, pp. 2914–2927, 2014.

[17] Mingjie Qian and Chengxiang Zhai, "Unsupervised feature selection for multi-view clustering on text-image web news data," in *The Conference on Information and Knowledge Management*, 2014, pp. 1963–1966.

[18] Hong Shi, Yin Li, Yahong Han, and Qinghua Hu, "Cluster structure preserving unsupervised feature selection for multi-view tasks," *Neurocomputing*, vol. 175, no. PA, pp. 686–697, 2016.

[19] Suhang Wang, Jiliang Tang, and Huan Liu, "Embedded unsupervised feature selection," in *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[20] Feiping Nie, Jing Li, Xuelong Li, et al., "Parameter-free auto-weighted multiple graph learning: A framework for multiview clustering and semi-supervised classification," International Joint Conferences on Artificial Intelligence, 2016.

[21] Yi Yang, Heng Tao Shen, Zhigang Ma, Zi Huang, and Xiaofang Zhou, "$l_{2,1}$-norm regularized discriminative feature selection for unsupervised learning," in *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, 2011, vol. 22, p. 1589.