Adaptive Feature Redundancy Minimization

Rui Zhang
Arizona State University
Tempe, Arizona, U.S.A.
ruizhang8633@gmail.com

Hanghang Tong*
University of Illinois at Urbana-Champaign
Urbana, Illinois, U.S.A.
htong@illinois.edu

Yifan Hu
Yahoo! Research
Sunnyvale, California, U.S.A.
yifanh@gmail.com

ABSTRACT

Most existing feature selection methods select the top-ranked features according to certain criterion. However, without considering the redundancy among the features, the selected ones are frequently highly correlated with each other, which is detrimental to the performance. To tackle this problem, we propose a framework regarding *adaptive redundancy minimization* (ARM) for the feature selection. Unlike other feature selection methods, the proposed model has the following merits: (1) The redundancy matrix is adaptively constructed instead of presetting it as the priori information. (2) The proposed model could pick out the discriminative and non-redundant features via minimizing the global redundancy of the features. (3) ARM can reduce the redundancy of the features from both supervised and unsupervised perspectives.

KEYWORDS

adaptive learning, feature redundancy

ACM Reference Format:
Rui Zhang, Hanghang Tong, and Yifan Hu. 2019. Adaptive Feature Redundancy Minimization. In *The 28th ACM International Conference on Information and Knowledge Management (CIKM '19), November 3–7, 2019, Beijing, China.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3357384.3358112

1 INTRODUCTION

Faced with the abundant high-dimensional data, it is advantageous to extract the most useful and relevant information. In addition, high-dimensional data inevitably contain the noised dimensions, so that they are difficult to process directly. Accordingly, feature selection has become indispensable and been widely exploited in diverse domains, such as image segmentation in computer vision [5], and promoting the clustering performance in machine learning [18]. The purpose of feature selection is to select a subset of raw features for the subsequent classification or clustering tasks. Depending on whether the class label information is utilized or not, feature selection methods are categorized into three different types, namely, supervised feature selection, semi-supervised feature selection, and unsupervised feature selection. Above all, unsupervised feature selection is a more challenging task due to a lack of class label information. On the other hand, feature selection methods can also be classified into three different types: filter method [11], wrapper method [16], and embedded method [17]. Filter method selects the features according to the statistical property of the data, e.g., max variance of the data. Therefore, it has less computational complexity. Wrapper method is associated with the specific classifier and thus has better classification performance. Embedded method incorporates feature selection into the optimization problems (such as dimensionality reduction and clustering) with rational computational cost.

In general, the procedures of all the feature selection methods are similar to some extent. More specifically, certain criterion is employed to evaluate the score for each feature, such as fisher score [8], Laplacian score [11], and the sparse regularization [3]. Therefore, all the feature scores are sorted in the descending order and top-ranked features are selected for the subsequent tasks. For instance, fisher score (a supervised method) [8] computes a score for each feature via fisher criterion. Accordingly, it selects the top-ranked features with larger inter-class instance and smaller intra-class instance simultaneously. Multi-cluster feature selection (MCFS) [3] utilizes the joint learning of spectral regression and ℓ_1 sparse regularization to calculate the score of each feature such that the top-ranked features are selected. However, due to lack of the redundancy minimization among the features, the selected ones frequently have high correlation with each other.

In the previous study, Peng *et al.* recognized this problem and proposed the minimum redundancy maximum relevance (mRMR) [14] approach to minimize the redundancy among the selected features. However, mRMR serves as a greedy method to minimize the redundancy of the features, and thus the features are selected without minimizing the global redundancy. In the literature [10], to select the non-redundant features, a maximum information and minimum redundancy (MIMR) criterion is developed based on the entropy and mutual information. Since MIMR is optimized via the evolutionary algorithm, it lack a global perspective of redundancy minimization and has higher computational cost. To address the issue previously mentioned, we propose an effective feature selection method via adaptive redundancy minimization (ARM). To measure the redundancy among the features, the redundancy matrix is adaptively constructed and updated instead of presetting it as the priori information. Compared with other feature selection methods, the proposed model can minimize the redundancy of the features from the global perspective. More importantly, all the existing feature selection methods could incorporate the proposed ARM, such that the redundancy within the associated methods can be largely reduced.

*Hanghang Tong is the Corresponding Author. The work was partly done while Hanghang Tong was at Arizona State University.

CIKM '19, November 3–7, 2019, Beijing, China

ACM ISBN 978-1-4503-6976-3/19/11…$15.00
https://doi.org/10.1145/3357384.3358112

Figure 1: Intuitive illustration of ARM

2 PROBLEM FORMULATION

As we all know, if feature i and feature j are correlated with each other, it is much better to keep one and eliminate the other to reduce the redundancy of the features. Generally speaking, after reducing the redundancy of the features, the newly obtained features should be more representative and provide useful information to perform subsequent tasks. As for the proposed ARM model, its major contribution is to select the representative and non-redundant features. Most importantly, how can we build up a unified framework, which could largely reduce the redundancy of the features? To answer this question, we first construct the redundancy matrix $A \in \mathbb{R}^{d\times d}$ (d is the feature number) to measure the redundancy of all the features. Given the column vectors $s \in \mathbb{R}^d$ (the original feature score) and $z \in \mathbb{R}^d$ (the new feature score obtained by the proposed ARM), the raw ARM problem can then be formulated as

$$\begin{cases} \min\limits_{z^T\mathbf{1}_d=1, z\geq\mathbf{0}} \langle z,z\rangle_A \\ \max\limits_{z^T\mathbf{1}_d=1, z\geq\mathbf{0}} \langle z,s\rangle_{I_d}, \end{cases} \tag{1}$$

where the first term $\langle z,z\rangle_A = z^TAz$ of problem (1) denotes the global redundancy of features and should be minimized. Besides that, the second term $\langle z,s\rangle_{I_d} = z^Ts$ of problem (1) measures the consistency between feature score z and s and should be maximized. To solve the bi-objective ARM problem (1) as

$$\begin{cases} \min\limits_{z^T\mathbf{1}_d=1, z\geq\mathbf{0}} \langle z,z\rangle_A = \min\limits_{z^T\mathbf{1}_d=1, z\geq\mathbf{0}} g(z) \\ \max\limits_{z^T\mathbf{1}_d=1, z\geq\mathbf{0}} \langle z,s\rangle_{I_d} = \max\limits_{z^T\mathbf{1}_d=1, z\geq\mathbf{0}} h(z), \end{cases} \tag{2}$$

we could utilize a weighted model as

$$\min_{z^T\mathbf{1}_d=1, z\geq\mathbf{0}, \lambda} \lambda g(z) - h(z), \tag{3}$$

where λ serves as the weight. However, problem (3) leads to the trivial solution w.r.t. λ as $\frac{\partial(\lambda g(z)-h(z))}{\partial\lambda} = 0 \Rightarrow g(z) = 0$ due to the linearity of the weight. To avoid the above trivial case, we can modify the model (3) via a simple yet effective technique. Specifically, problem (2) can be reformulated into the following quadratic weighted optimization (QWO) model as

$$\min_{z^T\mathbf{1}_d=1, z\geq\mathbf{0}, \lambda} \lambda(\lambda g(z) - h(z)), \tag{4}$$

where an additional weight λ is multiplied.

To adaptively evaluate the redundancy matrix A, the bi-objective problem (1) can be further reformulated into the ARM model as

$$\begin{aligned} &\min_{\lambda,z,A} \lambda\left(\lambda\left(\langle z,z\rangle_A + \alpha Tr\left(A\right)\right) - \langle z,s\rangle_{I_d}\right) \\ &s.t.\ z^T\mathbf{1}_d = 1, z\geq\mathbf{0}, A\succ\mathbf{0}, Tr\left(A^{-1}\right) = 1, \end{aligned} \tag{5}$$

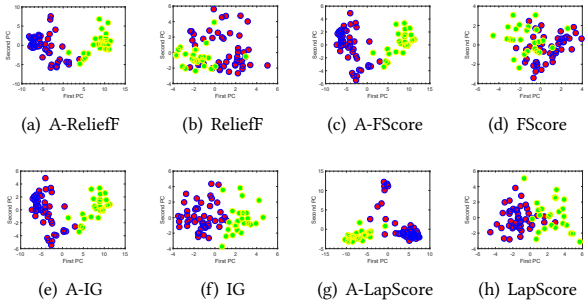


Figure 2: Projecting top 80 features to the first two principle components via PCA, where A- denotes that raw score is refined by ARM.

where the matrix $A \in \mathbb{R}^{d\times d}$ is optimized to measure the redundancy of all the features[1], λ is a self-adaptive parameter to balance the first term and second term, and $\alpha > 0$ is a regularization parameter. Note that λ serves as a leverage to incorporate the two terms in (1) into an optimization problem and the regularization $Tr\left(A\right)$ is added to prevent the potential trivial solution of the variable A.

Besides that, the mechanism of the proposed ARM is illustrated in Figure 1. In Figure 1, four features has been sorted by certain feature score. The first feature is correlated with the second and third features, but the second and third features are uncorrelated with each other, which is measured by the adaptive redundancy matrix A, namely, the deeper the color is, the more redundancy the features are correlated with each other. Via the proposed ARM, the ranking of the first feature will descend, and the ranking of the second and third features will ascend. In this case, the representative features, i.e., the second and third features can be selected.

3 OPTIMIZATION

In this section, we attempt to solve problem (5) by an iterative optimization algorithm.

Update A: When λ and z are fixed, problem (5) becomes

$$\begin{aligned} &\min_{A}\ z^TAz + \alpha Tr\left(A\right) \\ &s.t.\ A\succ\mathbf{0}, Tr\left(A^{-1}\right) = 1. \end{aligned} \tag{6}$$

According to the Cauchy-Schwartz inequality and constraint $Tr\left(A^{-1}\right) = 1$, we could infer that

$$\begin{aligned} &z^TAz + \alpha Tr\left(A\right) \\ &= Tr(A^{\frac{1}{2}}\left(zz^T + \alpha I_d\right)^{\frac{1}{2}}\left(zz^T + \alpha I_d\right)^{\frac{1}{2}}A^{\frac{1}{2}})Tr(A^{-\frac{1}{2}}A^{-\frac{1}{2}}) \\ &\geq \left(Tr\left(zz^T + \alpha I_d\right)^{\frac{1}{2}}\right)^2. \end{aligned} \tag{7}$$

With an arbitrary constant γ, the above equality holds if and only if

$$\gamma\left(zz^T + \alpha I_d\right)^{\frac{1}{2}}A^{\frac{1}{2}} = A^{-\frac{1}{2}} \Rightarrow \gamma\left(zz^T + \alpha I_d\right)^{\frac{1}{2}} = A^{-1}. \tag{8}$$

[1]In practice, we initialize A by the square of cosine similarity between each feature pair.

Table 1: Classification accuracy of top 80 features

Datasets	ReliefF	**A-ReliefF**	FScore	**A-FScore**	IG	**A-IG**
COIL20	0.7791	**0.9403**	0.8431	0.9333	0.8431	0.9333
USPS	0.8524	**0.9008**	0.8701	0.8922	0.8701	0.8950
ORL	0.7550	0.8400	0.7650	**0.8550**	0.7650	0.8550
ISOLET	0.7821	0.8744	0.7833	**0.8782**	0.7833	0.8782
LEU	0.9444	0.9591	0.9387	0.9590	0.9388	**0.9600**
AR10P	0.5385	**0.8154**	0.5539	0.7846	0.5538	0.7538

Table 2: Datasets Summary

Data Name	COIL20	USPS	ORL	ISOLET	LEU	AR10P
# Samples	1440	9298	400	1560	72	130
# Features	1024	256	1024	617	3571	2400
# Classes	20	10	40	26	2	10

Under constraint $Tr\left(A^{-1}\right) = 1$, we can obtain $\gamma = \frac{1}{Tr\left(\left(zz^T+\alpha I_d\right)^{\frac{1}{2}}\right)}$, such that the optimal A is achieved as

$$A = Tr\Big(\Big(zz^T + \alpha I_d\Big)^{\frac{1}{2}}\Big)\Big(zz^T + \alpha I_d\Big)^{-\frac{1}{2}}. \tag{9}$$

Apparently, the obtained A in Eq. (9) is positive definite and the constraint $A \succ \mathbf{0}$ is naturally satisfied.

Update λ: When z and A are fixed, problem (5) becomes

$$\min_{\lambda} \lambda^2 \left(z^T Az + \alpha Tr\left(A\right)\right) - \lambda z^T s. \tag{10}$$

By taking the derivative of problem (10) with respect to λ, and setting it to zero, we have

$$\lambda = \frac{z^T s}{2\left(z^T Az + \alpha Tr\left(A\right)\right)}. \tag{11}$$

Update z: When λ and A are fixed, problem (5) is simplified into

$$\begin{aligned} &\min_{z} \lambda^2 \left(z^T Az\right) - \lambda z^T s \\ &s.t.\ z^T \mathbf{1}_d = 1, z \geq \mathbf{0}. \end{aligned} \tag{12}$$

To effectively solve this problem, we introduce an auxiliary variable $v \in \mathbb{R}^d$, and rewrite problem (12) as

$$\min_{z^T \mathbf{1}_d = 1, z \geq \mathbf{0}, z = v} \lambda^2 \left(v^T Az\right) - \lambda v^T s. \tag{13}$$

By using Augmented Lagrangian Multiplier (ALM) [1] method (more details can refer to [4]), problem (13) is reformulated into

$$\min_{z^T \mathbf{1}_d = 1, z \geq \mathbf{0}, v} \lambda^2 \left(v^T Az\right) - \lambda v^T s + \frac{\mu}{2}\left\|z - v + \frac{\beta}{\mu}\right\|_2^2, \tag{14}$$

where $\mu > 0$ is the penalty parameter to control the violation of equality constraint, and $\beta \in \mathbb{R}^d$ is the Lagrangian multipliers. Subproblem (14) now contains two variables v and z and it can also be solved by iterative optimization strategy. When z is fixed, we take the derivative of subproblem (14) with respect to v and set it to zero as

$$v = z + \frac{\lambda s + \beta - \lambda^2 A^T z}{\mu}. \tag{15}$$

When v is fixed, subproblem (14) becomes

$$\min_{z^T \mathbf{1}_d = 1, z \geq \mathbf{0}} \lambda^2 \left(v^T Az\right) + \frac{\mu}{2}\left\|z - v + \frac{\beta}{\mu}\right\|_2^2. \tag{16}$$

Algorithm 1 The algorithm to solve problem (5).

Input: Feature score $s \in \mathbb{R}^d$, and the selected feature number t.
Initialize: Feature score $z \in \mathbb{R}^d$, redundancy matrix $A \in \mathbb{R}^{d\times d}$, parameters α, μ and the Lagrangian multipliers $\beta \in \mathbb{R}^d$.
while not converge **do**
1. Update λ by Eq. (11).
2. Using ALM method to solve problem (12), update v by Eq. (15), update z by solving problem (18), and update parameters μ and β of ALM, respectively.
3. Update A by Eq. (9).
end while
Output: Optimal z, then sort in descending order, and select top-ranked t features.

By merging this two terms, we rewrite problem (16) as the following equivalent form

$$\min_{z^T \mathbf{1}_d = 1, z \geq \mathbf{0}} \frac{\mu}{2}\left\|z - v + \frac{1}{\mu}\left(\beta + \lambda^2 A^T v\right)\right\|_2^2, \tag{17}$$

which can be further rewritten as

$$\min_{z^T \mathbf{1}_d = 1, z \geq \mathbf{0}} \frac{1}{2}\left\|z - m\right\|_2^2, \tag{18}$$

where $m = v - \frac{1}{\mu}\left(\beta + \lambda^2 A^T v\right)$. Problem (18) can be solved by [7] or an efficient algorithm in [12]. According to [4], we update the parameters μ and β of ALM, respectively.

Based on the analysis above, we summarize the detailed procedure for solving problem (5) in Algorithm 1.

4 EXPERIMENTS

In this section, we conduct the extensive experiments on six benchmark datasets including COIL20, USPS, ORL, ISOLET, LEU, and AR10P, which are briefly summarized in Table 2. As for α, it is searched in the grid of $\left\{10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 10^2, 10^3\right\}$ and parameters μ and β in the ALM are updated as in [4]. Particularly, **A-** denotes that ARM is applied to refining the raw score.

4.1 Visualization on Principle Components

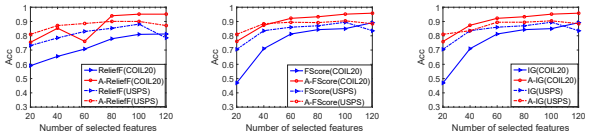
To visualize the feature redundancy, we select top 80 features from LEU dataset to perform the principle component analysis (PCA) [6], and plot the first two principle components for the selected features in Figure 2. In general, the projected features become much far away from each other after using ARM model. For example, the originally projected features by ReliefF [13] are tangled together in Figure 2. (b), but the newly projected features by A-ReliefF are much far away from each other in Figure 2. (a). That is to say, the selected features have less correlation via the proposed ARM model.

4.2 The Performance of Feature Selection by ARM

Classification result comparison: For the supervised feature selection, we use ReliefF [13], Fisher Score (FScore) [8], and Information Gain (IG) [15] as original methods. After using ARM model, we denote the corresponding methods as A-ReliefF, A-FScore, and

Table 3: Clustering accuracy and NMI of top 80 features

	datasets	LapScore	**A-LapScore**	MCFS	**A-MCFS**	UDFS	**A-UDFS**	LLCFS
Accuracy	COIL20	0.5631	0.6410	0.6298	0.6451	0.2187	**0.6521**	0.5875
	USPS	0.5944	0.6154	0.6540	0.6470	0.3428	**0.6729**	0.5096
	ORL	0.4500	0.5200	0.4975	0.5225	0.3750	**0.5450**	0.4775
	ISOLET	0.5109	0.5397	0.5256	**0.5961**	0.3109	0.5097	0.5583
	LEU	0.8772	0.8750	0.8611	**0.9027**	0.7639	0.8194	0.8611
	AR10P	0.2923	**0.3231**	0.2154	0.2308	0.2615	0.2693	0.2308
	Average	0.5430	0.5857	0.5640	0.5907	0.3788	0.5780	0.5375
NMI	COIL20	0.5631	0.6410	0.6298	0.6451	0.2187	**0.6521**	0.5875
	COIL20	0.6705	0.7261	0.7349	0.7349	0.2928	**0.7432**	0.6946
	USPS	0.5530	0.5417	0.5845	0.5880	0.2652	**0.6213**	0.5707
	ORL	0.6664	0.7122	0.7102	**0.7271**	0.5935	0.7127	0.6825
	ISOLET	0.6691	0.6788	0.6940	**0.7099**	0.4599	0.6554	0.6909
	LEU	0.4876	0.4222	0.5123	**0.5471**	0.2227	0.2694	0.5123
	AR10P	**0.3205**	0.3142	0.1768	0.2438	0.2249	0.3123	0.1462
	Average	0.5612	0.5659	0.5687	0.5918	0.3432	0.5532	0.5495



(a) A-ReliefF (b) A-FScore (c) A-IG

Figure 3: Supervised classification accuracy on COIL20 and USPS datasets with different number of selected features.

A-IG, respectively. As a convention in [2], we use classification accuracy (the proportion of correctly predicted labels) as evaluation metric, and SRDA [2] as the basic classifier. Half of the data are utilized as the training set, and the remaining part are set as the test set in the experiment. We summarize the classification accuracy for supervised feature selection in Table 1, where the best results are in bold face and the second-best results are underlined. From Table 1, we conclude that the classification accuracy is largely improved by using ARM framework. The best classification accuracy via A-ReliefF is increased around 27% in AR10P dataset.

In addition, the experiments with different number of selected features are also performed on two datasets (i.e. COIL20 and USPS), and the results are shown in Figure 3. It is noted that the classification accuracy is consistently increased by using ARM compared with the original methods. This is due to the fact that the redundancy of the selected features are largely decreased, and the selected features become more representative and discriminative.

Clustering result comparison: Similar to the supervised feature selection, we use Laplacian Score (Lap) [11], MCFS [3], and UDFS [18] as original methods. After using ARM model, we denote the corresponding methods as A-Lap, A-MCFS, and A-UDFS, respectively. As in [18] , we use clustering accuracy and NMI [9] as the basic evaluation metrics, and k-means (repeating 20 times) to perform the clustering task.

We summarize the clustering accuracy and NMI in Table 3, respectively, where best results are bolded, and the second-best results are underlined. From Table 3, it is noted that both the clustering accuracy and NMI on most datasets are significantly improved by A-LapScore and A-UDFS. As for MCFS, its improvement is not as obvious as A-LapScore and A-UDFS methods, because MCFS tends to selecting the features with less correlation.

5 CONCLUSION

To effectively reduce the negative impact of the redundancy among the selected features, we propose a practical framework ARM for both supervised and unsupervised feature selection. Unlike other feature selection methods, we adaptively construct the redundancy matrix to measure the redundancy of the features instead of presetting it as the priori information. Via the redundancy matrix and the re-sorted feature score, we can largely reduce the redundancy of the selected features from the global perspective, such that more representative and non-redundant features can be selected. Consequently, the experimental results on six benchmark datasets validate the effectiveness of the proposed ARM framework.

ACKNOWLEDGEMENT

This work is supported by NSF (IIS-1651203, IIS-1715385), and DHS (2017-ST-061-QA0001).

REFERENCES

[1] Dimitri P Bertsekas. 1982. Constrained optimization and Lagrange multiplier methods. (1982), 383–392.
[2] Deng Cai, Xiaofei He, and Jiawei Han. 2008. SRDA: An Efficient Algorithm for Large-Scale Discriminant Analysis. *IEEE Trans. Knowl. Data Eng.* 20, 1 (2008), 1–12.
[3] Deng Cai, Chiyuan Zhang, and Xiaofei He. 2010. Unsupervised feature selection for multi-cluster data. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, Dc, Usa, July*. 333–342.
[4] Xiao Cai, Feiping Nie, and Heng Huang. 2013. Exact top-k feature selection via $\ell_{2,0}$-norm constraint. In *International Joint Conference on Artificial Intelligence*. 1240–1246.
[5] Ming Ming Cheng, Yun Liu, Qibin Hou, Jiawang Bian, Philip Torr, Shi Min Hu, and Zhuowen Tu. 2016. *HFS: Hierarchical Feature Selection for Efficient Image Segmentation.* Springer International Publishing. 867–882 pages.
[6] Hans Peter Deutsch. 2002. *Principle Component Analysis.* Palgrave Macmillan UK.
[7] Duchi, John, ShalevShwartz, Shai, Singer, Yoram, Chandra, and Tushar. 2008. Efficient projections onto the ℓ_1-ball for learning in high dimensions. In *International Conference on Machine Learning*. 272–279.
[8] Richard O. Duda and Peter E. Hart. 1973. *Pattern classification and scene analysis.* Wiley.
[9] P. A Estevez, M Tesmer, C. A Perez, and J. M Zurada. 2009. Normalized Mutual Information Feature Selection. *Neural Networks IEEE Transactions on* 20, 2 (2009), 189–201.
[10] Jie Feng, Licheng Jiao, Fang Liu, Tao Sun, and Xiangrong Zhang. 2016. Unsupervised feature selection based on maximum information and minimum redundancy for hyperspectral images. *Pattern Recognition* 51 (2016), 295–309.
[11] Xiaofei He, Deng Cai, and Partha Niyogi. 2005. Laplacian score for feature selection. In *International Conference on Neural Information Processing Systems*. 507–514.
[12] Jin Huang, Feiping Nie, and Heng Huang. 2015. A new simplex sparse learning model to measure data similarity for clustering. In *International Conference on Artificial Intelligence*. 3569–3575.
[13] Igor Kononenko. [n. d.]. Estimating Attributes: Analysis and Extensions of RELIEF. In *Machine Learning: ECML-94, European Conference on Machine Learning, Catania, Italy, April 6-8, 1994*. 171–182.
[14] Hanchuan Peng, Fuhui Long, and Chris Ding. 2005. Feature Selection Based on Mutual Information: Criteria of Max-Dependency, Max-Relevance, and Min-Redundancy. *IEEE Transactions on Pattern Analysis & Machine Intelligence* 27, 8 (2005), 1226–1238.
[15] Laura Elena Raileanu and Kilian Stoffel. 2004. Theoretical Comparison between the Gini Index and Information Gain Criteria. *Annals of Mathematics and Artificial Intelligence* 41, 1 (2004), 77–93.
[16] Sina Tabakhi, Parham Moradi, and Fardin Akhlaghian. 2014. An unsupervised feature selection algorithm based on ant colony optimization. *Engineering Applications of Artificial Intelligence* 32, 6 (2014), 112–123.
[17] Suhang Wang, Jiliang Tang, and Huan Liu. 2015. Embedded Unsupervised Feature Selection. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA*. 470–476.
[18] Yi Yang, Heng Tao Shen, Zhigang Ma, Zi Huang, and Xiaofang Zhou. 2011. $\ell_{2,1}$-norm regularized discriminative feature selection for unsupervised learning. In *International Joint Conference on Artificial Intelligence*. 1589–1594.