Nonlinear Feature Selection Algorithm Based on Local Structure Learning (Postprint)

Authors: Li Jiaye, Zhang Leyuan, Lei Cong, Gan Jiangzhang, Lü Zhizheng

Date: 2018-12-13T00:00:00+00:00

Abstract

To address the characteristics of most high-dimensional data, which exhibit not only similarity but also nonlinear relationships, we propose a nonlinear attribute selection algorithm based on local structure learning. The algorithm first maps the data to a high-dimensional space via a kernel function to represent the nonlinear relationships among data attributes in this high-dimensional space; then, in the low-dimensional space, it thoroughly exploits the similarity among attributes through local structure learning, while simultaneously employing low-rank constraints to mitigate noise interference; finally, it performs attribute selection through a sparse regularization factor, uses kernel function mapping to identify nonlinear relationships among data attributes, and employs local structure learning to identify similarity among data attributes. This algorithm is a nonlinear attribute selection algorithm embedded with local structure learning. Experimental results demonstrate that the proposed algorithm achieves superior performance compared to other baseline algorithms.

Full Text

Preamble

Nonlinear Feature Selection Algorithm via Local Structure Learning

Li Jiaye†, Zhang Leyuan, Lei Cong, Gan Jiangzhang, Lyu Zhizheng
(Guangxi Key Lab of Multi-source Information Mining & Security, Guangxi Normal University, Guilin, Guangxi 541004, China)

Abstract: Most high-dimensional data exhibit not only similarities but also nonlinear relationships among attributes. To address this characteristic, this paper proposes a nonlinear feature selection algorithm based on local structure learning. The algorithm first maps data to a high-dimensional space through kernel functions to express nonlinear relationships between attributes in that space. It then exploits similarity among attributes in the low-dimensional space via local structure learning while simultaneously eliminating noise interference through low-rank constraints. Finally, feature selection is performed using a sparse regularization term. This algorithm integrates local structure learning into a nonlinear feature selection framework. Experimental results demonstrate that the proposed algorithm achieves superior performance compared to other benchmark algorithms.

Key words: feature selection; kernel function; low-rank; local structure learning; sparse regularization

---

0 Introduction

The advent of the information age, driven by advances in computer science and technology, has brought about massive volumes of high-dimensional data [?]. Concurrently, fields such as artificial intelligence and data mining have flourished. Processing thousands of features becomes extremely difficult, often leading to the "curse of dimensionality" [?, ?]. Effective preprocessing of these high-dimensional datasets is essential, and feature selection represents one of the most effective approaches [?, ?] for reducing dimensionality.

Feature selection methods [?] encompass both linear and nonlinear approaches, with the fundamental goal of identifying a relatively small yet representative subset of attributes. While numerous feature selection methods exist [?], most fail to capture nonlinear relationships among data attributes. Local structure learning was initially applied to samples, constructing similarity matrices to capture inter-sample structures [?] and achieving favorable results. However, it cannot adequately represent structural relationships among attributes. To address this limitation, we map each attribute to a high-dimensional space using kernel functions, enabling linear separability of nonlinear relationships in that space while simultaneously applying local structure learning to attributes to better represent their local structural relationships in the low-dimensional space. This yields a more effective feature selection algorithm termed the Nonlinear Feature Selection algorithm via Local Structure Learning (LS_NFS).

Our approach first processes data through kernel functions to obtain kernel matrices, thereby overcoming the limitation of linear feature selection. Second, we construct similarity matrices for data attributes to perform local structure learning, which improves classification accuracy. The model incorporates low-rank constraints to exclude noise interference. Finally, we embed an $\ell_{2,1}$-norm for sparse learning and feature selection. By simultaneously considering both nonlinear relationships and similarities among attributes, our method demonstrates superior performance compared to single linear feature selection approaches. Experimental validation confirms that the algorithm achieves excellent classification accuracy.

1.1 Kernel Functions

Kernel functions were introduced to machine learning long ago, mitigating the "curse of dimensionality" and significantly reducing computational complexity. Any symmetric function that produces a positive semi-definite kernel matrix on arbitrary datasets qualifies as a kernel function. With diverse forms and parameters, different kernel functions and parameter settings alter the mapping from low-dimensional to high-dimensional space, thereby affecting the properties of the high-dimensional space and ultimately changing kernel performance.

Commonly used kernel functions include Gaussian, polynomial, perceptron, and spline kernels. Since the Gaussian kernel generally exhibits superior performance in most scenarios compared to the other three, our algorithm employs the Gaussian kernel:

$$k(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right)$$

where x_i and x_j represent the i-th and j-th attributes, respectively, and σ is the width parameter controlling the radial scope of the function.

Consequently, we utilize the Gaussian kernel function to map data attributes into kernel space, thereby thoroughly mining nonlinear relationships among attributes.

1.2 Introduction to Local Structure Learning

Previous research has demonstrated that constructing local structures among data can achieve dimensionality reduction [**?**]. Therefore, we perform local structure learning by constructing similarity matrices among attributes in low-dimensional space.

Given a sample dataset $\mathbf{X} \in \mathbb{R}^{n\times d}$, where n and d denote the number of samples and attributes respectively, local structure learning yields:

$$\min_{\mathbf{W}} \sum_{i=1}^{d} \|x_i - \sum_{j=1}^{d} s_{ij} x_j\|_2^2$$

where x_i represents the i-th attribute, $\mathbf{W}$ is a transformation matrix in low-dimensional space, and s_{ij} is an element of matrix $\mathbf{S}$ indicating similarity between attributes x_i and x_j. If attribute x_j is the k-nearest neighbor of attribute x_i, s_{ij} is computed via the Gaussian kernel function in equation (1); otherwise, $s_{ij} = 0$.

2.1 Algorithm Description

We first decompose dataset $\mathbf{X}$ into d column vectors, each vector $x_i \in \mathbb{R}^{n\times 1}$; then treat each element in x_i as an independent sample $x_{ij} \in \mathbb{R}$ for $j = 1, ..., n$; and project them into kernel space to obtain kernel matrices. Thus, the original $\mathbf{X}$ becomes d kernel matrices $\mathbf{K}_i$.

Unsupervised feature selection aims to identify more representative attributes. Without class labels $\mathbf{Y}$, using the data matrix $\mathbf{X}$ as a response matrix better preserves the internal structure of original features [**?**, **?**].

To thoroughly mine nonlinear relationships among attributes, we derive:

$$\min_{\alpha} \sum_{i=1}^{d} \|\mathbf{K}_i \alpha_i - \mathbf{X}\|_F^2$$

where α performs feature selection, equivalent to an attribute weight vector; α_i corresponds to an element of vector α; $\mathbf{K}_i$ is the kernel coefficient matrix; and $\mathbf{K}_i$ is the kernel matrix.

Since the similarity matrix $\mathbf{S}$ is highly sensitive to parameter σ, to reduce parameter tuning while learning a more effective similarity matrix, we alternately perform structure learning and low-dimensional space learning to achieve optimal results. Specifically:

$$\min_{\mathbf{W},\mathbf{S}} \sum_{i=1}^{d} \|x_i - \sum_{j=1}^{d} s_{ij} x_j\|_2^2 + \lambda_1 \sum_{i=1}^{d} \|w_i\|_2^2 + \lambda_2 \sum_{i=1}^{d} \|s_i\|_2^2$$

subject to:

$$\sum_{j=1}^{d} s_{ij} = 1, \quad s_{ij} \geq 0, \quad s_{ij} = 0 \text{ if } j \notin \mathcal{N}_i$$

where λ_1, λ_2 are tuning parameters, s_i is the i-th column of similarity matrix $\mathbf{S}$, and $\mathcal{N}_i$ represents the neighbor set of the i-th attribute. This formulation ensures larger s_{ij} values for closer attributes and smaller values for distant ones.

To exclude outlier interference and remove noisy samples [**?**], we impose low-rank constraints on matrix $\mathbf{W}$, i.e., $\min \text{rank}(\mathbf{W})$, while applying orthogonal constraints to matrix $\mathbf{A}$ to fully consider correlations among output variables, and adding an $\ell_{2,1}$-norm for sparse learning and feature selection. The final objective function becomes:

$$\min_{\mathbf{A},\mathbf{B},\mathbf{S},\alpha} \sum_{i=1}^{d} \|\mathbf{K}_i \alpha_i - \mathbf{X}\mathbf{A}\mathbf{B}\|_F^2 + \lambda_1 \|\mathbf{S}\|_F^2 + \lambda_2 \|\mathbf{A}\|_{2,1} + \lambda_3 \text{rank}(\mathbf{W})$$

where $\lambda_1, \lambda_2, \lambda_3$ are tuning parameters. The kernel matrix $\mathbf{K}$ is computed via the Gaussian kernel function, primarily mapping data to kernel space to mine nonlinear relationships among attributes. The final $\ell_{2,1}$-norm term sparsifies α to perform feature selection. If an element in vector α equals zero, the corresponding attribute is not selected.

The proposed algorithm offers three advantages:
a) Conventional feature selection algorithms can only identify linear relationships among attributes, failing to discover nonlinear relationships. Our algorithm maps each attribute through kernel functions to create separate kernel matrices, thoroughly mining complex nonlinear relationships among attributes.
b) Unlike standard local structure learning that computes optimal similarity relationships among samples, our algorithm targets attributes and alternates similarity matrix learning with low-dimensional space learning to achieve optimal feature selection.
c) Low-rank constraints significantly reduce computational complexity while representing data redundancy. Noisy samples increase the rank of the coefficient matrix; low-rank constraints mitigate noise interference and consider global data structure, thereby improving efficiency and classification accuracy.

Algorithm 1: LS_NFS Algorithm
Input: Training samples $\mathbf{X} \in \mathbb{R}^{n \times d}$, control parameters $\lambda_1, \lambda_2, \lambda_3$
Output: Classification accuracy

1. Derive class indicator matrix from training samples
2. Construct kernel matrix $\mathbf{K}_i$ for each attribute via equations (1) and (3)
3. Construct structural similarity matrix $\mathbf{S}$ among attributes via equation (6)
4. Solve for global optimal solution using Algorithm 3 to obtain feature selection vector α^*
5. Perform feature selection on original attribute set $\mathbf{X}$ using optimal solution α^* to obtain new attribute set
6. Classify samples with new attribute set using SVM

2.2 Algorithm Optimization

Since the objective function is not jointly convex, a closed-form solution cannot be obtained directly. Therefore, we propose an alternating iterative optimization method with four steps:

1) Fix $\mathbf{S}, \mathbf{B}, \alpha$, optimize $\mathbf{A}$:
With $\mathbf{S}, \mathbf{B}, \alpha$ fixed, optimization problem (8) becomes:

$$\min_{\mathbf{A}} \|\mathbf{X}-\mathbf{AB}\|_F^2 + \lambda_2\|\mathbf{A}\|_{2,1} \quad \text{s.t.} \quad \mathbf{A}^T\mathbf{A}=\mathbf{I}$$

Since each data attribute has neighbors, we sort each s_i in descending order. Equation (9) can be rewritten as:

$$\min_{\mathbf{A}} \operatorname{tr}(\mathbf{A}^T\mathbf{X}^T\mathbf{LXA}) + \lambda_2\|\mathbf{A}\|_{2,1}$$

where $\operatorname{tr}(\cdot)$ denotes matrix trace and $\mathbf{L}$ is a Laplacian matrix. $\mathbf{Q}$ is a diagonal matrix with elements $q_{ii} = \sum_{j=1}^{d} s_{ij}$. Due to orthogonal constraints on $\mathbf{A}$, we optimize it using the method from reference [**?**].

2) Fix $\mathbf{A}, \mathbf{S}, \alpha$, optimize $\mathbf{B}$:
With $\mathbf{A}, \mathbf{S}, \alpha$ fixed, problem (8) becomes:

$$\min_{\mathbf{B}} \|\mathbf{XA}-\mathbf{B}\|_F^2 + \lambda_1\|\mathbf{B}\|_F^2$$

Taking the derivative of $\mathbf{B}$ and setting it to zero yields:

$$\mathbf{B} = (\mathbf{A}^T\mathbf{X}^T\mathbf{XA}+\lambda_1\mathbf{I})^{-1}\mathbf{A}^T\mathbf{X}^T\mathbf{X}$$

3) Fix $\mathbf{A}, \mathbf{B}, \alpha$, optimize $\mathbf{S}$:
With $\mathbf{A}, \mathbf{B}, \alpha$ fixed, problem (8) becomes:

$$\min_{\mathbf{S}} \sum_{i=1}^{d} \|x_i - \sum_{j=1}^{d} s_{ij}x_j\|_2^2 + \lambda_1 \sum_{i=1}^{d} \|s_i\|_2^2$$

subject to the same constraints as above. This is equivalent to Euclidean projection of $\mathbf{Z}$ onto a convex set. Due to its separable form, we optimize each s_i individually:

$$\min_{s_i} \|s_i + \frac{1}{2\lambda_1} z_i\|_2^2 \quad \text{s.t.} \quad s_i^T\mathbf{1}=1, \quad s_{ij} \geq 0, \quad s_{ij}=0 \text{ if } j \notin \mathcal{N}_i$$

where $\mathbf{Z} = \mathbf{X}^T\mathbf{X}$ and z_i is the i-th column of $\mathbf{Z}$. The closed-form solution is:

$$s_{ij} = \begin{cases} \frac{1}{k} + \frac{1}{2\lambda_1 k}\sum_{l=1}^{k}(z_{il}-z_{ij}) & \text{if } j \in \mathcal{N}_i \\ 0 & \text{otherwise} \end{cases}$$

4) Fix $\mathbf{A}, \mathbf{B}, \mathbf{S}$, optimize α:
With $\mathbf{A}, \mathbf{B}, \mathbf{S}$ fixed, problem (8) becomes:

$$\min_{\alpha} \sum_{i=1}^{d} \|\mathbf{K}_i \alpha_i - \mathbf{XAB}\|_F^2 + \lambda_3 \|\alpha\|_{2,1}$$

Since this is convex but non-smooth, we use proximal gradient method. The update rule is:

$$\alpha^{t+1} = \arg\min_{\alpha} \frac{1}{2} \|\alpha - (\alpha^t - \eta_t \nabla f(\alpha^t))\|_2^2 + \lambda_3 \eta_t \|\alpha\|_{2,1}$$

where η_t is the step size and $\nabla f(\alpha^t)$ is the gradient. To accelerate convergence, we introduce auxiliary variables $\mathbf{U}^t$ and step size parameters β_t.

Algorithm 2: Optimization for Equation (22)
Input: $\mathbf{X} \in \mathbb{R}^{n \times d}$, λ_3, α^0
Output: Optimal α^*

1. Initialize $t = 0$, α^0 as random vector, $\mathbf{U}^0 = \alpha^0$, $\beta_0 = 1$
2. Do:
   a. Compute gradient $\nabla f(\mathbf{U}^t)$
   b. Update α^{t+1} via proximal operator
   c. Update $\beta_{t+1} = \frac{1+\sqrt{1+4\beta_t^2}}{2}$
   d. Update $\mathbf{U}^{t+1} = \alpha^{t+1} + \frac{\beta_t - 1}{\beta_{t+1}}(\alpha^{t+1} - \alpha^t)$
3. While not converged

Algorithm 3: Optimization for Equation (8)
Input: Training samples $\mathbf{X} \in \mathbb{R}^{n \times d}$, control parameters $\lambda_1, \lambda_2, \lambda_3$
Output: $\mathbf{A}, \mathbf{B}, \mathbf{S}, \alpha$

1. Randomly initialize matrices $\mathbf{A}^{(0)}, \mathbf{B}^{(0)}$, set $\mathbf{S}^{(0)}$ as zero matrix
2. Do:
   a. Update $\mathbf{A}^{(t+1)}$ with fixed $\mathbf{S}^{(t)}, \mathbf{B}^{(t)}, \alpha^{(t)}$
   b. Update $\mathbf{B}^{(t+1)}$ with fixed $\mathbf{S}^{(t)}, \mathbf{A}^{(t+1)}, \alpha^{(t)}$
   c. Update $\mathbf{S}^{(t+1)}$ with fixed $\mathbf{A}^{(t+1)}, \mathbf{B}^{(t+1)}, \alpha^{(t)}$
   d. Update $\alpha^{(t+1)}$ with fixed $\mathbf{A}^{(t+1)}, \mathbf{B}^{(t+1)}, \mathbf{S}^{(t+1)}$
3. While Equation (8) not converged

2.3 Algorithm Convergence Proof

From equation (19), for all $i = 1, ..., d$, $s_{ij}^{(t+1)}$ has a closed-form solution. We can derive:

$$F(\mathbf{A}^{(t+1)}, \mathbf{B}^{(t+1)}, \mathbf{S}^{(t+1)}, \alpha^{(t+1)}) \leq F(\mathbf{A}^{(t)}, \mathbf{B}^{(t)}, \mathbf{S}^{(t)}, \alpha^{(t)}) - \frac{\gamma}{2}\|\alpha^{(t+1)} - \alpha^{(t)}\|_2^2$$

Theorem 1: Let $\{\alpha^t\}$ be the sequence generated by Algorithm 2. Then for all $t \geq 1$, equation (30) holds, proving algorithm convergence.

According to reference [?], γ is a predefined constant, and L is the Lipschitz constant of the gradient in equation (23). From the above inequality and Theorem 1, we can easily see that the proposed algorithm converges.

3.1 Experimental Datasets and Comparison Algorithms

We evaluate LS_NFS on six datasets: Movements, Ecoli, Urban_land, Ionosphere, Colon, and Lung_discrete. The first three are from UCI [?], while the latter three are from feature selection datasets [?]. Dataset details are shown in Table 1 .

All experiments run on MATLAB 2014a under Windows 7. We compare LS_NFS against five algorithms:

- **LS** [?]: Uses Laplacian score to preserve local structure based on the assumption that nearby data points share similar relationships.

- **CSFS** [?]: A convex semi-supervised multi-label feature selection algorithm for large-scale multimedia analysis that considers attribute correlations and initializes unlabeled data with zero labels.

- **NetFS** [?]: A robust unsupervised feature selection method that embeds latent representation learning to mitigate noise effects.

- **RUFS** [?]: Another robust unsupervised feature selection algorithm that simultaneously performs clustering and feature selection while reducing time and space complexity.

- **RSR** [?]: Uses $\ell_{2,1}$-norm constrained self-representation coefficient matrices to select representative features with outlier robustness.

These algorithms have respective strengths but none simultaneously considers nonlinear relationships and similarities among attributes. Our LS_NFS algorithm combines local structure learning with kernel methods to more thoroughly mine attribute relationships for better subset selection.

3.2 Experimental Results and Analysis

We employ ten-fold cross-validation for dataset partitioning, construct new attribute sets using selected features, and classify using SVM. Classification accuracy evaluates performance. All algorithms run in identical environments, with results reported as mean ± standard deviation over 10 runs.

Classification accuracy is defined as:

$$\text{acc} = \frac{X_{\text{correct}}}{X_{\text{total}}}$$

where X_{total} is the total sample count and X_{correct} is correctly classified samples. Stability is measured by standard deviation:

$$\text{std} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\text{acc}_i - \mu)^2}$$

where N is the number of experiments, acc_i is the i-th accuracy, and μ is the mean accuracy. Lower std indicates better stability.

Figure 1 [Figure 1: see original paper] compares results across six datasets, with detailed statistics in Table 2 .

LS_NFS demonstrates strong performance across all datasets. For binary classification (Ionosphere and Colon), it shows excellent results. For multi-class datasets, LS_NFS also exhibits outstanding performance. While each dataset has unique characteristics and no algorithm performs best in every case, LS_NFS achieves the highest accuracy in most scenarios.

Table 2 reveals LS_NFS outperforms comparison algorithms: 12.14% improvement over LS, 5.79% over CSFS, 13.07% over NetFS, 9.95% over RUFS, and 13.08% over RSR on average. Regarding stability, LS_NFS' s average standard deviation is second only to LS (higher by just 0.07), indicating good robustness. LS_NFS' s effectiveness stems from considering both attribute similarities and nonlinear relationships. Despite varying data distributions and interference factors, LS_NFS consistently selects the most representative attributes with highest classification accuracy.

4 Conclusion

This paper proposes a novel unsupervised nonlinear feature selection algorithm that considers both similarity and nonlinear relationships among attributes. Local structure learning identifies attribute similarities, kernel methods capture nonlinear relationships, and sparse regularization performs feature selection. Low-rank constraints further enhance the model. Experimental results confirm significant improvements in classification accuracy and stability. Future work will explore integration with more advanced theoretical developments.

References

[1] Zhu Xiaofeng, Suk H I, Shen Dinggang. Matrix-similarity based loss function and feature selection for Alzheimer' s disease diagnosis [C]// Proc of IEEE Conference on Computer Vision and Pattern Recognition. 2014: 3089-3096.
[2] Zhu Xiaofeng, Huang Zi, Shen Hengtao, et al. Dimensionality reduction by mixed kernel canonical correlation analysis [J]. Pattern Recognition, 2012, 45(8): 3003-3016.
[3] Zhu Xiaofeng, Zhang Shichao, Jin Zhi, et al. Missing value estimation for mixed attribute dataSets [J]. IEEE Transactions on Knowledge and Data Engineering, 2011, 23(1): 110-121.
[4] Zhu Xiaofeng, Li Xuelong, Zhang Shichao. Block-row sparse multi-view multi-label learning for image classification. [J]. IEEE Trans on Cybernetics, 2016, 46(2): 450-461.
[5] Wei Xiaokai, Yu P S. unsupervised feature selection by preserving stochastic neighbors [C]// Proc of the 19th International Conference on Artificial Intelligence and Statistics 2016: 1.
[6] Zhu Xiaofeng, Huang Zi, Yang Yang, et al. Self-taught dimensionality reduction on the high-dimensional small sized data[J]. Pattern Recognition, 2013, 46(1): 215-229.
[7] Zhang Shichao, Jin Zhi, Zhu Xiaofeng. Missing data imputation by utilizing information within incomplete instances[J]. Journal of Systems and Software, 2011, 84(3): 452-459.
[8] Nie Feiping, Zhu Wei, Li Xuelong. Unsupervised feature selection with structured graph optimization [C]//Proc of the 30th AAAI Conference on Artificial Intelligence. Palo Alto:AAAI Press, 2016: 1302-1308.
[9] Hou Chenping, Nie Feiping, Li Xuelong, et al. Joint embedding learning and sparse regression: a framework for unsupervised feature selection [J]. IEEE Trans on Cybern, 2017, 44(6): 793-804.
[10] Shao Weixiang, He Lifang, Lu C T, et al. Online unsupervised multi-view feature selection [C]//Proc of the 16th IEEE International Conference on Data Mining.Piscatraway,NJ: IEEE Press 2016:
[11] Jian Ling, Li Jundong, Shu Kai, et al. Multi-label informed feature selection [C]//Proc of International Joint Conference on Artificial Intelligence. Palo Alto:AAAI Press, 2016: 1627-1633.
[12] Liu Xinwang, Wang Lei, Zhang Jian, et al. Global and local structure preservation for feature selection [J]. IEEE Trans on Neural Networks & Learning Systems, 2017, 25 (6): 1083-1095.
[13] Lu Canyi, Lin Zhouchen, Yan Shuicheng. Smoothed low rank and sparse

matrix recovery by iteratively reweighted least squares minimization [J].IEEE Trans on Image Processing, 2015,24(2):

[14] Wen Zaiwen, Yin Wotao. A feasible method for optimization with orthogonality constraints [J]. Mathematical Programming, 2013, 142 (1-2): 397-434.

[15] Nesterov Y. Introductory lectures on convex optimization [J]. Applied Optimization, 2004, 87(5): xviii, 236.

[16] UCI repository of machine learning datasets [EB/OL]. [2016-05-27]. http://archive.ics.uci.edu/ml/

[17] Feature selection datasets [EB/OL]. [2016-05-27]. http://featureselection.asu.edu/datasets.php

[18] He Xiaofei, Cai Deng, Niyogi P. Laplacian score for feature selection [C]//Proc of International Conference on Neural Information Processing Systems. Cambridge : MIT Press, 2005: 507-514.

[19] Chang Xiaojun, Nie Feiping, Yang Yi, et al. A convex formulation for semi-supervised multi-label feature selection [C]// Proc of the 28th AAAI Conference on Artificial Intelligence.Palo Alto: AAAI Press, 2014: 1171-1177.

[20] Li Jundong, Hu Xia, Wu Liang, et al. Robust unsupervised feature selection networked data [C]//Proc of SIAM International Conference on Data Mining. 2016: 387-395.

[21] Qian Mingjie, Zhai Chengxiang. Robust unsupervised feature selection [C]//Proc of International Joint Conference on Artificial Intelligence. 2013: 1621-1627.

[22] Zhu Pengfei, Zuo Wangmeng, Zhang Lei, et al. Unsupervised feature selection by regularized self-representation [J]. Pattern Recognition, 2015, 48(2): 438-446.

Note: Figure translations are in progress. See original paper for figures.