Journal of Network Intelligence ©2019 ISSN 2414-8105 (Online)
Taiwan Ubiquitous Information Volume **4**, Number **3**, August **2019**

High-dimensional Data Clustering Using K-means Subspace Feature Selection

Xiao-Dong Wang[1,2], Rung-Ching Chen[2,*], Fei Yan[3]

[1]College of Computer and Information Engineering,
Xiamen University of TechnologyXiamen, 361024, China
[2]Department of Information Management, Chaoyang University of Technology, Taichung, Taiwan
[3]College of Computer and Information Engineering,
Xiamen University of Technology, Xiamen, 361024, China
*Corresponding author: crching@cyut.edu.tw
xdwang@xmut.edu.cn, fyan@xmut.edu.cn

Received Feb 2019, Revision May 2019

Abstract. *K-means is an effective way to gain experience in a variety of applications. Although existing methods attempt to solve this problem by reducing the size, it is powerless to process high dimensional data that typically includes noise and redundant features. What is worse, an increase in complexity with a robust loss function performing a similarity measure in the original limits their ability. This paper aims at developing an adaptive algorithm which utilizes group sparse technique and feature learning via a joint framework to reduce the impact of outliers. Thanks to the framework with special feature selection matrix, the eigenvalue decomposition operation can be properly avoided, making it be easily applied to data with high dimensions. To verify the effectiveness of the proposed method, we have applied it to six datasets. The advantages over several state-of-the-art methods shows that it is suitable for real-world applications.*

Keywords: K-means clustering; High-dimensional data; Feature learning; Dimension reduction; Feature selection.

1. **Introduction.** Clustering is a conventional technique designed to distribute data with similar properties into the same group by exploring the underlying structure between the data. In the past few decades, clustering has been widely used in various practical applications, including image processing, computer vision and so on. Of all the recent published clustering algorithms, K-means clustering (KM) achieves the highest interests for its simplicity. However, due to the rapid development of information technology, a great quantity of high-dimensional data has been generated, which poses a considerable challenge to traditional knowledge management. For example, for a 128 x 128 image with a relatively small resolution, if we use it as an instance, it contains 16384 features. Due to the curse of dimensions and higher complexity, it is difficult to apply KM directly to these high-dimensional data. Another challenge for KM is that it is sensitive to outliers. Specifically, as shown in previous work [1], KM needs to iteratively update its centroid in an Equivalent l_2-orm in Euclidean space. This expectation maximization (EM) update method reduces the performance of clustering, especially when dealing with noisy data.

In order to cope with the first challenge of manipulating high-dimensional data, KM has found the best low-dimensional feature space [2]. One conventional solution for clustering high-dimensional data is to first project these data into low-dimensional data, such as principal component analysis (PCA) and local linear embedding (LLE), and then perform KM through a dimensionality reduction algorithm. For example, in [3], PCA is executed to learn the feature subspace of KM. However, due to the separation between subspace learning and clustering, the obtained subspace may not be optimal for subsequent clustering tasks. In order to solve this problem, Based on linear discriminant analysis (LDA), Ding et al. [2] proposed a coherent objective function, in which KM and LDA are executed simultaneously in such a way that KM generates ”cluster tags” for LDA, and in turn LDA finds the representative features for

KM. Nevertheless, when facing "small sample size" data, this algorithm is degraded, where the number of instances is less than the number of features. However, all of these subspace learning methods involve Eigenvalue Decomposition (ED), which is very time to consume for high dimensional data.

Clustering with feature selection is another solution for clustering high dimensional data. Feature selection can preserve the intrinsic structure of the original features compared to dimensionality reduction, and thus has recently gained more and more interest. For example, Chen et al. [4] designed a joint framework to combine feature learning and KM in an objective function, in which the weighted feature mechanism searches for the optimal feature subset. Similarly, Huang et al. [5] proposed to find the representative features of KM clustering by feature weighting method and extend it to multi-view clustering application. However, all of these subspace clustering algorithms are "soft subspace" methods that need to update their cluster centroids in the original high dimensional feature space. Therefore, they are inefficient and susceptible to noise and redundant features.

To address the second challenge of KM's sensitivity to outliers, the researchers propose to extend KM by imposing a new cluster centroid update mechanism or a new distance measurement between two data points [6]. For example, under the promotion of the group sparse regularization technique [7], several KM type methods have recently been developed [8]. Cai et al. [6] designed a KM based on the $l_{2,1}$-norm specification to reduce the impact of outliers. However, it is difficult to detect redundant features. Pan et al. proposed an α-Fraction first method to solve wireless sensor networks arrangement problem [9].

In order to solve the above problem, we have proposed a K-means clustering for high-dimensional data[10]. We discuss the Fast and Robust K-means Clustering with feature learning, namely FRKC. It utilizes feature learning and clustering jointly by a newly designed objective function. Furthermore, the feature selection matrix can be efficiently optimized without the assistance of ED. In addition, it imposes a loss function based on the $l_{2,1}$-norm, so it is robust to outliers. The remainders of the paper are organized as follows. Section 2 is related works about clustering and feature learning with regularization. The proposed method described in Section 3. Section 4 is Experimental arrangement and discussion and give conclusions and future works on Section 5.

2. **Related works.**

2.1. **Notations.**

Given n training instances $X = [x_1, x_2, ..., x_n] \in R^{(d \times n)}$ with d dimensionality. For concise illustration, we assume the data are centered. We use $||.||_F$ to denote the Frobenius norm and $F = [f_1, .., f_n]^T \in R^{(n \times g)}$ be the cluster indicator matrix with g clusters. $f_i \times R^{(g \times 1)}$ is the cluster indicator vector of sample i, if data x_i belongs to class $j, f_{ij} = 1$, otherwise $f_{ij} = 0$). For brevity, we use Inf to denote the cluster indicator matrix set.

2.2. **K-means clustering.**

The general process of KM can be divided into two steps. It first assigned several random centroids and grouped these data into k groups $G_1, G_2, ..., G_k$ according to their distance to centroids. After that, it recalculated the centroids of each cluster. These two steps interleave repeat until these centroids are no longer changed. Mathematically, KM has the following formulation:

$$\min_F \sum_{j=1}^{k} \sum_{i \in \mathbf{g}_j} (x_i - c_j)^T (x_i - c_j) \tag{1}$$

where c_j is the $j - th$ cluster centroid.

From Eq.(1), it can be found that KM judges the closeness among centroids and data samples using squared l_2-norm, which will generate larger bias when the outliers with great distances to the centroids appear [6] [8].

2.3. **Feature learning for clustering with regularization.**

Feature selection (FS) is a well-known technology to locate the most representative features and has shown its efficiency in many applications. Recently, several FS methods have been published, among which the regularized FS achieves comprehensive performance and updates the researchers' interest. For example, an unsupervised FS was developed in [11]. It can explore clear structures by constructing graphs under l_1-norm constraint and find the importance of features by $l_2, 1$-norm group sparsity.

To improve the performance of knowledge management and retain the original feature structure, some researchers try to integrate feature learning into knowledge management. For example, Wang et al. [12] imposed an $l_2, 1$-norm standard regularization and discriminative technique on traditional knowledge management and showed good performance in improving clustering. However, this algorithm relies on ED, which is very complex for practical applications. The "soft subspace" algorithm can solve this problem. For example, De Amorim et al. [13] proposed a Minkowski weighted K-means clustering in which feature weights are converted into feature rescaling factors. However, all of these algorithms need to iteratively update their cluster centroids in the original high dimensional data space, which is very time consuming and difficult to apply to large datasets.

3. **The proposed method.**

3.1. **Robust KM with Feature Selection.**

The objective function of KM in Eq.(1) can be reformulated as:

$$\min_{G,\ F\in Inf} \sum_{i=1}^{g} \sum_{x_i\in\mathcal{C}_j} x_i - g_j{}_2^2 \tag{2}$$

where $C = [c_1, c_2, \cdots, c_g] \in \mathbb{R}^{\mathrm{d}\times\mathrm{g}}$ is the cluster centroid matrix. Eq.(2) can be rewritten as:

$$\min_{C,F\in Inf} \sum_{i=1}^{n} ||\, x_i - Cf_i^T \,||_2^2 \tag{3}$$

Nevertheless, as discussed in previous studies [14], the l_2-norm term in Eq.(3) is sensitive to noise, which frequently appears in many real-world applications. To address this problem, a non-squared l_2-norm based loss function is imposed as:

$$\min_{C,F\in Inf} \sum_{i=1}^{n} \left\|x_i - Cf_i^T\right\|_2 \tag{4}$$

To reduce the influence of redundant features, following[4], we introduce a special selection matrix into Eq.(4) and have:

$$\min_{c,F\in Inf} \sum_{i=1}^{n} \left\|W^T x_i - Cf_i^T\right\|_2 \tag{5}$$

where $W = [w_1, w_2, \cdots,\ w_m] \in \mathbb{R}^{d\times m}$ $(m << d)$ is the selection matrix, whose i-th column vector w_i is defined as:

$$w_i = \left[\overbrace{0,...,0}^{i-1}, 1, \overbrace{0,..,0}^{D-i}\right]^T \tag{6}$$

It can be easily observed that W in Eq.(6) is a column-full-rank transformation matrix and is extremely sparse, making the subsequent optimal procedure for feature subset efficient and independent of ED. We will show this optimization strategy in Section 3.2.

3.2. **Optimization.**

Our proposed objective function in Eq.(5) involves a non-square l_2-norm term, which is non-smooth and hard to optimize directly. In this section, we recommend alternately optimizing it. More specifically, we optimize one of these variables by constantly fixing other variables. Before that, we first rewrite equation (5) as follows.

$$\min_{c,F\in Inf} \sum_{i=1}^{n} \left\|d_i W^T x_i - Cf_i^T\right\|_2^2 \tag{7}$$

where $d_i = \frac{1}{2\sqrt{\left\|x_i - cf_i^T\right\|_2 + \sigma}}$ and σ is a small enough positive constant. D is a diagonal matrix with its i-th diagonal element equals to d_i. Then Eq.(7) can be reformulated as:

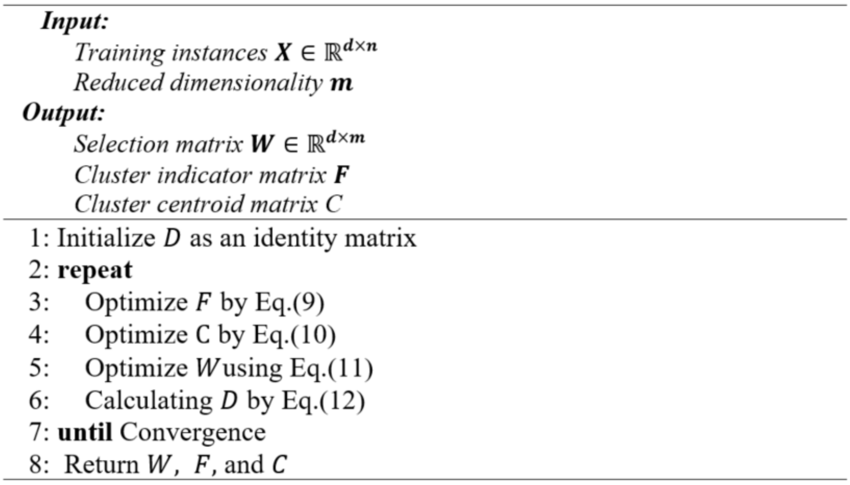
$$\min_{C,D,F\in Inf} Tr\big(X^T W - FC^T\big)^T D \left(X^T W - FC^T\right) \tag{8}$$

a.Fixing W,D,C, and optimizing F. When fixing W and C, D, Eq.(8) becomes:

$$f_{ij} = \begin{cases} 1,\ j = arg\ \min\limits_{k}\ \left\|W^T x_i - c_k\right\|_2^2 \\ \qquad 0,\ Otherwise. \end{cases} \tag{9}$$

Obviously, Eq.(9) is the traditional K-means in low-dimensional feature space.

Table 1. The procedure of the proposed method



```
Input:
    Training instances $X \in \mathbb{R}^{d\times n}$
    Reduced dimensionality $m$
Output:
    Selection matrix $W \in \mathbb{R}^{d\times m}$
    Cluster indicator matrix $F$
    Cluster centroid matrix $C$
Initialize $D$ as an identity matrix
repeat
    Optimize $F$ by Eq.(9)
    Optimize C by Eq.(10)
    Optimize $W$ using Eq.(11)
    Calculating $D$ by Eq.(12)
until Convergence
Return $W$, $F$, and $C$
```

b.Fixing D,F, and optimizing C,W. Taking the derivative of Eq.(8) w.r.t. C to zero, we have:

$$C = W^T XDF\left(F^T DF\right)^{-1} \tag{10}$$

Substituting Eq.(9) into the Eq. (7), we have:

$$\min_{W} Tr\left(W^T MW\right) = \min_{W} \sum_{i=1}^{m} Tr\left(w_i^T Mw_i\right) \tag{11}$$

where $M = XNX^T$ and $= D - DF(F^T DF)^{-1}F^T D$

Considering the feature selection matrix in Eq.(6), Eq.(11) can be effectively optimized by locating the first m smallest elements of $u \in \mathbb{R}^d$, where $u_i = \left\| \left(XN^{\frac{1}{2}}\right)_{i:} \right\|_2^2$

c.Fixing F, G, W and updating D by:

$$D = \begin{bmatrix} \frac{1}{2\left\|W^T x_i - Cf_i^T\right\|_2} & & \\ & \ddots & \\ & & \frac{1}{2\|W^T x_n - Cf_n^T\|_2} \end{bmatrix} \tag{12}$$

Table 1 shows the procedure of the proposed method.

3.3. Complexity analysis.

To show the effectiveness of the proposed method, we briefly discuss its complexity. From Table 1, to find the optimal F, we need *O(mcn)* multiplications. Then, to calculate C need $O(mcn + c^2n)$ complexity. To optimize W, we have to locate the first m smallest elements of M in Eq.(11), whose computational complexity is $O(dn + mlogm)$. As $m << d$ and $g << n$, thereby, the complexity of FRKC is $O(dn)$. Thus, FRKC can be applied to high-dimensional data.

4. Experiments and Discussion.

In our experiments, we first demonstrated the effectiveness of FRKC on the toy sample dataset, and six selected datasets (summarized in Table 2), including three UCI datasets (Cars, Wine, and Vote), two image datasets (MSRA25 and Yale), and one text dataset (WebKB). We then compared FRKC with several closely related KM type algorithms as follows.

- **KM**: the widely used K-means clustering algorithm.
- **RKM**: a single-view version of the robust multi-view KM in [6].

4.1. Experiment Setup.

The compared KM type clustering algorithms are sensitive to the initial cluster centroids. For fair comparison, we independently repeat clustering procedure 50 times and report the average results for each dataset. For the KM type subspace clustering, i.e. FRKC, we record the average clustering with optimal feature subset. Accuracy (ACC) is used as the evaluation metric. Mathematically, given an arbitrary data point x_i, denote p_i as the predicted cluster labels vector and q_i as the ground truth label vector. ACC is defined as follows:

Table 2. Details of selected datasets

Datasets	Classes	# of instances	Dimensions	Reduced dimensions
Cars	3	392	8	{2,3,4,5,6}
Wine	3	178	13	{2,3,4,5,6,7,8,9}
Vote	2	435	16	{2,3,4,5,6,7,8,9}
MSRA25	12	1799	256	{20,30,40,50,60,70,80}
Yale	15	165	1024	{20,30,40,50,60,70,80}
WebKB	7	814	4029	{20,30,40,50,60,70,80}

Table 3. Clustering performance comparison of different algorithms

Datasets	Cars	Wine	Vote	MSRA25	Yale	WebKB
KM	44.90 ± 0.00	65.17 ± 6.53	82.32 ± 0.54	49.47 ± 4.79	40.58 ± 3.55	56.27 ± 2.41
RKM	49.02 ± 0.09	71.78 ± 0.29	83.45 ± 0.00	46.21 ± 5.30	42.02 ± 4.59	45.20 ± 4.89
FRKC	$\mathbf{59.08 \pm 0.08}$	$\mathbf{87.94 \pm 1.83}$	$\mathbf{87.39 \pm 0.58}$	$\mathbf{55.60 \pm 3.18}$	$\mathbf{42.92 \pm 3.18}$	$\mathbf{66.86 \pm 0.86}$

$$ACC = \frac{\sum_{i=1}^{n} \delta\left(p_i, map\left(q_i\right)\right)}{n} \tag{13}$$

where n is number of instances, $\sigma(x, y) = 1$ if $x = y$; $\sigma(x, y) = 0$ otherwise, and $map(q_i)$ is the mapping function based on Kuhn-Munkres algorithm. A larger ACC indicates a better clustering performance.

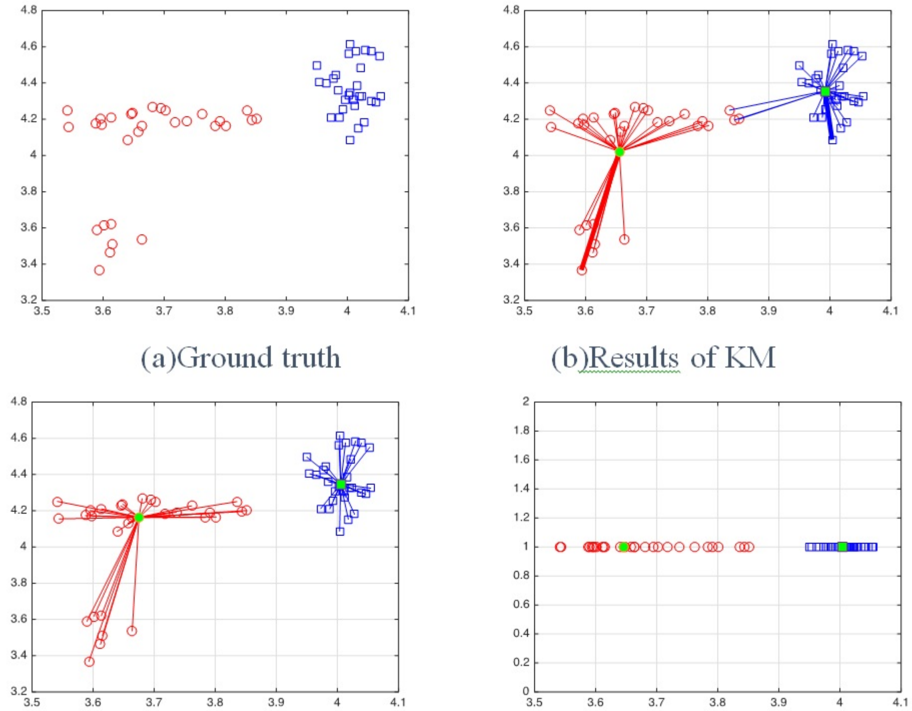
4.2. **Toy Example.** In this section, we will present an example of a two-dimensional synthetic dataset (it contains the x-direction and the y-direction) to illustrate how FRKC works. This dataset can be categorized into two parts. Every part contains 30 samples. More concretely, data points of the right part (blue square) are generated from a two-dimensional normal distribution with the mean [4,4.4] and covariance matrix [0.001,0;0,0.02]. Data points of the left part (red circle) contains two components. The first component is 25 samples generated by a two-dimensional normal distribution with the mean [3.7,4.2] and covariance matrix [0.01,0;0,0.02]. The second component is 7 samples, which can be taken as the outliers for the left part shown in Fig 1(a) (red circle). The samples in this component are generated randomly by a normal distribution with the mean [3.6,3.6] and covariance matrix [0.001,0;0,0.01].

To show the effectiveness of FRKC, we first perform FRKC and KM on the 2-D dimensional dataset without feature selection, that is we set $m = d$ in FRKC. Then we perform FRKC in 1-D dimensional data space to demonstrate its embedding clustering performance. The clustering results of KM on 2-D dimensional data are shown in Figure 1(b). We can observe that KM has misclassified 3 data points (3 red circles connected by blue lines) by the effect of outliers. The reason may be that KM always assigns a higher weight value for the data far away from its cluster centroid. For example, in Figure 1(b) the data points with a large distance to its cluster centroid will have a large weight (thick red line or thick blue line). Such an assignment will cause a vast bias when updating the cluster centroids.

Compared with KM, FRKC is able to obtain more stable results in Fig.1(c) by evaluating the importance of data points by their contributions to the cluster (that is the fitness to their clusters). Besides, FRKC can precisely find the optimal feature subset, which can be seen from the results show in Fig.1(d). We can observe that FRKC can correctly separate these two groups by preserving the x-direction features.

From the results shown in Fig.1, we can conclude that: 1). With the help of clustering, FRKC improves the performance of clustering once finding the suitable feature subset; (2) With the help of feature selection, the clustering performance, including accuracy and efficiency, can also be increased obviously.

4.3. **Clustering Performance Comparison.** Table 2 gives the clustering comparison results. We can conclude that (1) mostly, RKM performs better than KM by integrating the $l_{2,1} - norm$ regularizaiton, particularly, when dealing with data encoded by different types of features, e.g. Cars and Wine. (2) RKM designs a re-weighted iterative approach to optimize the $l_{2,1} - norm$ loss function. However, it is sensitive to small loss. Therefore, when processing data with plenty of redundant features, i.e. MSRA25, Yale, and WebKB, RKM fails. 3) FRKC gets the best clustering results over the other compared algorithms. The primary reason may be that it combines feature selection and robust learning simultaneously. This



(a)Ground truth

(b)Results of KM

Figure 1. Clustering results of KM and FRKC on toy example dataset

also demonstrates that it is beneficial to select the most discriminative feature subset for $l_{2,1}-norm$ loss function.

4.4. **Computational Time Comparison.** Since computational efficiency is a very important quality metric for clustering, we conduct a corresponding experiment to verify FRKC.

We selected all the 6 public datasets for comparison. For impartiality, KM and FRKC, are implemented in their original formulation, without any other accelerating operation. For RKM, we download the source code from the author's websites and follow their experimental setting. All algorithms were realized by Matlab 2015b and executed on Intel® Xeon® CPU E5-2620 2.00GHz with 48G memory and Windows Server 2008 operating system. Following [11], for the subspace clustering algorithm, i.e. FRKC, we fixed the reduced dimensionality as g-1. The computational time of different algorithms is listed in Table 3, where the subscript "ms" means milliseconds and "s" means seconds. We can make the following observations:

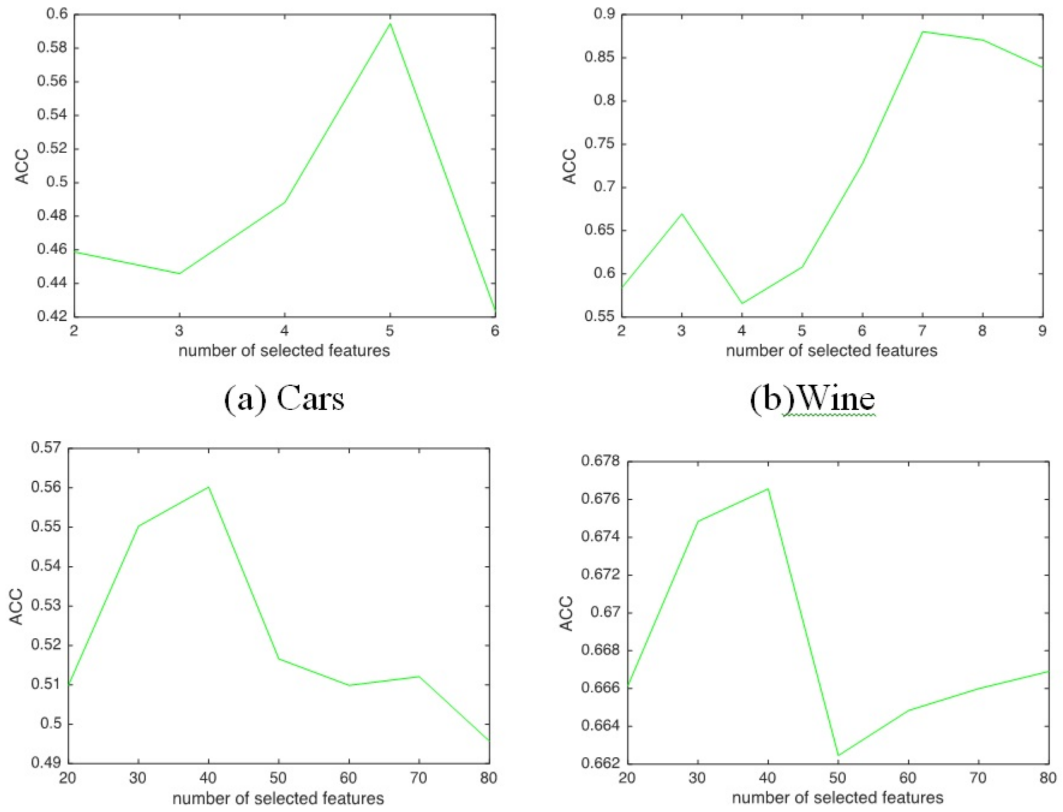
(1) Of the different methods on the different datasets, KM and FRKC consume the least time for most of the datasets. RKM takes the most computational times compared with the others except WebKB. The reason for this may be that RKM iteratively updates the cluster centroids and feature weights in the high-dimensional feature space. Besides, the implementation of RKM involves too many loop operations, which are hard to be optimized by MATLAB.

(2) When dealing with datasets with few features, i.e. Cars, Wine, and Vote, KM needs few computational time compared with FRKC. However, FRKC is more capable of handling high-dimensional data, i.e. MSRA25, Yale, WebKB. This may be caused by the fact that FRKC iteratively updates the cluster centroids in a low-dimensional feature space. For example, FRKC is nearly 84 times faster than KM on WebKB dataset. Therefore, we can conclude that FRKC is suitable for high-dimensional data.

4.5. **Influence of Selected Features.** As feature selection is a key component in subspace clustering, we perform an experiment to study how the number of selected features can affect the clustering performance. This experiment can help us to better understand the general tradeoff between performance and computational cost for real-world applications. From these results in Figure 2, we can observe when supplied with more features, not all methods achieve higher performance. Therefore, we can conclude that feature selection is beneficial for clustering. What is more, the optimal feature subset varies with different kinds of the datasets, e.g., 40 for WebKB and MSRA25, 7 for Wine. The underlying reason lies in that different datasets have different intrinsic dimensionality. Thus, we conclude that FRKC can both

Table 4. Computational time (in A microsecond)of different methods on various datasets (mean ± STD)

Datasets	KM	RKM	FRKC
Cars$_{(ms)}$	11.40 ± 3.60	344.48 ± 3.94	14.83 ± 4.41
Wine$_{(ms)}$	9.21 ± 5.56	145.99 ± 2.06	11.57 ± 3.78
Vote$_{(ms)}$	8.24 ± 2.13	161.27 ± 34.00	13.33 ± 6.55
MSRA25$_{(s)}$	0.60 ± 0.37	6.10 ± 0.38	0.33 ± 0.05
Yale$_{(s)}$	0.24 ± 0.03	0.83 ± 0.13	0.08 ± 0.00
WebKB$_{(s)}$	29.32 ± 12.60	7.99 ± 1.02	0.35 ± 0.12



(a) Cars

(b)Wine

Figure 2. Clustering results on 4 datasets with different number of selected features

improve the clustering performance and computational cost (by performing on low-dimensional feature space), if the optimal feature subset is determined. However, how to select the optimal number of reduced features is still an open problem for FRKC, which will be investigated in the future.

5. **Conclusion.** To cope with the difficulty of clustering high-dimensional data, we have designed a fast and robust clustering framework. In this framework, we have introduced a special selection matrix to locate the representative features by extracting the discriminative power among clusters. To achieve a robustness clustering, we also utilized the $l_{2,1}$-norm group sparsity technique to constrain the loss function. As the proposed objective function is non-smooth and is hard to be optimized directly, we have constructed a efficient optimizing algorithm to solve it. Experimental results on six public datasets demonstrated that our algorithm is simple and less energy-consuming, and can be applied to high-dimensional data.

Acknowledgment. This paper was supported by l Ministry of Science and Technology, Taiwan, (Grant Nos. MOST-107- 2221-E-324-018-MY2).

REFERENCES

[1] J. Xu, J. Han, and F. Nie, *Discriminatively Embedded K-Means for Multi-view Clustering*, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5356–5364, 2016.

[2] C. Ding and T. Li, *Adaptive dimension reduction using discriminant analysis and K -means clustering*, conf. Machine learning, ICML, pp. 521–528, 2007 .

[3] C. Hou, F. Nie, Y. Jiao, C. Zhang, and Y. Wu, *Learning a subspace for clustering via pattern shrinking*, Information Processing and Management, Vol. 49, No. 4, pp. 871–883, 2013.

[4] X. Chen, Y. Ye, X. Xu, and J. Z. Huang, *A feature group weighting method for subspace clustering of high-dimensional data*, Pattern Recognition, Vol. 45, No. 1, pp. 434–446, 2012.

[5] X. Huang, Y. Ye, and H. Zhang, *Extensions of Kmeans-Type Algorithms: A New Clustering Framework by Integrating Intracluster Compactness and Intercluster Separation*, IEEE Transactions on Neural Networks and Learning Systems, Vol. 25, No. 8, pp. 1433–1446, 2014.

[6] X. Cai, F. Nie, and H. Huang, *Multi-View K -Means Clustering on Big Data*, The 23rd International Joint Conference on Artificial Intelligence, pp. 2598–2604, 2013.

[7] X. Wang, R.-C. Chen, F. Yan, and Z. Zeng, *Semi-supervised feature selection with exploiting shared information among multiple tasks*, Journal of Visual Communication and Image Representation, Vol. 41, pp. 272–280, 2016.

[8] L. Du et al., *Robust multiple kernel K-means using l2,1-Norm*, IJCAI International Joint Conference on Artificial Intelligence, pp. 3476–3482.

[9] Jeng-Shyang Pan, Lingping Kong, Tien-Wen Sung, Pei-Wei Tsai and Vaclav Snasel, *alpha-Fraction First Strategy for Hirarchical Wireless Sensor Neteorks*, Journal of Internet Technology, Vol. 19, No. 6, pp. 1717 1726, 2015.

[10] X.-D. Wang, R.-C. Chen, and F. Yan, *Fast and robust K-means clustering via feature learning on high-dimensional data*, IEEE 8th International Conference on Awareness Science and Technology (iCAST), pp. 194–198, 2017.

[11] X. Wang, R.-C. Chen, C. Hong, Z. Zeng, and Z. Zhou, *Semi-supervised multi-label feature selection via label correlation analysis with l 1 -norm graph embedding*, Image and Vision Computing, Vol. 63, pp. 10–23, 2017.

[12] D. Wang, F. Nie, and H. Huang, *Unsupervised Feature Selection via Unified Trace Ratio Formulation and K-means Clustering (TRACK)*, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), pp. 306–321, 2014.

[13] R. Cordeiro de Amorim and B. Mirkin, *Minkowski metric, feature weighting and anomalous cluster initializing in K-Means clustering*, Pattern Recognition, Vol. 45, No. 3, pp. 1061–1075, 2012.

[14] F. Nie, H. Huang, X. Cai, and C. Ding, *Efficient and Robust Feature Selection via Joint l2,1-Norms Minimization*, Advances in Neural Information Processing Systems, Vol. 23, pp. 1813–1821, 2010.