Finding Well-Clusterable Subspaces for High Dimensional Data

A Numerical One-Dimension Approach

Chuanren Liu[1], Tianming Hu[2,⋆], Yong Ge[3], and Hui Xiong[1]

[1] Rutgers University, New Jersey, USA
{chuanren.liu,hxiong}@rutgers.edu
[2] Dongguan University of Technology, Guangdong, China
tmhu@ieee.org
[3] UNC Charlotte, North Carolina, USA
yong.ge@uncc.edu

Abstract. High dimensionality poses two challenges for clustering algorithms: features may be noisy and data may be sparse. To address these challenges, subspace clustering seeks to project the data onto simple yet informative subspaces. The projection process should be fast and the projected subspaces should be well-clusterable. In this paper, we describe a numerical one-dimensional subspace approach for high dimensional data. First, we show that the numerical one-dimensional subspaces can be constructed efficiently by controlling the correlation structure. Next, we propose two strategies to aggregate the representatives from each numerical one-dimensional subspace into the final projected space, where the clustering problem becomes tractable. Finally, the experiments on real-world document data sets demonstrate that, compared to competing methods, our approach can find more clusterable subspaces which align better with the true class labels.

Keywords: numerical one-dimension, clusterable subspace, subspace learning.

1 Introduction

People often face a dilemma when analyzing high dimensional data. On one hand, more features imply more information available for the learning task. On the other hand, irrelevant/contradicting features introduce noise and may mislead the learning algorithms. This difficulty has been studied extensively in the literature from different perspectives including dimension reduction, feature selection, model ensembling, etc.

Among them, multiple subspace learning is a promising paradigm to address the high dimensional difficulty. In this approach, we construct multiple simple

⋆ Corresponding Author. This research was partially supported by National Science Foundation via grant CCF-1018151 and IIS-1256016. Also, it was supported in part by Natural Science Foundation of China (No. 61100136 and 71329201).

V.S. Tseng et al. (Eds.): PAKDD 2014, Part II, LNAI 8444, pp. 311–323, 2014.

yet informative subspaces of the original high dimensional data. For example, principle component analysis (PCA) chooses the subspaces that best preserve the variance of the data. Then we can either build learning models in the aggregated space, or build models collaboratively in each of the subspaces. This paradigm brings several desirable advantages. First, we can construct the subspaces by grouping related features together and separating contradicting features simultaneously. This is superior to simple feature reduction which may lose information carried by contradicting features. Second, such collaborative learning mode in the aggregated space is superior to separately learning one submodel at a time and finally combining them. In fact, this mode share some spirit with multi-source learning [3] in the literature. In the language of multi-source learning, directly learning the original high dimensional data is actually the early-source-combination based approach, which might be too difficult for a single model. At the other extreme, directly assembling the separately learned submodels is actually the late-source-combination based approach, which might make very limited or even no information to be shared among different submodels. The aggregated/collaborative learning mode is actually the intermediate-source-combination approach, which can balance between the learning difficulty of too many features for individual models and the ensemble difficulty of many too isolated and non-cooperative models.

Along this line, in this paper, we focus on the task of subspace learning for clustering high dimensional data. Specifically, we first construct numerical one-dimensional subspaces consisting of highly related features. In theory, such subspaces can substantially alleviate the unstable difficulties often encountered by clustering algorithms such as K-means. In practice, we show such subspaces can be efficiently constructed by leveraging correlation coefficients. Next, by further exploiting the one-dimension nature, we propose strategies to aggregate the representatives from the numerical one-dimensional subspaces into the final projected space. Finally, we use real-world document data sets to compare our approach with several competing methods in terms of performance lift and clustering separability. The experimental results demonstrate that our approach can find more clusterable subspaces which align better with the true class labels.

The rest of the paper is organized as follows. Section 2 summarizes recent works related to subspace learning. In Section 3, we show the numerical one-dimensional subspaces can be constructed by controlling the correlation structure. In Section 4, we propose strategies to build the final projected subspace by aggregating the representatives from the numerical one-dimensional subspaces. Section 5 validates the effectiveness of our idea on real-world document data sets. Section 6 concludes this paper with some remarks on future work.

2 Related Work

Our work can be categorized as dimension reduction for clustering. Although there have been extensive studies of dimension reduction techniques in the literature, few of them are designed specially for the general clustering problems.

In [10], the idea of grouping correlated features was exploited for the regression of the DNA microarray data. Specifically, the authors defined the "supergenes" by averaging the genes within the correlated feature subspaces and then used them to fit the regression models. In our case of unsupervised clustering, however, we do not have response for learning, which was used in [10] to analyze the accuracy improvement of the regression with the averaged features. Instead, we show that the subspaces of correlated features are actually of numerical one-dimension, which speaks to the improved clustering stability. Furthermore, empirical studies on real-world data sets suggest that they enjoy higher clustering separability which aligns better with the true class labels. In [1], another approach of dimension reduction, random projection, was exploited for the clustering problems. It is shown that any set of N points in D dimensions can be projected into $O(K/\epsilon^2)$ dimensions, for $\epsilon \in (0, 1/3)$, where optimal K-means can be preserved. In the later experiments, we will compare our methods with this baseline approach.

Another category of related work includes the validation measures of the clustering results. [17] gave an organized study of the external validation measures. Normalization solutions and major properties of several measures were provided. Later, [9] investigated more widely used internal clustering validation measures. Recently, [5] studied the effectiveness of the validation measures with respect to different distance metrics. It is shown that the validation measures might biasedly prefer some distance metrics. Thus, we should be careful with the choice of validation measures involving distance computation.

3 Numerical 1-Dimensional Subspace Construction

In this section, we first use the simpleness of 1-dimensional clustering to introduce the motivation of our work. Then we show how to construct numerical 1-dimensional subspaces by controlling the correlation structure of the features.

3.1 1-Dimensional Clustering

The clustering problem can be formulated as:

Problem 1. Given a set of observations $\mathbf{X}$, and the number of clusters K, the optimal clustering solution $C = \{C_1, \cdots, C_K\}$ minimizes the so-called within-cluster sum of squares (WCSS):

$$\text{WCSS}(\mathbf{X}|C) = \sum_{k=1}^{K} \sum_{x \in \mathbf{X} \cap C_k} \|x - \mu_k\|^2$$

where μ_k is the centroid of cluster C_k.

The most common solver for this problem, K-means [18], can only achieve local optima, which are not stable. Indeed, we might have more than one solutions, which are often inconsistent with one another. However, there is a special place where K-means yields more stable clustering results: 1-dimensional space.

Proposition 1. *For any two K-means clustering solutions on a 1-dimensional data set, $C^1 = \{C_1^1, C_2^1, \cdots\}$ and $C^2 = \{C_1^2, C_2^2, \cdots\}$, with cluster centers $c_i^j \in C_i^j$ where $c_1^j < c_2^j < \cdots$ for $j = 1, 2$, there are no data points x_1 and x_2 such that $x_1 \in C_1^1, x_2 \in C_2^1$ but $x_1 \in C_2^2, x_2 \in C_1^2$.*

The proof is straightforward and is omitted due to space limit. In other words, K-means clustering is very simple in 1-dimensional space, which is equivalent to finding the cut points. This can also be intuitively visualized in the *clustergram* [12], as we will see later in Figure 1. In short, the *clustergram* examines how data points in each cluster are assigned to new clusters in the next round as the number of clusters increase. When Proposition 1 holds, it is expected that there are few cross lines connecting the consecutive solutions. However, few data are so perfectly "1-dimensional" in reality. Hence, in the following, we seek 1-dimension-like subspaces, where Proposition 1 can be preserved approximately.

3.2 Numerical 1-Dimensional Subspace

In 1-dimension-like subspaces (subset of features), it is observed that, if most of the variation of the data can be captured by the first principle component, then K-means is roughly equivalent to clustering in 1-dimensional space (along the first principle direction). In this case, Proposition 1 will still hold under the mild assumption that all cluster centers can be roughly connected by a line parallel to the first principle direction. Specifically, note that, if data point x is closer to cluster center c, its projection $\langle x, v\rangle$ is also closer to c on the axis of the first principle direction v. Formally, this notion is captured by the numerical 1-dimensional space define below [11, 7]:

Definition 1. *A data set $\mathbf{X}$ is numerical 1-dimensional with error ϵ, if and only if $\sigma^2 \leq \epsilon\sigma^1$, where $\sigma_1 \geq \sigma_2 \geq \cdots$ are singular values of $\mathbf{X}$ (standardized to be of zero-mean and unit-variance along each feature).*

At first glance, we need to perform singular value decomposition many times to find such subspaces, which is expensive in high dimensional space. Nevertheless, as we will show below, the error ϵ is bounded with a term of correlation among features, which can be leveraged to construct the desired subspaces efficiently.

Theorem 1. *If the average correlation of different features in the d-dimensional data set $\mathbf{X}$ is $\rho > 0$, then $\mathbf{X}$ is numerical 1-dimensional with error*

$$\epsilon \leq \sqrt{\frac{(1-\rho)d - 1 + \rho}{\rho d + 1 - \rho}} < \sqrt{\frac{1-\rho}{\rho}}.$$

Proof. Suppose matrix $\mathbf{X} \in \mathbb{R}^{N \times d}$ is already standardized to be of zero-mean and unit-variance along each feature (column). Then the feature correlations of $\mathbf{X}$ can be expressed by $\mathbf{C} = \frac{1}{N}\mathbf{X}'\mathbf{X}$ where the diagonal coefficients are all 1. With the singular value decomposition (SVD) $\mathbf{X} = U\Sigma V'$ where U, V are unitary matrices

and the diagonal coefficients of Σ are $\sigma_1, \sigma_2, \cdots$, we have $\mathbf{C} = \frac{1}{N} V \Sigma' \Sigma V'$ where $\Sigma'\Sigma = \text{diag}(\sigma_1^2, \sigma_2^2, \cdots)$. It follows that

$$\frac{1}{N}(\sigma_1^2 + \sigma_2^2) \leq \text{tr}(\mathbf{C}) = d.$$

Let J be the column vector with 1 as all coefficients, then on one hand we have

$$\begin{aligned} NJ'\mathbf{C}J &= (V'J)'\Sigma'\Sigma(V'J) = \sum_i (\sum_j v_{ji})^2 \sigma_i^2 \\ &\leq \sigma_1^2 \sum_i (\sum_j v_{ji})^2 = \sigma_1^2 (V'J)'(V'J) = \sigma_1^2 J'J = d\sigma_1^2. \end{aligned}$$

On the other hand, with the average of non-diagonal coefficients in $\mathbf{C}$, ρ, we have $J'\mathbf{C}J = \sum_{i,j} c_{ij} \geq (d^2 - d)\rho + d$. Hence, it follows that

$$\begin{aligned} \frac{1}{N}\sigma_1^2 &\geq \rho d + 1 - \rho \\ \frac{1}{N}\sigma_2^2 &\leq (1-\rho)d - 1 + \rho \end{aligned}$$

and this concludes our proof.

Theorem 1 suggests that, with a proper threshold of average correlation, the agglomerative hierarchical clustering over the feature set with average linkage can unambiguously group the original space into numerical 1-dimensional subspaces with error lower than the desired level. The standard Euclidean distance between features can be used as the linkage when the data matrix is of zero-mean and unit-variance along each feature. In the general case, the computational complexity of the agglomerative average linkage algorithm for D-dimensional data is $O(D^3)$, which is not efficient for big data applications. However, we note that Theorem 1 still holds if we denote ρ as the minimal correlation between features. This leads to the complete linkage clustering for which the computational complexity can be reduced to roughly $O(D^2)$. We will use this procedure in our experiments and denote it by $\mathcal{F} = N1dSpaces(\mathbf{X}, \epsilon)$ in the following discussions, where $\mathbf{X}$ is the data matrix, ϵ is the maximal error of numerical 1-dimensional subspaces, and $\mathcal{F}$ is the constructed subspaces.

The effectiveness of the subspace construction algorithm can be visualized in Figure 1, as mentioned earlier. Specifically, for a given high dimensional data set $\mathbf{X}$, we can produce a *clusgtergram* by directly applying a clustering algorithm, such as K-means with increasing number of clusters. Then we can construct the numerical 1-dimensional subspaces $\mathcal{F}$, and produce the same *clusgergram* in each subspace $\mathbf{S}$ in $\mathcal{F}$. The results show that, in the subspaces, there are few cross lines connecting the consecutive solutions.

4 Collaborative Ensemble of Subspaces

Now we have constructed subspaces where the clustering problem can be approached stably. However, clustering algorithms directly applied to the isolated

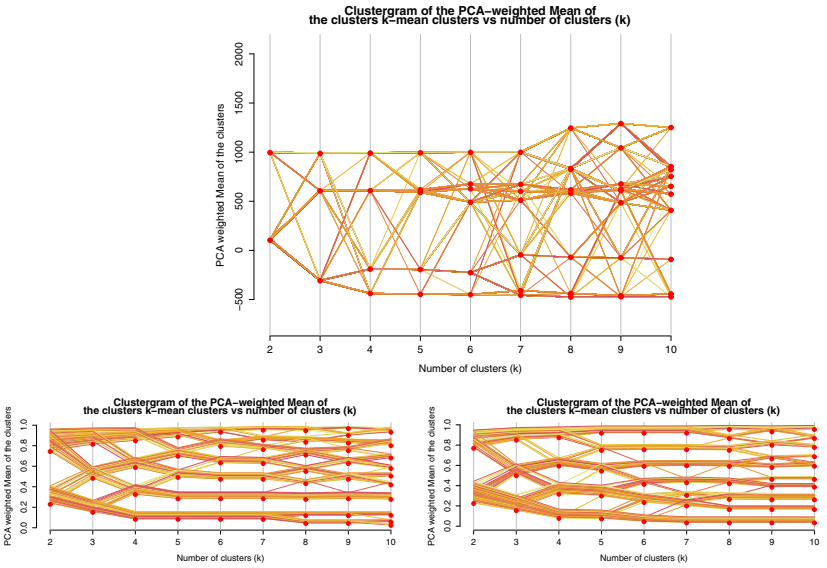

Fig. 1. Comparison of *clustergram*, where cluster means of consecutive cluster solutions are connected with parallelograms whose widths are proportional to the size of data assigned from the previous clusters. The top figure shows the *clustergram* of the high dimensional space. The bottom figures show the *clustergram* of two numerical 1-dimensional subspaces.

subspaces might produce degenerated solutions, since no information is shared between the subspaces. On the other hand, since each subspace $\mathbf{S}$ is numerically only of 1 dimension, it can be approximated by a few observation features. A natural way to this end is to investigate the SVD $\mathbf{S} = \mathbf{U}\Sigma\mathbf{V}'$, where $\Sigma = \text{diag}(\sigma_1, \cdots, \sigma_s)$ is a diagonal matrix consisting of s positive singular values of $\mathbf{S}$: $\sigma_1 \geq \cdots \geq \sigma_s$. In general, we can transform $\mathbf{S}$ to $\mathbf{SV}$ by the principal directions in $\mathbf{V}$. Then, guaranteed by Theorem 1, we can use only the first principal component $\mathbf{Sv}$ where $\mathbf{v}$ is the first principal direction in $\mathbf{V}$ corresponding to σ_1. Note that, this is often computationally more efficient, since we only need the first singular vector and it is not necessary to fully decompose $\mathbf{S}$. Also, when the number of features are small in $\mathbf{S}$, the computation can be further boosted by decomposing $\mathbf{S}'\mathbf{S}$ as in Theorem 1. This collaborative strategy of subspace ensemble is detailed in Algorithm 1, where $mSpace(\mathcal{F})$ denotes the combination of the projected components of the multiple subspaces in $\mathcal{F}$, and $mCluster$ denotes the clustering problem solver applied to $mSpace(\mathcal{F})$.

In addition to the above strategy of aggregating projected components, we can also progressively approximate the subspaces in the light of [8]. Specifically, suppose we have the approximation $\widehat{\mathbf{S}}^d$ for the first d subspaces $\mathbf{S}^1, \mathbf{S}^2, \cdots, \mathbf{S}^d$. To approximate the next new subspace $\mathbf{S}^{d+1}$, we compute the SVD $\mathbf{S} = \mathbf{U}\Sigma\mathbf{V}'$, where $\mathbf{S} = (\widehat{\mathbf{S}}^d, \mathbf{S}^{d+1})$ is concatenation of $\widehat{\mathbf{S}}^d$ and $\mathbf{S}^{d+1}$. Then the new approximation $\widehat{\mathbf{S}}^{d+1} = \mathbf{SP}$ where $\mathbf{P}$ are the top $d+1$ principal directions in $\mathbf{V}$.

Algorithm 1. The multiple subspaces clustering algorithm

Signature: $C = mCluster(\mathbf{X}, K, \epsilon)$
Input: The data matrix $\mathbf{X}$; The number of clusters K; The maximal error of numerical 1-dimensional subspaces ϵ.
Output: The clustering C.

1. Construct subspaces $\mathcal{F} \leftarrow N1dSpaces(\mathbf{X}, \epsilon)$.
2. **for** Each subspace $\mathbf{S} \in \mathcal{F}$ **do**
3. Compute the first singular vector $\mathbf{v}$ of $\mathbf{S}$.
4. Replace $\mathbf{S}$ in $\mathcal{F}$ by $\mathbf{Sv}$.
5. **end for**
6. Construct $\widehat{\mathbf{X}} = mSpace(\mathcal{F})$ by combining the approximated subspaces in $\mathcal{F}$.
7. Solve Problem 1 in the space $\widehat{\mathbf{X}}$ with the parameter K, e.g., compute the Kmeans clustering $C \leftarrow kmeans(\widehat{\mathbf{X}}, K)$.

Table 1. The characteristics of data sets

data	fbis	k1a	la1	re0	re1	wap
#doc	2463	2340	3204	1504	1657	1560
#term	2000	4707	6188	2886	3758	8460
#class	17	20	6	13	25	20
MinClass	38	9	273	11	10	5
MaxClass	506	494	943	608	371	341
Min/Max	0.075	0.018	0.290	0.018	0.027	0.015

The details are given in Algorithm 2, where $pSpace(\mathcal{F})$ denotes the approximation described above for the subspaces in $\mathcal{F}$, and $pCluster$ denotes the clustering problem solver applied to $pSpace(\mathcal{F})$.

5 Experimental Evaluation

5.1 Experimental Data Sets

For evaluation, we used six real data sets from different domains, all of which are available at the website of CLUTO [4]. Some characteristics of these data sets are shown in Table 1. One can see diverse characteristics in terms of size (#doc), dimension (#term), number of clusters (#class) and cluster balance are covered by the investigated data sets. The cluster balance is measured by the ratio MinClass/MaxClass, where MinClass and MaxClass are the sizes of the smallest class and the largest class, respectively.

5.2 Comparison of Performance Lift

To see how much improvements can be achieved by the subspaces, regardless which solver of Problem 1 is used, we compute the performance lift [14, 5] in

Algorithm 2. The progressive subspaces clustering algorithm

Signature: $C = pCluster(\mathbf{X}, K, \epsilon)$
Input: The data matrix $\mathbf{X}$; The number of clusters K; The maximal error of numerical 1-dimensional subspaces ϵ.
Output: The clustering C.

1. Construct subspaces $\mathcal{F} = \{\mathbf{S}^1, \mathbf{S}^2, \cdots\} \leftarrow N1dSpaces(\mathbf{X}, \epsilon)$.
2. Order subspaces in $\mathcal{F}$ by the descending order of numerical 1-dimensional error.
3. Initialize the $pSpace(\mathcal{F})$ as $\widehat{\mathbf{X}} \leftarrow ()$, i.e., empty space.
4. $d \leftarrow 0$.
5. **repeat**
6. $\quad d \leftarrow d + 1$.
7. $\quad \mathbf{S} \leftarrow (\widehat{\mathbf{X}}, \mathbf{S}^d)$.
8. $\quad$ Compute the first d singular vectors $\mathbf{P}$ of $\mathbf{S}$.
9. $\quad \widehat{\mathbf{X}} \leftarrow \mathbf{SP}$.
10. **until** d reaches the number of subspaces in $\mathcal{F}$
11. Solve Problem 1 in the space $\widehat{\mathbf{X}}$ with the parameter K, e.g., compute the Kmeans clustering $C \leftarrow kmeans(\widehat{\mathbf{X}}, K)$.

the approximated subspaces. Specifically, the performance lift can be defined by the expectation: $\text{lift}(\mathbf{X}|Y) = \text{E}[\frac{\text{WCSS}(\mathbf{X}|C)}{\text{WCSS}(\mathbf{X}|Y)}]$ where C is a random clustering assignments for the data set $\mathbf{X}$ and Y is the true class labels. The performance lift actually represents the difference between the ground truth of the clustering structure and the random clustering solution. The higher the lift is, the easier it will be for the solver of Problem 1 to find the optimal solutions. Thus, we can use this lift to see which subspaces help most. To estimate the $\text{lift}(\mathbf{X}|Y)$, we can generate T (e.g., 10) random clustering assignments $\{C_1, \cdots, C_T\}$, and compute the average: $\frac{1}{T}\sum_{t=1}^{T}\frac{\text{WCSS}(\mathbf{X}|C_t)}{\text{WCSS}(\mathbf{X}|Y)}$. In Figure 2, we show the performance lifts in different approximated subspaces for all the data sets.

Specifically, we generate $T = 10$ random clustering assignments to estimate the performance lift. By controlling the error ϵ used in $\mathcal{F} = N1dSpaces(\mathbf{X}, \epsilon)$, we can construct approximation $mSpace(\mathcal{F})$ and $pSpace(\mathcal{F})$ with different dimensions, e.g., $d = 100, 200, \cdots, 1000$. For comparison, we also compute the performance lifts with top d principal components constructed by simple PCA, as denoted by "PC" in Figure 2. The line denoted by "RP" stands for Random Projection [1], which constructs the low dimensional approximation of $\mathbf{X} \in \mathbb{R}^{N\times D}$ by $\mathbf{X\Omega}$ where $\mathbf{\Omega} \in \mathbb{R}^{D\times d}$ is random matrix with entries $+1/\sqrt{d}$ or $-1/\sqrt{d}$ with equal probability. We can see that $mSpace$, $pSpace$, and PC are all effective to boost the performance lift. Also, while $mSpace$ and $pSpace$ outperform others consistently, $mSpace$ achieves significantly higher lift of performance.

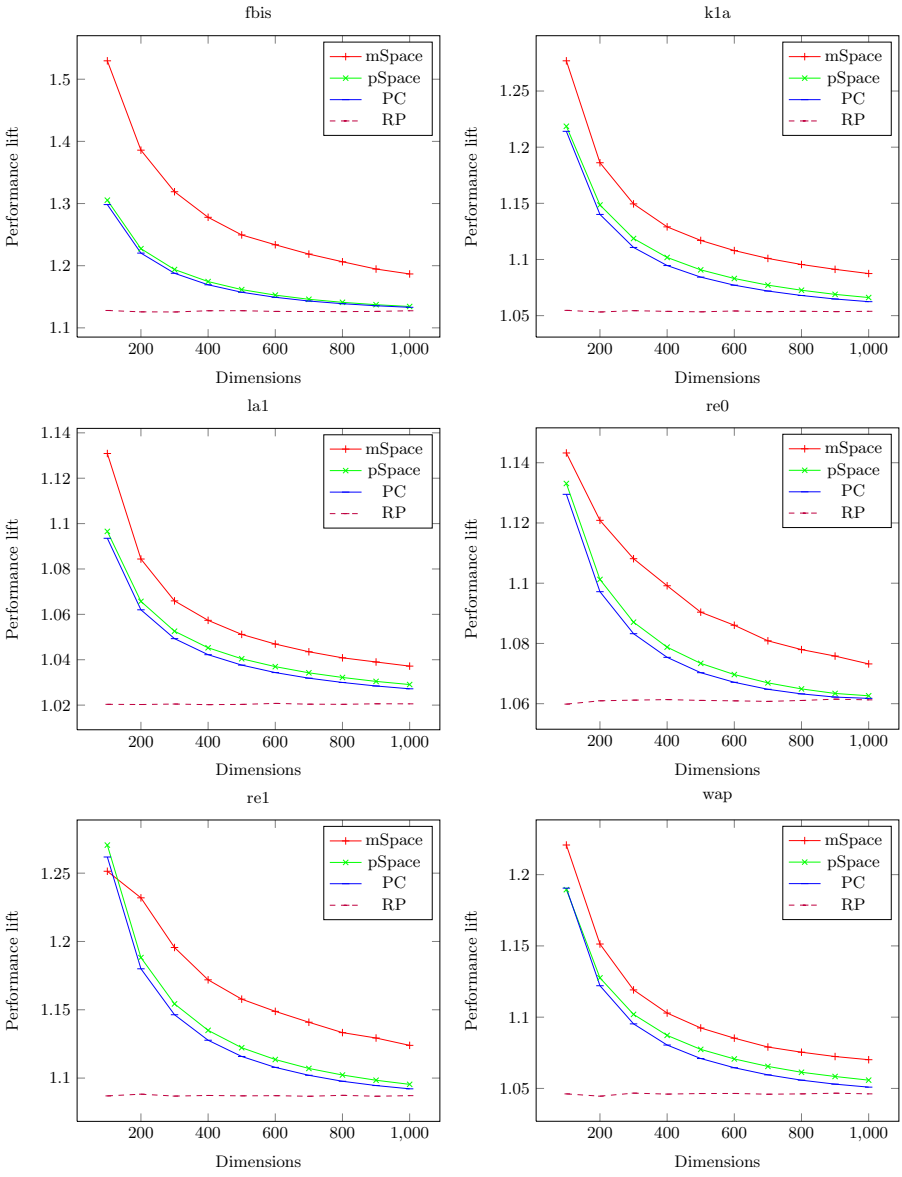

Fig. 2. The performance lift in different subspaces

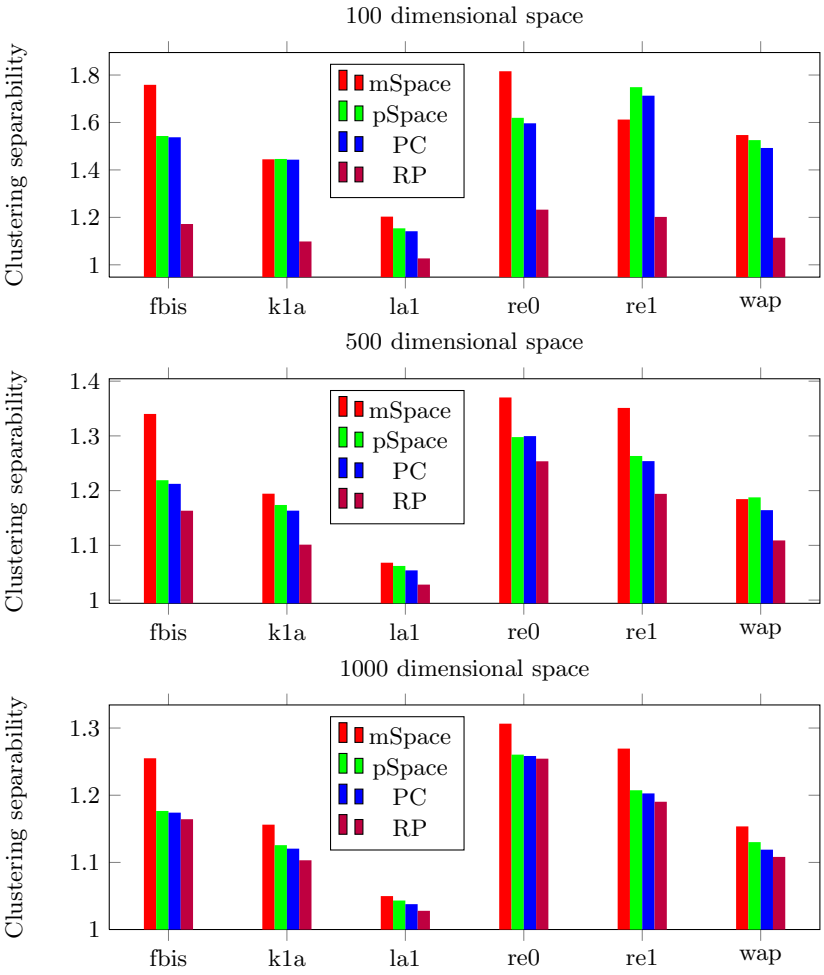

Fig. 3. The clustering separability in different subspaces

5.3 Comparison of Clustering Separability

Table 2. The clustering separability in 500 dimensional subspaces constructed with different methods on 'la1'

Cluster ID	Cluster Label	mSpace	pSpace	PC	RP
1	Entertainment	1.1596	**1.2010**	1.1482	1.0328
2	Financial	**1.0628**	1.0287	1.0372	1.0299
3	Foreign	1.0160	1.0176	1.0223	**1.0355**
4	Metro	1.0287	**1.0442**	1.0395	1.0136
5	National	1.0225	**1.0418**	1.0347	1.0191
6	Sports	**1.1200**	1.0410	1.0440	1.0390
Average		**1.0684**	1.0624	1.0543	1.0283

Adopted in [4, 5], one can investigate the data separability for the unsupervised clustering problem. Specifically, for each cluster C_i in the clustering solution $\{C_1, \cdots, C_K\}$, we can compute the ratio $\frac{EDis(C_i)}{IDis(C_i)}$ of the average external distance, $EDis(C_i)$, over the average internal distance, $IDis(C_i)$. The average internal distance $IDis(C_i)$ is the average distance between the instances in C_i, and the average external distance $EDis(C_i)$ is the average distance between the instances in C_i and the instances in the rest of the clusters C_j where $j \neq i$. The higher the ratio is for a cluster, the more compact and isolated the cluster will be, which, in turn, makes it easier for a clustering solver to identify the cluster. The ratio results of data set 'la1' are listed in Table 2, which clearly indicates that *mSpace* and *pSpace* provide better clustering separability. The last row also reports the average separability for the 6 clusters, where *mSpace* performs best. Besides, Figure 3 shows the average cluster separability for all of the six data sets. One can see that *mSpace* performs best on all data sets with few exceptions where *pSpace* performs better.

5.4 Analysis of Computational Cost

To reduce the dimensionality of the data matrix $\mathbf{X} \in \mathbb{R}^{N \times D}$ to d, our methods first construct the numerical 1-dimensional spaces. In the general case, the complexity of this step is $O(D^2)$, as we discussed in Section 3. To construct *mSpace*, we need to perform SVD further d times in the subspaces, each of $O(N)$ time, since most of the subspaces are of very low dimensions and we only need the first principal component. Thus the total computational cost for *mSpace* is $O(D^2 + dN)$. For *pSpace*, the computational complexity of the progressive SVD is costly $O(d^2N)$ and the total cost is $O(D^2 + d^3N)$. For PCA, we have the computational cost of $O(N^2D)$ when $N \leq D$ or $O(ND^2)$ when $D \leq N$. In our experiments, since most of the data sets are very high dimensional, we have the order of computational cost for the evaluated methods: $RP < PC < mSpace < pSpace$, which aligns with the order of performance.

6 Concluding Remarks

In this paper, we proposed a numerical one-dimension approach to high dimensional data clustering. An efficient correlation-based method was provided to construct the numerical one-dimensional subspace, which is well-clusterable and thus makes the clustering stable. Also, we discussed two strategies to collaboratively aggregate them into the final projected space. The experiments on real-world data sets demonstrated that such transformed data aligns better with the true class labels with respect to clustering.

This paper focused on the collaborative ensembling of the one-dimensional subspaces. For the future work, we plan to investigate collaboratively building clustering submodels directly in each of these one-dimensional subspaces. In the literature, this is related to the areas of multiple clustering [19, 6], clustering kernel [16] and clustering ensembles [13, 2, 15], where there are still many open problems to be answered.

References

1. Boutsidis, C., Zouzias, A., Drineas, P.: Random projections for k-means clustering. In: NIPS, pp. 298–306 (2010)
2. Fred, A.L.N., Jain, A.K.: Combining multiple clusterings using evidence accumulation. IEEE Transactions on Pattern Analysis and Machine Intelligence 27(6), 835–850 (2005)
3. Gönen, M., Alpaydın, E.: Multiple kernel learning algorithms. The Journal of Machine Learning Research 12, 2211–2268 (2011)
4. Karypis, G.: CLUTO: Data clustering software, `http://glaros.dtc.umn.edu/gkhome/views/cluto`
5. Liu, C., Hu, T., Ge, Y., Xiong, H.: Which distance metric is right: An evolutionary k-means view. In: SDM, pp. 907–918 (2012)
6. Liu, C., Xie, J., Ge, Y., Xiong, H.: Stochastic unsupervised learning on unlabeled data. Journal of Machine Learning Research - Proceedings Track 27, 111–122 (2012)
7. Liu, G., Lin, Z., Yu, Y.: Robust subspace segmentation by low-rank representation. In: ICML, vol. 3 (2010)
8. Liu, J., Chen, S., Zhou, Z.-H.: Progressive principal component analysis. In: Yin, F.-L., Wang, J., Guo, C. (eds.) ISNN 2004. LNCS, vol. 3173, pp. 768–773. Springer, Heidelberg (2004)
9. Liu, Y., Li, Z., Xiong, H., Gao, X., Wu, J.: Understanding of internal clustering validation measures. In: ICDM, pp. 911–916. IEEE (2010)
10. Park, M.Y., Hastie, T., Tibshirani, R.: Averaged gene expressions for regression. Biostatistics 8(2), 212–227 (2007)
11. Rangan, A.V.: Detecting low-rank clusters via random sampling. Journal of Computational Physics 231(1), 215–222 (2012)
12. Schonlau, M.: The clustergram: A graph for visualizing hierarchical and nonhierarchical cluster analyses. The Stata Journal 3, 316–327 (2002)
13. Strehl, A., Ghosh, J.: Cluster ensembles—a knowledge reuse framework for combining multiple partitions. The Journal of Machine Learning Research 3, 583–617 (2003)

14. Strehl, A., Ghosh, J., Mooney, R.: Impact of similarity measures on web-page clustering. In: Workshop on Artificial Intelligence for Web Search (AAAI 2000), pp. 58–64 (2000)
15. Topchy, A., Jain, A.K., Punch, W.: Clustering ensembles: Models of consensus and weak partitions. IEEE Transactions on Pattern Analysis and Machine Intelligence 27(12), 1866–1881 (2005)
16. Weston, J., Leslie, C., Zhou, D., Elisseeff, A., Noble, W.S.: Semi-supervised protein classification using cluster kernels. In: Thrun, S., Saul, L., Schölkopf, B. (eds.) NIPS (2003)
17. Wu, J., Xiong, H., Chen, J.: Adapting the right measures for k-means clustering. In: SIGKDD, pp. 877–886. ACM (2009)
18. Wu, X., et al.: Top 10 algorithms in data mining. Knowledge and Information Systems 14(1), 1–37 (2008)
19. Zhang, J., et al.: Pattern classification of large-scale functional brain networks: Identification of informative neuroimaging markers for epilepsy. PloS One 7(5), e36733 (2012)