



Adaptive multi-view subspace clustering for high-dimensional data

Fei Yan, Xiao-dong Wang*, Zhi-qiang Zeng, Chao-qun Hong

College of Computer and Information Engineering, Xiamen University of Technology, Xiamen 361024, China

ARTICLE INFO

Article history:

Available online 29 January 2019

Keywords:

Subspace clustering
Multi-view clustering
Adaptive learning
Feature selection

ABSTRACT

With the rapid development of multimedia technologies, we frequently confront with high-dimensional data and multi-view data, which usually contain redundant features and distinct types of features. How to efficiently cluster such kinds of data is still a great challenge. Traditional multi-view subspace clustering aims to determine the distribution of views by extra empirical parameters and search the optimal projection matrix by eigenvalue decomposition, which is impractical for real-world applications. In this paper, we propose a new adaptive multi-view subspace clustering method to integrate heterogeneous data in the low-dimensional feature space. Concretely, we extend *K*-means clustering with feature learning to handle high-dimensional data. Besides, for multi-view data, we evaluate the weights of distinct views according to their compactness of the cluster structure in the low-dimensional subspace. We apply the proposed method to four benchmark datasets and compare it with several widely used clustering algorithms. Experimental results demonstrate the effectiveness of the proposed method.

© 2019 Elsevier B.V. All rights reserved.

1. Introduction

Clustering is one of the most popularly used technologies in machine learning, which aims to partition data objects into several groups such that objects within the same group are close to each other while dissimilar objects in different clusters are far away from each other. With several decades' efforts, a series of remarkable algorithms have been yielded, such as *K*-means (KM) [1] clustering, spectral clustering [2] and graph-based clustering [3], and have been shown their success in various kinds of applications, including event detection [4] and multimedia understanding [5].

Along with the rapid development of the social network and computer technologies, a large number of multimedia data with high dimensionality are explosively generated in real-world applications. For example, in face recognition applications, given a face image data with a relatively small resolution 128×128 , it will generate a 16384-dimensional feature vector. It is hard to directly cluster these high-dimensional data with plenty of noises, outliers and redundant features, which easily leads to high computational complexity and degraded performance [6,7]. To solve this problem, a straightforward way is to first project the original data to a low-dimensional subspace by dimension reduction, such as PCA [8] or LDA [9] type methods, and to perform clustering subsequently. However, some researchers claim that the separation of dimension reduction and clustering may deteriorate the performance of clustering [10].

In other words, the obtained embedded feature space may not be the optimal one for clustering.

Another solution for high-dimensional problem is called subspace clustering, which tries to search the optimal cluster structure of data in the low-dimensional feature space. For instance, Ding et al. [11] built an adaptive clustering framework, where KM and LDA are performed jointly. Finding that PCA has close relation with KM, Hou et al. [12] proposed a general subspace clustering, which can flexibly explore the discriminative information among data in the embedded feature space. Feature selection is a powerful method to find the most representative feature subsets as well as to preserve the original structure of data [13,14]. Recently, researchers studied to integrate clustering with feature selection [15,16]. For example, Shang et al. [15] proposed a graph-based subspace clustering, where they built a feature map to explore the geometric structure information and to constrain the feature selection with group sparse learning. To reduce the influence of the data representation selection, Zeng et al. [16] incorporated a kernel based weighted regularization approach, which could iteratively shrink the weights for irrelevant features to zero. However, it depends on the graph or kernel construction, making it hard to deal with large-scale datasets.

In addition, data usually are collected from various feature extraction ways. Taking the web page classification for example, one can use many heterogeneous data or multi-view data, such as texts, musics and videos, to describe the web pages. As these multi-view data could offer different perspectives for the same instance, it could get more abundant information compared with the single-view data. However, due to the non-comparability among

* Corresponding author.

E-mail address: xdwangjsj@xmut.edu.cn (X.-d. Wang).

different views, it is still a significant research challenge to combine multiple views for clustering tasks. One of the earlier reports is to extend the traditional single-view clustering methods to the multi-view scenario by simply concatenating the heterogeneous data. Nevertheless, such a rough combination ignores the difference among views and lacks physical meaning. To fix this problem, some efforts attempt to investigate the Spectral Learning with the Multi-view Clustering (SLMC) [17]. For example, Bai et al. [17] proposed to leverage the complementary information among different views by a similarity framework. Zhuge et al. [18] introduced an auto-weighted multi-view subspace clustering, which utilizes a common representation matrix to reflect the sharing structure among views. Luo et al. [19] designed an adaptive loss function to automatically explore the local structure of data. However, the above methods need to build the similarity matrix, which is impractical for large-scale data. Although the adaptive weighting strategy proposed in [20] is independent of similarity matrix and is suitable for large-scale data, it only works in a supervised manner.

KM type Multi-view Clustering (KMMC) is another school of methods to deal with heterogeneous features. Compared with SLMC, KMMC is widely used for its simplicity and efficiency. For instance, Cai et al. [21] proposed a Robust Multi-view K -means Clustering (RMKMC), which extended traditional KM to multi-view application scenario with a group sparsity based loss function. However, it only performs clustering in the original feature space, which is inefficient to handle high-dimensional data. For this reason, Xu et al. [22] proposed a Discriminatively Embedded K -means for Multi-view clustering (DEKM), which imposes LDA into the K -means multi-view clustering. To reduce the complexity of parameter selection, they also proposed a re-weighted multi-view clustering approach [23]. Nevertheless, these algorithms depend on the eigenvalue decomposition to obtain the projection matrix, which is time-consuming to handle the high-dimensional data and is easy to get a trivial solution when dealing with “small-sample-size” data as well.

In this paper, we propose a new method called Adaptive Multi-view Subspace Clustering (AMSC). Our method is built on two advancements of the state-of-the-art: First, to solve high-dimensional data problem, AMSC imposes a special feature subset selection matrix, which is easy to be optimized and also can eliminate the dependency on the eigenvalue decomposition, thus it can be easily applied to high-dimensional data; Second, AMSC is able to adaptively calculate the weights for different views, which is quite suitable for the unsupervised clustering tasks.

The rest of this paper is organized as follows. The related works are first briefly reviewed in Section 2. We derive and optimize the proposed multi-view clustering model in Sections 3 and 4. Section 5 presents our experimental results, followed by the conclusion in Section 6.

2. Related works

In this section, we briefly review the related works on K -means type multi-view clustering methods, which are most relevant to our work.

2.1. K -means clustering

Given a dataset $X \in R^{D \times n}$, where n is the number of instances and D is the number of dimensionality. Let Ind be the cluster indicator matrix set. Traditional K -means aims to partition X into k classes $\{C_1, C_2, \dots, C_k\}$ such that the distance between each data point and cluster centroids is minimized:

$$\min_{h_k, F \in Ind} \sum_k \sum_{i \in C_k} \|x_i - h_k\|_2^2. \quad (1)$$

where $F \in R^{n \times c}$ is the cluster indicator matrix, h_k is the cluster centroid of k th cluster.

Previous work showed that K -means has a close relation with non-negative matrix factorization. Thus, we can reformulate the above objective function of K -means as follows:

$$\min_{H, F \in Ind} \|X^T - FH^T\|_F^2. \quad (2)$$

where $H = \{h_1, h_2, \dots, h_c\} \in R^{D \times c}$ is cluster centroid matrix.

2.2. Robust multi-view K -means clustering

As the objective function of traditional KM involves the squared Euclidean distance, it is sensitive to outliers. Besides, it only works for the single-view data clustering. To solve these problems, RMKMC was proposed. Suppose that we have n samples collected from c classes and each sample contains M types of heterogeneous features. Let $X_{(v)} \in R^{D_v \times n}$ be the data matrix and $H_{(v)} \in R^{D_v \times c}$ be the cluster centroid matrix for the v th view respectively. RMKMC aims to minimize the following objective function:

$$\min_{W_{(v)}, H_{(v)}, F \in Ind} \sum_{v=1}^M (\alpha_{(v)})^\gamma \|X_{(v)}^T - F_{(v)} H_{(v)}^T\|_{2,1}. \quad (3)$$

where $\alpha_{(v)}$ is the weight factor for the v th view and γ is the parameter to control the weights distribution of different views.

2.3. Discriminatively embedded K -means for multi-view clustering

Although RMKMC is robust to outliers, it performs clustering in the original high-dimensional feature space, which usually contains plenty of noises and redundant features. To address this problem, DEKM was proposed. It aims to integrate the subspace learning into K -means type multi-view clustering as follows:

$$\min_{W_{(v)}, G_{(v)}, F \in Ind} \sum_{v=1}^M (\alpha_{(v)})^\gamma \frac{\|X_{(v)}^T W_{(v)} - F_{(v)} G_{(v)}^T\|_F^2}{W_{(v)}^T S_{(v)}^{(v)} W_{(v)}}. \quad (4)$$

where $W_{(v)} \in R^{D_{(v)} \times d_{(v)}}$ and $d_{(v)}$ are the projection matrix and reduced dimensionality for v th view respectively, $S_{(v)}^{(v)} = X_{(v)} X_{(v)}^T$ is the total scatter matrix in the v th view, $G_{(v)} = \{g_{(v)}^1, g_{(v)}^2, \dots, g_{(v)}^c\} \in R^{d_{(v)} \times c}$ is the embedded cluster centroid matrix for v th view.

In despite of considering the discriminative information among different views, DEKM is inefficient to deal with high-dimensional data since it depends on eigenvalue decomposition, which needs a high computational cost.

3. The proposed framework

The key issue for multi-view clustering is to properly explore the sharing information and complementary information among data. The most popular techniques are usually designed to impose an extra parameter to control the distribution of distinct views. Nonetheless, such approaches need to manually tune the optimal parameters, which are unrealistic for unsupervised applications. To achieve this goal, inspired by the development of auto-weight learning [18,23–25], we propose to automatically determine the importance of different views by their intrinsic structure. Concretely, we aim to minimize the following objective function:

$$\min_{G_{(v)}, F \in Ind} \sum_{v=1}^M \sqrt{\|X_{(v)}^T - F G_{(v)}^T\|_{2,1}}. \quad (5)$$

where F is the cluster indicator matrix shared by all the views.

Moreover, to efficiently deal with high-dimensional data and to reduce the computational complexity involved by the eigenvalue

decomposition, following previous feature selection works [26–28], we extend Eq. (5) to multi-view subspace clustering as follows:

$$\min_{W_{(v)}, G_{(v)}, F \in \text{Ind}} \sum_{v=1}^M \sqrt{\|X_{(v)}^T W_{(v)} - F G_{(v)}^T\|_{2,1}}. \quad (6)$$

where $W_{(v)} = \{w_{(v)}^1, w_{(v)}^2, \dots, w_{(v)}^{d_{(v)}}\} \in \mathbb{R}^{D_{(v)} \times d_{(v)}}$ is the feature selection matrix for v th view, whose i th column vector $w_{(v)}^i$ is defined as:

$$w_{(v)}^i = [\underbrace{0, \dots, 0}_{i-1}, 1, \underbrace{0, \dots, 0}_{D_{(v)}-i}]^T. \quad (7)$$

From Eq. (7), one can easily find that $W_{(v)}$ is extremely sparse and is actually a column-full-rank projection matrix, which is quite suitable for feature selection and can be efficiently optimized. We will give a detailed explanation in Section 4.

4. Optimization

Note that the framework in Eq. (6) is different from the previous works [25], which involves the $l_{2,1}$ -norm and is non-smooth. Besides, it is also not joint convex with respect to all the variables. In this section, we propose an alternative optimization method to solve this problem. More concretely, we alternate between optimizing $W_{(v)}$ and $F, G_{(v)}$ for each view as follows. Before that, we first rewrite the objective function in Eq. (6) as:

$$\min_{W_{(v)}, G_{(v)}, \alpha_{(v)}, F \in \text{Ind}} \sum_{v=1}^M \alpha_{(v)} \|X_{(v)}^T W_{(v)} - F G_{(v)}^T\|_{2,1}. \quad (8)$$

where $\alpha_{(v)}$ is:

$$\alpha_{(v)} = \frac{1}{2\sqrt{\|X_{(v)}^T W_{(v)} - F G_{(v)}^T\|_{2,1}}}. \quad (9)$$

Then we rewrite Eq. (8) as:

$$\min_{W_{(v)}, G_{(v)}, \alpha_{(v)}, \Delta_{(v)}, F \in \text{Ind}} \sum_{v=1}^M \alpha_{(v)} \text{Tr}(E_{(v)}^T \Delta_{(v)} E_{(v)}). \quad (10)$$

where $E_{(v)} = \{e_{(v)}^1, e_{(v)}^2, \dots, e_{(v)}^{d_{(v)}}\}^T = X_{(v)}^T W_{(v)} - F G_{(v)}^T$ with $e_{(v)}^i \in \mathbb{R}^{d_{(v)} \times 1}$ is the i th column vector of E , and $\Delta_{(v)}$ is a diagonal matrix with its i th element defined as:

$$\Delta_{(v)}^{ii} = \frac{1}{2\|e_{(v)}^i\|_2}. \quad (11)$$

Based on the above objective function reformulation in Eq. (10), we give the following optimization steps.

Step 1: Fixing $\alpha_{(v)}, W_{(v)}, G_{(v)}, \Delta_{(v)}$ and optimizing F

Let $\tilde{\Delta}_{(v)} = \alpha_{(v)} \Delta_{(v)}$, we have:

$$\begin{aligned} & \sum_{v=1}^M \text{Tr}((X_{(v)}^T W_{(v)} - F G_{(v)}^T)^T \tilde{\Delta}_{(v)} (X_{(v)}^T W_{(v)} - F G_{(v)}^T)) \\ &= \sum_{v=1}^M \sum_{i=1}^{d_{(v)}} \tilde{\Delta}_{(v)}^{ii} \|W_{(v)}^T x_{(v)}^i - G_{(v)} f_i\|_2^2 \\ &= \sum_{i=1}^n \left(\sum_{v=1}^M \tilde{\Delta}_{(v)}^{ii} \|W_{(v)}^T x_{(v)}^i - G_{(v)} f_i\|_2^2 \right). \end{aligned} \quad (12)$$

Considering that each column vector of F satisfies 1-of- K coding scheme, Eq. (12) can be solved by performing the traditional K -means on data points in the embedded feature space $W_{(v)}^T x_{(v)}^i$ as follows:

$$F_{ij} = \begin{cases} 1, & j = \arg \min_k \sum_{v=1}^M \tilde{\Delta}_{(v)}^{ii} \|W_{(v)}^T x_{(v)}^i - g_{(v)}^k\|_2^2 \\ 0, & \text{Otherwise.} \end{cases} \quad (13)$$

Step 2: Fixing $\alpha_{(v)}, F, \Delta_{(v)}$ and optimizing $W_{(v)}, G_{(v)}$

Taking the derivative of the objective function in Eq. (10) with respect to $G_{(v)}$, we get:

$$G_{(v)} = W_{(v)}^T X_{(v)} \tilde{\Delta}_{(v)} F (F^T \tilde{\Delta}_{(v)} F)^{-1} \quad (14)$$

Substituting Eq. (14) into Eq. (10), the objective function becomes:

$$\begin{aligned} & \min_{W_{(v)}} \text{Tr}(W_{(v)}^T S_{(v)} W_{(v)}) \\ &= \min_{W_{(v)}} \sum_{i=1}^{d_{(v)}} \text{Tr}((w_{(v)}^i)^T S_{(v)} (w_{(v)}^i)). \end{aligned} \quad (15)$$

where $S_{(v)} = X_{(v)} N_{(v)} X_{(v)}^T$ and $N_{(v)} = \tilde{\Delta}_{(v)} - \tilde{\Delta}_{(v)} F (F^T \tilde{\Delta}_{(v)} F)^{-1} F^T \tilde{\Delta}_{(v)}$.

Considering the particular structure of $W_{(v)}$, the optimal solution of the problem in Eq. (15) can be effectively obtained by locating the first $d_{(v)}$ smallest diagonal elements of matrix $S_{(v)}$.

Step 3: Fixing $\alpha_{(v)}, F, W_{(v)}, G_{(v)}$, and updating $\Delta_{(v)}$ by Eq. (11).

Step 4: Fixing $F, W_{(v)}, G_{(v)}, \Delta_{(v)}$, and updating $\alpha_{(v)}$ by Eq. (9).

By applying the above four steps, we alternatively update all the variables and repeat the process iteratively until the objective function becomes converged. We summarize the proposed algorithm in Algorithm 1.

Algorithm 1 The iterative optimization algorithm of AMSC.

Input:

- Data For M views $\{X_{(1)}, X_{(2)}, \dots, X_{(M)}\}$ with v th view $X_{(v)} \in \mathbb{R}^{d_{(v)} \times n}$
- The number of clusters c .
- The reduced dimensionality d_v for each view.

Output:

Optimized feature subset selection matrix W_v , cluster indicator matrix F and the cluster centroid matrix $G_{(v)}$.

- 1: Set $t = 0$ and initialize $F \in \text{Ind}$.
 - 2: **repeat**
 - 3: Calculate F by Eq.(13).
 - 4: Calculate $G_{(v)}$ by $G_{(v)} = W_{(v)}^T X_{(v)} \tilde{\Delta}_{(v)} F (F^T \tilde{\Delta}_{(v)} F)^{-1}$ and update $W_{(v)}$ by minimizing the objective function in Eq.(15).
 - 5: Update $\Delta_{(v)}$ by Eq.(11).
 - 6: Update $\alpha_{(v)}$ by Eq.(9).
 - 7: $t = t + 1$;
 - 8: **until** Convergence
 - 9: Return $W_{(v)}, G_{(v)}, F$.
-

5. Experiments

In this section, we present empirical results to evaluate AMSC on four benchmark datasets in terms of three standard clustering evaluation metrics, i.e. Normalized Mutual Information (NMI) [29], Accuracy (ACC) [30] and Purity [31]. We also compare AMSC with several baseline single-view clustering algorithms, i.e. KM, and multi-view clustering algorithms, i.e. RMKMC, SMKMC [21] (a variant of RMKMC with equal weight for each view), DEKM.

For a fair comparison, following [21], we utilize the same cluster indicator matrix initialization approach for all the compared algorithms. For the subspace multi-view clustering, i.e. DEKM and AMSC, we determine the optimal reduced dimensionality for each view by the grid search strategy. For DEKM and RMKMC, we tune

Table 1

A brief description of the selected datasets.

View	Handwritten	Caltech	Jaffe	Yale
1	PIX(240)	GAB(48)	CLR(420)	CLR(420)
2	FOU(76)	WM(40)	GIST(512)	GIST(512)
3	FAC(216)	CENT(254)	SIFT(420)	SIFT(420)
4	ZER(47)	-	-	-
5	KAR(64)	-	-	-
6	MOR(6)	-	20	-
Samples	2000	1474	213	165
Classes	10	7	10	15

the view distribution parameter, i.e. γ , following the author's suggestions. All of the compared algorithms are independently repeat 50 times and the mean and standard deviation of the results are reported.

5.1. Datasets

In our experiments, four benchmark datasets, including Caltech [32], Handwritten [33], Jaffe [34] and Yale [35] were selected for evaluations (Table 1). The brief description of these datasets is reported as follows:

1) *Caltech*: It is an image dataset that contains 8677 images collected from 101 categories. We chose the most frequently used 7 classes, i.e. Face, Motorbikes, DollaBill, Garfield, Snoopy, Stop-Sign and Windsor-chair, eventually getting 1474 images. Three features were extracted to represent each image: 48 Gabor feature (GAB), 40 Wavelet Moments (WM), 254 Centrist feature (CENT).

2) *Handwritten*: This dataset consists of features of handwritten digits of (0–9) and contains totally 2000 samples with 200 instances per class. We extracted six published features: 76 Fourier coefficients of the character shapes (FOU), 216 profile correlations (FAC), 64 Karhunen-lov coefficients (KAR), 240 pixel averages in 2×3 windows (PIX), 47 Zernike moment (ZER) and 6 morphological (MOR) features.

3) *Jaffe*: This is a facial expression dataset collected from 10 Japanese female models. Each image has been rated on 6 emotion adjectives by 60 Japanese subjects. Each image was represented in terms of three heterogeneous features: 420 Color names feature (CLR), 512 Gist features (GIST), 420 Sift features (SIFT).

4) *Yale*: It contains 165 grayscale images of 15 individuals with various facial expressions, facial configurations and light conditions: center-light, with glasses, happy, left-light, without glasses, normal, right-light, sad, sleepy, surprised, and wink. Similar to Jaffe dataset, three features were extracted from this dataset: 420 Color names feature (CLR), 512 Gist features (GIST), 420 Sift features (SIFT).

5.2. Toy example

To show the validity of the proposed algorithm, we conduct several experiments on a synthetic dataset given by Kumar et al. [36]. This synthetic dataset contains three views and is sampled from two classes, each of which consists of two components generated by the Gaussian mixture model. We set the reduced dimensionality for three views as $d_1 = 2$, $d_2 = 2$, $d_3 = 1$ respectively. From the clustering results shown in Table 2, we can find that all of the multi-view clustering algorithms get better results than KM, which indicates that exploring the shared information among multiple views of data is helpful for improving clustering performance. Besides, different views maintain different discriminative clustering power, e.g. KM gets the best results on view 1. Thus, the way of mining the discriminative power among different views plays an important role in multi-view clustering. For example, DEKM and RMKMC outperform SMKMC by imposing an additional view

Table 2

Clustering results comparison on the toy example.

Methods	NMI	ACC	Purity
KM(1)	91.02 $\pm$ 0.00	98.80 $\pm$ 0.00	98.80 $\pm$ 0.00
KM(2)	47.43 $\pm$ 0.00	85.70 $\pm$ 0.00	85.70 $\pm$ 0.00
KM(3)	48.92 $\pm$ 0.00	88.30 $\pm$ 0.00	88.30 $\pm$ 0.00
SMKMC	94.67 $\pm$ 7.03	99.21 $\pm$ 1.75	99.21 $\pm$ 1.75
RMKMC	94.98 $\pm$ 3.45	99.37 $\pm$ 0.56	99.37 $\pm$ 0.56
DEKM	96.21 $\pm$ 0.00	99.60 $\pm$ 0.00	99.60 $\pm$ 0.00
AMSC	98.31 $\pm$ 0.13	99.83 $\pm$ 0.00	99.83 $\pm$ 0.13

weighting strategy. Additionally, AMSC is able to adaptively determine the importance of different views according the nature cluster structure of each view, resulting in the best accuracy compared with the other baseline methods.

To verify the validity of the adaptive multi-view weighting mechanism in Algorithm 1, we also visually illustrate the weights among views, weights between data points (farthest 5 data points and nearest 5 data points to their centroids) and their cluster centroids during the iteration. We use line width to represent weight. Bigger line width indicates larger weight. As shown in Fig. 1(b), the weights for all the views are initially set to be identical, so do the weights for data points (have the same line width). After several iterations in Fig. 1(c) and (d), the data points and views that contributes most to minimizing the objective function will be assigned larger weights. Zoom in the figures will get better visualization.

5.3. Comparison results and discussion

Tables 3 and 4 show the clustering results of the compared algorithms on different datasets. We begin the comparison results with single-view clustering method, i.e. KM. We can observe that nearly all of the multi-view clustering get better results than KM, which indicates the effectiveness of multi-view learning. However, the way of utilizing the information among different views has a significant effect on the clustering performance. For example, SMKMC fails KM(1) and KM(2) on Jaffe, KM(2) on Yale. The primary reason is that simply combining different types of features may not be beneficial for clustering.

Moreover, for four compared multi-view clustering, in most of the cases AMSC performs better than the others. Specifically, compared with RMKMC, AMSC gets better results by a large margin, except that on Yale dataset in terms of NMI and on Handwritten dataset in terms of Purity, AMSC is slightly worse than RMKMC. Note that owing to the subspace learning component in AMSC, it can tremendously save the computational resource for clustering especially for high-dimensional data.

Finally, AMSC outperforms the multi-view subspace clustering algorithm, i.e. DEKM, which demonstrates the effectiveness of our objective function and the usefulness of the non-eigenvalue-decomposition subspace clustering.

5.4. Convergence and complexity analysis

Following the previous works [18,24], we can easily demonstrate that the convergence of Algorithm 1 theoretically. Due to space limits, we only conduct an experimental analysis on the speed of convergence. The number of reduced dimensionality is set to c , where c is the number of classes. Fig. 2 shows the results on four datasets. We can find that our algorithm converges fast, usually within 30 iterations for all the datasets.

To show the efficiency of our algorithm, we provide a brief time complexity analysis of Algorithm 1. For v th view of data with n samples, the complexity of our algorithm is $O(D_{(v)}n)$, which is linear with respect to the number of samples n and the number of

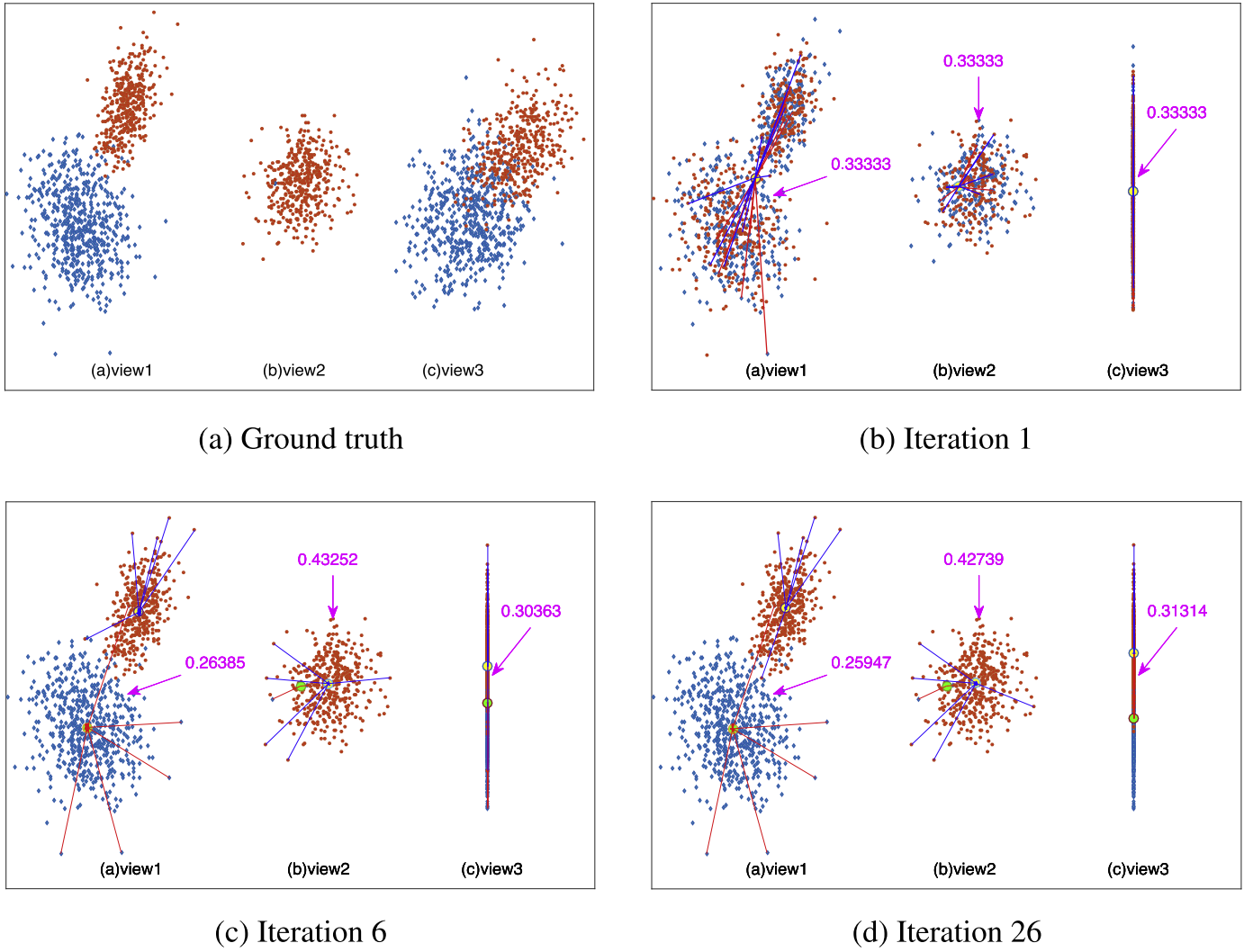


Fig. 1. The clustering results of toy example during the iteration by Algorithm 1.

Table 3

Clustering results of compared algorithms on Jaffe and Yale datasets.

Methods	Jaffe			Yale		
	NMI	ACC	Purity	NMI	ACC	Purity
KM(1)	89.24 ± 5.77	80.03 ± 9.81	84.00 ± 7.77	63.55 ± 4.26	52.93 ± 6.98	54.69 ± 6.02
KM(2)	88.09 ± 5.20	82.16 ± 8.17	84.98 ± 6.72	70.79 ± 4.42	62.22 ± 5.96	64.92 ± 5.07
KM(3)	82.40 ± 4.84	74.32 ± 8.58	78.87 ± 6.32	63.78 ± 2.93	51.98 ± 5.16	52.61 ± 4.89
SMKMC	87.45 ± 5.10	79.59 ± 7.83	83.04 ± 6.23	69.73 ± 3.59	62.18 ± 6.00	63.30 ± 5.67
RMKMC	89.77 ± 5.59	82.79 ± 9.51	85.87 ± 7.66	72.20 ± 3.65	64.79 ± 5.89	65.84 ± 5.58
DEKM	85.29 ± 5.37	77.56 ± 8.15	80.87 ± 7.10	62.68 ± 4.86	53.13 ± 7.04	54.12 ± 7.01
AMSC	90.64 ± 4.99	85.01 ± 8.19	87.56 ± 6.55	71.65 ± 3.29	64.96 ± 5.42	66.11 ± 5.25

Table 4

Clustering results of compared algorithms on Caltech and Handwritten datasets.

Methods	Caltech			Handwritten		
	NMI	ACC	Purity	NMI	ACC	Purity
KM(1)	14.93 ± 0.57	32.95 ± 1.81	69.39 ± 0.87	68.26 ± 3.99	68.63 ± 7.09	70.12 ± 6.06
KM(2)	29.28 ± 2.72	40.95 ± 2.80	79.59 ± 0.95	53.73 ± 2.42	55.86 ± 4.28	56.73 ± 3.95
KM(3)	30.79 ± 0.89	42.30 ± 4.87	79.20 ± 0.85	64.65 ± 2.96	67.53 ± 5.89	69.53 ± 4.50
KM(4)	-	-	-	49.99 ± 2.23	53.60 ± 5.30	55.78 ± 4.15
KM(5)	-	-	-	63.30 ± 2.53	65.71 ± 4.37	67.43 ± 3.77
KM(6)	-	-	-	64.11 ± 3.01	62.90 ± 5.61	64.53 ± 4.17
SMKMC	32.35 ± 2.67	45.41 ± 6.02	80.35 ± 1.24	78.68 ± 3.93	76.90 ± 6.76	80.01 ± 5.51
RMKMC	32.88 ± 3.25	47.19 ± 7.41	80.51 ± 1.28	81.34 ± 3.50	81.58 ± 7.67	84.26 ± 5.69
DEKM	32.07 ± 6.43	58.57 ± 8.35	80.79 ± 5.09	82.05 ± 3.36	78.97 ± 7.05	82.49 ± 5.25
AMSC	34.66 ± 3.24	49.16 ± 6.55	81.15 ± 1.40	82.52 ± 3.63	81.98 ± 6.47	82.55 ± 5.06

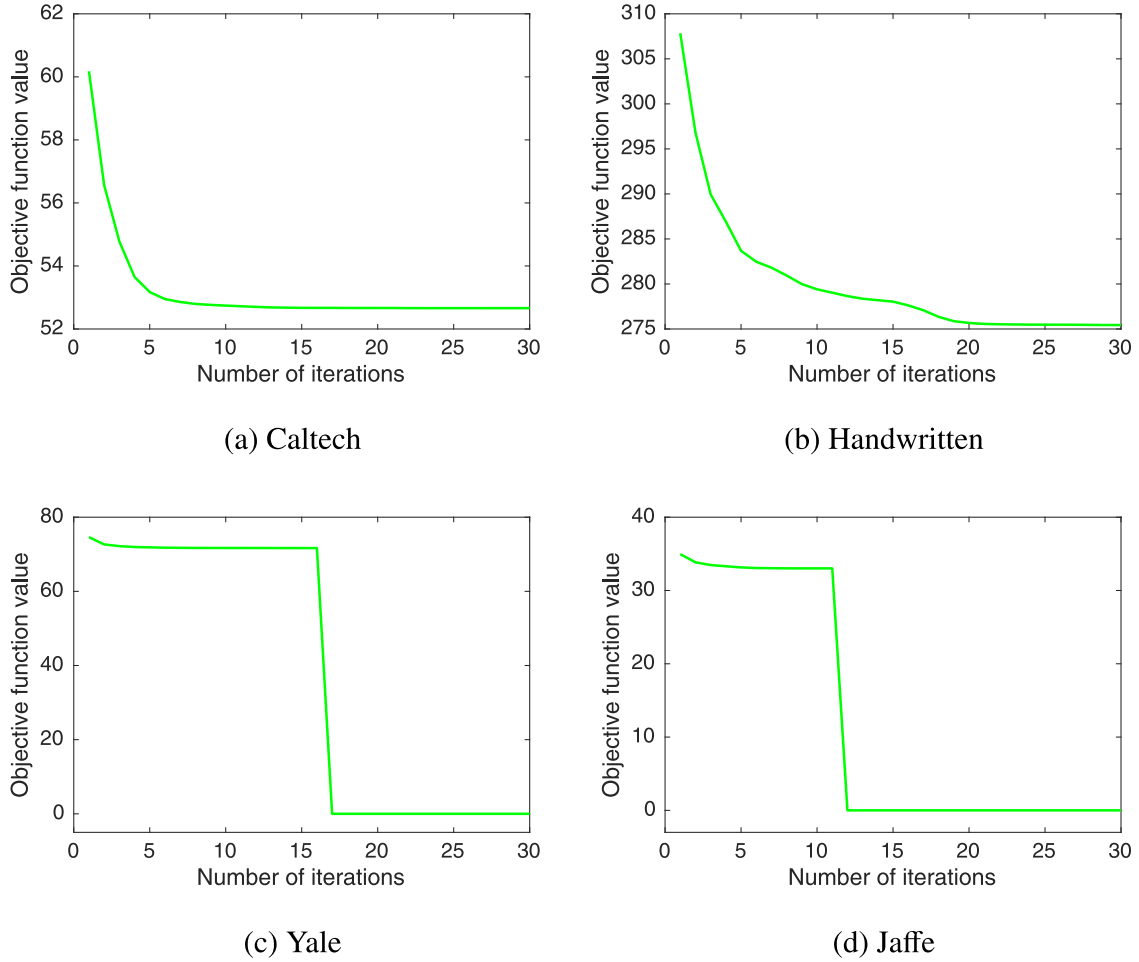


Fig. 2. The clustering results of toy example during the iteration by Algorithm 1.

Table 5
Computational time (in seconds) comparison.

Methods	Handwritten	Caltech	Jaffe	Yale
KM	0.82 ± 0.36	0.30 ± 0.16	0.10 ± 0.07	0.18 ± 0.09
SMKMC	10.16 ± 2.51	2.26 ± 0.81	0.66 ± 0.39	0.77 ± 0.32
RMKMC	6.77 ± 2.45	4.26 ± 0.96	0.39 ± 0.12	0.39 ± 0.08
DEKM	13.25 ± 1.56	4.40 ± 0.46	5.14 ± 0.16	5.14 ± 0.21
AMSC	2.58 ± 0.77	0.65 ± 0.22	0.06 ± 0.02	0.07 ± 0.03

dimensionality $D_{(v)}$. Thus, it can be used for large-scale datasets with high-dimensional features.

What is more, we also recorded the clustering time for experiment on four datasets. All the algorithms were realized by Matlab 2015b and executed on an Intel(R) Xeon(R) CPU E5-2630 V4 2.20GHz with 32G memory and ubuntu 16.04 LTS operating system. Note that the compared algorithms are standardly implemented or downloaded the source code from the author's web site, they might not be best tuned for efficiency. From Table 5, we can conclude that: 1) compared with RMKMC, SMKMC and DEKM, AMSC and KM achieve the remarkable efficiency on all of the datasets; 2) when dealing with low-dimensional data, i.e. Handwritten and Caltech, KM spends the least time, especially the number of features is smaller than that of instances. 3) when handling the high-dimensional data, i.e. Jaffe and Yale, AMSC performs faster than KM. The reason lies in that AMSC iteratively updates each cluster centroids in the embedded feature space, while KM operates in the original high-dimensional feature space.

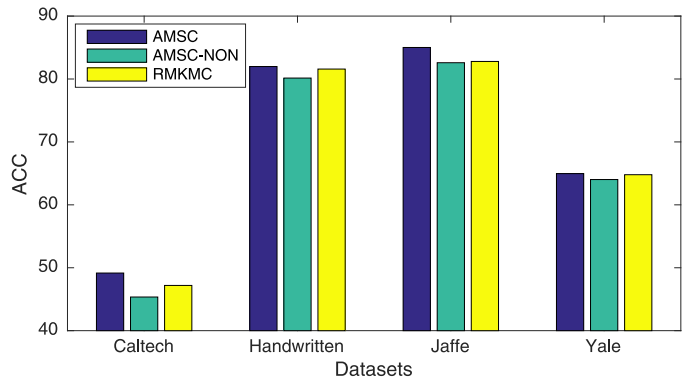


Fig. 3. Influence of subspace learning in AMSC.

5.5. Influence of subspace learning

In this section, we conduct a thorough experiment to study the influence of subspace learning in AMSC. We simply set the selection matrix in Eq. (6) as an identity matrix and generate the following equation:

$$\min_{G_{(v)}, F \in \text{Ind}} \sum_{v=1}^M \sqrt{\|X_{(v)}^T - FG_{(v)}^T\|_{2,1}}. \quad (16)$$

Obviously, without subspace learning AMSC is reduced to RMKMC with square root loss function and adaptive weighting

strategy, for better presentation, we named it AMRC-NON. The comparison results of AMSC, ARMC-NON and RMKMC are illustrated in Fig. 3. We can observe that AMSC outperforms AMSC-NON on all of the datasets, which demonstrates the importance of subspace learning. Besides, AMRC-NON gets competitive accuracy with RMKMC, especially when dealing with high-dimensional data, i.e. Jaffe and Yale. Such results confirm the validity of the adaptive weighting strategy in AMSC.

6. Conclusions

It still is a great challenge to deal with high-dimensional data along with heterogeneous features. In this paper, we propose a new subspace multi-view clustering. In contrast of manually selecting the distribution of different views in traditional approaches, we integrate an adaptive weighting mechanism in the proposed method, which can automatically determine the weights for views according to their own compactness of cluster structure. Besides, to efficiently handle high-dimensional data, we impose a special selection matrix, which can properly avoid the eigenvalue decomposition. The convergence behavior, computational complexity and the influence of subspace learning have also been analyzed. Experimental results on several benchmark datasets have demonstrated the effectiveness of the proposed method.

Acknowledgments

This work is supported by the National Natural Science Foundation of China, China (Grant No. 61871464), National Natural Science Foundation of Fujian Province, China (Grant No. 2017J01511), Scientific Research Fund of Fujian Provincial Education Department, China (Grant Nos. JAT170417, JAT160357), the “Climbing” Program of Xiamen university of technology (No. XPDQK18012).

References

- [1] S.Z. Selim, M.A. Ismail, K-means-type algorithms: a generalized convergence theorem and characterization of local optimality, *IEEE Trans. Pattern Anal. Mach. Intell.* 6 (1984) 81–87, doi:10.1109/TPAMI.1984.4767478.
- [2] X. Chang, F. Nie, Z. Ma, Y. Yang, X. Zhou, A convex formulation for spectral shrunk clustering, in: *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, in: AAAI'15, AAAI Press, 2015, pp. 2532–2538.
- [3] Y. Yang, D. Xu, F. Nie, S. Yan, Y. Zhuang, Image clustering using local discriminant models and global integration, *IEEE Trans. Image Process.* 19 (2010) 2761–2773, doi:10.1109/TIP.2010.2049235.
- [4] X. Chang, Z. Ma, Y. Yang, Z. Zeng, A.G. Hauptmann, Bi-level semantic representation analysis for multimedia event detection, *IEEE Trans. Cybern.* 47 (2017) 1180–1197, doi:10.1109/TCYB.2016.2539546.
- [5] X.-d. Wang, R.-C. Chen, F. Yan, Z.-q. Zeng, C.-q. Hong, Semi-supervised adaptive feature analysis and its application for multimedia understanding, *Multimedia Tools Appl.* (2017), doi:10.1007/s11042-017-4990-5.
- [6] X. Chang, F. Nie, S. Wang, Y. Yang, X. Zhou, C. Zhang, Compound rank- k projections for bilinear analysis, *IEEE Trans. Neural Netw. Learn. Syst.* 27 (2016) 1502–1513, doi:10.1109/TNNLS.2015.2441735.
- [7] Z. Ma, X. Chang, Y. Yang, N. Sebe, A.G. Hauptmann, The many shades of negativity, *IEEE Trans. Multimedia* 19 (2017) 1558–1568, doi:10.1109/TMM.2017.2659221.
- [8] X. Chang, F. Nie, Y. Yang, C. Zhang, H. Huang, Convex sparse PCA for unsupervised feature learning, *ACM Trans. Knowl. Discov. Data* 11 (2016) 1–16, doi:10.1145/2910585.
- [9] F. Nie, S. Xiang, Y. Liu, C. Hou, C. Zhang, Orthogonal vs. uncorrelated least squares discriminant analysis for feature extraction, *Pattern Recognit. Lett.* 33 (2012) 485–491, doi:10.1016/j.patrec.2011.11.028.
- [10] X.-d. WANG, R.-C. CHEN, C.-q. HONG, Z.-q. ZENG, Unsupervised feature analysis with sparse adaptive learning, *Pattern Recognit. Lett.* 102 (2018) 89–94, doi:10.1016/j.patrec.2017.12.022.
- [11] C. Ding, T. Li, Adaptive dimension reduction using discriminant analysis and K-means clustering, in: *Proceedings of the 24th International Conference on Machine Learning*, in: ICML '07, ACM, New York, NY, USA, 2007, pp. 521–528.
- [12] C. Hou, F. Nie, D. Yi, D. Tao, Discriminative embedded clustering: a framework for grouping high-dimensional data, *IEEE Trans. Neural Netw. Learn. Syst.* 26 (2015) 1287–1299.
- [13] X. Chang, F. Nie, Y. Yang, H. Huang, A convex formulation for semi-supervised multi-label feature selection, in: *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014, pp. 1171–1177.
- [14] X. Chang, Y. Yang, Semisupervised feature analysis by mining correlations among multiple tasks, *IEEE Trans. Neural Netw. Learn. Syst.* 28 (2017) 2294–2305, doi:10.1109/TNNLS.2016.2582746.
- [15] R. Shang, W. Wang, R. Stolkin, L. Jiao, Subspace learning-based graph regularized feature selection, *Knowl. Based Syst.* 112 (2016) 152–165, doi:10.1016/j.knsys.2016.09.006.
- [16] H. Zeng, Y.-M. Cheung, Feature selection and kernel learning for local learning based clustering, *IEEE Trans. Pattern Anal. Mach. Intell.* 33 (2010) 1532–1547, doi:10.1109/TPAMI.2010.215.
- [17] S. Bai, Z. Liu, X. Bai, Co-spectral for robust shape clustering, *Pattern Recognit. Lett.* 83 (2016) 388–394, doi:10.1016/j.patrec.2016.07.014.
- [18] W. Zhuge, C. Hou, Y. Jiao, J. Yue, H. Tao, D. Yi, Robust auto-weighted multi-view subspace clustering with common subspace representation matrix, *PLoS ONE* 12 (2017) e0176769, doi:10.1371/journal.pone.0176769.
- [19] M. Luo, X. Chang, L. Nie, Y. Yang, A.G. Hauptmann, Q. Zheng, An adaptive semisupervised feature analysis for video semantic recognition, *IEEE Trans. Cybern.* 48 (2018) 648–660, doi:10.1109/TCYB.2017.2647904.
- [20] C. Yan, M. Luo, H. Liu, Z. Li, Q. Zheng, Top-k multi-class svm using multiple features, *Inf Sci (Ny)* 432 (2018) 479–494, doi:10.1016/j.ins.2017.08.004.
- [21] X. Cai, F. Nie, H. Huang, Multi-view K-means clustering on big data, in: *The 23rd International Joint Conference on Artificial Intelligence*, 2013, pp. 2598–2604.
- [22] J. Xu, J. Han, F. Nie, Discriminatively Embedded K-Means for Multi-View Clustering, in: *Cvpr*, 2016, pp. 5356–5364, doi:10.1109/CVPR.2016.578.
- [23] J. Xu, J. Han, F. Nie, X. Li, Re-weighted discriminatively embedded K-means for multi-view clustering, *IEEE Trans. Image Process.* 26 (2017) 3016–3027, doi:10.1109/TIP.2017.2665976.
- [24] F. Nie, H. Huang, X. Cai, C.H. Ding, Efficient and Robust Feature Selection via Joint L2.1-Norms Minimization, in: *Advances in Neural Information Processing Systems* 23, Curran Associates, Inc., 2010, pp. 1813–1821.
- [25] F. Nie, J. Li, X. Li, Parameter-free auto-weighted multiple graph learning: a framework for multiview clustering and semi-supervised classification, in: *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, in: IJCAI'16, AAAI Press, 2016, pp. 1881–1887.
- [26] F. Nie, S. Xiang, Y. Jia, C. Zhang, S. Yan, Trace Ratio Criterion for Feature Selection, in: *Twenty-Third AAAI Conference on Artificial Intelligence*, 2008, pp. 671–676.
- [27] M. Luo, F. Nie, X. Chang, Y. Yang, A.G. Hauptmann, Q. Zheng, Adaptive unsupervised feature selection with structure regularization, *IEEE Trans. Neural Netw. Learn. Syst.* PP (2017) 1–13, doi:10.1109/TNNLS.2017.2650978.
- [28] X.-D. Wang, R.-C. Chen, F. Yan, Fast and robust K-means clustering via feature learning on high-dimensional data, in: *2017 IEEE 8th International Conference on Awareness Science and Technology (ICAST)*, IEEE, 2017, pp. 194–198, doi:10.1109/ICAwST.2017.8256444.
- [29] A. Strehl, J. Ghosh, Cluster ensembles – a knowledge reuse framework for combining multiple partitions, *J. Mach. Learn. Res.* 3 (2003) 583–617.
- [30] D. Cai, X. He, J. Han, Document clustering using locality preserving indexing, *IEEE Trans. Knowl. Data Eng.* 17 (2005) 1624–1637, doi:10.1109/TKDE.2005.198.
- [31] D. Wang, H. Nie, F. Huang, Unsupervised Feature Selection via Unified Trace Ratio Formulation and K-means Clustering (TRACK), Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 306–321.
- [32] L. Fei-Fei, R. Fergus, P. Perona, Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories, *Comput. Vis. Image Understanding* 106 (2007) 59–70, doi:10.1016/j.cviu.2005.09.012.
- [33] M. Lichman, UCI machine learning repository, 2013.
- [34] M. Lyons, S. Akamatsu, M. Kamachi, J. Gyoba, Coding facial expressions with Gabor wavelets, in: *Proceedings Third IEEE International Conference on Automatic Face and Gesture Recognition*, 1998, pp. 200–205, doi:10.1109/AFGR.1998.670949.
- [35] D. Cai, X. He, Y. Hu, J. Han, T. Huang, Learning a spatially smooth subspace for face recognition, in: *Proc. IEEE Conf. Computer Vision and Pattern Recognition Machine Learning (CVPR'07)*, 2007.
- [36] A. Kumar, P. Rai, H. Daume, Co-regularized multi-view spectral clustering, *Adv. Neural Inf. Process. Syst.* 24 (2011) 1413–1421.