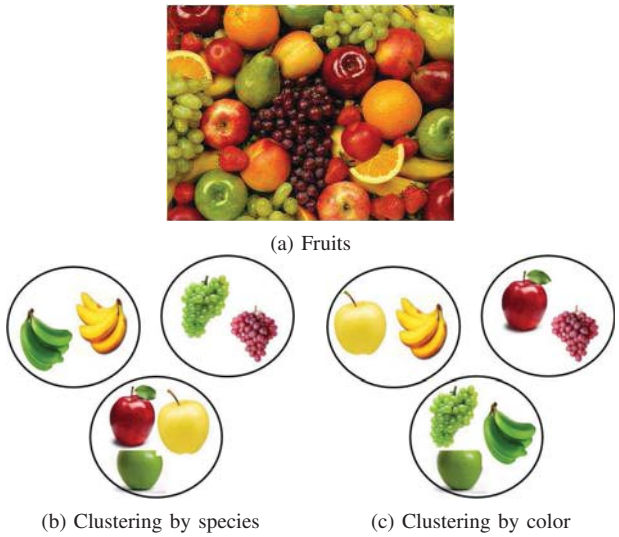
# Finding Multiple Stable Clusterings*

Juhua Hu*, Qi Qian†, Jian Pei*, Rong Jin† and Shenghuo Zhu‡
*School of Computing Science, Simon Fraser University, Burnaby, BC, V5A 1S6, Canada
†Department of Computer Science and Engineering, Michigan State University, East Lansing, MI, 48824, USA
‡Alibaba Group, Seattle, WA, 98101, USA
juhuah@sfu.ca, qianqi@cse.msu.edu, jpei@cs.sfu.ca, rongjin@cse.msu.edu, shenghuo.zhu@alibaba-inc.com

***Abstract*—Multi-clustering, which tries to find multiple independent ways to partition a data set into groups, has enjoyed many applications, such as customer relationship management, bioinformatics and healthcare informatics. This paper addresses two fundamental questions in multi-clustering: how to model the quality of clusterings and how to find multiple stable clusterings. We introduce to multi-clustering the notion of clustering stability based on Laplacian eigengap, which was originally used in the regularized spectral learning method for similarity matrix learning. We mathematically prove that the larger the eigengap, the more stable the clustering. Consequently, we propose a novel multi-clustering method MSC (for Multiple Stable Clustering). An advantage of our method comparing to the existing multi-clustering methods is that our method does not need any parameter about the number of alternative clusterings in the data set. Our method can heuristically estimate the number of meaningful clusterings in a data set, which is infeasible in the existing multi-clustering methods. We report an empirical study that clearly demonstrates the effectiveness of our method.**

***Keywords*—multi-clustering, clustering stability, feature subspace**



(a) Fruits

(b) Clustering by species (c) Clustering by color

Fig. 1. Fruits can be clustered in different ways (Images are from Internet).

## I. Introduction

Clustering, also known as unsupervised learning, is the real process of discovery and exploration by investigating the inherent and, more often than not, hidden structures with data. For example, a major part of the development of biology was to cluster species into categories and identify features that define the categorization.

Many clustering methods were proposed in literature. Most of the existing methods [1] focus on finding one way to partition data into groups. However, in many situations, different orthogonal ways may exist to partition a given data set, each way presents an unique aspect to understand the structure of the data. For example, fruits can be clustered by species or by color (Figure 1). They can even be clustered by nutrition components and in some other ways. In customer relationship management, customers can be clustered by gender, region, job, age, religion, purchase behavior, credit history and many other ways. As another example, mining phenotypes [2] is very useful in bioinformatics and healthcare informations, and is essentially a multi-clustering problem.

* Hu and Pei's research is supported in part by an NSERC Discovery grant, the Canada Research Chair program, and a Yahoo! Faculty Research and Engagement Program (FREP) award. Qian and Jin's research is supported in part by NSF (IIS-1251031) and ONR (N000141410631). All opinions, findings, conclusions and recommendations in this paper are those of the authors and do not necessarily reflect the views of the funding agencies.

Multi-clustering and alternative clustering methods try to find more than one way to partition a given data set, where a specific way of partitioning the data is called a *clustering*. Specifically, alternative clustering [3], [4], [5] attempts to find one or multiple alternative clusterings with respect to a given clustering. For example, COALA [6] aims to find an alternative clustering of high quality and high dissimilarity comparing to the given clustering. The major idea is to add instance level constraints, such as linked pairs in the given clustering being transformed to cannot-link constraints.

Alternative clustering methods highly rely on a given clustering as the input. Consequently, the alternative clustering result may not capture a user's interest exactly. To handle this problem, some researchers as Caruana *et al*. [7] proposed to generate many alternatives. However, it is computationally expensive to generate many alternative clusterings. Moreover, it is hard to constrain the differences between the clusterings computed, and overwhelming for users to absorb and understand the results.

To interpret multiple possible clusterings, subspace multi-clustering approaches produce multiple clusterings by considering different subsets of attributes that may represent different perspectives about the objects, such as species or color in the motivation example in Figure 1. For example, CLIQUE [8] divides a multidimensional data space into grid-cells, each dimension being partitioned into equal width intervals. Then,

dense cells in each subspace are identified using a density threshold. A group of connected dense cells in a subspace is regarded as a cluster. A clustering can be produced accordingly within a subspace. Obviously, CLIQUE has to search an exponential number of subspaces with respect to the number of attributes. Although some fast heuristic variants, such as INSCY [9], were proposed, the scalability remains a challenge. Another drawback is that the subspace multi-clustering approaches cannot explicitly consider the dissimilarity between different clusterings. Such methods tend to produce many clusterings in order to cover some interesting ones, which may likely overwhelm users.

In this paper, we challenge ourselves two fundamental questions: how can we model the quality of clusterings and how can we find multiple stable clusterings in a given data set? We make a few technical contributions. First, we borrow the idea of clustering stability based on Laplacian eigengap, originally introduced by Meilă and Shortreed in the regularized spectral learning method for similarity matrix learning [10], and apply to multi-clustering. The intuition is that a clustering is stable if small distortions on the attribute values do not affect the discoverability of the clustering. Mathematically, we show that the larger the eigengap, the more stable the clustering. Second, based on the notion of stability of clusterings and the underlying analysis on the Laplacian eigengap, we propose a novel multi-clustering method, named MSC (for Multiple Stable Clustering) to obtain a certain number of stable clusterings. We model the problem of finding a stable clustering as an optimization problem maximizing the eigengap. The problem is unfortunately non-convex, and thus we propose a heuristic randomized method using iterative gradient ascent. In order to find multiple stable clusterings, we introduce to the optimization problem a constraint on the difference from the clusterings found so far. An advantage of our method comparing to the existing multi-clustering methods is that our method does not need any parameter about the number of alternative clusterings in the data set. Our method can heuristically estimate the number of meaningful clusterings in a data set, which is infeasible in the existing multi-clustering methods. We also discuss techniques to speed up the search process. Last, we report an empirical study on synthetic and real data sets that clearly demonstrates the effectiveness of our method.

The rest of the paper is organized as follows. Section II reviews the related work briefly. Section III models the stability of clusterings and presents an algorithm to find a stable clustering. In Section IV, we develop MSC, an algorithm to find multiple stable clusterings. Section V reports the results of an empirical study. Section VI concludes this work and discusses the future research directions.

## II. Related Work

In this section, we briefly review the existing work related to our study.

### A. Multi-clustering

*Alternative clustering* is a major type of multi-clustering methods in literature. Given a clustering, an alternative clustering method tries to find clusterings that are different from the input clustering. Bailey [4] provided a thorough survey on alternative clustering methods. Meta clustering [7] is a typical unguided alternative clustering method, which first generates many different clusterings using different clustering algorithms or different settings, and then clusters the clusterings according to their similarities. In some application scenarios, users may want to obtain a clustering as different as possible from the input one. To address this demand, guided alternative clustering methods were developed. For example, MAXIMUS [11] utilizes a programming model to calculate the maximum dissimilarity between a new possible clustering and all clusterings obtained. Recently, Dang and Bailey [5] used regularized PCA to find an alternative subspace that is highly independent from the input clustering.

In multidimensional data sets, different subsets of attributes, that is, different subspaces, may represent different perspectives of the data, some researchers proposed to find different clustering solutions by using different feature subspaces, which can be referred as subspace multi-clustering. CLIQUE [8] is the first subspace multi-clustering approach, which is a grid cell based method and aims to explore all potential subspaces and find the dense clusters. Since the clustering results by CLIQUE are highly sensitive to the grid positions, algorithms MAFIA [12] and SCHISM [13] were developed to enhance the grid cell based method. Their subspace search strategy is later adopted by SUBCLU [14], which is an extension of DBSCAN [15]. Some efficient variants [16], [9], [17] were developed as well.

In summary, the existing multi-clustering methods mainly focus on finding multiple good but different clusterings. At the same time, those methods may not be able to capture a user's interest, and may overwhelm a user when they generate too many clusterings.

### B. Regularized Spectral Clustering

Spectral clustering [18] clusters data represented in pairwise similarities between data points. Spectral clustering has been studied extensively. Please see [18] for a thorough treatment. Here, we only review the regularized spectral learning method [10], where Meilă and Shortreed tackled the problem of automatically learning similarities between data points for spectral clustering. They proved that a large Laplacian eigengap corresponds to clustering stability. Thus, using the eigengap as a regularizer is natural for learning problems where some ground truth labels are available. In this paper, we adopt their clustering stability notion, and apply it to the multi-clustering problem.

## III. Stable Clusterings

In this section, we model the stability of a clustering and discuss how to find a stable clustering. Let us start with some preliminaries and the intuition.

### A. Preliminaries and Ideas

In this paper, we consider a data set $X \in \mathbb{R}^{d \times n}$, that is, $X$ contains $n$ instances, each of $d$ features. We do not assume any knowledge about how the instances in $X$ are partitioned into groups. We assume that a user wants $k$ clusters from $X$, where $k$ is a parameter.

A *clustering* $c = \{X_1, X_2, \ldots, X_k\}$ is a partitioning of the instances in $X$, such that $\cup_{m=1}^{k} X_i \subseteq X$ and $X_i \cap X_j = \emptyset$ for $1 \leq i < j \leq k$. Each $X_m$ $(1 \leq m \leq k)$ is called a *cluster*. In clustering analysis, we are interested in clusterings where objects in a cluster are similar and objects in different clusters are dissimilar. Here, similarity can be defined in many different ways and thus lead to various clustering methods.

In multi-clustering, we want to find multiple clusterings that are independent and thus different from each other. How to measure the independencies or differences among clusterings is a key in multi-clustering design. In this paper, we explore stabilities of clusterings as the measure. While we will develop the technical details later in this section, the intuition is as follows. We want to find stable clusterings. A clustering is stable if any small distortions on the attribute values will not affect the quality of the clustering. Unstable clusterings are not desirable in multi-clustering since their quality is sensitive to noise and thus may be caused by overfitting the current observed data.

Now, the technical questions are how we can model the stability of a clustering, and how to find a stable clustering.

### B. Stability of a Clustering

We thoroughly consider clusterings in different subspaces. In general, we consider all possible weighting designs of the features. We use a simplex $\Delta^d$ to denote all possible feature subspaces, which can be represented as a set of points

$$\Delta^d = \{w_1\mathbf{q}_1 + w_2\mathbf{q}_2 \cdots + w_d\mathbf{q}_d | w_m \geq 0, \sum_{m=1}^{d} w_m = 1\},$$

where $\mathbf{q}_m$ is a unit vector corresponding to the $m$-th feature, that is,

$$\begin{aligned}\mathbf{q}_1 &= (1, 0, 0, \cdots, 0)\\ \mathbf{q}_2 &= (0, 1, 0, \cdots, 0)\\ &\cdots\\ \mathbf{q}_d &= (0, 0, 0, \cdots, 1)\end{aligned}$$

and $w_m$ is the weight assigned to the $m$-th feature. Denote by $\mathbf{w} = (w_1, w_2, \cdots, w_d)$ the feature weight vector. When all weights are set to $1/d$, it is the conventional full feature space.

For each data point $\mathbf{x}_i$, we can obtain the weighted vector representation by multiplying each weight $w_m$ with the $m$-th feature as $\mathbf{x}'_i = \mathbf{w} \odot \mathbf{x}_i$, where $\odot$ is the operation multiplying each element of $\mathbf{w}$ with the corresponding feature in $\mathbf{x}_i$. Then, the similarity between two instances $\mathbf{x}_i$ and $\mathbf{x}_j$ can be written as

$$S_{i,j} = e^{-\|\mathbf{x}'_i - \mathbf{x}'_j\|_2^2} \tag{1}$$

Now we apply spectral clustering to the similarity matrix $S$. Note that, although we discuss only spectral clustering here, any clustering method, such as $k$-means, can be applied to the similarity matrix. We calculate the *normalized Laplacian* by

$$L = D^{-1/2} S D^{-1/2} \tag{2}$$

where $D$ is a diagonal matrix formed by

$$D_i = \sum_{j=1}^{n} S_{i,j}, \qquad i = 1, 2, \ldots, n$$

It is easy to verify that the eigenvectors of the normalized Laplacian here having the largest eigenvalues are identical to the eigenvectors of $I - D^{-1/2} S D^{-1/2}$ having the smallest eigenvalues, as stated in [19].

In spectral clustering, we conduct eigen-decomposition for the Laplacian matrix $L$ and conduct clustering based on the top $k$ eigenvectors. Now we show the main theoretical result of this paper, which indicates a nice property of stable clusterings. Denote by $\lambda_k(L)$ the $k$-th largest eigenvalue of $L$. Essentially, we show that, if the eigengap $\lambda_k(L) - \lambda_{k+1}(L)$ is sufficiently large, a small perturbation on the similarity matrix $S$ or on the weight vector $\mathbf{w}$ will not affect the top $k$ eigenvectors, and thus the clusterings obtained upon them are stable.

*Theorem 1 (Stability):* Given a Laplacian matrix $L$, if the eigengap $\lambda_k(L) - \lambda_{k+1}(L)$ is large enough, the top $k$ eigenvectors of $L_{perb} = L + \epsilon$ are the same as those of $L$, where $\epsilon$ is a symmetric perturbation matrix of small spectral norm $\|\epsilon\|_2$.

*Proof:* We prove that, for any $\epsilon$ such that $\|\epsilon\|_2 < \frac{\lambda_k(L) - \lambda_{k+1}(L)}{2}$, the top $k$ eigenvectors of $L_{perb} = L + \epsilon$ are the same as those of $L$.

Since $L$ is positive semi-definite, $L$ has $n$ non-negative real-valued eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n = 0$ with the corresponding eigenvectors $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n$. Denote by $\beta = \lambda_k(L) - \lambda_{k+1}(L)$. $\epsilon$ must be in one of the following two cases.

Case 1. $\epsilon$ is from the space spanned by the eigenvectors of $L$. Then, $L_{perb} = L + \epsilon$ can be represented as $\sum_{i=1}^{n}(\lambda_i \mathbf{v}_i \mathbf{v}_i^\top + \lambda_i^\epsilon \mathbf{v}_i \mathbf{v}_i^\top)$, where each $\mathbf{v}_i$ is an eigenvector of $L$ and $\epsilon = \sum_{i=1}^{n} \lambda_i^\epsilon \mathbf{v}_i \mathbf{v}_i^\top$. In this case, $\epsilon$ only affects the eigenvalues of $L_{perb}$ comparing to those of $L$ by a factor $\lambda_i^\epsilon$, and the eigenvectors remain the same. Furthermore, since $\|\epsilon\|_2 < \beta/2$, $\max_i |\lambda_i^\epsilon| < \beta/2$. Therefore,

$$\begin{aligned}\lambda_k + \lambda_k^\epsilon - (\lambda_{k+1} + \lambda_{k+1}^\epsilon) &> \lambda_k - \beta/2 - (\lambda_{k+1} + \beta/2)\\ &= \lambda_k - \lambda_{k+1} - \beta = 0\end{aligned}$$

Thus, the top $k$ eigenvectors of $L_{perb}$ remain the same, though the ordering of the top $k$ eigenvectors may vary.

Case 2. $\epsilon$ is not solely from the space spanned by the eigenvectors of $L$. Then, $L_{perb} = L + \epsilon$ can be decomposed into two parts, one from the space spanned by the eigenvectors of $L$ and the other from the orthogonal space. That is, $L + \epsilon = \sum_{i=1}^{n}(\lambda_i \mathbf{v}_i \mathbf{v}_i^\top + \lambda_i^\epsilon \mathbf{v}_i \mathbf{v}_i^\top + \lambda_i^\perp \mathbf{u}_i \mathbf{u}_i^\top)$, where $\sum_{i=1}^{n} \lambda_i^\perp \mathbf{u}_i \mathbf{u}_i^\top$ is part of $\epsilon$ from the orthogonal space. The first part is similar to the first case. For the second part, since $\|\epsilon\|_2 < \beta/2$, $\max_i \lambda_i^\perp <= \max_i |\lambda_i^\perp| < \beta/2$. Therefore,

$$\lambda_k = \lambda_{k+1} + \beta \geq \beta > \max_i \lambda_i^\perp$$

Thus, the top $k$ eigenvectors of $L_{perb}$ remain the same as $L$, though the ordering of the top $k$ eigenvectors may vary. ■

*Remark 1:* In spectral clustering, if the top $k$ eigenvectors are the same for $L$ and $L_{perb}$, the clusterings based on the same eigenvectors are the same.

According to Theorem 1 and Remark 1, the larger the eigengap between the $k$-th and the $(k+1)$-th eigenvalues of $L$, the more stable the clustering obtained upon $L$.

**Algorithm 1:** Finding a Stable State (FSS)

**Input:** Dataset $X \in \mathbb{R}^{d\times n}$, the number of clusters $k$, the number of iterations $T$, and step size $\eta$
Randomly initialize weight vector $\mathbf{w}_0$ in the simplex
**for** $t = 1$ **to** $T$ **do**
  Compute $S$ by Eq. 1
  Calculate the normalized Laplacian $L$ by Eq. 2
  Conduct eigen-decomposition for $L$
  Compute $\mathbf{G}$ by Eq. 4
  $\mathbf{w}_t = \mathbf{w}_{t-1} + \eta\mathbf{G}$
  Project $\mathbf{w}_t$ onto the simplex by $\mathcal{P}_{1^+}$
**end for**
**return** $\mathbf{w}_T$

### C. Finding a Stable Clustering

According to Theorem 1, the most stable clustering can be obtained by maximizing the eigengap, that is,

$$\arg\max_{\mathbf{w}\in\Delta^d} \lambda_k(L) - \lambda_{k+1}(L) \tag{3}$$

where the *most stable weight vector* $\mathbf{w}$ is searched in the simplex $\Delta^d$. Note that the simplex constraint has a good sparse property – it automatically eliminates those features of too low weights. This property is desirable since sparse feature selection has been demonstrated effective by many previous studies [20], [21], [22], [23].

Although Eq. 3 models the most stable clustering nicely, apparently the optimization problem in Eq. 3 is non-convex, and thus is hard to find an exact solution.

As a heuristic solution, we can randomly initialize $\mathbf{w}$ as $\mathbf{w}_0$ in the simplex $\Delta^d$ and solve the optimization problem by iterative gradient ascent. At the $t$-th iteration ($t \geq 1$), we set

$$\mathbf{w}_t = \mathbf{w}_{t-1} + \eta\mathbf{G}$$

where the $m$-th element of vector $\mathbf{G}$ ($1 \leq m \leq d$) is

$$G_m = \langle \mathbf{v}_k\mathbf{v}_k^\top, \frac{\partial L}{\partial w_m}\rangle - \langle \mathbf{v}_{k+1}\mathbf{v}_{k+1}^\top, \frac{\partial L}{\partial w_m}\rangle \tag{4}$$

and $\mathbf{v}_k$ is the $k$-th eigenvector. To constrain $\mathbf{w}$ within the simplex $\Delta^d$, in each gradient ascent step, we adopt the projection algorithm proposed by Kyrillidis *et al.* [24]. Concretely, we project $\mathbf{w}_t$ obtained in the $t$-th step onto the simplex by

$$(\mathcal{P}_{1^+}(\mathbf{w}_t))_m = [(\mathbf{w}_t)_m - \alpha]_+,$$

where $\alpha$ is a threshold that is set to $\alpha = \frac{1}{\rho}(\sum_{i=1}^{\rho}(\mathbf{w}_t)_m - 1)$, $\rho = \max\{m : (\mathbf{w}_t)_m > \frac{1}{m}(\sum_{i=1}^{m}(\mathbf{w}_t)_m - 1)\}$, and $\alpha$ and $\rho$ are calculated by first sorting the elements in $\mathbf{w}_t$ in descending order. Instead of directly calculating the gradient under the simplex constraint, we first calculate the gradient without the constraint, conduct the gradient ascent, and then project $\mathbf{w}$ back to the simplex. Kyrillidis *et al.* [24] proved that the projection algorithm still obtains the same $\mathbf{w}$ as considering the simplex constraint in the gradient calculation.

Algorithm 1 shows the pseudo-code of FSS (for Finding a Stable State), which finds a local optima for the Problem in Eq. 3.

**Algorithm 2:** Finding Alternative Stable State (FASS)

**Input:** Dataset $X \in \mathbb{R}^{d\times n}$, the number of clusters $k$, the number of iterations $T$, step size $\eta$, the set of previously found stable states $W$, and the tradeoff parameter $\delta$
Randomly initialize weight vector $\mathbf{w}_0$ in the simplex
**for** $t = 1$ **to** $T$ **do**
  Compute $S$ by Eq. 1
  Calculate the normalized Laplacian $L$ by Eq. 2
  Conduct eigen-decomposition for $L$
  Compute $\mathbf{G}$ by Eq. 4
  $\mathbf{w}_t = \mathbf{w}_{t-1} + \eta(\mathbf{G} + \delta\frac{1}{|W|}\sum_{\mathbf{w}_p\in W}(\mathbf{w}_{t-1} - \mathbf{w}_p))$
  Project $\mathbf{w}_t$ onto the simplex by $\mathcal{P}_{1^+}$
**end for**
**return** $\mathbf{w}_T$

## IV. MSC: Finding Multiple Stable Clusterings

In this section, we develop MSC, an algorithm that finds multiple stable clusterings. We first present the framework, and then discuss the techniques to speed up the algorithm.

### A. Finding Multiple Stable States

To find various stable clusterings, we need to search all stable states in the simplex. Although random initialization can give a good start point, two random initialization values may converge to the same local optimal clustering. To overcome this problem, we introduce a constraint on the difference between the current weight vector and those previously obtained. Let $W$ be the set of weight vectors obtained so far. The optimization problem can be rewritten as

$$\arg\max_{\mathbf{w}\in\Delta^d} \lambda_k(L) - \lambda_{k+1}(L) + \frac{\delta}{2}\frac{1}{|W|}\sum_{\mathbf{w}_p\in W}\|\mathbf{w} - \mathbf{w}_p\|_2^2 \tag{5}$$

In other words, we want to maximize the sum of the distances between the current weight vector and those weight vectors obtained previously, so as to keep the current clustering far away from all previous ones. Here, $\delta \geq 0$ is a tradeoff parameter for balancing the maximization of eigengap and the dissimilarity.

Algorithm 2 describes how to find a new stable state (weight vector), given the set $W$ of stable states found so far, by solving the optimization problem in Eq. 5 using gradient ascent. The regularization term incurs only very light computational cost.

We can run Algorithm 1 to find the first stable state, and then run Algorithm 2 repeatedly with different initialization values to find more stable states. Heuristically, if Algorithm 2 does not lead to any new stable state in the last $l$ runs, then we can terminate the process, where $l > 0$ is a parameter.

After gathering a set of stable weight vectors, we can compute the similarity matrix $S$ for each stable weight vector $\mathbf{w}$ and apply spectral clustering to obtain the corresponding stable clusterings. Each stable clustering obtained as such has a corresponding sparse feature subspace $\mathbf{w}$ for user understanding. The whole algorithm is summarized in Algorithm 3.

**Algorithm 3:** Multiple Stable Clustering (MSC)

1: **Input:** Dataset $X \in \mathbb{R}^{d\times n}$, the number of clusters $k$, the number of iterations $T$, step size $\eta$, the tradeoff parameter $\delta$, and the stopping threshold $\tau$
2: Initialize the stable state set $W = \emptyset$, and the clustering solution set $C = \emptyset$
3: Obtain the 1st stable state $\mathbf{w}$ using Alg. 1
4: $W = W \cup \mathbf{w}$
5: **repeat**
6: Obtain a new stable state $\mathbf{w}$ using Alg. 2
7: $W = W \cup \mathbf{w}$
8: **until** $\min_{\mathbf{w}_i,\mathbf{w}_j\in W, i\neq j} \|\mathbf{w}_i - \mathbf{w}_j\|_2^2 \leq \tau$
9: Delete the $\mathbf{w} \in W$ that is last obtained
10: **for each** $\mathbf{w} \in W$ **do**
11: Compute $S$ by Eq. 1
12: Calculate the normalized Laplacian $L$ by Eq. 2
13: Conduct eigen-decomposition for $L$
14: Get the top $k$ eigenvectors
15: Obtain the clustering $c$ using $k$-means on the eigenvectors
16: $C = C \cup \{(\mathbf{w}, c)\}$
17: **end for**
18: **return** $C$

An advantage of our method comparing to the existing multi-clustering methods is that our method does not need any parameter about the number of alternative clusterings in the data set. The optimization on both the eigengap and the weight vector dissimilarity leads to stable clusterings discovered in the iterations. The mining method terminates when no new stable and substantially different clustering can be found. As demonstrated in the experimental results in Section V, our method can heuristically estimate the number of meaningful clusterings in a data set, which is infeasible in the existing multi-clustering methods.

### B. Speedup

To find a stable state vector $\mathbf{w}$, we conduct gradient ascent for $T$ iterations. In each iteration, we have to compute the full similarity matrix $S$ based on the weight vector obtained in the previous iterations. Thus, the time and space complexities are $\mathcal{O}(n^2)$. We also have to conduct eigen-decomposition for the Laplacian matrix $L$ to obtain the largest $k+1$ eigenvalues and their corresponding eigenvectors, whose computational cost is $\mathcal{O}(kn^2)$. All other computation steps are cheaper comparing to these two. This heavy computational cost limits the application of MSC to large-scale data sets that have many instances.

The Nyström method [25], [26] has been commonly used to approximate kernel matrices. We can use it to approximate our similarity matrix $S \in \mathbb{R}^{n\times n}$. We randomly sample $m \ll n$ columns from matrix $S$ and obtains a symmetric sub-matrix $M \in R^{m\times m}$. Without loss of generality, we can take the first $m$ columns of $S$ as the sample. Let $A$ and $B$ be two sub-matrices of $S$ such that $A$, $B$ and $M$ do not overlap with one another, and $A$, $B$ and $M$ together cover every element in $S$. Then, $S$ can be rewritten as

$$S = \begin{bmatrix} M & B^\top \\ B & A \end{bmatrix}$$

Let $C \in \mathbb{R}^{n\times m}$ be the sub-matrix containing the selected columns, that is,

$$C = \begin{bmatrix} M \\ B \end{bmatrix}$$

The Nyström approximation of $S$ is given by

$$\tilde{S} = CM^{-1}C^\top$$

Using the Nyström approximation, the cost for calculating the similarity matrix $S$ is reduced to $\mathcal{O}(nm)$.

Since only the top $k+1$ eigenvalues and their corresponding eigenvectors are needed in MSC, our goal is to find a rank-$(k+1)$ approximation $S_{k+1}$ of $S$.

Let $S_{k+1} = U\Lambda U^\top$, where $\Lambda$ contains the largest $k+1$ eigenvalues of $S$, and $U$ contains their eigenvectors. Following the recent work [27], we define an analogous matrix $S_* \in \mathbb{R}^{m\times m}$ as

$$S_* = D_*^{-\frac{1}{2}} M D_*^{-\frac{1}{2}}$$

where $D_* = diag(M\mathbf{1})$.

Since $S_*$ is a $m \times m$ matrix, it can be decomposed efficiently to get its top $k+1$ eigenvalues $\Lambda$ that are also the top $k+1$ eigenvalues of $S_{k+1}$ and their corresponding eigenvectors, denoted by set $V$. Thereafter, the top $k+1$ eigenvectors of $S_{k+1}$ can be obtained by

$$U = D^{\frac{1}{2}} Q$$

where $Q = CD_*^{\frac{1}{2}} V\Lambda^{-1}$ and $D = diag(Q\Lambda Q^\top \mathbf{1})$.

By replacing Steps 11 to 14 in Algorithm 3 as well as the similar steps in Algorithms 1 and 2 accordingly by this speedup method, for each iteration of our gradient ascent method, we can reduce the computational cost from $\mathcal{O}(n^2)$ to $\mathcal{O}(n)$. For the space complexity, if storing $C$ in memory, it can be reduced from $\mathcal{O}(n^2)$ to $\mathcal{O}(nm)$. Since $C$ is only accessed once, it can be computed row by row, and thus the space complexity can be further reduced to $\mathcal{O}(nk)$. This makes our MSC method applicable on large-scale data.

## V. Empirical Study

To test our proposed MSC method, we conducted an empirical study on both synthetic and real data sets. We report the results in this section.

### A. Settings

To test whether the MSC method can find at least one clustering that is consistent with the ground truth labels, we set the parameter $k$ as the number of clusters given by the ground truth. The number of iterations $T$ was set to 30 for all the experiments, which is sufficient to allow the gradient ascent method to converge, as observed in our experimental results (Figs. 2, 6 and 7). All other parameters were tuned by the holdout for each dataset. Specifically, we randomly sampled 25% of the data points in the data set following the same distribution of the clusters in the ground truth for holdout. Empirically we found that, after normalization so that each data instance has norm 1, setting $\eta = 1$ and $\delta = 0.001$ achieved the best performance, which will be used as default values in this section.

TABLE I. STATISTICS OF THE DATA SETS.

| Data | # Instances | # Features | # Clusters |
|---|---|---|---|
| *synthetic data* | 50 | 3 | 2 |
| *fruit* | 105 | 6 | 3 |
| *balance* | 625 | 4 | 3 |
| *iris* | 150 | 4 | 3 |
| *wine* | 178 | 13 | 3 |

To the best of our knowledge, the existing multi-clustering methods either need some parameters to specify the number of clusterings to be found or do not guarantee the dissimilarity or quality of clusterings found. We cannot find an existing method that does not need a parameter about the number of clusterings and, at the same time, can guarantee the quality (stability in our method) of and dissimilarity among clusterings found. Therefore, in the rest of this section, we focus on testing our method against finding high quality clusterings, and do not compare with the existing multi-clustering methods. A thorough comparison will be left as future work.

Specifically, we compare the clusterings produced by MSC with the results from two well known clustering methods: $k$-means [28] and normalized spectral clustering [18], denoted by "$k$-means" and "Spectral" hereafter, respectively. When those two methods were applied, we weighted each feature the same, because those two methods cannot find the weights of features automatically.

We report the matching quality between two clusterings using five popularly used measures: Normalized Mutual Information (NMI), Rand Index (RI), Adjusted Rand Index (AR), Mirkin's Index Distance (MI), and Hubert's Index (HI) [29]. The smaller MI, the more similar the two clusterings under comparison. For the other four measures, the larger the values, the more similar the two clusterings.

The information about the data sets used in the experiments is summarized in Table I.

### B. A Case Study Using Synthetic Data

We use a synthetic data set to demonstrate that MSC can in practice determine the number of stable clusterings and discover the clusterings.

We randomly generated 50 data points with 3 binary features. That is, the synthetic data $X \in \{0,1\}^{50\times 3}$. We set the number of clusters $k = 2$. Due to the binary features, each feature solely can group the data points into 2 clusters, where the feature itself is a subspace. Therefore, we have 3 clusterings in the ground truth denoted by "Clustering 1", "Clustering 2", and "Clustering 3", respectively. For this synthetic data, we first normalize the data so that each instance has norm 1, and set the stopping threshold as $\tau = 0.001$ in Algorithm 3. MSC output 3 stable clusterings as shown in Table II.

The clustering produced by $k$-means matches Clustering 1 in the ground truth, manifested by the first feature, though all attributes are equivalently weighted in $k$-means. However, $k$-means can only produce a single partitioning, and thus cannot produce the other two clusterings in the ground truth, which is also the case for the normalized spectral clustering method. The spectral clustering method output a clustering solution relatively consistent with but not perfectly matching Clustering 2 in the ground truth, manifested by the second feature.

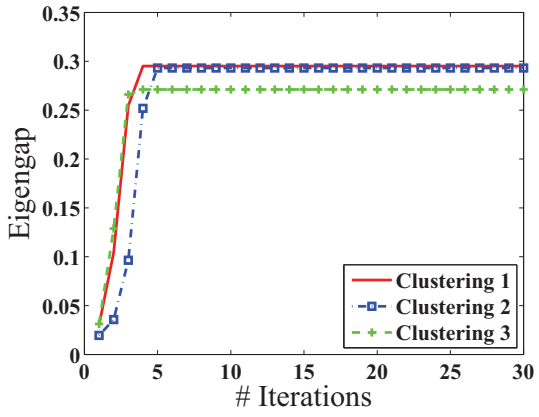



Fig. 2. The changes of eigengap in iterations on the synthetic data.

As shown in Table II, MSC found the three clusterings perfectly in the ground truth. Moreover, MSC also found the three stable states (weight vectors) exactly as expected (each feature provides a perfect feature subspace for a 2-way clustering), as shown in the last column of Table II. These weight vectors obtained confirm the sparse property of the simplex constraint. MSC output the three clusterings in the order of Clustering 1, Clustering 2 and Clustering 3. The order was the decreasing order of eigengap. The eigengaps for those three stable states are much larger than the eigengap in the clustering produced by the spectral clustering method, which demonstrates that by maximizing the eigengap, MSC can find stable clusterings.

Fig. 2 shows how the eigengap of each clustering solution converges as the number of iterations increases. It can be easily observed that our gradient ascent method converges very quickly on the synthetic data, which also demonstrates the effectiveness of our method.

### C. A Case Study on an Image Data Set

We conducted a case study on images of three types of fruits, namely apples, bananas and grapes, in different colors, namely red, yellow and green for apples, yellow and green for bananas, and red and green for grapes. There are two different clusterings in the ground truth: the clustering by fruit category, denoted by Clustering-by-Category, and the clustering by color, denoted by Clustering-by-Color.

*1) Data Preparation:* We collected from Internet 15 images for each sub-group, i.e., red apples, yellow apples, green apples, yellow bananas, etc. In total, there were $15 \times 7 = 105$ images, 45 about apples, 30 about bananas, and another 30 about grapes. Orthogonally, there were 30 images about red fruits, 30 about yellow fruits, and 45 about green fruits.

For each image, we firstly partitioned it into blocks of 4 pixels by 4 pixels, and then extracted the color and texture features for each block. For the color features, we transformed the RGB images into the HSI color space, which is appropriate for object recognition as suggested by Shapiro and Stockman [30]. Thereafter, the color feature of each block

TABLE II. CLUSTERINGS ON THE SYNTHETIC DATA. (The best cases are highlighted in bold. The AR values are not provided here because the denominators in the AR calculation are all zeros here.)

| Clustering produced by methods | Clustering in ground truth | NMI | RI | AR | MI | HI | Eigengap | Weight vector $\mathbf{w}$ |
|---|---|---|---|---|---|---|---|---|
| $k$-means | Clustering 1 | **1.000** | **1.000** | - | **.0000** | **1.000** | - | (.3333;.3333;.3333) |
| | Clustering 2 | .0043 | .4783 | - | .5217 | -.043 | | |
| | Clustering 3 | .0039 | .4737 | - | .5263 | -.053 | | |
| Spectral | Clustering 1 | .0283 | .5569 | - | .4431 | .1138 | .0039 | (.3333;.3333;.3333) |
| | Clustering 2 | **.7263** | **.7628** | - | **.2372** | **.5257** | | |
| | Clustering 3 | .0718 | .6491 | - | .3509 | .2983 | | |
| Clustering 1 | Clustering 1 | **1.000** | **1.000** | - | **.0000** | **1.000** | .2951 | (1.000;.0000;.0000) |
| | Clustering 2 | .0043 | .4923 | - | .5077 | -.015 | | |
| | Clustering 3 | .0039 | .5292 | - | .4708 | .0585 | | |
| Clustering 2 | Clustering 1 | .0043 | .4783 | - | .5217 | -.044 | .2931 | (.0000;1.000;.0000) |
| | Clustering 2 | **1.000** | **1.000** | - | **.0000** | **1.000** | | |
| | Clustering 3 | .0154 | .5573 | - | .4427 | .1146 | | |
| Clustering 3 | Clustering 1 | .0039 | .4737 | - | .5263 | -.053 | .2711 | (.0000;.0000;1.000) |
| | Clustering 2 | .0154 | .5088 | - | .4105 | .4912 | | |
| | Clustering 3 | **1.000** | **1.000** | - | **.0000** | **1.000** | | |

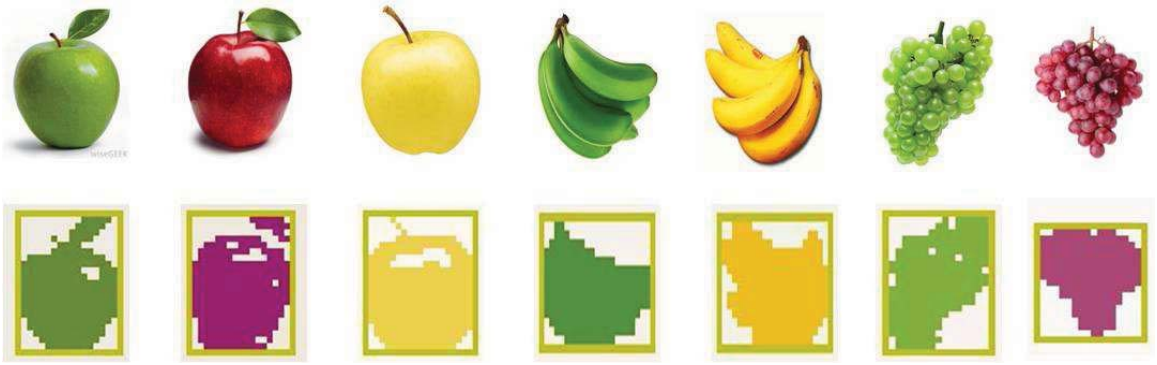


Fig. 3. Example images in the fruit data set.

TABLE III. CLUSTERINGS ON THE FRUIT IMAGE DATA SET. (The best cases are highlighted in bold.)

| Clustering produced by methods | Clusterings in ground truth/by Spectral | NMI | RI | AR | MI | HI | Weight vector $\mathbf{w}$ |
|---|---|---|---|---|---|---|---|
| $k$-means | Clustering-by-Category | .1486 | .5659 | .0684 | .4341 | .1319 | [.1667;.1667;.1667;.1667;.1667;.1667] |
| Spectral | | .1432 | .5650 | .0611 | .4350 | .1300 | [.1667;.1667;.1667;.1667;.1667;.1667] |
| Clustering 1 | | .1394 | .5857 | .0818 | .4143 | .1714 | [.2538;.0011;.0765;.0655;.1004;.5027] |
| Clustering 2 | | **.1627** | **.6045** | **.1289** | **.3954** | **.2092** | [.3222;.0000;.0000;.0000;.6778;.0000] |
| Clustering 3 | | .1449 | .5886 | .0883 | .4114 | .1773 | - |
| Clustering 4 | | .1151 | .5716 | .0465 | .4284 | .1432 | [.4012;.0000;.0000;.5988;.0000;.0000] |
| $k$-means | Clustering-by-Color | .5905 | .7626 | .4905 | .2374 | .5253 | - |
| Spectral | | .5522 | .7559 | .4730 | .2441 | .5117 | - |
| Clustering 1 | | .6160 | .7711 | .4926 | .2289 | .5241 | - |
| Clustering 2 | | .5564 | .7474 | .4436 | .2526 | .4949 | - |
| Clustering 3 | | **.6886** | **.8051** | **.5681** | **.1949** | **.6103** | [.4468;.0000;.5532;.0000;.0000;.0000] |
| Clustering 4 | | .5124 | .7291 | .3971 | .2709 | .4582 | - |
| $k$-means | Spectral | .8839 | .9581 | .9118 | .0419 | .9161 | - |

was represented by the average value of each channel. Thus, each block had three color features, corresponding to the channels of Hue, Saturation, and Intensity, respectively. For the texture features, we employed the one-level two-dimensional Daubechies-4 wavelet transformation [31] to decompose each block to four frequency bands. Each band had four parameters $\{c_{k,l}, c_{k,l+1}, c_{k+1,l}, c_{k+1,l+1}\}$. We then calculated the features by

$$f = (\frac{1}{4}\sum_{i=0}^{1}\sum_{j=0}^{1} c_{k+i,l+j}^2)^{\frac{1}{2}}$$

with the bands about activities in horizontal, vertical and diagonal directions [32], which also led to three features for a block.

For an image with multiple six dimensional blocks, we applied $k$-means for segmentation as recommended in [32]. After segmentation, we then used raster scanning [30] to get all connected components. A region whose number of blocks is below a threshold was set to background. We erased background regions. Then, each fruit region was represented by the average color features and the normalized inertia as the shape features [32]

$$l(H,\gamma) = \frac{\sum_{x} \in H \|x - \hat{x}\|^{\gamma}}{V(H)^{1+\gamma/k}}$$

where $H$ is the fruit region, $V(H)$ is the number of blocks in the region and $\hat{x}$ is the center of the region. We respectively set $\gamma = 1, 2, 3$ to calculate the shape features, and thus the total number of features for each fruit region was six. Fig. 3 shows some example images. The first row is the original images collected from Internet sources, and the second row is the recognized fruit regions bounded and represented by the average RGB colors.

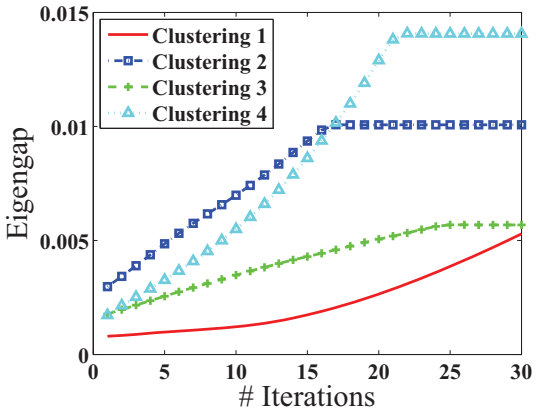


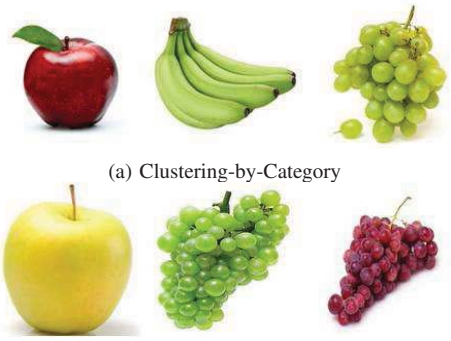

Fig. 4. The changes of eigengap in iterations on the fruit image data set.

*2) Results:* Table III shows the results of $k$-means, spectral clustering and MSC on this image dataset. The parameters for MSC were set to $T = 30$, $\eta = 1$, $\delta = 0.01$, and $\tau = 0.1$. Interestingly, the clustering results produced by $k$-means and spectral clustering were almost identical, which were more consistent with the ground truth clustering by color than that by category.

MSC produced 4 stable clusterings on this data set. Each clustering emphasized on different feature subspaces. For instance, Clustering 3 produced by MSC corresponded to a feature subspace that only considers the color features. Consequently, Clustering 3 was very consistent with Clustering-by-Color in the ground truth, and was also more consistent than the clusterings generated by $k$-means and spectral clustering with respect to Clustering-by-Color. Simultaneously, Clustering 2 focused on both the color and shape features and put more weight on the shape feature. Clustering 2 was substantially close to Clustering-by-Category than those produced by $k$-means and Spectral. A clustering using both the shape and color features is highly reasonable. To distinguish between apples, bananas and grapes, only the shape features may not be enough. For example, a bunch of bananas may happen to have a shape similar to that of a bunch of grapes. Thus, we need to get help from the color attributes. For example, this data set does not have red bananas.

In addition to those two clusterings, MSC also produced two other stable clusterings, Clustering 1 and Clustering 4 in Table III. Those two stable clusterings weighted on totally different subspaces. The 4 stable clusterings produced by MSC had low redundancy – all pairwise NMIs between them were smaller than 0.55. Moreover, MSC provided different subspaces for the stable clusterings, which were informative for user understanding. Interestingly, two different subspaces led to two somehow similar clusterings on the data, e.g., Clustering 2 and Clustering 4 in Table III. An important application of different weight vectors leading to similar clusterings is multi-view clustering [33], [34], which aims to combine results from different views to generate a consensus clustering.

Fig. 4 shows how the eigengap of each clustering solution converges as the number of iterations increases. Figs. 5(a) and 5(b) show the images that are closest to the cluster centers of Clustering 2 and Clustering 3, respectively. The



(a) Clustering-by-Category

(b) Clustering-by-Color

Fig. 5. The images closest to the cluster centers.

TABLE V. CLUSTERING REDUNDANCY. Clustering 1 and Clustering 2 were produced by MSC.

| Clusterings compared | NMI | RI | AR | MI | HI |
|---|---|---|---|---|---|
| **Data set** *balance* | | | | | |
| $k$-means vs. Spectral | .1598 | .6103 | .1233 | .3897 | .2206 |
| Clustering 1 vs. Clustering 2 | .2714 | .6438 | .1988 | .3562 | .2875 |
| **Data set** *iris* | | | | | |
| $k$-means vs. Spectral | .9398 | .9825 | .9610 | .0175 | .9649 |
| **Data set** *wine* | | | | | |
| $k$-means vs. Spectral | .3733 | .6962 | .3328 | .3038 | .3924 |
| Clustering 1 vs. Clustering 2 | .3754 | .7411 | .4232 | .2589 | .4821 |

cluster centers for Clustering 2 are apple, banana and grape, respectively, which are consistent with Clustering-by-Category in the ground truth. The cluster centers of Clustering 3 are yellow, green, and red, respectively, which match the ground truth of Clustering-by-Color. Please note that colors green and red here are both represented by grapes. This further verified that MSC can find meaningful weight vectors (subspaces) corresponding to different stable clusterings hidden in data.

### *D. More Results on the UCI Data Sets*

We also tested MSC on some real datasets from the UCI Machine Learning Data Repository [35]. We report the results on 3 data sets here as examples. Note that each data set in this subsection provides only one ground truth clustering.

*1) Results on Data Sets Balance and Iris:* We used two low-dimensional UCI datasets: *balance* and *iris*. The parameters were set as the same as on the synthetic data.

Table IV compares the clusterings produced by the methods tested against the ground truth. MSC found two stable states on the *balance* dataset. One of the stable states weighted more on the first and second features, while the other stable state put more weights on the second and third features. The sparse property of the simplex constraint can also be observed from the values of these two obtained weight vectors.

The clustering generated by the first stable state was much more consistent with the provided ground truth than the clusterings produced by $k$-means and Spectral. The clustering produced by MSC had a larger eigengap than that produced by the spectral clustering method.

To understand the redundancy between clusterings, Table V compares the clusterings by $k$-means and Spectral, and the clusterings produced by MSC. The redundancy between the clusterings produced by $k$-means and Spectral on the *balance*

TABLE IV. CLUSTERINGS ON THE UCI DATA SETS *Balance*, *Iris* AND *Wine*. All clusterings produced by the methods tested are compared with the ground truth.

| Clusterings produced by methods tested | NMI | RI | AR | MI$^{\downarrow}$ | HI | Eigengap | Weight vector $\mathbf{w}$ |
|---|---|---|---|---|---|---|---|
| **Data set *balance*** | | | | | | | |
| $k$-means | .2702 | .6551 | .2765 | .3449 | .3102 | - | [.2500;.2500;.2500;.2500] |
| Spectral | .2769 | .6636 | .2942 | .3364 | .3271 | .0000 | [.2500;.2500;.2500;.2500] |
| Clustering 1 | **.3215** | **.6928** | **.3556** | **.3072** | **.3856** | .0126 | [.5000;.5000;.0000;.0000] |
| Clustering 2 | .1238 | .5895 | .1389 | .4105 | .1791 | .0126 | [.0000;.5006;.4994;.0000] |
| **Data set iris** | | | | | | | |
| $k$-means | .7582 | .8797 | .7302 | .1203 | .7595 | - | [.2500;.2500;.2500;.2500] |
| Spectral | .7347 | .8679 | .7037 | .1321 | .7358 | .0519 | [.2500;.2500;.2500;.2500] |
| Clustering 1 | **.8366** | **.9341** | **.8510** | **.0659** | **.8683** | .2733 | [.0000;.0000;.8658;.1342] |
| **Data set wine** | | | | | | ($\times 10^{-5}$) | |
| $k$-means | .3823 | .7030 | .3470 | .2970 | .4061 | - | - |
| Spectral | .2945 | .6813 | .2890 | .3187 | .3627 | .0349 | - |
| Clustering 1 | .2859 | .6780 | .2814 | .3220 | .3516 | .0757 | - |
| Clustering 2 | **.5893** | **.8200** | **.5987** | **.1800** | **.6401** | .0526 | - |

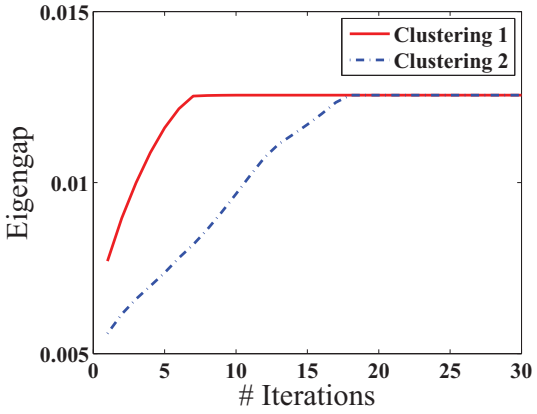



Fig. 6. The changes of eigengap in iterations on data set *balance*.

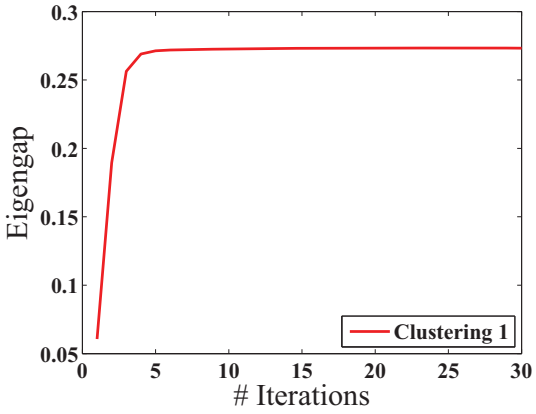



Fig. 7. The changes of eigengap in iterations on data set *iris*.

data set was low. This confirmed that using different clustering methods may generate substantially different alternative clusterings. The second clustering produced by MSC is substantially different from the first one, which confirms the effectiveness of the stability measure.

On the *iris* dataset, MSC found a stable state that weighted more on the third and the fourth features. The clustering is much more consistent with the ground truth comparing to the clusterings produced by $k$-means and Spectral that had much smaller eigengaps. These two methods generated very similar clusterings on this data set and thus failed to provide different alternative clusterings. Remarkably, MSC produced only one stable clustering on this dataset. The results from MSC heuristically indicated that the data set may not allow multiple interesting clusterings. This information is very helpful in practice, since the existing multi-clustering methods cannot determine the number of alternative clusterings in a data set exactly or heuristically.

Figs. 6 and 7 show how the eigengap converges as the number of iterations increases. On both data sets, the eigengaps converge in less than 20 iterations.

*2) Results on Data Set Wine:* *Wine* is a data set of higher dimensionality. The statistics of the data set are included in Table I. The stopping threshold $\tau$ was set to 0.2, and all other parameters were set to the same as the data sets *balance* and *iris*. The results are also shown in Tables IV and V, where we omit the weight vectors limited by space.

$k$-means and Spectral produced quite different clusterings. However, those clusterings were not matching the ground truth. MSC computed two stable clusterings. The second stable clustering output by MSC was very consistent with the ground truth. The first stable clustering is substantially different from the ground truth and the second stable clustering.

Please note that, since the feature values were normalized and then distributed by the weight vector, the eigengap obtained on this data set was much smaller than those obtained on the other two data sets, mainly due to the high dimensionality. The two clusterings produced by MSC had larger eigengaps than that by Spectral, which may be a factor leading to that our clustering is more consistent with the ground truth.

## VI. CONCLUSIONS

In this paper, to address the practical demands on multi-clustering, we tackled the challenges of how to model the quality of clusterings and how to find multiple stable clusterings in a given data set. We applied the idea of clustering stability based on Laplacian eigengap, originally introduced by Meilă and Shortreed in the regularized spectral learning method for similarity matrix learning [10], to multi-clustering. The intuition is that a clustering is stable if small distortions on the attribute values do not affect the discoverability of the clustering. Mathematically, we proved that the larger the eigengap, the more stable the clustering. Based on the notion of stability of clusterings and the underlying analysis on the

Laplacian eigengap, we developed a novel multi-clustering method MSC to obtain a certain number of stable clusterings. We modeled the problem of finding a stable clustering as an optimization problem maximizing the eigengap. The problem is unfortunately non-convex, and thus we proposed a heuristic randomized method using iterative gradient ascent. In order to find multiple stable clusterings, we introduced to the optimization problem a constraint on the difference from the clusterings found so far. An advantage of our method comparing to the existing multi-clustering methods is that our method does not need any parameter about the number of alternative clusterings in the data set. Our method can heuristically estimate the number of meaningful clusterings in a data set, which is infeasible in the existing multi-clustering methods. We also discussed techniques to speed up the search process. An empirical study on synthetic and real data sets clearly demonstrated the effectiveness of our method.

There are a series of interesting questions for future work. For example, for each dataset, we only studied the case when the number of clusters is fixed. How to address data sets where different stable clusterings may have very different numbers of clusters is an interesting direction.

## REFERENCES


[1] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: A review," *ACM Computing Surveys*, vol. 31, no. 3, pp. 264–323, 1999.

[2] C. Tang, A. Zhang, and J. Pei, "Mining phenotypes and informative genes from gene expression data," in *Proceedings of the Nineth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'03)*, Washington, D.C, Aug. 2003.

[3] S. Dasgupta and V. Ng, "Mining clustering dimensions," in *Proceedings of the 27th International Conference on Machine Learning*, Haifa, Israel, 2010, pp. 263–270.

[4] J. Bailey, "Alternative clustering analysis: A review," in *Data Clustering: Algorithms and Applications*, C. C. Aggarwal and C. K. Reddy, Eds. Taylor & Francis, 2013.

[5] X. Dang and J. Bailey, "A framework to uncover multiple alternative clusterings," *Machine Learning*, vol. 98, no. 1-2, pp. 7–30, 2015.

[6] E. Bae and J. Bailey, "COALA: A novel approach for the extraction of an alternate clustering of high quality and high dissimilarity," in *Proceedings of the IEEE International Conference on Data Mining*, Hong Kong, China, 2006, pp. 53–62.

[7] R. Caruana, M. F. Elhawary, N. Nguyen, and C. Smith, "Meta clustering," in *Proceedings of the IEEE International Conference on Data Mining*, Hong Kong, China, 2006, pp. 107–118.

[8] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan, "Automatic subspace clustering of high dimensional data for data mining applications," in *Proceedings of ACM SIGMOD International Conference on Management of Data*, Seattle, WA, 1998, pp. 94–105.

[9] I. Assent, R. Krieger, E. Müller, and T. Seidl, "INSCY: Indexing subspace clusters with in-process-removal of redundancy," in *Proceedings of the IEEE International Conference on Data Mining*, Pisa, Italy, 2008, pp. 719–724.

[10] M. Meilă and S. Shortreed, "Regularized spectral learning," *Journal of Machine Learning Research*, vol. 2006, pp. 1–20, 2006.

[11] E. Bae, J. Bailey, and G. Dong, "A clustering comparison measure using density profiles and its application to the discovery of alternate clusterings," *Data Mining and Knowledge Discovery*, vol. 21, no. 3, pp. 427–471, 2010.

[12] H. Nagesh, S. Goil, and A. Choudhary, "Adaptive grids for clustering massive data sets," in *Proceedings of the 1st SIAM International Conference on Data Mining*, Chicago, IL, 2001, pp. 1–17.

[13] K. Sequeira and M. J. Zaki, "Schism: A new approach to interesting subspace mining," *International Journal of Business Intelligence and Data Mining*, vol. 1, no. 2, pp. 137–160, 2005.

[14] K. Kailing, H.-P. Kriegel, and P. Kröger, "Density-connected subspace clustering for high-dimensiona data," in *Proceedings of the 4th SIAM International Conference on Data Mining*, Lake Buena Vista, FL, 2004, pp. 246–257.

[15] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databased with noise," in *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*, Portland, OR, 1996, pp. 226–231.

[16] H.-P. Kriegel, P. Kröger, M. Renz, and S. Wurst, "A generic framework for efficient subspace clustering of high-dimensional data," in *Proceedings of the IEEE International Conference on Data Mining*, Houston, TX, 2005, pp. 250–257.

[17] E. Müller, I. Assent, R. Krieger, S. Günnemann, and T. Seidl, "Densest: Density estimation for data mining in high dimensional spaces," in *Proceedings of the 9th SIAM International Conference on Data Mining*, Sparks, NV, 2009, pp. 173–184.

[18] U. Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, no. 4, pp. 395–416, 2007.

[19] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Advances in Neural Information Processing Systems 14*, 2001, pp. 849–856.

[20] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society, Series B*, vol. 58, pp. 267–288, 1994.

[21] P. Zhao and B. Yu, "On model selection consistency of lasso," *Journal of Machine Learning Research*, vol. 7, pp. 2541–2563, 2006.

[22] P. Bickel, Y. Ritov, and A. Tsybakov, "Simultaneous analysis of lasso and dantzig selector," *The Annals of Statistics*, vol. 37, no. 4, pp. 1705–1732, 2009.

[23] S. Xiang, X. Tong, and J. Ye, "Efficient sparse group feature selection via nonconvex optimization," in *Proceedings of the 30th International Conference on Machine Learning*, Atlanta, GA, 2013, pp. 284–292.

[24] A. T. Kyrillidis, S. Becker, V. Cevher, and C. Koch, "Sparse projections onto the simplex," in *Proceedings of the 30th International Conference on Machine Learning*, Atlanta, GA, 2013, pp. 235–243.

[25] C. K. I. Williams and M. Seeger, "Using the nyström method to speed up kernel machines," in *Advances in Neural Information Processing Systems 13*, 2001, pp. 682–688.

[26] P. Drineas and M. W. Mahoney, "On the nyström method for approximating a gram matrix for improved kernel-based learning," *Journal of Machine Learning Research*, vol. 6, pp. 2153–2175, 2005.

[27] M. Li, X. Lian, J. T. Kwok, and B. Lu, "Time and space efficient spectral clustering via column sampling," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Colorado Springs, CO, 2011, pp. 2297–2304.

[28] S. P. Lloyd, "Least squares quantization in PCM," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–136, 1982. [Online]. Available: http://dx.doi.org/10.1109/TIT.1982.1056489

[29] L. Hubert and P. Arabie, "Comparing partitions," *Journal of Classification*, vol. 2, pp. 193–218, 1985.

[30] L. G. Shapiro and G. C. Stockman, *Computer Vision*. San Diego, CA: Prentice Hall, 2001.

[31] I. Daubechies, *Ten Lectures on Wavelets*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 1992.

[32] J. Z. Wang, J. Li, and G. Wiederhold, "Simplicity: Semantics-sensitive integrated matching for picture libraries," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 9, pp. 947–963, 2001.

[33] X. Z. Fern and C. E. Brodley, "Random projection for high dimensional data clustering: A cluster ensemble approach," in *Proceedings of the 20th International Conference on Machine Learning*, Washington, DC, 2003, pp. 186–193.

[34] C. Domeniconi and M. Al-Razgan, "Weighted cluster ensembles: Methods and analysis," *ACM Transactions on Knowledge Discovery from Data*, vol. 2, no. 4, 2009.

[35] A. Frank and A. Asuncion, "UCI machine learning repository," 2010. [Online]. Available: http://archive.ics.uci.edu/ml