Noname manuscript No.
(will be inserted by the editor)

Generating Multiple Alternative Clusterings Via Globally Optimal Subspaces

Xuan Hong Dang · James Bailey

Received: date / Accepted: date

Abstract Clustering analysis is important for exploring complex datasets. Alternative clustering analysis is an emerging subfield involving techniques for the generation of multiple different clusterings, allowing the data to be viewed from different perspectives. We present two new algorithms for alternative clustering generation. A distinctive feature of our algorithms is their principled formulation of an objective function, facilitating the discovery of a subspace satisfying natural quality and orthogonality criteria. The first algorithm is a regularization of the Principal Components analysis method, whereas the second is a regularization of graph-based dimension reduction. In both cases, we demonstrate a globally optimum subspace solution can be computed. Experimental evaluation shows our techniques are able to equal or outperform a range of existing methods.

Keywords Alternative clustering · Multiple clusterings · Unsupervised learning

Majority of this work was done while the first author was with The University of Melbourne

Xuan-Hong Dang
Department of Computer Science
Aarhus University, 8200 Aarhus N, Denmark
Tel.: +45 8715 6224
Fax: +45 8715 4113
E-mail: dang@cs.au.dk

James Bailey
Department of Computing and Information Systems
The University of Melbourne, VIC 3010, Australia
Tel.: +61 3 8344 1319
Fax: +61 3 9349 4596
E-mail: baileyj@unimelb.edu.au

1 Introduction

Data clustering categorizes similar data instances into the same clusters. However, when clustering complex data, many solutions may exist and more than one may be reasonable. Addressing this challenge is the goal of the growing research area of alternative clustering, where the aim is to generate multiple dissimilar, yet high quality clusterings of a dataset.

In this paper, we present two algorithms that can generate multiple alternative clusterings. The first is suitable for when the dataset structure is linear, whilst the second can further handle nonlinear cases. Both adopt a transformed feature space approach: a new feature space (subspace) is generated to satisfy certain criteria and then a clustering algorithm is executed using this new feature space to obtain an alternative clustering.

A key contribution of our approach is that the new feature space is generated via the optimization of two objectives, one based on orthogonality (dissimilarity), the other based on quality. We formulate our objective functions in the framework of the eigendecomposition problem. *This has the major advantage that a closed form for the subspace solution is guaranteed to exist and the solution is globally optimum.* Our first technique operates by regularizing the objective function of the Principal Component Analysis (PCA) method [16], whereas the second technique regularizes the graph-based dimension reduction method. PCA attempts to preserve the global variance of the data by projecting into a lower subspace spanned by the leading eigenvectors of the data covariance matrix. We regulate this subspace learning process by incorporating the information from reference clusterings into the PCA optimization function, by using the Hilbert Schmidt Independence Criterion (HSIC) [2]. Similar to mutual information, this criterion is effective in measuring the dependence amongst different random variables. The output is a novel subspace that is independent from any reference clusterings, yet which ensures the global property of data variance being maximized.

In our second approach, rather than retaining the global variance property, we maintain a local property, the geometrical proximity of the data instances. This helps address the cases where the clustering structures may exhibit nonconvex shapes (i.e., clustering boundaries can be of any non-linear form). Experimental results show that the proposed techniques are either superior to or competitive with existing methods across a range of datasets.

In summary, the contributions of our work are:

- We develop the formulation of two novel algorithms for the task of generating multiple alternative clusterings.
- The chief advantage of these two algorithms is their theoretical formulation. To discover each alternative clustering, these algorithms operate by first generating an alternative feature subspace satisfying an objective function that combines both quality and dissimilarity. This feature space
  - Is guaranteed to exist.
  - Is globally optimal with respect to the objective function.

- We perform a number of experiments to demonstrate the advantages of the proposed algorithms and compare against a number of well-known algorithms in the literature.

An outline of the remainder of this paper is as follows. Background about related work in the field of alternative clustering is provided in Section 2. Section 3 defines terminology and formulates the problem. In Section 4, we describe our first alternative clustering algorithm known as RPCA, based on regularized principal components analysis. In Section 5, we describe our second alternative clustering algorithm known as RegGB, based on a regularized graph-based embedding. This is followed by an experimental analysis in Section 6 and conclusions in Section 7.

2 Related Work

A number of algorithms exist for alternative clustering and they can be generally categorized into two groups: those seeking an alternative clustering using the full original data space and those seeking an alterative clustering based on transformed/projected subspaces. We review major studies falling into these two themes in the following discussion.

Studies using the entire original data space: In this first group, the full original feature space is utilized and most studies differ in the way they develop a clustering objective that optimizes both the clustering quality and dissimilarity of an alternative solution. Algorithms developed in [4, 14, 15, 8, 9, 21] can be categorized into this group. In [4], an hierarchical clustering technique named COALA is developed that incorporates the cannot-link constraints (generated from a given clustering) into each its agglomerative merging step. COALA achieves the goal by attempting to satisfy as many of these cannot-link constraints as possible. In [8], the CAMI algorithm is developed to seek two alternative clusterings at the same time and use the entire original data space. Formulating the clustering problem under mixture models, CAMI optimizes a dual-objective function in which the log-likelihood (accounting for clustering quality) is maximized while the mutual information between two mixture models (accounting for the distinction between two clusterings) is minimized. Two algorithms, named Dec-kmeans and Conv-EM, proposed in [15] are also in this line, which aim to respectively regularize the k-means and EM objective functions by incorporating a term accounting for the decorrelation between the two clustering solutions. Representing each clustering solution by a set of mean vectors, these algorithms attempt to maximize the orthogonality of a mean in one clustering with respect to any mean vector of the other clustering (in addition to the maximizing clustering objective of the conventional k-means and EM techniques). The work in [9] takes a different approach which is rooted from information theory. Using the entire original data space, its clustering objective is to maximize the mutual information (MI) between the full feature data instances and the alternative clustering labels, while minimizing such information between alternative and a provided clustering solution. However, instead of using the traditional Shannon entropy [6], this work is developed based on the use of Renyi's entropy, with the corresponding quadratic mutual information [17, 25]. Such an approach allows the MI to be practically approximated when combined with the non-parametric Parzen window technique [24]. Recently, this dual-optimized clustering objective is also exploited in work [21] with an iterative approach, rather than the hierarchical technique adopted in [9].

Studies using a transformed data space: In this second group, most studies seek alternative clusterings via data space transformation, following the general idea that if the transformed spaces are independent (e.g., orthogonal), corresponding clustering solutions discovered from them are dissimilar as well. Algorithms developed in [7, 11, 26, 22] follow this approach, yet they differ in the way of formulating independent (sub)spaces, as well as the functions using for transformation/projection. Work in [7] develops two techniques to find an alternative clustering using orthogonal projections. From least mean square error theory, one knows that the projection of a vector, say $\mathbf{b}$, onto the column space of a matrix A is computed by $P * \mathbf{b}$, of which $P = A(A^T A)^{-1} A^T$ is called a projection matrix. Hence, $(I - P)$ is also another projection matrix which projects $\mathbf{b}$ onto the null space of A^T (i.e., perpendicular to A's column space). The two algorithms developed in [7] exploit this projection equation by viewing each data instance $\mathbf{x}_i$ as a vector $\mathbf{b}$ and in the first algorithm, the columns of A directly are the provided clustering's means, whereas in the second algorithm, they are the features learnt from PCA applied on the provided reference clustering's means. A second, yet potentially uncorrelated, clustering can be found by partitioning the orthogonally transformed data $\mathbf{y}_i = (I - P)\mathbf{x}_i$. Another work along this theme is developed in [11], in which the transformation is applied on the distance matrix learnt from the provided clustering. Compared to the two methods developed in [7], this work has an advantage that it can avoid the problem that the data dimension is smaller than the number of clusters (e.g., spatial datasets). The algorithm developed in [26] takes a different approach, by attempting to transform the data such that data points belonging to the same cluster in the provided reference clustering are now mapped far apart in the newly transformed space. However, a key difference of this method from the other ones is that it does not seek for a completely novel clustering. Instead, it allows some part of the previously known clustering to be retained in the alternative clustering, by adjusting a threshold accounting for the dissimilarity between two solutions. Algorithms proposed in [10] and [22] are based on spectral clustering. The first shows that alternative clusterings can be found by looking at different eigenvectors of the Laplacian matrix, whereas the second (mSC) combines dissimilar subspace learning into the process of spectral clustering. Its clustering objective is thus a dual-function and at each iteration, mSC fixes one term for optimizing the other term. Similar to the first algorithm we will present, mSC uses the HSIC to quantify the correlation, but between two subspaces Our algorithm uses HSIC to quantify the independence of a novel subspace to the provided clustering label directly (rather than to a subspace indirectly learnt from it).

Moreover, although the core algorithm of mSC is a spectral technique, the subspace learning is limited to a linear transformation.

3 Problem Formulation

Assume a d-dimensional dataset X with instances $\{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\}$ and an existing *reference* clustering $C^{(1)}$ (found by any clustering algorithm) which is a partition of X. Let $\mathcal{C}$ be the space of all clusterings of X.

Assume that the *quality* of a clustering can be measured by a quality function $\mathcal{Q} : \mathcal{C} \rightarrow \Re^+$, which captures the inherent "goodness" of the clustering. Assume also, the existence of a *dissimilarity* function $\mathcal{D} : \mathcal{C} \times \mathcal{C} \times \ldots \mathcal{C} \rightarrow \Re^+$, which can measure how different a clustering is compared to an existing set of clusterings. Then, the goal is:
Base Case: Given a single reference clustering $C^{(1)}$, we must generate $C^{(2)}$, an alternative clustering over X, whose clusters $C_i^{(2)}$'s of $C^{(2)}$ satisfy $\bigcup_i C_i^{(2)} = X$ and $C_i^{(2)} \cap C_j^{(2)} = \emptyset$ for $\forall i \neq j$. The quality of the alternative clustering $C^{(2)}$ should be high and $C^{(2)}$ should be dissimilar from $C^{(1)}$.
Recursive case: Given a set of reference clusterings $\{C^{(1)}, C^{(2)}, \ldots\}$, we must generate an alternative clustering $C^{(k)}$, such that $C^{(k)}$ is of high quality and $C^{(k)}$ is dissimilar from all previously found reference clusterings $\{C^{(1)}, C^{(2)}, \ldots\}$.

To generate an alternative clustering, our work will use a subspace learning approach. It aims to map data from an original space X into a new subspace which preserves certain well-defined characteristics of X and is also independent from one or more reference clusterings. Any clustering algorithm can then be executed in this new subspace to generate an alternative clustering. We will mostly focus on the case where only a single reference clustering is provided and briefly indicate how one extends to multiple reference clusterings.

4 Regularized Principal Component Analysis

Principal Component Analysis is a widely used unsupervised technique to find a subspace which maximally preserves the global variance of the data.

Mathematically, let $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n$ be the column vectors of the matrix X, the coordinates of $\mathbf{x}_i$ are considered as random variables and a row in X is the sample of the values associated with a random variable drawn from an unknown probability distribution. PCA finds a new basis $\mathbf{w}_1, \ldots, \mathbf{w}_q$ arranged in columns of a matrix W such that the projection of $\mathbf{x}_i$'s onto these new vectors is as spread as possible; i.e. if $Y = W^T X$, then the variance of Y should be maximized. This can be solved via the equation $Cov(X)\mathbf{w} = \lambda \mathbf{w}$ for eigenvalues $\lambda \geq 0$ (and $Cov(X)$ is the X's covariance). Solutions for $\mathbf{w}$'s all lie in the span of $\mathbf{x}_1, \ldots, \mathbf{x}_n$ since $Cov(X)\mathbf{w} = \sum_i (\mathbf{x}_i . \mathbf{w})\mathbf{x}_i$; and the optimal $\mathbf{w}_1, \ldots, \mathbf{w}_q$ are the q leading eigenvectors (corresponding to the q largest eigenvalues) of $Cov(X)$. Notice that up to d eigenvectors can be found. However in practice, only a small number q leading eigenvectors are retained ($q \ll d$), sufficient to cover most of the data variance (e.g. 95%).

In our problem, given $C^{(1)}$ as a reference clustering, we regularize PCA such that the learned subspace W is independent from $C^{(1)}$. This ensures any clustering solution $C^{(2)}$ derived from W will be dissimilar from $C^{(1)}$. To compute the dependency between subspaces, mutual information could be applied [8]. Nevertheless, this approach requires approximating the joint distribution. Instead, we employ the Hilbert Schmidt Independence Criterion (HSIC) [2], which can achieve the same purpose, but does not require computation of the joint distribution. Furthermore, combining the use of HSIC with PCA can lead to an eigendecomposition problem, for which a globally optimal solution can be computed.

Specifically, given X and Y as two random variables, $HSIC(X,Y)$ quantifies independence between them by computing the squared norm of the cross-covariance operator over the domain $\mathcal{X} \times \mathcal{Y}$ in the Hilbert Space. This norm is close to zero if X and Y are highly independent and is zero iff they are completely independent [2]. At a high level, we are going to use the HSIC measure to assess the correlation between X and Y, where X is the collection of transformed data instances (in the transformed subspace) and Y is the cluster memberships of these data instances in the reference clustering $C^{(1)}$. An advantage of using the HSIC criterion for this task, is that it can naturally accommodate assessing the correlation between domains whose samples X and Y may have complex structures.

More formally, let $\phi(\mathbf{x})$ be a mapping of data sample $\mathbf{x}$ in the input space $\mathcal{X}$ to the reproducing kernel Hilbert space (RKHS) $\mathcal{F}$. We call $\mathcal{F}$ an RKHS if the inner product of two mappings can be represented by a kernel function $k(\mathbf{x},\mathbf{x}') = \langle\phi(\mathbf{x}),\phi(\mathbf{x}')\rangle$. Likewise, we define $\psi(\mathbf{y})$ as the mapping of $\mathbf{y}$ in $\mathcal{Y}$ to an RKHS $\mathcal{G}$ along with the kernel function $l(\mathbf{y},\mathbf{y}') = \langle\psi(\mathbf{y}),\psi(\mathbf{y}')\rangle$. The cross-covariance operator $\mathbb{C}_{\mathbf{xy}}$: $\mathcal{G} \mapsto \mathcal{F}$ is defined as $\mathbb{C}_{\mathbf{xy}} = E_{\mathbf{xy}}[(\phi(x)-\mu_x)\otimes(\psi(y)-\mu_y)]$ with $\otimes$ is the tensor product. The HSIC is defined as the square of the Hilbert-Schmidt norm of $\mathbb{C}_{\mathbf{xy}}$: $HSIC(P_{\mathbf{xy}},\mathcal{F},\mathcal{G}) = \|\mathbb{C}_{\mathbf{xy}}\|^2_{HS}$ where $P_{\mathbf{xy}}$ is the joint distribution of X and Y. We do not have the joint distribution $P_{\mathbf{xy}}$, but given n observations $Z = \{(\mathbf{x}_1,\mathbf{y}_1),\ldots,(\mathbf{x}_n,\mathbf{y}_n)\}$ from $P_{\mathbf{xy}}$, the HSIC can be empirically estimated by [2]:

$$HSIC(Z,\mathcal{F},\mathcal{G}) = (n-1)^{-2}tr(KHLH) \tag{1}$$

where $K,L \in R^{n\times n}$ are Gram matrices with $K_{ij} = k(\mathbf{x}_i,\mathbf{x}_j)$, $L_{ij} = l(\mathbf{y}_i,\mathbf{y}_j)$, $H = I - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T$ and $tr(.)$ is the trace of a matrix and $\mathbf{1}_n$ is a column vector of size n with all 1's and $\mathbf{1}_n^T$ is a row vector with all 1's with size n. We will use $l(\mathbf{y}_i,\mathbf{y}_j) = \langle\mathbf{y}_i,\mathbf{y}_j\rangle$ where $\mathbf{y}_i$ is a binary vector encoding $\mathbf{x}_i$'s cluster label and as shown later, we use the inner product for the kernel function $k(\mathbf{x}_i,\mathbf{x}_j) = \langle\phi(\mathbf{x}_i),\phi(\mathbf{x}_j)\rangle$. For simplicity, the notation $HSIC(X,Y)$, instead of $HSIC(Z,\mathcal{F},\mathcal{G})$, is used for measuring the independence between two random variables X and Y.

Example 1 To give a simple illustration, let us assume that we have 6 instances $\{\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_6\}$ and 3 clusters, where every two consecutive instances belong to the same cluster. The vectors $\mathbf{y}_1 = \mathbf{y}_2 = (1,0,0)^T$ can thus be used to encode cluster labels for $\mathbf{x}_1$ and $\mathbf{x}_2$. Similarly, $\mathbf{y}_3 = \mathbf{y}_4 = (0,1,0)^T$ for $\mathbf{x}_3$ and $\mathbf{x}_4$ and $\mathbf{y}_5 = \mathbf{y}_6 = (0,0,1)^T$ for $\mathbf{x}_5$ and $\mathbf{x}_6$. Y is then a matrix with rows corresponding to each of the $\mathbf{y}_i$. L is a 6×6 matrix giving pairwise similarities between $\mathbf{y}_i$ and $\mathbf{y}_j$ according to their dot product. Likewise, K is a 6×6 matrix giving pairwise similarities between two mappings $\phi(\mathbf{x}_i)$ and $\phi(\mathbf{x}_j)$. Also note that, by its definition, H is a constant matrix fixed for any matrix of size of $n \times n$ (in this example it is 6×6) and the summation over any of one its rows or columns is equal to 0. In this example, each row (and column) of H will contain five entries with value $-\frac{1}{6}$ and one entry with value $\frac{5}{6}$.

Given our measure $HSIC(X,Y)$ we can use it for regularizing the objective function of PCA. We must find the transformation matrix W which leads to a subspace that is independent from $C^{(1)}$ and which also maintains the global variance property of the data. In other words, among all subspaces that are dissimilar to the existing reference solution $C^{(1)}$, we select one that can optimally preserve the data variance. Thus, we regulate the PCA objective function as follows:

$$\begin{aligned} W^* &= \underset{W \in R^{d\times q}}{\arg\max}\ var(W^T X) - HSIC(W^T X, C^{(1)}) \\ &= \underset{W \in R^{d\times q}}{\arg\max}\ var(W^T X) - tr(HKHL) \end{aligned} \tag{2}$$

in which W^* is used to denote the optimal solution for W and $tr(HKHL) = tr(KHLH)$ due to the invariant property of the matrix trace under cyclic permutations. Different kernel functions result in different approximations of the dependency between the variables. In our problem, we use a linear kernel in order to be consistent with the linear projection of the PCA and so that such kernel corresponds to the dot product between two variables. The mapping function is defined as $\phi(\mathbf{x}) = W^T\mathbf{x}$ and hence $K = \langle\phi(X), \phi(X)\rangle = X^T W W^T X$. This gives us:

$$\begin{aligned} & var(W^T X) - tr(HKHL) \\ &\qquad = W^T X X^T W - tr(HX^T W W^T XHL) \\ &\qquad = W^T X X^T W - W^T XHLHX^T W \\ &\qquad = W^T (XX^T - XHLHX^T) W \\ &\qquad = \sum_{i=1}^{q} \mathbf{w}_i^T (XX^T - XHLHX^T)\mathbf{w}_i \end{aligned} \tag{3}$$

The matrix $XHLHX^T$ is symmetric, since both H and L are symmetric. Hence, the eigenvalues of the symmetric matrix $XX^T - XHLHX^T$ are real and the corresponding eigenvectors are pairwise orthogonal. As an eigenvalue decomposition problem, the optimal solution for W^* is the set of q most important eigenvectors $W^* = [\mathbf{w}_1, \ldots, \mathbf{w}_q]$ corresponding to the q largest eigenvalues of the $XX^T - XHLHX^T$. In practice, we select q such that the sum of q largest eigenvalues occupies 90% of the sum of all eigenvalues. Then, to find an alternative clustering $C^{(2)}$ dissimilar from $C^{(1)}$, the k-means clustering technique can be performed in the subspace spanned by these selected eigenvectors. We name this algorithm RPCA and its pseudo code for the base case (provided one reference clustering and find another alternative one) is provided in Figure 1 (extending this to the recursive case is straightforward as sketch in the following discussion).

Algorithm RPCA: Discovery of an alternative clustering (clustering boundary expected to be linear).

```
Input:
  (1) A dataset $X$;
  (2) A reference clustering $C^{(1)}$ over $X$;
Output: An alternative clustering $C^{(2)}$ over $X$;

Compute $L$ with $L_{ij} = \langle \mathbf{y}_i, \mathbf{y}_j \rangle$ where $\mathbf{y}_i$ is the binary vector encoding $\mathbf{x}_i$'s cluster label in $C^{(1)}$;
Compute $H = I - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T$;
Compute $XX^T - XHLHX^T$;
Calculate eigenvalues/eigenvectors of $XX^T - XHLHX^T$;
Sorting eigenvalues in descending order;
Select $W = [\mathbf{w}_1, \ldots, \mathbf{w}_q]$ corresponding to the $q$ largest eigenvalues covering 90% of the sum of all eigenvalues;
$C^{(2)} = \text{k-means}(W^T X)$;
```

Fig. 1 Pseudo code of the RPCA technique for generating one alternative clustering

It can be seen that the value of $XHLHX^T$ can directly affect the computation of the variance matrix XX^T. However, as observed from Eq.(2), since this quantity stems from the measure of independence between two clustering solutions, one may view it somewhat as a constraint added to the main objective of maximizing the variance. In other words, we desire a projection in which the data variance is maximized, yet is subject to the independence condition with respect to the previously given clustering solution $C^{(1)}$. Also from this perspective, we can observe from Eq.(2) that a straightforward extension of our technique to find multiple alternative clusterings (recursive case) is to add more terms of HSIC measures (like constraints), each with respect to a known clustering solution. For example, in seeking for the third clustering $C^{(3)}$ given two previously found $C^{(1)}$ and $C^{(2)}$ solutions, the second term in Eq.(2) is replaced by $(HSIC(W^T X, C^{(1)}) + HSIC(W^T X, C^{(2)}))$. Analogously, deploying this as shown in Eq.(3) gives us $(W^T XHL_1HX^T W + W^T XHL_2HX^T W) =$

$W^T XH(L_1 + L_2)HX^T W$. That means L is replaced by $(L_1 + L_2)$[1] and other matrices remain unchanged.

Theorem 1 *Let q be the number of dimensions of the data subspace that we aim to look for satisfying Eq.(2), then the data subspace found by the regularized PCA technique is ensured to be a globally optimum solution.*

Proof. The proof of the theorem is straightforward, as seeking for the optimal data subspace satisfying Eq.(2) turns out to be the eigen-decomposition problem of the symmetric matrix $XX^T - XHLHX^T$. From linear algebra, the solutions are unique as the set of the matrix's eigenvectors/eigenvalues. Sorting its eigenvalues from large to small, one can easily select q eigenvectors corresponding to the q largest eigenvalues as the global optimum solution.

□

Theorem 2 *Given d as the dimension of the dataset, n as the number of the data instances, the computational complexity of the Regularized PCA technique is $O(n^2 d) + O(d^2)$.*

Proof. The complexity of Eq.(3) depends on the time to compute XX^T, which is $O(n^2 d)$, and the L matrix, which is $O(n^2 c)$, where c is the number of clusters in $C^{(1)}$. Finding eigenvalues/eigenvectors of the $XX^T - XHLHX^T$ matrix generally costs time $O(d^3)$ [13], since its dimension is $d \times d$. However, if only the first few leading eigenvectors are required, techniques such as the power method [30] can reduce computation to $O(d^2)$. Since c is usually smaller than the number of data dimension d, the overall complexity is $O(n^2 d) + O(d^2)$, the same as that of a conventional principal component analysis. Note also that adding more HSIC terms into Eq.(2) (i.e., conditioning on more than one reference clustering) does not affect the time complexity, since it only affects the values in L, but does not change the matrix size.

□

5 Regularized Graph-based Method

We have described a method based on PCA to learn a subspace that is independent from a reference clustering solution, but which preserves the global variance property of the data. The method is linear, making it practically suitable for applications where the boundaries between clusters are linear or close to linear functions. Nonetheless, if the structures in the data are not simple, then the clustering boundaries can be non-linear, requiring a more complex subspace learning technique. We therefore next propose another method to deal with this challenge. Specifically, the algorithm aims to preserve a *local property* of the data, the neighborhood proximity of the data instances.

[1] In order to keep the values in L not proportional to the number of reference clusterings, we normalize L's values within the range of 0 and 1.

Similar to Local Linear Embedding [27] and Laplacian Eigenmap [20], the philosophy is to map the original data into a subspace such that the local neighborhood information in X is optimally preserved. Furthermore, we also require the mapped data to be uncorrelated from any reference clusterings.

We formulate the approach using graph theory. Let $G = \{V, E\}$ be an undirected graph, where $V = \{v_1, \ldots, v_n\}$ is a set of vertices and $E = \{e_{ij}\}$ is a set of edges, each connecting two vertices (v_i, v_j). A vertex v_i corresponds to a data instance $\mathbf{x}_i$ in the dataset X and the edge e_{ij} between v_i and v_j exists if the respective points $\mathbf{x}_i, \mathbf{x}_j$ are close to each other. We define closeness between $\mathbf{x}_i$ and $\mathbf{x}_j$ using the ℓ-nearest neighbors, i.e. $\mathbf{x}_i$ is close to $\mathbf{x}_j$ if it is among the ℓ-nearest neighbors of $\mathbf{x}_j$. We associate a weight $K(i, j)$ between each pair of vertices v_i and v_j which measures their closeness. The non-linear RBF kernel function is widely used for this measure and by definition, $K(i, j) = \exp\left(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/\sigma^2\right)$, where σ is a given parameter. For two nodes that are not connected, the respective weight $K(i, j)$ is set to zero. The $n \times n$ matrix K contains the weights $K(i, j)$ as elements. K is symmetric and typically sparse, since each vertex is only connected to a limited number of neighbors. Furthermore, each $K(i, j) \in [0, 1]$ can loosely be interpreted as the possibility of $\mathbf{x}_i$ and $\mathbf{x}_j$ to be clustered together.

Given the weight matrix K derived from the graph G and a reference clustering $C^{(1)}$, our algorithm learns a novel set $Y = \{\mathbf{y}_1, \ldots, \mathbf{y}_n\}$ (where each $\mathbf{y}_i \in R^q$ $(q \ll d)$ is the mapping of $\mathbf{x}_i \in R^d$) that optimally preserves the local proximity of the data instances while at the same time are independent from the reference solution $C^{(1)}$. The objective function is as follows:

$$\mathbf{y}^* = \arg\min_{\mathbf{y}} \sum_{i=1}^{N} \sum_{j=1}^{N} \|y_i - y_j\|^2 K(i, j) \quad \text{s.t. } S^T\mathbf{y} = 0 \tag{4}$$

where S is a subspace encoding the reference solution $C^{(1)}$ (discussed shortly). In order to keep the problem simple, in Eq.(4) we present for the case where the original data instances are mapped into R^1 space, yielding a 1-dimensional vector $\mathbf{y} = \{y_1, \ldots, y_n\}$ (generalization to multiple dimensions will be presented later once the solution for $\mathbf{y}$ is clear).

Looking at this objective function, we can see that it aims to identify a new subspace $\mathbf{y}$, having two properties:

- The similarity of pairs of transformed data instances is required to be high, if the pair of instances was similar in the original data space. Specifically, if $K(i, j)$ is large ($\mathbf{x}_i$ and $\mathbf{x}_j$ are close in the original space), there is a large penalty in the objective function if the respective points y_i and y_j are mapped far apart. Therefore, optimally preserving the local proximity of the data is equivalent to minimizing this cost function.
- The new data space $\mathbf{y}$ is required to be orthogonal to the subspace S which characterises the reference clustering $C^{(1)}$ (captured by the constraint $S^T\mathbf{y} = 0$). Consequently, a second clustering $C^{(2)}$ learnt from subspace $\mathbf{y}$ is thus likely to be be independent or dissimilar from $C^{(1)}$, due to the orthogonality of the two respective subspaces.

We note that one can show $\mathbf{y} = 0$ is a solution of Eq.(4) since all elements in K matrix are non-negative and so is the objective function. However, note that such a trivial solution is not unique, since the function also reaches minimum when $y_i = y_j$ for any i, j. We therefore will later add more constraints over $\mathbf{y}$ in order to remove these trivial solutions.

Learning the subspace S which characterizes the reference clustering $C^{(1)}$: We now discuss how to find the subspace S that characterizes the provided reference clustering $C^{(1)}$. Different from our regularized PCA method, where the mapping function is clear and linear, we do not have such an explicit mapping function with this nonlinear graph-based approach. We hence employ a non-linear projection technique to find a new set of mapped data characterizing $C^{(1)}$. To achieve this goal, we use the kernel discriminant analysis (KDA) technique [5], a generalization of linear discriminant analysis (LDA). Briefly recall that LDA can be used to discover a lower dimensional representation for a dataset, that is a good characterization for the classes (in our case clusters) existing in that dataset. In particular, it seeks a lower dimensional representation which maximizes the separation between clusters. Mathematically, this is achieved by maximizing the difference between the projected means of the clusters, while also ensuring instances from the same cluster are projected close to one another.

In our context, we are seeking to find a subspace S which is a good characterisation for the reference clustering $C^{(1)}$. However, the shapes of clusters and the boundaries between them may be non linear. We thus employ KDA, which is able to more effectively handle such non linear situations (compared to the LDA). More specifically, KDA maps the original data into the Hilbert space $\mathcal{F}$ using a nonlinear mapping $\phi : \mathbf{x}_i \in R^d \rightarrow \phi(\mathbf{x}_i) \in \mathcal{F}$ and then performs the LDA on this $\mathcal{F}$ space. An optimal direction $\mathbf{v}$ in the $\mathcal{F}$ space is sought for which the between-cluster scatter matrix S_b^{ϕ} is maximized whilst the within-cluster scatter matrix S_w^{ϕ} is minimized, i.e.,

$$\mathbf{v}^* = \arg\max \frac{\mathbf{v}^T S_b^{\phi} \mathbf{v}}{\mathbf{v}^T S_w^{\phi} \mathbf{v}} \tag{5}$$

with $S_b^{\phi} = \sum_{j=1}^{k} (\boldsymbol{\mu}_j^{\phi} - \boldsymbol{\mu}^{\phi})(\boldsymbol{\mu}_j^{\phi} - \boldsymbol{\mu}^{\phi})^T$, $S_w^{\phi} = \sum_{j=1}^{k} \sum_{\mathbf{x}_i \in C_j^{(1)}} (\phi_{\mathbf{x}_i} - \boldsymbol{\mu}_j^{\phi})(\phi_{\mathbf{x}_i} - \boldsymbol{\mu}_j^{\phi})^T$ and $\boldsymbol{\mu}^{\phi}$ and $\boldsymbol{\mu}_j^{\phi}$ are respectively the centroids of the entire data and the $C_j^{(1)}$ cluster (computed in the $\mathcal{F}$ space).

According to the theory of reproducing kernels, it is known that solutions for $\mathbf{v}$ must lie in the span of $\phi(\mathbf{x}_1), \ldots, \phi(\mathbf{x}_n)$ and thus $\mathbf{v}$ can be represented as a linear combination of these vectors: $\mathbf{v} = \sum_{i=1}^{N} \alpha_i \phi(\mathbf{x}_i)$. Instead of directly seeking $\mathbf{v}$, KDA searches for $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_N]^T$. The corresponding optimal $\boldsymbol{\alpha}$ is the leading eigenvector of the generalized eigen-problem $UWU\boldsymbol{\alpha} = \lambda UU\boldsymbol{\alpha}$ [5], in which W and U are diagonal blocking matrices with their entries being defined respectively as:

- $W_{ij} = 1/|C_\ell^{(1)}|$ if $\mathbf{x}_i$ and $\mathbf{x}_j$ belong to the same cluster $C_\ell^{(1)}$ and zero otherwise;
- $U_{ij} = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle = \mathcal{K}(\mathbf{x}_i, \mathbf{x}_j)$ with $\mathcal{K}(.,.)$ is a non-linear kernel function.

The optimal eigenvector $\boldsymbol{\alpha}$ provides the corresponding optimal projection direction $\mathbf{v}$. We cannot compute $\mathbf{v}$ explicitly since recall that the mapping function $\phi(\mathbf{x})$ is unknown. Fortunately, *what we need is just the set of data projected onto* $\mathbf{v}$*, rather than the* $\mathbf{v}$ *itself.* We therefore define our S matrix in Eq.(4) as follows:

$$S = \langle \mathbf{v}, \phi(\mathbf{x}) \rangle = \sum_i \alpha_i \langle \phi(\mathbf{x}) \phi(\mathbf{x}_i) \rangle = \sum_i \alpha_i \mathcal{K}(\mathbf{x}_i, \mathbf{x})$$

Each row in S corresponds to a projected data instance and the number of columns in S equals the number of retained eigenvectors (the dimensionality of the transformed feature space). As a rule of thumb, this number is selected equal to the number of clusters in the reference clustering $C^{(1)}$ minus one.

Seeking an alternative clustering using the mapped data orthogonal to the reference clustering: It is important to note that by learning the subspace S as presented above, it is obvious that S strongly supports the provided reference clustering $C^{(1)}$ (i.e., highly correlates to the cluster labels in $C^{(1)}$). Therefore, by exploiting the orthogonal constraint $S^T\mathbf{y}$ as shown in our objective function Eq.(4), we ensure that the newly mapped data y_i's will be uncorrelated from the reference solution $C^{(1)}$ and subsequently a novel alternative clustering $C^{(2)}$ derived from y_i's is likely to be dissimilar from $C^{(1)}$. We now discuss how to compute this mapped data (the y_i's) and the corresponding novel alternative clustering $C^{(2)}$.

Let us define D as the diagonal matrix with $D_{ii} = \sum_j K(i,j)$ and $L = D - K$, then expanding the sum in Eq.(4) results in $\sum_i y_i^2 D_{ii} + \sum_j y_j^2 D_{jj} - 2\sum_i \sum_j y_i y_j K(i,j) = 2\mathbf{y}^T L \mathbf{y}$. Additionally, as we need the direction of $\mathbf{y}$ rather than its magnitude, the constraint $\mathbf{y}^T D \mathbf{y}$ is further taken into account to remove the freedom of $\mathbf{y}$'s magnitude. Hence, in combination with the clear form of S, the optimization objective with the constraints in Eq.(4) can be solved using the Lagrange method:

$$\mathcal{L}(\beta, \gamma, \mathbf{y}) = \mathbf{y}^T L \mathbf{y} - \beta(\mathbf{y}^T D \mathbf{y} - 1) - \gamma S^T \mathbf{y} \tag{6}$$

in which β and γ are the Lagrange multipliers. Solving this objective function for $\mathbf{y}$ will automatically satisfy our two added constraints over $\mathbf{y}$. Notice that the graph is connected and thus D is a positive definite diagonal matrix and the variable $\mathbf{y}$ can be changed to $\mathbf{y} = D^{-1/2}\mathbf{z}$. The minimization objective is therefore:

$$\begin{aligned} \mathbf{y}^T L \mathbf{y} &= \mathbf{z}^T D^{-1/2} L D^{-1/2} \mathbf{z} \\ &= \mathbf{z}^T Q \mathbf{z} \end{aligned}$$

and the two constraints are:

$$\mathbf{y}^T D\mathbf{y} = \mathbf{z}^T\mathbf{z} = 1 \quad \text{and} \quad S^T\mathbf{y} = S^T D^{-1/2}\mathbf{z} = 0$$

Let us denote $R^T = S^T D^{-1/2}$ and notice that adding the constant 1/2 to the two leading terms does not affect our optimization objective, Eq.(6) can be re-written as follows:

$$\mathcal{L}(\beta,\gamma,\mathbf{z}) = \frac{1}{2}\mathbf{z}^T Q\mathbf{z} - \frac{1}{2}\beta(\mathbf{z}^T\mathbf{z} - 1) - \gamma R^T\mathbf{z} \tag{7}$$

Taking the derivative with respect to $\mathbf{z}$ and equating it to zero gives us:

$$\frac{\delta\mathcal{L}}{\delta\mathbf{z}} = Q\mathbf{z} - \beta\mathbf{z} - \gamma R = 0$$
$$Q\mathbf{z} - \beta\mathbf{z} = \gamma R \tag{8}$$

Pre-multiplying R^T to both sides, it is straightforward to derive $\gamma = (R^T R)^{-1}R^T Q\mathbf{z}$. Substituting this result into Eq.(8) leads to:

$$\begin{aligned}\beta\mathbf{z} &= Q\mathbf{z} - R(R^T R)^{-1}R^T Q\mathbf{z}\\ &= \left(I - R(R^T R)^{-1}R^T\right)Q\mathbf{z}\\ &= PQ\mathbf{z}\end{aligned} \tag{9}$$

which is an eigenvalue problem with $P = I - R(R^T R)^{-1}R^T$. Note that PQ might not be symmetric, though each of its individual matrices is. However, it is possible to show that P is indeed a projection matrix. Specifically, notice that $R(R^T R)^{-1}R^T$ is unchanged under its transpose due to $((R^T R)^{-1})^T = ((R^T R)^T)^{-1} = (R^T R)^{-1}$, which is used to verify $P^T = P$. In addition, $P^2 = I^2 - 2R(R^T R)^{-1}R^T + R(R^T R)^{-1}(R^T R)(R^T R)^{-1}R^T = I - R(R^T R)^{-1}R^T = P$. Consequently, it is true $\beta(PQ) = \beta(PQP)$, meaning that the eigenvalues of both matrices PQ and PQP are the same. Therefore, instead of directly solving $\beta\mathbf{z} = PQ\mathbf{z}$, we solve an easier equation $\beta\mathbf{v} = PQP\mathbf{v}$ (with $\mathbf{v} = P^{-1}\mathbf{z}$) since PQP is a symmetric matrix.

The quadratic form of the symmetric matrix PQP is non-negative. Its eigenvalues β_i's are thus always non-negative and the smallest eigenvalue is $\beta_0 = 0$, corresponding to the eigenvector $\mathbf{v}_0 = P^{-1}D^{1/2}\mathbf{1}$ (with $\mathbf{1}$ is the vector having all unit elements). Such trivial eigenvalues/eigenvectors are removed from our solution, and the final representation is the set of q eignenvectors $\mathbf{y} = D^{-1/2}P\mathbf{v}$ corresponding to the q smallest positive eigenvalues of PQP (in our work, we select q as the number of clusters desired for the alternative clustering minus 1). Notice that the final solutions naturally satisfy the two specified constraints imposed on $\mathbf{y}$ due to the Lagrange multipliers method.

Again, a k-means clustering technique can be applied to the transformed data to obtain the novel alternative clustering $C^{(2)}$. In Figure 2, we provide the pseudo code of this regularized graph-based (RegGB) algorithm for the base case. Similar to our first technique, this approach can be straightforwardly

Algorithm RegGB: Discovery of one alternative clustering (clustering boundary can be non-linear).

```
Input:
  (1) A dataset $X$;
  (2) A reference clustering $C^{(1)}$ over $X$;
Output: An alternative clustering $C^{(2)}$ over $X$;

/*Computing kernel and weight matrices based on graph $G$ */
Compute $K$ with $K(i,j) = \exp\left(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/\sigma^2\right)$ if $x_i$ and $x_j$ are neighbors to each
  other; Otherwise, $K(i,j) = 0$ ;
Compute $W$ with $W_{ij} = 1/|C_\ell^{(1)}|$ if $\mathbf{x}_i, \mathbf{x}_j \in C_\ell^{(1)}$; Otherwise, $W_{ij} = 0$;

/*Learning subspace $S$ characterized for $C^{(1)}$ */
Compute $U$ with $U_{ij} = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle = \mathcal{K}(\mathbf{x}_i, \mathbf{x}_j)$ in that $\mathcal{K}(.,.)$ is the Gaussian kernel;
Calculate $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_N]^T$ as the leading eigenvector of the equation $UWU\boldsymbol{\alpha} =$
  $\lambda UU\boldsymbol{\alpha}$;
Compute $S = \sum_i \alpha_i \mathcal{K}(\mathbf{x}_i, \mathbf{x})$;

/*Learning mapped data $\mathbf{y}_i$'s and novel clustering $C^{(2)}$ */
Compute $D$ with $D_{ii} = \sum_j K(i,j)$
                 $L = D - K$;
Compute $Q = D^{-1/2} L D^{-1/2}$
        $R^T = S^T D^{-1/2}$
        $P = I - R(R^T R)^{-1} R^T$;
Compute $\beta_i$'s and $\mathbf{v}_i$'s as the set of eigenvalues/eigenvectors of $PQP$;
Remove $\beta_i$'s and $\mathbf{v}_i$'s having $\beta_i = 0$ and sort $\beta_i$'s in ascending order;
Select $Y = [\mathbf{y}_1, ....\mathbf{y}_q]$ with $\mathbf{y}_i = D^{-1/2} P \mathbf{v}_i$ corresponding to the $q$ smallest $\beta_i$'s;
$C^{(2)} \leftarrow$ k-means($Y$);
```

Fig. 2 Pseudo code for the RegGB technique when generating one alternative clustering

extended to find multiple alternative clusterings by just including all reference clusterings' subspaces into the S matrix (as S's rows).

Theorem 3 *Let q be the number of dimensions of the data subspace that we aim to look for satisfying Eq.(4), then the data subspace found by the Regularized Graph-Based technique is ensured to be a globally optimum solution.*

Proof. Similar to the case of the Regularized PCA technique, the proof of this theorem is straightforward since searching the optimal subspace data leads to solving the eigen-decomposition problem of the symmetric matrix PQP. Its solutions are always unique as the set of the matrix's eigenvectors/eigenvalues. Ranking these eigenvalues from small to large and removing those that are equal to 0, one can select the set of q eigenvectors (corresponding to the q smallest eigenvalues) as the global optimum solution. □

Theorem 4 *Given d as the dimension of the dataset, n as the number of the data instances, the computational complexity of the Regularized Graph-Based technique is $O(n^2 d)$.*

Proof. The algorithm complexity is dependent on the time to compute nearest neighbors and the K matrix, which both are $O(n^2 d)$. The size of PQP is the same as the size of L, and also the K matrix. So, its eigendecomposition complexity amounts to $O(n^3)$. By keeping only the first few eigenvectors, the power method [30] can be employed to reduce the time to $O(n^2)$. Thus, the overall complexity is $O(n^2 d)$.

□

6 Experimental Evaluation

We next undertake experimental comparisons of our two algorithms, the regularized PCA (RPCA) and the regularized graph-based (RegGB), against the following methods: two algorithms from [7] denoted by Algo1 and Algo2, the ADFT [11], the SC [10] and mSC [22] (which are all subspace based learning approaches); we also compare with COALA [4], Dec-kmeans [15], CAMI [8] and NACI [9] (which all use the entire data space). We set RegGB's parameters as follows: the nearest neighbors to 10 and the kernel width $\sigma = \widehat{\sigma}\,(4/(n(2d+1)))^{\frac{1}{d+4}}$ as in [9] (where $\widehat{\sigma} = \sum_i \sigma_i/d$ and σ_i's are the diagonal elements of the sample covariance matrix), since it was shown to work fairly well by balancing out the bias and the data variance [29]. Each algorithm (except hierarchical technique COALA and NACI) was run ten times for different initializations and we report the average values.

6.1 Clustering Measurements

Clustering results are evaluated according to clustering quality and dissimilarity. For clustering quality, we divide into two cases: if ground truth cluster labels are known, the agreement between alternative clusterings and the correct labels is calculated by the F-measure: $F = 2P \times R/(P + R)$, in which P and R are respectively the precision and recall. Otherwise, we use the Dunn Index denoted by $DI(C)$ (similar to the work [4, 11]), which measures clusters's separation normalized by cluster diameters within the clustering solution C. Mathematically, the Dunn Index is defined by: $DI(C) = \frac{\min_{i \neq j}\{\ \delta(c_i,c_j)\}}{\mathbf{x}_{1 \leq \ell \leq k}\{\triangle(c_\ell)\}}$ with δ: $C \times C \rightarrow \mathbb{R}_0^+$ is the cluster-to-cluster distance and $\triangle$: $C \rightarrow \mathbb{R}_0^+$ is the cluster diameter measure.

For measuring dissimilarity between alternative clusterings, we report the values of two different measures. The first and also the most popular one is the normalized mutual information *NMI* [18, 12, 15]. The second is the Jaccard index (JI): $J(C^{(1)}; C^{(2)}) = n_{11}/(n_{11} + n_{01} + n_{10})$ in which n_{11} is the number of pairs of samples in the same cluster for both $C^{(1)}$ and $C^{(2)}$, n_{01} and n_{10} are the number of samples' pairs belonging to the same cluster in one solution, but not in the other.

Note that *a smaller value of NMI and JI is desirable* (indicating higher dissimilarity between clusterings), while *a larger value of F-measure and Dunn Index is desirable* (indicating a better clustering quality). Also, since methods like Dec-kmeans, SC, CAMI and mSC do not require reference clusterings and instead seek two alternative clusterings simultaneously, we compare by reporting the higher values of F-measure in the case true labels are available, and averaging the Dunn Indexes otherwise.

6.2 Synthetic Datasets

We use two popular synthetic datasets to evaluate the performance of our proposed clustering techniques against the other algorithms. For visualization purposes, these two synthetic datasets are generated in low dimensions (and we leave the cases of high dimensional data for other real-world datasets). The first Syn1 consists of four Gaussian sub-classes with each containing 200 data points generated in 2-dimensional space [7, 11, 15]. The goal of using this dataset is to verify if our two developed techniques are able to uncover an alternative clustering that is orthogonal to a supplied one. For the second Syn2 dataset, we generate a more sophisticated clustering structure of which each sub-class has a non-convex shape. The purpose of using this dataset is to test whether our second technique, which exploits the local proximity property of the data, is able to uncover not only uncorrelated but also non-linear clustering structure.

Row 1 of Figure 3 shows the clustering results of our algorithms for dataset Syn1. The clustering in the first column is provided as the reference solution $C^{(1)}$ and in the second column, we see the alternative clustering $C^{(2)}$ returned by both the RPCA and RegGB techniques. Both our algorithms can find the alternative clustering that is orthogonal to the supplied reference clustering. Furthermore, if we include this clustering as an extra reference solution and search for a second alternative clustering $C^{(3)}$, RegGB returns the solution shown in the third column. In terms of Euclidean distance, this solution seems to be less natural compared to the two previous ones. Nonetheless, it still can be considered as an interesting one since it groups two opposite Gaussians into a cluster and such a clustering is completely independent from the two reference clusterings. We also note that, except for the SC method, all the other alternative clustering techniques are unable to find this second alternative clustering. We report the performance of all algorithms in Table 1 (to be fair, the second alternative $C^{(3)}$ is excluded).

For the Syn2 dataset (which is designed to test the RegGB's ability in uncovering a non-linearly shaped clustering function), looking at row 2 of Figure 3 we show the reference clustering $C^{(1)}$ (column 1), the first alternative clustering $C^{(2)}$ (column 2) and the second alternative $C^{(3)}$ (column 3) found by our RegGB technique. Moreover, to provide more insights, we show the corresponding subspace (the top eigenvector in this 2-clusters case) returned by RegGB at row 3. For this row, the first graph is the leading eigenvector output by the kernel general discriminant analysis method. The second graph is the leading eigenvector output by our algorithm by conditioning on the first

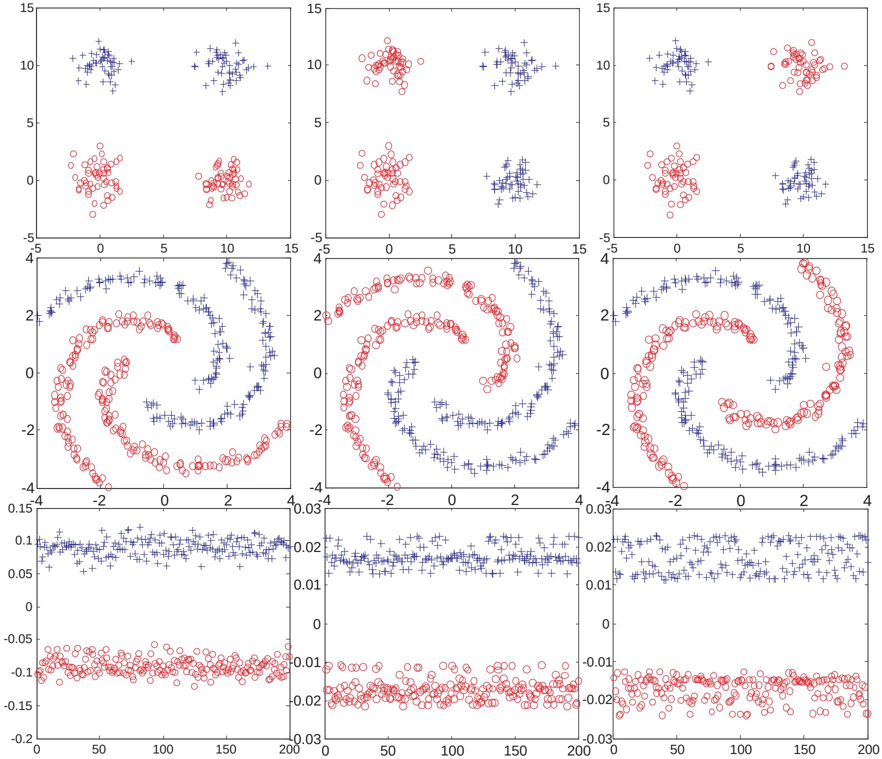

Fig. 3 Alternative clusterings uncovered from Syn1 (1st row) and Syn2 (2nd row) datasets. Images on the 3rd row shows corresponding top eigenvectors of RegGB technique. (Clusters are best visualized in colors.)

solution, and the third one is the top eigenvector found by conditioning on the two previously uncovered solutions.

It is clear that by learning a subspace orthogonal the first solution and attempting to maintain the local proximity of the data, the RegGB method has successfully discovered the second important clustering from the data. The separation between two non-Gaussian shape clusters, as graphically observed from the Figure 3(row 2), is remarkably far apart. This strongly demonstrates the advantage of a graph-based technique in learning a non-linear clustering boundary. For comparison against the other techniques over this second alternative clustering, we list the clustering measurements in Table 1 under the Syn2 row. Obviously, due to their core algorithms being tied to a particular spherical clustering technique (k-means and EM), the methods like Algo1, Algo2, CAMI and Dec-kmeans are not able to find the accurate clustering. Similarly, the RPCA is a linear technique and thus also has relatively poor results. SC and mSC perform better since their underlying clustering tech-

Table 1 Clustering performance for synthetic datasets (excluding second alternative clustering)

	Syn1			Syn2		
Methods	NMI	JI	F	NMI	JI	F
Algo1	0.25	0.41	0.83	0.28	0.34	0.63
Algo2	0.26	0.43	0.81	0.28	0.34	0.63
ADFT	0.12	0.39	0.92	0.30	0.36	0.62
COALA	0.00	0.33	1.00	0.25	0.37	0.58
mSC	0.00	0.33	1.00	0.05	0.35	0.76
SC	0.00	0.33	1.00	0.00	0.33	1.00
NACI	0.00	0.33	1.00	0.00	0.33	1.00
CAMI	0.11	0.38	0.95	0.21	0.34	0.63
Deckm	0.12	0.39	0.93	0.22	0.34	0.62
RPCA	0.00	0.33	1.00	0.03	0.35	0.66
RegGB	0.00	0.33	1.00	0.00	0.33	1.00

nique is the spectral clustering. However, as we can see in overall only the performance of NACI and RegGB can achieve the optimum. Nonetheless, if we further keep searching for another different clustering, only RegGB can find the third one (shown in the column 3, rows 2&3, Figure 3), which merges two opposite spirals into one cluster. This demonstrates RegGB's strength, which uses the eigen-decomposition technique with graph theory, over NACI which exploits the mutual information based approach.

6.3 Pen Digit Dataset

We next provide an experimental comparison on the Pen Digit dataset from [3], consisting of 1602 data samples, where each sample corresponds to a hand written digit. As a digit is being written on a pen-based tablet, 8 x, y positions of the pen are recorded and they form the 16 attributes of the digit. Users could write the digits in any form. We apply our algorithms on this dataset to find different explanations about how the digits have been written. Selecting 2 as the number of clusters within each clustering, we show in Figure 4(a) the reference clustering $C^{(1)}$ found by k-means . In Figures 4(b) and (c), we show two alternative solutions $C^{(2)}$ and $C^{(3)}$ returned by the RegGB. Each picture in the figure corresponds to a cluster centroid.

It is observed that three resultant clusterings provide three different interpretations regarding how the digits have been written. As seen from the first clustering $C^{(1)}$, the writing style of the digits seems to follow clockwise trend with a slightly constant speed in the first cluster but having a slow speed for initial strokes and increasingly high speed for later strokes. Notice that a shorter distance between two adjacent x, y co-ordinates indicates a slow writing speed and inversely, a longer one reveals a faster speed of strokes' writing. For the second clustering $C^{(2)}$, it is possible to observe from the first cluster that the digit writing style is in counter-clockwise, as opposed to the first clustering, with a smooth speed for most of the strokes. Analogically, though digits are mostly written from left to right and going down, the writing style in the second cluster of $C^{(2)}$ demonstrates a non-constant writing speed with initial strokes writing in high speed and subsequent ones with more and more slower. This writing style is clearly different from the style uncovered in the second cluster of $C^{(1)}$, where the speed of writing digits is backward. For the third clustering, we further observe that two clusters' centroids demonstrate two different novel writing styles. While the digit writing manner in the first cluster starts with a stroke from left to right, then with strokes going down to create a very far distance of two ends, the writing style in the second cluster begins with a stroke from right to left, going down then up again to create a closed-end circle. Furthermore, the writing speed in two clusters is also quite different with faster speeds for middle strokes in the first cluster while almost constant speed for all strokes in the second cluster. These two writing styles are not only themselves contrasted to each other but they are also clearly distinguished from those discovered from the first two clusterings $C^{(1)}$ and $C^{(2)}$.

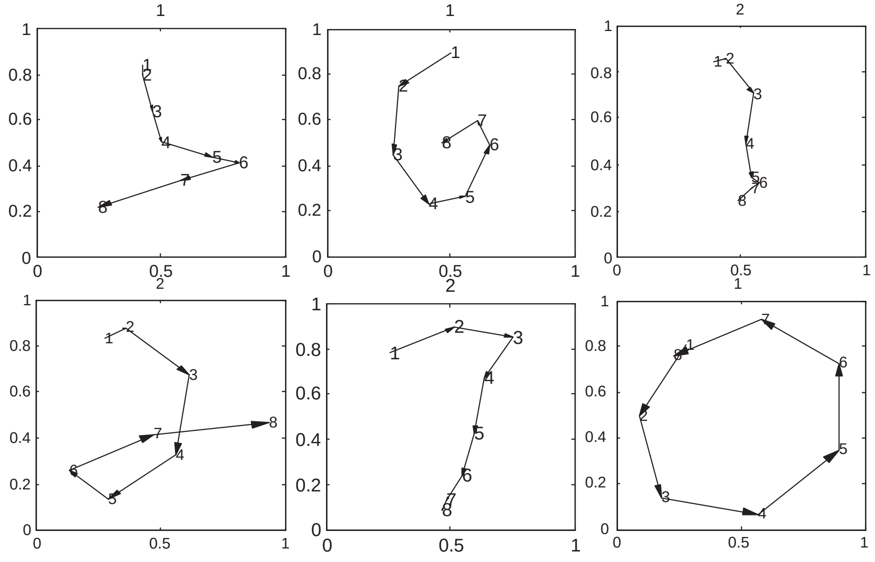

Fig. 4 Reference Clustering ($C^{(1)}$) and two Alternative Clusterings ($C^{(2)}$ and $C^{(3)}$) returned by RegGB on Pen Digit dataset

Quantitative results are in Table 2. Recall that ADFT, NACI and COALA cannot discover multiple alternative clusterings. For these algorithms, the results related to $C^{(3)}$ in Table 2, were computed by providing $C^{(2)}$ as the reference clustering. For these algorithms, $C^{(3)}$ is very close to clustering $C^{(1)}$. Similar behavior was found with Algo1 and Algo2, although they are able to uncover for more than one alternative. In Table 2, both their NMI and Jaccard Index between $C^{(1)}$ and $C^{(3)}$ are large, demonstrating a similar clustering

Table 2 Clustering performance of all algorithms on Pen Digit dataset. (NMI_{ij} stands for the NMI between $C^{(i)}$ and $C^{(j)}$ clusterings. A similar interpretation is applied to JI_{ij} and DI_i.)

	NMI_{12}	NMI_{13}	NMI_{23}	JI_{12}	JI_{13}	JI_{23}	DI_1	DI_2	DI_3
Algo1	0.01	0.74	0.02	0.38	0.83	0.37	1.70	1.59	1.68
Algo2	0.02	0.75	0.02	0.37	0.86	0.39	1.70	1.57	1.67
ADFT	0.01	0.83	0.01	0.42	0.90	0.44	1.70	1.60	1.70
COALA	0.04	0.85	0.01	0.45	0.90	0.36	1.70	1.64	1.67
mSC	0.02	0.22	0.02	0.36	0.44	0.49	1.67	1.60	1.55
SC	0.01	0.25	0.04	0.38	0.45	0.49	1.65	1.59	1.54
NACI	0.01	0.84	0.01	0.35	0.84	0.39	1.70	1.61	1.70
CAMI	0.02	0.04	0.24	0.36	0.42	0.49	1.67	1.62	1.65
Deckm	0.04	0.11	0.26	0.36	0.44	0.49	1.64	1.60	1.59
RPCA	0.01	0.18	0.02	0.33	0.48	0.47	1.70	1.61	1.57
RegGB	0.00	0.15	0.01	0.33	0.44	0.46	1.70	1.62	1.59

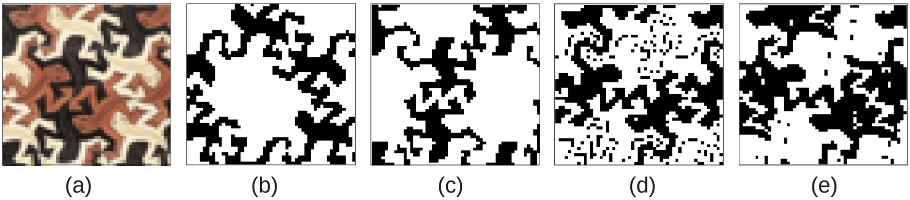

Fig. 5 Image segmentation results on Escher image data. The original image is shown in the first graph (a). Images in the three subsequent graphs (b-d) correspond to three alternative clusterings returned by RegGB. The image in the last graph (e) is the third clustering returned by RPCA (its second clustering is similar to the one in graph (c).

structure between $C^{(3)}$ and $C^{(1)}$. The performances of SC and mSC are similar to each other and slightly better than Algo1,2 over the third clustering. However, in terms of mutual information and Dunn index, they are less successful than those found by RegGB. The performance of RPCA in discovering $C^{(3)}$ is good in terms of clustering dissimilarity. Nevertheless, its clustering quality is still worse than that of RegGB, likely the result of its linear approach to the searching subspace.

6.4 Escher Image Data

For another set of experiments on discovering multiple alternative clusterings, we choose the Escher image data as introduced by [26], where there exists different interpretations of image segmentation (i.e., clustering) to the human eyes. For this sort of image segmentation, each pixel in the image is considered as a data object represented in RGB and HSV features. In Figure 5(a), we show the original Escher image. Similar to the Pen Digit dataset, we set $K = 2$ and the first segmentation $C^{(1)}$ (Figure 5(b)) found by k-means is provided as the reference clustering for all related algorithm. We present the second

Table 3 Clustering performance of all algorithms on the Escher image data (notations's interpretation is analogous to Table2).

	NMI12	NMI13	NMI23	JI12	JI13	JI23	DI1	DI2	DI3
Algo1	0.25	0.81	0.38	0.43	0.57	0.46	3.48	1.19	1.34
Algo2	0.27	0.80	0.41	0.44	0.61	0.45	3.48	1.33	1.21
ADFT	0.25	0.78	0.37	0.41	0.59	0.42	3.48	1.15	2.01
COALA	0.31	0.79	0.44	0.42	0.55	0.47	3.48	1.22	1.90
mSC	0.21	0.09	0.31	0.35	0.36	0.38	3.24	1.17	2.29
SC	0.12	0.08	0.04	0.40	0.54	0.47	3.39	1.31	1.03
NACI	0.03	0.11	0.14	0.40	0.44	0.51	3.48	1.24	1.31
CAMI	0.21	0.36	0.34	0.44	0.58	0.41	3.13	1.28	2.27
Deckm	0.32	0.28	0.35	0.42	0.45	0.56	3.04	1.21	2.30
RPCA	0.24	0.12	0.39	0.41	0.35	0.48	3.48	1.18	2.31
RegGB	0.26	0.08	0.31	0.42	0.39	0.45	3.48	1.17	2.34

alternative segmentation $C^{(2)}$ found by RegGB algorithm (also RPCA) in Figure 5(c) and the third alternative segmentation $C^{(3)}$ found by RegGB and RPCA in Figures 5(d) and (e) respectively. For other algorithms like ADFT, NACI or COALA, their third segmentation $C^{(3)}$ is sought by providing $C^{(2)}$ as the reference segmentation. As observed from Figure 5, our two proposed algorithms are able to uncover three different yet interpretable segmentations from this Escher image data. The first segmentation shown in Figure 5(b) is quite dominant and it corresponds to the yellow reptiles aligned horizontally. The second alternative segmentation shown in Figure 5(c) (showing segmentation with reptiles aligned vertically) is also successfully uncovered by both RegGB and RPCA. However, it is more interesting to observe the third segmentation returned by the RegGB and RPCA algorithms. Our two methods graphically seem to achieve the goal but different levels of noise are presented in their segmentations (see Figures 5(d) and (e)). However compared to RPCA's, the segmentation returned by RegGB is better, as the reptiles aligned in the diagonal way are more visible. Without being aware of the original Escher image, it would be somewhat difficult to realize the third segmentation uncovered by RPCA. For the performance of other algorithms, we observed their behaviors similarly to the Pen Digit data and thus do not repeat their justification. Nevertheless, it is worth mentioning here that none of the algorithms can return a third segmentation close to the one found by RegGB. Amongst them, the $C^{(3)}$ found by mSC is the best and quite similar to that of RPCA but with some minor noise added to its other segmentations. We summarize the overall segmentation results of all algorithms in Table 3.

6.5 CMUFace Dataset

The CMUFace dataset collected from the UCI repository [3] has samples which can be partitioned in different ways (by individual, by pose, etc.). It contains images of 20 people having various facial expressions (neutral, happy, sad, angry), head positions (left, right or straight), and eye states (open or sun-

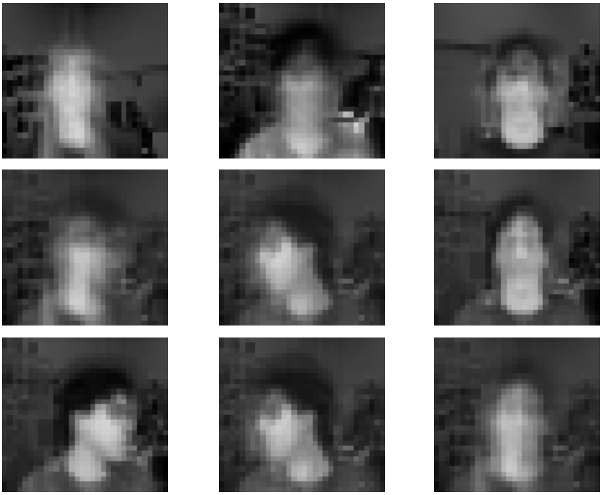

Fig. 6 Results on CMUFace data. 1st row corresponds to supplied clustering. 2nd and 3rd rows respectively correspond to RPCA's and RegGB's alternative clusterings

Table 4 Performance of all algorithms on CMUFace

	Algo	Alg2	ADFT	Coala	mSC	SC	NACI	CAMI	Deckm	RPCA	RegGB
NMI	0.31	0.33	0.29	0.27	0.32	0.37	0.20	0.24	0.26	0.22	0.21
JI	0.34	0.36	0.33	0.32	0.36	0.39	0.24	0.31	0.32	0.27	0.25
F.pose	0.68	0.67	0.69	0.71	0.59	0.51	0.81	0.74	0.72	0.71	0.78
F.per	0.87	0.84	0.89	0.87	0.87	0.81	0.94	0.89	0.90	0.90	0.97

glasses). There are 32 images for each person covering every combination of these features. We randomly select 3 people and all their images. Since it is known which image comes from which person, this ground truth can be used as a reference clustering.

We show the cluster means of this reference in row 1 of Figure 6, and in rows 2 and 3, cluster means of the alternative clustering found by the RPCA and RegGB are respectively shown. Graphically, one observes that the alternative clustering returned by both algorithms has provided another different, yet equally important clustering solution for this dataset. While pictures in the first row show that they represent for different individuals, pictures in the second and third row reveal images have been partitioned according to various poses. This obviously provides another different yet interesting interpretation about the data.

We compare against other algorithms via the results reported in Table 4. The methods like Dec-kmeans and CAMI, which seek two alternative clusterings concurrently, perform fairly well for the clustering based on individuals but achieve only a moderate accuracy on the clustering based on poses. Looking deeper, we found that the clustering based on poses is quite hidden and

Table 5 Results on Ionosphere and Glass data.

	iono.			glass		
Methods	NMI	JI	DI	NMI	JI	DI
Algo1	0.11	0.46	1.46	0.18	0.32	1.25
Algo2	0.13	0.47	1.44	0.20	0.36	1.24
ADFT	0.09	0.42	1.54	0.16	0.33	1.24
COALA	0.11	0.42	1.52	0.12	0.32	1.26
mSC	0.08	0.44	1.55	0.07	0.29	1.22
SC	0.10	0.48	1.51	0.10	0.43	1.21
NACI	0.03	0.36	1.56	0.09	0.31	1.29
CAMI	0.08	0.38	1.50	0.11	0.38	1.26
Deckm	0.10	0.39	1.49	0.14	0.42	1.23
RPCA	0.04	0.39	1.52	0.08	0.29	1.26
RegGB	0.04	0.36	1.59	0.05	0.28	1.32

non-linearly separable, but the configuration based on persons is very obvious. SC and mSC show good results for persons but less successful for the poses. This is because SC simply exploits a single eigenvector (i.e., solely 1-dimension) for each of its suboptimal solution, whereas mSC only uses a linear transformation in its subspace learning. In contrast our algorithms, especially the RegGB, are able to outperform these techniques for this dataset since they not only ensure the dissimilarity in subspace learning, but also make sure the important properties of the data being retained. We also evaluated the case that the pose-based clustering is used as a reference clustering. The accuracy for person-based clustering is summarized in the last row of Table 4 and we see that the performance of RPCA and RegGB is very close to the ground truth labels, with F-measures all above 90%.

6.6 Other real-world datasets

We further test our algorithms against other techniques on two real-world datasets collected from the UCI repository: Ionosphere and Glass. Since these datasets already contain class labels, we utilize them as reference clusterings. The Dunn index (instead of F-measure) is used for clustering quality comparison as ground truth for the alternative clusterings is not known. Results for all techniques are reported in Table 5. Inspecting this table we see that the performances of both RPCA and RegGB are consistently better than most of the other examined methods, especially in term of clustering dissimilarity. Our proposed techniques achieve the clustering dissimilarity with small values of both NMI and JI. These are only slightly larger than that of NACI in the ionosphere, but are far better in the glass data. Of the existing techniques, the methods like Algo1, Algo2 tend to have worse performance compared to the others. The RPCA, albeit a linear method, can achieve better results here since it not only ensures orthogonality but further looks for preserving data variance. The mSC and SC methods seem to achieve highly uncorrelated clusterings, but looking deeper we found that their alternative clusterings are somewhat imbalanced, causing small NMI but much larger JI. Overall, our developed techniques achieve more dissimilar clusterings and their clustering quality is very competitive. For the glass dataset, the performance of RegGB is better than all other algorithms.

6.7 Scalability:

Finally, we make some remarks on the scalability of our proposed algorithms. For both algorithms, recall that the most expensive computational step is their matrix eigenvalue decomposition. Our implementation used Matlab (version 7.11 implemented on Windows 7 with 4 CPUs of 3.4GHz, 8GB RAM) and employed the Lanczos method (which is an adaption of the power method) [19, 28] to compute the leading eigenvalues and eigenvectors.

To provide an idea about typical running times: For the Glass dataset, the RPCA method took 0.3 seconds and RegGB needed 0.5 seconds to find an alternative clustering. For the Ionosphere data, RPCA required 0.7 seconds and RegGB required 0.9 seconds. For the Pen Digit dataset (the largest of all our datasets), RPCA required 19.2 seconds and RegGB required 26.4 seconds. In all datasets, the RegGB algorithm required slightly more time since it further needed the learning subspace computation step (see Fig.2).

These running times indicate our methods are practical for use on modest sized datasets. However, the running time of both our proposed algorithms remains a potential limitation for deployment on very large datasets. One interesting future direction here would be to incorporate random sampling[23] and random projection[1] into our techniques. The challenge here will be how to balance dimensionality reduction (fewer instances and features) against the resultant quality and dissimilarity of the resulting alternative clustering.

7 Conclusions

In this paper, we have developed two algorithms for alternative clustering based on subspace learning. Our methods, each focuses on different data properties, but share the same advantages flowing from their characterization via the eigenvalue decomposition problem. *Importantly, closed form solutions can be found and the subspaces are guaranteed to be globally optimum.* This differentiates our approach from existing work. We evaluated and demonstrated the appealing performance of both methods on a range of datasets and compared against most well-known algorithms in the literature. The experimental results showed that the performance of the RPCA technique is highly competitive whilst that of the RegGB is consistently equal to or better than the state-of-the-art.

References

1. Dimitris Achlioptas. Database-friendly random projections. In *Proceedings of the twentieth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, PODS '01, pages 274–281, New York, NY, USA, 2001. ACM.
2. G. Arthur, B. Olivier, S. Alexander, and S. Bernhard. Measuring statistical dependence with Hilbert-Schmidt norms. In *Algorithmic Learning Theory*, 2005.
3. A. Asuncion and D.J. Newman. UCI machine learning repository, 2007.
4. E. Bae and J. Bailey. COALA: A novel approach for the extraction of an alternate clustering of high quality and high dissimilarity. In *The IEEE International Conference on Data Mining.*, pages 53–62, 2006.
5. G. Baudat and F. Anouar. Generalized discriminant analysis using a kernel approach. *Neural Computation*, 12:2385–2404, 2000.
6. T. M. Cover and J. A. Thomas. *Elements of Information Theory.* Wiley-Interscience, August 1991.
7. Y. Cui, X. Fern, and J. Dy. Non-redundant multi-view clustering via orthogonalization. In *The IEEE International Conference on Data Mining.*, pages 133–142, 2007.
8. X.H. Dang and J. Bailey. Generation of alternative clusterings using the cami approach. In *SIAM International Conference on Data Mining (SDM).*, pages 118–129, 2010.
9. X.H. Dang and J. Bailey. A hierarchical information theoretic technique for the discovery of non linear alternative clusterings. In *ACM Conference on Knowledge Discovery and Data Mining (SIGKDD).*, pages 573–582, 2010.
10. Sajib Dasgupta and Vincent Ng. Mining clustering dimensions. In *International Conference on Machine Learning*, pages 263–270, 2010.
11. I. Davidson and Z. Qi. Finding alternative clusterings using constraints. In *The IEEE International Conference on Data Mining.*, pages 773–778, 2008.
12. X. Fern and W. Lin. Cluster ensemble selection. *Statistical Analysis and Data Mining*, 1(3):128–141, 2008.
13. G.H. Golub and C.F. Van Loan. *Matrix Computations.* Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press, 1996.
14. D. Gondek and T. Hofmann. Non-redundant data clustering. In *The IEEE International Conference on Data Mining.*, pages 75–82, 2004.
15. P. Jain, R. Meka, and I. Dhillon. Simultaneous unsupervised learning of disparate clusterings. In *SIAM International Conference on Data Mining (SDM).*, pages 858–869, 2008.
16. I.T. Jolliffe. *Principal Component Analysis.* Springer Series in Statistics. Springer, 2010.
17. J. Kapur. *Measures of Information and their Application.* John Wiley, 1994.

18. M. Law, A. Topchy, and A. Jain. Multiobjective data clustering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*, pages 424–430, 2004.
19. R. B. Lehoucq, D. C. Sorensen, and C. Yang. *ARPACK Users Guide: Solution of Large-Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods.* SIAM, 1998.
20. B. Mikhail and N. Partha. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *International Conference on Neural Information Processing Systems (NIPS).*, pages 585–591, 2001.
21. X. V. Nguyen and J. Epps. minCEntropy: a novel information theoretic approach for the generation of alternative clusterings. In *The IEEE International Conference on Data Mining.*, pages 521–530, 2010.
22. D. Niu, J. Dy, and I. M. Jordan. Multiple non-redundant spectral clustering views. In *International Conference on Machine Learning*, pages 831–838, 2010.
23. Frank Olken and Doron Rotem. Random sampling from database files: a survey. In *Proceedings of the 5th international conference on Statistical and Scientific Database Management*, SSDBM'1990, pages 92–111, London, UK, UK, 1990. Springer-Verlag.
24. E. Parzen. On estimation of a probability density function and mode. *The Annals of Mathematical Statistics*, 33(3):1065–1076, 1962.
25. J. Principe, D. Xu, and J. Fisher. *Information Theoretic Learning.* John Wiley & Sons, 2000.
26. Z. Qi and I. Davidson. A principled and flexible framework for finding alternative clusterings. In *ACM Conference on Knowledge Discovery and Data Mining (SIGKDD).*, pages 717–726, 2009.
27. T. R. Sam and K. S. Lawrence. Nonlinear dimensionality reduction by locally linear embedding. *Science Journal*, 290(5500):2323–2326, 2000.
28. G. W. Stewart. *Matrix Algorithms Volume II: Eigensystems.* SIAM, 2001.
29. Wand and M. Jones. *Kernel Smoothing-Monographs on Statistics and Applied Probability.* 1994.
30. J. H. Wilkinson. *The Algebraic Eigenvalue Problem.* Claredon Press, 1965.