Iterative Discovery of Multiple Alternative Clustering Views

Donglin Niu, *Student Member, IEEE*, Jennifer G. Dy, *Member, IEEE*, and Michael I. Jordan, *Fellow, IEEE*

Abstract—Complex data can be grouped and interpreted in many different ways. Most existing clustering algorithms, however, only find one clustering solution, and provide little guidance to data analysts who may not be satisfied with that single clustering and may wish to explore alternatives. We introduce a novel approach that provides several clustering solutions to the user for the purposes of exploratory data analysis. Our approach additionally captures the notion that alternative clusterings may reside in different subspaces (or views). We present an algorithm that simultaneously finds these subspaces and the corresponding clusterings. The algorithm is based on an optimization procedure that incorporates terms for cluster quality and novelty relative to previously discovered clustering solutions. We present a range of experiments that compare our approach to alternatives and explore the connections between simultaneous and iterative modes of discovery of multiple clusterings.

Index Terms—Kernel methods, non-redundant clustering, alternative clustering, multiple clustering, dimensionality reduction

1 Introduction

The goal of exploratory data analysis is to find structure and interesting patterns in data, and to summarize, organize, and/or extract information from data. Many of these goals can be formulated as clustering problems, but existing clustering methods are often not sufficiently flexible to cover the range of desired data analytic goals. In particular, most clustering algorithms find only a single partitioning of the data [1], but data items can often be grouped together in several different ways for different purposes. For example, face images can be grouped based on their pose or on identity. Given the same medical data, what is interesting to physicians might be different from what is interesting to insurance companies.

Clustering algorithms are generally based on some notion of cluster quality and/or some notion of similarity among data items. Each such criterion has a particular bias, and given that "Different classifications [clusterings] are right for different purposes, so we cannot say any one classification is best" [2], users of clustering are often compelled to try a variety of different algorithms. It would be desirable, however, to capture this goal of diversity formally, and optimize directly for diversity within an algorithmic framework; this may provide a more systematic exploration of the range of alternatives.

- *D. Niu and J.G. Dy are with the Department of Electrical and Computer Engineering, Northeastern University, 409 Dana Research Bldg., Boston, MA 02115. E-mail: {dniu, jdy}@ece.neu.edu.*
- *M.I. Jordan is with the Computer Science Division and Department of Statistics, University of California, 387 Soda Hall 1776, Berkeley, CA 94720. E-mail: jordan@cs.berkeley.edu.*

Manuscript received 21 Dec. 2011; revised 9 Sept. 2012; accepted 7 Sept. 2013. Date of publication 22 Sept. 2013; date of current version 13 June 2014. Recommended for acceptance by I.S. Dhillon. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below. Digital Object Identifier no. 10.1109/TPAMI.2013.180

Our approach is built on spectral clustering, which, relative to traditional clustering methods, has two major advantages in the multiple clustering setting. First, spectral clustering captures a flexible notion of cluster shape which is not restricted to convex or homogeneous clusters, a particularly useful feature when attempting to discover multiple clusterings. Second, as we show here, the standard spectral clustering objective function turns out to be closely related to the Hilbert-Schmidt independence criterion [3], a general measure of nonlinear dependence. This same criterion can be used to capture a notion of alternativeness or non-redundancy of clusterings. This motivates our general optimization-based framework that combines cluster quality and diversity criteria into a single optimization functional.

1.1 Related Work

Although the literature on clustering is enormous, there has been relatively little attention paid to the problem of finding multiple non-redundant clusterings. There are two general ways to find alternative clustering solutions: one is to find multiple solutions *simultaneously* and the other is to find one alternative solution given known clusterings *iteratively*.

An early paper on alternative clustering was by Gondek and Hofmann [4], who suggest finding an alternative clustering via conditional information bottleneck. This approach is dependent on distributional assumptions. Bae and Bailey [5] utilize "cannot-link constraints" imposed on data points belonging to the same group (as defined by a previous clustering) and agglomerative clustering in order to find an alternative clustering. Both the information bottleneck approach and the agglomerative clustering approach are designed to find a single alternative solution given an existing one. In general, however, there may be more than two alternative clustering interpretations. Cui et al. [6], [7] developed an iterative approach that was not dependent on a specific clustering algorithm and that can find multiple

alternative views by clustering in the subspace orthogonal to the clustering solutions found in previous iterations. They directly address the problem of finding several (more than two) alternative clustering solutions by iteratively finding one alternative solution given the previously found clustering solutions. Davidson and Qi [8] also developed an approach that was not dependent on a specific clustering algorithm. They represent the existing clustering as a matrix of distances and obtain an alternative distance metric by retaining the left and right singular vectors and taking the inverse of the singular values of the original distance matrix. In [9], they suggest preserving properties of the original data while searching for alternativeness by minimizing the Kullback-Leibler divergence between the original data and the transformed data subject to the constraint that the sum-of-squared error between samples in the projected space with the means of the existing clusters is smaller than a pre-specified threshold. Dang and Bailey [10] finds an alternative clustering that preserves data characteristics by maximizing the mutual information between the new clusters and data at the same time as minimizing the alternative from a reference clustering.

Simultaneous approaches, on the other hand, discover multiple alternative solutions jointly. Caruana et al. [11] generate a diverse set of clustering solutions by either random initialization or random feature weighting. These solutions are then "meta-clustered" using an agglomerative clustering based on Rand index for measuring similarity between pairwise clustering solutions. Jain et al. [12] find two disparate clusterings simultaneously by minimizing a sum-of-squared objective for the two clustering solutions while at the same time minimizing the correlation between these two clusterings. CAMI [13] simultaneously discovers two disparate clusterings by optimizing for cluster quality, quantifying these criteria by maximizing the likelihood of Gaussian mixture models and minimizing the mutual information between them. The method [12] is based on K-means and CAMI [13] is based on Gaussian mixtures; both are thus limited to convex clusters. Dasgupta and Ng [14] obtain multiple clustering views by using each eigenvector in standard spectral clustering for each view. Poon et al. [15] formulated a probabilistic latent pouch tree model for selecting features in each clustering solution. Guan et al. [16], Mansinghka et al. [17], and Niu et al. [18] formulated nonparametric Bayesian models to learn multiple clusterings and features in each clustering solution.

Another line of work which is related yet different from the general direction of the methods in the above two paragraphs is that of subspace clustering [19], [20]. The goal of subspace clustering is to discover clusters hidden in high-dimensional spaces, where each cluster is embedded in its own subspace. It is similar to our work in that the broader definition of subspace clustering allows samples to belong to multiple clusters and is also concerned about discovering quality clusters that are non-redundant vis-a-vis the other clusters [21], [22]. However, the main distinction between subspace clustering and this paper is that, here each clustering view consists of multiple clusters that partition the data, and an alternative view means another partitioning of the data. Thus, cluster quality and alternativeness, in our case, is defined in terms of data partitionings; and the features or low-dimensional subspace in each view describes the subspace which defines the notion of similarity on which the samples are partitioned. Subspace clustering, on the other hand, defines cluster quality based on each individual cluster embedded in its own subspace.

1.2 Contributions of This Work

Learning from all these initial attempts to address this new clustering paradigm, we advance the field in the following way. We address finding multiple alternative views, as in Cui et al. [6], [7]. We utilize an independence criterion to evaluate alternativeness/non-redundant views. Our criterion can measure nonlinear dependencies, making it more flexible than the orthogonalization approach [6] and the sum-squared-distance approach [9]; both of these approaches capture only linear dependencies. Moreover, we endow our method with an adjustable parameter to tradeoff the quality of clustering and alternativeness. This is similar in spirit to previous work [9], [10], [13], but differs in that our approach does not need to estimate probability distributions or make restrictive assumptions regarding these distributions. We achieve this by using a kernel dependence measure, the Hilbert-Schmidt independence criterion (HSIC) [3], to quantify alternativeness between clustering solutions. Moreover, we both find an alternative clustering and the lower-dimensional subspace in which this clustering resides in a single optimization formulation; in contrast, previous work [6], [9] finds the transformed space first, then applies a clustering algorithm. In addition, while [5], [10], [13] find an alternative clustering in the original space; our approach finds alternative clusterings in an alternative projected (potentially lower-dimensional) spaces. An additional benefit of our formulation is that in the special case of using a linear kernel, our approach can be solved via an eigen-decomposition. Furthermore, if we do not need to obtain the low-dimensional subspace explicitly but only need to learn the cluster embeddings, our method also reduces to an eigenvalue problem.

A preliminary version of the work reported here first appeared in [23]. That work differs from the current paper in that it focuses on discovering the different clusterings *simultaneously*; this manuscript introduces approaches for discovering alternative clusterings *iteratively*. Moreover, we provide connections and comparisons between these alternative modes of discovery (simultaneous and iterative). Finally, we identify two special cases of particular interest which can be solved via eigen-decomposition and present empirical comparisons between these special cases and the general case.

1.3 Organization of This Paper

In Section 2, we present our general formulation for discovering alternative clusterings. In Section 3, we discuss the special case of linear kernels and show that this leads to an eigenvalue problem. In Section 4, we present another special case in which only a cluster embedding is desired and show that this also leads to an eigenvalue problem. In Section 5, we explore experimentally the various settings for alternative clustering discovery: (1) simultaneous versus iterative

discovery, (2) dimensionality reduction versus none (original space), and (3) sufficient dimensionality reduced subspace versus embedding view. Then, in Section 7 we present an empirical study of our iterative approach compared to other methods on synthetic and real data. Finally, we report our conclusions in Section 8.

2 Iterative Discovery of Alternative Clusterings

Given a current partitioning or clustering solution P_0, the goal is to find an alternative partitioning P_t. A partitioning P_t here denotes a set of clusters $\{C_1, \ldots, C_c\}$. The number of possible partitionings of a data set of size n is very large (it is given by the Bell number), but not all of these groupings are meaningful. We would like to find an alternative partitioning P_t that is of *good quality* and that is *novel* (meaning, different or non-redundant) when compared to the current partitioning P_0 so as to provide a possibly new discovery to the analyst. When additional alternative clusterings are desired, the process continues—searching for an alternative partitioning P_t given all the previous partitionings $\{P_j\}$, where $j = t-1, \ldots, 0$, up to t equal to the number of desired alternative solutions, v.

2.1 Formulation

Let our n data samples be denoted $\{x_1, \ldots, x_n\}$, with each x_i a column vector in $\mathbb{R}^d$, and $X = [x_1, \ldots, x_n]^T \in \mathbb{R}^{n \times d}$ represent our data matrix, where $(\cdot)^T$ is the transpose of a matrix. Given an existing clustering solution, P_0, let us define a cluster labeling matrix Y_0 of size n by c, where n is the number of instances and c is the number of clusters. If x_i belongs to cluster j in P_0, $y_{ij} = 1$, otherwise it is 0. Similarly, let us define a cluster labeling matrix U for the alternative partitioning P_t. At iteration t, we consider $\{P_j\}$ $(j = t-1, \ldots, 0)$ as the previously found partitionings. At this point, we now have more than one existing clustering solutions. We represent this multiple existing clusterings by augmenting all the existing cluster labeling matrices in a single matrix $Y = [Y_0, \ldots, Y_{t-1}]$ which is now of size n by $\sum_{j=0}^{t-1} c_j$ (Y_j is the cluster labeling matrix and c_j is the number of clusters per iteration). Thus, in each iteration, we are given a matrix Y (representing all the previously presented solutions) and our goal is to find an alternative clustering solution U that is both *novel* and of *good quality*.

2.1.1 Cluster Quality and Spectral Clustering

There are many ways to define the quality of clusters resulting in a variety of clustering algorithms in the literature [1], [24]. In this paper, we focus on spectral clustering because it is a flexible clustering algorithm that is applicable to different types of data and makes relatively weak assumptions on cluster shapes (clusters need not be convex or homogeneous). Spectral clustering can be presented from different points of view [25]; here, we focus on the graph partitioning viewpoint. We are given a set of n data samples, $\{x_1, \ldots, x_n\}$, with each x_i a column vector in $\mathbb{R}^d$, and we are given a set of similarities, $\{k_{ij}\}$, between all pairs x_i and x_j, where $k_{ij} \geq 0$. Let $G = \{V, E\}$ be a graph, with $V = \{v_1, \ldots, v_n\}$ the set of vertices and E the set of edges. Each vertex v_i in this graph represents a data sample x_i, with the similarities k_{ij} treated as edge weights. When there is no edge between v_i and v_j, $k_{ij} = 0$. Let us represent the similarity matrix as a matrix K with elements k_{ij}. This matrix is generally obtained from a kernel function, examples of which are the Gaussian kernel ($k(x_i, x_j) = \exp(-\|x_i - x_j\|^2 / 2\sigma^2)$) and the polynomial kernel ($k(x_i, x_j) = (x_i \cdot x_j + c)^p$).

The goal of spectral clustering is to partition the data $\{x_1, \ldots, x_n\}$ into k disjoint groups, $C_1, \ldots, C_k$, such that the similarity of the samples *between groups* is low, and the similarity of the samples *within groups* is high. There are several objective functions that capture this desideratum; in this paper we focus on the *normalized cut* objective. The k-way normalized cut, $NCut(G)$, is defined as follows: $NCut(C_1, \ldots, C_k) = \sum_{c=1}^{k} \frac{cut(C_c, V \backslash C_c)}{vol(C_c)}$, where the cut between sets $\mathcal{A}, \mathcal{B} \subseteq V$, $cut(\mathcal{A}, \mathcal{B})$, is defined as $cut(\mathcal{A}, \mathcal{B}) = \sum_{v_i \in \mathcal{A}, v_j \in \mathcal{B}} k_{ij}$, the *degree*, d_i, of a vertex, $v_i \in V$, is defined as $d_i = \sum_{j=1}^{n} k_{ij}$, the volume of set $\mathcal{A} \subseteq V$, $vol(\mathcal{A})$, is defined as $vol(\mathcal{A}) = \sum_{i \in \mathcal{A}} d_i$, and $V \backslash \mathcal{A}$ denotes the complement of $\mathcal{A}$. In this objective function, note that $cut(C_c, V \backslash C_c)$ measures the between-cluster similarity and the within-cluster similarity is captured by the normalizing term $vol(C_c)$. The next step is to rewrite $NCut(G)$ using an indicator matrix U of cluster membership of size n by k and to note that $NCut(G)$ takes the form of a Rayleigh coefficient in U. Relaxing the indicator matrix to allow its entries to take on any real value, we obtain a generalized eigenvalue problem. That is, the problem reduces to the problem

$$\begin{aligned} \max_{U \in R^{n \times k}} \quad & \mathrm{tr}(U^T D^{-1/2} K D^{-1/2} U) \\ \text{s.t.} \quad & U^T U = I, \end{aligned} \tag{1}$$

where $\mathrm{tr}()$ stands for trace of a matrix. The solution is to set U equal to the k eigenvectors corresponding to the largest k eigenvalues of the matrix $D^{-1/2} K D^{-1/2}$. This yields the spectral embedding. Based on this embedding, the discrete partitioning of the data is obtained from a "rounding" step. One specific rounding algorithm, due to Ng et al. [26], is based on renormalizing each row of U to have unit length and then applying K-means to the rows of the normalized matrix. We then assign each x_i to the cluster that the row u_i is assigned to.

2.1.2 Novelty and HSIC

To ensure that the alternative clustering is novel compared to existing clustering solutions, we minimize the dependence between the transformed data $XW \in \mathbb{R}^{n \times q}$ in the alternative view and the existing clustering solutions Y. W is a d by q transformation matrix which transforms the data X with dimension d to a lower dimensional space with dimension q. The work [6] applies orthogonal projection to find an alternative view; however, orthogonal projection only captures linear dependencies; similarly the work [9] use sum-of-squared distance to measure dependence, which also only captures linear dependence. Gondek and Hofmann [4] utilize a conditional information bottleneck to capture nonlinear dependence, but this requires estimating the joint probability distribution. In this paper, we measure dependence in terms of a kernel dependence measure, the Hilbert-Schmidt independence criterion [3]. This criterion

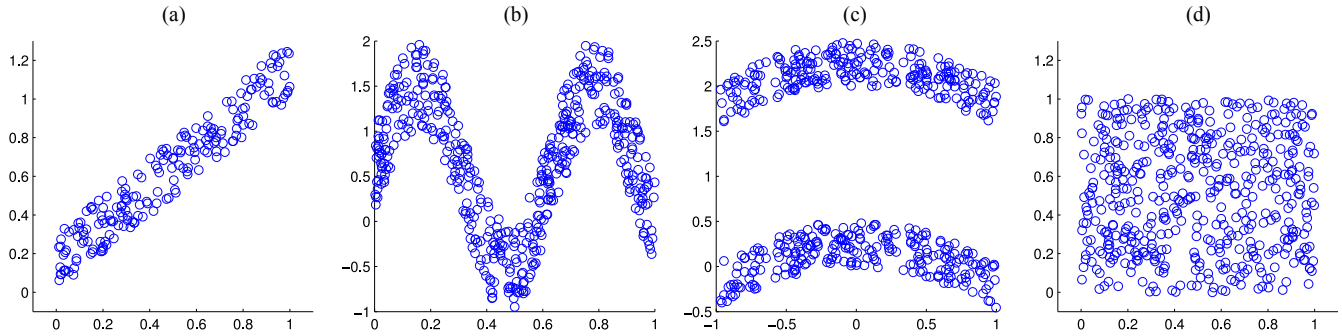


Fig. 1. Illustrative example comparing HSIC with correlation coefficient, ρ: (a) x and y are linearly correlated with some noise (HSIC $= 0.53$, $\rho = 0.81$), (b) uncorrelated but dependent (HSIC $= 0.41$, $\rho = 0$), (c) uncorrelated but dependent (HSIC $= 0.14$, $\rho = 0$), (d) independent (HSIC $= 0$, $\rho = 0$).

measures dependence by mapping variables into a reproducing kernel Hilbert space (RKHS) such that correlations measured in that space correspond to high-order joint moments between the original distributions [27]. This approach is able to estimate dependence between variables without explicitly estimating the joint distribution of the random variables and without having to apply discretization to continuous variables. We now describe this method in some detail.

Consider $\mathcal{X}$ and $\mathcal{Y}$ to be two domains with samples (x, y) that are drawn from these two domains. Let us define a mapping $\phi(x)$ from $x \in \mathcal{X}$ to kernel space $\mathcal{F}$, such that the inner product between vectors in that space is given by a kernel function $k_1(x, x') = \langle \phi(x), \phi(x') \rangle$. Let $\mathcal{G}$ be a second kernel space on $\mathcal{Y}$ with kernel function $k_2(\cdot, \cdot)$ and mapping $\varphi(y)$. A linear cross-covariance operator $C_{xy} : \mathcal{G} \to \mathcal{F}$ between these feature maps is defined as: $C_{xy} = E_{xy}[(\phi(x) - \mu_x) \otimes (\varphi(y) - \mu_y)]$, where $\otimes$ is the tensor product. The Hilbert-Schmidt norm $\|\cdot\|^2_{\text{HS}}$ of this cross-covariance operator defines the HSIC measure of dependence between two random variables, x and y, as follows:

$$\begin{aligned}\text{HSIC}(p_{xy}, \mathcal{F}, \mathcal{G}) &= \left\|C_{xy}\right\|^2_{\text{HS}} \\ &= E_{x,x',y,y'}[k_1(x, x')k_2(y, y')] \\ &\quad + E_{x,x'}[k_1(x, x')]E_{y,y'}[k_2(y, y')] \\ &\quad - 2E_{x,y}[E_{x'}[k_1(x, x')]E_{y'}[k_2(y, y')]].\end{aligned}$$

Given n observations $Z := \{(x_1, y_1), \ldots, (x_n, y_n)\}$, we can estimate the HSIC by

$$\text{HSIC}(Z, \mathcal{F}, \mathcal{G}) = (n-1)^{-2}\text{tr}(K_1 H K_2 H), \tag{2}$$

where K_1, $K_2 \in R^{n \times n}$ are the Gram matrices $K_{1,ij} = k_1(x_i, x_j)$, $K_{2,ij} = k_2(y_i, y_j)$, and $H_{ij} = \delta_{ij} - n^{-1}$ centers the Gram matrix to have zero mean in the feature space. To ensure that subspaces in different views provide non-redundant information, we use HSIC to penalize for dependence between data in these subspaces.

Fig. 1 shows an illustrative example suggesting why HSIC is better than correlation for capturing high order dependencies between two random variables. In Fig. 1a, x and y are linearly correlated with some noise. In Figs. 1a and 1c, two variables are uncorrelated but dependent. In Fig. 1d, they are independent. The HSIC values are 0.53, 0.41, 0.14 and 0 respectively, whereas, the correlation coefficient values are 0.81, 0, 0 and 0 respectively. This figure confirms that compared to the correlation coefficient, HSIC is better in measuring the dependence between variables because it takes higher-order moments into account.

2.1.3 Relation between HSIC and Spectral Clustering

In this section, we show that spectral clustering can be expressed as the HSIC between the variable x and the embedding U as indicated in [28], [29]. Let $K_1 = D^{-1/2}KD^{-1/2}$ be the kernel defined by x, where K is the similarity kernel with elements $k_{ij} = k(x_i, x_j)$. $K_2 = UU^T$ be the second kernel defined by embedding U. For notational convenience, let us assume that K_1 and K_2 are centered and ignore the scaling factor $(n-1)^{-2}$, and use $\text{HSIC}(X, U) = \text{tr}(K_1 K_2)$. Then,

$$\begin{aligned}\text{HSIC}(X, U) &= \text{tr}(D^{-1/2}KD^{-1/2}UU^T) \\ &= \text{tr}(U^T D^{-1/2}KD^{-1/2}U),\end{aligned}$$

which is the spectral clustering objective.

2.1.4 Learning the Low-Dimensional Subspace

When we search for quality alternative clustering U, unlike standard spectral clustering which utilizes all the original features to compute the kernel similarity matrix K between samples, here we compute the kernel similarity in a reduced dimensional subspace W (W is d by q, where $q < d$) as follows:

$$k_{ij} = k(W^T x_i, W^T x_j).$$

In this paper, we refer to this subspace as a *sufficient dimensionality reduced subspace*. This notion allows us to search for alternative clustering solutions in different subspace views W. We incorporate learning of the subspace in our approach to learning an alternative clustering because we observe that in real data, different subspaces provide different clustering interpretations.

2.1.5 Overall Objective

Given a matrix Y (representing all the previously presented solutions), we find an alternative clustering solution U in subspace W that is both *novel* and of *good quality* by optimizing the following:

$$\begin{aligned} \max_{U,W} \quad & \text{HSIC}(XW, U) - \lambda \, \text{HSIC}(XW, Y) \\ \text{s.t.} \quad & U^T U = I \\ & W^T W = I. \end{aligned} \tag{3}$$

Equivalently, we can express the first term using the spectral clustering criterion as follows:

$$\begin{aligned} \max_{U,W} \quad & \text{tr}(U^T D^{-1/2} K D^{-1/2} U) - \lambda \text{HSIC}(XW, Y) \\ \text{s.t.} \quad & U^T U = I \\ & k_{ij} = k(W^T x_i, W^T x_j) \\ & W^T W = I, \end{aligned} \tag{4}$$

where I is the identity matrix.

Maximizing (3) and equivalently (4) is achieved when the first term is large (large dependence between XW and clustering U which implies a good clustering) and the second term is small (small dependence between XW and existing clusterings Y which implies not much similarity to Y). The first term measures *cluster quality* while the second term measures *novelty*. Note that the kernel similarity matrix K is defined in subspace W. λ is a regularization parameter that controls the tradeoff between these two criteria.

2.2 Optimization

We optimize the objective function 6 by alternately optimizing the relaxed clustering indicator matrix U and the transformation matrix W in two steps as follows:

Assuming W fixed, we optimize for U. With projection operators W fixed, we can calculate the similarity and degree matrices K and D. Similar to spectral clustering, here we relax the indicator matrix U to take on real values. The problem now becomes a continuous optimization problem resulting in an eigenvalue problem. The solution for U is equal to the first k eigenvectors (corresponding to the largest k eigenvalues) of the matrix $D^{-1/2} K D^{-1/2}$, where k is the number of clusters in the alternative clustering solution. Note that unlike applying spectral clustering in the projected space $W^T x$, this optimization step stops here; it retains U as a real-valued matrix and does not need to explicitly assign the cluster membership to the samples.

Assuming U fixed, we optimize for W. Interestingly, when U is fixed, the objective can be written in the following form:

$$\sum_{ij} \gamma_{ij} k_{ij} = \sum_{ij} \left(\frac{\mathbf{u}_i^T \mathbf{u}_j}{d_i d_j} - \lambda \tilde{y}_{ij} \right) k_{ij}, \tag{5}$$

optimized under the constraint $W^T W = I$, where u_{ij} are the elements of U, $\mathbf{u}_i = [u_{i1} \dots u_{ik}]^T$ (the soft cluster assignment for sample i), d_{ii} are the diagonal elements of D, $d_i = \frac{1}{\sqrt{d_{ii}}}$, and $k_{ij} = k(W^T x_i, W^T x_j)$. The first term in 4, $\text{tr}(U^T D^{-1/2} K D^{-1/2} U) = \text{tr}(D^{-1/2} U U^T D^{-1/2} K)$ using the cyclic property of the trace function. The elements of $(D^{-1/2} U U^T D^{-1/2})_{ij} = \frac{\mathbf{u}_i^T \mathbf{u}_j}{d_i d_j}$. Let A be an arbitrary matrix, $\text{tr}(AK) = \sum_{ij} a_{ij} k_{ji}$. Since our kernel matrix K is symmetric, the first term can be expressed as $\sum_{ij} \frac{\mathbf{u}_i^T \mathbf{u}_j}{d_i d_j} k_{ij}$. Let K_Y be the kernel for the label matrix Y (in our experiments, we used a linear kernel). The second term can be expressed as $\text{tr}(K_Y HKH) = \text{tr}(HK_Y HK) = \sum_{ij} \tilde{y}_{ij} k_{ij}$, where $\tilde{y}_{ij}$ are the elements of $HK_Y H$. Thus, we arrive at formulation 5. The objective becomes a linear combination of kernel functions with coefficients γ_{ij}. Basically, γ_{ij} is the normalized embedding of this alternative clustering minus the labeling in other clustering views. The labeling kernel in other views is used to penalize the dependence on previous clustering views.

With U fixed, optimizing the objective with respect to W is a nonlinear optimization problem with orthonormality constraints. We utilize a dimension growth algorithm, introduced in [29], to optimize for W. First, we set the dimensionality of the subspace to be one, and we let w_1 denote a vector spanning that subspace. We use gradient ascent to optimize w_1, where w_1 is initialized by random projection and normalized to have unit norm. We then increase the dimensionality by one and optimize for w_2. w_2 is initialized by random projection, then projected to the space orthogonal to w_1, and finally normalized to have unit norm. We decompose the gradient of w_2 into two parts,

$$\nabla f = \nabla f_{proj} + \nabla f_{\perp}, \tag{6}$$

where ∇f_{proj} is the projection of ∇f onto the space spanned by w_1 and w_2, and $\nabla f_{\perp}$ is the component orthogonal to ∇f_{proj} ($\nabla f_{proj} \perp \nabla f_{\perp}$). $\nabla f_{\perp}$ is normalized to have norm one. We update w_2 according to the following equation:

$$w_{2,new} = \sqrt{1 - \alpha^2} w_{2,old} + \alpha \nabla f_{\perp}. \tag{7}$$

The step size $\alpha > 0$ is set by line search satisfying the two Wolfe conditions [30]

$$\begin{aligned} \Phi(\alpha) &\geq \Phi(0) + a_1 \alpha \Phi'(0), \\ \|\Phi'(\alpha)\| &\leq a_2 \Phi'(0), \end{aligned} \tag{8}$$

where $\Phi(\cdot)$ is a univariate function

$$\Phi(\alpha) = \sum_{ij} \gamma_{ij} k\big(\big(\sqrt{1 - \alpha^2} w_{2,old} + \alpha \nabla f_{\perp} \big)^T x_i, \big(\sqrt{1 - \alpha^2} w_{2,old} + \alpha \nabla f_{\perp} \big)^T x_j \big),$$

and where $0 < a_1 < 1$ and $0 < a_2 < 1$. We repeat Equation (7) until convergence. Because w_1 and w_2 are initially set to be orthonormal and w_2 is updated according to the above equation, w_2 and w_1 will remain orthonormal. w_j is optimized in the same way; it is updated orthogonal to $w_1, w_2, \dots, w_{j-1}$. Once we have the desired number of dimensions q, we repeat Equation (7) for each w_j, $j = 1, \dots, q$ until convergence.

We repeat these two steps iteratively until convergence. After convergence, we obtain the discrete clustering by using K-means in the embedding space U. Algorithm 1 provides a summary of our approach. We call this general approach *kernel dimension alternative clustering* (KDAC).

2.2.1 Example Kernels

To make the steps concrete, we provide details for the examples of Gaussian and polynomial kernels.

For the *Gaussian kernel*, we can rewrite the optimization problem in Step 2 as follows:

$$\max_{W} \quad \sum_{ij} \gamma_{ij} \exp\left(-\frac{\Delta x_{ij}^T W W^T \Delta x_{ij}}{2\sigma^2}\right) \quad \text{s.t.} \quad W^T W = I, \tag{9}$$

where Δx_{ij} is the vector $x_i - x_j$, and $\Delta x_{ij}^T W W^T \Delta x_{ij}$ is the l_2-norm of $x_i - x_j$ in the subspace defined by W. This objective function can be expressed as

$$\max_{W} \quad \sum_{ij} \gamma_{ij} \exp\left(-\frac{\mathrm{tr}(W^T \Delta x_{ij} \Delta x_{ij}^T W)}{2\sigma^2}\right) \quad \text{s.t.} \quad W^T W = I, \tag{10}$$

or

$$\max_{W} \quad \sum_{ij} \gamma_{ij} \exp\left(-\frac{w_1^T A_{ij} w_1 + w_2^T A_{ij} w_2 + \cdots}{2\sigma^2}\right) \quad \text{s.t.} \quad W^T W = I, \tag{11}$$

where w_l is the lth column of W, and A_{ij} is the $d \times d$ positive semidefinite matrix $\Delta x_{ij} \Delta x_{ij}^T$. In this step, we assume γ_{ij} is fixed. Note that $w_l^T A_{ij} w_l$ is a convex function. Thus, the summation of $w_l^T A_{ij} w_l$ is convex. The function $\exp(-y)$ is decreasing, so $\exp(-\frac{w_1^T A_{ij} w_1 + w_2^T A_{ij} w_2 + \cdots}{2\sigma^2})$ is a concave function. The components w_l are mutually orthogonal and have norm one. Unfortunately, due to the orthonormality constraints on W, the optimization problem is not concave. The dimension growth algorithm aims to solve this problem. We rewrite the objective as

$$\max_{W} \sum_{ij} \gamma_{ij} \exp\left(-\frac{w_1^T A_{ij} w_1}{2\sigma^2}\right) \exp\left(-\frac{w_2^T A_{ij} w_2}{2\sigma^2}\right) \cdots. \tag{12}$$

With w_1 fixed, we can absorb the term with w_1 into the coefficient. Taking the derivative with respect to w_2, we get

$$\sum_{ij} -\gamma_{ij} \frac{1}{\sigma^2} g(w_1) \exp\left(-\frac{w_2^T A_{ij} w_2}{2\sigma^2}\right) A_{ij} w_2, \tag{13}$$

where $g(w_1)$ is $\exp(-\frac{w_1^T A_{ij} w_1}{2\sigma^2})$.

For the *polynomial kernel* with degree p, we can rewrite the objective as follows:

$$\max_{W} \quad \sum_{ij} \gamma_{ij} \left(\left(W^T x_i\right)^T W^T x_j + 1 \right)^p \quad \text{s.t.} \quad W^T W = I. \tag{14}$$

This is equivalent to the following:

$$\max_{W} \quad \sum_{ij} \gamma_{ij} \left(w_1^T B_{ij} w_1 + \cdots + w_l^T B_{ij} w_l + 1\right)^p \quad \text{s.t.} \quad W^T W = I, \tag{15}$$

where w_l is the lth column of W, and B_{ij} is the $d \times d$ matrix $x_i x_j^T$. In this step, we assume γ_{ij} is fixed. We then apply the dimension growth algorithm to solve this problem. With w_1 fixed, we can absorb the term with $w_1^T B_{ij} w_1$ into the constant in the polynomial kernel. Taking the derivative with respect to w_2, we get

$$\sum_{ij} \gamma_{ij} d \left(w_2^T B_{ij} w_2 + g(w_1) + 1\right)^{p-1} \left(B_{ij} + B_{ij}^T\right) w_2, \tag{16}$$

where $g(w_1)$ is $w_1^T B_{ij} w_1$.

Algorithm 1 Kernel Dimension Alternative Clustering (KDAC)

Input: Data X, existing labeling/s Y, reduced dimension q, cluster number c for non-redundant clustering.
Initialize: $W = I$.
Step 1: Project data on subspaces W, calculate the kernel similarity matrix K and degree matrix D in each subspace, calculate the top c eigenvectors of $D^{-1/2} K D^{-1/2}$ to form matrix U.
Step 2: Given U, update W according to the dimension growth algorithm.
REPEAT steps 1 and 2 until convergence.
K-means Step: Form n samples $y_i \in \mathbb{R}^c$ from rows of U for each view. Cluster the points y_i, $i = 1, \ldots, n$, using K-means into c partitions, $C_1, \ldots, C_c$.
Output: Alternative clustering and transformation matrix W.

2.2.2 Discussion of Convergence

The dimension growth algorithm will converge to a local optimum. The algorithm is based on Equation (7), with $\alpha > 0$ satisfying the two Wolfe conditions. We have $\langle \nabla f_\perp, \nabla f(w) \rangle = \langle \nabla f_\perp, \nabla f_\perp + \nabla f_{proj} \rangle = \langle \nabla f_\perp, \nabla f_\perp \rangle \geq 0$, thus $\nabla f_\perp$ is an ascent direction (i.e., it gives $f(w_{new}) > f(w_{old})$). The algorithm will generate a sequence of w with $f(w_n) > f(w_{n-1}) > f(w_{n-2}) \ldots$. The objective function is upper bounded in both steps. In Step 1, the objective is bounded because the eigenvalues of a normalized similarity matrix are bounded. In Step 2, if each element in the kernel similarity matrix is bounded, the objective is bounded. For the Gaussian kernel, $\exp(-\frac{w^T A w}{2\sigma^2}) \leq 1$. For the polynomial kernel, using the Cauchy-Schwartz inequality, $(x_i^T W W^T x_j + 1)^p \leq (|x_i^T W W^T x_j| + 1)^p \leq (|W^T x_i \| W^T x_j| + 1)^p \leq (|x_i \| x_j| + 1)^p$. This kernel is then bounded for finite p if each original input x_i is finite. Assuming these conditions hold, the algorithm will converge to a local optimum.

2.2.3 Implementation Details

Our approach is dependent on initialization. Our suggested initialization procedure is to start by setting the kernel similarity based on all the features $W_{(0)} = I$. We then calculate the alternative embedding U using all of the features.

Calculating the kernel similarity matrix K and the eigen-decomposition of K can be time consuming. Suppose we have n instances, the complexity of calculating K itself is $O(n^2)$ and the eigen-decomposition of K has complexity $O(n^3)$. Since the eigenvalues of the kernel similarity matrix drops very fast, we can find low-rank approximations of the kernel similarity matrix with rank s ($s \ll n$). Here, we use incomplete Cholesky decomposition [27] giving us an approximate similarity matrix $\widetilde{K}$. The complexity of calculating this matrix is $O(ns^2)$, where n is the number of data points, s is the size of the Cholesky factor $\widetilde{G}$, where

$\widetilde{K} = \widetilde{G}\widetilde{G}^T$. We set s such that the approximate error is less than $\epsilon = 10^{-4}$. Thus, the complexities of the eigen-decomposition and derivative computations are now $O(ns^2)$ and $O(ns)$. The complexity of the overall algorithm is $O((ns^2 + nsrd)t)$, where d is the reduced dimensionality, r is the number of steps in the gradient ascent in Step 2, and t is the number of overall iterations.

3 Eigen-Decomposition Solution with Linear Kernels

For the special case of linear kernels, the approach leads to an eigenvalue problem that provides us with a global solution. With linear kernels, the objective of spectral clustering is equivalent to the PCA objective if we ignore the normalizing degree matrix. In the linear case, the spectral embedding U is a linear transformation of data $U = XW$. Note that in [31], it has been shown that PCA is the continuous solution to the K-means algorithm. Setting the kernel similarity matrix to $K = XX^T$ and assuming X is zero-centered, the spectral objective term for cluster quality becomes

$$\begin{aligned}\mathrm{tr}(U^T K U) &= \mathrm{tr}(W^T X^T X X^T X W) \\ &= \mathrm{tr}(W^T \Sigma^2 W),\end{aligned}$$

where Σ is the covariance matrix of the data. With a linear kernel, the empirical estimate for HSIC between XW and Y becomes

$$\begin{aligned}\mathrm{HSIC}(XW, Y) &= \mathrm{tr}(K_1 K_2) = \mathrm{tr}(XWW^T X^T Y Y^T) \\ &= \mathrm{tr}(W^T X^T Y Y^T X W),\end{aligned}$$

where K_1 and K_2 are the Gram matrices for the data and labeling respectively. The overall objective is

$$\begin{aligned}\max_{W} \quad & \mathrm{tr}(W^T \Sigma^2 W) - \lambda \mathrm{tr}(W^T X^T Y Y^T X W) \\ \text{s.t.} \quad & W^T W = I.\end{aligned} \tag{17}$$

The solution for W is the first q eigenvectors of matrix $\Sigma^2 - \lambda X^T Y Y^T X$. We then cluster the transformed data $U = XW$ using K-means. A summary of this algorithm is provided in Algorithm 2. We call this approach KDAC_{linear} to emphasize that it is a linear variant of our general framework.

Algorithm 2 Alternative Clustering with Linear Kernels (KDAC_{linear})

Input: Data X, existing cluster labeling/s Y, reduced dimension q, and number of clusters c.
Solve for W**:** Set W equal to the q eigenvectors corresponding to the first q eigenvalues of the matrix $X^T X - \lambda X^T Y Y^T X$.
K-means Step: Cluster the samples in the transformed data, $U = XW$, by K-means.
Output: Alternative clustering solution and transformation matrix W.

4 Eigen-Decomposition Formulation in Finding an Alternative Embedding View

In the previous section, we minimize the dependence between the transformed data and existing labeling $\mathrm{HSIC}(W^T X, Y)$. For alternative clustering, if we are not interested in learning a low-dimensional subspace W, we can simply minimize the embedding and labeling directly using $\mathrm{HSIC}(U, Y)$. In the discrete partitioning step in spectral clustering, we apply K-means in the embedding space U. This makes the assumption that the cluster boundaries in U have a linear structure. We thus find it is reasonable to use the linear kernel for $\mathrm{HSIC}(U, Y)$. HSIC can be expressed as $\mathrm{tr}(UU^T Y Y^T)$. The overall objective becomes

$$\begin{aligned}\max_{U} \quad & \mathrm{tr}(U^T D^{-1/2} K D^{-1/2} U) - \lambda \mathrm{tr}(U^T Y Y^T U) \\ \text{s.t.} \quad & U^T U = I.\end{aligned} \tag{18}$$

Note that K is the Gram matrix defined in all the original features. This is an eigenvalue problem. The solution for the alternative clustering embedding U is then the first q eigenvectors of the matrix $D^{-1/2} K D^{-1/2} - \lambda Y Y^T$. We then cluster in the embedding space U. Algorithm 3 gives a summary of the procedure. We call this approach $\mathrm{KDAC}_{embedding}$ to emphasize that it is an embedding variant of our general model.

Algorithm 3 Alternative Clustering Embedding ($\mathrm{KDAC}_{embedding}$)

Input: Data X, existing cluster labeling/s Y, embedding dimension q, and number of clusters c.
Solve for U**:** Set U equal to the q eigenvectors corresponding to the first q eigenvalues of the matrix $D^{-1/2} K D^{-1/2} - \lambda Y Y^T$.
K-means Step: Cluster the samples in the embedding space U by K-means.
Output: Alternative clustering solution.

5 Comparative Analysis of Various Alternative Clustering Scenarios

In this section, we present some experiments with synthetic data that explore some of the dimensions of variation of solutions to the multiple alternative non-redundant clustering problem. In particular, we compare and contrast (1) simultaneous and iterative approaches, (2) methods that search for clusterings in the original space versus those that search in reduced dimensionality subspaces, and (3) methods that are based on sufficient dimensionality reduction versus based on embedding.

5.1 Simultaneous versus Iterative Approaches

Given data, one can discover multiple clusterings either simultaneously or iteratively. In general, the choice of approach is driven by the exploratory analysis situation the data analyst faces. If the data analyst has some existing clustering solutions that are known and would like to search for alternative solutions, then iterative approaches are appropriate. Otherwise, either approach is applicable.

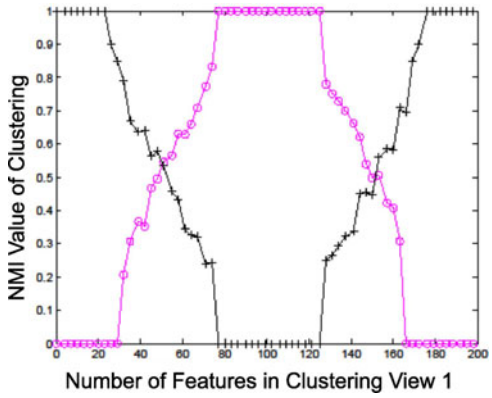


Fig. 2. The NMI value of the results: the simultaneous method with view 1 is shown in magenta with "circle" sign; and the iterative method with view 1 is shown in black with "plus" sign.

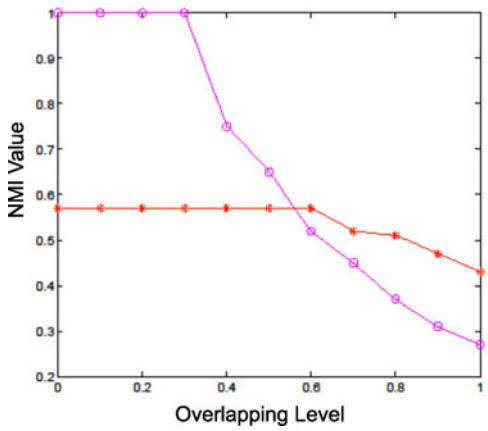


Fig. 3. The NMI values with respect to view 1 for different overlapping level of feature subspaces: our method (KDAC) is shown in magenta with "circle" sign; and decorrelated kmeans is shown in red with "asterisk" sign.

Generally, one can convert a simultaneous approach to an iterative one by setting the number of views to t and setting the $t-1$ clustering views equal to the previously found or existing clustering solutions.

Besides mechanistic differences, these two approaches lead to different biases in the cluster structures they discover. Because simultaneous approaches discover multiple clusterings at the same time, they provide a global perspective to optimizing the multiple alternatives and tend to prefer balanced clusterings. In contrast, because iterative methods sequentially find clustering solutions which is essentially a sequential search strategy, they tend to greedily favor and find the more dominant cluster structure first. We test this intuition by designing a synthetic experiment as follows. We generate a synthetic data with 200 equal variance features and two clustering views based on two feature subsets. Each clustering view has three Gaussian clusters with identity covariances. We associate "strength" or "dominance" of a clustering by the number of features in this clustering view. We generated several data sets by varying the number of features utilized to create each clustering view. We then compare the results of a simultaneous version of our multiple clustering approach [23] and the iterative algorithm (KDAC) described in this paper (Algorithm 1).

In Fig. 2, we plot the normalized mutual information (NMI) [32] between the clustering results for the simultaneous method (magenta line with circle symbol) and the iterative method (black line with plus symbol) compared to the true cluster labeling in view 1 as the number of features in view 1 is increased. Let U be the alternative clustering and L be the known labels, $NMI(L,U) = \frac{I(L,U)}{\sqrt{H(L)H(U)}}$, where $I(L,U)$ is the mutual information between L and U, and $H(L)$ and $H(U)$ are the entropies of L and U respectively. The higher the NMI value, the more similar the clustering and the labels are. In Fig. 2, we observe that when the underlying clustering structures are more balanced, i.e., when the number of features in each view are equal, the simultaneous approach works well as reflected by high NMI values in the middle portion of the x-axis. On the other hand, the iterative approach works well when the underlying cluster structures are imbalanced (i.e., one view is more dominant than the other) as reflected by high NMI values for the black curve in the extreme (left or right) portions of the x-axis. In summary, simultaneous approaches provide a global perspective and works well when the views are balanced; whereas, iterative approaches are greedy and finds the more dominant cluster structures first. Iterative methods work well when the underlying clustering views are imbalanced.

5.2 Importance of Dimensionality Reduction

[5], [13], [10] and [12] explore alternative clustering solutions using all of the original features. In this paper, we allow the clustering solutions in different alternative views to reside in different low-dimensional subspaces. In this section, we investigate when incorporating dimensionality reduction in searching for alternative clustering views is beneficial and when it is not.

We generate a synthetic experiment as follows. Let the original dimension be 100. We generate two clustering views from two feature subspaces and create multiple such synthetic data sets by varying the overlapping levels between the two feature subspaces. An overlap of zero means that the two views have disjoint features with 50 features each; overlap of 0.3 means that 30 features are common to the clustering views.

In Fig. 3, we show the NMI results of our method with view 1 in magenta with circle symbol and the NMI results for decorrelated K-means [12] in red with asterisk symbol. The results confirm that dimensionality reduction helps (and indeed, importantly, leads to an increase in average NMI of different clusterings from less than 0.6 to 1) when the clustering views reside in different subspaces. When the overlap in features is high, dimensionality reduction is not needed.

5.3 Embedding versus Sufficient Dimensionality Reduced Subspace

We can reduce the dimension by either finding a lower dimensional embedding U (Algorithm 3, KDAC$_{embedding}$) or a sufficient lower dimensional subspace W (Algorithm 1, KDAC). The first approach, KDAC$_{embedding}$, has the advantage of leading to an eigen-decomposition whose relaxed clustering/embedding solution is guaranteed to achieve the

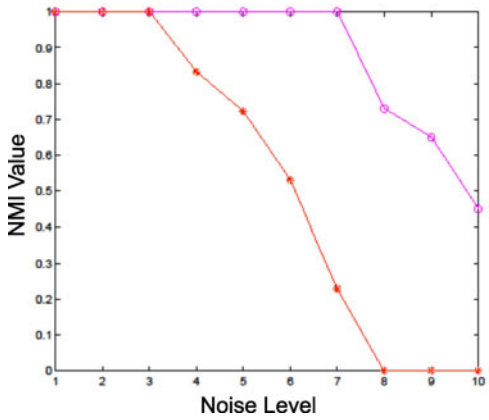


Fig. 4. NMI values with view 2 as the noise level is increased: alternative sufficient dimensionality reduced subspace, KDAC, is shown in magenta with "circle" symbol; and alternative embedding, KDAC$_{embedding}$ is shown in red with "asterisk" symbol.

global optimum. However, the embedding method is not as robust to noise compared to finding a sufficient lower dimensional subspace W because KDAC also learns the kernel by learning the low-dimensional subspace in which the cluster embedding lies.

To demonstrate this, we generate synthetic data as follows. We generated two clustering views from subspaces, s_1 and s_2, both with underlying dimensionality two. We then transform this underlying subspace, $s = [s_1, s_2]$, to a higher dimensional space of dimensionality ten by random projection R and adding uncorrelated Gaussian noise: $Rs + noise$. Assume the cluster labeling in view 1 (s_1) is given. We discover the second clustering in the data. We generate several data sets as we vary the noise level. In Fig. 4, we plot the NMI results of alternative sufficient dimensionality reduced subspace, KDAC, with respect to view 2 (in magenta with circle symbol), and the NMI results for alternative embedding, KDAC$_{embedding}$, with view 2 (in red). We observe that as the noise level increases, the NMI value of the embedding method decreases faster than the sufficient dimensionality reduction method.

6 Experiments

In this section, we perform experiments on both synthetic and real data to investigate whether our algorithm gives reasonable alternative clustering solutions with high quality. We first test our method on synthetic data sets in Section 6.1 to get a better understanding of the method. Then we test our method on real data in Section 6.2. In particular, we perform experiments on a face image, two image segmentation tasks and text data. We compare our kernel dimension reduction alternative clustering (KDAC) method and two variations (KDAC$_{linear}$ and KDAC$_{embedding}$) against five recently proposed algorithms for alternative clustering: the conditional information bottleneck (CIB) approach [4], COALA [5], orthogonal projection clustering (OP) [6], the constrained optimization of the Kullback-Leibler divergence (cons-KL) approach [9] and CAMI [13]. Finally, in Section 6.3 we test our method's ability to iteratively find more than two clustering solutions on a synthetic data set and a machine sound data set.

Evaluation Measures. The evaluation of clustering results is a challenging problem. Two types of criteria are generally used for measuring cluster quality: external and internal criteria. External criteria measure the agreement between the clustering result and an external input (usually from known labels). Internal criteria, on the other hand, measure quality based on characteristics of the data and the partitioning result. Here, we evaluate our results on both type of criteria. The external criteria serve two purposes in our setting. The first purpose is to measure the dissimilarity with the existing clustering. The more dissimilar, the better the result is. If the data set has two labelings, we consider the second labeling as the alternative labeling. The second purpose is to measure the similarity between the alternative clustering solution and the second labeling. The higher the similarity, the better the result is. We utilize the normalized mutual information suggested in [32] and defined earlier in Section 5.1 as our external measure.

Sometimes there is no external labeling available and in this case internal criteria are used to evaluate the clustering results by measuring cluster quality. However, clustering algorithms generally optimize some internal criteria; thus, internal criteria are necessarily biased to favor those algorithms with the same objective. With this caveat in mind, we nonetheless present these measures to give at least some indication of cluster quality. We utilize two standard measures: mean-squared-error (MSE) and Dunn index (DI). MSE is a widely used criterion for clustering which measures the error of instances to its corresponding cluster centroid. It is defined as $MSE = \frac{1}{n}\sum_{j=1}^{c}\sum_{x\in C_j}\|x-\mu_j\|^2$, where n is the number of instances and μ_j is the centroid of cluster C_j. To make this criterion suitable for data with nonlinear structure, we also employ a kernel version of MSE, which is defined as $MSE_{kernel} = \frac{1}{n}\sum_{j=1}^{c}\sum_{x\in C_j}\|(\phi(x)-\mu_j)\|^2$, where $\mu_j = \frac{1}{n_j}\sum_{x\in C_j}\phi(x)$ is the mean in kernel space, $\phi(x)$ transforms x to some nonlinear space and n_j is the cardinality of clustering C_j. Expanding the formula, the inner product in the kernel version of MSE can be calculated using kernel functions in the input space. Lower MSE values mean better cluster quality. The Dunn index is a ratio of the between-cluster separation normalized by the within-cluster distance. For a clustering $C = \{c_1, \ldots, c_k\}$, where $\delta : c \times c \rightarrow \mathbb{R}^+$ is a cluster-to-cluster distance and $\Delta : c \rightarrow \mathbb{R}^+$ is a cluster diameter measure, the Dunn index is $DI(C) = \frac{min_{i\neq j}\{\delta(c_i,c_j)\}}{max_{1\leq l\leq k}\{\Delta(c_l)\}}$. Higher values of this index indicate higher quality. In all our experiments, the internal criteria are calculated using all the original features.

6.1 Experiments on Synthetic Data

To get a better understanding of our method and test its applicability, we first perform our approach on two synthetic data sets. The first synthetic data set has two possible clustering structures. The second synthetic data set has two clusters with complex shapes. In this data set, we investigate whether or not our approach can discover alternative nonlinear structures. We apply the Gaussian kernel for KDAC and KDAC$_{embedding}$ on both data sets.

The first synthetic data set is generated from six features with 600 instances (see Fig. 5). Three Gaussian clusters are generated in features $\{F_1, F_2\}$ with 200, 200 and 200 instances

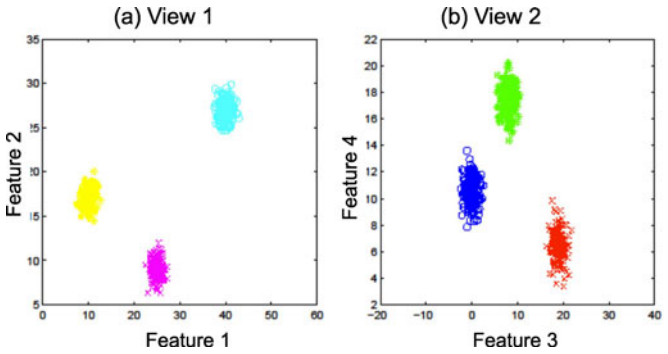


Fig. 5. Two alternative cluster labeling interpretations for synthetic data 1. The color and symbol indicate the labeling.

in each cluster. The other three Gaussian clusters are generated in features $\{F_3, F_4\}$ with 200, 200 and 200 instances respectively. The other two features are Gaussian noise with variance equal to 10. We assume the first clustering view is the existing clustering solution. Given this current labeling, we apply the different methods to find the alternative clustering. The second synthetic data set is generated from four dimensions with 600 instances. Two nonlinear structures reside in the two subspaces with two and three clusters respectively, as shown in Fig. 6. The color and symbol of points indicate cluster labelings. We use the labeling in features $\{F_1, F_2\}$ as the given existing clustering. Fig. 7 shows the NMI results with the existing labeling (view 1) and the NMI with the alternative labeling (view 2) for the different methods. Low NMI with the existing (left) and high NMI with the new clustering (right) are desired. For synthetic data 1, all the methods work well, with KDAC and KDAC$_{linear}$ obtaining the best results. For synthetic data 2, KDAC and KDAC$_{embedding}$ perform much better than the other methods because of their ability to capture nonlinear dependence.

6.2 Experiments on Real-World Data

We now test our method on four real-world data sets to see whether our method can find meaningful alternative clusterings. We select data that have multiple possible partitionings. In particular, we test our method on a face image data set, two image segmentation data sets and a web-page text data set. We use a Gaussian kernel for the image data sets, and a polynomial kernel with degree $p = 3$ for the text data.

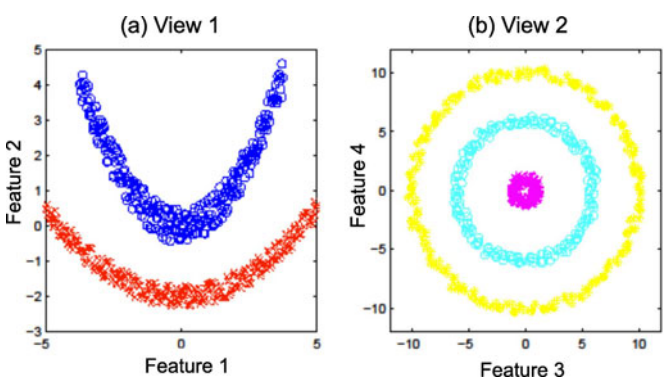


Fig. 6. Two alternative cluster labeling interpretations for synthetic data 2. The color and symbol indicate the labeling.

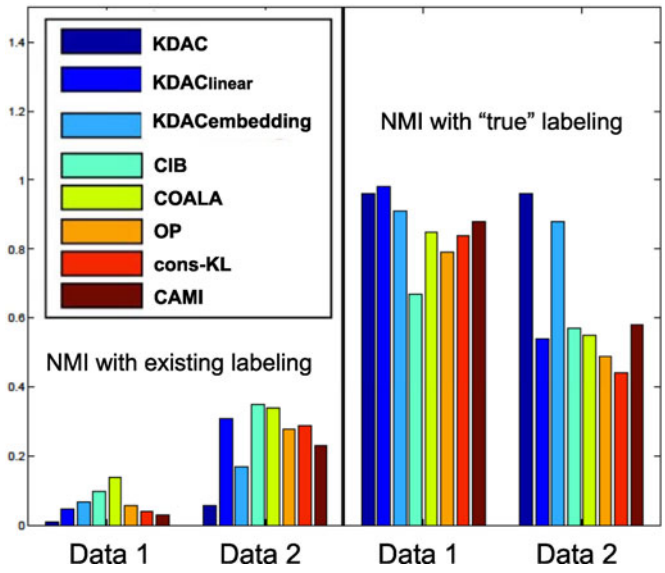


Fig. 7. Results on synthetic data.

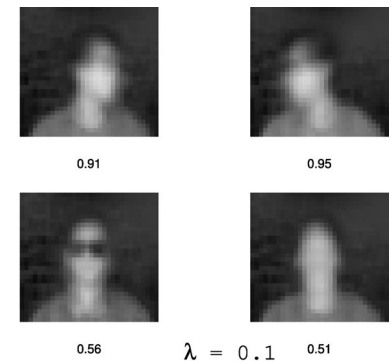
6.2.1 Experiments on Face Data

The face data set from UCI KDD repository [33] consists of 640 face images of 20 people taken at varying poses (straight, left, right, up), expressions (neutral, happy, sad, angry), eyes (wearing sunglasses or not). The two dominant clusterings of this face data set are: the identity of the person and their pose. Each person has 32 images with four equally distributed poses. The image resolution is 32×30. In summary, this results in a data set with 640 instances and 960 features. Each feature represents a pixel value. Given one existing clustering solution, we test whether our method can find the alternative clustering. In the experiment, we use the person's identity as the existing clustering solution and pose as the alternative clustering.

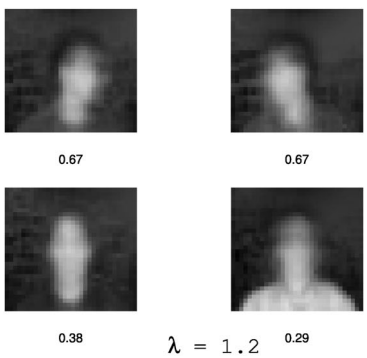
Table 1 shows the results for all methods based on the different evaluation measures: NMI$_e$(↓) measures similarity with the existing clustering; NMI$_a$(↑) measures similarity with the alternative labeling; and MSE(↓), MSE$_G$(↓), MSE$_P$ (↓), and DI(↑) are internal criteria that measure the cluster quality of the alternative clusters. MSE$_G$ stands for MSE with the Gaussian kernel and MSE$_P$ with the polynomial kernel. (↓) reminds us that lower values are desired and, similarly, higher values for the (↑) are desired. We highlight the best values in bold font. The results show that our methods, KDAC and variations, successfully find alternative clusterings of the face data with higher NMI$_a$(↑) values and the best cluster internal criteria values compared to the

TABLE 1
Results on the Face Data

	NMI$_e$↓	NMI$_a$↑	DI↑	MSE↓	MSE$_G$↓
KDAC	**0.037**	**0.482**	**0.738**	29.32	**0.535**
KDAC$_{linear}$	0.043	0.451	0.720	**29.22**	0.557
KDAC$_{embedding}$	0.083	0.478	0.703	30.22	0.542
CIB	0.126	0.372	0.710	30.39	0.603
COALA	0.073	0.424	0.704	30.38	0.589
OP	0.051	0.442	0.722	29.36	0.576
cons-KL	0.082	0.451	0.707	29.39	0.565
CAMI	0.061	0.439	0.712	30.19	0.574



(a) Mean faces of the alternative clustering discovered with $\lambda = 0.1$.



(b) Mean faces of the alternative clustering discovered with $\lambda = 1.2$.

Fig. 8. Mean face images for each cluster discovered by KDAC.

other five methods. We also have the lowest $\text{NMI}_e(\downarrow)$ similarity with the existing clustering.

Given face identity as the current clustering, our approach is able to find the alternative clustering based on pose as shown in Fig. 8. This figure shows the mean of each cluster discovered and the number below each image is the percentage of time this pose appears in this cluster. We show results for two different values of the control parameter λ. In our approach, KDAC, the parameter λ allows us to control the tradeoff between cluster quality and alternativeness. Cons-KL [9] also has a parameter that controls the tradeoff between quality and alternativeness. In practice during interactive exploration, the control parameter can be tuned until a desired alternative solution is provided. In Fig. 9, we compare KDAC and cons-KL in terms of both MSE (to measure cluster quality) and NMI with the existing clustering (to measure novelty). The plot shows that KDAC is able to discover better quality clusters (smaller MSE) than cons-KL for the same level of alternativeness (NMI).

6.2.2 Experiments on Image Segmentation

Image data usually have a rich structure that can be interpreted in several ways. Figs. 10 and 11 show the Escher fish and butterfly data sets respectively. Each pixel is treated as a sample and there are three features corresponding to the RGB color values of these pixels. The goal is to segment the image into foreground and background (two clusters). Note that there are several ways to segment these images. There is no external labeling available for this data. Column (b) of Figs. 10 and 11 show the result of spectral clustering which we set as our existing clustering. We compare the different methods in terms of their $\text{NMI}_e(\downarrow)$ values with the existing clustering and the quality of their alternative clustering solution based on internal criterion measures, MSE $(\downarrow)$, $\text{MSE}_G(\downarrow)$, and DI$(\uparrow)$.

Tables 2 and 3 show the results for all methods based on the different evaluation measures on the fish and butterfly data respectively. The results show that our methods, KDAC and variations, outperform the other methods in terms of finding an alternative clustering that is most dissimilar with the existing clustering in terms of NMI and the best in terms of cluster quality measures DI, MSE and MSE_G.

In Figs. 10 and 11, columns (c) and (d) show two alternative clustering results based on different values of the

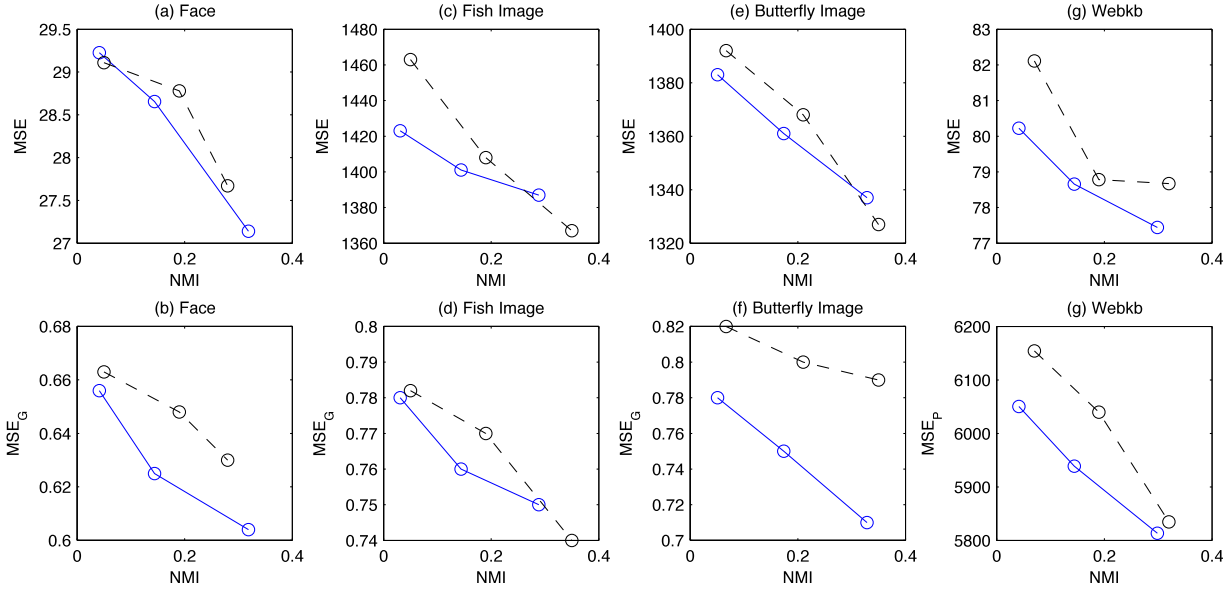


Fig. 9. NMI with existing labeling versus mean-squared error. Blue solid curves are KDAC results and black dashed curves are cons-KL results.

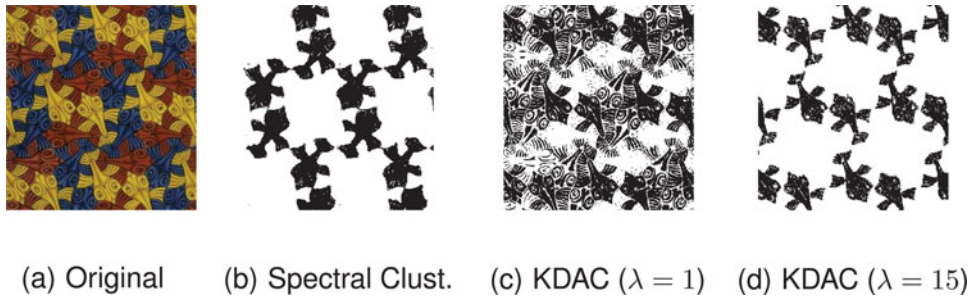


Fig. 10. Escher fish image data.

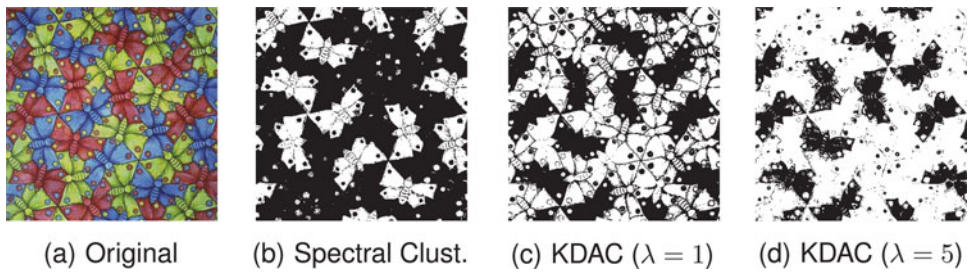


Fig. 11. Escher butterfly image data.

tradeoff parameter λ. Column (c) images reveal alternatives that capture a more textured pattern compared to the existing one in column (b). These segmentation results are more novel with respect to the existing labeling (low NMI_e) but have low cluster quality (high MSE). Column (d) images provide alternative segmentations that capture the fish or butterflies from colors other than the existing one in column (b). These segmentation results are less novel with respect to the existing labeling (high NMI_e) but have high cluster quality (low MSE). In Fig. 9, we compare KDAC and cons-KL in terms of both MSE (to measure cluster quality) and NMI with the existing clustering (to measure novelty). The plot shows that KDAC is able to discover better quality clusters (smaller MSE and MSE_G respectively) than cons-KL for the same level of alternativeness (NMI) for the fish and butterfly data; but at higher NMI for the linear case and for the fish data, cons-KL starts to have better MSE values.

6.2.3 Experiments on WebKB Text Data

The WebKB data set [34] is a sub-sample of $1,041$ html documents from four universities: Cornell University, University of Texas, Austin, University of Washington and University of Wisconsin, Madison. These web pages can be alternatively labelled as from four topics: course, faculty, project and student. We preprocessed the data by removing the rare words, stop words, and words with low variances, retaining 350 words. We use the university information as the existing clustering and the web page's topic as the alternative clustering. Results are shown in Table 4. From Table 4, we see that our algorithm with the polynomial kernel (KDAC) obtains the highest NMI with the alternative labeling (topics) and lowest NMI with the existing labeling (universities).

6.2.4 Summary of Results on Real-World Data

The results on face, fish, butterfly, and WebKB data show that the proposed approach, KDAC, consistently out-performs the competing methods. This is because KDAC provides a more flexible model, able to capture nonlinear dependence structures nonparametrically. In contrast, OP only considers linear independence; CAMI and cons-KL assume Gaussian mixture models; COALA utilizes average linkage to measure similarity/dissimilarity between clusters; and although CIB applies conditional mutual information (which can model nonlinear dependencies), it needs to learn and make parametric assumptions regarding the joint distributions.

6.3 Multiple Alternative Clustering Solutions

In this section, we explore an application of our method to a multiple iterative discovery process in which one finds

TABLE 2
Results for the Escher Fish Data

	$\text{NMI}_e(\downarrow)$	DI(↑)	MSE(↓)	$\text{MSE}_G(\downarrow)$
KDAC	**0.031**	0.885	1431	**0.783**
KDAC_{linear}	0.131	**0.903**	**1423**	0.793
$\text{KDAC}_{embedding}$	0.113	0.823	1583	0.813
CIB	0.143	0.744	1534	0.801
COALA	0.176	0.738	1573	0.823
OP	0.230	0.892	1489	0.893
cons-KL	0.239	0.801	1467	0.795
CAMI	0.241	0.751	1511	0.815

TABLE 3
Results for the Escher Butterfly Data

	$\text{NMI}_e(\downarrow)$	DI(↑)	MSE(↓)	$\text{MSE}_G(\downarrow)$
KDAC	**0.072**	**0.830**	1383	**0.723**
KDAC_{linear}	0.137	0.785	**1375**	0.751
$\text{KDAC}_{embedding}$	0.114	0.725	1411	0.841
CIB	0.223	0.754	1438	0.839
COALA	0.276	0.742	1484	0.823
OP	0.230	0.692	1484	0.788
cons-KL	0.279	0.739	1438	0.823
CAMI	0.208	0.776	1411	0.753

TABLE 4
Results for the WebKB Text Data

	$NMI_e\downarrow$	$NMI_a\uparrow$	DI↑	MSE↓	$MSE_P\downarrow$
KDAC	**0.088**	**0.467**	**0.633**	**79.2**	**6023**
$KDAC_{linear}$	0.135	0.307	0.603	85.1	6219
$KDAC_{embedding}$	0.125	0.417	0.607	82.3	6113
CIB	0.137	0.326	0.593	83.5	6139
COALA	0.124	0.373	0.604	81.7	6134
OP	0.142	0.331	0.622	80.4	6192
cons-KL	0.131	0.412	0.507	81.3	6213
CAMI	0.115	0.362	0.617	82.5	6145

TABLE 5
Results for the Multiple Iterative Discovery

	Synthetic			Machine Sound		
	NMI1	NMI2	NMI3	NMI1	NMI2	NMI3
KDAC	0.87	0.82	0.76	**0.81**	**0.82**	0.73
$KDAC_l$	**0.94**	**0.90**	**0.91**	0.65	0.64	0.63
$KDAC_e$	0.85	0.81	0.77	0.62	0.72	**0.75**
OP	0.85	0.74	0.82	0.62	0.54	0.58
cons-KL	0.83	0.69	0.79	0.65	0.54	0.61
CAMI	0.81	0.71	0.84	0.73	0.52	0.60

multiple clustering solutions by iteratively computing alternative clusters given one or more previously discovered clustering solutions. Among the competing methods, we compare our method to orthogonal projection clustering [6], cons-KL [9] and CAMI [13]. OP allows iteratively discovering multiple solutions; cons-KL can be adapted to multiple iterative solutions by accumulating constraints generated by all previous clusterings; and CAMI can be adapted by adding penalty for the mutual information of all the previous clustering results in its objective function.

We generate totally unsupervised multiple solutions by first applying spectral clustering to give us the first partitioning P_0, then iteratively apply our nonlinear method to find P_t given all previous clusterings, P_0 to P_{t-1}. Orthogonal projection iteratively generates solutions by first applying PCA followed by K-means to get P_0; then iteratively applies orthogonal projection and PCA followed by K-means in the orthogonal space until the desired number of views. For our linear method, we first apply PCA followed by K-means to get P_0 to give us a fair comparison with OP and to be consistent with being a linear model. In all the methods running PCA, we set the reduced dimensionality so as to retain at least 90 percent of the total variance.

We test these methods on a synthetic data set with three independent cluster labelings and a machine sound data set also with three known labelings. The synthetic data has 100 dimensions with $1{,}000$ instances and three independent clustering solutions. In each feature set, ($F_{(1..30)}$, $F_{(31..60)}$ and $F_{(61..100)}$), random vectors with three Gaussian components are generated. One of the projects in our lab is to classify different machine sounds inside buildings. We have collected 280 machine sounds. We applied fast Fourier transformation (FFT) on this data and selected $1{,}000$ highest points in the frequency domain as our features. In this data, there are three kinds of machine sounds: *pump*, *fan*, *motor*. Each instance of sound can be from one machine, or a mixture of two, or a mixture of three machine types. We set *pump* versus *no pump* as one clustering interpretation; *fan* versus *no fan* as another; and *motor* versus *no motor* as the third labeling.

Table 5 provides the NMI results of these two methods on each of the possible labelings for both data sets. The higher the NMI values the better. The results show that on both data sets our algorithms are able to find clustering solutions that match in terms of NMI with the different labelings better than competing methods. $KDAC_{linear}$ did much better than KDAC on the synthetic data because this data only requires linear relationships. Both $KDAC_{linear}$ and orthogonal projection clustering only take linear relationships into account. However, $KDAC_{linear}$ does better because it considers both novelty and quality. Orthogonal projection only accounts for novelty. On the machine sound data, KDAC obtained the best match in terms of NMI because this data has nonlinear structure and KDAC takes nonlinear dependencies into account. Moreover, KDAC does not need to estimate probability distributions or make restrictive assumptions regarding these distributions.

7 Conclusions

We have introduced a new methodology for allowing a user to iteratively discover alternative clustering solutions given previously discovered clustering structures for exploratory data analysis. In finding alternative solutions, it is important to find solutions that are both novel and of good cluster quality. Our approach provides a flexible model that can discover alternative clusters with complex shapes and simultaneously learns the linear subspace in which the clustering resides. These clusters are as independent as possible from the previously learned solutions. We achieve this by utilizing a kernel dependence criterion for assessing cluster quality and similarity/dissimilarity between clustering solutions. Moreover, for the special case of a linear kernel or when we only search for an alternative embedding, we obtain an eigenvalue problem. Our experiments on both synthetic and real data show that our algorithm outperforms competing alternative clustering algorithms.

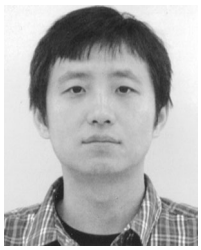
Acknowledgments

This work was supported by US National Science Foundation NSF IIS-0915910 and by the Office of Naval Research under contract/grant number N00014-11-1-0688.

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] J.A. Hartigan, "Statistical Theory in Clustering," *J. Classification*, vol. 2, pp. 63-76, 1985.
[3] A. Gretton, O. Bousquet, A. Smola, and B. Schölkopf, "Measuring Statistical Dependence with Hilbert-Schmidt Norms," *Proc. Int'l Conf. Algorithmic Learning Theory*, pp. 63-77, 2005.
[4] D. Gondek and T. Hofmann, "Non-Redundant Data Clustering," *Proc. IEEE Int'l Conf. Data Mining*, pp. 75-82, 2004.
[5] E. Bae and J. Bailey, "COALA: A Novel Approach for the Extraction of an Alternate Clustering of High Quality and High Dissimilarity," *Proc. IEEE Int'l Conf. Data Mining*, pp. 53-62, 2006.
[6] Y. Cui, X.Z. Fern, and J. Dy, "Non-Redundant Multi-View Clustering via Orthogonalization," *Proc. Seventh IEEE Conf. Data Mining (ICDM '07)*, pp. 133-142, 2007.
[7] Y. Cui, X.Z. Fern, and J.G. Dy, "Learning Multiple Nonredundant Clusterings," *ACM Trans. on Knowledge Discovery from Data*, vol. 4, no. 3, Article 15, 2010.
[8] Z.J. Qi and I. Davidson, "Finding Alternative Clusterings Using Constraints," *Proc. IEEE Int'l Conf. Data Mining*, pp. 773-778, 2008.

[9] Z.J. Qi and I. Davidson, "A Principled and Flexible Framework for Finding Alternative Clusterings," *Proc. ACM Int'l Conf. Knowledge Discovery and Data Mining (KDD)*, pp. 717-726, 2009.
[10] X.H. Dang and J. Bailey, "A Hierarchical Information Theoretic Technique for the Discovery of Non Linear Alternative Clusterings," *Proc. ACM Int'l Conf. Knowledge Discovery and Data Mining (KDD)*, pp. 573-582, 2010.
[11] R. Caruana, M. Elhawary, N. Nguyen, and C. Smith, "Meta Clustering," *Proc. IEEE Sixth Int'l Conf. Data Mining (ICDM '06)*, pp. 107-118, 2006.
[12] P. Jain, R. Meka, and I.S. Dhillon, "Simultaneous Unsupervised Learing of Disparate Clustering," *Proc. SIAM Int'l Conf. Data Mining*, pp. 858-869, 2008.
[13] X.H. Dang and J. Bailey, "Generation of Alternative Clusterings Using the CAMI Approach," *Proc. SIAM Int'l Conf. Data Mining*, pp. 118-129, 2010.
[14] S. Dasgupta and V. Ng, "Mining Clustering Dimensions," *Proc. Int'l Conf. Machine Learning*, pp. 263-270, 2010.
[15] L. Poon, N. Zhang, T. Chen, and Y. Wang, "Variable Selection in Model-Based Clustering: To Do or to Facilitate," *Proc. Int'l Conf. Machine Learning*, pp. 887-894, 2010.
[16] Y. Guan, J.G. Dy, D. Niu, and Z. Ghahramani, "Variational Inference for Nonparametric Multiple Clustering," *1st International Workshop on MultiClust: Discovering, Summarizing and Using Multiple Clusterings at KDD*, 2010.
[17] V. Mansinghka, E. Jonas, C. Petschulat, B. Cronin, P. Shafto, and J. Tenenbaum, "Cross-Categorization: A Method for Discovering Multiple Overlapping Clusterings," *Nonparametric Bayes Workshop at NIPS*, 2009.
[18] D. Niu, J.G. Dy, and Z. Ghahramani, "A Nonparametric Bayesian Model for Multiple Clustering with Overlapping Feature Views," *Proc. Int'l Conf. Artificial Intelligence and Statistics*, vol. 22, pp. 814-822, 2012.
[19] L. Parsons, E. Haque, and H. Liu, "Subspace Clustering for High Dimensional Data: A Review," *ACM SIGKDD Explorations Newsletter*, vol. 6, no. 1, pp. 90-105, 2004.
[20] K. Sim, V. Gopalkrishnan, A. Zimek, and G. Cong, "A Survey on Enhanced Subspace Clustering," *Data Mining and Knowledge Discovery*, vol. 26, no. 2, pp. 332-397, 2013.
[21] G. Moise and J. Sander, "Finding Non-Redundant, Statistically Significant Regions in High Dimensional Data: A Novel Approach to Projected and Subspace Clustering," *Proc. ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining*, pp. 533-541, 2008.
[22] E. Múller, I. Assent, S. Gúnnemann, R. Krieger, and T. Seidl, "Relevant Subspace Clustering: Mining the Most Interesting Non-Redundant Concepts in High Dimensional Data," *Proc. IEEE Int'l Conf. Data Mining*, pp. 377-386, 2009.
[23] D. Niu, J.G. Dy, and M.I. Jordan, "Multiple Non-Redundant Spectral Clustering Views," *Proc. Int'l Conf. Machine Learning*, pp. 831-838, 2010.
[24] A.K. Jain, "Data Clustering: 50 Years Beyond k-Means," *Pattern Recognition Letters*, vol. 31, pp. 651-666, 2010.
[25] U.V. Luxburg, "A Tutorial on Spectral Clustering," *Statistics and Computing*, vol. 5, pp. 395-416, 2007.
[26] A.Y. Ng, M.I. Jordan, and Y. Weiss, "On Spectral Clustering: Analysis and an Algorithm," *Advances in Neural Information Processing Systems*, vol. 14, pp. 849-856, 2001.
[27] F.R. Bach and M.I. Jordan, "Kernel Independent Component Analysis," *J. Machine Learning Research*, vol. 3, pp. 1-48, 2002.
[28] L. Song, A.J. Smola, A. Gretton, and K.M. Borgwardt, "A Dependence Maximization view of Clustering," *Proc. Int'l Conf. Machine Learning*, pp. 815-822, 2007.
[29] D. Niu, J. Dy, and M.I. Jordan, "Dimensionality Reduction for Spectral Clustering," *Proc. Int'l Conf. Artificial Intelligence and Statistics*, pp. 552-560, 2011.
[30] J. Nocedal and S.J. Wright, *Numerical Optimization*. Springer, 2006.
[31] C. Ding and X. He, "K-Means Clustering via Principal Component Analysis," *Proc. Int'l Conf. Machine Learning*, pp. 29-37, 2004.
[32] A. Strehl and J. Ghosh, "Cluster Ensembles—A Knowledge Reuse Framework for Combining Multiple Partitions," *J. Machine Learning Research*, vol. 3, pp. 583-617, 2002.
[33] S.D. Bay, "The UCI KDD Archive," 1999. http://kdd.ics.uci.edu.
[34] CMU, "CMU 4 Universities WebKB data," 1997.



Donglin Niu received the bachelor of science degree in electrical engineering from Nanjing University, China, and his master's of science degree in material science and engineering from the University of California, Irvine. He is currently working toward the PhD degree in electrical and computer engineering at Northeastern University, Boston, under Prof. Dy's supervision. His research interests include machine learning, data mining, and numerical optimization with a research focus on novel data clustering algorithms, high-dimensional data clustering, and multiple data clusterings.

Jennifer G. Dy received the BS degree (Magna Cum Laude) from the Department of Electrical Engineering, University of the Philippines, in 1993, and the MS and Phd degrees from the School of Electrical and Computer Engineering, Purdue University, West Lafayette, Indiana, in 1997 and 2001, respectively. She is an associate professor in the Department of Electrical and Computer Engineering, Northeastern University, Boston, Massachusetts, where she first joined the faculty in 2002. Her research interests include machine learning, data mining and their application to computer vision, health, security, science and engineering, with a particular focus on clustering, multiple clusterings, dimensionality reduction, feature selection and sparse methods, large margin classifiers, learning from the crowds and Bayesian nonparametric models. She received an NSF Career award in 2004. She serves as an action editor for *Machine Learning*, an editorial board member for the *Journal of Machine Learning Research*, organizing/senior/program committee member for ICML, ACM SIGKDD, AAAI, IJCAI, AISTATS and SIAM SDM, and program chair for SIAM SDM 2013.

Michael I. Jordan received the master's in mathematics from Arizona State University, and the PhD degree in cognitive science in 1985 from the University of California, San Diego. He is the Pehong Chen Distinguished Professor in the Department of Electrical Engineering and Computer Science and the Department of Statistics at the University of California, Berkeley. He was a professor at MIT from 1988 to 1998. His research in recent years has focused on Bayesian nonparametric analysis, probabilistic graphical models, spectral methods, variational methods, kernel machines and applications to problems in statistical genetics, signal processing, computational biology, information retrieval and natural language processing. He is a member of the National Academy of Sciences, a member of the National Academy of Engineering and a member of the American Academy of Arts and Sciences. He is a Fellow of the American Association for the Advancement of Science. He has been named a Neyman Lecturer and a Medallion Lecturer by the Institute of Mathematical Statistics. He is an Elected Member of the International Institute of Statistics. He is a Fellow of the AAAI, ACM, ASA, CSS, IMS, IEEE, and the SIAM.