

Max-margin Clustering: Detecting Margins from Projections of Points on Lines

Raghuraman Gopalan and Jagan Sankaranarayanan*
Center for Automation Research, University of Maryland
{raghuram, jagan}@umiacs.umd.edu

Abstract

Given a unlabelled set of points $\mathbb{X} \in \mathbb{R}^N$ belonging to k groups, we propose a method to identify cluster assignments that provides maximum separating margin among the clusters. We address this problem by exploiting sparsity in data points inherent to margin regions, which a max-margin classifier would produce under a supervised setting to separate points belonging to different groups. By analyzing the projections of \mathbb{X} on the set of all possible lines L in \mathbb{R}^N , we first establish some basic results that are satisfied only by those line intervals lying outside a cluster, under assumptions of linear separability of clusters and absence of outliers. We then encode these results into a pair-wise similarity measure to determine cluster assignments, where we accommodate non-linearly separable clusters using the kernel trick. We validate our method on several UCI datasets and on some computer vision problems, and empirically show its robustness to outliers, and in cases where the exact number of clusters is not available. The proposed approach offers an improvement in clustering accuracy of about 6% on the average, and up to 15% when compared with several existing methods.

1. Introduction

Unsupervised identification of patterns in data, broadly referred to as clustering, is an important problem that has been extensively studied [9, 11] over the last several decades. Existing approaches can be characterized based on pattern representation, criteria for similarity between patterns, and cost functions that determine the grouping mechanism [14, 15]. The goal of this work is to find maximally separable clusters, given the knowledge of number of clusters, and an appropriate representation of data that depends on the specific application of interest.

There are two broad approaches to this problem, both of which draw inspiration from supervised classification. The first class of methods performs clustering by reducing the original dimensionality of data. Subspace selection is performed using discriminative methods such as linear discriminant analysis (LDA) [11], which starts with random assignments of class labels, or using generative methods

such as principal component analysis (PCA) [11], locally linear embedding (LLE) [23] and Laplacian Eigenmaps [3]. Standard clustering algorithms like K-means [19] and spectral methods [21, 24] are then applied in the resulting subspace to determine the cluster assignments. However, the absence of ‘true’ data labels makes this a chicken-and-egg problem, and there are methods addressing this issue by studying the feedback between subspace selection and clustering (e.g. [8, 29, 30]). The second class of approaches is based on obtaining clusters with maximum separating margins [4, 22, 28], and are primarily motivated by the paradigm of max-margin supervised classifiers, such as support vector machines (SVM) [5]. Most of these methods can be visualized as implicitly running an SVM with different possible label combinations to obtain a final cluster assignment having maximum margin. However, as this process results in a non-convex integer optimization problem, subsequent efforts [26, 27, 31–33] have proposed approximation strategies that obtain a solution in polynomial time.

Contributions: Our approach belongs to the latter category. However, unlike most existing solutions that optimize over all possible cluster assignments, we seek a more basic understanding of the relationship between data points and margins. Since regions corresponding to the separating margins have (ideally) no data points, our goal is to identify these sparse regions by analyzing the projections of unlabelled points $\mathbb{X} \in \mathbb{R}^N$ on the set of all possible lines L in \mathbb{R}^N . In this process,

- We first derive certain properties which the projections of \mathbb{X} on a line interval will satisfy, if and only if that interval lies *outside* of a cluster, under assumptions of linear separability of clusters and absence of outliers;
- We extend these results to define a similarity measure, which computes the probability of finding a margin in the line interval between a pair of points, and use it to perform global clustering. We relax the assumption of linear separability of clusters using kernel methods, and address the problem of outliers through methods that emphasize a balance between cluster sizes.

Outline of the paper: Section 2 studies the properties of projections of data on line intervals for two cluster and multi-cluster cases. Section 3 proposes a method to determine cluster assignments. Section 4 validates the proposed

*Now at NEC Labs.

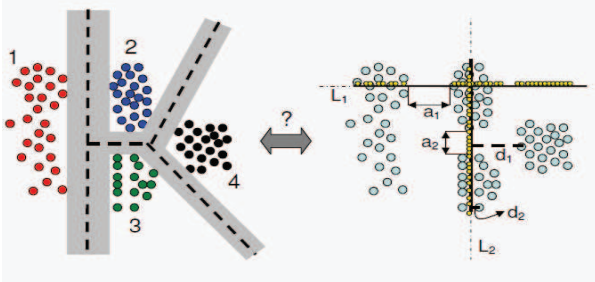


Figure 1. Left: A four class, linearly separable problem with $\mathbb{X} \in \mathbb{R}^2$. With known class labels, a max-margin classifier produces margins (shaded regions) with the separating hyperplanes indicated by the dashed lines. Right: In an unsupervised setting, how to identify these margin regions? Consider two lines L_1 and L_2 , and project \mathbb{X} on them (small yellow dots). Interval a_1 of L_1 has no projected points since it lies in margin region \perp to the hyperplane that separates a cluster from *all other clusters*; whereas interval a_2 of L_2 (whose margin separates *only a pair of clusters*) has projected points from other clusters, with their minimum distance of projection d_1 more than that of d_2 for points projected elsewhere on L_2 . In this work, we study the statistics of location and distance of projections of \mathbb{X} on all lines L , to identify margins and perform clustering. (All figures are best viewed in color.)

method through experiments on standard UCI datasets [10], and on computer vision applications, such as face recognition under illumination variations [12, 25], and 2D shape matching [13, 16]. Section 5 concludes the paper. Figure 1 provides an illustration of our approach.

2. Properties of projection of \mathbb{X} on L

Let the input \mathbb{X} contain a set of M unlabelled data points, $\{x_i\}_{i=1}^M \in \mathbb{R}^N$, belonging to k clusters. For the ease of discussion, we make the following assumptions that will be relaxed later; (i) Points in \mathbb{X} belong to clusters that are (pair-wise) linearly separable in their input space, and (ii) No outliers are present in the data (specific details regarding this assumption will be provided in the following sections). In what follows, we try to detect the presence of margins by studying the patterns in projections of \mathbb{X} on the set of lines L . We will motivate our method by drawing parallels to the supervised max-margin classification scenario.

2.1. Case A: Two clusters

We first study a two-cluster problem, i.e. when $k = 2$. For now, let us assume that the true labels of $x_i, y_i \in \{-1, +1\}$, are available. A max-margin classifier, such as a linear support vector machine (LSVM), produces a decision boundary that optimizes the following objective function,

$$\min_{w,b} \frac{1}{2} \|w\|_2^2, \text{ s.t. } y_i(w^t x_i + b) \geq 1, \forall i = 1 \text{ to } M \quad (1)$$

where $(\cdot)^t$ is the transpose operator. Essentially, the separating hyperplane $S : w^t x + b = 0$, where w is the normal to S , is chosen such that it has a maximum separation of $1/\|w\|_2$ from the tangent of support vectors from either classes given by $H_1 : w^t x + b = 1$, and $H_2 : w^t x + b = -1$, respectively. The margin region bounded by parallel hyperplanes H_1 and H_2 is denoted by M_S , which is characterized by no data points \mathbb{X} , and therefore provides a separating margin

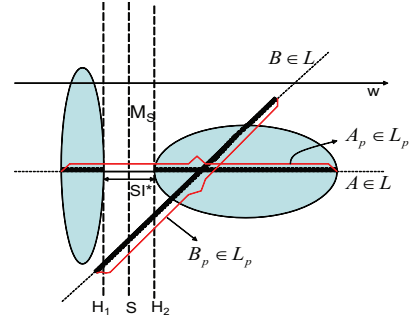


Figure 2. An illustration of projection of points on different line segments. The two clusters are represented by ellipses. Assume that points \mathbb{X} are present everywhere inside the ellipses. When labels of \mathbb{X} are available, S will be the separating hyperplane, and H_1 and H_2 are tangents to support vectors of either classes. M_S denotes the margin region (bounded by H_1 , and H_2). $SI^* = \gamma$ is the margin, and w is the normal to S . In a clustering scenario, where labels of \mathbb{X} are unknown, consider two lines ($A, B \in L$ in \mathbb{R}^2). L_p refers to the segment of L enclosing all projections x_{i_p} (dots in black). It can be seen that on intervals in $A_p \perp S$, there is no x_{i_p} in the region corresponding to margin M_S ; hence, there exist SI^* . For any other line segment not perpendicular to S , say B_p , maximum possible $SI < \gamma$.

of $\gamma = 2/\|w\|_2$ between the two classes. An illustration is provided in Figure 2.

To identify M_S from an unlabelled set of points \mathbb{X} , we now consider the projections¹ of \mathbb{X} onto the set of all lines L in \mathbb{R}^N . Let x_{i_p} denote the location of projection of $x_i \in \mathbb{X}$ on a line. It is not hard to visualize that the projection of \mathbb{X} creates *patterns* on the line, which is shown using black dots in Figure 2 for lines A , and B . Notice that for line A , the projection of \mathbb{X} on them creates two dark patterns with a sparse region in between, which clearly captures the margin between the left and right clusters. On the other hand, line B due to its orientation fails to capture the margin, which makes it unsuitable for our purposes. The intuition behind our algorithm is that if we draw sufficient number of lines between points in \mathbb{X} , we may be able to capture the margins that separate the clusters, which in turn would aid in clustering of \mathbb{X} . Furthermore, we can now discard A and B , in favor of line segments A_p and B_p , which are obtained by walking on those lines and truncating their bounds to lie between the first and last projected points of \mathbb{X} that we encountered. Let the set of these truncated line segments across all L be referred as L_p .

Before analyzing L_p in pursuit of M_S , under the assumption of no outliers in the data, we constrain the maximum margin γ to exist only between points belonging to different clusters, and not otherwise. We now define the following.

Definition Sparsity index of a line segment $z \in L_p$, $SI(z) \in \mathbb{R}$, is the maximum distance² travelled along z where there are no projected points x_{i_p} . Let $SI^* = \max_{z \in L_p} SI(z)$.

¹We are interested in the shortest (perpendicular) projection. Let x_1 and x_2 be any two points through which w passes. To project a new point x_i onto w , we first compute the line passing through x_1 and x_i , say w_{x_1} , and then obtain the location of its projection, $x_{i_p} = x_1 + \frac{w_{x_1} \cdot w}{w \cdot w}$. The distance of projection, d_{i_p} , is given by $\|w_{x_1} - x_{i_p} w\|_2$.

²By distance, we refer to the standard Euclidean norm $\|\cdot\|_2$ between the end points of the interval of z containing no x_{i_p} . Further, we might occasionally drop the argument for $SI(\cdot)$ for sake of simplicity. A glossary of symbols used in the paper is given in the supplementary material.

Proposition 2.1 $SI^* = \gamma$ is realized only by those set of line segments $C \subset L_p$ that are normals to the separating hyperplane S , and the intervals in C where SI^* occurs are those that correspond to the margin region M_S . Furthermore, $\forall C = L_p \setminus C, SI(C) < \gamma$.

Proof Follows directly from (1), provided there exist a unique max-margin separating hyperplane S . ■

Hence in an unsupervised setting, we directly obtain cluster assignments of \mathbb{X} by identifying a line segment (in C) with maximum SI , where the minimum distance between a pair of points belonging to different clusters is SI^* .

2.2. Case B: Multiple clusters

We now consider the general case where the number of clusters $k \geq 2$. We again draw motivation from the supervised max-margin classification problem, for which there are two popular strategies; (i) directly solve for the multi-class problem by optimizing a single objective function (e.g. [7]), and (ii) decompose the problem into one that combines several binary classifiers (e.g. [2]). We will motivate our study using the latter strategy, where we are primarily interested in understanding the information conveyed by a margin, its effect on the distribution of x_{i_p} on w , and the existence of SI^* to perform clustering.

Consider a set of points $\mathbb{X} = \{x_i\}_{i=1}^M$ with known labels $y_i \in \{1, 2, \dots, k\}$ belonging to one of the k linearly separable classes. A supervised classifier produces the final decision boundary \hat{S} by combining several independent binary separating hyperplanes S_i ,

$$\hat{S} = g(S_1, S_2, \dots, S_l) \quad (2)$$

where g is a combination function³ that determines the piece-wise linear boundaries of the decision regions $R_i, i = 1$ to k , belonging to the k classes. An illustration is provided in Figure 3 for a three-class problem.

Notations: Let $\mathbb{X}_i \subset \mathbb{X}$ represent the set of points that are separated by S_i . Let w_i be the normal to S_i , and let the length of the corresponding margins be denoted by γ_i . Let M_{S_i} denote the margin region corresponding to γ_i when S_i is considered in isolation (i.e. a two-class problem with $\mathbb{X} = \mathbb{X}_i$), and let $M'_{S_i} \subseteq M_{S_i}$ denote the bounded margin region in a multi-class setting where S_i independently classifies \mathbb{X}_i (2). For subsequent analysis, we partition the space of \mathbb{X} into two regions; (i) cluster regions $CL = \cup_{i=1}^k CL_i$, where CL_i is the convex hull of all points belonging to the i^{th} cluster, and (ii) non-cluster regions CL' that include $\cup_{i=1}^k M'_{S_i}$ pertaining to margins, and T comprising of $\cup_{i=1}^k R_i \setminus CL_i$, and regions where more than one S_i is involved in decision making. Figure 3 illustrates this for $k = 3$.

³The value of l depends on the type of binary classifiers used, for instance, one-vs-all or one-vs-one, and the mode of combination g . The maximum number of such classifiers, l' , is therefore k for one-vs-all, and $\binom{k}{2}$ for one-vs-one. Since not all hyperplanes might contribute to decision making, $l \leq l'$.

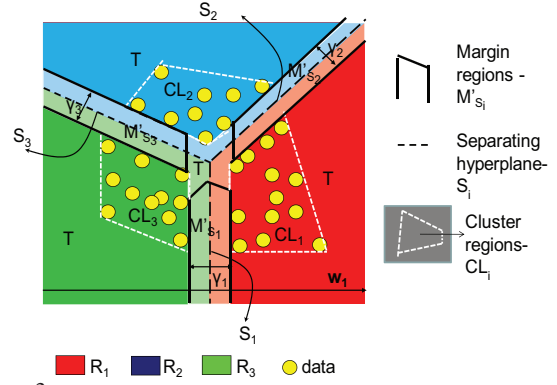


Figure 3. Partitioning the space of \mathbb{X} into different regions. The data \mathbb{X} , shown in yellow circles, belong to three linearly separable clusters ($k = 3$). With known class labels, a supervised classifier \hat{S} produces decision regions $R_i, i = 1$ to 3 belonging to the three classes, shown in red, blue, and green respectively. The margins M'_{S_i} are bounded by solid black lines, with their corresponding margins denoted by γ_i . The separating hyperplanes S_i are given in black dashed lines. We now divide the space of \mathbb{X} into, (i) cluster regions CL_i in white dotted lines, and (ii) non-cluster regions that comprise of margin regions M'_{S_i} and T .

To study the validity of Proposition 2.1 in clustering unlabelled \mathbb{X} belonging to multiple linearly separable groups, we first seek to understand the interference of $\mathbb{X}'_i = \mathbb{X} \setminus \mathbb{X}_i$ on the pair of clusters an S_i separates. To visualize what we mean by this, consider the line interval $a_2 \in L_p$ in Figure 1 that lies in a margin region perpendicular to S_i which separates \mathbb{X}_i belonging to group 2 and 3. Although a_2 does not contain any points from \mathbb{X}_i , many points \mathbb{X}'_i belonging to group 1 and 4 get projected on a_2 . Therefore, we first analyze the relevance of SI^* for a multi-cluster problem.

2.2.1 Existence of SI^* - Information conveyed by x_{i_p}

Instead of analyzing the projections of \mathbb{X} directly on L_p , we consider the set of all continuous intervals that are contained in L_p . Let $Int = \{Int^{CL}\} \cup \{Int^{CL'}\}$ be a set, such that Int^{CL} denotes intervals within the cluster regions $CL_i, \forall i = 1$ to k , and $Int^{CL'}$ denotes those outside the cluster. For example, the line segment $A_p \in L_p$ in Figure 2 has Int^{CL} corresponding to its intervals within the ellipses, and $Int^{CL'}$ corresponding to those in M_S . We now analyze the existence of SI^* , in this case, $SI = \gamma_i, \forall i = 1$ to l , for intervals in the corresponding margin regions M'_{S_i} . In doing so, we assume that there are no outliers in the data; (i.e.) if the maximum margin between points belonging to a same cluster is M_m , we require that $M_m < \min_{i=1}^l \gamma_i$.

Proposition 2.2 For any $Int^{CL'}$ in $M'_{S_i} \perp S_i$, a $SI^* = \gamma_i$ will be realized iff $M'_{S_i} \equiv M_{S_i}$.

Proof The basic criteria for $SI^* = \gamma_i$ to exist is that there should be no \mathbb{X} in M_{S_i} . From the definition of the margin of a separating hyperplane, M_{S_i} will not contain \mathbb{X}_i . If $\exists \mathbb{X}'_i$ in M_{S_i} , then there will exist a $S_j, j \neq i$ (as determined by g), to classify \mathbb{X}'_i from \mathbb{X}_i . This, in turn, leads to an $M'_{S_i} \subset M_{S_i}$ containing no \mathbb{X} , which results in a maximum realizable $SI < \gamma_i$ for any $Int^{CL'}$ in $M'_{S_i} \perp S_i$. ■

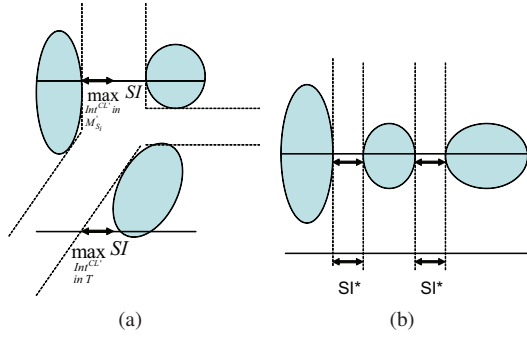


Figure 4. Illustrating the data-dependent nature of projections in intervals Int in T . Consider a three cluster problem, where the ellipses are completely filled with points. (a): Since $M_{S_i}' \subset M_{S_i}$, SI^* does not exist. However, the maximum possible SI occurs for intervals both in margin regions, and in T (shown with double head arrows). (b): When $M_{S_i}' \equiv M_{S_i}$, SI^* exists, and such intervals only belong to the margin regions (as in the case of a two-cluster problem).

We now focus on intervals belonging to other regions.

Corollary 2.3 For any interval Int^{CL} within the cluster region, $SI < \min_{i=1}^l \gamma_i$; and for an interval $Int^{CL'}$ in $M_{S_i}' \not\equiv M_{S_i}$, $SI < \gamma_i$.

Proof Follows from Propositions 2.1 and 2.2. ■

However, SI for intervals belonging to T is completely dependent on the spatial configuration of the data. Unless otherwise $M_{S_i}' \equiv M_{S_i}, \forall i = 1$ to k , the maximum SI realizable at an interval in M_{S_i}' can be realized for intervals in T also. An illustration is provided in Figure 4. Hence, with regard to the information conveyed by x_{i_p} , we finally state the following without a proof.

Corollary 2.4 Irrespective of whether $M_{S_i}' \equiv M_{S_i}$, an interval with $SI \geq \min_{i=1}^l \gamma_i$ can exist only outside a cluster. ■

2.2.2 Role of distance of projection d_{i_p}

Since existence of SI^* is itself dependent on the data, the location information of projected points x_{i_p} alone is insufficient to characterize margin properties for a multi-cluster problem. We make the following observation. M_{S_i}' , the informative subset of M_{S_i} , is obtained by *spatially bounding* M_{S_i} to remove the interactions of \mathbb{X}_i' . Hence, one way of translating this spatial neighborhood information for clustering is to use the distance of projection of points d_{i_p} . To understand the role of d_{i_p} , let us define D_{min} of a line interval to be the minimum⁴ d_{i_p} of all x_{i_p} projected in that interval. In similar vein, let D_{max} of an interval denote the maximum d_{i_p} of all x_{i_p} from that interval.

⁴To facilitate later discussion, for intervals with a $D_{min} = 0$, we set $D_{min} = \epsilon_{min}$, where ϵ_{min} is a positive real number slightly greater than zero. $\epsilon_{min} = .001$ in our experiments.

Proposition 2.5 D_{min} for intervals Int^{CL*} within a cluster of length more than M_m is less than that for all intervals Int^* in a margin region perpendicular to their corresponding separating hyperplanes, i.e. $Int^* = \cup_i \{Int^{CL'} \text{ in } M_{S_i}' \perp S_i\}$. Specifically, for intervals:

1. within a cluster region of length more than M_m , $\max_{Int^{CL*}} D_{min} \leq M_m/2$;
2. in the margin region perpendicular to the separating hyperplane, $\min_{Int^*} D_{min} \geq \min_{i=1}^l \gamma_i$.

Proof (i) This result comes directly from the no-outlier assumption in \mathbb{X} . When the maximum margin between any two points belonging to a cluster $M_m < \min_{i=1}^l \gamma_i$, for any Int^{CL*} there will exist an x_{i_p} with $0 \leq d_{i_p} \leq M_m/2$. Furthermore, there exists a pair of projections $(x_{i_p}, x_{j_p}), j \neq i$ such that, $0 \leq d_{i_p} \leq M_m/2, 0 \leq d_{j_p} \leq M_m$ and $0 \leq |d_{i_p} - d_{j_p}| \leq M_m$. (ii) For intervals $Int_i^* \in Int^*$ in $M_{S_i}' \perp S_i$, the points \mathbb{X}_i' need to travel a minimum distance of their corresponding margins to interfere with Int_i^* .

Hence, across all such intervals Int^* , $D_{min} \geq \min_{i=1}^l \gamma_i$. ■

The salient points of these discussions are captured by Figure 1 for a four-cluster problem where, (i) the intervals $a_1, a_2 \in Int^*$ illustrate that SI^* need not be realized at all margin regions, and (ii) D_{min} for intervals belonging to Int^* , for instance $a_2 \in L_2$ whose $D_{min} = d_1$, is *always* larger than that for intervals within a cluster of length more than M_m . Hence, d_{i_p} conveys much more data-independent⁵ information than that portrayed by x_{i_p} alone. We now define the following.

Definition Sparsity index of a line interval for a multi-cluster problem, $SI_m = [SI]_{\mathcal{D}} \in \mathbb{R}$, is the maximum distance travelled on that interval in which there exist no projected points x_{i_p} with $d_{i_p} < \mathcal{D}$. The dependency of SI_m on d_{i_p} is controlled by \mathcal{D} , which can take any value in the closed interval $[D_{min}, D_{max}]$.

As in the case of two-cluster problem, where $D_{min} = D_{max} = \infty$ for an interval with SI^* (Proposition 2.1), SI_m can be used to determine if an interval is associated within a cluster or outside cluster regions, as follows.

Proposition 2.6 An interval with $[SI]_{D_{min}^*} \geq \min_{i=1}^l \gamma_i$ can lie only outside a cluster, where $D_{min}^* \geq \min_{i=1}^l \gamma_i$.

Proof Follows from Corollary 2.4 and Proposition 2.5. Since $\max_{Int^{CL*}} D_{min} < \min_{Int^*} D_{min}$, intervals satisfying the above condition, say \tilde{Int} , can belong only to, (i) Int^* , and (ii) an interval in T depending on the data configuration. ■

⁵The properties of D_{min} for intervals in T , however, are completely dependent on data, as was the case with x_{i_p} (Figure 4).

However, unlike the two-cluster problem, such informative intervals \tilde{Int} do not provide the cluster assignments of \mathbb{X} directly. This is due to the inherent limitation of linear classifiers which, at the most, can separate only a pair of classes. Figures 1 and 2 illustrate this contrast, where although the interval a_1 on L_1 , and the interval in A_p pertaining to M_S realize a SI^* , only the latter interval could provide the cluster assignments. Methods by which the information contained in \tilde{Int} can be modeled to estimate the cluster assignments is the focus of the following section. At that point, we will also relax our assumption of requiring linear separability of clusters in their input space, and address the issue of outliers in data.

3. A Maximum-margin clustering algorithm

Determining the minimum value of D_{min}^* and the corresponding lower bound of $[SI]_{D_{min}^*}$, in an unsupervised setting, would require identifying a line perpendicular to separating hyperplane with the *least margin*. This is an ill-posed problem because the notion of D_{min}^* and $[SI]_{D_{min}^*}$ are relative with respect to the data configuration. Further, this process would *ideally* necessitate an analysis of projections of \mathbb{X} on all possible lines, and is therefore computationally intensive. Hence, we evaluate the *probability of presence* of \tilde{Int} between *all pair of points* in \mathbb{X} using Proposition 2.6, and perform global clustering using it to obtain the cluster assignments.

Since an interval belonging to \tilde{Int} will have a D_{min} (and the corresponding $[SI]_{D_{min}}$) greater than that for all intervals within a cluster, we define a pair-wise similarity measure,

$$f(x_i, x_j) = \exp(-\max_{\mathcal{D}: Int_{ij}} \mathcal{D}[SI]_{\mathcal{D}}) \quad (3)$$

which determines how probable is the absence of \tilde{Int} between the points x_i and x_j . Int_{ij} is the line interval between x_i and x_j containing projections of \mathbb{X} , from which the bounds for \mathcal{D} are determined to compute (3). Since Int_{ij} can contain intervals belonging to both Int^{CL} and $Int^{CL'}$, maximization over \mathcal{D} helps to identify the presence of \tilde{Int} (Proposition 2.6). We now make the following observations,

- Maximum value $f(x_i, x_j) = 1$ occurs only when $x_i = x_j$ since, (i) there exist no ‘interval’ between them ($SI = 0$), and (ii) for any point-pair $(x_i, x_j), j \neq i$, one can always find an infinitesimal interval (up to a discretization error) in Int_{ij} with $D_{min} > 0$, which would make $f(x_i, x_j) < 1$;
- Minimum value $f(x_i, x_j) \approx 0$ occurs only if x_i and x_j belong to different clusters, and Int_{ij} is perpendicular to the hyperplane that separates x_i and x_j , i.e., $Int_{ij} \in Int^* \subset \tilde{Int}$. Such cases will have a large $\max_{\mathcal{D}} \mathcal{D}[SI]_{\mathcal{D}}$, and from previous discussions, this value will be much higher than those when x_i and x_j belong to same cluster.

Essentially, the most significant edges connecting nodes from different clusters are those with least weights, $f(x_i, x_j) \approx 0$, which need to be ‘cut’ in order to obtain the cluster assignments. We use normalized cuts [24] for this purpose. Details are presented in Algorithm 1. Since we restrict our analysis to Int_{ij} between a pair of points (instead of using L_p in \mathbb{R}^N), we examine the fraction of such ‘meaningful’ edges obtained from each x_i in the Appendix.

Input: Set \mathbb{X} of M unlabelled points $\{x_i\}_{i=1}^M \in \mathbb{R}^N$, and number of clusters $k (> 0)$.
Output: The cluster assignments $y_i \in \{1, 2, \dots, k\}, \forall i = 1 \text{ to } M$, providing the maximum separating margin (Proposition 2.6). Do:

1. Compute projections of \mathbb{X} on the set of line intervals between all possible points pairs, $Int_{ij} : (x_i, x_j), \forall 1 \leq i, j \leq M$.
2. Compute a symmetric $M \times M$ similarity matrix S^* , with its entries $f(x_i, x_j), \forall 1 \leq i, j \leq M$ obtained from (3).
3. Perform normalized cuts (NCut) [24]; $y = NCut(S^*, k)$ to obtain the cluster assignments.

Algorithm 1: Maximum-margin clustering algorithm.

3.1. Design Issues

Computing f : Since we use an exponential function to compute (3), we first normalize SI and d_{i_p} with the maximum distance between two data points, and the maximum value of d_{i_p} across projections of \mathbb{X} on lines between all pairs of points, respectively. Then while evaluating (3), we need to account for the possibility of existence of no x_{i_p} in a small interval ($SI \approx 0$, and/or $SI < \gamma$) within a cluster. Such a condition results in $D_{min} = \infty$, which makes $f = 0$. To avoid such instances, we place an upper bound on the maximum value of D_{min} : $\bar{D}_{min} = n_1 * (\max_{Int \in L_p} d_{i_p}), n_1 >$

1. Since we normalize d_{i_p} , $\bar{D}_{min} = n_1$, and we chose $n_1 = 7$ for our experiments. This choice would make the minimum value of $f \approx 10^{-3}$, when the corresponding (normalized) $SI \approx 1$. From previous results, the instance with $SI \approx 1$ and $D_{min} = n_1$ will happen only when x_i and x_j belong to different clusters.

When data is not linearly separable: Since the basic information needed to compute (3) comes from x_{i_p} and d_{i_p} , and these computations involve dot products, we accommodate non-linearly separable data using the kernel trick [1].

Effect of outliers: The choice of normalized cuts to perform clustering based on (3) is primarily to obtain balanced clusters, which offers some resistance to outliers. Hence, our method is less prone to the presence of *isolated points* belonging to a cluster. An illustration is given in Figure 5. However, unlike outlier-robust *supervised* max-margin classifiers that use slack variables (eg. [5]), we cannot deal with conditions where a point belonging to cluster 1 is present inside cluster 2, and both clusters are well-balanced.

Computational complexity: Obtaining the projections of \mathbb{X} on line segments between all pairs of points has a cost of $O(M)$ for each line segment, and $O(M^2)$ for all point-pairs, thereby yielding a total cost of $O(M^3)$. To compute

f for a point-pair (3) with this information, we need to analyze the maximum distance between adjacent x_{i_p} 's (to compute $[SI]_{\mathcal{D}}$) for a maximum of M possible values of d_{i_p} . However, we discretized the d_{i_p} values into five equal intervals between 0 and 1. Hence, this stage has a cost of $O(M \log M)$ to sort x_{i_p} 's between a point-pair, and when performed for all point pairs incurs a cost of $O(M^3 \log M)$. These two stages, though, permit parallelization to improve efficiency. We then perform normalized cuts, which involves eigen-decomposition with M nodes, and thereby has a maximum cost of $O(M^3)$. Hence, the overall computational complexity of our method is $O(M^3 \log M)$, which is slightly more than that of normalized cuts.

4. Experiments

We performed experiments both on synthetic, and real data to evaluate our method. In all these experiments, we used the following set of kernels: linear, polynomial, RBF (radial basis function), and sigmoid. We then chose the kernel with least Ncut cost (Algorithm 1) to determine the cluster assignments. These results are then matched with the ground truth to compute the clustering accuracy. More details on computing clustering accuracy, and the error statistics across different kernels are provided in the supplementary material. On the whole, we saw an improvement in clustering accuracy of about 6% on average, and up to a maximum 15% using our method on several synthetic, and real datasets. For cases where we did not perform the best, we were outperformed by an average of about 1% and a maximum of 3.5%.

4.1. Synthetic data

We experimented with synthetic data⁶ containing multiple clusters (with maximum $k = 10$), and with cases where the clusters are not linearly separable in their input space. We generated 100 synthetic data, where the first set of 50 samples had outliers, and the second set of remaining samples had no outliers. The outlier instances were not restricted to cases with isolated points, which normalized cuts can handle relatively well. We tested the sensitivity of our algorithm to outliers, and to the absence of the exact value of k (we ran the algorithm for $2 \leq k < 10$) on this dataset and present the results of clustering accuracy in Table 1(a). Some clustering results using our method are shown in Figures 5 and 6, where $\mathbb{X} \in \mathbb{R}^2$, and $\mathbb{X} \in \mathbb{R}^3$. We also show some results using K-means (KM) [19] and using normalized cuts (NC) [24] with the pairwise-similarity measure $f^l(x_i, x_j) = \exp(-\|x_i - x_j\|_2^2 / \sigma)$, to illustrate the sensitivity of these algorithms to cluster-center initialization, and the value of σ , respectively. Hence, one advantage of our approach is its reduced dependence on parameter tuning. The results of KM and NC, for each kernel parameter setting (and number of clusters), were averaged over 50 trials, and different values of σ (set by a exhaustive

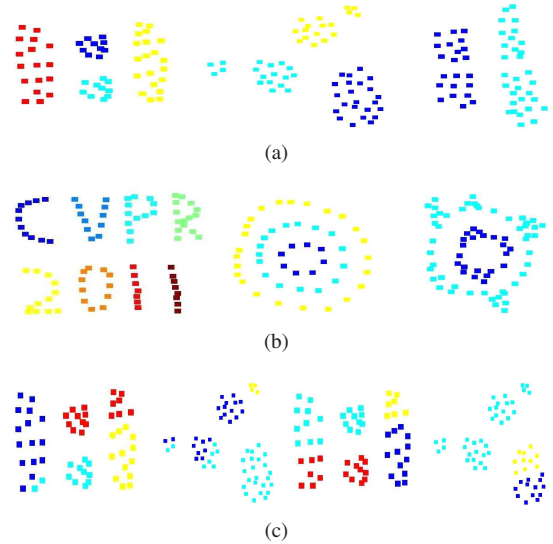


Figure 5. Clustering results on synthetic data $\mathbb{X} \in \mathbb{R}^2$. (a),(b): Results using our method showing robustness to outliers, and in characterizing margin properties. (c): the first two figures shows sample mis-clustering result from KM, and the last two from NC - to illustrate the sensitivity of these algorithms to cluster center initialization, and parameter tuning respectively. (Data magnified by a factor of 5.)

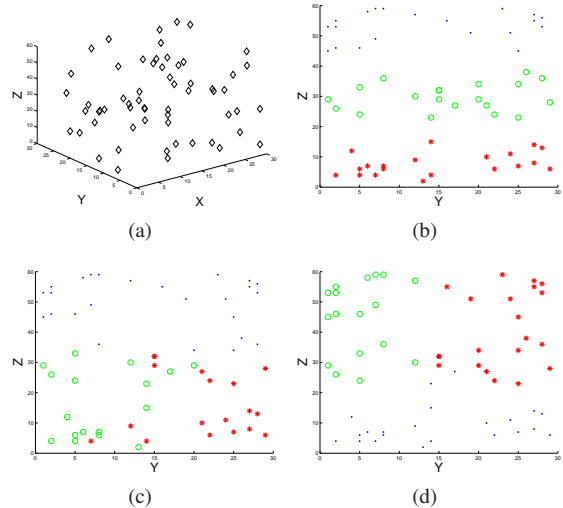


Figure 6. Clustering results on synthetic data $\mathbb{X} \in \mathbb{R}^3$. (a) original data. Data is randomly distributed in x- and y- directions from 0 to 30, and has three splits in the z- direction: 0 to 15, 23 to 38, and 45 to 60. Clustering results are shown in the y-z plane. (b) our method, (c) KM, (d) NC.

search over the distance between all point-pairs in the data) respectively, and the mean and standard deviation of clustering accuracy for the best parameter set are reported. With an improvement of around 9% in accuracy, our method has better tolerance to outliers. It also shows much better performance when the exact value of k is unknown.

4.2. Comparison with existing methods on real data

We then evaluated our method on the experimental setup of Wang et al. [27] that performs maximum margin clustering by optimizing over all possible cluster assignments, and that of Ye et al. [30] which performs discriminative clustering by integrating linear discriminant analysis-based

⁶www.umiacs.umd.edu/users/raghuran/Datasets/MaxMarginClustering.zip

(a)					(b)							
Data	KM [19]	NC [24]	Ours		Data	KM [19]	NC [24]	MMC [28]	GMMC [26]	IterSVR [31]	CPMMC [27]	Ours
set 1, k known	72.2±3.41	67.5±4.22	76.5		UCI-Iono.	54.28	75.00	78.75	76.50	77.70	75.48	86.17
set 1 & 2, k known	76.6±2.50	77.1±3.02	85.5		UCI-Let.	82.06	76.80	-	-	92.80	95.02	97.25
set 2, k unknown	70.9±3.28	78.55±2.31	92.3		UCI-Sat.	95.93	95.79	-	-	96.82	98.79	98.35
set 1 & 2, k unknown	58.78±3.96	61.43±4.55	78.1		Text-1	50.53	93.79	-	-	97.02	95.00	98.56
					Text-2	50.38	91.35	-	-	93.99	97.21	96.98
					Digits 3-8	94.68	65.00	90.00	94.40	96.64	96.88	97.33
					Digits 1-7	94.45	55.00	68.75	97.8	99.45	100.0	100.0
					Digits 2-7	96.91	66.00	98.75	99.50	100.0	100.0	100.0
					Digits 8-9	90.68	52.00	96.25	84.00	96.33	98.12	99.56
					UCI-Digit	96.38	97.57	-	-	98.18	99.40	99.52
					MNIST	89.21	89.92	-	-	92.41	96.21	98.55

(c)					(d)					
Data	KM [19]	NC [24]	CPM3C [27]	Ours	Data	DisKmeans [8] (max,mean)	DisCluster [30] (max,mean)	LLE [23]	LEI [3]	Ours
UCI-digits 0689	42.23	93.13	96.74	96.11	UCI-banding	(77.1,76.8)	(77.1,76.7)	64.8	76.4	83.2
UCI-digits 1279	40.42	90.11	94.52	96.54	UCI-soybean	(64.1,63.4)	(63.3,63.2)	63.0	64.9	68.7
USPS	92.15	92.81	95.03	97.11	UCI-segment	(68.7,66.4)	(67.6,67.2)	59.4	66.3	73.1
Cora-DS	28.24	36.88	44.15	56.31	UCI-pendigits	(69.9,69.0)	(69.6,69.0)	59.9	69.7	70.1
Cora-HA	34.02	42.00	59.80	69.86	UCI-satimage	(70.1,65.1)	(65.4,64.2)	62.7	66.3	69.5
Cora-ML 3-8	27.08	31.05	45.49	42.33	UCI-leukemia	(77.5,76.3)	(73.8,73.8)	71.4	68.6	77.4
Cora-OS 1-7	23.87	23.03	59.16	75.87	ORL	(74.4,73.8)	(73.9,73.8)	73.3	31.7	79.1
Cora-PL 2-7	33.80	33.97	47.21	53.33	USPS	(71.2,62.8)	(69.2,68.3)	63.1	70.0	75.3
WebKB-Corn. 8-9	55.71	61.43	72.05	73.11						
WebKB-Texas	45.05	35.38	69.10	75.60						
WebKB-Wash.	53.52	32.85	78.17	82.43						
WebKB-Wisc.	49.53	33.31	74.25	79.23						
20-newsgroup	35.27	41.89	71.34	71.44						
Reuters-RCVI	27.05	-	62.35	72.81						

Table 1. (a) Clustering accuracy (in %) on a synthetic dataset of around 100 samples with $X \in \mathbb{R}^2$ and $X \in \mathbb{R}^3$. Set 1 had 50 samples with outliers, and Set 2 had the remaining samples with no outliers. For the experiment with unknown k , trials with $2 \leq k \leq 10$ were done. The clustering accuracy of parameter set with the least Ncut cost (Algorithm 1) is given for our method, whereas for [19] and [24] the results correspond to mean and standard deviation of clustering accuracy of the best performing parameter set (across initialization and σ values respectively). (b),(c) Comparison with max-margin clustering methods. Clustering accuracy (in %) for, (b): two-cluster problems, and (c): multi-cluster problems. The experimental setup in [27] was followed, and the results presented in their work have been reproduced here. Only linear kernel was used. (d) Comparison with methods that integrate dimensionality reduction and clustering. Clustering accuracy (in %) on multi-cluster problem. Experimental setup of [30] was followed, and their results are reproduced here. Some methods have (max,mean) for clustering accuracy since they have a regularization parameter operating on a pre-specified kernel. Additional information on UCI datasets: banding - 0689, soybean - 1279, Iono. - Ionosphere, Let. - Letters, Sat. - Satellite.

dimensionality reduction and K-means clustering. The datasets belonged to the UCI repository [10], text data (20-newsgroup⁷, WebKB⁸, Cora [20] and RCVI [18]), digits data (USPS⁹, and MNIST [17]), and ORL face dataset¹⁰. A detailed explanation of these datasets is given in the supplementary material. The results of clustering accuracy comparison with max-margin clustering methods are given in Tables 1(b) and 1(c), and the comparison with the discriminative clustering methods is given in Table 1(d). Experiments with unknown k on these datasets are given in supplementary material. It can be seen that our method compares favorably with other methods on many datasets, offering an overall improvement of 3 to 4%.

4.3. Experiments on vision problems

4.3.1 Face recognition across lighting variations

We used the YaleB dataset [12] and CMU-PIE Illumination dataset [25]. The YaleB dataset has images of 38 subjects under 64 different lighting conditions, and the PIE dataset has 68 subjects with 21 lighting conditions. Sample images are given in supplementary material. No other facial variations such as pose, alignment etc. were present. The images were resized to 48×40 , and the gradient orientation information was computed at each pixel. This feature, which was

shown to be robust against lighting changes [6], was vectorized to constitute x_i 's. Clustering was then performed using normalized cuts on the pair-wise information f (3).

4.3.2 2D Shape matching

We used the MPEG-7 shape retrieval dataset [16] and an articulation dataset [13]. The MPEG-7 dataset contains 70 classes of shapes with 20 instances per class containing general shape deformations. The articulation dataset contains 5 classes, with 10 shapes per class, where the main source of variation is non-planar articulations. Sample images are given in supplementary material. The underlying shape representation was a shape context descriptor invariant to non-planar articulations. For each aligned shape (2D silhouettes), 100 points were sampled uniformly along the contour, and a log-polar histogram was associated with each point using the method of [13]. We used 5 radial bins, and 12 angular bins resulting in a 100×60 shape descriptor for each shape. The vectorized form of this descriptor represents $x_i \in \mathbb{X}$, using which clustering is done.

We compared our method with Zhang et al. [31], and Ye et al. [30], and the results are given in Table 2. We used the publicly available source code for [31]¹¹, implemented [30], and verified results on the datasets used for this work. Experiments with actual value of k not known, are provided in the supplementary material. These results, with roughly

⁷ people.csail.mit.edu/frennie/20Newsgroups/

⁸ www.cs.cmu.edu/~WebKB/

⁹ www.kernel-machines.org/data.html

¹⁰ www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html

¹¹ www.cse.ust.hk/~twinsen

Data	DisKmeans [8]	IterSVR [31]	Ours
Face-YaleB	65.6	68.1	77.4
Face-PIE	69.2	71.0	79.5
Shape-MPEG7	55.9	51.2	59.3
Shape-Articulation	42.3	38.5	51.4

Table 2. Clustering accuracy (in %) on datasets for face recognition across lighting condition, and shape matching. As before, the result for our method correspond to kernel parameters with least NCut cost, whereas for the other two methods we report the maximum clustering accuracy.

a 7% improvement, demonstrate potential applications of our method towards unsupervised pattern discovery in vision problems. As a final note, although it is desirable to have a good grouping mechanism, we would like to emphasize the equally important component of ‘data representation’ on which we operate on.

5. Conclusions

We addressed the problem of obtaining clusters with maximum separating margins, by studying the pattern of projections of points on all possible lines in the data space. By drawing parallels with supervised max-margin classification, we derived properties that projections on a line interval would satisfy *if and only if* that interval lies outside a cluster, under assumptions on linear separability of clusters and absence of outliers. We then proposed a pair-wise similarity measure to model this information to perform clustering, by accommodating non-linearly separable data using kernel methods, and (partially) handling outliers by placing emphasis on the cluster size. The experiments illustrated the utility of our method when applied to standard datasets, and to problems in computer vision.

Appendix - On realizing D_{min}^* with a restricted analysis on Int_{ij} between all pairs of points

We analyze the consequence of a restricted analysis on line intervals between a pair of points in \mathbb{X} , rather than all L_p in \mathbb{R}^N . Let $Int_{ij}^{full} \subset L_p$ denote line intervals between all pair of points $(x_i, x_j) \in \mathbb{X}$ containing the projections of \mathbb{X} . Let $Int_{ij}^{full} = Int_{ij}^1 \cup Int_{ij}^2$ comprise of two disjoint sets that denote line intervals between a pair of points belonging to same, and different clusters, respectively. From Proposition 2.6, only those intervals in $Int_{ij}^2 \perp S_i$, where S_i is the hyperplane separating the pair of points connected by Int_{ij}^2 , can have a $D_{min} \geq D_{min}^*$. Let us now analyze the possibility of obtaining such intervals in Int_{ij}^2 .

Let $0 < \theta \leq 90^\circ$ denote the angle¹² between an interval Int_{ij}^2 with its corresponding S_i . Let us analyze the distribution of θ tended by the set of all lines joining a point x_i to all points x_j belonging to a different cluster. Since this is a data-dependent analysis, without the loss of generality, let us assume θ to be uniformly distributed between 0 and (including) 90° . Let us split this into n_2 equally spaced angular bins. Essentially, $\forall x_i \in \mathbb{X}$, at least $1/n_2$ of its connections with points x_j in other clusters will *almost surely* realize D_{min}^* (since their $\theta \approx 90^\circ$). Hence, for each point,

¹² θ can not equal zero, since the line between a pair of points belonging to different clusters can never be parallel to the hyperplane that separates them.

these are the connections that need to be ‘cut’ in order to group points ($f \ll 1$).

Acknowledgements

This work was supported by a MURI grant N00014-10-1-0934 from the Office of Naval Research. The authors thank Prof. Rama Chellappa for helpful comments on the work.

References

- [1] A. Aizerman, E. M. Braverman, and L. I. Rozoner. Theoretical foundations of the potential function method in pattern recognition learning. *Automation and Remote Control*, 25:821–837, 1964.
- [2] E. L. Allwein, R. E. Schapire, and Y. Singer. Reducing multiclass to binary: a unifying approach for margin classifiers. *JMLR*, 1:113–141, 2001.
- [3] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, 2003.
- [4] A. Ben-Hur, D. Horn, H. T. Siegelmann, and V. Vapnik. Support vector clustering. *JMLR*, 2:125–137, 2001.
- [5] C. J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Min. Knowl. Discov.*, 2(2):121–167, 1998.
- [6] H. Chen, P. Belhumeur, and D. Jacobs. In search of illumination invariants. In *CVPR*, volume 1, pages 254–261, 2000.
- [7] K. Crammer and Y. Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *JMLR*, 2:265–292, 2002.
- [8] F. De la Torre and T. Kanade. Discriminative cluster analysis. In *ICML*, pages 241–248, 2006.
- [9] R. Duda, P. Hart, and D. Stork. *Pattern classification*. Wiley, 2001.
- [10] A. Frank and A. Asuncion. UCI machine learning repository, 2010.
- [11] K. Fukunaga. *Introduction to statistical pattern recognition (2nd ed.)*. Academic Press Professional, Inc., 1990.
- [12] A. Georghiadis, P. Belhumeur, and D. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE TPAMI*, 23(6):643–660, 2001.
- [13] R. Gopalan, P. Turaga, and R. Chellappa. Articulation-invariant representation of non-planar shapes. In *ECCV*, pages 286–299, 2010.
- [14] A. K. Jain and R. C. Dubes. *Algorithms for clustering data*. Prentice-Hall, Inc., 1988.
- [15] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. *ACM Comput. Surv.*, 31(3):264–323, 1999.
- [16] L. Latecki, R. Lakämper, and T. Eckhardt. Shape descriptors for non-rigid shapes with a single closed contour. In *CVPR*, pages 424–429, 2000.
- [17] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *IEEE*, 86(11):2278–2324, nov. 1998.
- [18] D. D. Lewis, Y. Yang, T. G. Rose, and F. Li. Rcv1: A new benchmark collection for text categorization research. *JMLR*, 5:361–397, 2004.
- [19] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297, 1967.
- [20] A. K. McCallum, K. Nigam, J. Rennie, and K. Seymore. Automating the construction of internet portals with machine learning. *Information Retrieval*, 3(2):127–163, 2000.
- [21] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *NIPS*, pages 849–856, 2001.
- [22] J. Peng, L. Mukherjee, V. Singh, D. Schuurmans, and L. Xu. An efficient algorithm for maximal margin clustering. *Journal of Global Optimization*, pages 1–15, Feb 2011.
- [23] S. Roweis and L. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290:2323–2326, 2000.
- [24] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE TPAMI*, 22(8):888–905, aug. 2000.
- [25] T. Sim, S. Baker, and M. Bsat. The cmu pose, illumination, and expression database. *IEEE TPAMI*, 25(1):1615–1618, December 2003.
- [26] H. Valizadegan and R. Jin. Generalized maximum margin clustering and unsupervised kernel learning. In *NIPS*, pages 1417–1424, 2007.
- [27] F. Wang, B. Zhao, and C. Zhang. Linear time maximum margin clustering. *IEEE Trans. Neural Networks*, 21(2):319–332, 2010.
- [28] L. Xu, J. Neufeld, B. Larson, and D. Schuurmans. Maximum margin clustering. *NIPS*, pages 1537–1544, 2005.
- [29] J. Ye, Z. Zhao, and H. Liu. Adaptive distance metric learning for clustering. *CVPR*, pages 1–7, 2007.
- [30] J. Ye, Z. Zhao, and M. Wu. Discriminative k-means for clustering. In *NIPS*, pages 1649–1656, 2008.
- [31] K. Zhang, I. W. Tsang, and J. T. Kwok. Maximum margin clustering made practical. *IEEE Trans. Neural Networks*, 20(4):583–596, 2009.
- [32] B. Zhao, J. Kwok, F. Wang, and C. Zhang. Unsupervised maximum margin feature selection with manifold regularization. In *CVPR*, pages 888–895, 2009.
- [33] B. Zhao, F. Wang, and C. Zhang. Efficient multiclass maximum margin clustering. In *ICML*, pages 1248–1255, 2008.