J Glob Optim
DOI 10.1007/s10898-011-9691-4

An efficient algorithm for maximal margin clustering

Jiming Peng · Lopamudra Mukherjee · Vikas Singh · Dale Schuurmans · Linli Xu

Received: 29 April 2009 / Accepted: 5 February 2011

Abstract Maximal margin based frameworks have emerged as a powerful tool for supervised learning. The extension of these ideas to the unsupervised case, however, is problematic since the underlying optimization entails a discrete component. In this paper, we first study the computational complexity of maximal hard margin clustering and show that the hard margin clustering problem can be *precisely* solved in $O(n^{d+2})$ time where n is the number of the data points and d is the dimensionality of the input data. However, since it is well known that many datasets commonly 'express' themselves primarily in far fewer dimensions, our interest is in evaluating if a careful use of dimensionality reduction can lead to practical and effective algorithms. We build upon these observations and propose a new algorithm that gradually increases the number of features used in the separation model in each iteration, and analyze the convergence properties of this scheme. We report on

J. Peng (✉)
Department of Industrial and Enterprise Systems Engineering,
University of Illinois at Urbana-Champaign, Urbana, IL, USA
e-mail: pengj@illinois.edu

L. Mukherjee
Department of Mathematical and Computer Sciences,
University of Wisconsin–Whitewater, Whitewater, WI, USA
e-mail: mukherjl@uww.edu

V. Singh
Departments of Biostatistics and Medical Informatics and Computer Sciences,
University of Wisconsin–Madison, Madison, WI, USA
e-mail: vsingh@biostat.wisc.edu

D. Schuurmans
Department of Computing Science, University of Alberta, Edmonton, AB, Canada
e-mail: dale@cs.ualberta.ca

L. Xu
School of Computer Science and Technology, University of Science and Technology of China,
Hefei, China
e-mail: linlixu@ustc.edu.cn

Published online: 22 February 2011

promising numerical experiments based on a 'truncated' version of this approach. Our experiments indicate that for a variety of datasets, good solutions *equivalent* to those from other existing techniques can be obtained in *significantly* less time.

Keywords Maximum margin · Clustering · Unsupervised · SDP

1 Introduction

The study of maximal margin-based learning algorithms [1] (or support vector machines) has become an active area of research in machine learning and data mining over the past few decades. SVM frameworks have been developed for a variety of classification problems and have been applied in a number of different application domains, see [2]. Inspired by the success of support vector machines for supervised learning, Xu et al. [3] recently proposed a maximal margin clustering algorithm for *unsupervised* learning problems. In subsequent work, Xu and Schuurmans [4] extended their earlier formulations [3] to the semi-supervised multi-class classification setting. Promising experiments on small-scale datasets were reported in [3,4]. In general, empirical results have demonstrated that maximal margin clustering achieves good quality solutions relative to other popular algorithms. Unfortunately, such experiments have also highlighted a few important limitations [5]. A major shortcoming as noted in [5] is the running time, which increases very rapidly with the size of the data—making it practically infeasible for use with larger datasets. Addressing this difficulty is our primary interest in this paper.

1.1 Related work

The idea of separating unlabeled data sets with maximal margin can be traced back to [6] where the authors considered semi-supervised or transductive SVMs. Transductive support vector machines (TSVMs) use unlabeled data to improve the generalization of SVMs. Here, a large margin separating hyperplane is determined using labeled training data. However, as an extension of the supervised setting, this hyperplane is forced to pass through low density regions of the unlabeled data. In recent work [7,8], a convex relaxation for transductive SVMs was proposed using spectral transduction to find the subspace of interest as means of approximating the underlying optimization problem (which will be introduced shortly). These algorithms essentially follow a mechanism similar to standard SVMs, i.e., they use the primal-dual structure of convex quadratic optimization problems. Let us discuss the relevant details of such an approach briefly. Consider the hard maximal margin clustering problem (without class balance constraints) defined as follows[1]:

$$\min_{y_i \in \{1,-1\}} f(y) = \min_{w} w^T w$$
$$s.t. \quad y_i(w^T v_i + b) \geq 1, \quad i = 1, \ldots, n. \tag{1}$$

In [3,7], the authors derived the dual of the low-level problem that allowed them to focus on the following model instead.

[1] We note that in [3,4], to accommodate the inseparable case, the authors used the so-called soft SVMs. However, as we shall discuss later, for clustering problems, the data set is *always* separable.

$$\min_{y_i \in \{-1,1\}} f(y) = \max_{u} 2u^T e - u^T Y M Y u$$
$$s.t. \quad u \geq 1, \quad u^T y = 0, \quad Y = \text{diag}(y), \tag{2}$$

where $M = [m_{ij}]$ is the square matrix whose entries are given as $m_{ij} = v_i^T v_j$. We can see that problems (1) and (2) are both challenging *discrete* optimization problems, and it is difficult to solve them (to optimality) efficiently. A simple analysis shows that the semidefinite programming (SDP) relaxations of these problems are nontrivial as well. For example, the SDP relaxation proposed in [3,4] has $O(n^2)$ constraints. Therefore, the model can only be applied to data sets of a relatively small size. To address this problem, the authors in [5] proposed another SDP based approach based on the Lagrangian dual of the problem. While the approach [5] reduces the complexity of earlier models in [3,4], there is no guarantee that the optimal solutions from the primal and dual problems are equivalent. Also, the algorithm in [5] is still dependent on the capacity of the underlying SDP solvers. For a given SDP problem of size n with m constraints, the computational cost at each iteration of interior-point methods (using the best available SDP solver) is $O(n^2m^2 + mn^3)$ [9]; problems with large-scale data sets lie much beyond what the fastest solvers [9] available at this time offer.[2] In an effort to at least partially mitigate this problem, a strategy of updating u and y in (2) in an iterative framework was investigated in a recent work [10]. However, the authors reported that the experimental results from the iterative SVM procedure were unsatisfactory, leading them to adopt a regression approach. In other words, [10] did not directly address the maximal margin model [3]; rather, proposed and argued for an alternate soft margin formulation.

In summary, all existing algorithms for maximal margin clustering follow a framework similar to SVMs for supervised learning. This strategy has certain advantages but makes it difficult to explore (and make use of) the clustering specific characteristics in the design of the algorithm. For example, while separability is a major concern in SVMs for *(semi)supervised learning*, it is not a key issue for *data clustering* since *any* hyperplane in the input space that passes through the center of a line segment between a point pair in the data set will be able to "*separate*" the data set into two subsets. Observe that we avoid asking how "good" the separation is—instead focus merely on the separability of *unlabeled* points. The "goodness" of a separator is a quantity we would like to optimize later. With this in mind, we propose a new approach for the maximal margin clustering. Under mild assumptions, we first show that the hard maximal margin clustering problem can be solved in $O(n^{d+2})$ time, where n is the size of the input data set, and d is the dimensionality of the input space. As we will see shortly, this yields an efficient and practical algorithm for problems involving datasets that can be characterized in lower dimensional spaces.

The discussion above brings up the issue of a balance between the quality of the solution and the algorithm efficiency. We note that exact algorithms for several clustering problems like k-means typically have a very high complexity, but nice approximation algorithms based on convex relaxation and subspace ideas have been suggested [8,11]. For more recent advances in optimization for clustering, we refer to [12–18] and the references therein. Inspired by these results, we consider how to incorporate feature selection concepts into the optimization model for maximal margin clustering, in an effort to improve efficiency. For instance, for a given parameter k, we consider the problem of finding the maximal margin separation by using at most k features of that instance. It can be verified that as k increases, the separating margin becomes larger. This yields a cascade for different choices of k. By this

[2] Also observe that the number of constraints m (where $m = n^2$) is much larger than the size of the matrix n. Therefore, there is a large gap between the complexity of the standard SDP problem (where $m < n$) and the SDP relaxation of the maximal-margin problem.

logic, the maximal margin clustering with a fixed number of features can also be cast as an approximate solution to the original maximal margin clustering problem and be used in a semi-supervised learning setting. Later, we investigate an algorithm based on this idea and analyze its convergence properties.

The rest of the paper is organized as follows. In Sect. 2, we discuss the complexity of the maximal margin problem and present an algorithm. Then, we analyze how the idea of maximal margin separation can be extended to some existing graph-cut models for clustering. In Sect. 3, we discuss the maximal margin clustering problem with a fixed number of features, which can be viewed as a combination of the maximal margin clustering and feature selection. In Sect. 4, we propose a new successive procedure to approximately solve the original hard maximal margin clustering problem and study its convergence properties. We also discuss how to deal with the semi-supervised learning case. Experimental results based on these ideas are reported in Sect. 5 and we conclude the paper in Sect. 6.

2 Complexity of hard maximal margin clustering

In this section, we investigate the complexity of the hard maximal margin clustering. Throughout the paper, we make the following assumption:

Assumption 1 [*General Position*] All the points in the input data set $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$ are in general position, i.e., any d points in $\mathcal{V}$ will define precisely one hyperplane in $\Re^d$.

We remark that the above assumption is quite reasonable, because if the data set $\mathcal{V}$ does not satisfy the assumption, we may perturb the data set slightly so that the assumption holds. We can now state the following result.

Theorem 1 *Suppose that the input data set satisfies Assumption 1 and (w^*, b^*) is the global solution to problem (1). Then, there must exist a subset $\mathcal{V}^* \subset \mathcal{V}$ with either d or $d+1$ points such that at least one of the following two conditions hold:*

(1) *The hyperplane defined by the points in $\mathcal{V}^*$ will create the same separation as the optimal separating hyperplane.*
(2) *The optimal separating hyperplane is also the global solution of problem (1) where $\mathcal{V}$ is replaced by $\mathcal{V}^*$.*

Proof Suppose that (w^*, b^*) is the global solution to problem (1), we can then separate the data set $\mathcal{V}$ into two subsets $\mathcal{V}_1$ and $\mathcal{V}_2$ such that

$$\begin{aligned} \mathcal{V}_1 &= \{v \in \mathcal{V} : (w^*)^T v + b^* > 0\}, \\ \mathcal{V}_2 &= \{v \in \mathcal{V} : (w^*)^T v + b^* < 0\}. \end{aligned} \tag{3}$$

Further, by the definition of maximal margin, there exist two subsets $\tilde{\mathcal{V}}_1, \tilde{\mathcal{V}}_2$ such that $\forall\, v_1, v_2 : v_1 \in \tilde{\mathcal{V}}_1 \subset \mathcal{V}_1, v_2 \in \tilde{\mathcal{V}}_2 \subset \mathcal{V}_2$,

$$(w^*)^T v_1 + b^* = -\left((w^*)^T v_2 + b^*\right) = 1. \tag{4}$$

For a given data set $\mathcal{V}$, let $|\mathcal{V}|$ denote the number of points in $\mathcal{V}$. Since all points in $\mathcal{V}$ are in general position, we can claim that

$$|\tilde{\mathcal{V}}_1| \le d, \quad |\tilde{\mathcal{V}}_2| \le d, \quad |\tilde{\mathcal{V}}_1| + |\tilde{\mathcal{V}}_2| \ge d. \tag{5}$$

If there exist at least $d+1$ points in $\tilde{\mathcal{V}}_1 \cup \tilde{\mathcal{V}}_2$, we may select $d+1$ points from $\tilde{\mathcal{V}}_1 \cup \tilde{\mathcal{V}}_2$ and these $d+1$ points will define the separating hyperplane $(w^*)^T v + b^* = 0$ due to the general

position assumption. If there exist precisely d points in $\tilde{\mathcal{V}}_1 \cup \tilde{\mathcal{V}}_2$, we can find the maximal margin separation hyperplane based on these d points. This statement of the theorem follows. □

Based on the above result, we can perform a comprehensive search over all the subsets of size d and find the global solution of problem (1). Let $C(n, d)$ be the combinatorial function of selecting d points out of n points in total, $\binom{n}{d}$. The algorithm above has a running time of $C(n, d+1)2^{d+1}n$, i.e., $O(n^{d+2})$. If $d = 1$, the complexity reduces to $O(n \log n)$—we first sort the data points, and then find the maximal margin separation. If $d = 2$, from relation (5) we can see that both sets $\tilde{\mathcal{V}}_1$ and $\tilde{\mathcal{V}}_2$ have either 1 or 2 points. Suppose that $\tilde{\mathcal{V}}_1$ contains two points, then those two points will define the separating hyperplane. If both $\tilde{\mathcal{V}}_1$ and $\tilde{\mathcal{V}}_2$ have only one point, i.e., $v_1 \in \tilde{\mathcal{V}}_1$, $v_2 \in \tilde{\mathcal{V}}_2$, then it is easy to see that the straight line perpendicular to the segment from v_1 to v_2 which passes through the midpoint $(v_1 + v_2)/2$ is the optimal separating line. Therefore, we can enumerate all possible point pairs. For every point pair, we compare the two margins derived by the two separating lines described in our previous discussion and choose the one with a larger margin, which gives a running time of $O(n^3)$. The procedure is quite efficient for datasets in lower dimensions, though it is still expensive for large values of d. In the next subsection, we briefly remark on the relationship of the maximal margin model with graph cut based clustering. We then discuss in detail how feature selection combined with the ideas above can be used to approximately solve the maximal margin clustering.

2.1 Connections to graph-cut based clustering

Graph partitioning based approaches have emerged as powerful tools for clustering in the last decade [11,19]. Such techniques have also been found to be extremely useful for image segmentation problems [19]. This popularity has led to investigations into the connection of such algorithms to other techniques such as kernel k-means [20] and transductive SVMs [6]. Here, we discuss how the maximal margin approach for unsupervised learning can also be extended to deal with graph partitioning based clustering problems (e.g., normalized cuts [19]). To see this, we first recall the definition of the normalized cut problem. Let W be the weight matrix of a graph, and $X = [x_{ij}] \in \Re^{n\times 2}$ be the assignment matrix and e_m be the all 1 vector in $\Re^m$. Let us define

$$\mathcal{F} = \{X : Xe_2 = e_n, \quad x_{ij} \in \{0, 1\}\}.$$

Let $\gamma = We_n$ and $\Gamma = \text{diag}(\gamma)$. The exact model for the normalized k-cut problem in [19] can be rewritten as

$$\min_{X\in\mathcal{F}} \text{tr}(\Gamma^{-1}W - X(X^T\Gamma X)^{-1}X^T W) \tag{6}$$

If we define

$$Z = \Gamma^{\frac{1}{2}}X(X^T\Gamma X)^{-1}X^T\Gamma^{\frac{1}{2}}, W_\gamma = \Gamma^{-\frac{1}{2}}W\Gamma^{-\frac{1}{2}}, \tag{7}$$

then the above model can be equivalently stated as:

$$\min \text{tr}(W_\gamma(I - Z)) \tag{8}$$

$$Z\gamma^{\frac{1}{2}} = \gamma^{\frac{1}{2}}, \ \text{tr}(Z) = 2, \tag{9}$$

$$Z \geq 0, \ Z^2 = Z, Z = Z^T. \tag{10}$$

It has been shown [20,21] that when Γ is the identity matrix (or $\gamma = e_n$) and $w_{ij} = v_i^T v_j$, the above model amounts to the classical k-means clustering. Therefore, we can interpret γ as a special scaling vector that enables us to normalize and project the input data (derived by using the singular value decomposition of W_γ) in a suitable subspace. Observe that since 1 is the largest eigenvalue of W_γ and $W_\gamma \gamma^{\frac{1}{2}} = \gamma^{\frac{1}{2}}$, we can consider the separation based on the data set projected onto the null space of $\gamma^{\frac{1}{2}}$, the popular spectral clustering indeed utilizes the first principal component of the projected data matrix to separate the data set. In fact, in [19], the authors suggested using maximal margin separation based on the first principal component of the projected matrix.

3 Maximal margin clustering with feature selection

In this section, we propose a new optimization model that can be viewed as a combination of feature selection and maximal margin clustering. For any vector $w \in \Re^d$, let $\mathcal{I}(w)$ denote the number of nonzero elements in w. For a given integer $k > 0$, we consider the following optimization problem

$$\begin{aligned} \min_{y_i \in \{1,-1\}} f(y) = \min\ & w^T w \\ s.t.\quad & y_i(w^T v_i + b) \geq 1, \quad i = 1, \ldots, n; \\ & \mathcal{I}(w) \leq k. \end{aligned} \tag{11}$$

In other words, we impose the condition $\mathcal{I}(w) \leq k$ on the solution of the maximum margin clustering problem. Since there are in total $C(d, k)$ different index sets of size k for a fixed k, we have

Theorem 2 *Suppose that the input data set satisfies Assumption 1. Then for any fixed k, we can find a global solution to problem* (11) *in* $O(C(d,k)n^{k+2})$ *time. Moreover, the objective value at the optimal solution of problem* (11) *is a decreasing function of k.*

The second conclusion in the above theorem sets up a cascade to approximate the original maximal margin clustering where $k = d$. The remaining difficulty is that solving problem (11) requires $O(C(d,k)n^{k+2})$ time and does not seem to be a very attractive option. To address this problem, let us recall (1)—it is straightforward to verify that

Theorem 3 *Let* $\mathcal{V} \in \Re^{d\times n}$ *be the data matrix such that each column of* $\mathcal{V}$ *represents a point in* $\Re^d$. *For any unitary matrix* $U \in \Re^{d\times d}$, *let* $\bar{\mathcal{V}} = U\mathcal{V}$ *be the data matrix after rotation. Then, the optimal solutions of problem* (1) *with respect to data matrices* $\mathcal{V}$ *and* $\bar{\mathcal{V}}$ *have precisely the* **same** *objective value.*

The above result gives us a simple and natural strategy to approximate the maximal margin clustering by using the eigenvalues and eigenvectors of matrix $\mathcal{V}$. Let us denote the eigenvalue decomposition (SVD) of $\mathcal{V}$ by

$$\mathcal{V} = U^T \Lambda Q, \quad U \in \Re^{d\times d}, \quad \Lambda \in \Re^{d\times n}, \quad Q \in \Re^{n\times n}$$

where U and Q are unitary matrices in suitable space, and

$$\Lambda = [\text{diag}\,(\lambda_1, \ldots, \lambda_d), 0]$$

where $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d$ are the eigenvalues of $\mathcal{V}$. We can use the product of the first k right eigenvectors (the first k columns of Q) with the corresponding k largest eigenvalues to project

Algorithm 1 Maximal Margin Clustering

MaxMarginClustering($\mathcal{V}, k$)
Input: $\mathcal{V} = \{v_i \in \Re^d : i = 1, \ldots, n\}$;
Output: Cluster label T;
begin
Compute the projected data set $\bar{\mathcal{V}} = \{\bar{v}_i \in \Re^k : i = 1, \ldots, n\}$,
Let $S = \{S_i \subset \bar{\mathcal{V}} : |S_i| = k, \}$ be the union of all the subsets of size k.
for every $S_i \in S$ **do**
  Find the hyperplane $h_i \in \Re^k$ induced by S_i.
  for $j = 1$ to n, $j \notin h_i$ **do**
    Calculate the distance from $\bar{v}_j$ to h_i, i.e., $\gamma_{ij} = L_2(\bar{v}_j, h_i)$.
  end for
end for
{Find the best separating hyperplane}
$h_{i*} = \arg \max_{i=1,\ldots,n} \min_{j=1,\ldots,n} \gamma_{ij}$;
for $j = 1$ to n **do**
  Assign the cluster label T_j based on the sign of r_{i*j};
end for
end

the data set into $\Re^k$, and then solve the maximal margin clustering problem in $\Re^k$. Notice that this closely resembles the well-known principal component analysis. For the graph-cut based clustering in Sect. 2.1, we must project the normalized data set instead. If we use the first k principal components, then the corresponding algorithm outlined in Algorithm 1 runs in $O(n^{k+2})$ time. Random Projection method based on the Johnson–Lindenstrauss (JL) lemma [22] can also be utilized, we present some experimental results in Sect. 5.

4 Successive approach

In the previous section, we discussed an approximate solution to the maximal margin clustering problem by means of feature selection. Though one might suspect that dimensionality reduction in this fashion restricts what we might be able to accomplish in a subsequent clustering step (due to loss of information), it is also widely held that many datasets can be well characterized in far fewer dimensions with only little distortion (if the choice of the lower dimensional space is done carefully), see [22]. For example, the JL lemma [22] has been successfully used for many machine learning problems. However, the application of our algorithm to dimensions $k \geq 4$ still remains a computational challenge, especially when key information regarding the underlying learning task is lost when projected to lower dimensional space. This difficulty can be tackled by an extension of the previous algorithm to one that uses successive feature selection for (1).

Our key idea is based on the following observation. Suppose we want to solve problem (1) by using a set of selected features $\mathcal{F} = \{f_1, f_2, \ldots, f_k\}$. Without loss of generality, we assume all features are ranked (using the eigenvalues, for example). We can start by solving problem (1) by using only the first two features and find the optimal separating hyperplane based on these two features. Note that associated with the optimal hyperplane are new features ($\bar{f}_1$ and $\bar{f}_2$) that can be *cast as a mixture of* f_1 *and* f_2, i.e., if the hyperplane, $h = af_1 + bf_2 + c$, then $\bar{f}_1 = af_1 + bf_2$ and $\bar{f}_2 = bf_1 - af_2$. We can rotate the features in the space (span(f_1, f_2)) spanned by f_1 and f_2 and thus derive a new set of features $\bar{\mathcal{F}} = \{\bar{f}_1, \bar{f}_2, \ldots, f_k\}$. From the construction of $\bar{f}_1$ and $\bar{f}_2 (h = 1\bar{f}_1 + 0\bar{f}_2 + c)$, we know that the feature $\bar{f}_2$ will not play a

role in the optimal separating hyperplane in the subspace span$(f_1, f_2) =$ span$(\bar{f}_1, \bar{f}_2)$. For simplicity of discussion, let us denote the updated feature set $\mathcal{F} = \{f_1, f_2, \ldots, f_k\} \leftarrow \bar{\mathcal{F}}$. Next, we pick the feature pair (f_1, f_3), find the optimal separating hyperplane based on the projected data set onto span(f_1, f_3) and update (f_1, f_3) in a similar manner. We can repeat the above process until all the features are scanned. The algorithm stops when no significant improvement can be obtained in the scan process or a prescribed number of scans have been performed. The algorithm can be summarized as shown in Algorithm 2.

Algorithm 2 (top) The successive approach for maximal margin clustering and (bottom) subroutine for scanning the ranked feature set

SuccessiveApproach$(\mathcal{V}, \mathcal{F}, \epsilon)$
Input: $\mathcal{V} = \{v_i \in \Re^d : i = 1, \ldots, n\}$, $\mathcal{F} = \{f_i \in \Re^n : i = 1, \ldots, d\}$ and a tolerance parameter ϵ;
Output: Cluster label T;
begin
Rank the features $\mathcal{F} = \{f_i : i = 1, \ldots, d\}$,
Set $h' = 0$
$\{\mathcal{F}, h, T\} \leftarrow$ Scan $(\mathcal{V}, \mathcal{F}, 2)$
$j \leftarrow 3$
while $h - h' > \epsilon$ **do**
  $h' = h$,
  $\{\mathcal{F}, h, T\} \leftarrow$ Scan $(\mathcal{V}, \mathcal{F}, j)$
  $j \leftarrow j + 1$
end while
Output the cluster label T found in the last iteration
end

Scan $(\mathcal{V}, \mathcal{F}, j)$
Input: $\mathcal{V} = \{v_i \in \Re^d : i = 1, \ldots, n\}$ and $\mathcal{F} = \{f_i \in \Re^n : i = 1, \ldots, d\}$ and index j of the current feature;
Output: $\mathcal{F} = \{f_i : i = 1, \ldots, d\}$, h and the cluster label set T due to h;
begin
Compute the projected data set $\bar{\mathcal{V}}$ of $\mathcal{V}$ onto the subspace span(f_1, f_j),
For $\bar{\mathcal{V}}$, find the normalized optimal separating hyperplane $a_j f_1 + b_j f_j + c_j = 0$
Set $f_1' = f_1$, $f_i' = f_i$
Update $f_1 \leftarrow a_j f_1' + b_j f_j'$
Update $f_j \leftarrow b_j \bar{f}_1' - a_j f_j'$
Compute the projected data set $\bar{\mathcal{V}}$ based on the updated subspace span(f_1, f_j);
Find the optimal separating hyperplane based on $\bar{\mathcal{V}}$
Calculate the corresponding maximal margin h and the cluster label set T
end

The complexity of the algorithm depends on the number of scans in the algorithm multiplied by the cost of every scan $O(dn^3)$. If we allow the algorithm to run only a few scans, then the total complexity of the algorithm remains $O(dn^3)$. We call such a process with a fixed number of scans as the truncated successive approach. Empirically, we found the truncated successive approach useful for extensions to the semi-supervised case in Sect. 4.2.

4.1 Convergence of the successive approach

Let us analyze the convergence of the proposed successive approach. First, we introduce the following definition.

Definition 1 Suppose that (y^*, w^*) is a vector pair with $y^* \in \{-1, 1\}^n$, $w^* \in \Re^d$.(y^*, w^*) is called a local minimizer of problem (1) if the following conditions are satisfied:

C.1 w^* is the optimal solution of the following problem

$$\begin{aligned} &\min w^T w \\ &s.t. \quad y_i^*(w^T v_i + b) \geq 1, \quad i = 1, \ldots, n. \end{aligned} \tag{12}$$

C.2 y^* is the optimal solution of the following problem:

$$\begin{aligned} &\min \alpha \\ &s.t. \quad y_i \left(\alpha v_i^T w^* + b\right) \geq 1, \quad i = 1, \ldots, n. \\ &\quad \alpha \geq 0, y_i \in \{-1, 1\}, \end{aligned} \tag{13}$$

and at the optimal solution we have $\alpha(y^*) = 1$.

Next, we show that the proposed successive approach will terminate at a local minimizer of problem (1) if the prescribed parameter in the algorithm, $\epsilon = 0$. First, observe that if all the features in $\mathcal{F}$ are orthogonal to each other, then the updated features remain orthogonal to each other after one scan since the orthogonality among the features is preserved due to the special update rule in the algorithm. Second, we note that the algorithm stops whenever the maximal margin separation does not change. For notational convenience, let us express any vector v in term of the features $f_1, \ldots, f_d$, i.e., $v = v^1 f_1 + v^2 f_2 \cdots + v^d f_d$. Note that at the beginning of every scan, we can partition the entire data set into two clusters by applying the maximal margin separation to the projected data onto span(f_1), or in other words, separate the data set based on the values of the first coordinate of the data points. Let us denote the corresponding cluster labels by $y^* \in \{-1, 1\}^n$. Then, the maximal margin separation is derived by solving the following problem:

$$\begin{aligned} &\min w_1^2 \\ &s.t. \quad y_i^* \left(w_1 v_i^1 + b\right) \geq 1, \quad i = 1, \ldots, n. \end{aligned} \tag{14}$$

Let us denote the optimal solution of the above problem by w_1^*. If the maximal margin separation stays invariant during the entire scanning process, then for any $j = 2, \ldots, d$, $(w_1^*, w_j^* = 0)$ is also the optimal solution of the following problem

$$\begin{aligned} &\min w_1^2 + w_j^2 \\ &s.t. \quad y_i^* \left(w_1 v_i^1 + w_j v_i^j + b\right) \geq 1, \quad i = 1, \ldots, n. \end{aligned} \tag{15}$$

Recall that $(w_1^*, 0, \ldots, 0)^T$ is a feasible solution for problem (12). The above fact also implies that $(w_1^*, 0, \ldots, 0)^T$ is a *stationary point* along every coordinate direction corresponding to the feature f_j, $j = 2, \ldots, d$. Consequently, it is also a stationary point for problem (12). It follows from convexity theory that $(w_1^*, 0, \ldots, 0)^T$ is an optimal solution to problem (12).

The following theorem summarizes the main results in this section.

Theorem 4 *Suppose that the parameter ϵ equals zero. Then the proposed successive approach will converge to a local minimum of problem* (1) *in finite number of steps.*

Proof The convergence to a local minimum follows from the discussion preceding the theorem, while the finite convergence is due to the fact that the number of local minimizers of problem (1) is finite. □

4.2 Semi-supervised case

Observe that the performance of the above algorithm depends on the set of ranked features—the eigenvectors ordered based on their eigenvalues can be used as the set of features. If a projection onto the space spanned by the leading eigenvectors does not induce a good margin, a semi-supervised approach can be used partly based on ideas from the previous section. Assume the dataset $D = D_1 \cup D_2$, where D_1 is the set of labeled items, and D_2 denotes the set of unlabeled items. Let $\mathcal{V} = (v_1, v_2, \ldots, v_l)$ be the set of eigenvectors (for non-zero eigenvalues) of D. We select all pairs of eigenvectors from $\mathcal{V}$, and then evaluate the max-margin algorithm after projecting the dataset in the space spanned by each pair. Besides the margin, we also determine the misclassification error on D_1 in each case. For instance, if v_i and v_j are the two vectors corresponding to the minimum misclassification error, and if the margin is greater than ϵ, we combine them into a single feature $\bar{v}$. The process of evaluating pair-wise features can now be repeated for $\bar{v}$ and the remaining vectors in $\mathcal{V}$ (assuming v_i and v_j have been removed), this determines which vector will be combined with $\bar{v}$. We continue this process until the margin returned is less than ϵ. The time complexity in this case is $O(l^2 T)$, where T is the time required for the max-margin algorithm. In practice, this approach works very well, since l is usually small for most datasets.

5 Experiment results

In this section, we summarize our experimental evaluation results. We discuss some toy examples, as well as simulation and publicly available data sets. We also report on comparisons with other clustering algorithms (with comparable running times) like k-means clustering (10 restarts) and a spectral clustering algorithm [19]. For the smaller datasets, where running a SDP solver was feasible, we report on comparisons with the max-margin algorithm in [3], for convenience this is summarized at the end of this section. We used both two class and multi-class datasets (using one versus all scheme). To avoid degenerate solutions, we used class balance constraints by only considering S_i (see Algorithm 1) which satisfy the class bounds. We performed clustering either in the original space or in lower dimensions (after projection), kernels were used for some datasets and are noted where applicable.

5.1 Toy Examples

Our first set of simulation experiments were performed on four toy examples. The results of our max-margin algorithm, k-means and spectral clustering [19] are summarized in Fig. 1. Since the two circles and the joined circles dataset (the two right-most images in Fig. 1) are not linearly separable, we used a RBF kernel before obtaining a lower dimension ($3D$) projection, the $3D$ distribution was then used as an input to the max-margin algorithm. For the examples in Fig. 1 our technique was able to separate the point sets precisely, except in the joined circles data set (error $\sim$2%). For the Letters (HELLO), Letters (AI), and two circles datasets, our algorithm matched the performance of the algorithm in [3]. For the joined circles distribution, our algorithm misclassified just one additional point relative to [3].

5.2 Effect of dimensionality reduction

We now demonstrate the effect of dimensionality reduction in maximal margin clustering and analyze the dependence on d. We generated several two class normally distributed data [23] in

Dataset	Ours	k-means	spectral [19]
Letters (HELLO)	0	0.18	0.11
Two circles	0	0.48	0.5
Joined circles	0.02	0.5 [3]	0.3
Letters (AI)	0	0.38 [3]	0

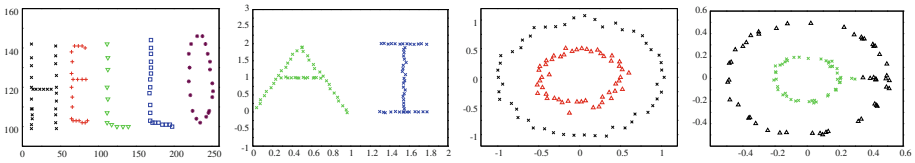

Fig. 1 Misclassification errors of our maximal margin clustering algorithm, k-means, and spectral clustering [19] on toy examples—letters (HELLO), letters (AI), two circles and joined circles, different clusters discovered by our algorithm indicated by *different markers* (and *different colors*)

Table 1 Misclassification errors of our maximal margin clustering algorithm when the high dimensional dataset is projected onto few dimensions

Original	$1D$	$2D$	$3D$	$4D$	k-Means	Spectral [19]
$6D$	0.36	0.25	0.21	0.13	0.23	0.36
$8D$	0.40	0.37	0.34	0.31	0.32	0.44
$10D$	0.44	0.42	0.33	0.23	0.46	0.32
$12D$	0.47	0.46	0.34	0.26	0.38	0.44

{6, 8, 10, 12} dimensions, these were projected onto {1, 2, 3, 4} dimensions. The misclassification errors are shown in Table 1. As can be expected, the classification error improves from the projection in $1D$ through $4D$ near-linearly. The performance of the technique is almost always better than k-means or spectral clustering [19] when projected to $3D$ or higher.

5.3 Multi-class clustering

The max-margin clustering technique can also be applied on multi-class dataset where we recursively bi-partition the data until the desired number of clusters are obtained. The letters dataset in Fig. 1 is an example of such an approach. We further evaluated the algorithm on two additional multi-class datasets. The first data set is the Iris dataset from UCI machine learning repository, where there are three classes (no. of instances = 150, no. of features = 4). The other dataset is the 8D2K dataset http://strehl.com/download/x8d5k.txt) with five classes (no. of instances = 1,000, no. of features = 8). The results are illustrated in Table 2.

We also evaluated our approach in context of unsupervised learning tasks on several two class data sets from the UCI machine learning repository (see http://archive.ics.uci.edu/ml). Specifically, we used the (a) voting-records data set (no. of instances = 435, no. of features = 16), (b) hepatitis data set (no. of instances = 155, no. of features = 19), and (c) heart data set (no. of instances = 270, no. of features = 11). Note that no training data was used. The data sets were projected onto 2 and 3 dimensions using Random Projections (RP), using the construction proposed in [24]. The results are shown in Table 3. As expected, the algorithm performs better when the data is projected onto $3D$ space, however it is much faster in the $2D$ case. Nonetheless, it performs better than conventional k-means clustering and spectral clustering [19] in almost all cases.

Table 2 Misclassification errors of our maximal margin clustering algorithm is used for multi-class datasets

Data	$1D$	$2D$	$3D$	k-Means	Spectral [19]
Iris	0.07	0.04	0.04	0.11	0.10
8D2K	0.25	0.02	0.01	0.09	0

5.4 Labeled and unlabeled data

We evaluated our semi-supervised learning max-margin algorithm on these data sets (using PCA). In this case, the items were divided equally into labeled and unlabeled categories. At each step, the classification accuracy of the labeled items was used to select the eigenvectors. In most cases, the combination of eigenvectors converged within six iterations. We report on the misclassification errors for the unlabeled set. Results shown in Table 3 (column 6) show a significant improvement over the results obtained for the unsupervised case.

The final set of experiments were performed in context of an application of clustering for face recognition. We chose this application because the UMist dataset for faces was used in [3] to evaluate maximum margin clustering. We report our results on the UMist faces dataset [25] and also the ORL face dataset [26].

5.5 UMist data

The UMist data set is comprised of gray-scale images (112×92) (multiple views) of 20 different people, a total of 575 images. We chose two human subjects at a time yielding 190 pairs for all images in the database. Each of these images were converted to a 10304 dimensional feature point. Typical examples of some of these images are shown in Fig. 2. The error rate was $\sim$4% over the entire dataset, and the algorithm matches the performance reported in [3].

5.6 ORL data

The ORL dataset comprises of 400 images of 40 subjects (10 images per subject). The images reflect variations in pose, expression, illumination and scale. The image sizes were 32×32, these were rescaled to yield a $1024D$ feature vector for each image. For our evaluations, we selected a pair of faces for each execution. The feature vectors were projected on to $1D$, $2D$, $3D$ using PCA and then clustering was performed. We also evaluated the semi-supervised algorithm on this dataset. As is seen in Fig. 3, the performance is similar to the UMist Dataset, with the mean error of 3–4% in both the unsupervised and supervised cases.

Table 3 Misclassification errors of our algorithm, k-means and spectral clustering [19] on UCI machine learning repository datasets. PCA was used only for the semi-supervised case

Data set (dimensions)	$2D$	$3D$	k-Means	Spectral [19]	Semi-supervised
Heart ($11D$)	0.37	0.35	0.40	0.48	0.26
Voting records ($16D$)	0.42	0.38	0.40	0.47	0.21
Hepatitis ($19D$)	0.33	0.31	0.36	0.46	0.26

Fig. 2 Sample face images from UMist face dataset

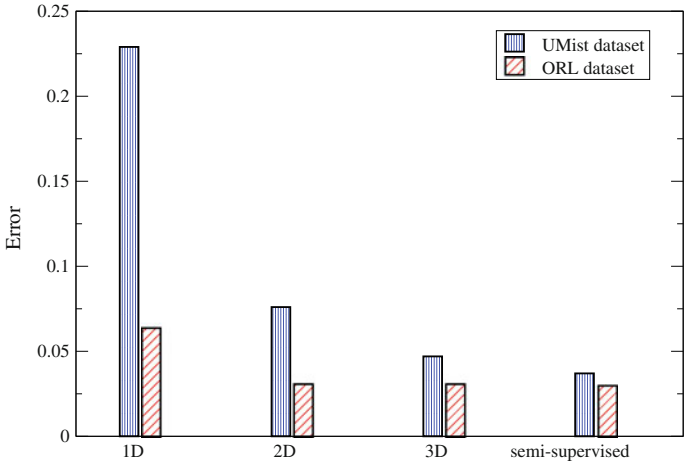

Fig. 3 Plot of average misclassification errors on face recognition tasks in UMist and ORL datasets

A word on the running time of the algorithm. The $1D$ and $2D$ cases take less than 1 s, the $3D$ case takes under 20 s on a Pentium IV 2.8 GHz machine in Matlab with the current implementation. As we discussed before, the computational cost for the $3D$ case is approximately the same as what one must expend in *one* iteration of a SDP-based approach for this problem (note that for larger datasets even obtaining the solution to the SDP model using standard solvers becomes infeasible). We evaluated our unsupervised algorithm (in dimensions $1D$, $2D$, $3D$ using PCA) and semi-supervised algorithm on all pairs of faces, the means of these values can be seen in Fig. 3. The mean error for the unsupervised case (in the $3D$) case is about 4.75%, where as the semi-supervised algorithm reported a mean error of about 3.74%.

5.7 Summary of comparisons with [3]

Here, we summarize the comparisons of our approach with the SDP model described in [3]. Direct comparison of all data sets presented in the previous section with the SDP approach is difficult because of the computational requirements of SDP solvers. However, we discuss comparisons on a subset of datasets. Results of our algorithm on three toy examples—Letters

Table 4 Misclassification plots

Summary of comparisons of our max-margin algorithm with [3] on some datasets

Dataset	Ours	Max-margin [3]
Letters (AI)	0	0
Two circles	0	0
Joined circles	0.02	0.01
Vote (Supervised)	0.21	0.14

(AI), Two Circles and Joined Circles are repeated in Table 4, alongside the results on these data sets reported in [3]. Among the UCI data sets, [3] reports on the performance accuracy on part of the voting records data set using their supervised version of the max-margin algorithm, where as our results are based on the entire dataset. Apart from the improvements in running time, it is encouraging that our approach achieves comparable accuracy on almost all data sets, when compared with the SDP approach.

6 Conclusions

In this paper, we first discuss the complexity of maximal margin clustering. Then, using some geometric ideas, we show that the proposed clustering algorithm based on feature selection and dimension reduction techniques works quite well in practice, both in terms of the quality of solutions generated and the running time. This yields a new way to approximately solve the maximal margin clustering problems for large scale data sets which has provable convergence properties. As the reader may have noticed, the algorithm is very easy to implement and we believe is applicable to a wide variety of clustering problems. It can be used either as stand-alone or as a "warm-up" to sophisticated algorithms that benefit from a good initial solution as a starting point.

Acknowledgments The authors would like to thank the anonymous referees and editors for their useful suggestions that improve an early version of the paper. The research of this work is supported by AFOSR grant FA9550-09-1-0098 and NSF grant DMS 09-15240 ARRA. Singh was supported in part by NIH grants R21-AG034315, R01-AG021155, UW ICTR (1UL1RR025011), and UW ADRC (P50-AG033514).

References

1. Vapnik, V.: The Nature of Statistical Learning Theory. Springer, New York (1995)
2. Schölkopf, B., Smola, A.: Learning with Kernels Support Vector Machines, Regularization, Optimization and Beyond. MIT Press, Cambridge (2002)
3. Xu, L., Neufeld, J., Larson, B., Schuurmans, D.: Maximum margin clustering. In: Advances in Neural Information Processing Systems (NIPS). MIT Press, Cambridge (2005)
4. Xu, L., Schuurmans, D.: Unsupervised and semi-supervised multi-class support vector machines. In: Proceedings of the 20th National Conference on Artificial Intelligence (AAAI) (2005)
5. Valizadegan, H., Jin, R.: Generalized maximum margin clustering and unsupervised kernel learning. In: Advances in Neural Information Processing Systems (NIPS). MIT Press, Cambridge (2006)
6. Bennett, K., Demiriz, A.: Semi-supervised support vector machines. In: Advances in Neural Information Processing Systems (NIPS). MIT Press, Cambridge (1998)
7. Bie, T. D., Cristianini, N.: Convex methods for transduction. In: Advances in Neural Information Processing Systems (NIPS). MIT Press, Cambridge (2004)
8. De Bie, T., Cristianini, N.: Fast SDP relaxations of graph cut clustering, transduction, and other combinatorial problems. J. Mach. Learn. Res. **7**, 1409–1436 (2006)
9. Vandenberghe, L., Boyd, S.: Semidefinite programming. SIAM Rev. **38**, 49–95 (1996)

10. Zhang, K., Tsang, I.W., Kwok, J.T.: Maximum margin clustering made practical. In: International Conference on Machine learning (ICML), pp. 1119–1126 (2007)
11. Ng, A., Jordan, M., Weiss, Y.: On spectral clustering: analysis and an algorithm. In: Advances in Neural Information Processing Systems (NIPS). MIT Press, Cambridge (2002)
12. Prokopyev, O.A., Busygin, S., Pardalos, P.M.: An optimization based approach for data classification. Optim. Methods Softw. **2**, 3–9 (2007)
13. Xu, R., Wunsch, D.: Survey of clustering algorithms. IEEE Trans. Neural Netw. **16**, 645–648 (2005)
14. Sherali, H.D., Desai, J.: A global optimization rlt-based approach for solving the hard clustering problem. J. Global Optim. **32**, 281–306 (2005)
15. Sherali, H.D., Desai, J.: A global optimization rlt-based approach for solving the fuzzy clustering approach. J. Global Optim. **33**, 597–615 (2005)
16. Butenko, S., Chaovalitwongse, W., Pardalos, P.M.: Clustering Challenges in Biological Networks. World Scientific, Singapore (2009)
17. Bradley, P.S., Mangasarian, O.L.: k-plane clustering. J. Global Optim. **16**, 23–32 (2000)
18. Du, D., Jung, Y., Park, H., Drake, B.L.: A decision criterion for the optimal number of clusters in hierarchical clustering. J. Global Optim. **25**, 91–111 (2003)
19. Shi, J., Malik, J.: Normalized cut and image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **22**, 888–905 (2000)
20. Dhillon, I.S., Guan, Y., Kulis B.: Kernel k-means, spectral clustering and normalized cuts. In: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), pp. 551–556 (2004)
21. Chen, H., Peng, J.: 0–1 semidefinite programming for graph-cut clustering: modelling and approximation. In: Pardalos, P.M., Hansen, P. (eds.) Data Mining and Mathematical Programming. CRM Proceedings and Lecture Notes of the American Mathematical Society, pp. 15–40 (2008)
22. Johnson, W. B., Lindenstrauss, J.: Extensions of lipshitz mapping into hilbert space. Contemp. Math. **26**, 189–206 (1984)
23. Musicant, D. R.: NDC: normally distributed clustered datasets (1998) http://www.cs.wisc.edu/dmi/svm/ndc/
24. Achlioptas, D.: Database-friendly random projections. In: ACM Symposium on Principles of Database Systems (PODS), pp. 274–281 (2001)
25. Graham, D. B., Allinson, N. M.: In: Face Recognition: From Theory to Applications, vol. 163, chapter Characterizing Virtual Eigensignatures for General Purpose Face Recognition. NATO ASI Series F, Computer and Systems Sciences, pp. 446–456 (1998)
26. Samaria, F., Harter, A.: Parameterisation of a stochastic model for human face identification. In: Proceedings of 2nd IEEE Workshop on Applications of Computer Vision (1994)