Unsupervised Maximum Margin Feature Selection with Manifold Regularization

Bin Zhao[†], James Kwok[‡], Fei Wang[†], Changshui Zhang[†]
[†]State Key Laboratory of Intelligent Technology and Systems
Tsinghua National Laboratory for Information Science and Technology
Department of Automation, Tsinghua University, Beijing, China
[‡]Department of Computer Science and Engineering, HKUST, Hong Kong

Abstract

Feature selection plays a fundamental role in many pattern recognition problems. However, most efforts have been focused on the supervised scenario, while unsupervised feature selection remains as a rarely touched research topic. In this paper, we propose Manifold-Based Maximum Margin Feature Selection (M3FS) to select the most discriminative features for clustering. M3FS targets to find those features that would result in the maximal separation of different clusters and incorporates manifold information by enforcing smoothness constraint on the clustering function. Specifically, we define scale factor for each feature to measure its relevance to clustering, and irrelevant features are identified by assigning zero weights. Feature selection is then achieved by the sparsity constraints on scale factors. Computationally, M3FS is formulated as an integer programming problem and we propose a cutting plane algorithm to efficiently solve it. Experimental results on both toy and real-world data sets demonstrate its effectiveness.

1. Introduction

Real-world data sets are often high-dimensional and contain many spurious features. For example, in face recognition, an image of size $m \times n$ is often represented as a vector in $\mathbb{R}^{mn}$, which can be very high-dimensional for typical values of m and n. Similarly, biological databases such as microarray data can have thousands or even tens of thousands of genes as features. Such a large number of features can easily lead to the curse of dimensionality and severe over-fitting. Hence, dimensionality reduction, in the form of either feature extraction or feature selection, plays a fundamental role in many pattern recognition problems.

In this paper, we will focus on feature selection, which selects a relevant subset of features. Excellent reviews on this topic can be found in [8, 10]. Note that not only can feature selection improve the generalization performance of the resultant classifier, the use of fewer features is also less computationally expensive and thus implies faster testing. Moreover, it can eliminate the need to collect a large number of irrelevant and redundant features, and thus reduces the cost. Finally, with the discovery of fewer features, the resultant model can be more easily understood by human.

In feature selection, the features may be scored either individually or as a subset. In general, there are three approaches to score them: filters, wrappers, and embedded methods [8]. Filters score the features as a pre-processing step, independently of the classifier. Wrappers score the features according to their prediction performance when used with the classifier. Both filters and wrappers rely on search strategies to guide the search for the "best" feature subset. While a large number of search strategies can be used, often one is limited to the computationally simple greedy (forward or backward) strategies. Finally, embedded methods combine feature selection with the classifier. While the design of embedded methods is tightly coupled with the specific classifier, they are often considered as more efficient than filters and wrappers [8].

While supervised feature selection has been extensively studied for decades, feature selection in the unsupervised learning setting has received relatively little attention. This is partly due to the fact that unsupervised feature selection is much more difficult because of the lack of label information to guide the search for relevant features. While most unsupervised feature selection methods are based on the filter approach [6, 12, 14], some wrappers [16] and embedded approaches that treat clustering and feature selection simultaneously have also been proposed [4, 7, 12]. However, these are often based on generative models (such as Gaussian mixtures) [4, 7, 12, 16]. As is well-known, generative models may lead to inferior performance when the model assumption does not match the observed data.

Instead of relying on model-based clustering, we will propose in this paper an embedded method that is based on discriminative clustering. This is motivated by the common belief that discriminative models are often better than gen-

978-1-4244-3991-1/09/$25.00 ©2009 IEEE

erative models in supervised learning. Among the discriminative methods, large margin methods, such as the support vector machines, are particularly successful. Indeed, inspired by the superiority of large margin methods in supervised learning, there is growing interest in extending them to unsupervised learning. For example, Xu *et al*. [21] proposed a novel approach called *maximum margin clustering (MMC)*, which performs clustering by simultaneously finding the large margin separating hyperplane between clusters. Experimental results showed that this large margin clustering method (and its variant [19]) have been very successful in many clustering problems.

Moreover, in many computer vision and pattern recognition applications (such as face recognition and hand-written digit recognition), it has been observed that the data examples often lie on a manifold. Hence, another novelty of the proposed approach is that manifold information can also be incorporated into the feature selection process. Note that the Laplacian score [9], which can be used as a filter approach for unsupervised feature selection, also utilizes manifold information. However, for the Laplacian score, a feature will be considered as good if two samples that are close to each other on the data manifold are also close to each other according to that feature. On the other hand, the proposed method uses the manifold information by directly considering the resultant decision function and ensures that it is smooth on the manifold. As will be seen in Section 4, since ours is an embedded method that explicitly considers the clustering objective, it performs much better than the filter method of Laplacian score.

In this paper, we propose *Manifold-Based Maximum Margin Feature Selection (M3FS)* to select the most discriminative features for clustering. *M3FS* targets to find those features that would result in the maximal separation of different clusters and incorporates manifold information by enforcing smoothness constraint on the clustering function. Specifically, we define a scale factor for each feature to measure its relevance to clustering, and irrelevant features are identified by assigning zero weights. Feature selection is then achieved by the sparsity constraints on the scale factors. Computationally, *M3FS* is formulated as an integer programming problem and we propose a cutting plane algorithm to efficiently solve it. Experimental results on both toy and real-world data sets demonstrate its effectiveness.

The rest of this paper is organized as follows. In Section 2, we present a brief introduction to *maximum margin clustering*. Section 3 presents the details of the *M3FS* algorithm, together with theoretical analysis on both the accuracy and time complexity of the algorithm, and extension to the multi-class clustering setting. Experimental results on both toy and real-world data sets are provided in Section 4, followed by some concluding remarks in Section 5.

2. Maximum Margin Clustering

Maximum margin clustering (MMC) is a recently proposed clustering algorithm that extends *support vector machines* (*SVM*) to unsupervised learning setting. Since the class labels are unknown in unsupervised learning, *MMC* tries to find a cluster labeling of the patterns, together with a hyperplane classifier, such that the resultant margin is maximized among all possible labelings [21].

For simplicity of exposition, assume that there are only two clusters. Given a set of examples $\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_n] \in \mathbb{R}^{d\times n}$, *MMC* targets to find the best label combination $\mathbf{y} = [y_1, \ldots, y_n] \in \mathbb{R}^n \in \{-1, +1\}^n$ such that an *SVM* trained on this $\{(\mathbf{x}_i, y_i), \ldots, (\mathbf{x}_n, y_n)\}$ yields the largest margin. Computationally, it can be formulated as the following problem

$$\begin{aligned} \min_{\mathbf{y}\in\{\pm 1\}^n} \min_{\mathbf{w},b,\boldsymbol{\xi}} \quad & \frac{1}{2}\mathbf{w}^T\mathbf{w} + \frac{C}{n}\sum_{i=1}^{n}\xi_i \\ \text{s.t.} \quad & \forall i \in \{1, \ldots, n\} : \\ & y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i,\ \xi_i \geq 0, \\ & -l \leq \sum_{i=1}^{n} y_i \leq l. \end{aligned} \tag{1}$$

where $\sum_{i=1}^{n}\xi_i$ is divided by n to better capture how C scales with the data set size. The last constraint in (1) is often known as the *class balance* constraint. It is introduced to avoid the trivially "optimal" solution that assigns all patterns to the same class and thus achieves "infinite" margin. Here, $l > 0$ is a constant controlling the class imbalance.

3. Maximum Margin Feature Selection with Manifold Regularization

In this section, we present the *manifold-based maximum margin feature selection* algorithm. We will first consider the two-cluster case. Extension to the multi-class case will be discussed in Section 3.5.

3.1. Two-Class Manifold-Based Maximum Margin Feature Selection

Manifold-based maximum margin feature selection (M3FS) is an embedded approach that performs clustering and feature selection simultaneously. It tries to find a subset of the d given features such that the resultant clusters will be maximally separated. As mentioned in Section 1, while previous efforts on unsupervised feature selection are often based on generative models which require strong model assumption, *M3FS* adopts *maximum margin clustering* which can often outperform conventional clustering methods.

Moreover, in many computer vision and pattern recognition applications, it has been observed that the data exam-

ples often lie on a manifold. Hence, our goal is to also utilize this manifold information in the feature selection process. As is well-known, the data manifold can be represented by a graph $\mathcal{G}$. In the following, let $\mathbf{W} \in \mathbb{R}^{n\times n}$ be the similarity (or adjacency) matrix of $\mathcal{G}$, $\mathbf{D} \in \mathbb{R}^{n\times n}$ be the diagonal degree matrix whose ith entry is the sum of the ith row of $\mathbf{W}$, and $\mathbf{L} = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}}\mathbf{W}\mathbf{D}^{-\frac{1}{2}}$ (where $\mathbf{I}$ is the $n \times n$ identity matrix) be the normalized graph Laplacian [3].

To achieve the first goal, we extend *MMC* by associating each feature k ($k = 1, 2, \ldots, d$) with a learnable scale factor σ_k, which is used to measure its "relevance" to clustering. When learning is completed, the irrelevant features can then be identified as those having zero scale factors [20]. Hence, the resultant decision function is $f(\mathbf{x}) = \mathbf{w}^T(\boldsymbol{\sigma} \circ \mathbf{x}) + b = (\mathbf{w} \circ \boldsymbol{\sigma})^T\mathbf{x} + b$, where $\boldsymbol{\sigma} = [\sigma_1, \sigma_2, \ldots, \sigma_d]^T$ and $\circ$ is the element-wise product. As for the second goal, we enforce that the decision function $f(\mathbf{x})$ is smooth on the whole data manifold. This smoothness can be achieved by adding the manifold regularizer [1]

$$\sum_{i,j=1}^{n} W_{ij}\left(\frac{f(\mathbf{x}_i)}{\sqrt{D_{ii}}} - \frac{f(\mathbf{x}_j)}{\sqrt{D_{jj}}}\right)^2 = \left[\mathbf{X}^T(\mathbf{w}\circ\boldsymbol{\sigma}) + b\mathbf{1}\right]^T \mathbf{L}\left[\mathbf{X}^T(\mathbf{w}\circ\boldsymbol{\sigma}) + b\mathbf{1}\right]$$

to the objective function. Here, $\mathbf{1} \in \mathbb{R}^n$ is the n-dimensional vector of all ones. Combining these two together, *M3FS* can thus be formulated as the following optimization problem:

$$\min_{\mathbf{y},\mathbf{w},b,\xi,\boldsymbol{\sigma}} \ \frac{1}{2}\sum_{k=1}^{d}\sigma_k w_k^2 + \frac{C}{n}\sum_{i=1}^{n}\xi_i + \lambda\left[\mathbf{X}^T(\mathbf{w}\circ\boldsymbol{\sigma})+b\mathbf{1}\right]^T\mathbf{L}\cdot\left[\mathbf{X}^T(\mathbf{w}\circ\boldsymbol{\sigma})+b\mathbf{1}\right] \tag{2}$$

$$\text{s.t.} \ \ \forall i \in \{1,\ldots,n\} : \xi_i \geq 0,$$

$$y_i\left(\sum_{k=1}^{d}\sigma_k w_k x_{ik} + b\right) \geq 1 - \xi_i, \tag{3}$$

$$\forall k \in \{1,\ldots,d\} : 0 \leq \sigma_k \leq 1; \mathbf{y} \in \{-1,+1\}^n$$

$$\sum_{k=1}^{d}\sigma_k = m, \tag{4}$$

$$-l \leq \sum_{i=1}^{n}\left(\sum_{k=1}^{d}\sigma_k w_k x_{ik} + b\right) \leq l \tag{5}$$

where λ is a user-defined regularization parameter, and m is the number of features to be selected. Note that we have also relaxed the constraint $\sigma_k \in \{0, 1\}$ on $\boldsymbol{\sigma}$ to $0 \leq \sigma_k \leq 1$. The ℓ_1 regularizer (4) on $\boldsymbol{\sigma}$ enforces sparsity. Moreover, a slightly relaxed class balance constraint is used in (5) [17].

Since σ_k and w_k are coupled together in the decision function, the objective in (2) and the constraints (3), (5) are non-convex. Therefore, we apply the change of variables [24]: $\forall k \in \{1,\ldots,d\} : v_k = \sigma_k w_k$. Let $\mathbf{v} = [v_1, v_2, \ldots, v_d]^T$, we have the following proposition:

Proposition 1 *M3FS can be equivalently formulated as*

$$\min_{\mathbf{v},b,\xi,\boldsymbol{\sigma}} \ \frac{1}{2}\sum_{k=1}^{d}\frac{v_k^2}{\sigma_k} + \frac{C}{n}\sum_{i=1}^{n}\xi_i + \lambda(\mathbf{X}^T\mathbf{v}+b\mathbf{1})^T\mathbf{L}(\mathbf{X}^T\mathbf{v}+b\mathbf{1}) \tag{6}$$

$$s.t. \ \forall i \in \{1,\ldots,n\} : \left|\mathbf{v}^T\mathbf{x}_i + b\right| \geq 1-\xi_i, \ \xi_i \geq 0$$

$$\forall k \in \{1,\ldots,d\} : \ 0 \leq \sigma_k \leq 1,$$

$$\sum_{k=1}^{d}\sigma_k = m, \ -l \leq \sum_{i=1}^{n}\left(\mathbf{v}^T\mathbf{x}_i + b\right) \leq l$$

where $\mathbf{y}$ *is calculated as* $y_i = \mathrm{sgn}(\mathbf{v}^T\mathbf{x}_i + b)$.

3.2. Cutting Plane Algorithm

The *M3FS* formulation in (6) has n slack variables ξ_i's, one for each data sample $\mathbf{x}_i$. We reformulate (6) as follows to reduce the number of slack variables,

$$\min_{\mathbf{v},b,\xi,\boldsymbol{\sigma}} \ \frac{1}{2}\sum_{k=1}^{d}\frac{v_k^2}{\sigma_k} + C\xi + \lambda(\mathbf{X}^T\mathbf{v}+b\mathbf{1})^T\mathbf{L}(\mathbf{X}^T\mathbf{v}+b\mathbf{1}) \tag{7}$$

$$\text{s.t.} \ \forall \mathbf{c} \in \{0,1\}^n : \frac{1}{n}\sum_{i=1}^{n}c_i\left|\mathbf{v}^T\mathbf{x}_i+b\right| \geq \frac{1}{n}\sum_{i=1}^{n}c_i - \xi, \tag{8}$$

$$\forall k \in \{1,\ldots,d\} : 0 \leq \sigma_k \leq 1,$$

$$\sum_{k=1}^{d}\sigma_k = m, \ \xi \geq 0, \ -l \leq \sum_{i=1}^{n}\left(\mathbf{v}^T\mathbf{x}_i+b\right) \leq l$$

Proposition 2 *Any solution* $(\mathbf{v}^*, b^*, \xi^*, \boldsymbol{\sigma}^*)$ *to problem (7) is also a solution to problem (6), and vice versa, with* $\xi^* = \frac{1}{n}\sum_{i=1}^{n}\xi_i^*$.

The number of slack variables is now reduced by $n - 1$. On the other hand, the number of constraints in (7) is increased from n to 2^n. To handle this exponential number of constraints, we employ an adaptation of the *cutting plane* algorithm [11]. It starts with an empty constraint subset Ω, and computes the optimal solution to problem (7) subject to the constraints in Ω. The algorithm then finds the most violated constraint in (8) and adds it to Ω. In this way, we construct a series of successively tightening approximations to problem (7). The algorithm stops when no constraint in (8) is violated by more than ϵ. The whole *cutting plane* algorithm for *M3FS* is presented in Algorithm 1.

3.2.1 Optimization via the CCCP

For the optimization problem in (7), the objective is convex (quadratic) and all the constraints except the first one are linear. Moreover, note that although the constraint in (8) is

Algorithm 1 Cutting plane algorithm for M3FS

Input: $\mathbf{X}$, C, l, λ and ϵ, set constraint subset $\Omega = \emptyset$.
repeat
Solve problem (7) for $(\mathbf{v}, b, \boldsymbol{\sigma}, \xi)$ under the current working constraint set Ω.
Select the most violated constraint $\mathbf{c}$; set $\Omega = \Omega \cup \{\mathbf{c}\}$.
until the newly selected constraint $\mathbf{c}$ is violated by no more than ϵ.

non-convex, it can be expressed as a difference of the two convex functions $\frac{1}{n}\sum_{i=1}^{n} c_i \left|\mathbf{v}^T\mathbf{x}_i + b\right|$ and $\frac{1}{n}\sum_{i=1}^{n} c_i - \xi$. Hence, we can solve problem (7) with the *constrained concave-convex procedure (CCCP)*, which is designed to solve these optimization problems with a concave-convex objective function and concave-convex constraints [18]. Specifically, given an initial estimate $(\mathbf{v}^{(0)}, b^{(0)})$, the *CCCP* computes $(\mathbf{v}^{(t+1)}, b^{(t+1)})$ from $(\mathbf{v}^{(t)}, b^{(t)})$ by replacing $\frac{1}{n}\sum_{i=1}^{n} c_i \left|\mathbf{v}^T\mathbf{x}_i + b\right|$ in the first constraint with its first-order Taylor expansion at $(\mathbf{v}^{(t)}, b^{(t)})$, leading to

$$\begin{aligned}
\min_{\mathbf{v},b,\xi,\boldsymbol{\sigma}} \; & \frac{1}{2}\sum_{k=1}^{d}\frac{v_k^2}{\sigma_k} + C\xi + \lambda(\mathbf{X}^T\mathbf{v} + b\mathbf{1})^T\mathbf{L}(\mathbf{X}^T\mathbf{v} + b\mathbf{1}) \qquad (9)\\
\text{s.t. } \forall \mathbf{c} \in \Omega\colon & \frac{1}{n}\sum_{i=1}^{n} c_i z_i^{(t)}\left(\mathbf{v}^T\mathbf{x}_i + b\right) \geq \frac{1}{n}\sum_{i=1}^{n} c_i - \xi,\\
& \forall k \in \{1, \ldots, d\}:\ 0 \leq \sigma_k \leq 1,\\
& \sum_{k=1}^{d} \sigma_k = m,\ \xi \geq 0\\
& -l \leq \sum_{i=1}^{n}\left(\mathbf{v}^T\mathbf{x}_i + b\right) \leq l
\end{aligned}$$

where $z_i^{(t)} = \operatorname{sgn}\left(\mathbf{v}^{(t)T}\mathbf{x}_i + b\right)$. Define t_k as the upper bound of $\frac{v_k^2}{\sigma_k}$, s as the upper bound of $(\mathbf{X}^T\mathbf{v} + b\mathbf{1})^T\mathbf{L}(\mathbf{X}^T\mathbf{v} + b\mathbf{1})$, and note that $\mathbf{L}$ is symmetric positive semi-definite, the above problem can be reformulated as the following *second order cone programming (SOCP)* [2].

$$\begin{aligned}
\min_{\mathbf{v},b,\xi,\boldsymbol{\sigma},\mathbf{t},s} \; & \frac{1}{2}\sum_{k=1}^{d} t_k + C\xi + \lambda s \qquad (10)\\
\text{s.t. } \forall \mathbf{c} \in \Omega\colon & \frac{1}{n}\sum_{i=1}^{n} c_i z_i^{(t)}\left(\mathbf{v}^T\mathbf{x}_i + b\right) \geq \frac{1}{n}\sum_{i=1}^{n} c_i - \xi,\\
& \forall k \in \{1, \ldots, d\}: 0 \leq \sigma_k \leq 1\\
& \forall k \in \{1, \ldots, d\}: \left\|\begin{bmatrix} 2v_k \\ t_k - \sigma_k \end{bmatrix}\right\| \leq t_k + \sigma_k\\
& \left\|\begin{bmatrix} 2\mathbf{L}^{\frac{1}{2}}(\mathbf{X}^T\mathbf{v} + b\mathbf{1}) \\ s - 1 \end{bmatrix}\right\| \leq s + 1,\ \xi \geq 0\\
& \sum_{k=1}^{d}\sigma_k = m,\ -l \leq \sum_{i=1}^{n}\left(\mathbf{v}^T\mathbf{x}_i + b\right) \leq l
\end{aligned}$$

The above *SOCP* problem can be solved in polynomial time [13]. Following the *CCCP*, the obtained solution $(\mathbf{v}, b, \boldsymbol{\sigma}, \xi, \mathbf{t}, s)$ from this *SOCP* problem is then used as $(\mathbf{v}^{(t+1)}, b^{(t+1)}, \boldsymbol{\sigma}, \xi, \mathbf{t}, s)$, and the iteration continues until convergence. The algorithm for solving problem (7) subject to the constraint subset Ω is summarized in Algorithm 2. As for its termination criterion, we check if the difference in objective values from two successive iterations is less than $\alpha\%$ (which is set to 0.01 in the experiments).

Algorithm 2 Solve problem (7) subject to constraint subset Ω via the constrained concave-convex procedure.

Initialize $(\mathbf{v}^{(0)}, b^{(0)})$.
repeat
Obtain $(\mathbf{v}, b, \boldsymbol{\sigma}, \xi, \mathbf{t})$ as the solution to problem (10).
Set $\mathbf{v}^{(t+1)} = \mathbf{v}$, $b^{(t+1)} = b$ and $t = t + 1$.
until the stopping criterion is satisfied.

3.2.2 Identifying the Most Violated Constraint

The most violated constraint is the one that results in the largest ξ. Since each constraint in (8) is represented by a vector $\mathbf{c}$, we have the following proposition:

Proposition 3 *The most violated constraint* $\mathbf{c}$ *in (8) can be computed as:*

$$c_i = \begin{cases} 1 & \text{if } \left|\mathbf{v}^T\mathbf{x}_i + b\right| < 1,\\ 0 & \text{otherwise.} \end{cases} \qquad (11)$$

The *cutting plane* algorithm iteratively selects the most violated constraint under the current hyperplane parameter and then adds it to the working constraint set Ω, until no constraint is violated by more than ϵ, i.e.,

$$\forall \mathbf{c} \in \{0, 1\}^n\colon \frac{1}{n}\sum_{i=1}^{n} c_i\left|\mathbf{v}^T\mathbf{x}_i + b\right| \geq \frac{1}{n}\sum_{i=1}^{n} c_i - (\xi + \epsilon) \qquad (12)$$

Moreover, note that in the objective function of problem (7), there is a single slack variable ξ measuring the clustering loss. Hence, we can simply select the stopping criterion in Algorithm 1 as being all the samples satisfying inequality (12). Then, the approximation accuracy ϵ of this approximate solution is directly related to the clustering loss.

3.3. Accuracy of the Cutting Plane Algorithm

The following proposition characterizes the accuracy of the solution computed by the *cutting plane* algorithm.

Proposition 4 *For any* $\epsilon > 0$*, the cutting plane algorithm for M3FS returns a point* $(\mathbf{v}, b, \boldsymbol{\sigma}, \xi)$ *for which* $(\mathbf{v}, b, \boldsymbol{\sigma}, \xi + \epsilon)$ *is feasible in problem (7).*

Based on this proposition, ϵ indicates how close one wants to be to the error rate of the best separating hyperplane. This justifies its use as the stopping criterion in Algorithm 1.

3.4. Time Complexity Analysis

In this section, we provide theoretical analysis on the time complexity of the *cutting plane* algorithm for *manifold-based maximum margin feature selection*. We will first obtain the time involved in each iteration of the algorithm. Next, we will show that the total number of constraints added into the working set Ω, i.e., the total number of iterations involved in the *cutting plane* algorithm, is upper bounded. Specifically, we have the following two lemmas,

Lemma 1 *Each iteration of the cutting plane algorithm for manifold-based maximum margin feature selection takes $O(d^{3.5}+nd+d^{2.5}|\Omega|)$ time for a working constraint set size $|\Omega|$.*

Lemma 2 *The cutting plane algorithm terminates after adding at most $\frac{CR}{\epsilon^2}$ constraints, where R is a constant independent of n and d.*

Lemma 2 bounds the number of iterations in our *cutting plane* algorithm by a constant $\frac{CR}{\epsilon^2}$, which is independent of n and d. Moreover, each iteration of the algorithm takes $O(d^{3.5}+nd+d^{2.5}|\Omega|)$ time. Therefore, the *cutting plane* algorithm for *manifold-based maximum margin feature selection* has a time complexity of $\sum_{|\Omega|=1}^{CR/\epsilon^2} O(d^{3.5}+nd+d^{2.5}|\Omega|) = O(\frac{d^{3.5}+nd}{\epsilon^2}+\frac{d^{2.5}}{\epsilon^4})$. Hence, we have the following proposition.

Proposition 5 *The cutting plane algorithm for manifold-based maximum margin feature selection takes $O(\frac{d^{3.5}+nd}{\epsilon^2}+\frac{d^{2.5}}{\epsilon^4})$ time.*

3.5. Multi-Class M3FS

For the multi-class scenario, we will start with an introduction to the *multi-class support vector machine* formulation proposed in [5]. Given a point set $\mathbf{X}=\{\mathbf{x}_1,\cdots,\mathbf{x}_n\}$ and their labels $\mathbf{y}=(y_1,\ldots,y_n)\in\{1,\ldots,M\}^n$, the *SVM* defines a weight vector $\mathbf{w}_p$ for each class $p\in\{1,\ldots,M\}$ and classifies sample $\mathbf{x}$ by $p^*=\arg\max_{p\in\{1,\ldots,M\}}\mathbf{w}_p^T\mathbf{x}$. The weight vectors are obtained as follows:

$$\begin{aligned}\min_{\mathbf{w}_1,\ldots,\mathbf{w}_M,\boldsymbol{\xi}}\quad & \frac{1}{2}\sum_{p=1}^{M}||\mathbf{w}_p||^2+\frac{C}{n}\sum_{i=1}^{n}\xi_i \qquad (13)\\ \text{s.t.}\quad & \forall i\in\{1,\ldots,n\}, r\in\{1,\ldots,M\}:\\ & \mathbf{w}_{y_i}^T\mathbf{x}_i+\delta_{y_i,r}-\mathbf{w}_r^T\mathbf{x}_i\geq 1-\xi_i;\ \xi_i\geq 0.\end{aligned}$$

Similar with the two-class scenario, we define a scale factor for each feature and obtain the following unsupervised *multi-class manifold-based maximum margin feature selection* formulation

$$\begin{aligned}\min_{\mathbf{y},\boldsymbol{\sigma},\mathbf{v},\boldsymbol{\xi}}\quad & \frac{1}{2}\sum_{k=1}^{d}\sum_{p=1}^{M}\frac{v_{pk}^2}{\sigma_k}+\frac{C}{n}\sum_{i=1}^{n}\xi_i+\lambda\sum_{p=1}^{M}\mathbf{v}_p^T\mathbf{X}\mathbf{L}\mathbf{X}^T\mathbf{v}_p \qquad (14)\\ \text{s.t.}\quad & \forall i\in\{1,\ldots,n\}, r\in\{1,\ldots,M\}:\\ & \sum_{k=1}^{d}(v_{y_ik}-v_{rk})x_{ik}+\delta_{y_i,r}\geq 1-\xi_i,\ \xi_i\geq 0\\ & \forall k\in\{1,\ldots,d\}:\ 0\leq\sigma_k\leq 1;\ \sum_{k=1}^{d}\sigma_k=m\\ & \forall p,q\in\{1,\ldots,M\}:-l\leq\sum_{i=1}^{n}\sum_{k=1}^{d}(v_{pk}-v_{qk})x_{ik}\leq l,\end{aligned}$$

Here, the subscript p in w_{pk} denotes the pth class, k denotes the kth feature, and we have applied the change of variables $\forall p\in\{1,\ldots,M\}, k\in\{1,\ldots,d\}:\ v_{pk}=\sigma_k w_{pk}$ to ensure that the objective function and the last constraint are convex. Similar to two-class clustering, we have also added class balance constraints (where $l>0$) in the formulation to control class imbalance. Again, the above formulation is an integer program, and is much more complex than the *QP* problem in *multi-class SVM*. Fortunately, we have the following proposition.

Proposition 6 *Problem (14) is equivalent to*

$$\begin{aligned}\min_{\boldsymbol{\sigma},\mathbf{v},\boldsymbol{\xi}}\quad & \frac{1}{2}\sum_{k=1}^{d}\sum_{p=1}^{M}\frac{v_{pk}^2}{\sigma_k}+\frac{C}{n}\sum_{i=1}^{n}\xi_i+\lambda\sum_{p=1}^{M}\mathbf{v}_p^T\mathbf{X}\mathbf{L}\mathbf{X}^T\mathbf{v}_p \qquad (15)\\ \text{s.t.}\quad & \forall i\in\{1,\ldots,n\}, r\in\{1,\ldots,M\}:\\ & \sum_{k=1}^{d}\left(\sum_{p=1}^{M}z_{ip}v_{pk}-v_{rk}\right)x_{ik}+z_{ir}\geq 1-\xi_i,\ \xi_i\geq 0\\ & \forall k\in\{1,\ldots,d\}:\ 0\leq\sigma_k\leq 1;\ \sum_{k=1}^{d}\sigma_k=m\\ & \forall p,q\in\{1,\ldots,M\}:-l\leq\sum_{i=1}^{n}\sum_{k=1}^{d}(v_{pk}-v_{qk})x_{ik}\leq l,\end{aligned}$$

where z_{ip} is defined as $\forall i\in\{1,\ldots,n\}, p\in\{1,\ldots,M\}$:

$$z_{ip}=\prod_{q=1,q\neq p}^{M}I_{[\sum_{k=1}^{d}v_{pk}x_{ik}>\sum_{k=1}^{d}v_{qk}x_{ik}]},$$

with $I(\cdot)$ being the indicator function and the label for sample $\mathbf{x}_i$ is determined as $y_i=\arg\max_p\sum_{k=1}^{d}v_{pk}x_{ik}=\sum_{p=1}^{M}pz_{ip}$.

To reduce the number of slack variables, we make use of the following proposition:

Proposition 7 *Problem (15) can be equivalently formulated as problem (16), with* $\xi^* = \frac{1}{n}\sum_{i=1}^{n}\xi_i^*$.

$$\min_{\boldsymbol{\sigma},\mathbf{v},\boldsymbol{\xi}} \ \frac{1}{2}\sum_{k=1}^{d}\sum_{p=1}^{M}\frac{v_{pk}^2}{\sigma_k} + C\xi + \lambda\sum_{p=1}^{M}\mathbf{v}_p^T\mathbf{X}\mathbf{L}\mathbf{X}^T\mathbf{v}_p \tag{16}$$

$$\begin{aligned}
s.t.\ & \forall \mathbf{c}_i \in \{\mathbf{e}_0, \mathbf{e}_1, \ldots, \mathbf{e}_k\}, i \in \{1, \ldots, n\}: \\
& \frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{d}\sum_{p=1}^{M}(\mathbf{c}_i^T\mathbf{e}z_{ip} - c_{ip})v_{pk}x_{ik} + \frac{1}{n}\sum_{i=1}^{n}\sum_{p=1}^{M}c_{ip}z_{ip} \\
& \geq \frac{1}{n}\sum_{i=1}^{n}\mathbf{c}_i^T\mathbf{e} - \xi, \\
& \forall k \in \{1, \ldots, d\}: \ 0 \leq \sigma_k \leq 1, \\
& \sum_{k=1}^{d}\sigma_k = m, \ \xi \geq 0 \\
& \forall p, q \in \{1, \ldots, M\}: -l \leq \sum_{i=1}^{n}\sum_{k=1}^{d}(v_{pk} - v_{qk})x_{ik} \leq l,
\end{aligned}$$

where we define $\mathbf{e}_p$ *as the* $M \times 1$ *vector with only the pth element being 1 and others 0,* $\mathbf{e}_0$ *as the* $M \times 1$ *zero vector and* $\mathbf{e}$ *as the vector of ones.*

A single slack variable ξ is shared across all the non-convex constraints in (16) and, again, the cutting plane algorithm can be used to handle the exponential number of constraints.

For the inner optimization, we use the *CCCP* to compute $\mathbf{v}^{(t+1)}$ from $\mathbf{v}^{(t)}$ by solving the following optimization problem

$$\min_{\boldsymbol{\sigma},\mathbf{v},\boldsymbol{\xi}} \ \frac{1}{2}\sum_{k=1}^{d}\sum_{p=1}^{M}\frac{v_{pk}^2}{\sigma_k} + C\xi + \lambda\sum_{p=1}^{M}\mathbf{v}_p^T\mathbf{X}\mathbf{L}\mathbf{X}^T\mathbf{v}_p \tag{17}$$

$$\begin{aligned}
\text{s.t.}\ & \forall [\mathbf{c}_1, \ldots, \mathbf{c}_n] \in \Omega, i \in \{1, \ldots, n\}: \\
& \frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{d}\sum_{p=1}^{M}(\mathbf{c}_i^T\mathbf{e}z_{ip}^{(t)} - c_{ip})v_{pk}x_{ik} + \frac{1}{n}\sum_{i=1}^{n}\sum_{p=1}^{M}c_{ip}z_{ip}^{(t)} \\
& \geq \frac{1}{n}\sum_{i=1}^{n}\mathbf{c}_i^T\mathbf{e} - \xi; \ \xi \geq 0 \\
& \forall k \in \{1, \ldots, d\}: \ 0 \leq \sigma_k \leq 1; \ \sum_{k=1}^{d}\sigma_k = m \\
& \forall p, q \in \{1, \ldots, M\}: -l \leq \sum_{i=1}^{n}\sum_{k=1}^{d}(v_{pk} - v_{qk})x_{ik} \leq l,
\end{aligned}$$

where $z_{ip}^{(t)} = \prod_{q=1, q\neq p}^{M} I_{[\sum_{k=1}^{d} v_{pk}^{(t)} x_{ik} > \sum_{k=1}^{d} v_{qk}^{(t)} x_{ik}]}$. Again, this can be formulated as an *SOCP* and solved efficiently. Finally, as for the most violated constraint, it is the one that results in the largest ξ and can be obtained by the following proposition.

Proposition 8 *The most violated constraint* $\mathbf{c} = [\mathbf{c}_1, \ldots, \mathbf{c}_n]$ *can be obtained as*

$$\mathbf{c}_i = \begin{cases} \mathbf{e}_{r^*} & \text{if } \left[\sum_{k=1}^{d} v_{p^*k}x_{ik} - \sum_{k=1}^{d} v_{r^*k}x_{ik}\right] < 1, \\ \mathbf{0} & \text{otherwise,} \end{cases}$$

where $p^* = \arg\max_p \sum_{k=1}^{d} v_{pk}x_{ik}$ *and* $r^* = \arg\max_{r\neq p^*} \sum_{k=1}^{d} v_{rk}x_{ik}$.

4. Experiments

In this section, we validate the effectiveness of *manifold-based maximum margin feature selection (M3FS)* on both toy and real-world data sets.

4.1. Setup

We use 5 data sets which are intended to cover a wide range of properties: ionosphere, digits, letter and satellite (these are from the UCI data repository[1]), and mnist[2]. The two-class data sets are created following the same setting as in [22]. We also create several multi-class data sets from the digits, letter and mnist data. All these are summarized in Table 1. For representing the manifold used

Data	Size	Feature	Class
digits1v7	361	64	2
digits2v7	356	64	2
ionosphere	354	64	2
letterAvB	1555	16	2
satellite	2236	36	2
digits0689	713	64	4
digits1279	718	64	4
letterABCD	3096	16	4
mnist01234	28911	196	5

Table 1. Descriptions of the data sets.

in *M3FS*, we use a fully-connected graph connecting all the samples, and set the pairwise similarity matrix $\mathbf{W}$ as $w_{ij} = \exp\left(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/2\rho^2\right)$, where ρ is the variance in the Gaussian function. Besides *M3FS*, for comparison, we also run the following algorithms which perform clustering with feature selection:

- **Feature selection based on Gaussian mixture model *(FSGMM)*** [12]: This is an embedded approach for unsupervised feature selection and, as its name implies, the clustering algorithm is based on the Gaussian mixture model. Its implementation is the same as in [12].
- **Laplacian score** [9]: This is a filter method for supervised/unsupervised feature selection which also uses manifold information. The implementation code is downloaded from

[1] http://archive.ics.uci.edu/ml/
[2] http://yann.lecun.com/exdb/mnist/

http://www.cs.uiuc.edu/homes/dengcai2. Since it is a filter, it is not particularly tied to any clustering algorithm. In the following, we experiment with both *MMC* and *K-Means*, and the corresponding methods are denoted ***LapMMC*** and ***LapKM***, respectively.

Moreover, we also experiment with *maximum margin clustering* without doing feature selection. The implementation is the same as in [23], and this will be denoted as ***MMC-all***.

In the experiments, we first take a set of labeled data, remove all the labels and run the clustering algorithms; then we label each of the resulting clusters with the majority class according to the original labels. Moreover, we always set the number of clusters to be the true number of classes M for all the methods. These clustering algorithms (with or without feature selection) will be evaluated by the following two performance measures:

Clustering Accuracy *(Acc)*. The first performance measure is the *Clustering Accuracy*, which discovers the one-to-one relationship between clusters and classes and measures the extent to which each cluster contained data points from the corresponding class. Specifically, *Acc* measures the number of correct classifications.

Rand Index *(RI)* [15]. Let $\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_M\}$ be the set of final clustering results such that $\mathcal{C}_k$ represents the kth cluster, and $\mathcal{L} = \{\mathcal{L}_1, \mathcal{L}_2, \ldots, \mathcal{L}_M\}$ denotes the set of true data classes such that $\mathcal{L}_k$ represents the kth class. We define the following four variables: a: the number of data pairs in $\mathbf{X}$ that are in the same set in both $\mathcal{C}$ and $\mathcal{L}$; b: the number of data pairs in $\mathbf{X}$ that are in different sets in both $\mathcal{C}$ and $\mathcal{L}$; c: the number of data pairs in $\mathbf{X}$ that are in the same set in $\mathcal{C}$ but different sets in $\mathcal{L}$; d: the number of data pairs in $\mathbf{X}$ that are in different sets in $\mathcal{C}$ but the same set in $\mathcal{L}$. Then the Rand Index R that measures the similarity between $\mathcal{C}$ and $\mathcal{L}$ can be computed as $R = \frac{a+b}{a+b+c+d}$. Intuitively, one can think of $a + b$ as the number of agreements between $\mathcal{C}$ and $\mathcal{L}$ and $c + d$ as the number of disagreements between $\mathcal{C}$ and $\mathcal{L}$. Clearly, R has a value between 0 and 1, with 0 indicating that $\mathcal{C}$ and $\mathcal{L}$ do not agree on any pair of data points, and 1 indicating that $\mathcal{C}$ and $\mathcal{L}$ are exactly the same.

4.2. Ability to Detect Relevant Features

In this section, we first illustrate the ability of *M3FS* in selecting relevant features by using the iris data set from UCI machine learning repository. The iris data contain 3 classes of 50 instances each, and each instance is characterized by 4 features. We add 10 noisy features (generated from the normal distribution $\mathcal{N}(0, 1)$) to the iris data, and thus obtain a data set of 150 14-dimensional instances.

The saliencies of all the 14 features as calculated by the various methods are shown in Figure 1. For simplicity of illustration, we order the features such that the first four are the original features, while the last ten are the noisy ones. As can be seen, *M3FS* successfully selects the 4 relevant features and assigns zero saliency to all the noisy features. On the other hand, both the **Laplacian score** and **FSGMM** assign non-zero saliencies to the noisy features.

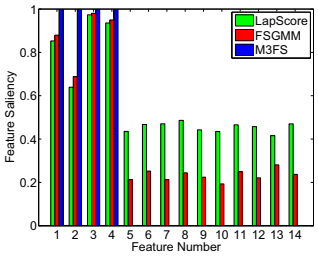

Figure 1. Feature saliencies on the iris data set with 10 noisy features added.

4.3. Clustering Performance

In this section, we report the clustering performance of the various algorithms on the data sets in Table 1. The clustering accuracy and Rand Index results are shown in Table 2. We also demonstrate the effect when manifold information is not used by setting $\lambda = 0$. As can be seen, even when no manifold information is used, both the clustering accuracy and *Rand Index* of *M3FS* are comparable to those attained by *maximum margin clustering* using all features and is often better than the other unsupervised feature selection algorithms. The addition of manifold regularization significantly improves the performance of *M3FS* and enables it to be even better than *MMC-all*.

4.4. Generalization Ability of M3FS

Manifold-based maximum margin feature selection adopts the maximum margin principle of *SVM*, which could allow good generalization on unseen data. In this experiment, we validate the generalization ability of *M3FS* on unseen data samples. We first learn the *M3FS* model on a data subset randomly drawn from the whole data set. Then we use the learned model to cluster the whole data set. As can be seen in Table 3, the clustering performance of the model learned on the data subset is comparable with that of the model learned on the whole data set. Thus, for a large data set, we can simply perform the feature selection and clustering process on a small subset of the data and then use the learned model to cluster the remaining data points.

5. Conclusions

In this paper, we propose a novel unsupervised feature selection method named *Manifold-Based Maximum Margin Feature Selection (M3FS)*. *M3FS* targets to identify those features that would result in the maximal separation of different clusters. As many computer vision and pattern recognition problems have intrinsic manifold structure, we add

Data	m	LapKM		LapMMC		FSGMM		MMC-all		M3FS		M3FS ($\lambda=0$)	
digits1v7	10	79.50	0.569	70.08	0.580	88.64	0.798	**100.0**	**1.00**	**100.0**	**1.00**	**100.0**	**1.00**
digits2v7	10	88.20	0.723	84.27	0.734	80.62	0.687	**100.0**	**1.00**	**100.0**	**1.00**	**100.0**	**1.00**
ionosphere	10	69.52	0.575	64.10	0.539	70.94	0.587	72.36	0.599	**85.57**	**0.755**	70.66	0.584
letterAvB	10	92.80	0.866	94.21	0.891	90.29	0.825	93.12	0.873	**96.33**	**0.929**	94.41	0.894
satellite	16	95.35	0.911	97.45	0.950	95.53	0.915	98.48	0.971	**98.75**	**0.975**	**98.75**	**0.975**
digits0689	20	54.84	0.735	93.41	0.600	75.32	0.863	96.63	0.968	**97.19**	**0.973**	95.65	0.960
digits1279	20	74.65	0.811	89.97	0.583	79.53	0.834	94.01	0.943	**96.66**	**0.968**	92.48	0.931
letterABCD	10	66.09	0.773	62.08	0.731	65.67	0.777	70.77	0.804	**85.53**	**0.867**	70.51	0.815
mnist01234	50	-	-	-	-	71.32	0.811	89.98	0.901	**90.85**	**0.919**	**90.85**	**0.919**

Table 2. Clustering accuracy (%) and Rand Index comparisons on the various data sets. For each method, the number on the left denotes the clustering accuracy, and the number on the right stands for the Rand Index. The symbol '-' means that the corresponding algorithm cannot handle the data set in reasonable time.

Data	from whole set		from data subset		
	Acc	RI	subset size	Acc	RI
letterAvB	96.33	0.929	500	95.60	0.912
satellite	98.75	0.975	500	98.57	0.972
letterABCD	85.53	0.867	500	83.98	0.852
mnist01234	90.85	0.919	1000	89.11	0.902

Table 3. Generalization ability on unseen samples when the *M3FS* model is learned only from a data subset.

Laplacian regularizer in the objective to enforce smoothness on the clustering function. Moreover, we also extend the *M3FS* algorithm to the multi-class setting. Finally, experimental results on both toy and real-world data sets demonstrate the effectiveness of the proposed approach.

Acknowledgements

Supported by NSFC (Grant No. 60835002 and No. 60675009), and Research Grants Council of the Hong Kong Special Administrative Region under grant 614907.

References

[1] M. Belkin, P. Niyogi, and V. Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *JMLR*, 7:2399–2434, 2006. 3

[2] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004. 4

[3] F. Chung. *Spectral Graph Theory*. American Mathematical Society, 1997. 3

[4] C. Constantinopoulos, M. Titsias, and A. Likas. Bayesian feature and model selection for gaussian mixture models. *TPAMI*, 28(6):1013–1018, 2006. 1

[5] K. Crammer and Y. Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *JMLR*, 2:265–292, 2001. 5

[6] M. Dash, K. Choi, P. Scheuermann, and H. Liu. Feature selection for clustering - a filter solution. In *ICDM*, 2002. 1

[7] J. Dy and C. Brodley. Feature selection for unsupervised learning. *JMLR*, 5:845–889, Dec. 2004. 1

[8] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *JMLR*, 3:1157–1182, 2003. 1

[9] X. He, D. Cai, and P. Niyogi. Laplacian score for feature selection. In *NIPS*, 2006. 2, 7

[10] A. Jain, R. Duin, and J. Mao. Statistical pattern recognition: A review. *TPAMI*, 22(1), 2000. 1

[11] J. E. Kelley. The cutting-plane method for solving convex programs. *Journal of the Society for Industrial Applied Mathematics*, 8:703–712, 1960. 3

[12] M. Law, M. Figueiredo, and A. Jain. Simultaneous feature selection and clustering using mixture models. *TPAMI*, 26(9):1154–1166, 2004. 1, 7

[13] M. Lobo, L. Vandenberghe, S. Boyd, and H. Lebret. Applications of second-order cone programming. *Linear Algebra Appl.*, 284:193–228, 1998. 4

[14] P. Mitra, C. Murthy, and S. Pal. Unsupervised feature selection using feature similarity. *TPAMI*, 24(3):301–312, Mar. 2002. 1

[15] W. Rand. Objective criteria for the evaluation of clustering methods. *JASA*, 66:846–850, 1971. 7

[16] V. Roth and T. Lange. Feature selection in clustering problems. In *NIPS*, 2004. 1

[17] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2000. 3

[18] A. J. Smola, S. Vishwanathan, and T. Hofmann. Kernel methods for missing variables. In *AISTATS*, 2005. 4

[19] H. Valizadegan and R. Jin. Generalized maximum margin clustering and unsupervised kernel learning. In *NIPS*, Cambridge, MA, 2007. MIT Press. 2

[20] J. Weston, S. Mukherjee, O. Chapelle, M. Pontil, T. Poggio, and V. Vapnik. Feature selection for svms. In *NIPS*, 2000. 3

[21] L. Xu, J. Neufeld, B. Larson, and D. Schuurmans. Maximum margin clustering. In *NIPS*, Cambridge, MA, 2005. MIT Press. 2

[22] K. Zhang, I. W. Tsang, and J. T. Kwok. Maximum margin clustering made practical. In *ICML*, 2007. 6

[23] B. Zhao, F. Wang, and C. Zhang. Efficient multiclass maximum margin clustering. In *ICML*, 2008. 7

[24] A. Zien and C. Ong. Multiclass multiple kernel learning. In *ICML*, 2007. 3