A Joint Optimization Framework of Sparse Coding and Discriminative Clustering

Zhangyang Wang†, Yingzhen Yang†, Shiyu Chang†,
Jinyan Li‡, Simon Fong‡, and Thomas S. Huang†
†University of Illinois at Urbana-Champaign, Urbana, IL, USA
‡University of Macau, Macau, China
{zwang119, yyang58, chang87, t-huang1}@illinois.edu {yb47432, ccfong}@umac.mo

Abstract

Many clustering methods highly depend on extracted features. In this paper, we propose a joint optimization framework in terms of both feature extraction and discriminative clustering. We utilize graph regularized sparse codes as the features, and formulate sparse coding as the constraint for clustering. Two cost functions are developed based on entropy-minimization and maximum-margin clustering principles, respectively, as the objectives to be minimized. Solving such a bi-level optimization mutually reinforces both sparse coding and clustering steps. Experiments on several benchmark datasets verify remarkable performance improvements led by the proposed joint optimization.

1 Introduction

Clustering [Yang *et al.*, 2014a] plays an important role in many real world data mining applications. To learn the hidden patterns of the dataset in an unsupervised way, existing clustering algorithms can be described as either generative or discriminative in nature. Generative clustering algorithms model categories in terms of their geometric properties in feature spaces, or as statistical processes of data. Examples include K-means and Gaussian mixture model (GMM) clustering [Biernacki *et al.*, 2000], which assume a parametric form of the underlying category distributions. Discriminative clustering techniques search for the boundaries or distinctions between categories. With fewer assumptions being made, these methods are powerful and flexible in practice. For example, maximum-margin clustering [Xu *et al.*, 2004], [Xu and Schuurmans, 2005], [Zhao *et al.*, 2008] aims to find the hyperplane, that can separate the data from different classes with maximum margins. Information theoretic clustering [Li *et al.*, 2004], [Barber and Agakov, 2005] minimize the conditional entropy of all samples. Many recent discriminative clustering methods have achievedsatisfactory performances [Zhao *et al.*, 2008].

Moreover, many clustering methods extract discriminative features from data prior to clustering. The Principal Component Analysis (PCA) feature is a common choice but not necessarily discriminative [Zheng *et al.*, 2011]. In [Roth and Lange, 2003], the features are selected for optimizing the discriminativity by Linear Discriminant Analysis (LDA). More recently, sparse codes prove to be both robust to noise and scalable to high dimensional data [Wright *et al.*, 2009]. Furthermore, ℓ_1-graph [Cheng *et al.*, 2010] builds the graph by reconstructing each data point sparsely and locally with other data. A spectral clustering [Ng *et al.*, 2002] is followed based on the constructed graph matrix. In [Sprechmann and Sapiro, 2010], dictionary learning is combined with the clustering process, which makes the use of Lloyds-type algorithms that iteratively re-assign data to clusters and then optimize the dictionary associated with each cluster. In [Zheng *et al.*, 2011], the authors learned the sparse codes that explicitly consider the local data manifold structures. Their results indicate that encoding geometrical information will significantly enhance the learning performance. However, their clustering step is neither discriminative nor jointly optimized.

In this paper, we propose to jointly optimize feature extraction and discriminative clustering, in which way they mutually reinforce each other. We focus on sparse codes as the extracted features, and develop our loss functions based on two representative discriminative clustering methods, the entropy-minimization [Li *et al.*, 2004] and maximum-margin [Xu *et al.*, 2004] clustering, respectively. A task-driven bi-level optimization model [Mairal *et al.*, 2012], [Wang *et al.*, 2015] is then built upon the proposed framework. The sparse coding step is formulated as the lower-level constraint, where a graph regularization is enforced to preserve the local manifold structure [Zheng *et al.*, 2011]. The clustering-oriented cost functions are considered as the upper-level objectives to be minimized. Stochastic gradient descent algorithms are developed to solve both bi-level models. Experiments on several popular real datasets verify the noticeable performance improvement led by such a joint optimization framework.

2 Model Formulation

2.1 Sparse Coding with Graph Regularization

Sparse codes prove to be an effective feature for clustering. In [Cheng *et al.*, 2010], the authors suggested that the contribution of one sample to the reconstruction of another sample was a good indicator of similarity between these two samples. Therefore, the reconstruction coefficients (sparse codes) can be used to constitute the similarity graph for spectral clustering. ℓ_1-graph performs sparse representation for each data point separately without considering the geometric information and manifold structure of the entire data. Further research shows that the graph regularized sparse representations produce superior results in various clustering and classification tasks [Zheng *et al.*, 2011], [Yang *et al.*, 2014c]. In this paper, we adopt the graph regularized sparse codes as the features for clustering.

We assume that all the data samples $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n], \mathbf{x}_i \in R^{m \times 1}, i = 1, 2, \cdots, n$, are encoded into their corresponding sparse codes $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_n]$, $\mathbf{a}_i \in R^{p \times 1}, i = 1, 2, \cdots, n$, using a learned dictionary $\mathbf{D} = [\mathbf{d}_1, \mathbf{d}_2, \cdots, \mathbf{d}_p]$, where $\mathbf{d}_i \in R^{m \times 1}, i = 1, 2, \cdots, p$ are the learned atoms. Moreover, given a pairwise similarity matrix $\mathbf{W}$, the sparse representations that capture the geometric structure of the data according to the manifold assumption should minimize the following objective: $\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\mathbf{W}_{ij}||\mathbf{a}_i - \mathbf{a}_j||_2^2 = Tr(\mathbf{ALA^T})$, where $\mathbf{L}$ is the graph Laplacian matrix constructed from $\mathbf{W}$. In this paper, $\mathbf{W}$ is chosen as the Gaussian Kernel: $\mathbf{W}_{ij} = \exp(-\frac{||\mathbf{x}_i - \mathbf{x}_j||_2^2}{\delta^2})$, where δ is the controlling parameter selected by cross-validation.

The graph regularized sparse codes are obtained by solving the following convex optimization

$$\begin{aligned}\mathbf{A} = \arg\min\nolimits_{\mathbf{A}} \tfrac{1}{2}||\mathbf{X} - \mathbf{DA}||_F^2 + \lambda \textstyle\sum_i ||\mathbf{a}_i||_1 \\ +\alpha Tr(\mathbf{ALA^T}) + \lambda_2 ||\mathbf{A}||_F^2.\end{aligned} \tag{1}$$

Note $\lambda_2 > 0$ is necessary for proving the differentiability of the objective function (see [7.1] in the Appendix). However, setting $\lambda_2 = 0$ proves to work well in practice, and thus the term $\lambda_2||\mathbf{A}||_F^2$ will be omitted by default hereinafter (except for the differentiability proof).

Obviously, the effect of sparse codes $\mathbf{A}$ largely depends on the quality of dictionary $\mathbf{D}$. Dictionary learning methods, such as K-SVD algorithm [Elad and Aharon, 2006], are widely used in sparse coding literature. In regard to clustering, the authors in [Cheng *et al.*, 2010], [Yang *et al.*, 2014c], [Yang *et al.*, 2014b] constructed the dictionary by directly selecting atoms from data samples. [Zheng *et al.*, 2011] learned the dictionary that can reconstruct input data well. However, it does not necessarily lead to discriminative features. In contrast, we will optimize $\mathbf{D}$ together with the clustering task.

2.2 Bi-level Optimization Formulation

The objective cost function for the joint framework can be expressed by the following bi-level optimization:

$$\begin{aligned}\min_{\mathbf{D},\mathbf{w}} \quad & C(\mathbf{A}, \mathbf{w}) \\ s.t. \quad & \mathbf{A} = \arg\min\nolimits_{\mathbf{A}} \tfrac{1}{2}||\mathbf{X} - \mathbf{DA}||_F^2 + \lambda \textstyle\sum_i ||\mathbf{a}_i||_1 \\ & \quad +\alpha Tr(\mathbf{ALA^T}).\end{aligned} \tag{2}$$

where $C(\mathbf{A}, \mathbf{w})$ is a cost function evaluating the loss of clustering, with $\mathbf{A}$ as its input and $\mathbf{w}$ as the parameter. It can be formulated differently based on various clustering principles, two of which will be discussed and solved in Section 3.

Bilevel optimization [Yang *et al.*, 2012] has been investigated in both theory and application sides. In [Yang *et al.*, 2012], the authors proposed a general bilevel sparse coding model for learning dictionaries across coupled signal spaces. Another similar formulation has been studied in [Mairal *et al.*, 2012] for general regression tasks.

3 Clustering-oriented Cost Functions

3.1 Entropy-Minimization Loss

Maximization of the mutual information with respect to parameters of the encoder model effectively defines a discriminative unsupervised optimization framework. The model is parameterized similarly to a conditionally trained classifier, but the cluster allocations are unknown [Barber and Agakov, 2005]. In [Dai and Hu, 2010], [Li *et al.*, 2004], the authors adopted an information-theoretic framework as an implementation of the low-density separation assumption by minimizing the conditional entropy. By substituting the logistic posterior probability into the minimum conditional entropy principle, the authors got the logistics clustering algorithm, which is equivalent to find a labelling strategy so that the total entropy of data clustering is minimized.

Assuming K clusters, since the true cluster label of each $\mathbf{x_i}$ is unknown, we introduce the predicted confidence probability p_{ij} that sample $\mathbf{x_i}$ belongs to cluster j, $i = 1, 2, \cdots, N$, $j = 1, 2, \cdots, K$, which is set as the likelihood of multinomial logistic (softmax) regression:

$$p_{ij} = p(j|\mathbf{w}, \mathbf{a}_i) = \frac{e^{-j\mathbf{w}^T\mathbf{a}_i}}{\sum_{l=1}^{K} e^{-l\mathbf{w}^T\mathbf{a}_i}}, \tag{3}$$

The loss function for all data could be defined accordingly in a entropy-like form:

$$C(\mathbf{A}, \mathbf{w}) = -\textstyle\sum_{i=1}^{n}\sum_{j=1}^{K} p_{ij}\log p_{ij}. \tag{4}$$

The predicted cluster label of $\mathbf{a}_i$ is the cluster j where it achieves the largest likelihood probability p_{ij}. The logistics regression can deal with multi-class problems more easily compared with the support vector machine (SVM). The next important thing we need to study is the differentiability of (2).

Theorem 3.1. *The objective* $C(\mathbf{A}, \mathbf{w})$ *defined in (4) is differentiable on* $\mathbf{D} \times \mathbf{w}$.

Proof: Denote $\mathbf{X} \in \mathcal{X}$, and $\mathbf{D} \in \mathcal{D}$. Also let the objective function $C(\mathbf{A}, \mathbf{w})$ in (4) be denoted as C for short. The differentiability of C with respect to $\mathbf{w}$ is easy to show, assuming the compactness of $\mathcal{X}$, as well as the fact that C is twice differentiable.

We will therefore focus on showing that C is differentiable with respect to $\mathbf{D}$, which is more difficult since $\mathbf{A}$, and thus $\mathbf{a}_i$, is not differentiable everywhere. Without loss of generality, we use a vector $\mathbf{a}$ instead of $\mathbf{A}$ for simplifying the derivations hereinafter. In some cases, we may equivalently express $\mathbf{a}$ as $\mathbf{a}(\mathbf{D}, \mathbf{w})$ in order to emphasize the functional dependence. Based on [7.1] in Appendix, and given a small perturbation $\mathbf{E} \in R^{m \times p}$, it follows that

$$\begin{aligned}C(\mathbf{a}(\mathbf{D}+\mathbf{E}), \mathbf{w}) - C(\mathbf{a}(\mathbf{D}), \mathbf{w}) = \\ \nabla_{\mathbf{z}} C_{\mathbf{w}}^T(\mathbf{a}(\mathbf{D}+\mathbf{E}) - \mathbf{a}(\mathbf{D})) + O(||\mathbf{E}||_F^2),\end{aligned} \tag{5}$$

where the term $O(||\mathbf{E}||_F^2)$ is based on the fact that $\mathbf{a}(\mathbf{D}, \mathbf{x})$ is uniformly Lipschitz and $\mathcal{X} \times \mathcal{D}$ is compact. It is then possible to show that

$$\begin{aligned}C(\mathbf{a}(\mathbf{D}+\mathbf{E}), \mathbf{w}) - C(\mathbf{a}(\mathbf{D}), \mathbf{w}) = \\ Tr(\mathbf{E}^T g(\mathbf{a}(\mathbf{D}+\mathbf{E}), \mathbf{w})) + O(||\mathbf{E}||_F^2),\end{aligned} \tag{6}$$

where g has the form given in Algorithm I. This shows that C is differentiable on $\mathcal{D}$.□

Building on the differentiability proof, we are able to solve (1) using a projected first order stochastic gradient descent (SGD) algorithm, whose detailed steps are outlined in Algorithm 1. At a high level overview, it consists of an outer SGD loop that incrementally samples the training data. It uses each sample to approximate gradients with respect to $\mathbf{w}$ and $\mathbf{D}$, which are then used to update them.

Convergence and Complexity Analysis SGD converges to stationary points under a few stricter assumptions than ones satisfied in this paper. A non-convex convergence proof assumes three times differentiable cost functions [Mairal *et al.*, 2012]. As a typical case in machine learning, we use SGD in a setting where it is not guaranteed to converge in theory, but behaves well in practice.

Assuming n samples and dictionary size p, in each iteration of Algorithm 1, step 8 takes $O(n)$ time. Step 4 is solved by the feature-sign algorithm [Lee *et al.*, 2006], which is reduced to a series of quadratic programming (QP) problems. The computational bottleneck lies in solving the inverse of matrix $\mathbf{D}^T\mathbf{D}$ of size $p \times p$, where applying the Gauss-Jordan elimination method takes $O(p^3)$ time per sample. Thus, Algorithm 1 takes $O(np^3)$ time per iteration, and $O(Cnp^3)$ in total (C is a constant absorbing epoch numbers, etc.). Further, if p is a constant, Algorithm I reaches O(Cn) time complexity.

3.2 Maximum-Margin Loss

Xu et al. [Xu *et al.*, 2004] proposed maximum margin clustering (MMC), which borrows the idea from the SVM theory. Their experimental results showed that the MMC technique could often obtain more accurate results than conventional clustering methods. Technically, what MMC does is just to find a way to label the samples by running an SVM implicitly, and the SVM margin obtained would be maximized over all possible labelings [Zhao *et al.*, 2008]. However, unlike supervised large margin methods which are usually formulated as convex optimization problems, maximum margin clustering is a non-convex integer optimization problem, which is much more difficult to solve. [Li *et al.*, 2009] made several relaxations to the original MMC problem and reformulated it as a semi-definite programming (SDP) problem. The cutting plane maximum margin clustering (CPMMC) algorithm was presented in [Zhao *et al.*, 2008] to solve MMC with a much improved efficiency.

To develop the multi-class max-margin loss of clustering, we refer to the classical multi-class SVM formulation in [Crammer and Singer, 2002]. Given the sparse code $\mathbf{a}_i$ are the features to be clustered, we define the multi-class model:

$$f(\mathbf{a}_i) = \underset{j=1,\dots,K}{\arg\max} f^j(\mathbf{a}_i) = \underset{j=1,\dots,K}{\arg\max} (\mathbf{w}_j^T \mathbf{a}_i), \tag{7}$$

where f^j is the prototype for the j-th cluster and $\mathbf{w_j}$ is its corresponding weight vector. The predicted cluster label of $\mathbf{a}_i$ is the cluster of the weight vector that achieves the maximum value $\mathbf{w}_j^T \mathbf{a}_i$. Let $\mathbf{w} = [\mathbf{w}_1, ..., \mathbf{w}_K]$, the multi-class max-margin loss for $\mathbf{a}_i$ could be defined as:

$$\begin{aligned} C(\mathbf{a}_i, \mathbf{w}) &= \max(0, 1 + f^{r_i}(\mathbf{a}_i) - f^{y_i}(\mathbf{a}_i)) \\ \text{where} \quad y_i &= \underset{j=1,\dots,K}{\arg\max} f^j(\mathbf{a}_i) \\ r_i &= \underset{j=1,\dots,K, j\neq y_i}{\arg\max} f^j(\mathbf{a}_i). \end{aligned} \tag{8}$$

Note that different from training a multi-class SVM classifier, where y_i is given as a training label, the clustering scenario requires us to jointly estimate y_i as a variable. The overall max-margin loss to be minimized is (λ as the coefficient):

$$C(\mathbf{A}, \mathbf{w}) = \tfrac{\lambda}{2}||\mathbf{w}||^2 + \textstyle\sum_{i=1}^n C(\mathbf{a}_i, \mathbf{w}). \tag{9}$$

But to solve (8) or (9) with respect to the same framework as logistic loss will involve two additional concerns, which needs to be handled specifically.

First, the hinge loss of the form (8) is non-differentiable, with only subgradients existing. That makes the objective function $C(\mathbf{A}, \mathbf{w})$ non-differentiable on $\mathbf{D} \times \mathbf{w}$, and further the analysis in Theorem [3.1] proof can not be applied. We could have used the squared hinge loss or modified Huber loss for a quadratically smoothed loss function [Lee and Lin, 2013]. However, as we checked in the experiments, the quadratically smoothed loss is not as good as hinge loss in training time and sparsity. Also, though not theoretically guaranteed, using the subgradient of $\mathrm{C}(\mathbf{A}, \mathbf{w})$ works well in our case.

Second, given that $\mathbf{w}$ is fixed, it should be noted that y_i and r_i are both functions of $\mathbf{a}_i$. Therefore, calculating the derivative of (8) over $\mathbf{a}_i$ would involve expanding both r_i and y_i, and become quite complicated. Instead, we borrow ideas from the regularity of the elastic net solution [Mairal *et al.*, 2012], that the set of non-zero coefficients of the elastic net solution should not change for small perturbations. Similarly, due to the continuity of the objective, it is assumed that a sufficiently small perturbation over the current $\mathbf{a}_i$ will not change y_i and r_i. Therefore in each iteration, we could directly pre-calculate y_i and r_i using the current $\mathbf{w}$ and $\mathbf{a}_i$ and fix them for $\mathbf{a}_i$ updates [1].

Given the above two handling, for a single sample $\mathbf{a}_i$, if the hinge loss is above 0, the derivative of (8) over $\mathbf{w}$ is:

$$\Delta_i^j = \begin{cases} \lambda \mathbf{w}_i^j - \mathbf{a}_i & \text{if} \quad j = y_i \\ \lambda \mathbf{w}_i^j + \mathbf{a}_i & \text{if} \quad j = r_i \\ \lambda \mathbf{w}_i^j & \text{otherwise,} \end{cases} \tag{10}$$

where Δ_i^j denote the j-th element of the derivative for the sample $\mathbf{a}_i$. If the hinge loss is less than 0, then $\Delta_i^j = \lambda \mathbf{w}_i^j$. The derivative of (8) over $\mathbf{a}_i$ is $\mathbf{w}^{r_i} - \mathbf{w}^{y_i}$ if the hinge loss is over 0, and 0 otherwise. Note the above deduction can be conducted in a batch mode. It is then similarly solved using a projected SGD algorithm, whose steps are outlined in Algorithm 2. The convergence and complexity analysis is similar to Algorithm 1.

Algorithm 1 Stochastic gradient descent algorithm for solving (2), with $C(\mathbf{A}, \mathbf{w})$ as defined in (4)

```
Require: $\mathbf{X}, \sigma$; $\lambda$; $\mathbf{D}_0$ and $\mathbf{w}_0$ (initial dictionary and classifier parameter); ITER (number of iterations); $t_0, \rho$ (learning rate)
 1: Construct the matrix $\mathbf{L}$ from $\mathbf{X}$ and $\sigma$.
 2: FOR t=1 to ITER DO
 3: Draw a subset $(\mathbf{X}_t, \mathbf{Y}_t)$ from $(\mathbf{X}, \mathbf{Y})$
 4: Graph-regularized sparse coding: computer $\mathbf{A}^*$:
        $\mathbf{A}^* = \arg\min_{\mathbf{A}} \frac{1}{2}||\mathbf{X} - \mathbf{DA}||_F^2 + \lambda \sum_i ||\mathbf{a}_i||_1 + Tr(\mathbf{ALA^T})$.
 5: Compute the active set $S$ (the nonzero support of $\mathbf{A}^*$)
 6: Compute $\boldsymbol{\beta}^*$: Set $\boldsymbol{\beta}^*_{SC} = 0$ and $\boldsymbol{\beta}^*_S = (\mathbf{D}_S^T \mathbf{D}_S + \lambda_2 \mathbf{I})^{-1} \nabla_{\mathbf{A_S}}[C(\mathbf{A}, \mathbf{w})]$
 7: Choose the learning rate $\rho_t = \min(\rho, \rho\frac{t_0}{t})$
 8: Update $\mathbf{D}$ and $\mathbf{W}$ by a projected gradient step:
        $\mathbf{w} = \prod_{\mathbf{w}}[\mathbf{w} - \rho_t \nabla_{\mathbf{w}} C(\mathbf{A}, \mathbf{w})]$
        $\mathbf{D} = \prod_{\mathbf{D}}[\mathbf{D} - \rho_t(\nabla_{\mathbf{D}}(-\mathbf{D}\boldsymbol{\beta}^* \mathbf{A}^T + (\mathbf{X}_t - \mathbf{DA})\boldsymbol{\beta}^{*T})]$
    where $\prod_{\mathbf{w}}$ and $\prod_{\mathbf{D}}$ are respectively orthogonal projections on the embedding spaces of $\mathbf{w}$ and $\mathbf{D}$.
 9: END FOR
Ensure: $\mathbf{D}$ and $\mathbf{w}$
```

Table 2: Accuracy and NMI performance comparisons on all datasets

		KM	KM + SC	EMC	EMC + SC	MMC	MMC + SC	joint EMC	joint MMC
ORL	Acc	0.5250	0.5887	0.6011	0.6404	0.6460	0.6968	0.7250	**0.7458**
	NMI	0.7182	0.7396	0.7502	0.7795	0.8050	0.8043	0.8125	**0.8728**
MNIST	Acc	0.6248	0.6407	0.6377	0.6493	0.6468	0.6581	0.6550	**0.6784**
	NMI	0.5142	0.5397	0.5274	0.5671	0.5934	0.6161	0.6150	**0.6451**
COIL20	Acc	0.6280	0.7880	0.7399	0.7633	0.8075	0.8493	0.8225	**0.8658**
	NMI	0.7621	0.9010	0.8621	0.8887	0.8922	0.8977	0.8850	**0.9127**
CMU-PIE	Acc	0.3176	0.8457	0.7627	0.7836	0.8482	0.8491	0.8250	**0.8783**
	NMI	0.6383	0.9557	0.8043	0.8410	0.9237	0.9489	0.9020	**0.9675**

Algorithm 2 Stochastic gradient descent algorithm for solving (2), with $C(\mathbf{A}, \mathbf{w})$ as defined in (9)

```
Require: $\mathbf{X}, \sigma$; $\lambda$; $\mathbf{D}_0$ and $\mathbf{w}_0$ (initial dictionary and classifier parameter); ITER (number of iterations); $t_0, \rho$ (learning rate)
 1: Construct the matrix $\mathbf{L}$ from $\mathbf{X}$ and $\sigma$.
 2: Estimate the initialization of $y_i$ and $r_i$ by pre-clustering, $i = 1, 2, ..., N$
 3: FOR t=1 to ITER DO
 4: Conduct the same step 4-7 in Algorithm 1.
 5: Update $\mathbf{D}$ and $\mathbf{W}$ by a projected gradient step, based on the derivates of (9) over $\mathbf{a}_i$ and $\mathbf{w}$ (10).
 6: Update $y_i$ and $r_i$ using the current $\mathbf{w}$ and $\mathbf{a_i}$, $i = 1, 2, ..., N$.
 7: END FOR
Ensure: $\mathbf{D}$ and $\mathbf{w}$
```

4 Experiments

4.1 Dataset and Evaluation

We conduct our clustering experiments on four popular real datasets, which are summarized in Table 1. We apply two widely-used measures to evaluate the performance of the clustering methods: the accuracy and the Normalized Mutual Information(NMI) [Zheng *et al.*, 2011], [Cheng *et al.*, 2010]. Suppose the predicted label of the $\mathbf{x}_i$ is $\hat{y}_i$ which is produced by the clustering method, and y_i is the ground truth label. The accuracy is defined as:

Table 1: Comparison of all datasets

Name	Number of Images	Class	Dimension
ORL	400	10	1,024
MNIST	70,000	10	784
COIL20	1,440	20	1,024
CMU-PIE	41,368	68	1,024

$$Acc = \frac{\mathbb{1}_{\Phi(\hat{y}_i) \neq y_i}}{n}, \tag{11}$$

where $\mathbb{1}$ is the indicator function, and Φ is the best permutation mapping function [Lovász and Plummer, 2009]. On the other hand, suppose the clusters obtained from the predicted labels $\{\hat{y}_i\}_{i=1}^n$ and $\{y_i\}_{i=1}^n$ as $\hat{C}$ and C, respectively. The mutual information between $\hat{C}$ and C is defined as:

$$MI(\hat{C}, C) = \sum_{\hat{c} \in \hat{C}, c \in C} p(\hat{c}, c) \log \frac{p(\hat{c}, c)}{p(\hat{c})p(c)}, \tag{12}$$

where $p(\hat{c})$ and $p(c)$ are the probabilities that a data point belongs to the clusters $\hat{C}$ and C, respectively, and $p(\hat{c}, c)$ is the probability that a data point jointly belongs to $\hat{C}$ and C.

[1] To avoid ambiguity, if y_i and r_i are the same, i.e., the max value is reached by two cluster prototypes simultaneously in current iteration, then we ignore the gradient update corresponding to $\mathbf{a}_i$.

The normalized mutual information(NMI) is defined as:

$$NMI(\hat{C}, C) = \frac{MI(\hat{C}, C)}{\max\{H(\hat{C}), H(C)\}}, \tag{13}$$

where $H(\hat{C})$ and $H(C)$ are the entropies of $\hat{C}$ and C, respectively. NMI takes values between [0,1].

4.2 Comparison Experiments

Comparison Methods We compare the following eight methods on all four datasets:

- **KM:** K-Means clustering on the input data.
- **KM + SC:** A dictionary $\mathbf{D}$ is first learned from the input data by K-SVD [Elad and Aharon, 2006]. Then KM is performed on the graph-regularized sparse code features (1) over $\mathbf{D}$
- **EMC:** Entropy-minimization clustering, by minimizing (4) on the input data.
- **EMC + SC:** EMC performed on the graph-regularized sparse codes over the pre-learned K-SVD dictionary $\mathbf{D}$.
- **MMC:** Maximum-margin clustering [Xu and Schuurmans, 2005].
- **MMC + SC:** MMC performed on the graph-regularized sparse codes over the pre-learned K-SVD dictionary $\mathbf{D}$.
- **Joint EMC:** The proposed joint optimization (2), with $C(\mathbf{A}, \mathbf{w})$ as defined in (4).
- **Joint MMC:** The proposed joint optimization (2), with $C(\mathbf{A}, \mathbf{w})$ as defined in (9).

All images are first reshaped into vectors, and PCA is then applied to reducing the data dimensionality by keeping 98% information, which is also used in [Zheng *et al.*, 2011] to improving efficiency. The multi-class MMC algorithm is implemented based on the publicly available CPMMC code for two-class clustering [Zhao *et al.*, 2008], following the multi-class case descriptions in the original paper. For all algorithms that involve graph-regularized sparse coding, the graph regularization parameter α is fixed to be 1, and the dictionary size p is 128 by default. For joint EMC and joint MMC, we set ITER as 30, ρ as 0.9, and t_0 as 5. Other parameters in competing methods are tuned in cross-validation experiments.

Comparison Analysis All the comparison results (accuracy and NMI) are listed in Table. 2, from which we could conclude the following:

- **1:** The joint EMC and joint MMC methods each outperform their "non-joint" counterparts, e.g., EMC + SC and MMC + SC, respectively. For example, on the *ORL* dataset, joint MMC surpasses MMC + SC by around 5% in accuracy and 7% in NMI. Those demonstrate that the key contribution of this paper, i.e., jointly optimizing the sparse coding and clustering steps, indeed leads to improved performances.
- **2:** KM + SC, EMC + SC, and MMC + SC all outperform their counterparts using raw input data, which verifies that sparse codes are effective features that help improve the clustering discriminability.

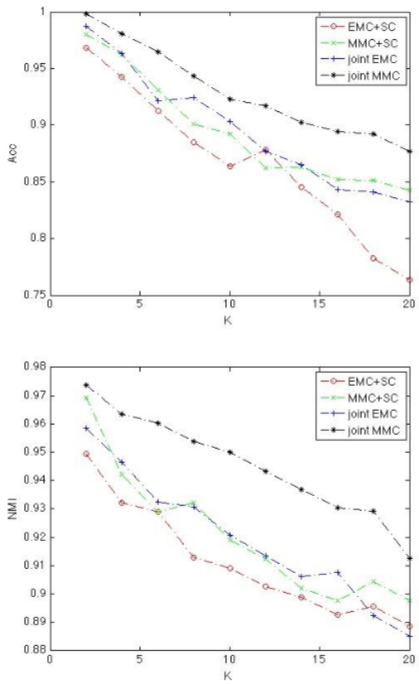

Figure 1: The clustering accuracy and NMI measurements versus the number of clusters K.

- **3:** The joint MMC obtains the best performances in all cases, outperforming the others, including joint EMC, with significant margins. The MMC + SC obtains the second best performance for the last three datasets (for ORL, it is joint EMC that ranks the second). The above facts reveal the power of the max-margin loss (9).

Varying the number of clusters On the COIL20 dataset, We re-conduct the clustering experiments with the cluster number K ranging from 2 to 20, using EMC + SC, MMC + SC, joint EMC, and joint MMC. For each K except for 20, 10 test runs are conducted on different randomly chosen clusters, and the final scores are obtained by averaging over the 10 tests. Fig. 1 shows the clustering accuracy and NMI measurements versus the number of clusters. It is revealed that the two joint methods consistently outperforms their non-joint counterparts. When K goes up, the performances of joint methods seem to degrade less slowly.

Initialization and Parameters As observed in our experiments, a good initialization of $\mathbf{D}$ and $\mathbf{w}$ can affect the final results notably. We initialize Joint EMC by the $\mathbf{D}$ and $\mathbf{w}$ solved from EMC + SC, and Joint MMC by the solutions from MMC + SC, respectively.

There are two parameters that we need to set empirically: the graph regularization parameter α, and the dictionary size p. The regularization term imposes stronger smoothness constraints on the sparse codes when α grows larger. Also, while a compact dictionary is more desirable computationally, more

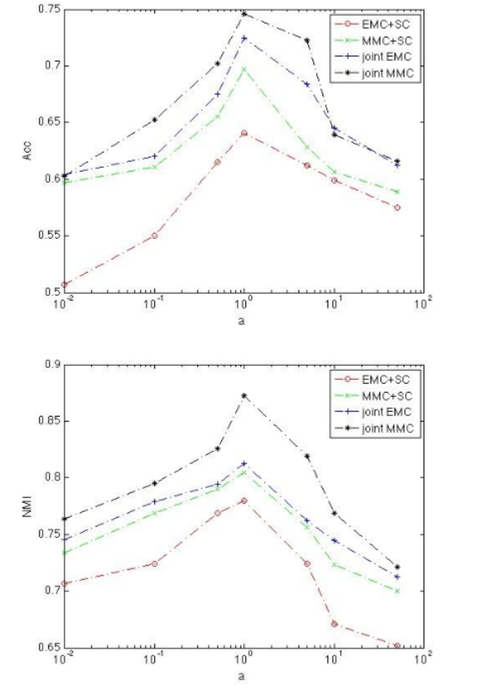

Figure 2: The clustering accuracy and NMI measurements versus the parameter choices of α.

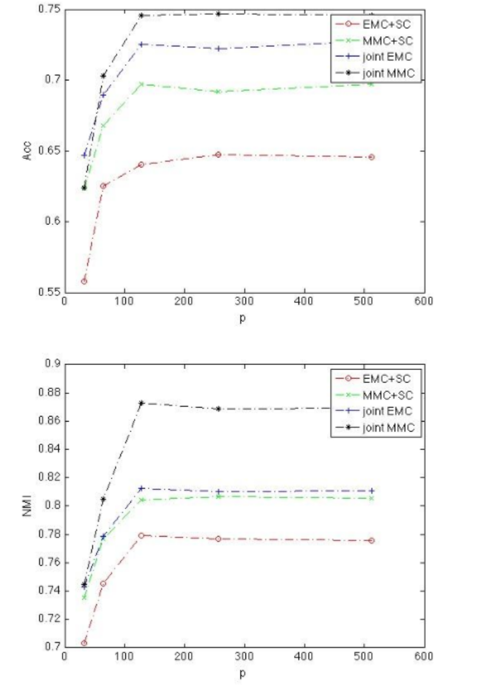

Figure 3: The clustering accuracy and NMI measurements versus the parameter choices of p.

redundant dictionaries may lead to less cluttered features that can be better discriminated. We investigate how the clustering performances EMC + SC, MMC + SC, joint EMC, and joint MMC change on the ORL dataset, with various α and p values. As depicted in Fig. 2 and 3, we observe that:

- **1:** While α goes up, the accuracy result will first grow up then go down (the peak is around α =1). That could be interpreted as when α is too small, the local manifold information is not sufficiently encoded. On the other hand, when α turns overly large, the sparse codes are "over-smoothened" with a reduced discriminability.

- **2:** Increasing dictionary size p will first improve the accuracy sharply, which however soon reaches a plateau. Thus in practice, we keep a medium dictionary size p =128 for all experiments.

5 Conclusion

We propose a joint framework to optimize sparse coding and discriminative clustering simultaneously. We adopt graph regularized sparse codes as the feature to be learned, and design two clustering-oriented cost functions, by entropy-minimization and maximum-margin principles, respectively. The task-driven bi-level optimization mutually reinforces both sparse coding and clustering steps. Experiments on several benchmark datasets verify the remarkable performance improvements led by the proposed joint optimization.

6 Acknowledgement

We sincerely appreciate the great help from Associate Professor Linli Xu, affiliated with University of Science and Technology of China (USTC), for sharing with us her original implementation of multi-class MMC algorithm [Xu and Schuurmans, 2005].

7 Appendix

We recall the following lemma [7.1] in [Mairal *et al.*, 2012]:

Theorem 7.1 (Regularity of the elastic net solution). *Consider the formulation in (1) (we may drop the last term to obtain the exact elastic net form, without affecting the differentiability conclusions). Let* $\lambda_2 > 0$ *, and* $\mathcal{X}$ *is assumed to be compact. Then,*

- $\mathbf{a}$ *is uniformly Lipschitz on* $\mathcal{X} \times \mathcal{D}$

- *Let* $\mathbf{D} \in \mathcal{D}$*,* σ *be a positive scalar and* $\mathbf{s}$ *be a vector in* $\{-1, 0, 1\}^p$*. Define* $K_s(\mathbf{D}, \sigma)$ *as the set of vectors* $\mathbf{x}$ *satisfying for all* j *in* $\{1, ..., p\}$*,*

$$\begin{array}{l} |\mathbf{d}_j^T(\mathbf{x} - \mathbf{D}\mathbf{a}) - \lambda_2 \mathbf{a}[j]| \leq \lambda_1 - \sigma \quad if \quad \mathbf{s}[j] = 0 \\ \mathbf{s}[j]\mathbf{a}[j] \geq \sigma \quad if \quad \mathbf{s}[j] \neq 0 \end{array} \tag{14}$$

Then there exists $\kappa > 0$ *independent of* $\mathbf{s}, \mathbf{D}$ *and* σ *so that for all* $\mathbf{x} \in K_s(\mathbf{D}, \sigma)$*, the function* $\mathbf{a}$ *is twice continuously differentiable on* $B_{\kappa\sigma}(\mathbf{x}) \times B_{\kappa\sigma}(\mathbf{D})$*, where* $B_{\kappa\sigma}(\mathbf{x})$ *and* $B_{\kappa\sigma}(\mathbf{D})$ *denote the open balls of radius* $\kappa\sigma$ *respectively centered on* $\mathbf{x}$ *and* $\mathbf{D}$*.*

References

[Barber and Agakov, 2005] David Barber and Felix V Agakov. Kernelized infomax clustering. In *Advances in Neural Information Processing Systems*, pages 17–24, 2005.

[Biernacki *et al.*, 2000] Christophe Biernacki, Gilles Celeux, and Gérard Govaert. Assessing a mixture model for clustering with the integrated completed likelihood. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(7):719–725, 2000.

[Cheng *et al.*, 2010] Bin Cheng, Jianchao Yang, Shuicheng Yan, Yun Fu, and Thomas S Huang. Learning with l1 graph for image analysis. *Image Processing, IEEE Transactions on*, 19(4):858–866, 2010.

[Crammer and Singer, 2002] Koby Crammer and Yoram Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *The Journal of Machine Learning Research*, 2:265–292, 2002.

[Dai and Hu, 2010] Bo Dai and Baogang Hu. Minimum conditional entropy clustering: A discriminative framework for clustering. In *ACML*, pages 47–62, 2010.

[Elad and Aharon, 2006] Michael Elad and Michal Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *Image Processing, IEEE Transactions on*, 15(12):3736–3745, 2006.

[Lee and Lin, 2013] Ching-Pei Lee and Chih-Jen Lin. A study on l2-loss (squared hinge-loss) multiclass svm. *Neural computation*, 25(5):1302–1323, 2013.

[Lee *et al.*, 2006] Honglak Lee, Alexis Battle, Rajat Raina, and Andrew Y Ng. Efficient sparse coding algorithms. In *Advances in neural information processing systems*, pages 801–808, 2006.

[Li *et al.*, 2004] XR Li, Keshu Zhang, and Tao Jiang. Minimum entropy clustering and applications to gene expression analysis. In *Computational Systems Bioinformatics Conference, 2004. CSB 2004. Proceedings. 2004 IEEE*, pages 142–151. IEEE, 2004.

[Li *et al.*, 2009] Yu-Feng Li, Ivor W Tsang, James T Kwok, and Zhi-Hua Zhou. Tighter and convex maximum margin clustering. In *International Conference on Artificial Intelligence and Statistics*, pages 344–351, 2009.

[Lovász and Plummer, 2009] László Lovász and MD Plummer. *Matching theory*, volume 367. American Mathematical Soc., 2009.

[Mairal *et al.*, 2012] Julien Mairal, Francis Bach, and Jean Ponce. Task-driven dictionary learning. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(4):791–804, 2012.

[Ng *et al.*, 2002] Andrew Y Ng, Michael I Jordan, Yair Weiss, et al. On spectral clustering: Analysis and an algorithm. *Advances in neural information processing systems*, 2:849–856, 2002.

[Roth and Lange, 2003] Volker Roth and Tilman Lange. Feature selection in clustering problems. In *Advances in neural information processing systems*, page None, 2003.

[Sprechmann and Sapiro, 2010] Pablo Sprechmann and Guillermo Sapiro. Dictionary learning and sparse coding for unsupervised clustering. In *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, pages 2042–2045. IEEE, 2010.

[Wang *et al.*, 2015] Zhangyang Wang, Nasser M Nasrabadi, and Thomas S Huang. Semisupervised hyperspectral classification using task-driven dictionary learning with laplacian regularization. 2015.

[Wright *et al.*, 2009] John Wright, Allen Y Yang, Arvind Ganesh, Shankar S Sastry, and Yi Ma. Robust face recognition via sparse representation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(2):210–227, 2009.

[Xu and Schuurmans, 2005] Linli Xu and Dale Schuurmans. Unsupervised and semi-supervised multi-class support vector machines. In *AAAI*, volume 5, 2005.

[Xu *et al.*, 2004] Linli Xu, James Neufeld, Bryce Larson, and Dale Schuurmans. Maximum margin clustering. In *Advances in neural information processing systems*, pages 1537–1544, 2004.

[Yang *et al.*, 2012] Jianchao Yang, Zhaowen Wang, Zhe Lin, Xianbiao Shu, and Thomas Huang. Bilevel sparse coding for coupled feature spaces. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2360–2367. IEEE, 2012.

[Yang *et al.*, 2014a] Yingzhen Yang, Feng Liang, Shuicheng Yan, Zhangyang Wang, and Thomas S Huang. On a theory of nonparametric pairwise similarity for clustering: Connecting clustering to classification. In *Advances in Neural Information Processing Systems*, pages 145–153, 2014.

[Yang *et al.*, 2014b] Yingzhen Yang, Zhangyang Wang, Jianchao Yang, Jiawei Han, and Thomas S Huang. Regularized l1-graph for data clustering. In *British Machine Vision Conference*, 2014.

[Yang *et al.*, 2014c] Yingzhen Yang, Zhangyang Wang, Jianchao Yang, Jiangping Wang, Shiyu Chang, and Thomas S Huang. Data clustering by laplacian regularized l1-graph. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014.

[Zhao *et al.*, 2008] Bin Zhao, Fei Wang, and Changshui Zhang. Efficient maximum margin clustering via cutting plane algorithm. In *SDM*, pages 751–762. SIAM, 2008.

[Zheng *et al.*, 2011] Miao Zheng, Jiajun Bu, Chun Chen, Can Wang, Lijun Zhang, Guang Qiu, and Deng Cai. Graph regularized sparse coding for image representation. *Image Processing, IEEE Transactions on*, 20(5):1327–1336, 2011.