21st International Conference on Pattern Recognition (ICPR 2012)
November 11-15, 2012. Tsukuba, Japan

Adaptive Graph Regularized Nonnegative Matrix Factorization via Feature Selection

Jing-Yan Wang, Islam Almasri and Xin Gao *
Mathematical and Computer Sciences and Engineering Division,
King Abdullah University of Science and Technology (KAUST),
Thuwal 23955-6900, Saudi Arabia
{*Jingyan.Wang, Islam.Almasri, Xin.Gao*}*@kaust.edu.sa*

Abstract

Nonnegative Matrix Factorization (NMF), a popular compact data representation method, fails to discover the intrinsic geometrical structure of the data space. Graph regularized NMF (GrNMF) is proposed to avoid this limitation by regularizing NMF with a nearest neighbor graph constructed from the input data feature space. However using the original feature space directly is not appropriate because of the noisy and irrelevant features. In this paper, we propose a novel data representation algorithm by integrating feature selection and graph regularization for NMF. Instead of using a fixed graph as GrNMF, we regularize NMF with an adaptive graph constructed according to the feature selection results. A uniform object is built to consider feature selection, NMF and adaptive graph regularization jointly, and a novel algorithm is developed to update the graph, feature weights and factorization parameters iteratively. Data clustering experiment shows the efficacy of the proposed method on the Yale database.

1 Introduction

Nonnegative matrix factorization (NMF) [3] decomposes the data matrix as a product of two matrices that are constrained by having nonnegative elements. This method results in a reduced representation of the original data that can be seen as a parts-based representation technique. However, NMF performs this learning in the Euclidean space. It fails to discover the intrinsic geometrical and discriminative structure of the data space. Recently, Cai et al. improved the transitional NMF to Graph regularized Nonnegative Matrix Factorization (GrNMF) in [1] to avoid this limitation by incorporating a geometry-based regularizer. In GrNMF, the geometrical information of the data space is encoded by constructing a nearest neighbor graph, and then the matrix factorization is built with respect to the graph structure. The key component of GrNMF is the graph. In the original GrNMF algorithm, the graph is constructed according to the original input feature space. The nearest neighbors of a data point are found by comparing the Euclidean distances between pairs of data points, while the weights of edges are also estimated in the Euclidean space. However, it is well acknowledged that in some data clustering and classification problems, using the original feature space directly is not appropriate because of the noisy and irrelevant features. If we use the original features of samples to construct the graph for GrNMF, problems might be caused because the graph itself is not a good representation of the manifold. Therefore, it is desirable to develop an effective feature selection algorithm by identifying those relevant features and to represent the data accurately. In fact, graph regularization has been applied to feature selection task [6], but the graph used in [6] is also fixed and will not be effected by the feature selection results.

In this paper, we investigate the inherent relationship between feature selection and NMF with graph regularization. The feature selection will provide a new data space for the graph construction of GrNMF, and GrNMF will also provide the criterion for feature selection. We will unify the feature selection and GrNMF within a single object function and repeat their optimizations alternately, so that they will effect the learning of each other. We propose a unified feature selection and graph regularization algorithm for NMF, referred to as **Adap**tive **G**raph **r**egularized **NMF** with **f**eature **s**election (AdapGrNMF$_{fs}$).

The rest of the paper is organized as follows: We briefly review the GrNMF in Section 2. We then intro-

*To whom all correspondence should be addressed.

duce our framework, AdapGrNMF, in Section 3. Experimental results on clustering are presented in Section 4. Finally, conclusive remarks are presented in Section 5.

2 Overview of Graph Regularized NMF

Given N nonnegative data points $\mathcal{X} = \{x_1, \cdots, x_N\} \in \mathbb{R}_+^D$ represented as a data matrix $X = [x_1, \cdots, x_N] \in \mathbb{R}_+^{D \times N}$, NMF aims to find two nonnegative matrices H and W whose product can well approximate the original matrix X as $X \approx HW$, where $H \in \mathbb{R}^{D \times R}$, and $W \in \mathbb{R}^{R \times N}$, with $R \leq D$. H can be regarded as containing a set of basis vectors, and w_n (the n-th column of W) can be regarded as a coding vector or new representations of the n-th data point with respect to the basis H. The most commonly used cost function is base on Squared Euclidean Distance (SED) between the two matrices :

$$O^{NMF}(H, W) = ||X - HW||^2 \tag{1}$$

The above objective function can be minimized by the algorithm proposed by Lee and Seung [3].

By performing this learning in the Euclidean space, NMF fails to discover the intrinsic geometrical and discriminative structure of the data space [1]. To avoid this limitation, Cai et al. [1] introduced the GrNMF algorithm, by incorporating a geometry-based regularizer. They modeled the local geometric structure by a P nearest neighbor graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, A\}$ on a scatter of data points. The node set $\mathcal{V}$ corresponds to N data points. $\mathcal{E}$ is the edge set, and $(n, m) \in \mathcal{E}$ if $x_m \in \mathcal{N}_n$, where $\mathcal{N}_n$ is the P nearest neighbors of x_n in $\mathcal{X}$ determined by the SED metric. $A \in R^{N \times N}$ is the weight matrix on the graph with A_{nm} equal to the weight of edge (n, m), which can be defined by Gaussian kernel weighting as $A_{nm} = e^{-\frac{||x_n - x_m||^2}{\sigma^2}}$, if $(n, m) \in \mathcal{E}$; 0, otherwise.

With the weight matrix A defined above, we can use the following **G**raph **r**egularization term to measure the smoothness of the low-dimensional coding vector representations in W

$$O^{Gr}(W; A) = \frac{1}{2} \sum_{n,m=1}^{N} ||w_n - w_m||^2 A_{nm} \tag{2}$$

By minimizing $O^{Gr}(W; A)$ regarding to W, we expect that if two data points x_n and x_m are close (i.e.,A_{nm} is big), w_n and w_m are also close to each other.

Combining this geometry-based regularizer $O^{Gr}(W; A)$ with the original NMF objective function $O^{NMF}(H, W)$ leads to the loss function of GrNMF [1]:

$$O^{GrNMF}(H, W; A) = O^{NMF}(H, W) + \alpha O^{Gr}(W; A) \tag{3}$$

in which α is the tradeoff parameter to balance the two terms. H and W can be solved in an iterative manner by updating them alternately [1].

3 Adaptive Graph Regularized NMF with Feature Selection

In this section, we will integrate the feature selection into the GrNMF algorithm.

3.1 Graph Adaptive to Selected Features

Feature selection tries to scale each feature, and thus obtains a weighted feature space [2], which is parameterized by a $D \times D$ nonnegative diagonal matrix Λ, whose diagonal entry Λ_{dd} is the scaling factor for the d-th feature. The original data matrix X and the basis matrix H with feature weighting can be represented as $\widetilde{X} = \Lambda X$ and $\widetilde{H} = \Lambda H$. By substituting $\widetilde{X}$ and $\widetilde{H}$ into (1), we have the augmented objective function for NMF with feature selection in an enlarged parameter space

$$O^{NMF_{fs}}(H, W, \Lambda) = ||\Lambda(X - HW)||^2 \tag{4}$$

With the new feature space defined by feature weight matrix Λ, the graph should also be updated to be adaptive to the selected features. First, the P nearest neighbors $\mathcal{N}_n^\Lambda$ of the n-th data point should be updated according to the Λ-weighted square Euclidean distance, i.e.,

$$d^\Lambda(x_i, x_j) = ||x_n - x_m||_\Lambda^2 = \sum_{d=1}^{D} \Lambda_{dd}^2 (x_{di} - x_{dj})^2 \tag{5}$$

Thus graph adaptive to Λ is also updated as $\mathcal{G}^\Lambda = \{\mathcal{V}, \mathcal{E}^\Lambda, A^\Lambda\}$ accordingly.

With the adaptive graph $\mathcal{G}^\Lambda$, we propose the **Adap**tive **G**raph **r**egularization term as

$$\begin{aligned} O^{AdapGr}(W; A^\Lambda) &= \frac{1}{2} \sum_{n,m=1}^{N} ||w_n - w_m||^2 {A^\Lambda}_{nm} \\ &= Tr(W L^\Lambda W^\top) \end{aligned} \tag{6}$$

where D^Λ is a diagonal matrix whose entries are column sums of A^D, $D_{nn} = \sum_{n=1}^{N} A_{nm}$ and $L^\Lambda = D^\Lambda - A^\Lambda$ is the graph Laplacian.

3.2 Proposed Algorithm

Combining (4) and (6) leads to the object function of our AdapGrNMF with **f**eature **s**election — AdapGrNMF$_{fs}$ algorithm:

$$\begin{aligned}&\min_{H,W,\Lambda} O^{AdapGrNMF_{fs}}(H,W,\Lambda)\\&\quad = Tr(\Lambda^2 XX^\top) - 2Tr(\Lambda^2 XW^\top H^\top)\\&\qquad + Tr(\Lambda^2 HWW^\top H^\top) + \alpha Tr(WL^\Lambda W^\top)\\&\quad s.t.\ H \geq 0,\ W \geq 0,\\&\qquad \Lambda\ is\ diagonal,\ \sum_{d=1}^{D}\Lambda_{dd} = 1,\ \Lambda_{dd} \geq 0.\end{aligned} \tag{7}$$

Since direct optimization to (7) is difficult, we propose an iterative, two-step strategy to alternately optimize (H, W) and Λ.

On optimizing (H, W): By fixing Λ and updating $\mathcal{G}^\Lambda$ with L^Λ, the optimization problem (7) is reduced to

$$\begin{aligned}&\min_{H,W} Tr(\Lambda^2 XX^\top) - 2Tr(\Lambda^2 XW^\top H^\top)\\&\qquad + Tr(\Lambda^2 HWW^\top H^\top) + \alpha Tr(WL^\Lambda W^\top)\\&\quad s.t.\ H \geq 0,\ W \geq 0.\end{aligned} \tag{8}$$

By setting the partial derivatives of Lagrange of the above optimization problem with respect to H and W to zero, and using KKT conditions, we get the following updating rules:

$$\begin{aligned}h_{dr} &\leftarrow \frac{(\Lambda^2 XW^\top)_{dr}}{(\Lambda^2 HWW^\top)_{dr}} h_{dr}\\ w_{rn} &\leftarrow \frac{(H^\top \Lambda^2 X + \alpha WA^\Lambda)_{rn}}{(H^\top \Lambda^2 HW + \alpha WD^\Lambda)_{rn}} w_{rn}\end{aligned} \tag{9}$$

On optimizing Λ: By fixing H, W and removing the terms irrelevant to Λ, the optimization problem (7) becomes

$$\begin{aligned}&\min_{\Lambda} Tr[\Lambda^2(XX^\top - 2XW^\top H^\top + HWW^\top H^\top)]\\&\quad = Tr[\Lambda^2(YY^\top)]\\&\quad s.t.\ \Lambda\ is\ diagonal,\ \sum_{d=1}^{D}\Lambda_{dd} = C,\ \Lambda_{dd} \geq 0.\end{aligned} \tag{10}$$

where $Y = X - HW$. Since Λ is diagonal, we introduce a vector $\lambda = [\lambda_1, \cdots, \lambda_D]^\top$ such that $\Lambda = diag(\lambda)$. (10) turns to

$$\begin{aligned}&\min_{\lambda} Tr[diag(\lambda^2)(Y^\top Y)] = \sum_{d=1}^{D}\lambda_d^2 \sum_{n=1}^{N} y_{dn}^2 = \sum_{d=1}^{D}\lambda_d^2 e_d\\&s.t.\ \sum_{d=1}^{D}\lambda_d = C,\ \lambda_d \geq 0.\end{aligned} \tag{11}$$

where $e_d = \sum_{n=1}^{N} y_{dn}^2$. The optimization of (11) with respect to the feature weights λ can be solved as a standard Quadratic Programming (QP) problem.

The AdapGrNM$_{fs}$ algorithm is summarized in Algorithm 1.

Algorithm 1 AdapGrNM$_{fs}$ Algorithm.

Input: Original data matrix X;
Input: Initial factorization matrices H^0 and W^0;
Initialize the feature weight variables as $\lambda_d^0 = \frac{1}{D}, d = 1, \cdots, D$;
for $t = 1, \cdots, T$ **do**
  Update the graph $\mathcal{G}^{\Lambda^t}$ and L^{Λ^t} according to $\Lambda^{t-1} = diag(\lambda^{t-1})$ as in section 3.1;
  Update the matrices H^t and W^t as in (9);
  Update the feature weights λ^t as in (11);
end for
Output: $H = H^{t-1}$, $W = W^{t-1}$ and $\lambda = \lambda^{t-1}$.

4 Experiment

In this section, we evaluate the performance of the proposed AdapGrNMF$_{fs}$ algorithm for face image clustering on the Yale face database.

4.1 Database and Setup

The Yale database contains 165 gray scale images of 15 individuals. There are 11 face images resized into 32×32 for each individual. Thus, each image is represented by a 1024-dimensional vector in image space, and the image set will be organized as a data matrix X, the size of which is $1,024 \times 165$.

Using our AdapGrNMF$_{fs}$ algorithm, the images are firstly represented as coding vectors $\{w_n\}$, $n = 1, \cdots, N$, and then the coding vectors will be input to K-means for the clustering procedure. We set the dimensionality of the new space R to be the same as the number of clusters (here an individual is a cluster). In this clustering experiment, we evaluated the performance with the Clustering Accuracy index.

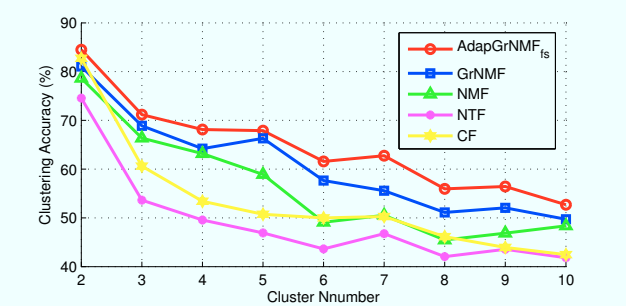

Figure 1. Clustering Performance on Yale Face Database.

4.2 Results

We now consider the clustering performance of different image representation methods. Performance of AdapGrNMF$_{fs}$ is compared with four different algorithms, namely, NMF, GrNMF, Nonnegative Tensor Factorization (NTF) [4] and Concept Factorization (CF) [5]. We conducted the evaluations with different numbers of clusters varying from 2 to 10. For the fixed cluster number R, we randomly chose R categories from the data set, and mix the images of these R categories as the collection X for clustering. Figure 1 summarizes the results. As can be seen from Figure 1, Graph optimized representations, i.e. GrNMF and AdapGrNMF$_{fs}$, significantly outperform generatively learned image representation methods, i.e. NMF, NTF and CF structures. The results summarized in Figure 1 also show that the proposed AdapGrNMF$_{fs}$ method consistently achieves the best performance (highest clustering accuracy rate). Even though the GrNMF already achieves very high performance, its adaptive graph and feature selection extension AdapGrNMF$_{fs}$ further improves performance in terms of clustering accuracy. Compared to GrNMF, AdapGrNMF$_{fs}$ improves the accuracy rate by 7.19%, which is significant. This is strong proof of our assumption that the graph regularization and feature selection can benefit from each other in the data representation.

The feature weights learned by AdapGrNMF$_{fs}$ are also shown in Figure 2 (c). It can be seen that the irrelevant region in the image, such as lower-right and lower-left corners, are assigned with smaller weights. While the important regions in the face image, such as the area around eyes, are assigned with larger weights.

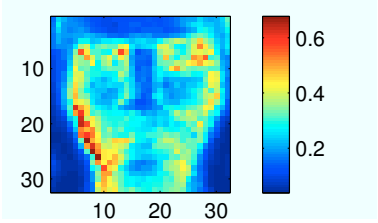

Figure 2. The feature weights learned by AdapGrNMF$_{fs}$.

5 Conclusion

This paper proposes a novel NMF data representation algorithm by learning the feature weights, reconstructing the adaptive graph and factorizing the data matrix iteratively. Performance evaluation is conducted on the Yale database. Experimental results show that the proposed AdapGrNMF$_{fs}$ algorithm significantly outperforms the standard GrNMF and other state-of-the-art methods.

Acknowledgement

The study was supported by grants from the Key Laboratory of High Performance Computing and Stochastic Information Processing, Ministry of Education of China, and the King Abdullah University of Science and Technology (KAUST).

References

[1] D. Cai, X. He, J. Han, and T. S. Huang. Graph Regularized Nonnegative Matrix Factorization for Data Representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(8):1548–1560, Aug 2011.

[2] B. Chen, H. Liu, J. Chai, and Z. Bao. Large Margin Feature Weighting Method via Linear Programming. *IEEE Transactions on Knowledge and Data Engineering*, 21(10):1475–1488, OCT 2009.

[3] D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In *NIPS*, pages 556–562, 2000.

[4] W. Peng and T. Li. On the equivalence between nonnegative tensor factorization and tensorial probabilistic latent semantic analysis. *Applied Intelligence*, 35(2):285–295, OCT 2011.

[5] Wei Hua and Xiaofei He. Discriminative concept factorization for data representation. *Neurocomputing*, pages 3800–7.

[6] Zenglin Xu, I. King, M.-T. Lyu, and Rong Jin. Discriminative Semi-Supervised Feature Selection Via Manifold Regularization. *IEEE Transactions on Neural Networks*, 21:1033–47, July 2010.