Automatic Rank Determination in Projective Nonnegative Matrix Factorization

Zhirong Yang, Zhanxing Zhu, and Erkki Oja

Department of Information and Computer Science*
Aalto University School of Science and Technology
P.O.Box 15400, FI-00076, Aalto, Finland,
{zhirong.yang,zhanxing.zhu,erkki.oja}@tkk.fi

Abstract. Projective Nonnegative Matrix Factorization (PNMF) has demonstrated advantages in both sparse feature extraction and clustering. However, PNMF requires users to specify the column rank of the approximative projection matrix, the value of which is unknown beforehand. In this paper, we propose a method called ARDPNMF to automatically determine the column rank in PNMF. Our method is based on automatic relevance determination (ARD) with Jeffrey's prior. After deriving the multiplicative update rule using the expectation-maximization technique for ARDPNMF, we test it on various synthetic and real-world datasets for feature extraction and clustering applications to show the effectiveness of our algorithm. For FERET faces and the Swimmer dataset, interpretable number of features are obtained correctly via our algorithm. Several UCI datasets for clustering are also tested, in which we find that ARDPNMF can estimate the number of clusters quite accurately with low deviation and good cluster purity.

1 Introduction

Since its introduction by Lee and Seung [1] as a new machine learning method, Nonnegative Matrix Factorization (NMF) has been applied successfully in many applications, including signal processing, text clustering and gene expression studies, etc. (see [2] for a survey). Recently much progress for NMF has been reported both in theory and practice. Also there are several variants to extend original NMF, (e.g. [3–5]). *Projective Nonnegative Matrix Factorization* (PNMF), introduced in [6–8], approximates a data matrix by its nonnegative subspace projection. Compared with NMF, the PNMF has a number of benefits such as better generalization, a sparser factorizing matrix without ambiguity, and close relation to principal component analysis, which are advantageous in both feature extraction and clustering [8].

However, a remaining difficult problem is how to determine the dimensionality of the approximating subspace in PNMF in practical applications. In most

* Supported by the Academy of Finland in the project *Finnish Centre of Excellence in Adaptive Informatics Research.*

cases, one has to guess a suitable component number, e.g. the number of features needed to encode facial images. Such trial-and-error procedures can be tedious in practice. In this work, we propose a variant of PNMF called ARDPNMF that can automatically determine dimensionality of factorizing matrix. Our method is based on the *automatic relevance determination* (ARD) [9] technique which has been used in Bayesian PCA [10] and adaptive sparse supervised learning [11]. The proposed algorithm is free of user-specified parameters. Such property is especially desired for exploratory analysis of the data structure. Empirical results on several synthetic and real-world datasets demonstrate that our method can effectively discover the number of features or clusters.

This paper is organized as follows. In Section 2, we summarize the essence of PNMF and model selection in NMF. Then, we derive our algorithm ARDPNMF in Section 3. In Section 4, the experimental results of the proposed algorithm on a variety of synthetic and real datasets for feature extraction and clustering are presented. Section 5 concludes the paper.

2 Related Work

2.1 Projective Nonnegative Matrix Factorization

Given a nonnegative input matrix $\mathbf{X} \in \mathbb{R}_+^{m \times n}$, Projective Nonnegative Matrix Factorization (PNMF) seeks a nonnegative matrix $\mathbf{W} \in \mathbb{R}_+^{m \times r}$ such that

$$\mathbf{X} \approx \mathbf{W}\mathbf{W}^T\mathbf{X}. \tag{1}$$

Compared with the NMF approximation scheme, $\mathbf{X} \approx \mathbf{W}\mathbf{H}$, PNMF replaces $\mathbf{H}$ matrix with $\mathbf{W}^T\mathbf{X}$. As a result, PNMF has a number of advantages over NMF [8], including high sparseness in the factorizing matrix $\mathbf{W}$, closer equivalence to clustering, easy nonlinear extension to a kernel version, and fast approximation of newly coming samples without heavy re-computation. The name "projective" comes from the fact that $\mathbf{W}\mathbf{W}^T$ is very close to a projection matrix because the $\mathbf{W}$ learned by PNMF is highly orthogonal. It can be made fully orthogonal by post-processing.

PNMF based on the Euclidean distance solves the following optimization problem:

$$\underset{\mathbf{W} \geq 0}{\text{minimize}}\, J_F(\mathbf{W}) = \frac{1}{2} \sum_{ij} \left[X_{ij} - \left(\mathbf{W}\mathbf{W}^T\mathbf{X} \right)_{ij} \right]^2. \tag{2}$$

Previously, Yuan and Oja [6] presented a multiplicative algorithm that iteratively applies the following update rule for the above minimization:

$$W'_{ik} \leftarrow W_{ik} \frac{A_{ik}}{B_{ik}} \tag{3}$$

$$\mathbf{W}^{\text{new}} \leftarrow \mathbf{W}' / \|\mathbf{W}'\|, \tag{4}$$

where $\mathbf{A} = 2\mathbf{X}\mathbf{X}^T\mathbf{W}$, $\mathbf{B} = \mathbf{W}\mathbf{W}^T\mathbf{X}\mathbf{X}^T\mathbf{W} + \mathbf{X}\mathbf{X}^T\mathbf{W}\mathbf{W}^T\mathbf{W}$, and $\|\mathbf{W}'\|$ calculates the square root of the maximal eigenvalue of $\mathbf{W}'^T\mathbf{W}'$.

2.2 Model Selection in NMF

In NMF, Tan and Févotte [12] addressed the model selection problem based on automatic relevance determination. First, a prior is added on the columns and rows of matrix $\mathbf{W}$ and $\mathbf{H}$. A Bayesian NMF model with the prior is then built. After maximizing the posterior, they obtain a multiplicative update rule to do both factorization and determination of component number simultaneously. The limitation of this method is that the prior distribution still depends on the hyper-parameters. For real-world applications, the hyper-parameters must be chosen suitably in advance to obtain reasonable results. In this sense, this method is not totally automatic for determining the component number.

In the following section, we overcome this problem and apply the ARD method to PNMF by selecting Jeffrey's prior [13] to get rid of hyper-parameters. Then our algorithm is totally automatic without any user-specified parameters.

3 ARDPNMF

Firstly, we construct a generative model for PNMF based on the Euclidean distance, where the likelihood function is a normal distribution.

$$p(X_{ij}|\mathbf{W}) = \mathcal{N}\left(X_{ij}|\left(\mathbf{W}\mathbf{W}^T\mathbf{X}\right)_{ij}, \mathbf{I}\right) \tag{5}$$

Following the approach of Bayesian PCA [10], we give a normal prior on the kth column of $\mathbf{W}$ with variance γ_k. Due to the nonnegativity in PNMF, we treat the distribution of each column of $\mathbf{W}$ as half-normal distribution.

$$p(W_{ik}|\gamma_k) = \mathcal{HN}(W_{ik}|0,\gamma_k) = \frac{\sqrt{2}}{\sqrt{\pi\gamma_k}} \exp\left(-\frac{W_{ik}^2}{2\gamma_k}\right) \tag{6}$$

for $W_{ik} \leq 0$, and zero otherwise.

Similar to [13], we impose a non-informative Jeffreys' hyper prior on the variances $\boldsymbol{\gamma}$ to control the sparseness of $\mathbf{W}$:

$$p(\gamma_k) \propto \frac{1}{\gamma_k} \tag{7}$$

We choose this prior because it expresses ignorance with respect to scale and the resulting model is parameter-free, which plays a significant role in determining the component number automatically.

The posterior of $\mathbf{W}$ for the above model is given by

$$p(\mathbf{W}|\mathbf{X},\boldsymbol{\gamma}) \propto p(\mathbf{X}|\mathbf{W})p(\mathbf{W}|\boldsymbol{\gamma}) \tag{8}$$

Because $\boldsymbol{\gamma}$ is unobserved, we apply the *Expectation-Maximization* (EM) algorithm by regarding $\boldsymbol{\gamma}$ as a hidden variable.

E-step. Given the current parameter estimates and observed data, E-step computes the expectation of the complete log-posterior, which is known as Q-function:

$$Q(\mathbf{W}|\mathbf{W}^{(t)}) = \int \log p(\mathbf{W}|\mathbf{X},\boldsymbol{\gamma}) p(\boldsymbol{\gamma}|\mathbf{W}^{(t)},\mathbf{X}) d\boldsymbol{\gamma} \tag{9}$$

Thanks to the property of Jeffrey's prior, we have a concise form of Q-function following the derivation in [13]:

$$Q(\mathbf{W}|\mathbf{W}^{(t)}) = -J_F(\mathbf{W}) - \frac{1}{2}\mathrm{Tr}(\mathbf{W}\mathbf{V}^{(t)}\mathbf{W}^T), \tag{10}$$

where $J_F(\mathbf{W})$ is the original objective function in PNMF (see Equation (2)), $\mathbf{V}^{(t)}$ is a diagonal matrix with $V_{ii}^{(t)} = \left\|\mathbf{w}_i^{(t)}\right\|^{-2}$, and $\left\|\mathbf{w}_i^{(t)}\right\|$ is the L_2-norm of the ith column of matrix $\mathbf{W}^{(t)}$. Note that we ignore the constants independent of $\mathbf{W}$ to present a simplified version of the Q-function.

M-step. This step maximizes the Q-function w.r.t parameters.

$$\mathbf{W}^{(t+1)} = \arg\max_{\mathbf{W}} Q(\mathbf{W}|\mathbf{W}^{(t)}), \tag{11}$$

which is equivalent to minimizing its negative form

$$Q_{\mathrm{ard}}(\mathbf{W}|\mathbf{W}^{(t)}) = -Q(\mathbf{W}|\mathbf{W}^{(t)}) = J_F(\mathbf{W}) + \frac{1}{2}\mathrm{Tr}(\mathbf{W}\mathbf{V}^{(t)}\mathbf{W}^T). \tag{12}$$

The derivative of $Q_{\mathrm{ard}}(\mathbf{W}|\mathbf{W}^{(t)})$ with respect to $\mathbf{W}$ is

$$\frac{\partial Q_{\mathrm{ARD}}(\mathbf{W}|\mathbf{W}^{(t)})}{\partial \mathbf{W}_{ik}} = -A_{ik} + B_{ik} + \left(\mathbf{W}\mathbf{V}^{(t)}\right)_{ik}. \tag{13}$$

For $\mathbf{A}, \mathbf{B}$, see eq. (3).

A commonly used principle that forms multiplicative update rule in NMF is

$$W'_{ik} \leftarrow W_{ik}^{(t)} \frac{\nabla_{ik}^-}{\nabla_{ik}^+}, \tag{14}$$

where ∇^- and ∇^+ denote the negative and positive parts of the derivative [1]. Applying this principle to the gradient given in Equation (13), we obtain the multiplicative update rule for ARDPNMF:

$$W'_{ik} \leftarrow W_{ik}^{(t)} \frac{A_{ik}^{(t)}}{B_{ik}^{(t)} + \left(\mathbf{W}^{(t)}\mathbf{V}^{(t)}\right)_{ik}}. \tag{15}$$

The ARDPNMF algorithm is summarized in Algorithm 1. After the algorithm converges, we apply a simple thresholding to keep the $\mathbf{W}$ columns whose norm is larger than a small constant ϵ. In practice such thresholding is insensitive to ϵ because the ARD prior forces these norms towards two extremes, as demonstrated in Section 4.1.

4 Experimental Results

We have implemented the ARDPNMF algorithm and tested it on various synthetic and real-world datasets to find out the effectiveness of our algorithm. The focus is on feature extraction and clustering.

Algorithm 1 ARDPNMF based on Euclidean distance

Usage: $\mathbf{W} \leftarrow \text{ARDPNMF}(\mathbf{X}, r)$, where $r < m$ is a large initial component number.
Initialize $\mathbf{W}^{(0)}$, $t \leftarrow 0$.
repeat
$\mathbf{V}^{(t)} \leftarrow \text{diag}(\|\mathbf{w}_1^{(t)}\|^{-2}, \ldots, \|\mathbf{w}_r^{(t)}\|^{-2})$
$W'_{ik} \leftarrow W_{ik}^{(t)} \frac{A_{ik}^{(t)}}{B_{ik}^{(t)} + (\mathbf{W}^{(t)}\mathbf{V}^{(t)})_{ik}}$
$\mathbf{W}^{(t+1)} \leftarrow \mathbf{W}'/\|\mathbf{W}'\|$
$t \leftarrow t + 1$
until convergent conditions are satisfied
Check the diagonal elements in matrix $\mathbf{V}$, and keep the columns of $\mathbf{W}$ with large L_2-norms as the effective components.

Fig. 1. Some sample images of Swimmer dataset

4.1 Swimmer Dataset

Swimmer dataset [14] consists of 256 images, each of which depicts a figure with one static part (torso)and four moving parts (limbs) with size 32×32. Each moving part has four different positions. Four of the 256 images are displayed in Figure 1. The task here is to extract the 16 limb positions and 1 torso position. Firstly, we vectorized each image matrix and treated it as one column of input matrix $\mathbf{X}$. The initial component number was set to $r = 36$. Each column of $\mathbf{W}$ learned by ARDPNMF has the same dimensionality as the input column vectors and thus can be displayed as base images in Figure 2. We found that our algorithm can correctly extract all the 17 desired features. The L_2-norms of all the columns of $\mathbf{W}$ are shown in Figure 3. We can easily see that the L_2-norms of ineffective basis images are equal to zero or very close to zero. The three values between 0 and 1 correspond to three duplicates of the torsos.

4.2 FERET Faces Dataset

The FERET face dataset [15] for feature extraction consists of the inner part of 2409 faces with size of 32×32. We normalized the images via dividing the pixel values by their maximal value 255. In ARDPNMF, the initial component number was chosen as $r = 64$. Figure 4 shows the resulting base images, which demonstrates high sparseness in the factorizing matrix $\mathbf{W}$ and captures nearly all facial parts.

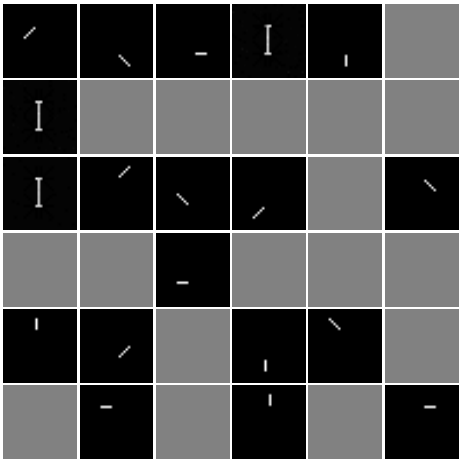

Fig. 2. 36 basis images of Swimmer dataset. The gray cells correspond to columns whose L_2-norms are zero or very close to zero.

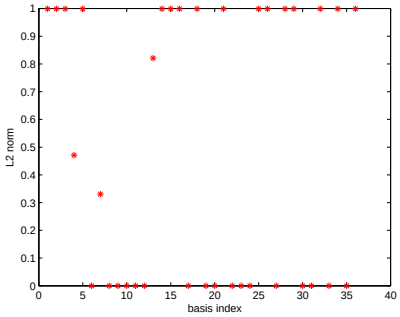

Fig. 3. L_2-norm of 36 basis images in Swimmer dataset.

4.3 Clustering for UCI Datasets

Clustering is another important application of PNMF. We construct the input matrix $\mathbf{X}$ by treating each sample vector as a row. Then the index of the maximal value in a row of $\mathbf{W}$ indicates the cluster membership of the corresponding sample [8]. We have adopted a widely-used measurement called *purity* [8] for quantitatively analyzing clustering results, which is defined as follows:

$$\text{purity} = \frac{1}{n} \sum_{k=1}^{r'} \max_{1 \le l \le q} n_k^l, \tag{16}$$

where q is the true number of classes, r' is the effective number of components (clusters), n_k^l is the number of samples in the cluster k that belongs to original

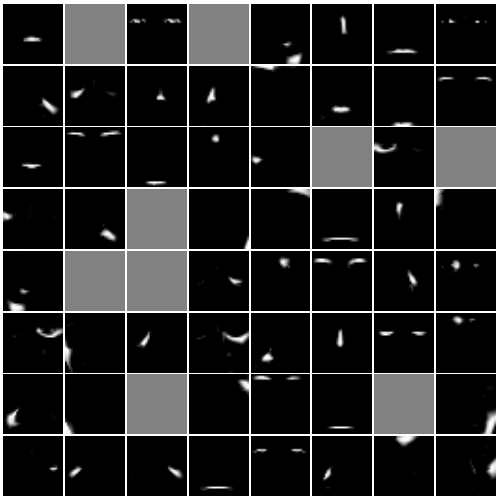

Fig. 4. 64 basis images of FERET dataset. 55 of them are effective basis. Remaining gray ones' L_2-norms are zero or close to zero.

Table 1. Clustering Performance

Datasets	iris	ecoli	glass	wine	parkinsons
Number of classes	3	5	6	3	2
Estimated cluster number	4.34 ± 0.71	2.74 ± 0.60	3.34 ± 0.61	3 ± 0.40	4.37 ± 0.58
Purity	0.95 ± 0.01	0.68 ± 0.06	0.67 ± 0.05	0.9 ± 0.09	0.77 ± 0.02

class l, and n is the total number of samples. Larger purity value indicates better clustering results, and value 1 indicates total agreement with the ground truth.

We chose several commonly used datasets in the UCI repository[1] as experimental data. In each dataset, ARDPNMF was run 100 times with different random seeds for $\mathbf{W}$ initialization, and we set the initial cluster number r as 36. Table 1 shows the mean and standard deviation of the number of clusters and purities, as well as the numbers of ground truth classes. ARDPNMF can automatically estimate the cluster number which is not far from the true class number, with small deviations. Furthermore, our method can achieve reasonably good clustering performance especially when the estimated r value is close to the ground truth.

5 Conclusion

In this paper, using Bayesian construction and EM algorithm, we have presented the ARDPNMF algorithm which can automatically determine the rank of the projection matrix in PNMF. By using Jeffreys' prior as the model prior, we

[1] http://www.ics.uci.edu/~mlearn/MLRepository.html

have made our algorithm totally free of human tuning in finding algorithm parameters. Through experiments on various synthetic and real-world datasets for feature extraction and clustering, ARDPNMF demonstrates its effectiveness in model selection for PNMF. Moreover, our algorithm is readily extended to other dissimilarity measures, such as the α or β divergences [2]. Our method however could be sensitive to the initialization of the factorizing matrix in some cases, which we should improve in the future for a more robust estimate of the rank.

References

1. Lee, D.D., Seung, H.S.: Learning the parts of objects by non-negative matrix factorization. Nature **401** (1999) 788–791
2. Cichocki, A., Zdunek, R., Phan, A.H., Amari, S.: Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-way Data Analysis. John Wiley (2009)
3. Dhillon, I.S., Sra, S.: Generalized nonnegative matrix approximations with bregman divergences. In: Advances in Neural Information Processing Systems. Volume 18. (2006) 283–290
4. Choi, S.: Algorithms for orthogonal nonnegative matrix factorization. In: Proceedings of IEEE International Joint Conference on Neural Networks. (2008) 1828–1832
5. Ding, C., Li, T., Jordan, M.I.: Convex and semi-nonnegative matrix factorizations. IEEE Transactions on Pattern Analysis and Machine Intelligence **32**(1) (2010) 45–55
6. Yuan, Z., Oja, E.: Projective nonnegative matrix factorization for image compression and feature extraction. In: Proc. of 14th Scandinavian Conference on Image Analysis (SCIA 2005), Joensuu, Finland (June 2005) 333–342
7. Yang, Z., Yuan, Z., Laaksonen, J.: Projective non-negative matrix factorization with applications to facial image processing. International Journal on Pattern Recognition and Artificial Intelligence **21**(8) (2007) 1353–1362
8. Yang, Z., Oja, E.: Linear and nonlinear projective nonnegative matrix factorization. IEEE Transaction on Neural Networks (2010) In press.
9. Mackay, D.J.C.: Probable networks and plausible predictions – a review of practical bayesian methods for supervised neural networks. Network: Computation in Neural Systems **6**(3) (1995) 469–505
10. Bishop, C.M.: Bayesian pca. In: Advances in Neural Information Processing Systems. (1999) 382–388
11. Tipping, M.E.: Sparse bayesian learning and the relevance vector machine. Journal of Machine Learning Research **1** (2001) 211–244
12. Tan, V.Y.F., Févotte, C.: Automatic relevance determination in nonnegative matrix factorization. In: Proceedings of 2009 Workshop on Signal Processing with Adaptive Sparse Structured Representations (SPARS'09. (2009)
13. Figueiredo, M.A.: Adaptive sparseness for supervised learning. IEEE Transactions on Pattern Analysis and Machine Intelligence **25**(9) (2003) 1150–1159
14. Donoho, D., Stodden, V.: When does non-negative matrix factorization give a correct decomposition into parts? In: Advances in Neural Information Processing Systems 16. (2003) 1141–1148
15. Phillips, P.J., Moon, H., Rizvi, S.A., Rauss, P.J.: The FERET evaluation methodology for face recognition algorithms. IEEE Trans. Pattern Analysis and Machine Intelligence **22** (October 2000) 1090–1104