Non-negative Matrix Factorization on Kernels

Daoqiang Zhang[1, 2], Zhi-Hua Zhou[2], and Songcan Chen[1]

[1] Department of Computer Science and Engineering
Nanjing University of Aeronautics and Astronautics,
Nanjing 210016, China
{dqzhang, s.chen}@nuaa.edu.cn
[2] National Laboratory for Novel Software Technology
Nanjing University, Nanjing 210093, China
zhouzh@nju.edu.cn

Abstract. In this paper, we extend the original non-negative matrix factorization (NMF) to kernel NMF (KNMF). The advantages of KNMF over NMF are: 1) it could extract more useful features hidden in the original data through some kernel-induced nonlinear mappings; 2) it can deal with data where only relationships (similarities or dissimilarities) between objects are known; 3) it can process data with negative values by using some specific kernel functions (e.g. Gaussian). Thus, KNMF is more general than NMF. To further improve the performance of KNMF, we also propose the SpKNMF, which performs KNMF on sub-patterns of the original data. The effectiveness of the proposed algorithms is validated by extensive experiments on UCI datasets and the FERET face database.

1 Introduction

Many data analysis tasks in machine learning require a suitable representation of the data. Typically, a useful representation can make the latent structure in the data more explicit, and often reduces the dimensionality of the data so that further computational methods can be applied [6]. Non-negative matrix factorization (NMF) [7] [8] is a recent method for finding such representation. NMF imposes the non-negativity constraints in its bases and coefficients. Due to its part-based representation property, NMF and its variations have been applied to image classification [2] [5], face expression recognition [3], face and object recognition [9] [10] [12], document clustering [13], etc.

However, NMF and many of its variants are essentially linear, and thus cannot disclose nonlinear structures hidden in the data. Besides, they can only deal with data with attribute values, while in many applications we do not know the detailed attribute values and only the relationships (similarities or dissimilarities) are available. NMF cannot be directly applied to such relational data. Furthermore, one requirement of NMF is that the values of data should be non-negative, while in many real-world problems the non-negative constraints can not be satisfied.

In this paper, we propose the kernel NMF (KNMF), which can overcome the above limitations of NMF. First, through using kernel-induced nonlinear mapping, KNMF could extract more useful features hidden in the original data. Second, we develop a method for KNMF to deal with data where only relationships between objects are known. Third, by using some specific kernel functions (e.g. Gaussian), KNMF can process data with negative values. Thus, KNMF is more general than NMF. Moreover, inspired by successes of so many 2D pattern representation methods [4] [14] and to further improve the performance of KNMF, we also propose the SpKNMF, which performs KNMF on sub-patterns of the original data. The effectiveness of the proposed algorithms is validated by extensive experiments on several UCI datasets and the FERET database for face recognition.

The rest of the paper is organized as follows: Section 2 introduces NMF briefly. This is followed by the detailed description of the KNMF algorithm in Section 3. In Section 4, we present the SpKNMF algorithm. In Section 5, experimental results are reported. Finally, we conclude this paper and raise some issues for future research in Section 6.

2 Non-negative Matrix Factorization

The key ingredient of NMF is the non-negativity constraints imposed on matrix factors. Assume that the observed data of the objects are represented as an $n \times m$ matrix ***V***, each column of which contains n non-negative attribute values of one of the m objects. In order to represent the data or reduce the dimensionality, NMF finds two non-negative matrix factors ***W*** and ***H*** such that

$$V_{i\mu} \approx (WH)_{i\mu} = \sum_{a=1}^{r} W_{ia} H_{a\mu} \tag{1}$$

Here the r columns of ***W*** are called NMF bases, and the columns of ***H*** are its combining coefficients. The dimensions of ***W*** and ***H*** are $n \times r$ and $r \times m$, respectively. The rank r of the factorization is usually chosen such that $(n+m)r<nm$, and hence the dimensionality reduction is achieved.

To find an approximate factorization $\boldsymbol{V} \approx \boldsymbol{W}\,\boldsymbol{H}$, a cost function is needed to quantify the quality of the approximation. NMF uses the divergence measure as the objective function

$$D(V \,||\, WH) = \sum_{i,j} \left(V_{ij} \log \frac{V_{ij}}{(WH)_{ij}} - V_{ij} + (WH)_{ij} \right) \tag{2}$$

NMF factorization is a solution to the following optimization problem: minimize $D(V \,||\, WH)$ with respect to ***W*** and ***H***, subject to the constraints $\boldsymbol{W}, \boldsymbol{H} \geq 0$, i.e. all terms in the matrix are non-negative. In order to obtain ***W*** and ***H***, a multiplicative update rule is given in [11] as follows

$$W_{ia} = W_{ia} \sum_{\mu=1}^{m} \frac{V_{i\mu}}{(WH)_{i\mu}} H_{a\mu} \tag{3a}$$

$$W_{ia} = \frac{W_{ia}}{\sum_{j=1}^{n} W_{ja}} \tag{3b}$$

$$H_{a\mu} = H_{a\mu} \sum_{i=1}^{n} W_{ia} \frac{V_{i\mu}}{(WH)_{i\mu}} \tag{3c}$$

3 Kernel Non-negative Matrix Factorization

Given *m* objects $\boldsymbol{O}_1$, $\boldsymbol{O}_2$, …, $\boldsymbol{O}_m$, with attribute values represented as an *n* by *m* matrix $\boldsymbol{V}$=[$\boldsymbol{v}_1$, $\boldsymbol{v}_2$, …, $\boldsymbol{v}_m$], each column of which represent one of the *m* objects. Define the nonlinear map from original input space $\boldsymbol{V}$ to a higher or infinite dimensional feature space $\boldsymbol{F}$ as follows

$$\phi : x \in V \rightarrow \phi(x) \in F \tag{4}$$

For the *m* objects, denote

$$\phi(V) = [\phi(v_1), \phi(v_2), ..., \phi(v_m)] \tag{5}$$

Similar as NMF, KNMF finds two non-negative matrix factors W_ϕ and $\boldsymbol{H}$ such that

$$\phi(V) = W_\phi H \tag{6}$$

Here, W_ϕ is the bases in feature space and $\boldsymbol{H}$ is its combining coefficients, each column of which denotes now the dimension-reduced representation for the corresponding object. It is worth noting that since $\phi(V)$ is unknown, it is impractical to directly factorize $\phi(V)$. In what follows, we will derive a practical method to solve this problem. From Eq. (6), we obtain

$$(\phi(V))^T \phi(V) = (\phi(V))^T W_\phi H \tag{7}$$

Before further explaining the meaning of Eq. (8), we first give the definition of kernels. According to [15], a kernel is a function in the input space and at the same time is the inner product in the feature space through the kernel-induced nonlinear mapping. More specifically, a kernel is defined as

$$k\left(x,y\right)=\left\langle \phi(x),\phi(y)\right\rangle =\left(\phi(x)\right)^{T}\phi(y) \tag{8}$$

Some commonly-used kernels in literature are [15]:
(1) Gaussian kernel

$$k(x,y)=\exp\left(\frac{-\|x-y\|^{2}}{\sigma^{2}}\right) \tag{9}$$

(2) Polynomial kernel

$$k(x,y)=\left(1+\left\langle x,y\right\rangle\right)^{d} \tag{10}$$

(3) Sigmoid kernel

$$K(x,y)=\tanh\left(\alpha\left\langle x,y\right\rangle+\beta\right) \tag{11}$$

From Eq. (8), the left side of Eq. (7) can be rewritten as

$$\left(\phi\left(V\right)\right)^{T}\phi\left(V\right)=\left\{\left(\phi(v_{i})\right)^{T}\phi(v_{j})\right\}_{i,j=1}^{m}=\left\{k(v_{i},v_{j})\right\}_{i,j=1}^{m}\triangleq K \tag{12}$$

Denote

$$Y=\left(\phi\left(V\right)\right)^{T}W_{\phi} \tag{13}$$

From Eqs. (12) and (13), Eq. (7) changes to

$$K=YH \tag{14}$$

Comparing Eq. (14) with Eq. (6), it can be found that the combining coefficient ***H*** is the same. Since W_{ϕ} is the learned bases of $\phi\left(V\right)$, similarly we call ***Y*** in Eq. (14) as the bases of the kernel matrix ***K***. Eq.(14) provides a practical way for obtaining the dimension-reduced representation ***H*** by performing NMF on kernels.

For a new data point, the dimension-reduced representation is computed as follows

$$\begin{aligned} H_{new} &=\left(W_{\phi}\right)^{+}\phi\left(v_{new}\right) \\ &=\left(W_{\phi}\right)^{+}\left(\left(\phi\left(V\right)\right)^{T}\right)^{+}\left(\phi\left(V\right)\right)^{T}\phi\left(v_{new}\right) \\ &=Y^{+}K_{new} \end{aligned} \tag{15}$$

Here A^{+} denote the generalized (Moore-Penrose) inverse of matrix A, and $K_{new} = \left(\phi(V)\right)^{T} \phi\left(v_{new}\right)$ is the kernel matrix between the *m* training instances and the new instance.

Eqs. (14) and (15) construct the key components of KNMF when used for classification. it is easy to see that, the computing of KNMF need not know the attribute values of objects, and only the kernel matrix ***K*** and ***K***$_{new}$ are required. Note that some kernels (e.g. Gaussian) can be seen as similarity measures between objects. Thus, the classification problem which KNMF can deal with is formulated as:

Given *m* training objects, we do not know their detailed attribute values, but the pair wise relationship between them can be measured (recorded in ***K***). Also, the attribute values of the test object is not known, but the relationship between it and the training objects can be computed (recorded in ***K***$_{new}$). Then, classify the new object into one of the training objects given ***K*** and ***K***$_{new}$.

Obviously, KNMF is more general than NMF because the former can deal with not only attribute-value data but also relational data. Another advantage of KNMF is that it is applicable to data with negative values since the kernel matrix in KNMF is always non-negative for some specific kernels (e.g. Gaussian).

4 Sub-pattern based KNMF

Given *m* objects ***O***$_1$, ***O***$_2$, …, ***O***$_m$, with attribute values represented by an *n* by *m* matrix ***V***=[***v***$_1$, ***v***$_2$, …, ***v***$_m$], each column of which represents one of the *m* objects. Assume *n* is divisible by *p*, then reassemble the original matrix ***V*** into *n*/*p* by *mp* matrix ***U*** as follows

$$U = \left[u_1, \ldots, u_p, u_{p+1}, \ldots, u_{2p}, \ldots, u_{(m-1)p+1}, \ldots, u_{mp}\right] \tag{16}$$

Here

$$v_i = \left[u_{(i-1)p+1}^{T}, \ldots, u_{ip}^{T}\right]^{T}, i = 1, 2, \ldots, m \tag{17}$$

From Eq. (16), compute the *mp* by *mp* kernel matrix as

$$K = \left(\phi(U)\right)^{T} \phi(U) \tag{18}$$

Factorizing Eq. (18) using Eq. (14), we obtain the dimension-reduced representation ***H***={***h***$_j$} with dimension of *r* by *mp*, where *r* is the number of reduced dimensions. Then reassemble the matrix ***H*** into *rp* by *m* matrix ***R*** as

$$R = \left\{r_i\right\}_{i=1}^{m} = \left\{\left[h_{(i-1)p+1}^{T}, \ldots, h_{ip}^{T}\right]^{T}\right\}_{i=1}^{m} \tag{19}$$

Similarly, for some new data v_{new}, first reassemble it into *n*/*p* by *p* matrix U_{new} as follows

$$U_{new} = \left[u_1, u_2, \ldots, u_p\right] \tag{20}$$

Here $v_{new} = \left[u_1^T, \ldots, u_p^T\right]^T$. From Eq. (20), compute the *mp* by *p* kernel matrix as

$$K_{new} = \left(\phi\left(U\right)\right)^T \phi\left(U_{new}\right) \tag{21}$$

From Eq. (15), we can obtain the dimension-reduced representation $H_{new} = [h_1, h_2, \ldots, h_p]$ with dimension of *r* by *p*, where *r* is the number of reduced dimensions. Then reassemble the matrix H_{new} into *rp* by 1 vector R_{new} as

$$R_{new} = \left[h_1^T, \ldots, h_p^T\right]^T \tag{22}$$

Finally, Eqs. (19) and (22) can be used for classification. For example, if the nearest neighborhood classifier is adopted, then classify the new data point to the same class of *i*-th column vector of R with minimum distance from R_{new}.

5 Experiments

In this section, we present a set of experiments to evaluate our proposed algorithms: KNMF and SpKNMF, compared with traditional NMF, on several UCI Machine Learning Repository datasets [1] and the FERET face database [11]. In our experiments Gaussian kernel is adopted and the kernel width is set to the standard variance $\sigma = \text{sqrt}(\sum_{j=1}^{n} \left\| x_j - \bar{x} \right\|^2 \big/ n)$. The nearest neighborhood classifier (1-NN) is used for classification. It is worthy noting that NMF, KNMF and SpKNMF are unsupervised dimensionality methods, and hence it is not comparable between them and some supervised dimensionality techniques or supervised classifiers.

5.1 UCI Data Sets

Four UCI datasets are used. No extra criterion is adopted for the selection of datasets except that the datasets should have relatively more dimensions and only numeric attributes without missing values are considered. Table 1 gives the statistics of them. For each dataset 10 independent runs are carried out and the results are averaged. At each run, half of the data are randomly picked for training, and the rest for testing. For each data set, we test the accuracies of NMF, KNMF and SpKNMF under different dimensions.

Table 1. Statistics of the UCI data sets

Dataset	Size	Dimension	# of classes
Ionosphere	351	34	2
Bupa	345	6	2
Glass	214	9	6
PID	768	8	2

Fig. 1 depicts the accuracies of NMF, KNMF and SpKNMF under different dimensions. It can be found that on all these data sets, SpKNMF consistently outperforms KNMF and NMF no matter which dimension is considered. For *Ionosphere* and *Glass*, KNMF outperforms NMF greatly. While for *Bupa* and *PID*, the performances of KNMF and NMF are close, while KNMF is slightly better under bigger dimensions.

Table 2 shows the accuracies averaged acrossthe dimensions shown in Fig. 1. From Table 2 it can be found that SpKNMF and KNMF outperform NMF on average, and SpKNMF always achieves the beest performance (see the bold).

Table 2. Comparisons of averaged accuracies (%) under different dimensions (the values in the bracket are the sizes of the reassembled matrices).

Datasets	NMF	KNMF	SpKNMF
Ionosphere	75.24	91.24	**92.69**(17x2)
Bupa	55.58	56.79	**60.19**(3x2)
Glass	39.87	44.89	**51.88**(3x3)
PID	64.52	64.75	**69.51**(4x2)

5.2 FERET Face Database

In this experiment, a partial FERET face database containing 400 gray-level frontal view face images from 200 persons are used, each of which is cropped with the size of 60×60. There are 71 females and 129 males; each person has two images (**fa** and **fb**) with different facial expressions. The **fa** images are used as gallery for training while the **fb** images as probes for test.

Fig. 2 shows the accuracies of the three algorithms under different feature dimensions on the partial FERET face database. From Fig2 it can be found that SpKNMF and KNMF consistently outperform NMF no matter how many dimensions are used. SpKNMF is slightly superior to KNMF on this database. The accuracies of NMF, KNMF and SpKNMF averaged across all the dimensions shown in Fig. 2 are 69.23, 80.37 and 84.44, respectively. The performance of SpKNMF is still the best (the size of its reassembled matrix is 900x4). it is impressive that SpKNMF and KNMF achieve nearly 10% and 15% higher accuracies than NMF, respectively.

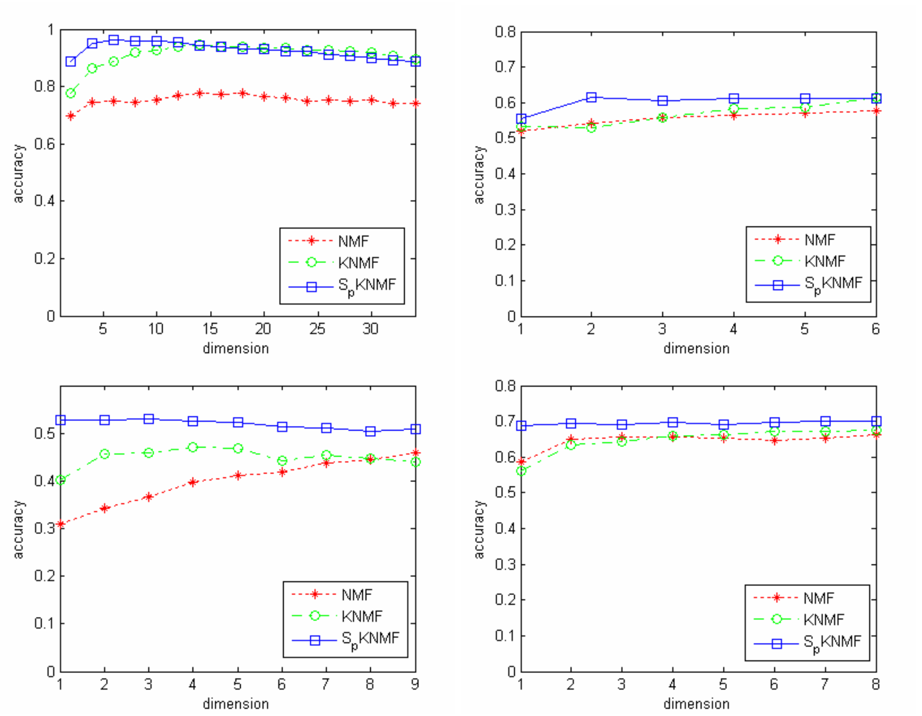

Fig. 1. Comparisons of accuracies under different dimensions on *Ionosphere* (top left), *Bupa* (top right), *Glass* (bottom left) and *PID* (bottom right).

6 Conclusions

In this paper, KNMF is developed. Compared with conventional NMF, KNMF can: 1) extract more useful features hide in the original data using some kernel-induced nonlinear mapping; 2) deal with relational data where only the relationships between objects are known; 3) process data with negative values by using some specific kernel functions (e.g. Gaussian). Thus, KNMF is more general than NMF. Furthermore, another algorithm SpKNMF is proposed to further improve the performance of KNMF by performing KNMF on sub-patterns of the original data. Experimental results on UCI datasets and the FERET face database validated the effectiveness of the proposed algorithms.

There are several issues for future research. First, as in other kernel-based methods, the selection of kernels and their parameters is crucial for the performances of KNMF and SpKNMF. In this paper, we only consider the Gaussian kernel and set the kernel width to the standard variance. We will investigate how to adaptively choose the kernels and parameters in the future. Also, choosing the appropriate size for the reassembled matrix in SpKNMF is also an interesting issue for future research. Moreover, comparing KNMF and SpKNMF with other dimensionality reduction methods, such

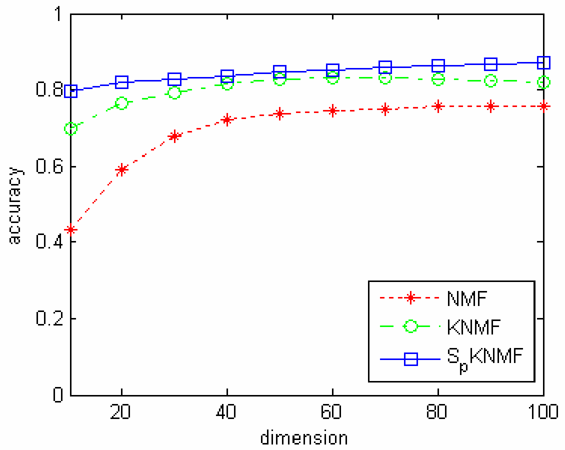

Fig. 2. Comparisons of accuracies under different dimensions on the partial FERET database.

as LDA and KFD, in particular, on their performances in classification, is left for future research.

Acknowledgements

This work was supported by NSFC (Nos. 60505004 and 60473035), JiangsuSF (BK2005122), FANEDD (200343), the Jiangsu Postdoctoral Research Foundation, and funds from Shanghai Key Laboratory of Intelligent Information Processing. Portions of the research in this paper use the FERET database of facial images collected under the FERET program.

References

1. Blake, C., Keogh, E., Merz, C.J.: UCI repository of machine learning databases [http://www.ics.uci.edu/~mlearn/MLRepository.html], Department of Information and Computer Science, University of California, Irvine, CA, 1998
2. Buchsbaum, G., Bloch, O.: Color categories revealed by non-negative matrix factorization of Munsell color spectra. Vision Research 42 (2002) 559-563
3. Buciu, I., Pitas,I.: Application of non-negative and local non-negative matrix factorization to facial expression recognition. In: ICPR, Cambridge, 2004
4. Chen, S.C., Zhu, Y.L.: Subpattern-based principle component analysis. Pattern Recognition 37 (5) 1081-1083
5. Guillamet, D., Bressan, M., Vitria, J.: A weighted non-negative matrix factorization for local representation. In: CVPR, Hawaii, 2001
6. Hoyer, P.O.: Non-negative matrix factorization with sparseness constraints. Journal of machine Learning Research 5 (2004) 1457-1469

7. Lee D.D., Seung, H.S.: Learning the parts of objects by non-negative matrix factorization. Nature 401(1999) 788-791

8. Lee D.D., Seung, H.S.: Algorithms for non-negative matrix factorization. In: NIPS 13 (2001) 556–562

9. Li, S.Z., Hou, X.W., Zhang, H.J., Cheng, Q.S.: Learning spatially localized, parts-based representation. In: CVPR, Hawaii, 2001.

10. Liu, W., Zheng, N.: Non-negative matrix factorization based methods for object recognition. Pattern Recognition Letters 25 (2004) 893-897

11. Phillips, P.J., Wechsler, H., Huang, J., Rauss, P.J.: The FERET database and evaluation procedure for face-recognition algorithms. Image and Vision Computing 16 (5) (1998) 295-306

12. Wild, S., Curry, J., Dougherty, A.: Improving non-negative matrix factorizations through structured initialization. Pattern Recognition 37 (11) (2004) 2217-2232

13. Xu, W., Liu, X., Gong, Y. Document clustering based on non-negative matrix factorization. In: SIGIR'03, Toronto, Canada, 2003

14. Zhang, D.Q., Chen, S.C., Liu, J.: Representing image matrices: Eigenimages vs. Eigenvectors. In: ISNN, Chongqing, China, 2005

15. Zhang, D.Q., Chen, S.C.: Clustering incomplete data using kernel-based fuzzy c-means algorithm. Neural Processing Letters 18(3) (2003) 155-162