NON-NEGATIVE MATRIX FACTORIZATION FOR VISUAL CODING

Weixiang Liu *Nanning Zheng* *Xiaofeng Lu*

Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, Xi'an 710049, P.R. China
{wxliu, xflu}@aiar.xjtu.edu.cn nnzheng@mail.xjtu.edu.cn

ABSTRACT

This paper combines linear sparse coding and non-negative matrix factorization into sparse non-negative matrix factorization. In contrast to non-negative matrix factorization, the new model can learn much sparser representation via imposing sparseness constraints explicitly; in contrast to a close model - non-negative sparse coding, the new model can learn parts-based representation via fully multiplicative updates because of adapting a generalized Kullback-Leibler divergence instead of the conventional mean square error for approximation error. Experiments on MIT-CBCL training faces data demonstrate the effectiveness of the proposed method.

1. INTRODUCTION

One primary goal of visual neuroscience is to understand processing of early visual system. Many experiments have concerned the mammalian primary visual cortex known as V1. Neurons in V1 are divided into two families: simple and complex cells. Linear sparse coding is an influential model to receptive fields of simple cells in V1 [9,4].
Parts-based representation by non-negative matrix factorization (NMF) [1] is proposed for visual sensory coding. In this model, non-negative training data are arranged as a matrix, and the same non-negative constraints are composed to its matrix factors. It has been widely applied to pattern recognition [3] such as image representation, document analysis [1] although it is claimed that non-negativity is not reasonable [7]. Furthermore, L. K. Saul and D. D. Lee's work shows that sparseness of training data controls the speed of learning [10]. There the sparseness of the training data is measured by the $1/s$, where the value of s is the maximum sum of features. In fact, although the non-negativity yields sparseness to some extent, much sparser representation can be leaned.
Recently, P. O. Hoyer combined non-negative matrix factorization and sparse coding into *non-negative sparse coding* (NNSC) [4]. We found that this NNSC model can only work well in simplex case, such as the binary bar images shown in the experiments of [4]. Adapting the projected gradient descent, it has the disadvantage of being sensitive to the choice of iterative step size which can only be set experientially. In addition, it is not guaranteed to make every element of factor matrices non-negative. Then it resorts to one 'forcible' step to set negative elements to zeros for non-negative constraints. This is not fully multiplicative update rule as that in NMF. In this paper we simply combine sparse coding and non-negative matrix factorization into *sparse non-negative matrix factorization* (SNMF) which can learn both parts-based representation and much sparser representation.
The rest of this paper is organized as follows. In section 2 we recall some basic theory of sparse coding, NMF and NNSC and discuss their relationships. In section 3 we introduce the proposed method - SNMF. We made some experiments on MIT-CBCL training faces data using NMF, NNSC, and SNMF for contrast in Section 4 and in Section 5 we give our conclusion.

2. SPARSE CODING, NMF AND NNSC

2.1. SPARSE CODING

The idea of linear sparse coding model is as follows: The observed data is a linear superposition of some basis vectors (called basis images in the case of image data), and the coefficients based on these basis vectors (images) stand for new feature vectors for further analysis. The sparse coding strategy assumes that these hidden variables exhibit sparseness. Then it maximizes sparseness via seeking a specific form of low-entropy code in which the probability densities are highly peaked around zero and have heavily long tails [9,4].
As noted in [9], let $I(x, y)$ denote an image, which is assumed as the following linear superposition of a set of basis functions $f_i(x, y)$

$$I(x,y) = \sum_i a_i f_i(x,y) \tag{1}$$

where a_i is the coefficients. As stated above, the linear sparse coding can be modeled as an optimization problem by minimizing the following cost function [9]:

$$F = -\text{ [preserve information] } - \lambda \text{ [sparseness of } a_i] \tag{2}$$

where the positive value of λ controls the preserve information, usually measured by the approximation error between the actual data and the reconstructed data, and the sparseness.

2.2. NMF

NMF is a linear multivariate analysis method in the following manner [2]. Let **X** denote an $n \times m$ matrix, each column of which contains an n-dimension observed data vector with non-negative values. In order to compress data or reduce dimensionality, we can find two non-negative matrix factors **B** and **H** such that

$$\mathbf{X} \approx \mathbf{BH} \tag{3}$$

where **B** is an $n \times r$ matrix and **H** an $r \times m$ matrix. The value of r is smaller than both n and m. We call the r columns of **B** basis images (vectors) and the columns of **H** encoding coefficients. We can consider the observed data matrix **X** as original features, and the encoding matrix **H** new features based on the basis matrix **B**.

In order to complete approximate factorization (3), we need to define some cost functions that quantify the quality of the approximation. Two algorithms for NMF are proposed in [2] using the simple and efficient multiplicative update rules. In [1, 5], NMF is carried out to solve the following optimization problem:

$$\begin{aligned} \min \quad & D(\mathbf{X}\|\mathbf{BH}) = \sum_{i,j}(x_{ij}\log(x_{ij}/y_{ij}) - x_{ij} + y_{ij}) \\ \text{s.t} \quad & \mathbf{B},\mathbf{H} \geq 0, \sum_i b_{ik} = 1 \end{aligned} \tag{4}$$

where y_{ij} = $[\mathbf{BH}]_{ij}$ for simplicity. In fact, this objective function is also called as generalized Kullback-Leibler divergence [2].

According to work in [10], the sparseness of the learned feature vectors for further analysis, which controls the speed of learning algorithm, is measured by $1/\max_j(\sum_k h_{kj})$ where h_{kj} are learned by NMF as above.

This is one criteria for the following experiment performance.

It is claimed that the exact form of the objective function is not crucial in [1] but we can find here, that different objective function or cost function will provide different factorizations.

2.3. NNSC

Inspired by linear sparse coding [9], P. O. Hoyer suggests that it is important to impose both the non-negativity and the sparseness for leaning parts-based representations [4]. He used the mean square function of error as the loss

$$D(\mathbf{X} \| \mathbf{BH}) = \|\mathbf{X}\text{-}\mathbf{BH}\|^2/2. \tag{5}$$

Then his NNSC model of a non-negative data matrix can be defined as the following optimization problem [4]

$$\begin{aligned} \min \quad & C(\mathbf{B},\mathbf{H}) = \|\mathbf{X}\text{-}\mathbf{BH}\|^2/2 + \lambda\sum_{k,j} h_{kj} \\ \text{s.t} \quad & \mathbf{B},\mathbf{H} \geq 0, \|b_{\cdot k}\| = 1, \forall k . \end{aligned} \tag{6}$$

There are two parameters in its learning algorithm, i.e. the positive constant λ and iterative step size for the projected gradient descent; see [4] for more details. In fact, the update rule of NNSC is not fully multiplicative and this leads to some disadvantages discussed above.

3. SPARSE NMF

Although NMF yields sparseness for its non-negativity to some extent, We believe that much sparser representation can be learned by imposing sparseness constraints on matrix factors. So simply from the viewpoint of linear sparse coding, we combine sparse coding and NMF to Spare NMF (SNMF). Here we select the generalized Kullback-Leibler divergence between **X** and **BH** to measure the approximation error.

Then we get our SNMF as formulated the following optimization problem

$$\begin{aligned} \min \quad & F(\mathbf{B},\mathbf{H}) = \sum_{i,j}(x_{ij}\log(x_{ij}/y_{ij}) - x_{ij} + y_{ij}) + \alpha\sum_{k,j} h_{kj} \\ \text{s.t} \quad & \mathbf{B},\mathbf{H} \geq 0, \sum_i b_{ik} = 1 . \end{aligned} \tag{7}$$

where y_{ij} = $[\mathbf{BH}]_{ij}$ for convenience as above, and α is similar to λ in (6). Now we can interpret our SNMF both from sparse coding and NMF. For the former, we select different approximation error in contrast to that of NNSC. For the latter, we impose constraint on encoding coefficients for sparseness explicitly via minimizing the sum of all h_{kj}. Then SNMF also is a variant of NMF.

The multiplicative update rules for constrained optimization (7) are as follows:

$$h_{kj} \leftarrow \frac{h_{kj}\sum_i x_{ij}\dfrac{b_{ik}}{\sum_l b_{il}h_{lj}}}{1+\alpha} \tag{8}$$

$$b_{ik} \leftarrow \frac{b_{ik}\sum_j x_{ij}\dfrac{h_{kj}}{\sum_l b_{il}h_{lj}}}{\sum_j h_{kj}} \tag{9}$$

$$b_{ik} \leftarrow \frac{b_{ik}}{\sum_i b_{ik}} \tag{10}$$

In contrast to NNSC this algorithm has fully multiplicative update rules with one parameter. Set α =0, then SNMF reduces to NMF. For our proof, now we must briefly mention another variant of NMF by S. Z. Li and co-authors. Their method, called local NMF (LNMF), imposes additional constraints both on basis vectors and coefficient besides non-negativity [5]. Here we focus on sparse coding, and the relationships among NMF, NNSC, SNMF and LNMF are discussed in other paper. The proof of SNMF algorithm given in Appendix is following that in [2, 5].

4. EXPERIMENTS

We investigated how SNMF can learn both sparse and parts representation in contrast to NMF and NNSC. We adapt the training set for faces of MIT-CBCL face database which was used in [1]. This training set contains 2429 faces. Each face has 19×19 pixels and has been histogram-equalized and normalized so that all pixel values are between 0 and 1 [6].

We compared NMF, NNSC, and our SNMF for learning the parts-based and representation with $r = 5 \times 5$, $7 \times 7, \ldots, 13 \times 13$. Set $\lambda = 100$ and step size equal to 1 for NNSC, and $\alpha = 100$ for SNMF. In fact, it is difficult to set step size for NNSC in the experiment. This is one shortcoming of NNSC because of projected gradient descent. For comparison, we set maximum of iteration steps equal to 200.

Firstly we investigate how SNMF can learn much sparser features than NMF according to the maximum sum of the feature vectors of training data. Table 1 shows the sparseness of the learned feature vectors by NMF, SNMF and NNSC, respectively. We can find that SNMF can improve the sparseness of feature vectors from 0.0054 of NMF to 0.5548, where the ratio of the latter to the former is close to α . Here we find that NNSC can learn the sparsest features, which can arrive to 1.1536e205.

In contrast to NMF, SNMF can learn much sparser features because additional sparseness constraint is imposed on encoding coefficient matrix explicitly.

Table 1 Sparseness of learned feature vectors

sqrt(r)	5	7	9	11	13
NMF	.0054	.0054	.0054	.0054	.0054
NNSC(1e205)	.0054	.0192	.0617	.1391	1.1536
SNMF	.5448	.5448	.5448	.5448	.5448

However we found that the learned basis images via NNSC doesn't has the property of 'parts-based' occurred both in the cases of NMF and SNMF as shown in Figure 2. As mentioned above, there is one 'forcible' step in the learning algorithm, i.e. setting negative elements to zero. This update rule is not fully multiplicative as that of NMF and SNMF. This implies that NNSC can't learn 'parts-based' representation in complex cases, such as faces, although it works well in simple case as reported in [4].

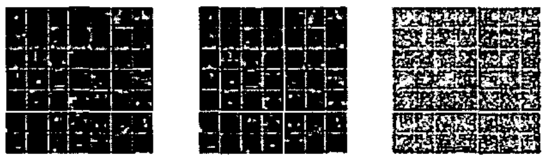

Figure 2 Basis images learned by NMF (left), SNMF (middle) and NNSC (right).

Now we can conclude that the proposed SNMF can learn both sparse and parts-based presentation in contrast to NMF and NNSC as in Table 2.

Table 2 Comparison of learned basis images and sparse features by NMF, NNSC and SNMF.

	Parts-based	Sparseness
NMF	yes	general
NNSC	no	most
SNMF	yes	more

5. CONCLUSION

We combine sparse coding and non-negative matrix factorization into sparse non-negative matrix. It is a new variant of linear spare coding model based on non-negative matrix. We compared three methods of non-negative matrix factorization on MIT-CBCL training faces data. The proposed SNMF method, simply from the viewpoint of sparse coding like NNSC, can learn not only much sparser representation than NMF, but also parts-based representation as NMF, in contrast to NNSC which can't learn parts-based representation in complex cases. SNMF overcomes the shortcomings of NNSC. It implies that different objective functions lead to different results. One future work is to use learned features via SNMF for further analysis such as classification and clustering and to investigate how the sparseness affects the accuracy. Another further work is to investigate other objective functions for matrix factorization and constraints on matrix factors.

APPENDIX

Because the objective function of SNMF is similar to that of NMF and LNMF, the proof will follow closely the proof given in [2,5], which makes use of an auxiliary function technique as the EM algorithm [8]. We can take turns updating **H** and **B** and get our learning algorithm.

Updating H: **H** is updated by minimizing L(**H**) = *F*(**B**, **H**) with **B** fixed. Following [2, 5], We define the auxiliary function for L(**H**) here as

$$G(\mathbf{H},\mathbf{H}') = \sum_{i,j} x_{ij}\log x_{ij} - \sum_{i,j,k} x_{ij}\frac{b_{ik}h'_{kj}}{\sum_l b_{il}h'_{lj}}\left[\log(b_{ik}h_{kj}) - \log\frac{b_{ik}h'_{kj}}{\sum_l b_{il}h'_{lj}}\right] - \sum_{i,j} x_{ij} + \sum_{i,j} y_{ij} + \alpha\sum_{k,j} h_{kj} \quad (11)$$

Firstly it is obviously to prove G(**H**,**H**) = L(**H**). Then we can prove G(**H**,**H'**) $\geq$ L(**H**). Because –log is a convex function, for any *k* non-negative numbers summing up to one ($\sum_k u_{ijk} = 1$), we get the following

$$-\log(\sum_k b_{ik}h_{kj}) \leq -\sum_k u_{ijk}\log(\frac{b_{ik}h_{kj}}{u_{ijk}}) \quad (12)$$

Let

$$u_{ijk} = \frac{b_{ik}h'_{kj}}{\sum_l b_{il}h'_{lj}} \quad (13)$$

then

$$-\log(\sum_k b_{ik}h_{kj}) \leq -\sum_k \frac{b_{ik}h'_{kj}}{\sum_l b_{il}h'_{lj}}\left[\log(b_{ik}h_{kj}) - \log\frac{b_{ik}h'_{kj}}{\sum_l b_{il}h'_{lj}}\right] \quad (14)$$

so G(**H**,**H'**) $\geq$ L(**H**) holds.

In order to minimize L(**H**), we can update **H** by following iterative rule

$$\mathbf{H}^{t+1} = \arg\min_H G(\mathbf{H},\mathbf{H}') \quad (15)$$

and such **H** can be found by $\frac{\partial G(\mathbf{H},\mathbf{H}')}{\partial h_{kj}} = 0$. Because

$$\frac{\partial G(\mathbf{H},\mathbf{H}')}{\partial h_{kj}} = -\sum_i x_{ij}\frac{b_{ik}h'_{kj}}{\sum_l b_{il}h'_{lj}}\cdot\frac{1}{h_{kj}} + \sum_i b_{ik} + \alpha = 0 \quad (16)$$

we get

$$h_{kj} = \frac{\sum_i x_{ij}\frac{b_{ik}h'_{kj}}{\sum_l b_{il}h'_{lj}}}{1+\alpha}. \quad (17)$$

For update rule, all variables h'_{kj} are the results of previous step. Then we get the rule (8).

Updating B: **B** is updated through minimizing L(**B**) = *F*(**B**, **H**) with **H** fixed. We here define the auxiliary function for L(**B**) as

$$G(\mathbf{B},\mathbf{B}') = \sum_{i,j} x_{ij}\log x_{ij} - \sum_{i,j,k} x_{ij}\frac{b'_{ik}h_{kj}}{\sum_l b'_{il}h_{lj}}\left[\log(b_{ik}h_{kj}) - \log\frac{b'_{ik}h_{kj}}{\sum_l b'_{il}h_{lj}}\right] - \sum_{i,j} x_{ij} + \sum_{i,j} y_{ij} + \alpha\sum_{k,j} h_{kj} \quad (18)$$

As above G(**B**,**B**) = L(**B**) and G(**B**,**B'**) $\geq$ L(**B**) hold. Let

$$\frac{\partial G(\mathbf{B},\mathbf{B}')}{\partial b_{ik}} = -\sum_j x_{ij}\frac{b'_{ik}h_{kj}}{\sum_l b'_{il}h_{lj}}\cdot\frac{1}{b_{ik}} + \sum_j h_{kj} = 0 \quad (19)$$

and set all variables b'_{ik} as the results of previous step, we can easily get the rule (9).

6. REFERENCES

[1] D.D. Lee and H.S. Seung, "Learning The Parts Of Objects By Non-Negative Matrix Factorization," *Nature*, vol. 401, pp. 788-791, 1999.

[2] D.D. Lee and H. S. Seung, "Algorithms For Non-Negative Matrix Factorization," in T. Leen, T. Dietterich, and V. Tresp (eds), Advances in Neural Information Processing Systems 13. MIT Press: Cambridge, MA, 2001.

[3] A. Jain, P. Duin, and J. Mao. "Statistical Pattern Recognition: A Review," IEEE Trans PAMI, vol. 22(1), pp. 4-37, 2000.

[4] P. O. Hoyer, "Non-negative Sparse Coding," To be presented at NNSP 2002. Available online, see http://www.cis.hut.fi/~phoyer/papers/.

[5] S.Z. Li, X.W. Hou, and H.J. Zhang, "Learning Spatially Localized Parts-Based Representation" Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, Hawaii, 2001.

[6] CBCL Face Database #1,MIT Center For Biological and Computation Learning. Available online, see http://www.ai.mit.edu/projects/cbcl.

[7] B.W. Mel, "Think Positive to Find Parts," *Nature*, vol. 401, pp. 759-760, 1999.

[8] A.P. Dempster, N.M. Laird, and D.B. Rubin. "Maximum Likelihood from Incomplete Data via the EM Algorithm," Roy. Statist. Soc., vol. 39, pp. 1-38, 1977.

[9] B. A. Olshausen and D. J. Field, "Emergence of Simple-Cell Receptive Field Properties by Learning A Sparse Code for Natural Images," *Nature*, vol. 381, pp. 607-609, 1996.

[10] L. K. Saul and D. D. Lee, "Multiplicative Updates for Classification by Mixture Models" in T. G. Dietterich, S. Becker, and Z. Ghahramani (eds.), Advances in Neural Information Processing Systems 14. MIT Press, Cambridge, MA, 2002.