Neural Process Lett (2016) 43:603–609
DOI 10.1007/s11063-015-9432-7

# Dispersion Constraint Based Non-negative Sparse Coding Model

**Xin Wang[1,2] · Can Wang[1] · Li Shang[3] · Zhan-Li Sun[4]**

Published online: 23 June 2015


**Abstract** Based on advantages of basic non-negative sparse coding (NNSC) model, and considered the prior class constraint of image features, a novel NNSC model is discussed here. In this NNSC model, the sparseness criteria is selected as a two-parameter density estimation model and the dispersion ratio of within-class and between-class is used as the class constraint. Utilizing this NNSC model, image features can be extracted successfully. Further, the feature recognition task by using different classifiers can be implemented well. Simulation results prove that our NNSC model proposed is indeed effective in extracting image features and recognition task in application.

**Keywords** Classification constraint · Non-negative sparse coding (NNSC) · Dispersion ratio · Classifiers · Within-class · Between-class

✉ Li Shang
sl0930@jssvc.edu.cn

Xin Wang
xinwang@zju.edu.cn

Can Wang
wcan@zju.edu.cn

Zhan-Li Sun
zhlsun2006@126.com

1 Zhejiang Provincial Key Laboratory of Service Robot, College of Computer Science, Zhejiang University, Hangzhou 310027, Zhejiang, China

2 Simon Fraser University, 8888 University Drive, Burnaby, BC V5A 1S6, Canada

3 Department of Communication Technology, College of Electronic Information Engineering, Suzhou Vocational University, Suzhou 215104, Jiangsu, China

4 School of Electrical Engineering and Automation, Anhui University, Hefei 230039, Anhui, China

## 1 Introduction

Non-negative sparse coding (NNSC) model was proposed by Hoyer in 2002 [1], which has been used widely in numerous domains, especially in image processing field [2]. And many documents published [1–3] have proved that NNSC, like as sparse coding (SC) [2], can successfully extract image features, denoise images, classify features and so on. However, Hoyer's model only considers image reconstruction error and sparse priori distribution of sparse coefficients in the cost function. And in learning feature basis vectors [3–6], only gradient projection method is used without considering multiplicative updating factor. Therefore, its performance is influenced hardly by the iterative step size, and the convergence precision can not be very high. Otherwise, the prior class information of image features isn't also considered in Hoyer's model, so, when being used to classification task, the high classification precision isn't also be obtained. To solve these faults above-mentationed, considered prior class constraint and the maximum sparseness, a novel NNSC model is proposed by us in this paper. In our NNSC model, the class constraint is selected the dispersion ratio of within-class and between-class of image features' sparse coefficients, in others words, our NNSC model is such a model based on dispersion constraint, denoted by DCB-NNSC here. In our DCB-NNSC model, the sparseness measure criterion is selected as the two-parameter density model behaving the priori sparse distribution of feature coefficients as described in document [6]. In the feature extraction task, to improve the feature separability, the dispersion ratio of within-class and between-class of feature coefficients is used. Further, using the PolyU palmprint database to test our DCB-NNSC model, simultaneously, compared with Hoyer's NNSC model in the same experimental condition, simulation results both testify that it is efficient indeed in image feature extraction and feature recognition task.

## 2 Dispersion Constraint

Assumed that $\mathbf{S} = \left[\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_j, \ldots, \mathbf{s}_M\right]$ denotes the sparse coefficient matrix, where $\mathbf{s}_j$ is the $jth$ column vector ($j = 1, 2, \ldots, M$). Let $M_k$ ($k = 1, 2, 3, \ldots, C$) denote the number of elements of the $kth$ class matrix $\mathbf{S_k}$, $\tilde{\mathbf{S}}$ denote the mean matrix of all samples, $\tilde{\mathbf{S}}_k$ denote the mean matrix of $\mathbf{S_k}$, and $\widehat{\mathbf{S}}_\mathbf{k}$ denote all the $k$th class samples. And then the within-class and between-class dispersion matrix $\mathbf{D}_\mathrm{W}$ and $\mathbf{D}_\mathrm{B}$ are respectively defined as follows [5]:

$$\mathbf{D}_\mathrm{W} = \frac{1}{M}\sum_{k=1}^{C}\sum_{\mathbf{S}_j \in \widehat{\mathbf{S}}_k}\left(\mathbf{S}_j - \tilde{\mathbf{S}}_k\right)^T\left(\mathbf{S}_j - \tilde{\mathbf{S}}_k\right) \tag{1}$$

$$\mathbf{D}_\mathrm{B} = \frac{1}{M}\sum_{k=1}^{C} M_k\left(\tilde{\mathbf{S}}_k - \tilde{\mathbf{S}}\right)^T\left(\tilde{\mathbf{S}}_k - \tilde{\mathbf{S}}\right) \tag{2}$$

where $\tilde{\mathbf{S}}_k = \frac{1}{M_k}\sum_{i=1}^{M_k}\mathbf{S}_i^{(k)}$ and $\tilde{\mathbf{S}} = \frac{1}{M}\sum_{i=1}^{M}\mathbf{S}_i$. After $\mathbf{D}_\mathrm{W}$ and $\mathbf{D}_\mathrm{B}$ are computed, the dispersion constraint is obtained by the following formula [5]:

$$Dis = \ln\left(\mathbf{D}_\mathrm{W}/\mathbf{D}_\mathrm{B}\right) \tag{3}$$

Noted that the smaller $\mathbf{D}_\mathrm{W}$ is, and the larger $\mathbf{D}_\mathrm{B}$ is, the smaller the $Dis$ is, thus, the better within-class aggregation can be obtained. So, the term of $\ln\left(\mathbf{D}_\mathrm{W}/\mathbf{D}_\mathrm{B}\right)$ is considered as the

class constraint in our NNSC model, which can make features trained to be better separability in implementing feature recognition task.

## 3 Our NNSC Model

### 3.1 Hoyer's NNSC Model

Hoyer's NNSC model combines two algorithms of SC and Non-negative Matrix Factorization (NMF) [4], and the cost function is formulated as follows [3,4]:

$$\min \quad J(\mathbf{A},\mathbf{S}) = \frac{1}{2}\|\mathbf{X}-\mathbf{AS}\|^2 + \lambda \sum_{ij} \mathbf{S}_{ij} \tag{4}$$

subject to $\mathbf{X} \geq 0$, $\mathbf{A} \geq 0$, $\mathbf{S} \geq 0$ and $\|\mathbf{A}\|_1 = 1$ in training. Here $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N]^T$ denotes the input matrix with the size of $N \times L$, and each row is an image patch. Matrix $\mathbf{A}$ is the feature basis set with $N \times M$ size, and matrix $\mathbf{S}$ with the size of $M \times L$ represents sparse coefficients. In training NNSC model, $\mathbf{A}$ and $\mathbf{S}$ are updated in turn by using the gradient optimization algorithm and a multiplicative method. The detailed learning process can be found in the document [3]. However, because of only using the gradient projection in updating $\mathbf{A}$, the property of Hoyer's model is influenced hardly by the size of step length, and its precision can not be high in practice.

### 3.2 The DCB-NNSC Model

On the basis of Hoyer's model, to ensure the sparsity of feature coefficients and improve the feature separability, considered the adaptive-self sparsity of data and class priori knowledge, a new NNSC model is proposed in this paper and its cost function is defined as follows:

$$\min \quad J(\mathbf{A},\mathbf{S}) = \frac{1}{2}\|\mathbf{X}-\mathbf{AS}\|^2 + \lambda_1 \sum_i f(\mathrm{s}_i) + \lambda_2 \ln(\mathbf{D}_\mathrm{W}/\mathbf{D}_\mathrm{B}) \tag{5}$$

where the constraint conditions is still subject to $\mathbf{X} \geq 0$, $\mathbf{A} \geq 0$, $\mathbf{S} \geq 0$ and $\|\mathbf{A}\|_1 = 1$. Vector $\mathrm{s}_i$ is the $i$th row vector of matrix $\mathbf{S}$ and the symbol $\langle\cdot\rangle$ means the mean value operation. The sparsity measure function $f(\cdot)$ is calculated by $-\log[p(\cdot)]$. Here $p(\cdot)$ is the prior sparse distribution of feature coefficients, and considered the strong sparse shape, $p(\cdot)$ is defined as follows for a stochastic vector $\mathbf{y}$ [6]:

$$p(\mathbf{y}) = \frac{1}{2b}\frac{(d+2)\left[0.5d\,(d+1)\right]^{(0.5d+1)}}{\left[\sqrt{0.5d\,(d+1)} + |\mathbf{y}/b|\right]^{(d+3)}} \tag{6}$$

where parameters $d, b > 0$, $d$ is a sparsity parameter and $b$ is a scale parameter. Parameters $d$ and $b$ are estimated according to the following equations [6]:

$$\begin{cases} b = \sqrt{E\left\{\mathrm{y}^2\right\}} \\ d = \frac{2-k+\sqrt{k(k+4)}}{2k-1} \end{cases} \tag{7}$$

where parameter $k = b^2 f_\mathrm{y}(0)^2$, and $f_\mathrm{y}(0)$ is the value of the function $f(\cdot)$ at zero. $\mathbf{A}$ and $\mathbf{S}$ are still updated in turn. In the inner loop, $\mathbf{A}$ is fixed, $\mathbf{S}$ is updated to minimize the object function $J(\mathbf{A},\mathbf{S})$. And in the external loop, $\mathbf{S}$ is fixed, and $\mathbf{A}$ is updated. Here, to reduce

the convergence time, **A** and **S** are at first updated by the gradient descent algorithm, and then they are further updated by using the multiplication factor referred to the document [4]. Combined the partial derivative of the $i$th row vector $\mathrm{a}_i$ of **A** and the multiplication factor rule, the updating process of $\mathrm{a}_i$ is deduced as follows:

$$\begin{cases} \nabla \mathbf{a}_i = \left[ \mathbf{X}(x, y) - \sum_{i=1}^{n} \mathbf{a}_i(x, y)\, \mathbf{s}_i \right] \mathbf{s}_i^T \\ \mathbf{a}_i^{(t+1)} = \left( \dfrac{\mathbf{a}_i^{(t)} \sum_i \mathbf{x}_i \dfrac{\mathbf{s}_k^{(t)}}{\sum_k \mathbf{a}_i^{(t)} \mathbf{s}_k^{(t)}}}{\sum_k \mathbf{s}_k^{(t)}} \right) \Big/ \left\| \mathbf{a}_i^{(t)} \right\| \end{cases} \tag{8}$$

And the updating process of matrix **S** is written as follows:

$$\begin{cases} \nabla \mathrm{s}_i = \mathrm{a}_i^T \left[ \mathbf{X}(x, y) - \sum_{i=1}^{n} \mathrm{a}_i(x, y) \mathrm{s}_i \right] + \lambda_1 f'(\mathrm{s}_i) + 2\lambda_2 \left[ \frac{(\mathrm{s}_i - \mathrm{s}_k)}{\mathbf{D}_\mathrm{W}} - \frac{(\mathrm{s}_k - \bar{\mathrm{S}})}{\mathbf{D}_\mathrm{B}} \right] \\ \mathrm{s}_i^{(t+1)} = \sqrt{\mathrm{s}_k^{(t)} \left( \dfrac{\left[ \mathrm{a}_i^{(t)} \right]^T \mathrm{x}_i}{\mathrm{a}_i^{(t)} \mathrm{s}_k^{(t)}} \right)} \end{cases} \tag{9}$$

where $f'(\mathrm{s}_i) = \left[ -\log(p(\mathrm{s}_i)) \right]'$ is the first-order derivative of $f(\mathrm{s}_i)$. By using the above updating methods described in Eq. (9), matrices **A** and **S** can be ensured to be positive, at the same time, the sparse coefficient vectors are guided to approach the true class center of samples.

## 4 Experimental Results Analysis and Conclusion

### 4.1 Feature Basis Extraction

In test, ten images of different individuals were chosen from the Hong Kong Polytechnic University (PolyU) palmprint image database to learn feature basis vectors of our NNSC model. This database includes 600 palmprint images pre-processed with the size of $128 \times 128$ from 100 individuals [8]. Each test image was randomly sampled 5000 times by using an $8 \times 8$ pixel window, and then each image patch was converted to one column vector with the size of $64 \times 5000$, thus, the training matrix **X** consisting of 10 test images was $64 \times 50,000$ pixels. Further, to reduce the calculation, matrix **X** was in advance centered and whiten by principal component analysis (PCA) method. Then, using the updating rules defined in Eqs. (8) and (9) of **A** and **S** in turn, the objective function given in Eq. (5) was minimized efficiently. The 64 feature basis vectors of palmprint images learned by our NNSC were shown in Fig. 1. Meanwhile, in the same condition, the 64 feature bases of Hoyer's NNSC were shown in Fig. 2. In Figs. 1 and 2, the white denotes positive pixels, the gray denotes zero pixels, and the black denotes negative pixels. Clearly, compared Figs. 1 with 2, in each grid, the gray area in Fig. 2 is larger than that in Fig. 1, namely, the bases of our NNSC behave much sparsity, which are similar to those of sparse pixels shown in the document [1]. At the same time, it is clearly to see that Fig. 2 has distincter white and black pixels than Fig. 1. This proves further that basis vectors of our NNSC model behave clearer locality than Hoyer's model.

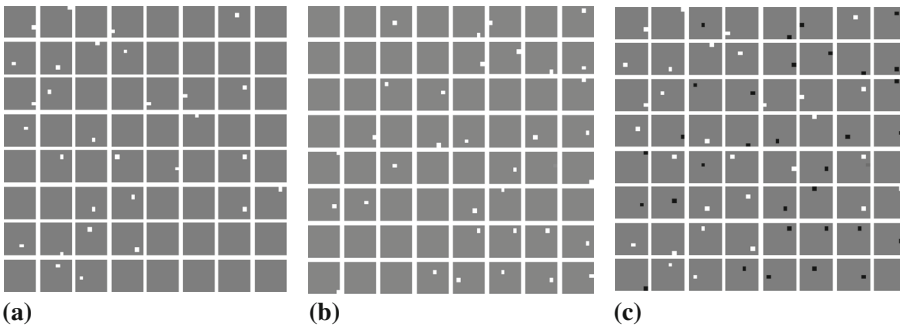


**Fig. 1** Basis vectors of our NNSC model. **a** ON-channel basis. **b** OFF-channel basis. **c** Bases of ON-channel minus OFF-channel

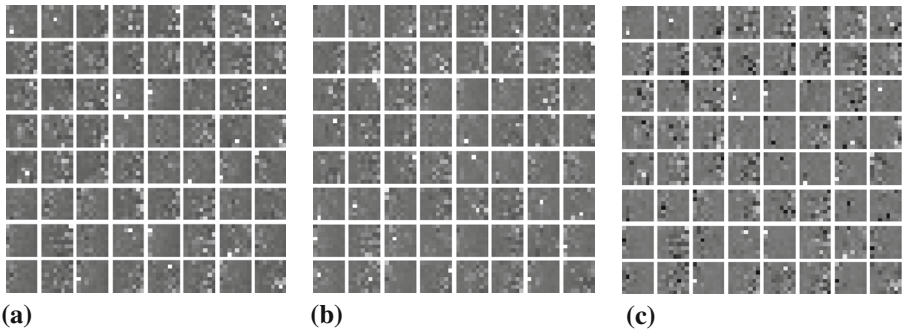


**Fig. 2** Basis vectors of basic NNSC model. **a** ON-channel basis. **b** OFF-channel basis. **c** Bases of ON-channel minus OFFchannel

### 4.2 Feature Recognition Test

Using feature bases learning by our NNSC algorithm, referring to the ICA frameworkI-Idescribed in [9], the feature classification task can be implemented. Here, the PolyU palmprint database was still used. To reduce computation, each image was reprocessed by wavelet to be the size of $64 \times 64$ pixels. Thus, the training sample set with $4096 \times 600$ pixels was obtained. For each person, the first three images were used as training images, and others were used as testing images. Therefore, the training and test set were the size of $4096 \times 300$. For the convenience for calculating, PCA was further used to reduce dimension so as to obtain an appropriate dimension $k$. Let $\mathbf{P}_k$ ($4096 \times k$ pixels) denote the matrix containing the first $k$ principal component (PC) axes in its columns and let $\breve{\mathbf{X}}$ denote the data set with zero-mean images, then the PC coefficient matrix $\mathbf{R}_k$ was represented by $\mathbf{R}_k = \breve{\mathbf{X}}^T \mathbf{P}_k$ ($300 \times k$ pixels). So, for testing set $\mathbf{X}_{train}$ with the size of $k \times 300$, the representation of train images was obtained in the columns of $\mathbf{S}_{train}$ as follows:

$$\mathbf{S}_{train} = \mathbf{A}^{-1} \cdot \mathbf{R}_{train}^T = \mathbf{A}^{-1} \cdot \left( \mathbf{X}_{train}^T \cdot \mathbf{P}_k \right)^T \tag{10}$$

where matrix $\mathbf{A}^{-1}$ with the size of $4096 \times k$ is the inverse or pseudo inverse of $\mathbf{A}$. In the same way, for the test set $\mathbf{X}_{test}$, the representation $\mathbf{S}_{test}$ of test images can be obtained. In recognition task, three types of popular classifiers, Euclidean distance, extreme learning

**Table 1** Recognition rates of different classifiers

| Algorithms ($k = 121$) | ELM (%) | SVM (%) | Euclidean distance (%) |
|---|---|---|---|
| PCA | 96.48 | 94.75 | 91.33 |
| DCB-NNSC | 97.82 | 97.64 | 93.72 |
| Basic NNSC | 96.73 | 96.252 | 92.68 |

**Table 2** The training time and classification time of different classifiers obtained by different algorithms

| Classifiers | Our NNSC (s) | | Basic NNSC (s) | | PCA (s) | |
|---|---|---|---|---|---|---|
| | Training | Classification | Training | Classification | Training | Classification |
| ELM | 0.031 | 0.015 | 0.063 | 0.031 | 0.078 | 0.035 |
| SVM | 4.827 | 0.253 | 5.762 | 3.527 | 23.326 | 12.625 |
| Euclidean distance | 8.524 | 5.526 | 18.846 | 7.836 | 35.627 | 16.163 |

machine (ELM) [9–12] and support vector machine (SVM) [13–15] were used to test features learned by our NNSC model. The SVM model was first developed by Vapnik for pattern recognition and function regression [10], and it has been proved to be very successful in pattern recognition [9,10]. Otherwise, it is also noted that the recognition rate obtained by SVM model has to do with the kernel functions selected [13,14]. Some published documents have proved that, among SVM kernels used frequently [13–15], the Gaussian radial basis function kernel and the wavelet kernel behave good recognition effect. ELM can provide the higher generalization performance at a much faster speed [10,11]. At present, there are many ELM variations that have been proposed, which have let to the state-of -the-art results in many applications, especially for the pattern recognition problem [11,12]. Here, we used the Kernel-based ELM model to implement the classification task.

In test, to determine the optimal feature length to reduce computation time, the PCA feature recognition with different $k$ dimension was first implemented by three classifiers mentioned-above. Based on this thought, according to experimental results, $k$ was finally selected as 121. Thus, used three classifiers, the recognition results of PCA and our DCB-NNSC features were obtained and listed in Table 1. At the same time, under the same test condition, the recognition results of Hoyer's NNSC features were also shown in Table 1. As well as, in order to compare the recognition speed of different classifiers, the training time and classification time of each classifier were also listed in Table 2. From Table 1, for DCB-NNSC features, it is clearly seen that the recognition results obtained by three classifiers all exceed 93 %. In some extent, it testifies also that DCB-NNSC model is successful in extracting image features, which are favorable to classify images. Otherwise, it also can see that no matter what kind of classifiers, the recognition rate of our DCB-NNSC model is the best, and that of PCA method is the worst. Otherwise, form Table 2, it is clear to see that, the training time and classification time of ELM classifier are the smallest than those of SVM and Distance classifiers. Moreover, despite of feature extraction algorithms, the training and classification time are both less than 1 minute. In other words, the experiment data testify that the classification speed of ELM is the best, and it behaves the fast classification speed.

Therefore, according to recognition results, it can be concluded that the DCB-NNSC model developed by us, like Hoyer's NNSC model, can model successfully the respective field of V1 in the primary visual system of human beings, and extract efficiently natural image features containing prior class information. Further, used in image classification task, the DCB-NNSC model is better than the basic NNSC model, and this mode is indeed a very promising in practical applications.

**Acknowledgments** This work was supported by the National Natural Science Foundation of China (Nos. 61373098, 61370109).

## References

1. Hoyer PO (2003) Modelling receptive fields with non-negative sparse coding. Neurocomputing 52:547–552
2. Olshausen BA, Field DJ (1996) Emergence of simple-cell receptive field properties by learning a sparse code for natural images. Nature 381:607–609
3. Li L, Zhang YJ (2009) SENSC: a stable and efficient algorithm for nonnegative sparse coding. Acta Autom Sin 35:439–443
4. Lee DD, Seng HS (1999) Learning the parts of objects by non-negative matrix factorization. Nature 401:788–891
5. Cao J, Lin Z (2014) Bayesian signal detection with compressed measurements. Inform Sci 289:241–253
6. Shang Li (2008) Non-negative sparse coding shrinkage for image denoising using normal inverse Gaussian density model. Image Vis Comput 26:1137–1147
7. Hyvärinen A (1997) Sparse coding shrinkage: denoising of nongaussian data by maximum likelihood estimation. Neural Comput 11:1739–1768
8. Cao J, Chen T, Fan J (2014) Fast online learning algorithm for landmark recognition based on BoW framework. In: Proceedings of the 9th IEEE Conference on Industrial Electronics and Applications. Hangzhou, China, June 2014, pp 1163–1168
9. Huang GB, Zhu Q, Siew CK (2006) Extreme learning machine: theory and applications. Neurocomputing 70:489–501
10. Cao J, Xiong L (2014) Protein sequence classification with improved extreme learning machine algorithms, BioMed Research International, vol. 2014, Article ID 103054, 12 pages. doi:10.1155/2014/103054
11. Huang GB, Chen L (2007) Convex incremental extreme learning machine. Neurocomputing 70:3056–3062
12. Huang GB, Chen L (2008) Enhanced random search based incremental extreme learning machine. Neurocomputing 71:3460–3468
13. Cortes C, Vapnik VN (1995) Support vector networks. Mach Learn 20:273–297
14. Chen GY, Xie WF (2006) Pattern recognition with SVM and dual-tree complex wavelets. Image Vis Comput 25:960–966
15. Zhang L, Zhou W, Jiao L (2004) Wavelet support vector machine. IEEE Trans Syst, Man, Cybern—Part B 34:34–39