LETTER

Nonnegative Component Representation with Hierarchical Dictionary Learning Strategy for Action Recognition

Jianhong WANG[†a)], ***Student Member***, **Pinzheng ZHANG**[†b)], ***and*** **Linmin LUO**[†c)], ***Nonmembers***

SUMMARY Nonnegative component representation (NCR) is a mid-level representation based on nonnegative matrix factorization (NMF). Recently, it has attached much attention and achieved encouraging result for action recognition. In this paper, we propose a novel hierarchical dictionary learning strategy (HDLS) for NMF to improve the performance of NCR. Considering the variability of action classes, HDLS clusters the similar classes into groups and forms a two-layer hierarchical class model. The groups in the first layer are disjoint, while in the second layer, the classes in each group are correlated. HDLS takes account of the differences between two layers and proposes to use different dictionary learning methods for this two layers, including the discriminant class-specific NMF for the first layer and the discriminant joint dictionary NMF for the second layer. The proposed approach is extensively tested on three public datasets and the experimental results demonstrate the effectiveness and superiority of NCR with HDLS for large-scale action recognition.

key words: *hierarchical dictionary learning strategy, nonnegative component representation, nonnegative matrix factorization, action recognition*

1. Introduction

Human action recognition in videos is one of the most active research topics in the field of computer vision and pattern recognition. The key problem for action recognition is how to represent different action video clips effectively and discriminatively. Bag of Visual Words (BoVW) recently appears as the most popular approach for this representation. In this framework, features are encoded with the visual words in codebook and a histogram of word occurrences is used to represent a video.

However, this BoVW representation only contains statistics of unordered visual words, the inside relationships between different visual words have not been considered. Recently, researchers have shown that nonnegative component representation (NCR) can overcome these limitations and achieve better result for action recognition [1], [2]. NCR is a mid-level representation which extracts action components from the low-level BoVW representation based on a nonnegative matrix factorization (NMF). The videos are further encoded using these action components. NCR is more compact and discriminative than the BoVW representation. However, it is shown in this paper that the NCR approach can be further improved by modifying the dictionary learning method for NMF.

Common dictionary learning methods can be roughly divided into three categories according to the structure of the dictionaries: (i) a single shared dictionary for all classes; (ii) multiple class-specific dictionaries (i.e. one for each class); (iii) a mixed approach. In the latter case, a joint dictionary is built with one common dictionary for all classes and a class-specific dictionary for each class. This method separates the shared and different parts of each sample, and allows boosting the classification performance for highly correlated or fine-grained categories [3].

However, none of the three methods is fully suitable for action recognition. With the growth of the dataset scale, it becomes more complicated to learn a single shared dictionary capable to well represent all classes. Meanwhile, the differences between classes may significantly vary: some classes are easy to separate, while some are highly correlated (e.g. running and jogging). Neither class-specific dictionary learning nor joint dictionary learning can perfectly handle this complex situation. To overcome this problem, and inspired by [4], we propose a hierarchical dictionary learning strategy (HDLS) in this paper to learn two-layer dictionaries. We partition the similar classes together into new groups based on a similarity matrix. After clustering, in the first layer, each group is disjoint from others, which is suitable for using class-specific dictionary learning. In the second layer, classes in each group are highly correlated and a joint dictionary learning method is applied to suppress the common part and amplify the individual features corresponding to the classes. To enhance the discrimination power of the dictionaries, the Fisher discrimination criterion is added for both learning algorithms. After dictionary learning, a two-layer NCR is obtained and we can use a two-layer hierarchical classifier for action recognition purpose.

The framework of our approach is illustrated in Fig. 1, and the rest of this paper is organized as follows: Section 2 briefly introduces the nonnegative component representation (NCR). Section 3 presents the hierarchical dictionary learning strategy (HDLS). Section 4 gives the classification scheme for NCR with HDLS. Experiments and comparisons with other methods are reported in Sect. 5 and Sect. 6 concludes the paper.

2. Nonnegative Component Representation

Nonnegative component representation is a mid-level rep-

Manuscript received July 26, 2015.
Manuscript revised November 9, 2015.
Manuscript publicized January 13, 2016.

[†]The authors are with School of Computer Science and Engineering, Southeast University, Nanjing, China.
a) E-mail: wjh.seu@gmail.com
b) E-mail: luckzpz@seu.edu.cn
c) E-mail: luo.list@seu.edu.cn

DOI: 10.1587/transinf.2015EDL8164

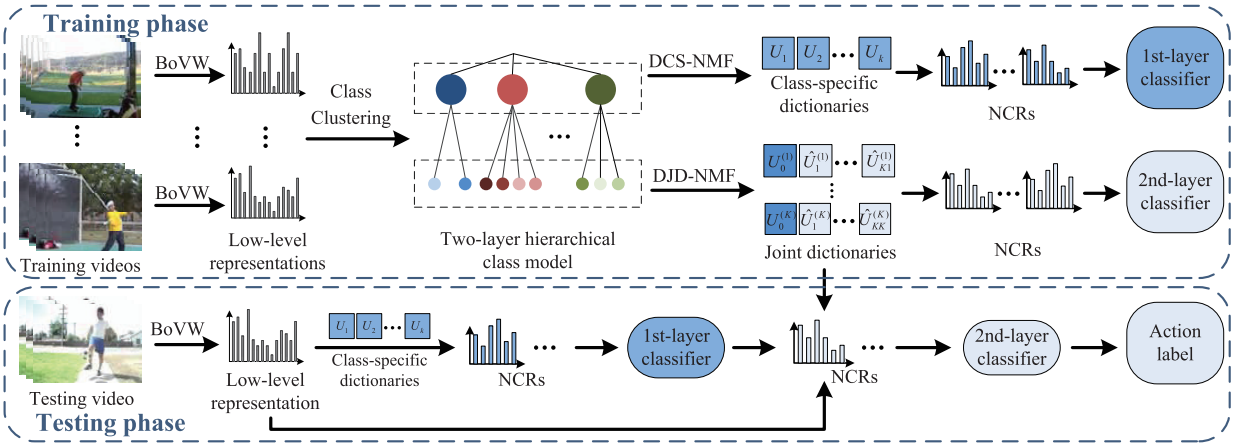

Fig. 1 Flowchart of the proposed work for action recognition.

resentation. It relies on NMF to decompose the low-level representation into a combination of nonnegative components, which reduces the dimension of representation vector. Moreover, NMF uses nonnegative constraints, and when compared to other matrix factorization methods, it can lead to a part-based and intuitive representation [5].

Let Y denote the low-level representation vectors for action videos, where each column represents one vector. NMF aims at finding two nonnegative matrices U and V to approximate the matrix Y: $Y \approx UV$. If we consider each column of matrix U as an action component constructed by several correlated visual words, then U becomes the action component dictionary, and each column of matrix V is the new mid-level representation for the corresponding video sample based on the dictionary U. U and V can be learned with two iterative update algorithm [5].

3. Hierarchical Dictionary Learning Strategy

3.1 Class Clustering

The purpose of class clustering is to separate various classes into two layers based on similarity measure, and to make further process easier to discriminate different classes.

The similarity between two classes is measured by the misclassification rate. We split the training data into a train set and a validation set, and train one-against-rest SVM model for each class based on the low-level representations. By evaluating the models on the validation set, the confusion matrix M can be obtained and the similarity matrix S can be defined as $S = (M + M^T)/2$.

As the number of clusters is not predefined, it is not suitable to use standard clustering algorithm such as K-means. We employ the Affinity Propagation (AP) technique [6] to cluster classes. AP is an exemplar-based cluster algorithm whose input is a similarity matrix and does not require the number of clusters. The only parameter to be set in AP is the preference value. In our experiments, the preference parameter of AP has been chosen larger than the median to make sure that classes in the same group are very similar.

After clustering, the classes form a two-layer hierarchical class model, the groups in the first layer being disjoint while, in the second layer, classes in each group are correlated.

3.2 Discriminant Class-Specific NMF

We propose the discriminant class-specific NMF (DCS-NMF) to learn the nonnegative component dictionaries for groups in the first layer. Firstly, let us consider the Fisher discriminant NMF (FD-NMF) formulated as:

$$\begin{aligned} &\min_{(U,V)} \|Y - UV\|_F^2 + \lambda \left[tr\left(S_W(V) - S_B(V)\right) + \eta \|V\|_F^2 \right] \\ &s.t.\ U, V \geq 0 \end{aligned} \tag{1}$$

where the second term is the Fisher discriminant term. $S_W(V)$ is the within-class scatter matrix of V, and $S_B(V)$ is the between-class scatter matrix of V. Based on the Fisher criterion, a more discriminative result can be achieved by minimizing the within-class scatter and maximizing the between-class scatter.

To learn the class-specific dictionary for each group, assume that $U = [U_1, U_2, \ldots, U_K]$ and that the data from each group are represented only by the words from the corresponding dictionary. Following [7], after some derivations from Eq. (1), we can obtain the DCS-NMF model:

$$\begin{aligned} &\min_{(U_i,V_i)} \sum_{i=1}^{K} \left[\|Y_i - U_i V_i\|_F^2 + \lambda_1 \|V_i - M_i\|_F^2 + \lambda_2 \|V_i\|_F^2 \right] \\ &s.t.\ U_i, V_i \geq 0 \end{aligned} \tag{2}$$

where $\lambda_1 = \lambda(1 + \kappa_i)$, $\kappa_i = 1 - n_i/n$ and $\lambda_2 = \lambda(\eta - \kappa_i)$. Y_i is the low level representation matrix for group i, V_i is the new representation for Y_i based on U_i, and M_i is the mean vector matrix of V_i. This model can be optimized group by group using the iterative update algorithm proposed in [5]. Multiplicative updates for U_i and V_i are given by

$$U_i \leftarrow U_i \odot \frac{Y_i V_i^T}{U_i V_i V_i^T} \tag{3}$$

$$V_i \leftarrow V_i \odot \frac{{U_i}^T Y_i}{{U_i}^T U_i V_i + 2\lambda_1 V_i P_i {P_i}^T + 2\lambda_2 V_i} \tag{4}$$

where $\odot$ is an element-wise product, $P_i = I_{n_i} - E_{n_i}/n_i$, I_{n_i} is the identity matrix of size n_i, $E_{n_i} = [1]_{n_i \times n_i}$ is the matrix of size $n_i * n_i$ with all entries being 1.

3.3 Discriminant Joint Dictionary NMF

The discriminant joint dictionary NMF (DJD-NMF) is designed to learn component dictionaries for correlated classes in each group in the second layer. The DJD-NMF learns a common dictionary and multiple class-specific dictionaries simultaneously. Similar to DCS-NMF, the model of DJD-NMF is also derived from the FD-NMF of Eq. (1). We slightly change the form of the between-class scatter matrix as $S_B(V) = \frac{1}{2}\sum_{i=1}^{K}\sum_{j=1,j\neq i}^{K}\left(m_i - m_j\right)\left(m_i - m_j\right)^T$. This formulation can simplify the derivation and leads to a more compact result.

Let $U = \left[U_0, \hat{U}_i, \ldots, \hat{U}_K\right]$ where U_0 is the shared dictionary and the rest sub-dictionaries are class-specific dictionaries. Assuming that the data are described only with the shared dictionary and the class-specific dictionary of its own class, we can derive the optimization of DJD-NMF from Eq. (1) as:

$$\begin{aligned}\min_{(U_0,\hat{U}_i,V_i)} \sum_{i=1}^{K} \Bigg[&\left\| Y_i - \left[U_0, \hat{U}_i\right] V_i \right\|_F^2 + \lambda_1 \left\| V_i - M_i \right\|_F^2 \\ &-\lambda_2 \sum_{j=1,j\neq i}^{K} \left\| m_i^0 - m_j^0 \right\|_2^2 + \lambda_3 \left\| V_i \right\|_F^2 \Bigg] \; s.t.\; U_0, \hat{U}_i, V_i \geq 0\end{aligned} \tag{5}$$

where $\lambda_1 = \lambda(1+\kappa_i)$, $\kappa_i = (K-1)/n_i$, $\lambda_2 = \lambda/2$ and $\lambda_3 = \lambda(\eta - \kappa_i)$. Define now $V_i = \left[V_i^0; \hat{V}_i\right]$, where V_i^0 and $\hat{V}_i$ are the coefficient matrices of class i for U_0 and $\hat{U}_i$ respectively, and m_i^0 in Eq. (5) is the mean vector of V_i^0.

The object function in Eq. (5) can be divided into two sub-procedures by optimizing the dictionaries in U and the coefficients V_i alternatively with the other one fixed.

Supposing that the dictionary U is fixed, we can compute the nonnegative coefficients V_i class by class. When updating V_i, all V_j, $j \neq i$ are fixed. Since m_i^0 can be represented as $m_i^0 = Q_i V_i R_i$, where $Q_i = [I_{n_i}, \mathrm{O}]$, $R_i = [1]_{n_i \times 1}$, then V_i can be computed with the following multiplicative update rule:

$$V_i \leftarrow V_i \odot \frac{{U_i}^T Y_i + 2\lambda_2 (K-1) {Q_i}^T Q_i V_i R_i {R_i}^T}{{U_i}^T U_i V_i + 2\lambda_1 V_i P_i {P_i}^T + 2\lambda_2 \sum\limits_{j=1,j\neq i}^{K} {Q_i}^T Q_j V_j R_j {R_i}^T + 2\lambda_3 V_i} \tag{6}$$

Then we update the dictionaries with the coefficients fixed. We first compute the class-specific dictionaries $\{\hat{U}_i\}_{i=1}^K$ class by class and after that we update the shared dictionary U_0. With the V_i and U_0 fixed, the optimization of class-specific dictionary $\hat{U}_i$ reduces to the following problem:

$$\min_{(\hat{U}_i)} \left\| Y_i - U_0 V_i^0 - \hat{U}_i \hat{V}_i \right\|_F^2 \; s.t.\; \hat{U}_i \geq 0 \tag{7}$$

After all the class-specific dictionaries $\{\hat{U}_i\}_{i=1}^K$ have been updated, fixing V_i and $\hat{U}_i$, Eq. (5) could be formulated as follows in order to update U_0:

$$\min_{(U_0)} \sum_{i=1}^{K} \left\| \left(Y_i - \hat{U}_i \hat{V}_i\right) - U_0 V_i^0 \right\|_F^2 \; s.t.\; U_0 \geq 0 \tag{8}$$

Equation (8) can be further written as:

$$\min_{(U_0)} \left\| Y^0 - U_0 V^0 \right\|_F^2 \; s.t.\; U_0 \geq 0 \tag{9}$$

where $V^0 = \left[{V_1}^0, \ldots, {V_K}^0\right]$, $Y^0 = [Y_1 - \hat{U}_1\hat{V}_1, \ldots, Y_K - \hat{U}_K\hat{V}_K]$. Both Eq. (7) and Eq. (9) are the standard form of NMF with coefficients fixed. Therefore, they can be optimized with the multiplicative update rule used for standard NMF.

4. Classification Scheme

Based on our hierarchical dictionary learning approach, a two-layer hierarchical classifier is built for classification. The first-layer classifier allows defining the group label of the sample, and the second-layer classifier predicts which class in such group this sample belongs to.

For the first layer, as the groups are disjoint to each other and relatively easy to separate, we simply use the reconstruction error and the distance between the representation vector and the mean vector as the classification metric. Define the query sample as y, and the representation of y over dictionary U_i as $\hat{\alpha}_i$. The metric function over each class-specific dictionary can be defined as:

$$e_i = \left\| y - U_i \hat{\alpha}_i \right\|_2^2 + \omega \left\| \hat{\alpha}_i - m_i \right\|_2^2 \tag{10}$$

where ω is the tradeoff parameter to balance the contribution of the two terms. The first layer will allocate y to the group that yields the smallest e_i.

For the second layer, considering that the classes in each group are correlated, we follow the classification scheme reported in [3] to get a more discriminative classifier. For a group i with K_i classes, Samples are trying to be represented over K_i different dictionaries $U_j^{(i)} = \left[U_0^{(i)}, \hat{U}_j^{(i)}\right]$, $j = 1, \ldots, K_i$. Therefore, we train K_i multiclass linear SVM models for the K_i different component representations. The final decision for the second layer is obtained by means of an equal voting scheme using the outputs of the K_i SVM models.

5. Experiments

5.1 Dataset and Experimental Setup

We evaluated the proposed method on three popular large-scale human action datasets: UCF50 [8], HMDB51 [9] and UCF101 [10]. All three datasets are complicated and include a number of different categories. For a fair comparison of results, the experimental settings selected in the original

papers [8], [9] and [10] were kept for UCF50, HMDB51 and UCF101 respectively.

Considering the success of the dense trajectory [11] in action recognition, we adopted three features HOG, HOF and MBH based on dense trajectory as low level features in our experiments. We employed the localized soft assignment [12] for low-level descriptor encoding. Localized soft assignment shows better accuracy than vector quantization and can keep the encoding results nonnegative, which is important for the NMF. The codebook size was set to 4000 for low-level features.

5.2 Evaluation of NCR with HDLS

We adopted the BoVW representation, the NCR with shared dictionary (NCR with SD) and the NCR with class-specific dictionaries (NCR with CSD) as baselines to evaluate our proposed NCR with HDLS. The BoVW representation is the most popular low-level representation method, and it is the basis for NCR. NCR with SD learns a single shared dictionary for all classes. Both BoVW and NCR with SD make use of SVM for classification. NCR with CSD relies on class-specific dictionaries for every class and follows the classification method introduced in Sect. 4 for the first layer of HDLS.

Table 1, Table 2 and Table 3 report the comparison results on UCF50, HMDB51 and UCF101 datasets respectively. The following observations can be drawn when analyzing these experimental results. First, by comparing the results of three NCR methods on the three datasets, we can observe that the proposed HDLS performs better than the other two common dictionary learning methods. This shows that the proposed hierarchical processing strategy and the proposed two NMF based dictionary learning algorithms for two layers can increase the discrimination ability of dictionary and improve the recognition accuracy. Second, we notice that the improvement brought on the UCF101 dataset by our approach is larger than the benefit obtained on the other datasets. This demonstrates that the proposed approach is more suitable for large-scale and complex situations. Third, all three NCR methods have higher accuracies than BoVW, This exemplifies that the mid-level component representation is more compact and effective than the low-level BoVW representation. The study on four methods demonstrates the effectiveness of our approach for large-scale action recognition.

Table 1 Comparison of NCR with HDLS and other methods with different low-level features on UCF50

	HOG	MBH
BoVW	78.3%	83.4%
NCR with SD	79.6%	85.2%
NCR with CSD	80.1%	86.1%
NCR with HDLS	**82.6%**	**88.7%**

Table 2 Comparison of NCR with HDLS and other methods with different low-level features on HMDB51

	HOG	MBH
BoVW	37.3%	46.5%
NCR with SD	38.6%	48.1%
NCR with CSD	39.1%	48.9%
NCR with HDLS	**40.9%**	**52.2%**

Table 3 Comparison of NCR with HDLS and other methods with different low-level features on UCF101

	HOG	MBH
BoVW	69.2%	73.9%
NCR with SD	71.1%	76.2%
NCR with CSD	72.7%	78.1%
NCR with HDLS	**75.4%**	**82.3%**

5.3 Comparison with State-of-the-Art Methods

We fused three low-level features (HOG, HOF and MBH) at the component representation level and compared our combination results with some recent attempts reported in the literature on UCF50, HMDB51 and UCF101 datasets. The comparison result is listed in Table 4. We can observe that our method leads to a better performance on all three datasets. The improvement brought by our method over the state-of-the-art ones is equal to 1.2% on the UCF50 dataset, 0.4% on the HMDB51 dataset and 0.6% on the UCF101 dataset.

6. Conclusion

This paper proposes a novel hierarchical dictionary learning strategy (HDLS) for NCR to overcome the limitations of traditional dictionary learning methods for action recognition. HDLS processes disjoint classes and correlated classes separately in order to face the high variability of action types. It firstly clusters the similar classes into groups and forms a two-layer hierarchical class model. Then, HDLS takes account of the different properties in the two layers using different algorithms for dictionary learning. This approach has been extensively tested on three large-scale datasets using different low-level features and compared with other NCR and state-of-the-art methods. The experimental results

Table 4 Comparison of our method with state-of-the-art methods

UCF50		HMDB51		UCF101	
Sadanand et al. [13]	57.9%	Sadanand et al. [13]	26.9%	Soomro et al. [10]	43.9%
Reddy et al. [8]	76.9%	Wang et al. [14]	42.1%	Karpathy et al. [15]	63.3%
Wang et al. [14]	78.4%	Shi et al. [16]	47.6%	Wu et al. [17]	84.2%
Shi et al. [16]	83.3%	Wang et al. [11]	57.2%	Wang et al. [11]	85.9%
Wang et al. [11]	91.2%	Hou et al. [4]	57.9%	Hou et al. [4]	87.0%
Our Method	**92.4%**	Our Method	**58.3%**	Our Method	**87.6%**

demonstrate the effectiveness and superiority of NCR with HDLS for large-scale action recognition.

Acknowledgments

This work was supported by the Natural Science Foundation of Jiangsu Province under Grants BK2010426.

References

[1] H. Wang, C. Yuan, W. Hu, H. Ling, W. Yang, and C. Sun, "Action recognition using nonnegative action component representation and sparse basis selection," IEEE Trans. Image Process., vol.23, no.2, pp.570–581, 2014.
[2] Y. Tian, Q. Ruan, G. An, and R. Liu, "Local non-negative component representation for human action recognition," ICSP, pp.1317–1320, 2014.
[3] N. Zhou, Y. Shen, J. Peng, and J. Fan, "Learning inter-related visual dictionary for object recognition," CVPR, pp.3490–3497, 2012.
[4] R. Hou, A.R. Zamir, R. Sukthankar, and M. Shah, "Damn–discriminative and mutually nearest: Exploiting pairwise category proximity for video action recognition," ECCV, vol.8691, pp.721–736, 2014.
[5] D.D. Lee and H.S. Seung, "Learning the parts of objects by non-negative matrix factorization," Nature, vol.401, no.6755, pp.788–791, 1999.
[6] B.J. Frey and D. Dueck, "Clustering by passing messages between data points," Science, vol.315, no.5814, pp.972–976, 2007.
[7] M. Yang, L. Zhang, X. Feng, and D. Zhang, "Sparse representation based fisher discrimination dictionary learning for image classification," Int. J. Comput. Vis., vol.109, no.3, pp.209–232, 2014.
[8] K.K. Reddy and M. Shah, "Recognizing 50 human action categories of web videos," Mach. Vision Appl., vol.24, no.5, pp.971–981, 2013.
[9] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "Hmdb: a large video database for human motion recognition," ICCV, pp.2556–2563, 2011.
[10] K. Soomro, A.R. Zamir, and M. Shah, "UCF101: A dataset of 101 human actions classes from videos in the wild," Tech. Rep. CRCV-TR-12-01, UCF, 2012.
[11] H. Wang and C. Schmid, "Action recognition with improved trajectories," ICCV, pp.3551–3558, 2013.
[12] L. Liu, L. Wang, and X. Liu, "In defense of soft-assignment coding," ICCV, pp.2486–2493, 2011.
[13] S. Sadanand and J.J. Corso, "Action bank: A high-level representation of activity in video," CVPR, pp.1234–1241, 2012.
[14] L. Wang, Y. Qiao, and X. Tang, "Motionlets: Mid-level 3d parts for human motion recognition," CVPR, pp.2674–2681, 2013.
[15] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," CVPR, pp.1725–1732, 2014.
[16] F. Shi, E. Petriu, and R. Laganiere, "Sampling strategies for real-time action recognition," CVPR, pp.2595–2602, 2013.
[17] J. Wu, Y. Zhang, and W. Lin, "Towards good practices for action video encoding," CVPR, pp.2577–2584, 2014.