Representative Video Action Discovery Using Interactive Non-negative Matrix Factorization

Hui Teng[1,2], Huaping Liu[2], Lianzhi Yu[1], and Fuchun Sun[2]

[1] School of Optical-Electrical and Computer Engineering,
University of Shanghai for Science and Technology, Shanghai
[2] Department of Computer Science and Technology, Tsinghua University, Beijing
State Key Lab. of Intelligent Technology and Systems, Beijing
hpliu@tsinghua.edu.cn

Abstract. In this paper, we develop an interactive Non-negative Matrix Factorization method for representative action video discovery. The original video is first evenly segmented into some short clips and the bag-of-words model is used to describe each clip. Then a temporal consistent Non-negative Matrix Factorization model is used for clustering and action segmentation. Since the clustering and segmentation results may not satisfy the user's intention, two extra human operations: MERGE and ADD are developed to permit user to improve the results. The newly developed interactive Non-negative Matrix Factorization method can therefore generate personalized results. Experimental results on the public Weizman dataset demonstrate that our approach is able to improve the action discovery and segmentation results.

Keywords: Interactive action summarization, Non-negative Matrix Factorization, video analysis.

1 Introduction

There has been a lot of interests in developing practical systems to automatically understand video data. Of the many related tasks, discovering representative actions from video clip is of considerable practical importance. Such algorithms could automatically extract representative actions within streaming or archival video and therefore significantly improve the efficiency of video understanding.

In Ref.[6], a Bayesian non-parametric model of sequential data is adopted to allow completely unsupervised activity discovery. The authors claim that this work need not predefine the relevant behaviors or even their numbers, as both of them are learned directly from data. However, such a method admits the following disadvantages: (1)Due to the complexity of non-parametric Beyasian method, its time burden is rather huge; (2) The number of behaviors, although need not to be determined by the user, is still sensitive to some parameters of the algorithm (especially, the Dirichlet prior parameter). That is to day, the task of determining the number of behavior does not diminish, but is replaced

X. Hu et al. (Eds.): ISNN 2015, LNCS 9377, pp. 205–212, 2015.
DOI: 10.1007/978-3-319-25393-0_23

by another task to determine a more uninterpreted parameter. (3) The inference algorithm may introduce randomness. This leads to inconsistent results from multiple runs when the human factor is incorporated in to the loop. Such a problem was pointed by Ref.[3]. In Ref.[1], a relevance feedback strategy is proposed to help action search and localization in video database. All of the above work do not consider how to use the human-machine interface to enhance the action discovery performance. Recently, Ref.[7] addresses this problem for image clustering by introducing some human operation, and Ref.[3] used interactive non-negative matrix method for document topic discovery. To the best of the knowledge of the authors, there is no related work to solve video action discovery using human operation. This motivates us to solve this problem. The main task of this work is to discover the action categories within a video sequence, and identify such actions in this video sequence. The main contributions are summarized as follows: (1)We develop an interactive non-negative matrix factorization method for representative video action discovery. (2)We design two human operations: ADD and MERGER to realize the relevance feedback and enhance the video summarization performance. (3)We develop a practical software system and perform extensive experimental validations for the proposed method.

The rest of this paper is organized as follows: Section 2 is about the video representation. In section 3 we give a detailed introduction about the proposed method and Section 4 presents the experimental results.

2 Video Representation

The first-of-all task to analyze a video is to transform it into some suitable structured form. In this work, we follow the popular Bag-of-Words framework which was successfully utilized many action analysis work. To this end, we use Spatio-Temporal Interest Points (STIPs) to detect interest points and obtain Histogram-Of-Gradients (HOG) and Histogram-Of-Optical flow (HOF) descriptors. The obtained default descriptors is of $d = 162$ dimensions. We evenly divide to original video into segments which length is T frames. The parameter T is specified by the users. It should ensure the action consistency within each segment. In this work, we select $T = 24$ frames,which means about one second. These segments, which are denoted as $\mathbf{P}_1, \mathbf{P}_2, \cdots, \mathbf{P}_N$, represent the basic units of the actions. The final action summary should include such segments. The value N is obtained by the ceil of the whole frame numbers divided by T.

To give a formal representation of the segments, we first cluster all of the descriptors in this video into K clusters.The parameter K is also a meta-parameter which is specified by the users. A larger K will give better accuracy, but will also slow down the summarization period. In this work we empirically set it as 128. The obtained K cluster centers are regarded as code-words. Then each descriptor is mapped to the nearest code-word and each segment can be represented as a K-dimensional BoW histogram[1]. We therefore can represent the whole video as $\{\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_N\}$, where $\mathbf{y}_i$ is the K-dimensional BoW histogram for the i-th segment. After this period, each video can be represented as a matrix $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_N] \in R^{K \times N}$.

3 Non-negative Matrix Factorization for Video Action Discovery

3.1 Basic Non-negative Matrix Factorization

Given the matrix $\mathbf{Y} \in R^{K \times N}$ which includes the low-level action information of the original video, where N is the number of examples in the video. We then face the problem of how to extract the representative action clips from the matrix $\mathbf{Y}$ and then project each column to the corresponding representative action clip, providing the action segmentation results. A representative method is the popular Non-negative Matrix Factorization (NMF) in Ref.[2], which solves the following optimization problem:

$$\min_{\mathbf{U},\mathbf{V}} \ ||\mathbf{Y} - \mathbf{UV}||_F^2 \qquad s.t. \ \mathbf{U} \geq 0, \ \mathbf{V} \geq 0 \tag{1}$$

where $\mathbf{U} \in R^{K \times r}$ and $\mathbf{V} \in R^{r \times N}$ are two non-negative matrices. The term-topic matrix $\mathbf{U}$ uncovers the latent topic structure of the actions and r is usually set by the users and chosen to be smaller than K or N.

Once the solutions of $\mathbf{U}$ and $\mathbf{V}$ are obtained, we can subsequently infer the topic presentations of segments, namely the topic-segment matrix $\mathbf{V}$ by projecting the segments into the latent topic space. Such a model was originally proposed in Ref.[3] and then was used in many fields such as document clustering and image clustering. However, in our work, since we deal with continuous video, the temporal consistence should be encouraged to reflect the continuity of action. Therefore the model is modified as

$$\begin{aligned} &\min_{\mathbf{U},\mathbf{V}} \ ||\mathbf{Y} - \mathbf{UV}||_F^2 + \beta \sum_{i=1}^{N-1} ||\mathbf{V}_{i+1} - \mathbf{V}_i||_F^2 \\ &s.t. \ \mathbf{U} \geq 0, \ \mathbf{V} \geq 0 \end{aligned} \tag{2}$$

where β is a parameter to encourage the temporal consistency term, and $\mathbf{V}_i$ represents the i-th column of $\mathbf{V}$.

After obtaining the solutions $\mathbf{U}$ and $\mathbf{V}$, we can easily obtain the discovered representative actions and the temporal action segmentation results. The details are described as follows. For $\mathbf{U}$, each column $\mathbf{U}_i \in R^K$ corresponds a representative action clip. By searching $i^* = \underset{j \in [1,N]}{\operatorname{argmin}} \frac{\mathbf{U}_i^T \mathbf{y}_j}{||\mathbf{U}_i||_2 \cdot ||\mathbf{y}_j||_2}$, we can use the the video clip $\mathbf{P}_{i^*}$ as the corresponding representative action clip. On the other hand, we use the column $\mathbf{V}_j \in R^r$ for $j = 1, 2, \cdots, N$ to determine the cluster assignment of the j-th video clip and therefore realize the action segmentation. Concretely speaking, we search the maximum element in the vector $\mathbf{V}_j$ and use the corresponding index as the clustering assignment results.

3.2 Interactive Non-negative Matrix Factorization

In Ref.[3], some interesting interaction operation, such as key-words operations are used for interactive topic discovery or refinement. Such operations cannot been exploited in the video scenarios. The main reason is that for document clustering, the dictionary atom is the conventional words (such as *dog*, *apple*, *play*, *eat*, and so on.) which have the semantic meanings, so we can use the keyword distribution of each topic to realize the visualization. However, it is impossible to construct such a dictionary for a video. In our case, the dictionary is learning using K-means clustering algorithm and therefore the words do not have any semantic meaning. As a result, the key-words based operation defined in Ref.[3] cannot be used. Due to the same reason, such a visualization manner is not suitable in our case. To this end, we developed two interaction operations: ADD and MERGE for the visualized video action discovery.

MERGE Operation. The merge operation tries to solve the problem that some similar video segments may be clustered into different topics. This is unavoidable due to at least two reasons: (1) The semantic gap between the human understanding and the adopted BoW model which is based on low-level feature descriptor. (2) The results are not consistent to the user's intention.

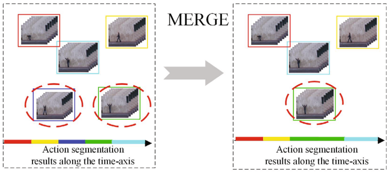

MERGE
Action segmentation results along the time-axis
Action segmentation results along the time-axis

Fig. 1. MERGE operation

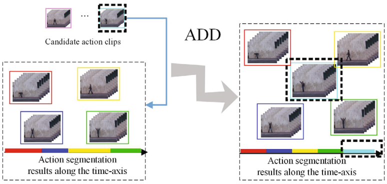

...
Candidate action clips
ADD
Action segmentation results along the time-axis
Action segmentation results along the time-axis

Fig. 2. ADD operation

To solve this problem, we permit the user to click the visualization action boxes and click the button *merge* to tell the computer that some segments should be merged into the same topic in the next iteration. This interaction also provides very important supervised information that we can exploit to enhance our model. In fact, the visualization of actions are shown as the video segments $\mathbf{P}_{t_1}, \mathbf{P}_{t_2}, \cdots, \mathbf{P}_{t_r}$. Without loss of generalization, we denote the selected merge segments as $\mathbf{P}_i$ and $\mathbf{P}_j$, then we add this pair into a set $\mathcal{M} = \mathcal{M} \cup \{(i,j)\}$, then we solve the following optimization problem in the next iteration:

$$\min_{\mathbf{U},\mathbf{V}} ||\mathbf{Y} - \mathbf{U}\mathbf{V}||_F^2 + \beta \sum_{i=1}^{N-1} ||\mathbf{V}_{i+1} - \mathbf{V}_i||_F^2 + \gamma \sum_{(i,j)\in\mathcal{M}} ||\mathbf{V}_i - \mathbf{V}_j||_F^2 \tag{3}$$
$$s.t.\ \mathbf{U} \geq 0,\ \mathbf{V} \geq 0.$$

The main characteristic of this model is the third term which encourages the i-th and j-th segments to share the similar topic pattern, and γ is a trade-off parameter. Please note that the pair set $\mathcal{M}$ is set to empty for the first iteration. During iteration procure, once $\mathcal{M}$ is added with some pair elements, it always play roles in the subsequent iterations.

ADD Operation. Though the above model can successfully discover most of the representative actions from the video, it is still possible that some important action clips cannot be discovered automatically. To this end, a candidate list of new action clips should be presented to the user for performing the ADD operation. Such a list should be short and representative. Ideally, it should contain only the actions which are not included in the list of the discovered topical actions. That is to say, it should not be well reconstructed by the discovered representative actions. Based on the above discussion, we design a performance index to evaluate the novelty of each action clip. To this end, for each video segments, we define its confidence about the topic assignment. We regard $\bar{\mathbf{V}}_i$ as the L_1 normalized i-th column of $\mathbf{V}$, and then adopt its entropy function as $En(\mathbf{V}_i) = -\sum\limits_{j=1}^{r} \bar{\mathbf{V}}_i(j) \log \bar{\mathbf{V}}_i(j)$. Obviously, when there is only one element of $\mathbf{V}_i$ is nonzero and equal to one, then the entropy is zero and the confidence score is maximum. On the other hand, when all the elements of $\mathbf{V}_i$ are equal to $1/r$, then the entropy is equal to $\log_2 r$ and the confidence score is minimum. Therefore, it is very convenient to adopt the entropy to select the most uncertain video segments for the operation ADD. In this work, we sort the entropies (in descending order) of all video segments which are not visualized and not deleted in the former stages, and then select the top N_a segments for visualization in a specifically design region and the user can browse them in a short time and then select some ones to add in the next iterations. The number N_a should not be too large, otherwise the user will be strongly confused. In this paper, it is set to 5. That is to say, at each iteration stage, we provide 5 most uncertain video segments for the user for possible ADD operation.

Once some action of which representation is $\mathbf{y}_i$ is selected to be added, then we should increase the number of r by one in the next operation and make some adjustments. Concretely speaking, we augment the topic matrix as $\bar{\mathbf{U}} = [\mathbf{U}\ \mathbf{y}_i] \in R^{N \times (r+1)}$. The optimization problem then becomes:

$$\min_{\mathbf{V}} ||\mathbf{Y} - \bar{\mathbf{U}}\mathbf{V}||_F^2 + \beta \sum_{i=1}^{N-1} ||\mathbf{V}_{i+1} - \mathbf{V}_i||_F^2 + \gamma \sum_{(i,j)\in\mathcal{M}} ||\mathbf{V}_i - \mathbf{V}_j||_F^2, s.t.\ \mathbf{V} \geq 0 \tag{4}$$

Note that in the above model, $\bar{\mathbf{U}}$ is known and only $\mathbf{V}$ should be calculated.

3.3 Optimization Method

All of the model in (2), (3) and (4) can be efficiently solve by the regularized NMF method proposed in Ref.[4]. To this end, we should construct a nearest neighbor graph to encode the consistency information of the data points. Consider a graph with vertices where each vertex corresponds to a data point. Define the edge weight matrix $\mathbf{W} \in R^{N \times N}$ as follows:

$$\mathbf{W}_{ij} = \begin{cases} \beta, & \text{if } |i-j| = 1 \\ \gamma, & \text{if } \{i,j\} \in \mathcal{M} \text{ and } |i-j| \neq 1 \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

Define a diagonal matrix $\mathbf{D}$, whose entries are column sums of $\mathbf{W}$, i.e., $\mathbf{D}_{ii} = \sum_{j=1}^{N} \mathbf{W}_{ij}$. Then the reformulated optimization problem leads to the two new following update rules[4]:

$$\mathbf{U}_{ij} \leftarrow \mathbf{U}_{ij}\frac{\left(\mathbf{Y}\mathbf{V}^T\right)_{ij}}{\left(\mathbf{U}\mathbf{V}\mathbf{V}^T\right)_{ij}}, \mathbf{V}_{ij} \leftarrow \mathbf{V}_{ij}\frac{\left(\mathbf{U}^T\mathbf{V}+\mathbf{V}\mathbf{W}\right)_{ij}}{\left(\mathbf{U}^T\mathbf{U}\mathbf{V}+\mathbf{V}\mathbf{D}\right)_{ij}} \tag{6}$$

where the subscript ij represents the i, j-th element in the corresponding matrix. The detailed algorithm flow and analysis can be found in [4].

4 Performance Evaluation

4.1 Dataset

We use the well-known Weizman database[5] of 90 low-resolution video sequences showing 9 different people, each performing 10 natural actions such as run, walk, skip, jack, jump, pjump, side, wave2, wave1 and bend. To evaluate the performance of our interactive method, we have created a "stitched" version of the weizman dataset into uninterrupted sequences. Each sequence depicts a single person performing 10 actions for a total duration of approximately 700 frames.

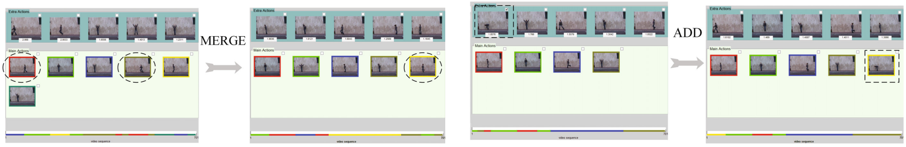
How to evaluate our approach is still an open problem. Generally, if ground truth is available, many evaluation metrics are available for clustering, such as purity, and normalized mutual information. To this end, the ground truth for each video is established manually based on the exact actions in every single sequence.

4.2 Operation Process

Figure 3 illustrates the two operations. In some cases, similar actions may be extracted. Such case often occurs when the number r is set to a large value. The operation MERGE allows us to merge the similar actions selected by the user when he press the *merge* button, which is shown in Figure 3(a). On the other hand, we need to find as more actions in the whole video sequence as possible. Figure 3(b) demonstrates the process of this operation. Users select the new actions from the list of candidate actions. By pressing the *add* button, the selected actions are added into the clustering results. Note that when performing either ADD or MERGE operations, we modify the model to produce the expecting clustering performance according to users' operations and the action segmentation result is demonstrated along the time axis in different color bars.

4.3 Performance Evaluation on the Interactive Interaction

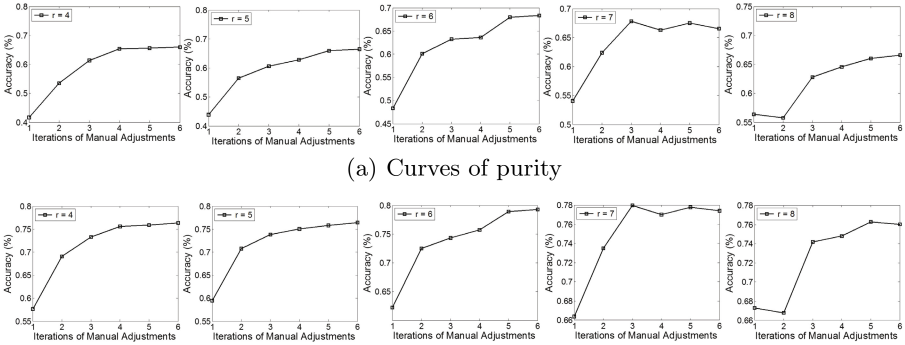
To evaluate the action segmentation results in each iteration, we adopt the purity and NMI indices which are popular in the community of clustering. Purity[8] is a simple and transparent evaluation measure. To compute purity, each cluster is assigned to the class which is most frequent in the cluster, and then the accuracy of this assignment is measured by counting the number of correctly assigned segments and dividing by N which is the total number of the whole video segments.



(a) MERGE Operation

(b) ADD Operation

Fig. 3. Demonstrations of MERGE and ADD operation



(a) Curves of purity

(b) Curves of NMI

Fig. 4. Clustering performance of purity and NMI.

Denote the ground-truch action clustering results as $\mathit{\Omega} = \{\omega_1, \omega_2, \ldots, \omega_g\}$ and $\mathbb{C} = \{c_1, c_2, \ldots, c_r\}$ as the practical clustering results, then the purity is defined as $Purity\,(\mathit{\Omega}, \mathbb{C}) = \frac{1}{N}\sum\limits_{k} \max\limits_{j} |\omega_k \cap c_j|$. Since high purity value is easy to achieve when the number of clusters is large and particularly, purity is 1 if each segment gets its own cluster. Thus we cannot use purity only to trade off the quality of the clustering against the number of clusters. Normalized mutual information(NMI)[8] is confident to make this tradeoff and can be information-theoretically interpreted $\mathbf{NMI}\,(\mathit{\Omega}, \mathbb{C}) = \frac{I(\mathit{\Omega};\mathbb{C})}{[H(\mathit{\Omega})+H(\mathbb{C})]/2}$, where I is the mutual information and $H(\cdot)$ represents the entropy.

Using these two evaluation metrics, we conduct experiments with Weizman dataset and there are 3 different users involved in this process. They make their adjustments to obtain the willing performance which is to find as more actions as possible during the whole process. By setting the different initial value of the number of clusters(r ranges from 4 to 8), we compute the average accuracy of the 9 video sequence. Figure 4 illustrates the clustering performance using our interactive method. We can see that the clustering performance can be significantly improved by adopting manual interactive adjustments. Note that when $r = 8$, the accuracy declined. Because when the number of cluster r is getting large, users have to compromise to the higher possibility of exploiting same actions so that they need to merge the very several actions, which results in the phenomenon that fewer number of actions leads to lower accuracy rate.

5 Conclusion and Future Work

This paper proposed an interactive method to detect representative actions within streaming or archival video. Incorporated with user's intention, expecting results have been obtained. However, there still exists a lot work to be further investigated. Firstly, we wish to extend the work on single video to video sets and discover more sensible behavior patterns for the end users; Secondly, we hope to develop more flexible interface and more high-level knowledge of the human can be incorporated in to the model. Finally, we wish to discover the hierarchical structure of the action in the video in a coarse-to-fine manner.

Acknowledgments. This work was supported in part by the National Key Project for Basic Research of China under Grant 2013CB329403; and in part by the Tsinghua University Initiative Scientific Research Program under Grant 20131089295.

References

1. Shao, L., Jones, S., Li, X.: Efficient Search and Localization of Human Actions in Video Databases. IEEE Trans. Circuits Syst. Video Techn. 24(3), 504–512 (2014)
2. Lee, D.D., Seung, H.S.: Algorithms for non-negative matrix factorization. Advances in Neural Information Processing Systems, 556–562 (2001)
3. Choo, J., Lee, C., Reddy, C.K., Park, H.: Utopian: User-driven topic modeling based on interactive nonnegative matrix factorization. IEEE Trans. on Visualization and Computer Graphics, 1992–2001 (2013)
4. Cai, D., He, X., Wu, X., Han, J.: Non-negative matrix factorization on manifold. In: Proc. of Eighth IEEE International Conference in Data Mining(ICDM), pp. 63–72 (2008)
5. Blank, M., Gorelick, L., Shechtman, E., Irani, M., Basri, R.: Actions as space-time shapes. In: Proc. of Tenth IEEE International Conference on Computer Vision (ICCV), pp. 1395–1402 (2005)
6. Hughes, M.C., Sudderth, E.B.: Nonparametric discovery of activity patterns from video collections. In: Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 25–32 (2012)
7. Wang, M., Ji, D., Tian, Q., Hua, X.S.: Intelligent photo clustering with user interaction and distance metric learning. Pattern Recognition Letters, 462–470 (2012)
8. Evaluation of clustering. `http://nlp.stanford.edu/IR-book/html/htmledition/evaluation-of-clustering-1.html`