LEARNING A SPARSE DICTIONARY OF VIDEO STRUCTURE FOR ACTIVITY MODELING

Nandita M. Nayak, Amit K. Roy-Chowdhury

University of California, Riverside

ABSTRACT

We present an approach which incorporates spatiotemporal features as well as the *relationships* between them, into a sparse dictionary learning framework for activity recognition. We propose that the dictionary learning framework can be adapted to learning complex relationships between features in an unsupervised manner. From a set of training videos, a dictionary is learned for individual features, as well as the relationships between them using a stacked predictive sparse decomposition framework. This combined dictionary provides a representation of the structure of the video and is spatio-temporally pooled in a local manner to obtain descriptors. The descriptors are then combined using a multiple kernel learning framework to design classifiers. Experiments have been conducted on two popular activity recognition datasets to demonstrate the superior performance of our approach on single person as well as multi-person activities.

Index Terms— Sparse coding, activity recognition, multiple kernel learning

1. INTRODUCTION

Most traditional approaches for activity recognition [1] extract a set of spatio-temporal features which represent the points of interest in the video. These feature descriptors are then fed to classifiers to learn different action categories. Recently, it has been widely acknowledged that, in addition to the features themselves, the structural similarities [2] between sets of features play an important role in discriminating between activities. In other words, the spatiotemporal arrangement of features can provide contextual information to distinguish between actions that result in similar feature sets. Researchers have proposed different ways to represent this structure. Approaches such as [2], [3] have encoded these relationships as logical relations or as graph-based models. However, such methods use "hand picked" attributes to represent structure and/or require graph matching algorithms to quantify structural similarities. We propose that sparse dictionary learning methods can be used to automatically learn a natural representation of the structural relationships between features.

Recently, sparse coding techniques have gained popularity in the field of activity recognition. Different methods of sparse dictionary learning such as deep Boltzmann machines [4], stacked auto-encoders and sparse coders [5] have been used to represent image data in the context of object recognition. A 3D convolutional network learned over a fixed set of input frames to represent the video was proposed in [6]. A cascading systems of independent subspace analysis and spatial pooling was used to learn a set of local features which was then classified using K-means vector quantization and χ^2 kernel in [7]. Action attributes were modeled using a sparse dictionary based representation in [8]. Anomaly detection was performed by measuring the encoding error of features learned using sparse coding in [9]. Sparse coding in conjunction with spatial temporal pyramid matching was proposed for activity recognition in [10]. Shift invariant sparse coding was used for activity recognition in [11]. However, most of these approaches work on a video representation using pixel data computed over the entire video and do not explore structural relationships between features.

As an alternative, we design a sparse dictionary learning approach to learn a compact dictionary from well-known features such as space time interest points or HOG features. We suggest that, these interest points provide an effective and compact representation of patterns in a video. However, for efficient activity recognition, an effective encoding of these features is essential. This is where sparse coding can be useful. Sparse dictionary learning provides us with a more efficient non-linear encoding method as compared to vector quantization [12]. In addition, we also utilize sparse dictionary learning to learn a "combination dictionary" for a set of "combination features" which can then be used along with the individual features for recognition. The combination features would encode more complex relationships between features as compared to traditional approaches.

Given a set of training videos, we first compute some well known features such as [13]. We then perform a dictionary learning for the features using a predictive sparse decomposition (PSD) algorithm. This is a sparse dictionary learning framework which learns a dictionary and a set of sparse coefficients, as well as an "encoder" through a feedback mechanism. The encoder can be used to generate the sparse coefficients for any given input by a simple matrix vector multiplication. Similarly, we also learn a relationship dictionary

This work has been partially supported by NSF grant CNS-1330110.

which is learned through a PSD with pairs of features as input. This dictionary encodes the structural relationship between different pairs of features as observed in the training data. A local pooling strategy is then applied to the individual features as well as the relationship features to generate feature descriptors. These feature descriptors are then combined to learn a discriminative classifier using a multiple kernel learning approach.

The main contributions of our work are: 1) We propose a novel representation to learn the *structural* relationships between features in the context of activity recognition. 2) We learn a compact dictionary of structure elements using predictive sparse decomposition, which can then be used to train classifiers in a manner similar to individual feature dictionaries. 3) We describe a method to combine the individual features with structural features using a multiple kernel learning framework for classification of activities.

2. SPARSE CODING FRAMEWORK

2.1. Video Representation

We use STIP points [13] as a set of individual features, or salient points in the video. We utilize the histogram of gradients (HOG) descriptor constructed over these interest points. Therefore, the video is composed of a set of p individual STIP features given by $F^{ind} = \{f_1^{ind}, f_2^{ind}....f_p^{ind}\}$.

We also define pairs of features in a video as "combination features" since they capture the structural information in the video. A combination feature is obtained by a concatenation of a pair of features along with their relative spatial and temporal information. A combination feature can be written as $f_{ij}^{comb} = [(x_i - x_j), (x_i - x_j), (t_i - t_j), f_i^{ind}, f_j^{ind}]$. The set of all combination features in a video can be given by $F^{com} = \{f_{ij}^{comb} \forall i, j \in \{1, 2, ..p\}\}$. Note that, we scale the space and time coordinates of the interest points to lie in a unit cube to make it independent of the actual co-ordinates in a video. Also, here, we restrict the model to computing combination features between points which are within a predefined spatio-temporal distance of each other. This distance can be fixed by the user.

2.2. Stacked Predictive Sparse Coding (SPSD)

A 2-layer stacked predictive sparse decomposition coder model is constructed to generate the feature dictionary and combination dictionary. We begin by describing a single layer of the predictive sparse decomposition coder (PSD) and then extend the model to the stacked PSD.

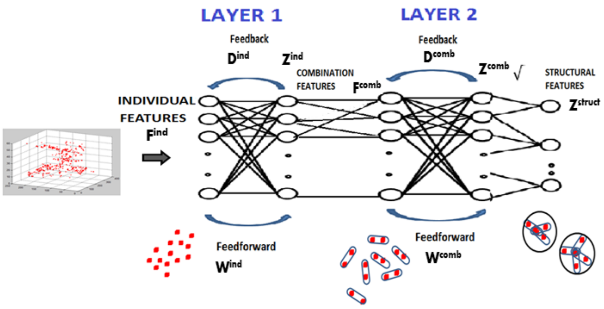

Fig. 1. The figure illustrates the architecture of the SPSD used in our approach. Layer 1 PSD is trained on individual features F^{ind} to obtain sparse coefficients Z^{ind}. These are used to generate combination features F^{comb} which are fed to Layer 2 PSD to obtain coefficients Z^{comb}. Coefficients of all pairs related to a single feature are combined to form structural features Z^{struct}.

2.2.1. *Predictive Sparse Decomposition (PSD)*

A predictive sparse decomposition algorithm [14] is an unsupervised learning algorithm for learning a one-layer model that computes a distributed representation for its input. Given a set of input features, $F \in \Re^m$, which are obtained from the training data, the objective of the PSD is to arrive at a representation $Z \in \Re^n$ for each input F using a set of k basis functions forming the columns of a dictionary matrix or decoder $D \in \Re^{n \times m}$. For an input vector of size n forming the input X, the PSD consists of three components: an encoder W, the dictionary D and a set of codes Z. The overall optimization function is expressed as:

$$G(F; Z, W, D) = \|WF - Z\|_2^2 + \lambda\|Z\|_1 + \|DZ - F\|_2^2, \quad (1)$$

where $F \in \Re^n$, $Z \in \Re^k$, dictionary $D \in \Re^{n \times k}$ and encoder $W \in \Re^{k \times n}$. The first term represents the feedforward or the encoding, the second term denotes the sparsity constraint and the last term denotes the feedback/decoding. λ is a parameter that controls the sparsity of the solution, i.e., sparsity is increased with higher value of λ. The parameter λ is varied between 0.01 and 1 in steps of 0.05 and the optimum value is selected through cross validation to minimizes $G(.)$. Here, we used $\lambda = 0.2$.

The learning protocol involves computing the optimal D, W and Z that minimizes $G(.)$. The process is iterative by fixing one set of parameters while optimizing others and vice versa, i.e., iterate over steps (2) and (3) as given below.

1. Randomly initialize D and W.
2. Fix D and W and minimize Equation 1 with respect to Z, where Z for each input vector is estimated via the gradient descent method.
3. Fix Z and estimate D and W, where D, W are approximated through stochastic gradient descent algorithm.

The stochastic gradient descent algorithm approximates the true gradient of the function by the gradient of a single example or the sum over a small number of randomly chosen training examples in each iteration.

2.2.2. Cascading the PSD

Layer 1: The first layer of SPSD is designed for representing individual features. Given a set of N_T training videos, the input to Layer 1 PSD are the set of all individual features detected in these videos $F^{ind} = \{f^{ind}_{(1,1)}, ..f^{ind}_{(1,p_1)}, ..f^{ind}_{(N_T,1)}.. f^{ind}_{(N_T,p_{N_T})}\}$, where video i contains p_i features. The PSD has connections from all input nodes to all output nodes. A layer 1 encoder W^{ind} and a layer 1 decoder D^{ind} are learned. The output nodes are the sparse coefficients computed for the input features $Z^{ind} = \{z^{ind}_{(1,1)}, z^{ind}_{(1,2)}..z^{ind}_{(1,p_1)}, ..z^{ind}_{(N_T,1)}.. z^{ind}_{(N_T,p_{N_T})}\}$.

Layer 2: The second layer of the SPSD is designed for representing combination features. We use the output of the first layer, which is the sparse coefficients to generate the input for the second layer. Since we intend to use both individual as well as combination features for recognition, using the same lower level representation to learn a higher level representation has the advantage that the statistical information of the first layer is also shared with the second layer [15]. We wish to model relationships between pairs of features which are within a pre-defined spatio-temporal distance.

The combination features for the second layer are obtained by concatenating the coefficients of related pairs of features along with their relative spatial and temporal information. The input to the second layer is therefore given by $F^{comb} = \{[(x_{(k,i)} - x_{(k,j)}), (y_{(k,i)} - y_{(k,j)}), (t_{(k,i)} - t_{(k,j)}), z^{ind}_{(k,i)}, z^{ind}_{(k,j)}], k \in \{1, 2..N_T\}\}, i, j \leq p_k$. The PSD has connections from all input nodes to all output nodes. A layer 2 encoder W^{comb} and a layer 1 decoder D^{comb} are learned. The output nodes are the sparse coefficients computed for the combination features $Z^{comb} = \{z^{comb}_{(k,i,j)}, k \in \{1, 2..N_T\}\}, i, j \leq p_k\}$. Since each individual feature can lie in the vicinity of multiple other features, we compute a structural feature by combining the coefficients of all pairs related to a single feature as $z^{struct}_{(k,i)} = \sqrt{\sum_{j=1}^{n_{k,i}} (z^{comb}_{k,i,j})^2} / n_{k,i}$, where $n_{k,i}$ is the number of pairs related to the i^{th} feature in the k^{th} training video. The output of the layer 2 PSD is therefore the collection of structural features Z^{struct}.

The two layers of the SPSD are learned sequentially in a greedy manner using the method described in Sec 2.2.1. The SPSD is illustrated in Figure 1.

Spatiotemporal Pooling: The obtained coefficients are pooled to form one individual descriptor V^{ind} and one combination descriptor V^{comb} per video. We divide the video into equal sized non-overlapping spatio-temporal blocks and perform a max pooling of the data over each block. For the i^{th} spatiotemporal block, the descriptor generated is given by $v_i = \max(\mid z^{struct}_{(i,1)} \mid, \mid z^{struct}_{(i,2)} \mid, .. \mid z^{struct}_{(m_i,1)} \mid)$, where m_i is the number of codes generated in the i^{th} spatiotemporal block. The final descriptor is obtained by concatenating all local descriptors.

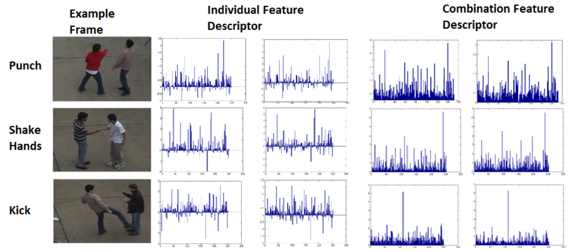

Fig. 2. The figure shows some sample frames from the UT Interaction dataset for different retrieved activities along with two instances of individual feature descriptors and combination feature descriptors of the corresponding activity. It can be seen that the combination feature descriptors look distinct for different activities and have well defined peaks.

2.3. Classification of Activities

For classification of activities, we wish to combine the two features in a discriminative framework. Having two different sets of features for a data, like the individual features and the combination features in our case, we can define the discriminant as a combination function of two kernels, one for each set of features.

$$\kappa_c(V_i, V_j) = f_c(\{\kappa_f(V_i^f, V_j^f)\}_{f=1}^2), \tag{2}$$

where the combination function $f_c : \Re^2 \rightarrow \Re$ can be linear or non linear. The kernel functions $\{\kappa_f : \Re^{D_m} \times \Re^{D_m} \rightarrow \Re\}_{f=1}^2$ is defined for the two sets of features, each of dimension D_m.

Here, we assume f_c to be a linear weighted combination of the two kernels. Therefore, the combination kernel is defined as

$$\kappa_c(V_i, V_j) = w_1 \kappa_1(V_i^{ind}, V_j^{ind}) + w_2 \kappa_2(V_i^{comb}, V_j^{comb}) \tag{3}$$

We choose the kernels to be polynomial. Optimal performance was achieved for a polynomial kernel of order 3. The function is solved using an SVM base learner as described in [16]. The weights w_1 and w_2 are decided experimentally using cross-validation.

3. EXPERIMENTS

We perform experiments on two publicly available datasets: the UT Austin Interaction data [2] and the UCLA office dataset [18]. The UT Interaction dataset consists of high resolution video of two actors performing actions such as handshake, kicking, hugging, pushing, punching and pointing. The videos are of different lengths and the activities are

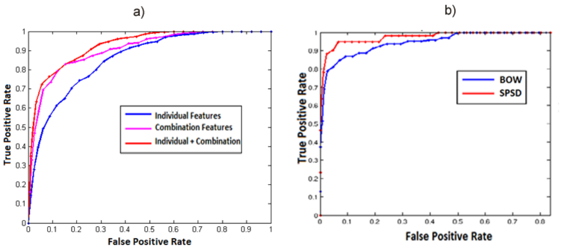

Fig. 3. Figure a) shows the ROC curves obtained using individual feature descriptors in an SVM framework, combination feature descriptors in an SVM framework and individual + combination feature descriptors using our approach for the UT Interaction dataset. Figure b) shows the ROC curves for individual features and individual+combination features using our approach for the UCLA dataset.

Method	**BOW**	**Ryoo[2]**	**Gaur[17]**	**SPSD**
Precision	50.3	70.1	71.5	76.7
Recall	52	73.1	73.1	74.2

Table 1. Precision and recall values of methods BOW, Ryoo et al [2] and Gaur et al [17] and our approach for the UT Interaction dataset.

performed from two different viewpoints. The UCLA office dataset consists of indoor and outdoor videos of single and two-person activities. Here, we perform experiments on the lab scene containing close to 35 minutes of video captured with a single fixed camera in a room. We work on 10 single person activities: Enter lab, exit lab, sit down, stand up, work on laptop, work on paper, throw trash, pour drink, pick phone receiver and place receiver down.

For both datasets, we utilize the first half of the data as training and the second half as testing. Each video is rescaled to lie in a unit volume cuboid for generality. The individual feature dictionary and combination dictionary are computed over the training dataset in an unsupervised manner (without considering the activity labels). The combination features are computed for all features lying in a neighborhood of 0.1 from the feature under consideration in the unit volume. The dictionary size for the UT Interaction data was taken to be 250 for the individual features and 250 for the combination features. The dictionary size for the UCLA data was taken as 500 elements for the individual as well as combination features.

3.1. Analysis of the results

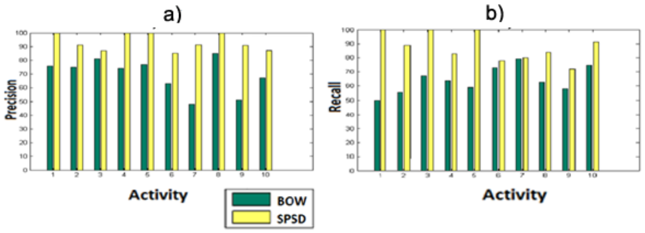

Fig. 4. Figures show the precision and recall obtained on the UCLA dataset with our approach. Comparison has been shown to the performance of BOW classifier [13]. The activities are mentioned in Section 3.

Some example of the individual feature descriptors and the combination feature descriptors generated for one block of data for the UT Interaction data are shown in Figure 2. It can be seen that there are distinct peaks in the histograms. It can also be noticed that the combination features look more distinct as compared to the individual features. The performance of our method on the UT Austin Interaction data is shown in Table 1. It is seen that the performance of our method is superior as compared to other state of the art methods like [2] and [17]. As shown in the table, we achieved an overall recognition accuracy of 76.7%, while [2] achieves an accuracy of 70.1% and [17] achieves an accuracy of 71.5% with half the data used for training. The ROC curves for recognition of activities using just the individual features in an SVM framework, just the combination features in an SVM framework and both sets of features in a MKL framework are shown in Figure 3 a).

For the UCLA dataset, we analyzed overall recognition accuracy against [18] and the Bag-of-Words classifier [13]. The output of baseline classifier on unsegmented data gives an accuracy of 78.7% while [18] has obtained an overall accuracy of 92.3%. Our SPSD approach gives an overall accuracy of 93.1%. The values of precision and recall for BOW and BOW+context are shown in Figure 4. The ROC curve in Figure 3 b) shows the improvement in performance achieved by using our approach as compared to the Bag-of-Words classifier.

4. CONCLUSION

In this paper, we have proposed a method to learn a natural representation for the structure in a video using predictive sparse decomposition framework. We have proposed a 2-layer stacked PSD where the first layer computes sparse coefficients for individual features and these are used to compute a sparse dictionary for combination of features. These combination features are seen to be distinctive for different activities, thereby increasing the performance of recognition in conjunction with the individual features. In the future, we plan to extend this work to have more levels of hierarchy to be able to analyze structure in videos at different resolutions. This work can also be extended to other applications where structure of the data is crucial.

5. REFERENCES

[1] J. K. Aggarwal and M. S. Ryoo, "Human activity analysis: A review," *ACM Computing Surveys*, 2011.

[2] M.S. Ryoo and J.K. Aggarwal, "Spatio-temporal relationship match: Video structure comparison for recognition of complex human activities," in *International Conference on Computer Vision*, 2009.

[3] M. R. Amer and S. Todorovic, "Sum-product networks for modeling activities with stochastic structure," in *Computer Vision and Pattern Recognition*, 2012.

[4] R. Salakhutdinov and G. Hinton, "A betterway to pretrain deep boltzmann machines," in *Neural Information Processing Systems*, 2012.

[5] Q. V. Le, M. A. Ranzato, R. Monga, M. Devin, K. Chen, G. Corrado, J. Dean, and A. Ng, "Building high-level features using large scale unsupervised learning," in *International Conference on Machine Learning*, 2012.

[6] S. Ji, W. Xu, M. Yang, and K. Yu, "3d convolutional neural networks for human action recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2012.

[7] Q. V. Le, W. Y. Zou, S. Y. Yeung, and A. Y. Ng, "Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis," in *Computer Vision and Pattern Recognition*, 2011.

[8] Z. Jiang Q. Qiu and R. Chellappa, "Sparse dictionary-based representation and recognition of action attributes," in *International Conference on Computer Vision*, 2011.

[9] B. Zhao, L. Fei-Fei, and E. Xing, "Online detection of unusual events in videos via dynamic sparse coding," in *Computer Vision and Pattern Recognition*, 2011.

[10] X. Zhang, H. Zhang, and X. Cao, "Action recognition based on spatial-temporal pyramid sparse coding," in *International Conference on Pattern Recognition*, 2012.

[11] C. Vollmer, H. Gross, and J. P. Eggert, "Learning features for activity recognition with shift-invariant sparse coding," in *International Conference on Artificial Neural Networks*, 2013.

[12] A. Coates and A. Y. Ng, "The importance of encoding versus training with sparse coding and vector quantization," in *International Conference on Machine Learning*, 2011.

[13] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," in *International Conference on Computer Vision*, 2005.

[14] K. Jarrett, K. Kavukcuoglu, M. Ranzato, and Y. LeCun, "What is the best multi-stage architecture for object recognition?," in *International Conference on Computer Vision*, 2009.

[15] H. Lee, "Tutorial on deep learning and applications," in *NIPS 2010 Workshop on Deep Learning and Unsupervised Feature Learning*, 2010.

[16] M. Gonen and E. Alpaydin, "Multiple kernel learning algorithms," *Journal of Machine Learning Research*, 2011.

[17] U. Gaur, Y. Zhu, B. Song, and A. Roy-Chowdhury, "String of feature graphs analysis of complex activities," in *International Conference on Computer Vision*, 2011.

[18] M. Pei, Y. Jia, and S.-C. Zhu, "Parsing video events with goal inference and intent prediction," in *International Conference on Computer Vision*, 2011.