Exploiting High Dimensional Video Features Using Layered Gaussian Mixture Models

Datong Chen
Computer Science Department
Carnegie Mellon University
datong@cs.cmu.edu

Jie Yang
School of Computer Science
Carnegie Mellon University
yang+@cs.cmu.edu

Abstract

Analysis of video data usually requires training classifiers in high dimensional feature spaces. This paper proposes a layered Gaussian mixture model (LGMM) to exploit high dimensional features for classifying various shots in video. LGMM decomposes a high dimensional feature space by building a pyramid structure and estimating the distribution of local partitions in each layer using Gaussian mixtures from the bottom of the pyramid to the top. We reduce the dimension of features in each local region at a lower layer by projecting them onto the estimated Gaussian components. These projected feature vectors are then used to estimate the Gaussian mixture models at a upper layer. The final dimension of the feature is adjustable by choosing the number of Gaussians at the top layer of the pyramid. We demonstrate the proposed method using motion features to classify video shots. The proposed method is independent from low level features and can be extended to other classification tasks.

1 Introduction

The booming usage of digital video presents a challenge to exploit rich information in video processing and content-based video indexing and retrieval. Many different features, such as motion, color histogram, text, and edge, have been utilized for video analysis. These features, however, are usually in high dimensional spaces. For example, most of high density motion features developed by computer vision researchers assign a value to each pixel of an image, i.e., the dimension of a feature vector of an image equals to the number of pixels in the image. Suppose we use image differences as the motion feature for a video with 720x480x30 resolution, a feature vector in a one second interval has 10,368,000 dimensions. Such a high dimensionality brings difficulty for training and classification. In fact, these feature vectors contain much redundancy, and dimension reduction approaches and granulation approaches should be employed to remove it.

Dimension reduction approaches transform a high dimensional feature space into a lower dimensional feature space without losing much discrimination am0ng classes. Many dimension reduction approaches use linear criterion, for example principal component analysis (PCA), factor analysis, projection pursuit, and independent component analysis (ICA). Linear approaches can be transformed to non-linear versions by applying the kernelization [3] technique, which can reduce non-linear redundancy from the data. There are also characterized non-linear approaches, such as principle curves, multidimensional scaling, topographic mapping, and vector quantization. A survey [2] is a good reference for the details of these approaches. Granulation approaches treat a document (video or image) as a bag of local features. Local features can be regular partitions of the input feature space of local sampling results under some criterion [4]. Granulation approaches transform the original high-dimensional problem (the video) into low dimensional problem (local features). The drawback is that characteristic structures among local features are not used in these approaches, which may not very efficient to model motion information. For example, lifting an arm vs. lifting a leg, these two motions consist of many similar local motion features but can be characterized spatially.

To exploit spatial-temporal information in video, in this paper, we propose a layered Gaussian mixture model (LGMM) to address high-dimensionality problem. A LGMM estimates the distribution of local features in multiple scales. It decomposes a high dimensional feature space by building a pyramid structure and estimating the distribution of observed local features in each layer using Gaussian mixtures from the bottom of the pyramid to the top. The observed features in each layer are then transformed into a lower dimensional space by projecting them onto the estimated Gaussian components. The projected features are then treated as the observed feature of the next upper layer.

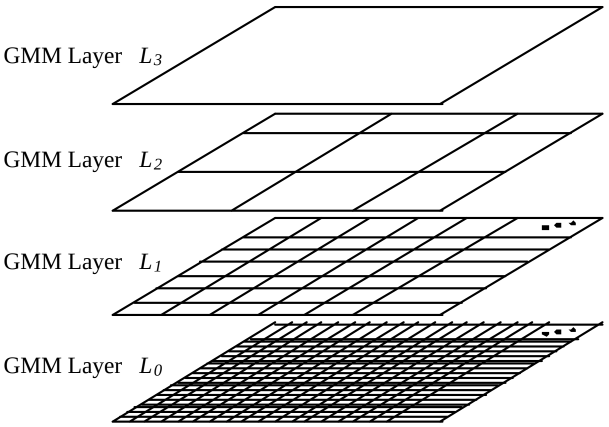

Figure 1. An illustration of a pyramid structure of a feature image with three layers.

The final dimension of the feature is adjustable by choosing the number of Gaussians at the top layer of the pyramid.

2 A LAYERED GAUSSIAN MIXTURE MODEL

The intuitive idea of LGMM is to bottom-up model a high-dimensional spatial-sensitive data by merging local models step by step. Given a high-dimensional spatial-sensitive data, e.g., a feature image with n dimensional feature vectors aligned in regular 2D latices, we can build a pyramid structure of the data, as shown in the Figure 1. To simplify the discussion, we only show a three-layer model in the figure. Each layer of the pyramid is defined as a triple-unit:

$$L_i = (PAR_i, M_i, F_i),$$

where PAR_i is the partition plan, M_i denotes the Gaussian mixture model and F_i is the observed feature.

The partition plan contains the partition information of the layer with N_i dimensions and is denoted as a set of subsets of dimensions $PAR_i = \{p_{i1}, ..., p_{iK_i}\}$, where $p_{ij} \subseteq \{1, ..., N_i\}$. The partition plan in each layer is predefined. Although the partition plan can be adaptive estimated and partitions can be in irregular shapes, we use only regular granularity as partition plans in each layer in this paper to simplify the discussion.

The model M_i is a mixture of m_i Gaussians:

$$M_i \sim \sum_{h=1}^{m_i} w_h^i N(\mu_h^i, \Sigma_h^i). \tag{1}$$

It models the joint probability density function of feature vectors in different partitions. Since all the partitions have the same dimensions, all the mean vectors and variance matrices can keep the same dimensions. The number of Gaussians m_i is estimated in the training process, which is discussed in Section 4.

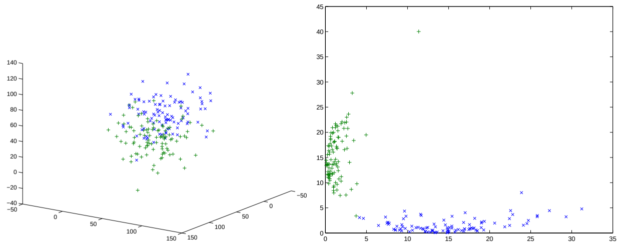

Figure 2. An illustration of a pyramid structure of a feature image with three layers.

The observed feature F_0 of the bottom layer can be features in pixels. The observed feature $F_i = \{f_{i1}, ..., f_{iN_i}\}$ denotes N_i feature vectors, where each feature vector f_{ij} is computed from the jth partition at the lower layer L_{i-1}. Given a partition $p_{i-1\ j}$, its observation feature vectors $f_{i-1\ k}$ $(k \in p_{i-1\ j})$ in the layer L_{i-1} and the model M_{i-1}, we first concatenate all the observed feature vectors as $f_{i-1}(p_{i-1\ j}) = (f_{i-1\ k})$ $(k \in p_{i-1\ j})$ and then compute the projection of the concatenated feature vector $f_{i-1}(p_{i-1\ j})$ onto the m_{i-1} Gaussian components:

$$y_{i-1\ h} = w_{i-1\ h} P\left(f_{i-1}(p_{i-1\ j}) | N(\mu_h^{i-1}, \Sigma_h^{i-1})\right), \tag{2}$$

where $w_{i-1\ h}$ and $N(\mu_h^{i-1}, \Sigma_h^{i-1})$ are weights and Gaussian components of the model M_{i-1}. The feature vector f_{ij} in the current layer L_i is then defined as a m_{i-1} dimensional vector razed by a specific function R:

$$f_{i\ j} = R\left(y_{i1}, ..., y_{im_{i-1}}\right), \tag{3}$$

Without considering the razing function, the vector y_i is a projection of the concatenated feature vector onto each component of the model M_i with the weight of the component. The intuitive meaning of this projection is shown in Fig. 2. Suppose the observed features in the layer L_i are 3-dimensional vectors as shown on the left. We choose a simple partition plan such that each partition contains only one data point and train the model M_i to have two components. Then, the projection vector y_i is plotted as the figure on the left.

The razing function R is used to reduce the variance in the projected vectors. The basic idea is to set values of $y_{i-1\ h}$ to zero if it is smaller than a threshold τ. We use adaptive threshold for each vector in this paper by it's median value.

The dimension D_i of observed features at layer L_i ($i > 0$) equals to the number partitions multiplying the number of

Gaussians in the layer below: $D_i = m_{i-1}N_i = m_{i-1}K_{i-1}$. Therefore, in order to reduce the dimension of the feature in a higher layer, e. g. the layer L_i, with predefined numbers of partitions K_i and K_{i-1}, the number of Gaussians in the layer should satisfy the follow condition:

$$m_i < m_{i-1}\frac{K_{i-1}}{K_i}. \tag{4}$$

Although the lower level feature can be different, in this paper, we will use motion as an example to illustrate the concept without losing generality.

3 Initial motion feature extraction

Many types of features have been proposed to characterize motions of various objects and activities with pixel level precision. In this paper, we use two types of motion features: image difference and edge motion history image.

3.1 Image difference

A simple way to produce dense motion feature is to compute the difference between two consecutive images. For any given frame of an image sequence I_t at time t and its previous image I_{t-1} at time $t-1$, the difference motion feature at each pixel x is computed as:

$$D_t(x) = I_t(x) - I_{t-1}(x). \tag{5}$$

The motion of all pixels between two images can be represented as a difference image D_t. This feature is easy to be computed but it has a serious drawback because the value of $D_t(x)$ indicates neither the direction of the movement of the pixel x nor the velocity value. Image differencing operation extracts a feature image from very frame, which is too expensive even for storage. To reduce, amount of the data, we compute the average of difference image through a video short.

3.2 Edge motion history image

Edge motion history image, computed by combining edge detection and motion history image (MHI) techniques, is another feature we can extract a temporal-compressed feature vector from a short video sequence

MHI was proposed by [1] to compress temporal information of the human activities. An MHI is a binomial model to represent recent object movements. In [1] an MHI is computed from silhouettes of objects segmented using background subtraction and stereo depth subtraction. However, the background is not easy to be extracted in news and sports videos with complex background scenes. Stereo depth information is usually not available in video data either. In this paper, we propose to use edge information detected in each frame image instead of silhouettes to compute edge motion history image (EMHI).

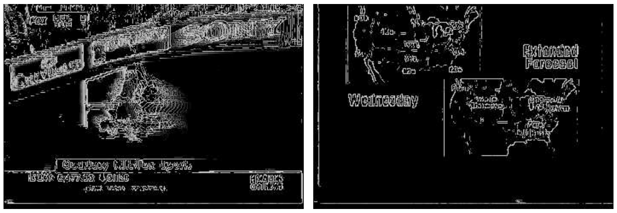

Figure 3. Two examples of EMHIs extracted from CNN news video.

Let $E_t(x)$ be a binary value to indicate if pixel x is located on an edge at time t. A EMHI $H_t^\tau(x)$ is computed from the EMHI of the previous frame $H_{t-1}^\tau(x)$ as:

$$H_t^\tau(x) = \begin{cases} \tau, & if \;\; E_t(x) = 1 \\ max(0,\; H_{t-1}^\tau(x) - 1), & otherwise. \end{cases} \tag{6}$$

The EMHI compresses the temporal dimensions using binomial model of bounded edge frequency at each pixel to obtain a low dimensional feature vector, for example a vector with 720×480 dimensions instead of $720 \times 480 \times 30$ dimensions. Figure 3 shows some examples of the EMHIs extracted from CNN news. Another advantage of the EMHI in comparison with the optical flow is that the computation of EMHI takes less time for the same mount of video data.

4 Training LGMM

The estimation of LGMM parameters is performed layer by layer from the bottom to the top. In each layer, the model is first estimated using Expectation-Maximization (EM) algorithm and then they are used to reduce the feature dimensions and to produce observed feature of the layer right above. The algorithm is described in the following table.

5 Experiments

To evaluate the proposed approach, we choose 60 minutes CNN video from the TRECVID'05 data set to estimate all the GMMs in layer 1 and layer 2. Classification topics of video shots based on the TRECVID'03 data set. We randomly extract 100 video shots for each of 6 topics: "indoor", "out-door", "news-person", "news-subject", "sport" and "weather" as a training set and 100 video shots for the first 4 topics, 78 video shots for "sports" and 84 shots for "weather" (there is no more) as a testing set. There is no

Table 1. Algorithm of LGMM training

- 1. Given the initial motion features F_0 of training videos and the partition plans of each layer $\{PAR_i\}$;
- 2. Loop from bottom layer L_0 to the top layer L_T, do steps 3 to 7;
- 3. For each layer L_i, set step counter the number of component $m_i = 1$;
- 4. Learn model $M_i^{m_i}$ using EM algorithm;
- 5. Compute BIC value $B_{m_i} = -log\,(P(F_i|M_i^{m_i}) + m_i log(n)$, where n is the number training samples in this layer;
- 6. If $B_{m_i} < B_{m_i-1}$, $m_i = m_i + 1$, go to step 4;
- 7. Compte F_{i+1} using the model $M_i^{m_i-1}$;
- 8. Output the low dimensional features F_{T+1} for each training video.

video shot contains more than one topic. There is also no training data and test data are extracted from the same video in each topic.

The partition plan of the bottom layer consists of $K_0 = 36 \times 24$ number of 10×10 non-overlapping blocks. The partition plan of the first layer consists of $K_1 = 6 \times 4$ number of 6×6 blocks. We treat the layer 2 as one partition.

We extract features of EMHI and temporal averaging image difference (Diff). In comparison, the feature dimensions are reduced by granulation (local) PCA and LGMM. Figure 4 shows the accuracies of four schemes: Diff.PCA, the first component of local PCA of Diff features in $n \times n$ cell partitions ($n = 1, \ldots, 10$); Diff.LGMM, the Diff feature compressed using LGMM; EMHI.PCA, the first component of local PCA of EMHI feature in $n \times n$ cell partitions; and EMHI.LGMM, the EMHI feature compressed using LGMM. Classifiers are trained by performing 10-fold cross validation on support vector machines and tested on the testing set.

In the granulation PCA schemes, the performance of both features are quite same. The LGMM approaches improved the accuracies with both EMHI and Diff features. The EMHI.LGMM out-performed than all the other schemes significantly, which indicates that the LGMM approach better fits the intrinsic structure of the data. The optimal numbers of Gaussian components found by BIC in EMHI.LGMM scheme are $m_0 = 67$, $m_1 = 24$ and $m_2 = 10$.

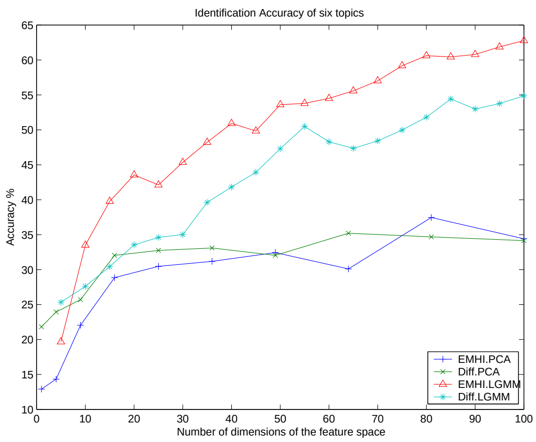

Figure 4. Accuracies of a six topics video shots classification using the proposed approach.

5.1 Acknowledgement

This research is supported by the National Science Foundation (NSF) under Grant No. IIS-0205219.

5.2 Conclusions

We have proposed a LGMM approach for exploiting high dimensional features for video shot classification. The LGMM performs as a non-liner method to reduce dimensions of the data with spatial lattice or temporal order constraints. We have shown the improvement of the proposed approach on top of the conventional PCA schemes. The proposed approach can be also applied on other high dimensional data with The layer stricture provides a feasible and quick (exponentially) way for dimension reduction by using more and more layers.

References

[1] J. Davis and A. Bobick. The representation and recognition of human movement using temporal templates. In *In Proc. Comp. Vis. and Pattern Rec.*, pages 928–934, 1997.

[2] I. Fodor. A survey of dimension reduction techniques. In *technical report UCRL-ID-148494, LLNL*, 2002.

[3] B. Scholkopf, C. Burges, and A. Smola. Advances in kernel methods: Support vector learning. In *MA: MIT Press, Cambridge*, 1999.

[4] J. Sivic, B. Russell, A. Efros, A. Zisserman, and W. Freeman. Discovering object categories in image collections. In *Proc. Int'l Conf. Computer Vision, Beijing*, 2005.