

---

# Recognizing Human Actions in Video Sequences

---

Goutham Patnaikuni  
gpanaiku@uci.edu

## Abstract

Recognizing human actions is a challenging problem in computer vision. In this work, a bag-of-words approach is taken to solving this problem. Each video sequence is represented as a document and each frame in a sequence corresponds to a "word". Although a bag-of-words model may not seem intuitive for data other than text, it has shown to be quite successful in object recognition problems in computer vision, and may prove to be a simple yet powerful model for recognizing human actions in video sequences. This particular model is probabilistic one trained in a semi-supervised way using a variant of the Latent Dirichlet Allocation model.

## 1 Introduction

The bag of words approach to human action recognition is motivated by some recent success in applying the same approach to object recognition problems in computer vision. In this particular model, each frame in a video sequence corresponds to a visual word and each video sequence corresponds to a "bag" of these words. It is important to note that the order in which the "words" occur in the video sequence does not matter and hence, some structure is lost by moving to this representation. On the plus side, this is a much simpler model than the ones that model temporal structure. Instead of capturing temporal structure, this model captures "temporal smoothing" via co-occurrence statistics among visual words. In particular, the model is based on the Latent Dirichlet Allocation (LDA)[2] model. One major issue with the LDA model is that it is not clear how to choose the right number of latent topics. Usually, this is done in an ad-hoc way i.e several different values are tried, but this is not a realistic approach. This issue is dealt with in two ways. First of all, the bag-of-words model for video sequences used here is much simpler than previous ones. Each frame corresponds to a "visual word" rather than a "collection of words" computed at spatial-temporal points of interest [3]. Secondly, some of the latent variables in the LDA model are observed in the training phase. This solves the problem of choosing the right number of latent topics. Section 2 describes the training and testing algorithms for the LDA model in detail. Experimental results are presented in Section 3 and conclusions in Section 4.

## 2 Approach

Video sequences are represented as a "bag of words". A "word" corresponds to a frame and a "document" corresponds to a video sequence in this representation. The model

is trained in a semi-supervised fashion using a variant of the LDA algorithm. First, a vocabulary of visual words, called a codebook is created. Using this codebook, a dataset of video sequences is converted to a bag of words representation which is then used to build a probabilistic model. The model is then used to classify video sequences. Algorithms are described in detail below.

## 2.1 Building a Codebook

The first step in moving toward a bag-of-words model is to build a vocabulary or codebook. For this, there needs to be a way to compare video frames so that frames that are "similar" can correspond to the same word. In this work, the motion descriptor in Efron et al.[4] is used. The first step is to track the human figures in video sequences. For this, the algorithm in Sabzmeydani and Mori [5][6] is used.

Given a video sequence centered around the human figure, the optical flow at each frame is computed using the Lucas-Kanade [7][9] algorithm. The optical flow field  $F$  is then split into  $F_x$  and  $F_y$  (flow fields corresponding to movement in the x direction and y direction respectively).  $F_x$  and  $F_y$  are further split into  $F_x^-$ ,  $F_x^+$ ,  $F_y^-$  and  $F_y^+$  such that  $F_x = F_x^+ - F_x^-$  and  $F_y = F_y^+ - F_y^-$ . These four non-negative channels are then blurred with a Gaussian kernel and normalized to obtain the final four channels  $Fb_x^-$ ,  $Fb_x^+$ ,  $Fb_y^-$  and  $Fb_y^+$ .

The motion descriptors of two frames are compared as follows: If the four channels for frame A are  $a_1, a_2, a_3$  and  $a_4$ , and the four channels for frame B are  $b_1, b_2, b_3$  and  $b_4$ , then the "similarity between frames A and B is:

$$S(A, B) = \sum_{c=1}^4 \sum_{x,y \in I} a_c(x, y) b_c(x, y) \quad (1)$$

where I are the indices of the motion descriptors.

Now, to build a codebook, a subset of all the frames are randomly selected and an affinity matrix  $A$  is computed for these frames where entry  $(i, j)$  is the affinity between frame  $i$  and frame  $j$ . Then k-medoid clustering is run on the affinity matrix  $A$ , obtaining  $V$  clusters. Words are then defined as the centers of the clusters. Then, all video sequences are converted to "documents" by replacing each frame with its corresponding word.

## 2.2 Latent Dirichlet Allocation

The model used in this work is based on the Latent Dirichlet Allocation (LDA) [7] model. The LDA model in the context of video sequences is described below. A dataset of videos is described as a collection  $D$  of video sequences  $(\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3, \dots, \mathbf{w}_N)$ . Each video sequence  $\mathbf{w}$  is represented as a collection of frames  $\mathbf{w} = (w_1, w_2, w_3, \dots, w_N)$ . A word  $w_i$  is an item from the codebook indexed by  $(1, 2, 3, \dots, V)$ . Assuming there are  $K$  underlying latent topics (action label classes), each topic is represented as a multinomial distribution over  $V$  motion words. A video sequence (represented by  $\theta$ ) is generated by sampling a mixture of the topics. LDA is a generative model and is described in Fig 1.:

The parameter  $\theta$  indicates the mixing proportion of different action labels in a video sequence.  $\alpha$  is the Dirichlet prior (a  $k$  element vector such that  $\forall i, \alpha_i > 0$ ) that controls how  $\theta$  varies among different video sequences.  $\beta$  is the parameter of a set of multinomial distributions indicating the the distribution of motion words within a action label. Learning the LDA model from a collection of video sequences  $D$  involves finding the  $\alpha$  and  $\beta$  that maximize the log likelihood of the data  $l(\alpha, \beta) = \sum_{d=1}^M \log P(\mathbf{w}_d | \alpha, \beta)$ .

1. Choose  $\theta$  from  $\text{Dir}(\alpha)$
2. For each of the  $N$  motion words  $w_n$ :
  - (a) Choose action label (topic)  $z_n$  from  $\text{Mult}(\theta)$
  - (b) Choose a motion word  $w_n$  from  $p(w_n|z_n, \beta)$ , a multinomial probability conditioned on  $z_n$

Figure 1: The generative LDA process

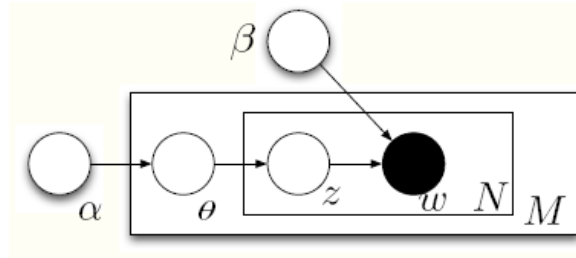


Figure 2: Representation of the LDA model

### 2.3 Semi-Latent Dirichlet Allocation

In LDA, the topic  $z_i$  for word  $w_i$  is not known and neither is the mixing proportion  $\theta$  for a document. In the action classification framework, all the frames in a training data set have class labels associated with them; each frame is labeled with the action label corresponding to the document that it belongs to. This suggests that there should be a direct correspondence between topics and class labels. Therefore, the topic of the word  $w_i$  in the video sequence  $\mathbf{w} = (w_1, w_2, w_3, \dots, w_N)$  will simply be the class label of  $w_i$  (i.e.  $z_i$ ). The representation of this model is shown in Fig 3.

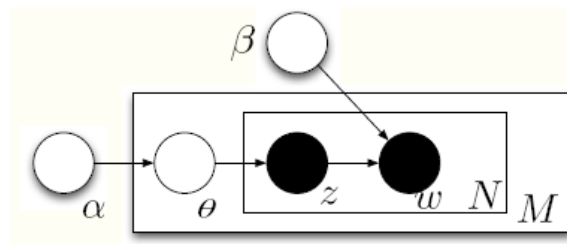


Figure 3: Representation of the S-LDA model

Notice that in Fig. 3, the topics  $z$  are observed. This makes the training phase much easier than the original LDA. In LDA, the parameters  $\alpha$  and  $\beta$  are coupled (conditioned on observed words  $\mathbf{w}$ ) and have to be estimated jointly. Several approaches (variational EM sampling etc.) are used to do this. In this case,  $\alpha$  and  $\beta$  are de-coupled and can be estimated separately. The parameter  $\beta$  is represented by a  $K \times V$  matrix, where  $K$  is the number of class labels and  $V$  is the number of words. Each row of the matrix  $\beta_i$ , must sum to 1. This means that the sum of the probabilities of generating all words in the vocabulary given a topic must sum to 1. Using maximum likelihood estimation,  $\beta_{ij} = n_{ij} / n_i$ , where  $n_i$  is the count of the  $i$ -th topic in the corpus and  $n_{ij}$  is the number of times the  $j$ -th word appears in the  $i$ -th topic.

To compute  $\theta$  for each document  $\mathbf{w} = (w_1, w_2, w_3, \dots, w_N)$ , assuming that the topics of the words in the document are  $\mathbf{z} = (z_1, z_2, z_3, \dots, z_N)$ , the  $i$ -th element of  $\theta$ ,  $\theta_i = |\{j : z_j = i, j = 1, 2, 3, \dots, N\}|/N$  i.e the number of words in the document that belong to topic  $z_i$ . After calculating  $\theta$ 's for all the documents in the dataset, a  $K \times M$  matrix  $\Theta = [\theta^1, \theta^2, \theta^3, \dots, \theta^M]$  can be computed where  $\theta^k$  is the  $\theta$  for the  $k$ -th document,

The parameter  $\alpha$  determines how the  $\theta$ s vary among documents (the  $\alpha$ s can essentially be thought of as pseudo-counts). The distribution  $p(\theta|\alpha)$  has the form  $\frac{\Gamma(\sum_k \alpha_k)}{\prod_k \Gamma(\alpha_k)} \prod_k \theta_k^{\alpha_k - 1}$ . This is the probability distribution function of the Dirichlet distribution.  $\alpha$  can be estimated from  $\Theta$  using a Newton Raphson algorithm [8]. Here, the  $\alpha$  parameter is set to a  $K$  dimensional vector of 1's.

## 2.4 Classification of Video Sequences

One of the advantages of the LDA model is that it provides a natural framework for well-defined inference procedures for previously unseen documents. The key problem that needs to be solved is that of computing the posterior distribution of the hidden variables given a document. Given a new, previously unseen video sequence  $\mathbf{w}$ , the task is now to classify each frame in the sequence. Assuming that the video sequence is represented by  $\mathbf{w} = (w_1, w_2, w_3, \dots, w_N)$ , the probability  $p(z_i|\mathbf{w}, \alpha, \beta)$  needs to be calculated. A frame  $w_i$  is classified to be action label  $k$  if  $k = \operatorname{argmax}_j p(z_i=j | \mathbf{w}, \alpha, \beta)$ . An interesting observation is that  $p(z_i|\mathbf{w})$  is calculated instead of  $p(z_i|w_i)$ . This means that the class label  $z_i$  depends not only on the word  $w_i$  but also depends on the entire video sequence  $\mathbf{w}$ .

A variational inference algorithm is used to calculate  $p(z_i|\mathbf{w}, \alpha, \beta)$ , proposed in Blei et al. [2]. The basic idea is to approximate the intractable distribution  $p(\theta, \mathbf{z}|\mathbf{w}, \alpha, \beta)$  by a simplified family of distributions  $q(\theta, \mathbf{z})$  where  $q(\theta, \mathbf{z}) = q(\theta|\gamma) \prod_{n=1}^N q(z_n|\phi_n)$ . The graphical representation of the distribution  $q(\theta, \mathbf{z}|\gamma, \phi)$  is shown in Fig. 4. Fig 4. is obtained from Fig.

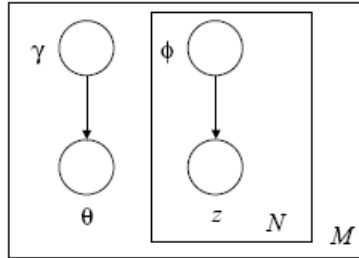


Figure 4: Representation of the variational distribution

2 by dropping the edges between  $\theta, \mathbf{z}$  and  $\mathbf{w}$  (which causes a coupling between  $\theta$  and  $\beta$ ) and the  $\mathbf{w}$  nodes.

In order to make the approximation as close to the original distribution as possible, the parameters  $\gamma^*$  and  $\phi^*$  that minimize the KL divergence between the distributions  $q(\theta, \mathbf{z}|\gamma, \phi)$  and  $p(\theta, \mathbf{z}|\mathbf{w}, \alpha, \beta)$  need to be estimated. The parameters  $(\gamma^*, \phi^*)$  can be found by iteratively updating  $(\gamma, \phi)$  using the variational inference algorithm in Fig. 5.

It should be noted that  $\gamma^*(\mathbf{w})$  and  $\phi^*(\mathbf{w})$  are document specific parameters. The  $i$ -th element of  $\gamma$  is approximately the  $i$ -th prior of the Dirichlet parameter  $\alpha_i$  plus the expected number of "words" generated by the  $i$ -th topic.  $\text{Dir}(\gamma^*(\mathbf{w}))$  is the Dirichlet distribution from which the mixing proportion  $\theta$  for the new video sequence is drawn [11]. The true mixing proportion  $\theta^*$  of the document can be approximated by taking the mean of a set

1. initialize  $\phi_{ni}^0 := 1/k \forall i, j$
2. initialize  $\gamma_i := \alpha_i + N/k \forall i$
3. **repeat**
4.     **for**  $n = 1$  **to**  $N$
5.         **for**  $i = 1$  **to**  $k$
6.              $\phi_{ni}^{t+1} := \beta_{iw_n} \exp(\Psi(\gamma_i^t))$
7.             normalize  $\phi_{ni}^{t+1}$  to sum to 1
8.              $\gamma^{t+1} := \alpha + \sum_{n=1}^N \phi_n^{t+1}$
9. **until convergence**

Figure 5: The variational inference algorithm

of samples drawn from  $\text{Dir}(\gamma^*(\mathbf{w}))$ . The  $\phi_n$  distributions ( $\phi_n$  is the  $n$ -th column of the  $\phi$  matrix, which is a  $K \times N$  matrix) approximate  $p(z_n|w_n)$ . Topic  $z_n$  is drawn from the distribution  $\text{Mult}(\theta^*)$ , which means  $\theta^*$  is an approximation of  $p(z_n)$ . Then  $p(z_n|\mathbf{w}) = \theta_{z_n}^* \phi_{z_n w_n}$ . This equation says that the class label  $z_n$  is based on  $\theta_{z_n}^*$ , the probability of generating topic  $z_n$  in the document and  $\phi_{z_n w_n}$ , the probability of generating topic  $z_n$  conditioned on word  $w_n$ . Alternatively, the  $\phi_n$  distribution can simply be read and the label of the  $n$ -th word can be determined by  $\max(\phi_n)$ .

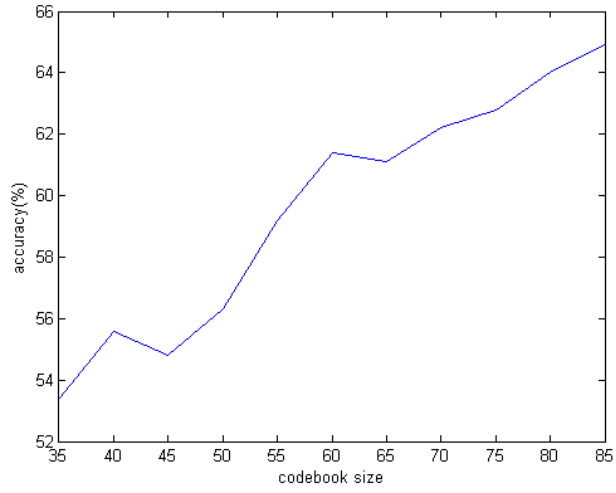
### 3 Experiments

The algorithm is tested on the KTH dataset [10] and results are show and discussed below.

#### 3.1 Results on the KTH Dataset

The KTH action dataset consists of six types of human actions (walking, jogging, running, boxing, hand waving and hand clapping) performed several times by 25 subjects in 4 different scenarios. The dataset contains 2391 video sequences. All sequences were taken over homogeneous backgrounds with a static camera with 25fps frame rate. For the purposes of this work, 300 of these video sequences (containing an equal number of videos in every class label) were chosen as a training set and 150 video sequences were chosen as the test set. First, a preprocessing step was run on all the video sequences to track the human figures appearing in them, using the algorithm in Sabzmeydani and Mori [5][6]. Optical flow was computed and the motion descriptor discussed in Sec. 2.1 was used to build an affinity matrix by choosing a small number of frames randomly from each video sequence.  $K$ -medoid clustering was then run on the affinity matrix to obtain clusters corresponding to words. The S-LDA algorithm was then performed on the bag of words version of the training set. Cross validation was performed on the training set with a hold out set of 60 video sequences. Fig 6(a) is a codebook size vs accuracy plot using the held out set. In the experiments performed, the accuracy obtained by the model when using 85 codewords was shown to be the highest. The corresponding confusion matrix when classifying videos in the test set is shown in Fig 7. As seen in the figure, the algorithm classifies most of the actions correctly. The "jogging" action seems to have the lowest accuracy rate (51%) followed by running (55%). The algorithm often mistakes these actions for one another. This is reasonable considering that the two actions are alike. Fig 6(b). below further illustrates this point.

From Fig 7., it is apparent that the "boxing" action has the highest accuracy rate. There is also some confusion between the "handwaving" and "handclapping" actions. This makes sense because in many cases, both actions start out the same way i.e with the hands of the subject stretched out by his/her side, with some of the following initial movements being



(a)



(b)

Figure 6: (a) Accuracy vs codebook size; (b) Frames from the original KTH dataset. The frames in the top row are from a jogging sequence, the frames in the bottom row from a running sequence

<b>ACTIONS</b>	<i>boxing</i>	<i>handclapping</i>	<i>handwaving</i>	<i>jogging</i>	<i>running</i>	<i>walking</i>
<i>boxing</i>	0.91	0.01	0.01	0.02	0.02	0.03
<i>handclapping</i>	0.00	0.84	0.14	0.01	0.00	0.01
<i>handwaving</i>	0.01	0.11	0.88	0.00	0.00	0.00
<i>jogging</i>	0.00	0.01	0.01	0.51	0.33	0.14
<i>running</i>	0.03	0.04	0.01	0.31	0.55	0.06
<i>walking</i>	0.02	0.01	0.00	0.06	0.01	0.90

Figure 7: Confusion matrix using 85 codewords. Horizontal rows represent true labels and vertical columns represent predictions.

similar. The overall accuracy of the S-LDA model is 76.5%.

## 4 Conclusion and Future Work

In this project, a new approach was taken toward the problem of human action recognition. Given that spatial and temporal information is lost, the bag of words approach for image and video data may not seem intuitive at first, but in this case, it does perform reasonably well. The reason for this is not very apparent and it is certainly conceivable that a more complex model that is based on temporal data will perform better, but the moderate success achieved here indicates that the bag of words approach is certainly worth further exploration.

One of the limitations of the method discussed here is that it requires a significant pre-processing stage of tracking human figures. This is reasonable for the training phase but it is not realistic to assume that given a new video to classify, it is "clean" enough that the human figures in it can be accurately tracked. For example, the new video may have a lot of noise in it. Even if it were the case that the human figures could be tracked, a tracking algorithm may be computationally expensive. Ideally, one would like to classify a new video as in a computationally less expensive way.

This is an ongoing project and the next step is to try and recognize human actions in video sequences where more complex actions take place (a KTH video sequence consists of only a simple action being performed by a single human subject). For this purpose, a dataset consisting of YouTube videos was collected. These videos range from political speeches to concert videos. Other categories include music, tennis, soccer and football and amateur videos. A potential problem to work on would be to devise methods to recognize the actions within a new, yet to be classified video sequence without explicitly having to track human figures. The bag-of-words approach may be tried on the YouTube dataset but other temporal based models may also be investigated.

## References

- [1] Wang, Y., Sabzmeydani, P. & Mori G (2007) Semi-Latent Dirichlet Allocation: A Hierarchical Model for Human Action. In IEEE International Conference on Computer Vision Recognition.
- [2] Blei, D.M., Ng, A.Y. & Jordan, M.I. (2003) Latent dirichlet allocation. Journal of Machine Learning Research.
- [3] Juan C.N., Hongcheng W. & Fei-Fei L. (2006) Unsupervised Learning of Human Action Categories Using Spatial-Temporal Words. British Machine Vision Conference.
- [4] Efros, A.A., Berg, A.C., Mori, G. & Malik, J. (2003) Recognizing action at a distance. In IEEE International Conference on Computer Vision.
- [5] Sabzmeydani, P. & Mori, G. (2003) Detecting pedestrians by learning shapelet features. In IEEE Conference on Computer Vision and Pattern Recognition.

- [6] Sabzmeydani,P. & Mori,G. (2003) Detecting pedestrians by learning shapelet features. Software retrieved from the Simon Fraser University website <http://www.cs.sfu.ca/mori/research/>. *Note: Software no longer available.*
- [7] Lucas,B.D. & Kanade,T. (1981) An iterative image registration technique with an application to stereo vision. In Proceedings of the DARPA Image Understanding Workshop.
- [8] Minka,T.P. (2000) Estimating a Dirichlet distribution. Massachusetts Institute of Technology (2000).
- [9] Khan. S. Software retrieved from website <http://www.cs.ucf.edu/khan/>.
- [10] Schuldt C., Laptev I. & Caputo B. (2004) Recognizing Human Actions: A Local SVM Approach. International Conference on Pattern Recognition.
- [11] Minka, T.P. (2006) The Fastfit Matlab toolbox. Software retrieved from Microsoft research website <http://research.microsoft.com/minka/software/fastfit/>