Embedding Visual Words into Concept Space for Action and Scene Recognition

Behrouz Saghafi
Behr0002@e.ntu.edu.sg
Elahe Farahzadeh
Elah0001@e.ntu.edu.sg
Deepu Rajan
ASDRAJAN@ntu.edu.sg
Andrzej Sluzek
ASSLUZEK@ntu.edu.sg

School of Computer Engineering
Nanyang Technological University
Singapore

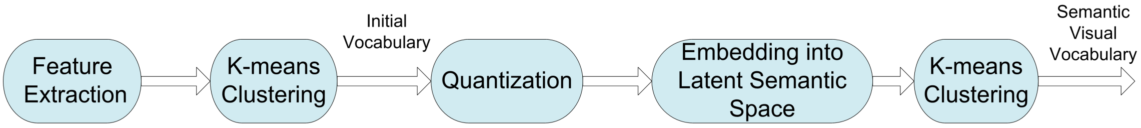

Figure 1: Constructing the semantic visual vocabulary.

In this paper we propose a novel approach to introducing semantic relations into the *bag-of-words* framework. We use the latent semantic models, such as LSA and pLSA, in order to define semantically-rich features and embed the visual features into a semantic space. The semantic features used in LSA technique are derived from the low-rank approximation of word-document occurrence matrix by SVD. Similarly, by using the pLSA approach, the topic-specific distributions of words can be considered dimensions of a concept space.

In the proposed space, the distances between words represent the semantic distances which are used for constructing a discriminative and semantically meaningful vocabulary. Figure 1 shows the flowchart for constructing the semantic visual vocabulary via embedding into concept space. We first extract features from patches (cuboids) in the images (videos). The initial vocabulary is constructed by performing k-means clustering on the extracted features and choosing the cluster centers as the codewords. The feature vectors are quantized based on the initial codebook to form the word-image (or video) matrix which describes the occurrences of words in images/videos. The codewords are then embedded into the concept space by latent semantic models - we demonstrate the embedding both by LSA as well as by pLSA.

LSA [1] finds a low-rank approximation for the word-document matrix. Here, document refers to the image or video sequence. Let X be the occurrence matrix whose rows correspond to words and columns correspond to documents. Decomposing X using SVD as $X = U\Sigma \mathrm{V}^T$ gives the orthogonal matrices U and V and the diagonal matrix Σ. Selecting the L largest singular values and their corresponding singular vectors, we find the rank-L approximation of X by $X \approx U_L\Sigma_L{V_L}^T$. The column vectors of U_L span the concept space of words and the columns of V_L span the concept space of documents.

pLSA [3] is the statistical version of LSA which defines a generative model on the data. It is assumed that there is a latent topic variable z_l associated with occurrence of the word w_i in the document d_j. The observed variables are w_i and d_j while z_l is latent. The probability of observation pair $P(w_i,d_j)$ is $P(w_i,d_j) = P(w_i|d_j)P(d_j)$. Since the occurrence of w_i and d_j is assumed to be independent, we can marginalize over latent topics z_l in order to find the conditional probability $P(w_i|d_j)$, i.e.,

$$P(w_i|d_j) = \sum_{l=1}^{L} P(w_i|z_l)P(z_l|d_j), \tag{1}$$

where L is the total number of latent topics. Equation (1) is a decomposition of the word-document matrix, similar to LSA, but with the condition that the values are normalized to be probability distributions. We fit the model by determining $P(z_l|d_j)$ and $P(w_i|z_l)$ given the observation occurrence matrix. Maximum likelihood estimation of the parameters is performed using Expectation Maximization (EM) algorithm. Assuming a vocabulary of M words and N documents, the likelihood function to be maximized is $\prod_{i=1}^{M}\prod_{j=1}^{N} P(w_i|d_j)^{n(w_i,d_j)}$, where $n(w_i,d_j)$ is the number of words w_i in the document d_j and $P(w_i|d_j)$ is obtained by equation (1). The original pLSA algorithm in the unsupervised learning framework tries to categorize the query document given the learned parameters [3]. However, we use the pLSA algorithm only to determine the probabilities $P(w_i|z_l)$. In fact $P(w_i|z_l)$ is equivalent to the l^{th} dimension of $\hat{t}_i$ in the LSA framework. Thus, using pLSA we obtain the concept space embedded vector $\hat{t}_i$ as:

$$\hat{t}_i = \begin{bmatrix} p(w_i|z_1) & p(w_i|z_2) & \dots & p(w_i|z_L) \end{bmatrix}^T. \tag{2}$$

We have tested our approach on the KTH action database and on the fifteen scene database and have achieved very promising results on both.We choose the size of the initial codebook as 1500 for both action and scene experiments.

For action recognition, we use the interest point detector proposed by Dollar et al.[2] The interest points are detected based on the local maxima of a response function, which incorporates a Gaussian kernel in the spatial domain and a Gabor filter in the temporal domain. Cuboids are extracted around each interest point. The cuboids are described by flattened gradients. The dimension of the descriptors are reduced to 100 using PCA to obtain the final feature vectors.For scene recognition, dense features are more discriminative than sparse ones. Accordingly, we use dense features sampled using regular grid with space of 8 pixels. SIFT descriptors of 16×16 patches are used on the grid.

One of the advantages of the proposed method is that it allows the number of topics to be varied, in contrast to pLSA using unsupervised framework where the number of topics is constrained to be the same as the number of classe, We have investigated the influence of number of topics L on the recognition accuracy As the number of topics is increased from L = 6, which is the number of classes, the recognition rate increases since the increased number of concepts enables better discrimination between topics. However, after around L = 30 topics, the recognition accuracy decreases. This is mainly because adding more dimensions to the concept space implies further division into semantic units that are not meaningful.This phenomenon occurs at L = 50 for pLSA. We have obtained the best result of **93.94%** accuracy using pLSA model and the semantic codebook size of 600, without any spatial or temporal information.

Similar to results of action recognition, the scene recognition accuracy improves with increasing number of topics until L = 70 after which, there is a deterioration in the accuracy.The best result achieved on fifteen scene dataset with our method is **79.22%** using pLSA model and the semantic codebook size of 600.

[1] S. Deerwester, S.-T. Dumais, G.-W. Furnas, T.-K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990.

[2] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *VS-PETS*, 2005.

[3] T. Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning*, 42(1-2):177–196, 2001.