Unsupervised Parameter Selection for Gesture Recognition with Vector Quantization and Hidden Markov Models

Przemysław Głomb, Michał Romaszewski,
Arkadiusz Sochan, and Sebastian Opozda

Institute of Theoretical and Applied Informatics of Polish Academy of Sciences,
Bałtycka 5, 44-100 Gliwice, Poland
przemg@iitis.pl

Abstract. This article presents an investigation of a heuristic approach for unsupervised parameter selection for gesture recognition system based on Vector Quantization (VQ) and Hidden Markov Model (HMM). The two stage algorithm which uses histograms of distance measurements is proposed and tested on a database of natural gestures recorded with motion capture glove. Presented method allows unsupervised estimation of parameters of a recognition system, given example gesture recordings, with savings in computation time and improved performance in comparison to exhaustive parameter search.

Keywords: Gesture recognition, Vector Quantization, Hidden Markov Model, automatic parameter selection.

1 Introduction

Human-computer interfaces (HCI) using natural human gestures are one of the promising new approaches in interaction design. The important component of a gesture interface is a recognition system for time series measurements of data (motion capture readings or image derived features). For this task, one popular and reported as effective approach is the combination of Vector Quantization (VQ) and Hidden Markov Model (HMM) [1,2]. While effective, this method needs to be provided with several parameters, notably number of VQ symbols, number of HMM states, and initial choice for HMM matrices. Those parameters are often chosen ad hoc or by trial and error, with random selection of initial conditions. Such approach is time consuming and can lead to suboptimal performance [3,4]. On the other hand, the performance of input recognition largely determines the usability of a HCI system.

Our approach and main contribution of this work is the investigation of a two step, heuristic algorithm for VQ+HMM parameter selection: first, computing the number of VQ symbols by considering class separation, followed by iterative hierarchical clustering of subsequences for finding initial HMM matrices. The standard model reestimation (Baum-Welch algorithm) and selection (Akaike/Bayesian information criterion or others) can be applied afterwards for completing the parameter selection. This algorithm allows for deterministic and unsupervised estimation of parameters of

P. Campos et al. (Eds.): INTERACT 2011, Part IV, LNCS 6949, pp. 170–177, 2011.

a recognition system, given example set of gesture recordings, with savings in computation time and improved performance in comparison to exhaustive or random parameter search.

1.1 Related Work

The choice of parameters is an integral element of a development of a VQ+HMM system, and as such has been a subject of active research. In [11], one of the most popular Vector Quantization algorithms was introduced, named after authors' initials as LBG. By an iterative process of splitting and optimizing cluster centers, it finds optimal VQ dictionary vectors with respect to the distortion measure. This approach produces good results for coding applications, but does not consider the discriminative power of the sequences of symbols for the classification task. The latter problem can be related to estimation of class separation and within/between class variability as considered in Linear Discriminant Analysis (LDA). The problem of parameterization of VQ for optimal class separation has been considered in [12] using Dynamic Time Warping (DTW) and Learned Vector Quantization with the objective of speech recognition with Self Organizing Maps. The similarity of symbol sequences with application for recognition has also been the subject of [13], where DTW was applied to sequences after VQ. The problem of optimization of VQ centroids has been addressed in [14], with application of Bhattacharyya distance.

Our approach builds on above works, focusing on maximization of the Bhattacharyya measure of within-class and between-class distance distributions, obtained using Levenshtein symbol distance, a technique conceptually similar to DTW.

The other problem with design of VQ+HMM recognizer, apart from the above, is the parameterization of a HMM. [1] note the importance of good initial HMM values, as the reestimation algorithm (Baum-Welch) finds local, not global maximum of likelihood function; a procedure is thus proposed for improving the initial values using K-means clustering of data vectors. The clustering can be also used to identify the number of HMMs needed for representation, as in [8] where DTW and hierarchical clustering is used to obtain a set of HMMs describing a set of human motion trajectories. In [2] authors optimize the performance of a HMM by merging states based on Kullback-Leibler similarity measure of symbol emission matrices, however, this is done after training, with the chief objective of reducing number of states of a HMM constructed as a sum of several others. In [5], several model selection strategies are investigated, including sequential state pruning strategy, where least probable states are iteratively deleted from the model. While effective, deletion could possibly lead to a loss of information in the final model. [6] perform an in-depth analysis and argue for using cross-validated likelihood to assess the number of states.

Our approach considers estimation of initial HMM parameters, which are to be improved with Baum-Welch optimization. We split the sequences into fragments and successively merge them with agglomerative hierarchical clustering based on the Bhattacharyya distance of distributions of symbol occurrences. At each moment, the clustered fragments can be translated into HMM initial parameters.

This article is organized as follows: section two contains the method description, section three experimental results with motion capture database, last section presents conclusions.

2 Method

We focus on the problem where the user interface (UI) designer has the following:

1. An acquisition system that captures human motion as a time sequence of vectors $X=(\boldsymbol{x}_t)$, where each $x_t \in \mathbb{R}^l$ represents the data from l sensors (finger bend, hand rotation, and similar when using motion capture device) or features (such as trajectory features, shape moments when working with images of motions).
2. A set of c gestures (defined motions) selected to be used in the interface.

The input sequence X is classified into one of the c gestures using VQ+HMM framework as follows. First, a quantization function

$$Q: \mathbb{R}^l \rightarrow \mathbb{Z}_m \quad x_t \rightarrow s_t \tag{1}$$

is used, transforming a sequence of vectors X into a sequence of discrete symbols $S=(s_t)$. Then, a set of HMM models corresponding to the gestures $\lambda_1,\ldots,\lambda_c$ is used to compute the estimate of the sequence being generated with each model. Either the forward or Vitterbi algorithm can be used for this purpose [1]. By comparing the result probabilities the class of the input can be derived.

Implementation of this approach requires definition of the VQ and HMM stages. For VQ a quantization algorithm must be selected, for example the Linde-Buzo-Gray (LBG) method [11]. Given a set of reference data, the algorithm can calculate the m reference ("dictionary") vectors. The number of VQ symbols m (or related parameter, like target distortion ε), however, has to be explicitly provided. For HMM, the defining stochastic matrices (starting state probability vector Π, state to state transmission probabilities A and symbol emission probabilities B) must be given. The HMM matrices are commonly estimated with Baum-Welch method [1], but this requires definition of their initial values and implicitly providing the number of states n.

2.1 Choosing the Number of Vector Quantization Symbols

The objective of this stage is to derive the number of VQ symbols m. The proposed method follows from the observation that for any two recorded symbol sequences S_1 and S_2 we can use approximate string matching to compute the distance $d_S(S_1,S_2)$ between them. The use of approximate methods is necessary, as subsequent recordings of the same gesture usually differ in both length and content of the symbol sequence, resulting from variations of movement (speed, hand configuration and so on). The nature of those differences can be formalized as symbol addition, deletion and replacement, thus Levenshtein distance is particularly fitting for this purpose.

Given a set of c gesture classes with k realizations of each class we can compare them both within their classes (by computing $d_S(S_1,S_2)$ where S_1 and S_2 belong to the same class) and between them (where each of S_1 and S_2 belong to a different class).

We can compute the histograms h_W of within-class d_S and h_B of between-class d_S. After normalizing the histograms, we can compute Bhattacharyya distance as

$$d_B(h_B, h_W) = -\ln \sum_{i \in I} \sqrt{h_B(i) h_W(i)}. \tag{2}$$

This distance has been selected as it is reported to have good properties [9]. We can make the following observations concerning its behavior in this context:

1. d_B can be treated as a measure of separation of the two histograms. In other words, it can be used to estimate the compactness of the clusters formed by realizations of each gesture. Good separation (preferable situation, with high d_B value) occurs when, on average, the distance between gestures sampled from one class is substantially smaller than to other classes. Poor separation occurs when the average distance within each class is comparable to other classes. In the second case the d_B value is low, and there can be expected a loss of performance due to misclassification of gestures.
2. d_B changes with m, as the number of symbols together with reference data define the quantization function, which produces symbol strings.
3. With a large value of m (large number of symbols), a situation can arise where a number of symbols will appear only once in single realizations (sequences). Hence the distances between sequences within-class will be large and comparable to those between-class, and d_B value will be low. This phenomenon is akin to overtraining of VQ.
4. With a small value of m, the pool of symbols will be small, hence all sequences will be similar (in terms of distance function). This can be viewed as "undertraining", and the result can be expected to be similar to above – distances comparable (although on average lower), and low d_B value.

Based on the observations, it seems fitting to observe the $d_B(m)$ and find its maximum, corresponding to best separation between the classes. While not formally proved, a number of experiments with different equipment and gesture sets confirm the validity of the above observations.

2.2 Choosing the Initial Parameters of HMM

For training the HMM, the Baum-Welch (BW) method is commonly used. It is noted [1,3,4], that the local optimization approach used in BW requires a good estimate of initial conditions, in particular of the symbol emission matrix B. One approach is to choose the number of states n with trial and error, training HMM for each choice with a number of randomized B candidates and choosing the combination providing maximum performance. This approach is essentially a random search in parameter space, and can be time consuming with a large number of models and/or states of each model, as the number of free parameters to be estimated is on the order of $O(n^2)$.

We propose to obtain initial estimates for HMM matrices for a given class using a hierarchical clustering of data sequences with the following steps:

1. Split the k training sequences into the l fragments each. Index the fragments as (t,i), $t=1,\dots,l, i=1\dots,k$ and compute the distribution of symbols in each fragment.
2. Generate candidate initial HMM (see below).

3. Perform reestimation and assess current performance (for example, empirical likelihood or Akaike information criterion [10])
4. Merge step: compute Bhattacharyya distances of distributions of symbols in fragments (a) for given *t* (b) between set of fragments at *t* and *t+1*. Merge the fragments (or sets) with smallest distance (most similar).
5. Repeat the process (goto 2) until either the desired performance target is satisfied, or until number of fragments is 1.

The initial HMM generation in step (2) is done as follows. Number of states is equal to the number of fragments. Matrix B is set with symbol distributions for each fragment. Matrix A and vector Π are initialized with left-to-right structure (with A matrix structured so that from each state i only the transfer to $j \geq i$ is possible). However, if, for given *t*, there is more than one fragment, it is assumed that different exclusive symbol distributions can appear (for example users will make the gesture with one of several hand configurations). This exclusivity is reflected in the state transition matrix structure: a state created from fragment *(t,i)* has non-zero probability of transfer only to itself and to all states *(t+1,·)*, while having zero probability of transfer to all others, including *(t,j), j≠i*.

3 Experiments and Results

We have performed experiments to observe and confirm validity of the proposed method. The data set used consists of motion capture recordings of 22 natural gestures ("A-OK", "thumbs up", etc) with four participants and five repetitions of each gesture per person (with different speeds: three times normal speed, one fast, one slow execution). The equipment used was DG5VHand motion capture glove, containing five resistance type finger bend sensors and three axis accelerometer producing three

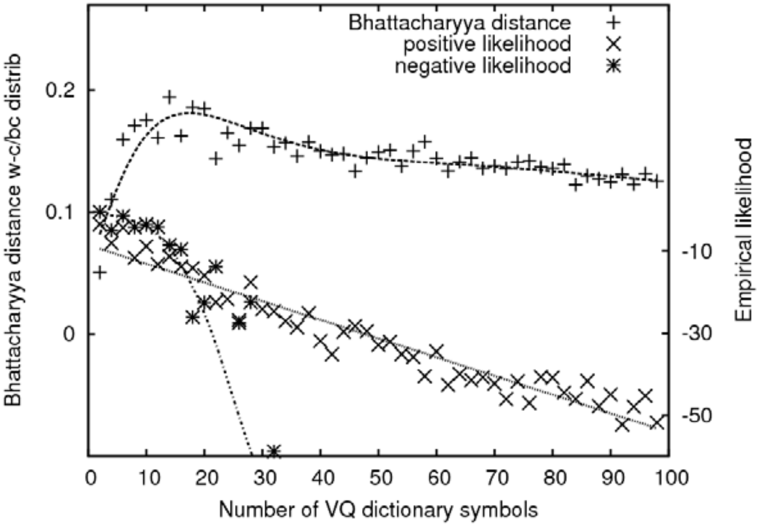

Fig. 1. Values of Bhattacharyya distance for different numbers of symbols, and log likelihoods: positive (of the class that model was trained on) and negative (of other classes)

acceleration and two orientation readings. The sampling frequency was ~30Hz. The movements used were a mix of natural, real-life gestures of several types (symbolic, iconic, deictic and manipulative), selected and captured with an objective on creating a diverse, rich testing set for natural gesture recognition methods. The length of recorded sequences was between 30 and 190 vectors. The experiment software was written in C++ using several publicly available numerical algebra libraries. For details on the creation of database and gesture choice, refer to [7].

For VQ dictionary the K-means algorithm was used with Mahalanobis distance (the covariance matrix was estimated from data). The range of symbols considered was m=2,...,200. The observed histograms of Levenshtein distance show noticeable overlap, mainly due to the fact that, with different execution speeds, the sequences for the same gesture differ in length (by 30-40% between fast/normal or normal/slow speeds). While Bhattacharyya distance estimates are noisy, the tendency to peak at m~18 can be seen (fig. 1). The behavior of HMMs for different number of symbols was then observed. First, one HMM was trained for each gesture (with number of states $n=20$; models chosen from l=10 candidates; left-to-right topology with uniform initialization for matrix A and vector Π; B matrix randomized with simplex method). Then, the log likelihood was computed – for given model i, of the true class i (positive) and $j{\neq}i$ (negative). As noted before, with small number of symbols, all sequences are similar, even across different gestures (small inter-class variation, hence high negative likelihood); with large number of symbols, all gesture realizations are dissimilar, even for the same gesture (large intra-class variation, decrease of positive likelihood). Proposed method selects maximum of Bhattacharyya distance, which corresponds roughly to the area of decline of negative likelihood.

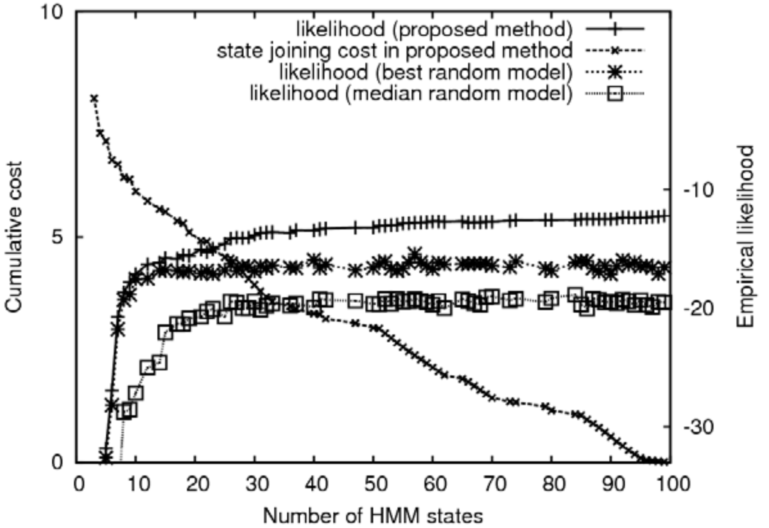

Fig. 2. Comparison of log likelihoods with proposed method and random initialization, with the cumulative cost of merging (sum of Bhattacharyya distances for subsequent fragment joining)

The second set of experiments was carried out to assess the performance of the fragment clustering method. For each number of states, one HMM was initialized with proposed method, and *l=100* randomly (with the same method as above). The log likelihoods of proposed method, minimum (best value) and median of random models are presented on fig 2. The improvement is increasing with number of states. The likelihood increase when *n>30* is not substantial, this can be used as a cue for *n* choice; for exact value, one can rely on selection methods such as [6].

4 Conclusions

We have presented an approach for unsupervised parameter selection of VQ+HMM classifier and experiments validating the algorithm on a real dataset of natural gestures. The main advantage of its application is the possibility to investigate the parameter space and find the best performing values with cost of several scales of magnitude smaller than exhaustive search. As the VQ+HMM is very common in the role of time-varying sequence classification, this approach has potentially wide range of applications, for example motion capture or camera based HCI systems employing gesture recognition.

Acknowledments. This work has been partially supported by a Polish Ministry of Science and Higher Education project NN516405137 “User interface based on natural gestures for exploration of virtual 3D spaces”.

References

1. Rabiner, L.R.: A tutorial on Hidden Markov Models and selected applications in speech recognition. Proceedings of the IEEE 77(2), 257–286 (1989)
2. Lee, H.-K., Kim, J.H.: An HMM-Based Threshold Model Approach for Gesture Recognition. IEEE Trans. Pattern Analysis and Machine Intelligence 21(10), 961–973 (2002)
3. Brand, M., Oliver, N., Pentland, A.: Coupled hidden Markov models for complex action recognition. In: Proc. Computer Vision and Pattern Recognition (CVPR), pp. 994–999 (1997)
4. Nathan, K., Senior, A., Subrahmonia, J.: Initialization of hidden Markov models for unconstrained on-line handwriting recognition. In: IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP), vol. 6, pp. 3502–3505 (1996)
5. Bicego, M., Murino, V., Figueiredo, M.A.T.: A sequential pruning strategy for the selection of the number of states in hidden Markov models. Pattern Recognition Letters 24(9-10), 1395–1407 (2003)
6. Celeux, G., Durand, J.-B.: Selecting hidden Markov model state number with cross-validated likelihood. Computational Statistics 23(4), 541–564 (2008)
7. Głomb, P., Romaszewski, M., Opozda, S., Sochan, A.: Choosing and modeling hand gesture database for natural user interface. In: Proc. of GW 2011: The 9th International Gesture Workshop Gesture in Embodied Communication and Human-Computer Interaction (2011)
8. Ouivirach, K., Dailey, M.N.: Clustering human behaviors with dynamic time warping and hidden Markov models for a video surveillance system. In: International Conference onElectrical Engineering/Electronics Computer Telecommunications and Information Technology (ECTI-CON), pp. 884–888 (2010)

9. Kailath, T.: The Divergence and Bhattacharyya Distance Measures in Signal Selection. IEEE Trans. on Communication Technology 15(1), 52–60 (1967)
10. Akaike, H.: A new look at the statistical model identification. IEEE Trans. on Automatic Control 19(6), 716–723 (1974)
11. Linde, Y., Buzo, A., Gray, R.: An Algorithm for Vector Quantizer Design. IEEE Transactions on Communications 28(1), 84–95 (1980)
12. Somervuo, P., Kohonen, T.: Self-Organizing Maps and Learning Vector Quantization for Feature Sequences. Neural Processing Letters 10(2), 151–159 (2004)
13. Martin, M., Maycock, J., Schmidt, F.P., Kramer, O.: Recognition of Manual Actions Using Vector Quantization and Dynamic Time Warping. In: Graña Romay, M., Corchado, E., Garcia Sebastian, M.T. (eds.) HAIS 2010. LNCS, vol. 6076, pp. 221–228. Springer, Heidelberg (2010)
14. Rigazio, L., Tsakam, B., Junqua, J.-C.: An optimal Bhattacharyya centroid algorithm for Gaussian clustering with applications in automatic speech recognition. In: Proc. ICASSP 2000 IEEE International Conf. on Acoustics, Speech, and Signal Processing, vol. 3, pp. 1599–1602 (2000)