DTW Based Clustering to Improve Hand Gesture Recognition

Cem Keskin, Ali Taylan Cemgil, and Lale Akarun

Bogazici University, Computer Engineering Department
keskinc@cmpe.boun.edu.tr, {taylan.cemgil,akarun}@boun.edu.tr

Abstract. Vision based hand gesture recognition systems track the hands and extract their spatial trajectory and shape information, which are then classified with machine learning methods. In this work, we propose a dynamic time warping (DTW) based pre-clustering technique to significantly improve hand gesture recognition accuracy of various graphical models used in the human computer interaction (HCI) literature. A dataset of 1200 samples consisting of the ten digits written in the air by 12 people is used to show the efficiency of the method. Hidden Markov model (HMM), input-output HMM (IOHMM), hidden conditional random field (HCRF) and explicit duration model (EDM), which is a type of hidden semi Markov model (HSMM) are trained on the raw dataset and the clustered dataset. Optimal model complexities and recognition accuracies of each model for both cases are compared. Experiments show that the recognition rates undergo substantial improvement, reaching perfect accuracy for most of the models, and the optimal model complexities are significantly reduced.[1]

Keywords: Dynamic time warping, DTW, hand gesture recognition, HMM, IOHMM, HSMM, HCRF, preprocessing, time series clustering.

1 Introduction

Vision based hand gesture recognition has been used in the last decade as a natural interface for a variety of applications, such as games, virtual reality and modeling tools. However, using hand gestures as an input to HCI systems is challenging due to the inherent sensor noise. The impact of illumination conditions on the image, the difficulty of segmenting the hand from a cluttered background, and the cumbersome procedures for calibration of multiple cameras and other sensors, have limited the spread of vision based hand gesture interaction.

The recent release of infrared equipped depth sensors such as Kinect has accelerated the use of hand gestures for HCI, since such depth sensors can be used to segment the hand from cluttered backgrounds. Moreover, Kinect works by emitting and sensing infrared structured light, and does not depend on illumination conditions. Hence, hands can be easily detected, segmented and tracked in real time.

[1] This work has been supported by research grants Tubitak 108E161 and BU-BAP 09M101.

A.A. Salah and B. Lepri (Eds.): HBU 2011, LNCS 7065, pp. 72–81, 2011.

In HCI applications, sensors detect the gesture signals by retrieving images of the hand while a gesture is performed. Features describing the shape and motion of the hand are extracted from each frame, forming a vector–valued time series, which we call a gesture sample. Gesture recognition is performed through classification of these gesture samples in real time. From a machine learning point of view, hand gestures can be considered as the output of partially observable stochastic processes [10]. Hence, the majority of related studies use graphical models such as the HMM for this task. These models have been traditionally compared in terms of their gesture classification accuracies. However, their classification speeds are also important, as the target applications are almost always meant to run in real–time.

1.1 Graphical Models for Hand Gesture Recognition

A hand gesture is generated by the hand as it assumes certain shapes while moving on a predefined trajectory. Sensors supply partial observations from this process. Both generative and discriminative graphical models have been employed for hand gesture recognition based on these observations. Generative models learn the joint distribution of their latent variables and the observations, and thus, they can produce new samples belonging to a gesture class by sampling from this distribution. On the other hand, discriminative models condition their hidden states on a suitable function of observations, and learn to distinguish between different gesture classes.

The most basic generative graphical model is the HMM [11]. The ability of generative models to generate samples is not required for classification. Instead, Markov random fields can be used to attack the problem by directly modeling the probability of model parameters conditioned on observation features. The simplest type of Markov random field is the conditional random field (CRF), which is the discriminative counterpart of HMM [4].

CRFs are not suitable for sequence classification tasks, since they associate a class label with each frame instead of the entire sequence. For such tasks, a CRF variant called hidden CRF (HCRF) is used, that incorporates a single class label with a sequence [13]. This is achieved by adding a new variable for the class label that is connected to all of the hidden state variables of the graph.

The input–output HMM (IOHMM) is an HMM variant, which conditions model parameters on an external input sequence [1]. This sequence is used to estimate the HMM parameters at each time frame. The sequence can contain any information that is known to be correlated with observations and state transitions, i.e., regime changes in the data. HMM parameter estimation is done through common regression methods such as artificial neural networks or radial basis functions.

HSMM is a natural extension to HMM, where each state produces a sequence of observations instead of a single item. These segments can be generated using a variety of methods, such as using counters to keep track of the number of symbols or employing local HMMs that produce subsequences. Explicit duration model (EDM) is a type of HSMM, where state visit durations and sojourn times are explicitly modeled [14]. Hence, EDM can be interpreted as a special case of HMMs, where the state variable is augmented by a counter variable that keeps track of the time that the process has spent in a given state. Likewise, an HMM is an EDM with durations set to a single frame.

Some gestures are subject to spatio–temporal variability, i.e., the exact starting point, speed or scale of the gesture do not change their meaning. Speed, scale and sampling rate have a direct effect on the gesture sample lengths. Graphical models need to take the variance of sample length into account, usually by modeling durations at each hidden state. Nevertheless, as long as there are no alternative trajectories or hand shapes for a gesture class, the model can have a *left–right architecture*, which has considerably lower complexity than an unrestricted model and a lower evaluation time complexity. A common example is the left–right HMM [5], which is also extensively used for speech recognition.

To analyze and compare different graphical models, we use a challenging dataset in the sense that it does not conform to the assumptions of a left–right architecture. This dataset is created from the ten digits written in the air by several users, and captured by Kinect. There are no universally accepted trajectories for drawing digits, and different gesturers are likely to follow different paths; e.g.,the digit zero can be drawn clockwise or counter–clockwise. Likewise, the changes in speed along the path do not change the meaning of a digit. Due to these additional challenges, a left–right architecture cannot be directly assumed.

In this work, we compare HMMs, HCRFs, IOHMMs, and EDMs on the basis of recognition accuracy and speed using the digit dataset. We show that the models need to be more complex (i.e., need more hidden states) and are not restricted to have a left–right architecture, due to the complexity of the dataset. We propose a a preprocessing method, which eliminates the need for more complex models by transforming the dataset. This new dataset is formed by rescaling, resampling and clustering of gesture samples.

1.2 Clustering Time Series

Clustering of time series has been shown to be effective in many application domains [6]. The goal of clustering is to identify sets of samples that form homogeneous groups, in the sense that a certain distance measure, such as Euclidean distance for static data, is minimized among the samples in the formed clusters. Thus, a direct benefit of clustering the dataset is that modeling clusters is easier than modeling the original classes. For instance, such clusters can be modeled using simple graphical models with left–right architecture in a hand gesture recognition framework.

There are two main approaches to time series clustering. In the first approach, a distance measure that is applicable to time series is used to calculate a distance matrix from pairwise distances of samples. A common measure is the DTW cost, which is the cost of aligning one sample to the other. Likewise, pairwise distances can be trivially transformed to similarities, forming a similarity matrix instead. Some clustering methods, such as hierarchical and spectral clustering use these similarity or distance matrices as input to cluster the data [6]. In the second approach, static features are extracted from each sample, essentially converting the time series data to static data. Common static data clustering methods such as k–means can then be used to cluster the features.

While DTW can esimate the similarity of two samples, graphical models such as HMM can measure similarity of a sample respective to a set of samples. This can be used to formulate a k–means type of clustering approach, where the HMMs play the role of cluster means. Hence, each sample is assigned to the *closest* HMM, and each HMM is re–estimated using their own set of sequences. For instance, Oates et al. use DTW to hierarchically cluster the data to form the initial clusters [9]. Hu et al. use DTW iteratively to form initial clusters and for model selection [2]. Ma et al. recursively model the dataset with HMM, calculate a feature called weighted transition occurring matrix and use normalized cut algorithm to divide the set into two clusters [7].

In this work, we first train HMM, IOHMM, HCRF and EDM on the digit dataset and optimize the model parameters. Next, we apply DTW to calculate pairwise distances of samples belonging to the same gesture class and form a distance matrix, and a corresponding similarity matrix. We use these matrices to apply spectral and hierarchical clustering to the digit dataset. Then, we train each graphical model on the resulting clusters and optimize model parameters for the new dataset. We compare the results and show that after clustering, model complexities are significantly reduced and the recognition accuracies reach nearly perfect scores.

The rest of the paper is organized as follows: We explain the time series clustering method in more detail in Section 2. Section 3 introduces the models used for hand gesture modeling. In Section 4, we explain the experiment setup and present the results. Finally, we conclude and discuss future work in Section 5.

2 Time Series Clustering Methodology

Hand gesture samples are time series consisting of concatenated observation vectors corresponding to each time frame, where the observations are features describing the shape and motion of the hand. Thus, any measure of similarity or distance is based on these observations and their sequence. The efficiency of clustering methods directly depends on the selection of these features.

2.1 Feature Selection

The digit dataset used in this work consists of motion–only gestures, i.e., the hand shape is not important. On the other hand, the shape of the trajectory contains most of the information for digits. Yet, clustering only according to the shape of the trajectory will not produce homogeneous clusters that can be modeled with simple graphical models, as the distribution of the hand speed over the trajectory might be different for two samples, even though they share the same path. Such datasets are not suitable for left–right architecture, and should be further clustered. To ensure production of homogeneous clusters in this sense, we use both location and velocity based features.

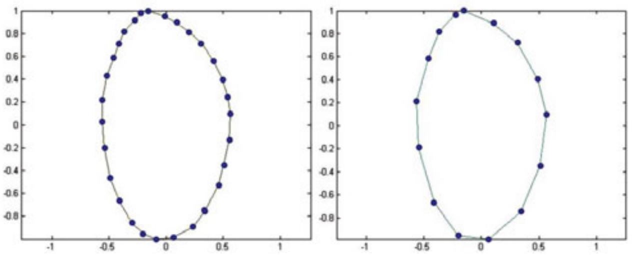

Fig. 1. Effect of resampling a signal. The digit zero is resampled to 15 samples.

First, we normalize the hand coordinates for each sample by tightly mapping the digit in the vertical interval $[-1, 1]$. Then, we resample each gesture sample using cubic interpolation, so that every sequence is of the same length. These steps ensure that some of the spatio–temporal variability is handled manually. Finally, we use these normalized locations, and the differences between consecutive frames as features. Since the digit dataset is essentially in 2D, each real valued observation vector consists of four numbers: two for location, two for velocity.

The effect of resampling a gesture signal can be seen in Figure 1. Here, a sample of digit zero is normalized and resampled to length 15.

2.2 Clustering Methods

As mentioned in Section 1.2, a distance measure for gesture pairs is needed for most clustering algorithms. We use DTW to estimate the cost of aligning two rescaled and resampled sequences. The cost of aligning individual frames of two sequences is taken as the Euclidean distance between the feature vectors corresponding to each frame. Therefore, if the normalized locations are close and the velocities in these locations are similar, this cost is small. Thus, the overall DTW cost is low if the shapes of the paths as well as the velocity distribution over the trajectories are similar.

We applied both spectral and hierarchical clustering methods to the digit dataset. Spectral clustering methods are based on the Min–Cut algorithm, which partitions graph nodes by minimizing a certain cost associated with each edge in the graph [12]. This is a binary clustering method, which can be used to hierarchically cluster data into multiple clusters. A related algorithm has been proposed by Meila and Shi [8], which can estimate multiple clusters. We first form the distance matrix D using pairwise DTW costs, and convert it into the similarity matrix S by taking the reciprocals of each element. Then we normalize each column using the row sums, to obtain the matrix P as follows:

$$D_{i,j} = DTW(G_i, G_j) \tag{1}$$

$$S_{i,j} = 1/\left(D_{i,j} + \epsilon\right) \tag{2}$$

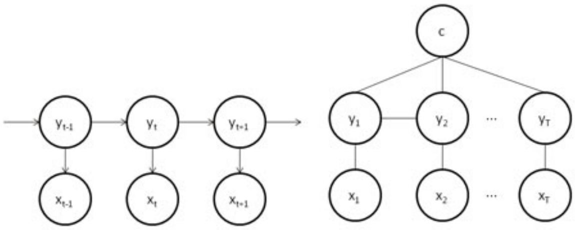

Fig. 2. Graphical models of HMM (left), HCRF (right)

$$R_{i,i} = \sum_j S_{i,j} \tag{3}$$

$$P = SR^{-1} \tag{4}$$

where $DTW(G_i, G_j)$ is the cost of aligning gesture samples G_i and G_j. This cost is symmetric due to resampling of the data. Finally, we take the eigenvectors corresponding to the k largest eigenvalues of the matrix P. We cluster these eigenvectors using the conventional k-means method.

Hierarchical clustering method creates a cluster tree using the distance matrix D [6]. Both bottom–up and top–down strategies can then be followed to merge or divide clusters according to certain criteria. We followed the bottom–up strategy called agglomerative hierarchical clustering. Initially, the algorithm regards each sample as a separate cluster and forms a tree. Then, starting from the leaves, the method merges clusters that have the minimum distance, until a termination criterion is satisfied. We force the algorithm to terminate if the number of clusters reaches a predefined number.

We cluster the digit dataset using both of these methods. Both of the methods manage to detect all the meaningful clusters in the dataset.

3 Hand Gesture Modeling

In this section, we briefly introduce the graphical models mentioned in Section 1.1. The graphical models are depicted in Figures 2 and 3. Here, c is the class label, y_t is the state variable, x_t is the observation, s_t is the input sequence and T_t is the counter value at time t.

3.1 Hidden Markov Models

HMM is one of the simplest graphical models, consisting of discrete states producing observations conditioned on the state and a state transition network with fixed probabilities. Each hidden state of an HMM represents a section of the sequence. Since HMMs are generative models, we train a separate model for each gesture class or cluster. The complexity of HMMs is directly based on the number of hidden states and the allowed transitions.

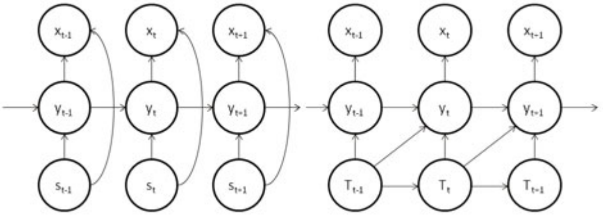

Fig. 3. Graphical model of IOHMM (left) and HSMM (right)

3.2 Hidden Conditional Random Fields

CRFs are the discriminative counterparts of HMMs [4]. CRFs do not have the naive Bayes assumption; each state is conditioned on features extracted from an overlapping set of observations. However, CRFs do not model intra-class dynamics, i.e., each gesture is represented by a single latent variable. The model needs to determine the class label at each time frame based on the current observations. Therefore, CRFs are not suitable for modeling time series. To extend the modeling capabilities of CRFs, Hidden Conditional Random Fields (HCRFs) [13] have been introduced. HCRFs relate a single class variable to the entire sequence. As HCRFs are discriminative models, we train a single model that learns to differentiate between every class or cluster pair.

3.3 Input-Output HMMs

Input-Output HMMs (IOHMM), as hybrids of generative and discriminative models, have shown considerable success in hand gesture recognition [3]. These models condition the state transition and emission probability distributions on an input sequence, which is a function of the observations [1]. The transition and observation probabilities are estimated via local models using the input sequence. In the literature, it is common to use multi–layer perceptrons (MLP) or radial basis function as local function approximators. Consequently, IOHMMs require careful design by an expert and are harder to train. In this study, we use MLPs as local models and train a separate IOHMM for each gesture.

3.4 Hidden Semi Markov Models

The special HSMM called EDM allows explicit modeling of state durations. As in the case of HMMs, the observations x_t are conditioned on the states y_t. Each hidden state is augmented by a positive counter variable τ_t that is initialized and deterministically decreased until it becomes zero. If the counter reaches zero, both the state y_t is allowed to make a transition, and the counter variable τ is reinitialized.

HSMMs are generative models, and a separate model is trained for each gesture.

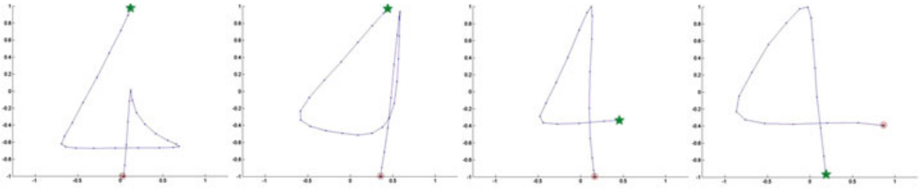

Fig. 4. Some common choices of trajectories by different users for the digit 4. Start and end points are swapped for the third and fourth trajectories.

4 Experiments

To justify the claim that pre–clustering is useful in terms of both speed and accuracy, we conduct several experiments on the digit dataset. First, the digit classes in the dataset are modeled with HMM, EDM, IOHMM and HCRF without pre–clustering. Then, the dataset is clustered using both spectral and hierarchical clustering, and the same graphical models are trained on the pre–clustered datasets. Finally, the accuracies and speeds of the models on these datasets are compared.

4.1 Gesture Dataset

The gestures were performed by 12 subjects, in 10 repetitions, yielding 120 exemplars for each class and a total of 1200 samples. Subjects were not instructed to follow a specific path. As a result, subjects used a wide variety of trajectories, yielding a difficult dataset with high variance. An example is given for the digit 4 in Figure 4.

4.2 Training Methods

HMMs are trained using the Baum-Welch algorithm, EDMs are trained using a generalized version of Baum-Welch algorithm extended for explicit durations [14], HCRFs are trained using the Broyden Fletcher Goldfarb Shanno [13] method, and IOHMMs are trained using generalized expectation maximization method [3]. Training IOHMMs and HCRFs take significantly more time than training HMMs and EDMs. To reduce training times of EDMs and IOHMMs, we initially constructed the models using a priori information obtained from trained HMMs. This reduced training times and increased the accuracy.

4.3 Parameter Optimization

We applied grid search and 5x2 cross validation for parameter optimization over all possible parameters. For HMMs and EDMs, the optimized parameter is the number of states. For HCRFs, both the number of states and the window size is considered. For IOHMMs, MLPs are used as local models. Therefore, IOHMMs have both two parameters: The number of hidden states, and the number of hidden nodes. The optimum parameters for the models on both datasets are shown in Table 1. Here, the number of hidden states is depicted as N_S. N_H is the number of hidden neurons, L is the maximum duration for EDMs, and w is the window size for HCRF.

4.4 Results

The recognition results for the models are given in Table 1. Here, D_O is the original dataset and D_C is the clustered dataset. The models trained on the clustered dataset are constrained to have a left–right architecture. HMM, EDM and IOHMM reach 100% accuracy on the clustered dataset, and HCRF has a recognition rate of 98.95%. This shows that clustering is significantly effective for this problem.

For the original dataset, the number of hidden states that maximize the recognition rates are 16 for the HMM, 19 for the EDM, 8 for the IOHMM and 7 for the HCRF. IOHMM uses 5 hidden neurons, EDM states have a maximum duration 15, and HCRF has a window size of 3 in this case. On the clustered dataset, a left–right HMM with 3 states is capable of achieving perfect accuracy. As HMMs are special cases of EDMs and IOHMMs, these too need only 3 states. HCRFs, however, need more states to be able to distinguish between the increased number of class labels.

Table 1. Recognition rates and optimum model parameters on the original dataset D_O and on the clustered dataset D_C. N is the number of states, H is the number of hidden neurons, w is the window size and L is the maximum duration allowed for EDMs.

	Accuracy on D_O	N_S	N_H	w	L	Accuracy on D_C	N_S	N_H	w	L
HMM	89.7%	16				100%	3			
EDM	91.17%	19			15	100%	3			5
IOHMM	94.33%	8	5			100%	3	2		
HCRF	95.17%	7		3		98.95%	13		3	

Furthermore, the models trained on the clustered dataset are faster in comparison to their original counterparts, both due to their lower complexities and due to their left–right architectures, which are N_S times faster in general.

5 Conclusion

In this study, we proposed unsupervised clustering of gesture samples belonging to gesture classes to improve gesture recognition accuracy of commonly used graphical models. To justify our claims, we collected a challenging digit dataset and trained several graphical models on this dataset. Then we applied a DTW based clustering method to the original dataset and formed a clustered dataset.

We trained the same models on this dataset and achieved perfect accuracy for even very simple models.

This study shows that, rather than solving the isolated gesture recognition task by increasing the complexity of models, one can decrease the complexity of the gesture classes through preprocessing and clustering. Then, fast and simple models are able to attain good accuracies.

References

1. Bengio, Y., Frasconi, P.: Input-output HMM's for sequence processing. IEEE Transactions on Neural Networks 7(5), 1231–1249 (1996)
2. Hu, J., Ray, B., Han, L.: An interweaved hmm/dtw approach to robust time series clustering. In: 18th International Conference on Pattern Recognition, ICPR 2006, vol. 3, pp. 145–148 (August 2006)
3. Keskin, C., Akarun, L.: Stars: Sign tracking and recognition system using input-output hmms. Pattern Recogn. Lett. 30, 1086–1095 (2009)
4. Lafferty, J.D., McCallum, A., Pereira, F.C.N.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: Proceedings of the Eighteenth International Conference on Machine Learning, pp. 282–289. Morgan Kaufmann Publishers Inc. (2001)
5. Lee, H.-K., Kim, J.-H.: Gesture spotting from continuous hand motion. Pattern Recognition Letters 19(5-6), 513–520 (1998)
6. Liao, T.W.: Clustering of time series data - a survey. Pattern Recognition, 1857–1874 (2005)
7. Ma, G., Lin, X.: Typical Sequences Extraction and Recognition. In: Sebe, N., Lew, M., Huang, T.S. (eds.) ECCV/HCI 2004. LNCS, vol. 3058, pp. 60–71. Springer, Heidelberg (2004)
8. Meila, M., Shi, J.: A random walks view of spectral segmentation (2001)
9. Oates, T., Firoiu, L., Cohen, P.: Using Dynamic Time Warping to Bootstrap Hmm-Based Clustering of Time Series. In: Sun, R., Giles, C.L. (eds.) Sequence Learning. LNCS (LNAI), vol. 1828, pp. 35–52. Springer, Heidelberg (2001)
10. Pavlovic, V., Sharma, R., Huang, T.S.: Visual interpretation of hand gestures for human-computer interaction: A review. IEEE Tran. on Patt. Anal. and Machine Intel. 19(7), 677–695 (1997)
11. Rabiner, L., Juang, B.: An introduction to hidden markov models. In: IEEE Acoustic Speech Signal Processing Magazine, pp. 3–4 (1986)
12. Shi, J., Malik, J.: Normalized cuts and image segmentation. In: Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR 1997), p. 731. IEEE Computer Society, Washington, DC (1997)
13. Wang, S.B., Quattoni, A., Morency, L.-P., Demirdjian, D.: Hidden conditional random fields for gesture recognition. In: CVPR 2006: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 1521–1527. IEEE Computer Society, Washington, DC (2006)
14. Yu, S.-Z., Kobayashi, H.: Practical implementation of an efficient forward-backward algorithm for an explicit-duration hidden markov model. IEEE Transactions on Signal Processing 54(5), 1947–1951 (2006)