A method to add Hidden Markov Models with application to learning articulated motion

Y.A.Hicks[1], P.M.Hall[2], A.D.Marshall[1]
1 School of Computer Science
University of Cardiff
Cardiff, CF24 3XF, UK
{Y.A.Hicks, dave}@cs.cf.ac.uk
2 Department of Computer Science
University of Bath
Bath, UK
pmh@cs.bath.ac.uk

Abstract

In this paper we present a method for adding Hidden Markov Models. The main advantages of our method are that it does not require the data the models had been trained on, allows a change in the number of components, does not assume independence of the components to be added and is resistant to the order in which the training data arrives. We assessed the method in the experiments with synthetic data, which showed good accuracy. Finally, we present an application in computer vision.

1 Introduction

Hidden Markov Models (HMMs) have been used in the speech recognition community since the early 1970s [1]. Their popularity is due to sound mathematical structure and wide applicability. More recently, they found use in computer vision for modelling temporal structure of gestures and articulated motion [7, 12].

A facility to update HMMs incrementally is a very desirable one. Among the advantages of incremental update are faster training times, as each time a model is trained only on the part of the data. Another advantage is the possibility of updating a model with new data as it becomes available, thus keeping the model consistent with any changes in the input data, or extending the model to represent a larger data set.

The HMMs that are usually used in computer vision applications are continuous, where the underlying data distribution is modelled with a mixture of Gaussians and each state is represented with a single Gaussian. Among the desirable features of a method to update such a HMM would be not only the ability to update the parameters of the Gaussian components, but also the ability to change the number of these components to represent the data in the most efficient way. Another desirable feature would be the absence of requirement to access the data the HMM had been trained on previously.

There has been recent research in the speech recognition community on incremental update of HMMs [3, 11]. Most of these methods work by updating the Gaussian Mixture

Model (GMM) representing the underlying data distribution and then updating the HMM transitions on the basis of the updated GMM. Among them, the method developed by Gotoh, [3] does not require the previous data, however, it does not allow updating of the number of Gaussians in the mixture, but only of their parameters. The method developed by Lu and Zhang [11] allows updating of the number of components in the mixture as well as their parameters. However, the last method depends on the temporal order of the data used to update a HMM.

There has been also research on updating just GMMs [5, 13, 15] as opposed to HMMs. Most notably, a method developed by Vasconcelos and Lippman [13], allows a change in the number of components in the model, however, it assumes independence between the components, which is not the case if we allow the Gaussians to overlap.

Most recently, a method for adding two or more GMMs, which allows a change in the number of components, does not assume independence between the components to be added, and largely is independent from the temporal order of the input data has been proposed [4]. Moreover, this method does not require any of the original data the GMMs had been trained on, nor does it need to resample the data from the GMMs. The result of adding two GMMs is a third GMM, which closely approximates the one which would have been constructed by a standard algorithm given as input all the data sets used for training of both original GMMs. The method not only allows the addition of two arbitrary GMMs but it also selects the optimum number of clusters to represent the underlying distributions. Up to date, this is the most general method for adding GMMs we know of.

In this paper we propose to utilise the above method [4] to update the underlying HMM data distribution and use the updated GMM to evaluate the transition probabilities between the states. We analyse the performance of the algorithm on a synthetic data set and apply it to the incremental learning of models of human motion from real world data.

2 Adding GMMs

In this section we describe a method for adding GMMs presented in detail in [4], which we then extend to adding HMMs.

Suppose we are given a pair of GMMs; an N-component GMM G_1 made from a data set having N_x points, and an M-component GMM G_2 constructed from a data set of N_y points. The GMMs G_1 and G_2 represent the distributions $p(\mathbf{x})$ and $q(\mathbf{x})$, respectively:

$$p(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i' g(\mathbf{x}; \mu'_i, \mathbf{C}'_i) \tag{1}$$

$$q(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i'' g(\mathbf{x}; \mu''_i, \mathbf{C}''_i) \tag{2}$$

The process of adding these GMMs starts with their concatenation, which consists of concatenating their descriptions and updating the priors in such a way that the sum of the new set of priors remains equal to one. This can be done by taking into account the respective numbers of points the two original distributions had been trained on. The result of this concatenation operation is a new GMM G_3 consisting of $N+M$ components from G_1 and G_2 respectively, and representing the distribution $r(\mathbf{x})$

$$r(\mathbf{x}) = \sum_{i=1}^{N+M} \alpha_i g(\mathbf{x}; \mu_i, \mathbf{C}_i) \tag{3}$$

In the second stage G_3 is "simplified" to a K-component GMM G_4, where $K \leq N+M$, through the application of $(N+M) \times K$ scalars w_{ij}, which are used to specify the contribution that the ith component of G_3 makes to the jth component of G_4. G_4 represents the distribution $s(\mathbf{x})$:

$$s(\mathbf{x}) = \sum_{i=1}^{K} \beta_i g(\mathbf{x}; \nu_i, \mathbf{D}_i) \tag{4}$$

where the parameters are calculated in the following way:

$$\beta_j = \sum_{i=1}^{N+M} w_{ij} \alpha_i \tag{5}$$

$$\nu_j = \frac{1}{\beta_j} \sum_{i=1}^{N+M} w_{ij} \alpha_i \mu_i \tag{6}$$

$$\mathbf{D}_j = \frac{1}{\beta_j} \left(\sum_{i=1}^{N+M} w_{ij} \alpha_i (\mathbf{C}_i + \mu \mu^T) \right) - \nu \nu^{\mathbf{T}} \tag{7}$$

with the following constraints:

$$\sum_{i=1}^{N+M} w_{ij} = 1 \tag{8}$$

$$0 < \sum_{i=1}^{N+M} w_{ij} \alpha_i < 1 \tag{9}$$

The weights w_{ij} for transforming $(N+M)$-component GMM G_3 to a K-component GMM G_4 are found through minimising the χ^2 distance between the $N+M$-component and K-component GMMs using Nelder-Mead search and the constraints (8) and (9).

The number of components K which represents best the sum of G_1 and G_2 is found through consequently simplifying the $N+M$-component GMM to 1,2,3,...,$N+M-1$ components and evaluating the penalised log-likelihood of each simplified GMM. The GMM with the largest value of the penalised log-likelihood is chosen as the result of addition of the two original GMMs G_1 and G_2.

3 Adding HMMs

In this section we consider adding two continuous HMMs where each state is represented using a single Gaussian.

Suppose we have two HMMs: $\lambda_1 = \{\pi^1, \mathbf{A}^1, \mathbf{B}^1\}$ consisting of M states, and $\lambda_2 = \{\pi^2, \mathbf{A}^2, \mathbf{B}^2\}$ consisting of N states, where π^i are the initial state probability vectors, $\mathbf{A}^i$ are the state transition probability matrices, and $\mathbf{B}^i$ are the Gaussians representing the states. Let the result of the addition be a HMM $\lambda_3 = \{\pi^3, \mathbf{A}^3, \mathbf{B}^3\}$

To add λ_1 and λ_2 we firstly add the two underlying observation distributions $\mathbf{B}^1$ and $\mathbf{B}^2$ using the method described above for adding GMMs to estimate the combined distribution $\mathbf{B}^3\}$ and then use the obtained matrix $\mathbf{W}$ to a estimate the new transition matrix and the new initial state probability vector.

The first stage of estimating the probability transition matrix of HMM λ_3 is to combine the two original transition matrices A^1 of size MxM and A^2 of size NxN into a single matrix $\mathbf{A}^c$ of size nxn, where $n = N + M$, as detailed below.

$$\mathbf{A^c} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1M} & a_{1M+1} & \cdots & a_{1M+N} \\ a_{21} & a_{22} & \cdots & a_{2M} & a_{2M+1} & \cdots & a_{2M+N} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \\ a_{M1} & a_{M2} & \cdots & a_{M} & a_{MM+1} & \cdots & a_{MM+N} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \\ a_{M+N1} & a_{M+N2} & \cdots & a_{M+NM} & a_{M+NM+1} & \cdots & a_{M+NM+N} \end{pmatrix}$$

$$\text{where } a_{ij} = \begin{cases} \{A^1\}_{ij} & \text{if } i \leq M \text{ and } j \leq M \\ \{A^2\}_{i-M,j-M} & \text{if } i > M \text{ and } j > M \\ 0 & \text{otherwise} \end{cases}$$

The elements of the probability transition matrix A^3 are obtained using the formula below, where α_i are $\mathbf{B}_3$ the component priors, a_{ij} are the elements of the matrix $\mathbf{A}^c$ and w_{ij} are the elements of the matrix $\mathbf{W}$.

$$\{A^3\}_{ij} = \frac{\sum_{l=1}^{n}\sum_{k=1}^{n}\alpha_k w_{ki} w_{lj} a_{kl}}{\sum_{j=1}^{N}\sum_{l=1}^{n}\sum_{k=1}^{n}\alpha_k w_{ki} w_{lj} a_{kl}} \tag{10}$$

Finally, we estimate the initial state probability vector π^3 of λ_3. To obtain its values we first concatenate the two initial state probability vectors π^1 and π^2 into a single vector π^c and then update the vector elements in the following way.

$$\pi_i^3 = \frac{\sum_{k=1}^{n} w_{ki}\pi_k^c}{\sum_{i=1}^{N}\sum_{k=1}^{n} w_{ki}\pi_k^c} \tag{11}$$

Sometimes the nominator value in the above expression takes on a negative value, in such eventuality we set it to zero before estimating the rest of the elements of π^3.

This approach seems to be sensible considering that minimising over the distances between different HMMs rather than GMMs as in [4] would involve calculating more computationally expensive measures than χ^2, such as, for example, Kullback-Liebler measure. Moreover, a standard distance measure used for assessing dissimilarity between two ergodic HMMs [6] is more sensitive to the difference between the underlying observation distributions than to the difference in the transition matrices. Thus, optimising the result of the addition of the HMM state probability densities first and then using the result to adjust the transition probability matrix has good theoretical and practical foundations. In the next sections we experimentally evaluate the proposed method and show that it produces very good results.

4 Experiments with synthetic data

In this section we present a series of experiments designed to measure the accuracy and efficiency of our method for adding HMMs. Unless otherwise stated, the data used in the experiments is three-dimensional. When we construct a random HMM of N components we ensure the data is *separable* in the sense that under full automation the optimal number of components for an *ab initio* GMM turned out to be N also, thus avoiding strongly overlapping components.

The random HMMs were generated by choosing random transition matrices, means and covariance matrices. The means were randomly distributed in a hypercube of edge d. A low value of d increased the likelihood of "overlapping" components, making separation, simplification, and selecting the correct number of components when adding two GMMs a more difficult task. We found when $d = 20$ the components were usually separable with some exclusions when another HMM had to be generated.

4.1 Measuring distance between two HMMs

In the next section we will describe the experiments we have undertaken to measure the accuracy and efficiency of our method to add two HMMs. To do so we need to employ a measure of distance between two HMMs.

There have been several HMM dissimilarity measures proposed in recent years. Early approaches were based on the Euclidian distance of the discrete observation probabilities [10]. However, these kinds of measures did not take into account the temporal structure represented in the Markov chain.

One of the first distance measures to take into account the temporal structure of Markov chains was proposed by B.-H.Juang et.al. [6], and is based on the Kullback-Liebler distance [9], which characterises the discriminating properties of two probabilistic models λ and ξ:

$$D_{KL}(\lambda,\xi) = \int_{O^\lambda} \frac{1}{G(O^\lambda)} log \frac{p_\lambda(O^\lambda)}{p_\xi(O^\lambda)} p_\lambda(O^\lambda) dO^\lambda \tag{12}$$

In the above expression O^λ is an observation generated by the model λ, and $p_\lambda(O^\lambda)$ is the likelihood of the sequence O^λ being generated by the model λ.

The measure proposed by Juang is an approximation based on the Monte Carlo method, with T being the length of the observation.

$$D_J(\lambda,\xi) = \lim_{T\to\infty} \frac{1}{T} log \frac{p_\lambda(O^\lambda)}{p_\xi(O^\lambda)} \tag{13}$$

Different variants of the above measure were proposed by Kohler [8], Falkhausen et.al. [2] and M.Vihola et.al. [14], but they were mainly concerned with adapting the above measure to left-to-right HMMs, which are usually used in speech recognition applications, or with finding an approximation to the original definition which could be estimated more efficiently.

In the following experiments we use the following variant proposed by Kohler to work with N observations:

$$D(\lambda,\xi) = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{T_i} log \frac{p_\lambda(O_i^\lambda)}{p_\xi(O_i^\lambda)} \tag{14}$$

However, the distance $D(\lambda,\xi)$ is not symmetric. We symmetrise it in the following way:

$$D_s(\lambda,\xi) = \frac{1}{2}[D(\lambda,\xi) + D(\xi,\lambda)] \tag{15}$$

4.2 Accuracy of adding two HMMs

In order to assess the accuracy of our method for HMM addition we require some comparison of the results our method produces to the ground truth. To achieve this objective we performed a series of experiments, where in each experiment we produce a random HMM λ_0, which is to be the ground truth. We use it to generate several observations O_i of time-lengths T_i, which we use to estimate an *ab initio* HMM λ_{init}

Next we separate the observations O_i into two sets, X_1 and X_2. We train two new HMMs λ_1 and λ_2 on the sets X_1 and X_2 respectively. Finally, we add λ_1 and λ_2 to produce λ_{add} using the proposed method and then measure the distance between the ground truth HMM λ_0 and the result of the addition λ_{add}, as well as the distance between λ_0 and λ_{init}. If the two distances are close, it means our method has performed as well as an automatic method for estimating HMM given all the data. We also find the distance between λ_{init} and λ_{add}, which tells us how close the result of addition is to the original HMM.

We repeat the above procedure 10 times for each number K of states in HMM, which we change from 1 to 5. We also investigate how the number of samples used to train the HMMs affects the accuracy of the *ab initio* and added HMM by repeating the whole series of experiments for 200, 400, 800 and 1600 samples from the underlying HMM distribution. The results are shown in Figure 1. In theory, when dealing with the observations of infinite length, the distance propose by Juang (13) will always stay positive. However, when we are dealing with observations of finite length, sometimes it can happen that the result of the expression (14) is a negative number. The longer the observations are, the smaller is the chance that the expression will produce a negative number. As we can see in the graphs, the distances between λ_0 and λ_{init}, and between λ_0 and λ_{add} are comparable. The results improve further when the number of samples increases (Figure 1), but they are more than acceptable even with the smallest number of samples (200) that we used.

5 Experiments with 3D human motion data

In this section we present the result of adding two HMMs, each trained on the 3D motion data collected from one of two people. The data represents the 3D Cartesian positions of 17 markers, attached to the body of a person in places such as elbows and knees, in the course of several cycles of walking motion. The origin of the coordinate system is placed approximately at the centre of gravity of a person's body and hence moves together with the person. Prior to HMM training the data had been normalised with respect to the person's height and to have zero mean. Finally, the dimensionality of the data has been reduced from 51 to 3 for visualisation purposes through PCA analysis.

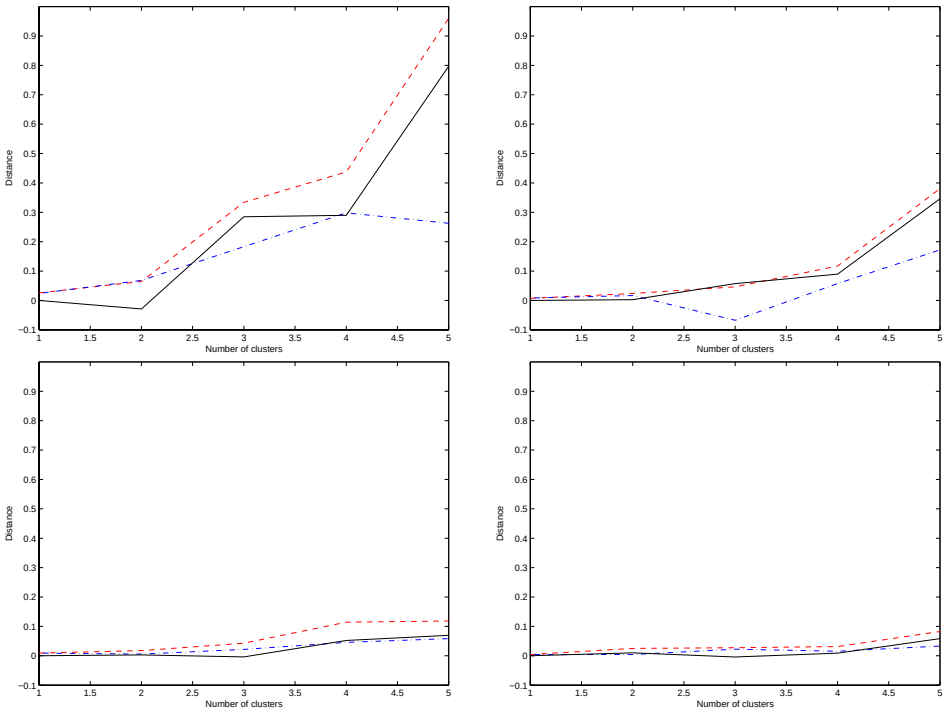

Figure 1: The mean distances, over 10 trials, between λ_{init} and λ_{add} (solid line), λ_0 and λ_{add} (dashed line), λ_0 and λ_{init} (dash-dot line). The distances are calculated for 200, 400, 800 and 1600 samples (displayed in left to right, top down order.)

Wishing to obtain a single model representing the walking motion of both people we add the above HMMs (Figures 2, 3). Both of the original HMMs have been initialised to have 12 Gaussian components, the number chosen by ourselves.

The resulting HMM (Figure 4) has 22 Gaussian components chosen automatically by the method. The number is close to the total number of components in both of the original HMMs. However, as you can see in Figure 4, most of the new components model the data from both distributions, complete with the state transitions, thus providing us with a single model of motion of two people.

6 Conclusions

We presented a novel method for HMM addition, which does not require the data the HMMs had been trained on, allows a change in the number of components, does not assume independence of the components to be added and is resistant to the order in which the training data arrives. The method allows for incremental learning of HMMs as the new data becomes available.

We assessed the method in the experiments with synthetic data, which showed good accuracy. We also presented a practical application of adding two HMMs modelling the walking motion of two different people. The resulting HMM is more compact than the

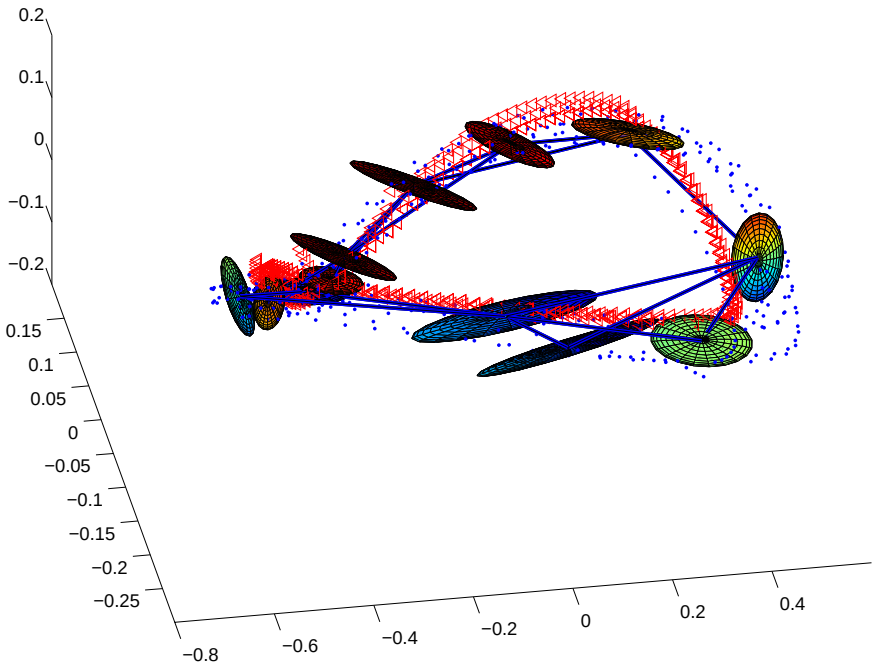

Figure 2: The motion data of two people (dots refer to the first person, triangles refer to the second person) with an overlaid HMM trained on the data of the first person. The Gaussians are represented with the ellipsoids, the possible state transitions are represented with the lines connecting the ellipsoids.

two separate models together, whilst representing the motion of two people. The added HMM, same as the original HMMs, can be used for tracking of walking people in video as proposed in [7]. However, the new model will be able to track the motion of both people.

References

[1] J.K.Baker, Stochastic modeling for automatic speech understanding. Speech Recognition, R.Reddy, ed., New York: Academic Press, 1975.

[2] M.Falkhausen, H.Reininger and D.Wolf. Calculation of distance measures between hidden Markov models. Proc. Eurospeech 1995, pp.1487-1490, Madrid, 1995.

[3] Y.Gotoh. Incremental algorithms and MAP estimation: efficient HMM learning of speech signals. PhD thesis, Brown University, 1996.

[4] P.Hall and Y.Hicks. A method to add Gaussian mixture models. Submitted to IEEE Transactions on Pattern Analysis and Machine Intelligence.

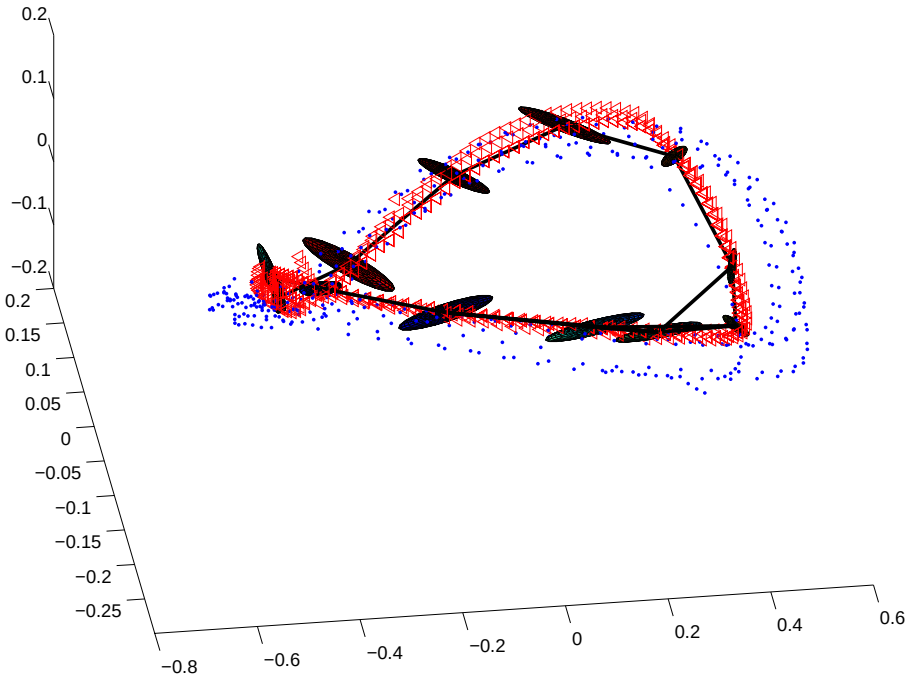

Figure 3: The motion data of two people with an overlaid HMM trained on the data of the second person

[5] M.Figueiredo and A.K.Jain. Unsupervised learning of finite mixture models. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(3), pp.381–396, March 2002.

[6] B.-H. Juang and L.R.Rabiner. A probabilistic distance measure for hidden Markov models. AT&T Technical Journal 64(2), pp.391–408, February 1985.

[7] I.A.Karaulova, P.M.Hall and A.D.Marshall. Tracking People in Three Dimensions Using a Hierarchical Model of Dynamics. Image and Vision Computing 20, pp.691-700, August 2002.

[8] J.Kohler. Multilingual phone models for vocabulary-independent speech recognition tasks. Speech Communication 35, pp.21-30, August 2001.

[9] S.Kullback. Information theory and statistics. Dover publications, New York, 1968.

[10] A.E.Levinson, L.R.Rabiner and M.M.Sondhi. An introduction to the application of the theory of probabilistic functions of a Markov process to automatic speech recognition. Bell System Technical Journal 62(4), pp.1035-74, April 1983.

[11] L.Lu, H.-J.Zhang, Real-time unsupervised speaker change detection. Proc. ICPR 2002, Vol. 2, pp. 358-361, Quebec City, Canada, August 11-15, 2002.

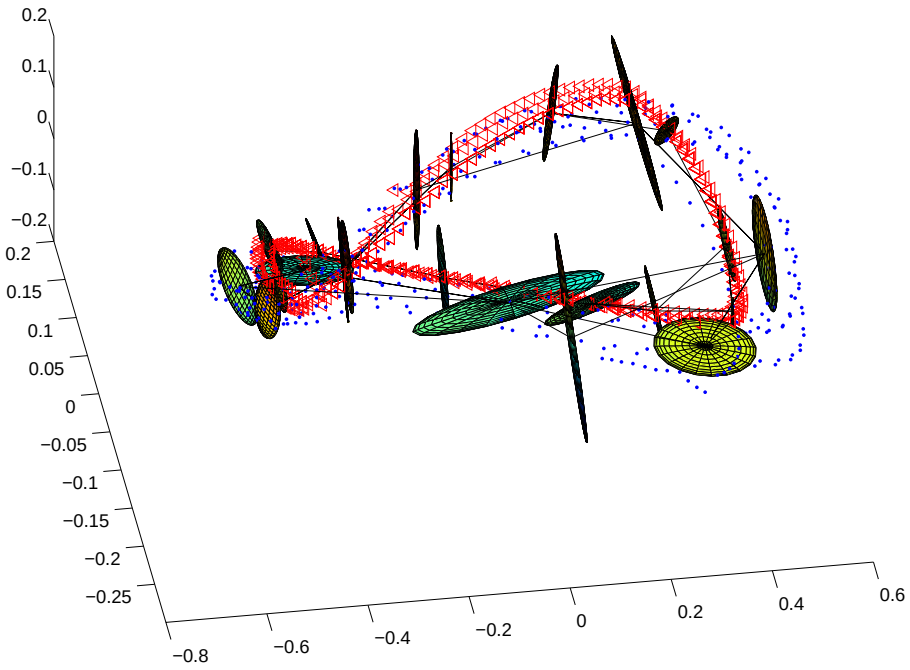

Figure 4: The motion data of two people with an overlaid added HMM modelling the motion of both people.

[12] T.Starner and A.Pentland. Real-Time American Sign Language Recognition from Video Using Hidden Markov Models. M.I.T. Media Laboratory Perceptual Computing Section Technical report No. 375, available at *http://vismod.www.media.mit.edu/cgi-bin/tr_pagemaker*.

[13] N.Vasconcelos and A.Lippman. Learning mixture hierarchies. Neural Information Processing System 11, Denver, Colorado, 1998.

[14] M.Vihola, M.Harju, P.Salmela, J.Suontausta and JSavela. Two dissimilarity measures for HMMs and their application in phoneme model clustering. Proc. ICASSP 2002, Vol.1, pp.933-936, May 13-17, 2002, Orlando, Florida.

[15] Z.R.Yang and M.Zwolinski. Mutual information theory for adaptive mixture models. IEEE Transactions on Pattern Analysis and Machine Intelligence, 23(4), pp.396–403, 2001.