Multimodal Authentication using Asynchronous HMMs

Samy Bengio

Dalle Molle Institute for Perceptual Artificial Intelligence (IDIAP),
CP 592, rue du Simplon 4, 1920 Martigny, Switzerland
bengio@idiap.ch
http://www.idiap.ch/~bengio

Abstract. It has often been shown that using multiple modalities to authenticate the identity of a person is more robust than using only one. Various combination techniques exist and are often performed at the level of the output scores of each modality system. In this paper, we present a novel HMM architecture able to model the joint probability distribution of pairs of asynchronous sequences (such as speech and video streams) describing the same event. We show how this model can be used for audio-visual person authentication. Results on the M2VTS database show robust performances of the system under various audio noise conditions, when compared to other state-of-the-art techniques.

1 Introduction

Biometric identity verification systems use the characteristics of a person to either accept or reject the identity claim made by a person [1]. While several such systems are based on only one characteristic, or modality (such as a spoken sentence or a face), several recent methods have been proposed in order to combine more that one modality in the hope to obtain more robust decisions [2]. Most of these combination methods are in fact based either on the decisions (accept or reject the access) or the scores (often real values) obtained by each unimodal algorithm in order to take a global and hopefuly more robust decision.

In this paper we would like to propose a combination method at the level of the raw data. We will concentrate for this purpose on the difficult task of audio-visual authentication based on two streams of data: an audio stream representing a spoken sentence of a person trying to access the system, and a corresponding video stream of the face of the person pronouncing the sentence.

Trying to combine the models at the level of the raw data in that case is complex for many reasons: first, each stream may have been preprocessed at different frame rates, chosen according to prior knowledge of each stream; second, simply *up-sampling* or *down-sampling* the streams in order to get the same number of frames in each stream might not be the optimal way of combining the streams. We propose in this paper a solution that could overcome this limitation.

In a recent paper [3], we proposed an algorithm to train Asynchronous Hidden Markov Models (AHMMs) in order to model the joint probability of pairs of sequences of data representing the same sequence of events, even when the events are not synchronized between the sequences. In fact, the model enables to *desynchronize* the streams by temporarily stretching one of them in order to obtain a better match between the corresponding frames. The model can thus be directly applied to the problem of audio-visual speaker verification where sometimes lips start to move before any sound is heard for instance.

The paper is organized as follows: in the next section, we review the model of AHMMs, followed by the corresponding EM training algorithm. Related models are then presented and implementation issues are discussed. Finally, experiments on an audio-visual text-dependent speaker verification task based on the M2VTS database are presented, followed by a conclusion.

2 The Asynchronous Hidden Markov Model

Let us denote the 2 asynchronous sequences to model as $X = x_1^T$ and $Y = y_1^S$ where T and S are respectively the length of sequences X and Y, with $S \leq T$ without loss of generality[1].

We are thus interested in modeling $p(x_1^T, y_1^S)$. As it is intractable if we do it directly by considering all possible combinations, we introduce a hidden variable Q which represents the *state* as in the classical HMM formulation [4], and which is synchronized with the longest sequence. Let N be the number of states.

Moreover, in the model presented here, we always emit x_t at time t and sometimes emit y_s at time t. Let us first define $\epsilon(i,t) = P(\tau_t{=}s|\tau_{t-1}{=}s{-}1, q_t{=}i, x_1^t, y_1^s)$ as the probability that the system emits the next observation of sequence Y at time t while in state i. The additional hidden variable $\tau_t = s$ can be seen as the alignment between Y and Q (and X, which is aligned with Q). Hence, we model $p(x_1^T, y_1^S, q_1^T, \tau_1^T)$.

2.1 Likelihood Computation

Using classical HMM independence assumptions, a simple **forward procedure** can be used to compute the joint likelihood of the two sequences, by introducing the following α intermediate variable for each state and each possible alignment between the sequences X and Y:

$$\alpha(i,s,t) = p(q_t{=}i, \tau_t{=}s, x_1^t, y_1^s) \tag{1}$$

$$\begin{aligned}\alpha(i,s,t) = {} & \epsilon(i,t) p(x_t, y_s|q_t{=}i) \sum_{j=1}^{N} P(q_t{=}i|q_{t-1}{=}j)\alpha(j,s-1,t-1) \\ & + (1-\epsilon(i,t))p(x_t|q_t{=}i) \sum_{j=1}^{N} P(q_t{=}i|q_{t-1}{=}j)\alpha(j,s,t-1)\end{aligned}$$

[1] In fact, we assume that for all pairs of sequences (X, Y), sequence X is always at least as long as sequence Y. If this is not the case, a straightforward extension of the proposed model is then necessary.

which is very similar to the corresponding α variable used in normal HMMs. It can then be used to compute the joint likelihood of the two sequences as follows:

$$
\begin{aligned}
p(x_1^T, y_1^S) &= \sum_{i=1}^{N} p(q_T{=}i, \tau_T{=}S, x_1^T, y_1^S) \\
&= \sum_{i=1}^{N} \alpha(i, S, T) \ .
\end{aligned} \tag{2}
$$

2.2 An EM Training Algorithm

An EM training algorithm can also be derived in the same fashion as in classical HMMs. We here sketch the resulting algorithm, without going into more details[2].

Backward Step: Similarly to the forward step based on the α variable used to compute the joint likelihood, a **backward variable**, β can also be derived as follows:

$$
\beta(i,s,t) = p(x_{t+1}^T, y_{s+1}^S | q_t{=}i, \tau_t{=}s) \tag{3}
$$

$$
\begin{aligned}
\beta(i,s,t) = &\sum_{j=1}^{N} \epsilon(j,t+1) p(x_{t+1}, y_{s+1} | q_{t+1}{=}j) P(q_{t+1}{=}j | q_t{=}i) \beta(j, s+1, t+1) \\
&+ \sum_{j=1}^{N} (1-\epsilon(j,t+1)) p(x_{t+1} | q_{t+1}{=}j) P(q_{t+1}{=}j | q_t{=}i) \beta(j, s, t+1) \ .
\end{aligned}
$$

E-Step: Using both the forward and backward variables, one can compute the posterior probabilities of the hidden variables of the system, namely the posterior on the state when it emits on both sequences, the posterior on the state when it emits on sequence X only, and the posterior on transitions.
Let $\alpha^1(i,s,t)$ be the part of $\alpha(i,s,t)$ when state i emits on Y at time t:

$$
\alpha^1(i,s,t) = \epsilon(i,t) p(x_t, y_s | q_t{=}i) \sum_{j=1}^{N} P(q_t{=}i | q_{t-1}{=}j) \alpha(j, s-1, t-1) \tag{4}
$$

and similarly, let $\alpha^0(i,s,t)$ be the part of $\alpha(i,s,t)$ when state i does not emit on Y at time t:

$$
\alpha^0(i,s,t) = (1-\epsilon(i,t)) p(x_t | q_t{=}i) \sum_{j=1}^{N} P(q_t{=}i | q_{t-1}{=}j) \alpha(j, s, t-1) \ . \tag{5}
$$

Then the posterior on state i when it emits jointly on both sequences X and Y is

$$
P(q_t{=}i, \tau_t{=}s | \tau_{t-1}{=}s-1, x_1^T, y_1^S) = \frac{\alpha^1(i,s,t)\beta(i,s,t)}{P(x_1^T, y_1^S)} \ , \tag{6}
$$

[2] The full derivations can be found in the appendix of [5].

the posterior on state i when it emits the next observation of sequence X only is

$$P(q_t{=}i, \tau_t{=}s|\tau_{t-1}{=}s, x_1^T, y_1^S) = \frac{\alpha^0(i,s,t)\beta(i,s,t)}{P(x_1^T, y_1^S)} \;, \tag{7}$$

and the posterior on the transition between states i and j is

$$P(q_t{=}i, q_{t-1}{=}j|x_1^T, y_1^S) = \frac{P(q_t{=}i|q_{t-1}{=}j)}{P(x_1^T, y_1^S)} \cdot \tag{8}$$
$$\left(\begin{array}{l} \displaystyle\sum_{s=1}^{S} \alpha(j, s-1, t-1) p(x_t, y_s|q_t{=}i)\epsilon(i,t)\beta(i,s,t) + \\ \displaystyle\sum_{s=0}^{S} \alpha(j, s, t-1) p(x_t|q_t{=}i)(1-\epsilon(i,t))\beta(i,s,t) \end{array} \right) .$$

M-Step: The Maximization step is performed exactly as in normal HMMs: when the distributions are modeled by exponential functions such as Gaussian Mixture Models, then an exact maximization can be performed using the posteriors. Otherwise, a Generalized EM is performed by gradient ascent, back-propagating the posteriors through the parameters of the distributions.

3 Related Models

The present AHMM model is related to the *Pair HMM* model [6], which was proposed to search for the best alignment between two DNA sequences. It was thus designed and used mainly for discrete sequences. Moreover, the architecture of the Pair HMM model is such that a given state is designed to always emit on either one OR two sequences, while in the proposed AHMM model, each state can always emit both on one or two sequences, depending on $\epsilon(i,t)$, which is learned. In fact, when $\epsilon(i,t)$ is deterministic and solely depends on i, we can indeed recover the Pair HMM model by slightly transforming the architecture.

It is also very similar to the asynchronous version of *Input/Output HMMs* [7], which was proposed for speech recognition applications. The main difference during recognition is that in AHMMs both sequences are considered as output, while in Asynchronous IOHMMs one of the sequence (the shortest one, the output) is conditioned on the other one (the input). The resulting Viterbi decoding algorithm (used in recognition experiments) is thus different since in Asynchronous IOHMMs one of the sequence, the input, is known during decoding, which is not the case in AHMMs.

4 Implementation Issues

The proposed algorithms (either likelihood estimation or training) have a complexity of $\mathcal{O}(N^2 ST)$ where N is the number of states (and assuming the worst case with ergodic connectivity), S is the length of sequence Y and T is the length of sequence X. This can become quickly intractable if both X and Y are longer than, say, 1000 frames. It can however be shortened when *a priori* knowledge is known about possible alignments between X and Y. For instance, one can force the alignment between x_t and y_s to be such that $|t - \frac{T}{S}s| < k$ where k is a constant representing the maximum stretching allowed between X and Y, which should not depend on S nor T. In that case, the complexity (both in time and space) becomes $\mathcal{O}(N^2Tk)$, which is k times the usual complexity of HMM algorithms.

In order to implement this system, we thus need to model the following distributions:

- $P(q_t{=}i|q_{t-1}{=}j)$: the transition distribution, as in normal HMMs;
- $p(x_t|q_t{=}i)$: the emission distribution in the case where only X is emitted at time t, as in normal HMMs;
- $p(x_t, y_s|q_t{=}i)$: the emission distribution in the case where both sequences are emitted at time t. This distribution could be implemented in various forms, depending on the assumptions made on the data:
  - x_t and y_s are independent given state i (which is not the same as saying that X and Y are independent of course):
$$p(x_t, y_s|q_t{=}i) = p(x_t|q_t{=}i)p(y_s|q_t{=}i) \tag{9}$$
  - y_s is conditioned on x_t:
$$p(x_t, y_s|q_t{=}i) = p(y_s|x_t, q_t{=}i)p(x_t|q_t{=}i) \tag{10}$$
  - the joint probability is modeled directly, eventually forcing some common parameters from $p(x_t|q_t{=}i)$ and $p(x_t, y_s|q_t{=}i)$ to be shared.

  In the experiments described later in the paper, we have chosen the latter implementation, with no sharing except during initialization;
- $\epsilon(i,t) = P(\tau_t{=}s|\tau_{t-1}{=}s{-}1, q_t{=}i, x_1^t, y_1^s)$: the probability to emit on sequence Y at time t on state i. With various assumptions, this probability could be represented as either independent on i, independent on s, independent on x_t and y_s. In the experiments described later in the paper, we have chosen the latter implementation.

5 Experiments

Audio-visual text-dependent speaker verification experiments were performed using the M2VTS database [8], which contains 185 recordings of 37 subjects, each containing acoustic and video signals of the subject pronouncing the French digits from zero to nine. The video consisted of 286x360 pixel color images with a 25 Hz frame rate, while the audio was recorded at 48 kHz using a 16 bit PCM coding.

The audio data was down-sampled to 8khz and every 10ms a vector of 16 MFCC coefficients and their first derivative, as well as the derivative of the log energy was computed, for a total of 33 features. Each image of the video stream (25 per seconds) was coded using 12 shape features and 12 intensity features, as described in [9]. The first derivatives of these features were also computed, for a total of 48 features.

In the following, we compared 6 different models:

- an AHMM trained on both voice and face data, as explained in the paper,
- an HMM trained on the fusion of voice and face data (by up-sampling correctly the face data to obtain the same number of frames in the two streams),
- an HMM trained on the voice data only,
- an HMM trained on the face data only,
- a Gaussian Mixture Model (GMM) trained on the voice data only,
- a fusion between the GMM on voice only and the HMM on face only. The fusion was performed using a multi-layer perceptron with the two scores as input.

In all the cases, we used the classical speaker verification technique, computing the difference between the log likelihood of the data given the client model and the log likelihood of the data given the world model (a model created with data no coming from the target client), and accepting the access when this difference was higher than a given threshold.

Although the M2VTS database is one of the largest databases of its type, it is still relatively small to obtain statistically significant results. Hence, in order to increase the significance level of the experimental results, a 4-fold cross-validation method was used as follows: We used only 36 subjects, separated into 4 groups. For each subject, there was 5 different recording sessions. We used the first 2 sessions to create a client model, and the last 3 sessions to estimate the quality of the model. For each group, we used the other 3 groups to create a world model (using only the first 2 sessions per client). Moreover, for each client in one of the other three groups, we adapted a client specific model (using a simple MAP adaptation method [10]) from the world model (again using only the first 2 sessions of the client). Using these client-specific models, we selected a global threshold such that it yielded an Equal Error Rate (EER, when the False Acceptance Rate, FAR, is equal to the False Rejection Rate, FRR). Finally, we adapted (using MAP again) a client-specific model from the world model for each client of the current test group and computed the Half Total Error Rate (HTER, the average of the FAR and the FRR) on the last three accesses of each test client using the global threshold previously found. Hence, all results presented here can be seen as unbiased since no parameters (including the threshold) were computed using the test accesses.

The HMM topologies were as follows: we used left-to-right HMMs for each instance of the vocabulary, which consisted of the following 11 (french) words: zero, un, deux trois, quatre, cinq, six, sept, huit, neuf, silence. Each model had between 3 to 9 states including non-emitting begin and end states.

In each emitting state, there was 3 distributions: $P(x_t|q_t)$, the emission distribution of audio-only data, which consisted of a Gaussian mixture of 10 Gaussians (of dimension 33), $P(x_t, y_s|q_t)$, the joint emission distribution of audio and video data, which consisted also of a Gaussian mixture of 10 Gaussians (of dimension 33+48=81), and $\epsilon(i,t)$, the probability that the system should emit on the video sequence, which was implemented for these experiments as a simple table (but still trained of course).

Training of the AHMM was done using the EM algorithm described in the paper. However, in order to keep the computational time tractable, a constraint was imposed in the alignment between the audio and video streams: we did not consider alignments where audio and video information were farther than 0.68 second from each other (equivalent to 17 video frames).

The GMM models used a silence removal technique based on an unsupervised bi-Gaussian method in order to remove all non-informative frames.

In order to show the interest of robust multimodal speaker verification, we injected various levels of noise in the audio stream during test accesses (training was always done using clean audio). The noise was taken from the Noisex database [11], and was injected in order to reach signal-to-noise ratios of 10dB, 5dB and 0dB.

Note that all the hyper-parameters of these systems, such as the number of Gaussians in the mixtures, the number of EM iterations, or the minimum value of the variances of the Gaussians, were not tuned using the M2VTS dataset, but instead on the task of speech recognition using the Numbers'95 database.

Figure 1 presents the results. For each method at each level of noise injected in the audio stream, we present the Half Total Error Rate (HTER), a measure often used to assess the quality of a verification system. As it can be seen, the AHMM yielded better and more stable results as soon as the noise level in the audio stream was significant. For almost clean data, the performance of the GMM using the audio stream only as well as the one of the fusion of the score of the GMM with the score of the face HMM model were better, but quickly deteriorated with the addition of noise.

6 Conclusion

In this paper, we proposed the use of a novel asynchronous HMM architecture for the task of text-dependent multimodal person authentication. An EM training algorithm was given, and speaker verification experiments were performed on a multimodal database, yielding significant improvements on noisy audio data. Various propositions were made to implement the model but only the simplest ones were tested in this paper. Other solutions should thus be investigated soon.

Acknowledgements

This research has been partially carried out in the framework of the Swiss NCCR project (IM)2. The author would like to thank Stephane Dupont for providing the extracted visual features used in the paper.

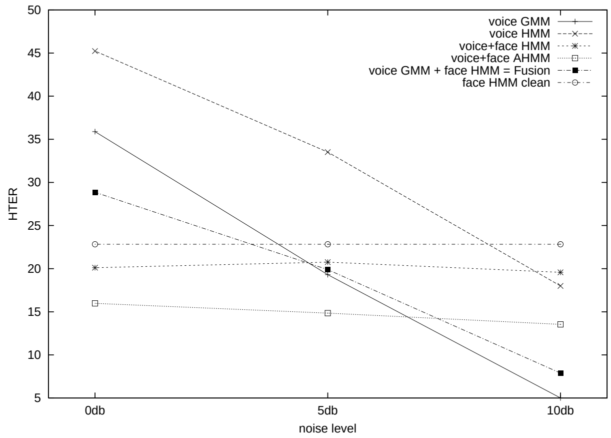

Fig. 1. HTER (the lower the better), of various systems under various noise conditions during test (from 10 to 0 dB additive noise). The proposed model is the AHMM using both audio and video streams.

References

1. Verlinde, P., Chollet, G., Acheroy, M.: Multi-modal identity verification using expert fusion. Information Fusion **1** (2000) 17–33
2. Ross, A., Jain, A.K., Qian, J.Z.: Information fusion in biometrics. In: Proceedings of the 3rd International Conference on Audio- and Video-Based Person Authentication (AVBPA). (2001) 354–359
3. Bengio, S.: An asynchronous hidden markov model for audio-visual speech recognition. In Becker, S., Thrun, S., Obermayer, K., eds.: Advances in Neural Information Processing Systems, NIPS 15. (2003)
4. Rabiner, L.R.: A tutorial on hidden markov models and selected applications in speech recognition. Proceedings of the IEEE **77** (1989) 257–286
5. Bengio, S.: An asynchronous hidden markov model for audio-visual speech recognition. Technical Report IDIAP-RR 02-26, IDIAP (2002)
6. Durbin, R., Eddy, S., Krogh, A., Michison, G.: Biological Sequence Analysis: Probabilistic Models of proteins and nucleic acids. Cambridge University Press (1998)
7. Bengio, S., Bengio, Y.: An EM algorithm for asynchronous input/output hidden markov models. In: Proceedings of the International Conference on Neural Information Processing, ICONIP, Hong Kong (1996)
8. Pigeon, S., Vandendorpe, L.: The M2VTS multimodal face database (release 1.00). In: Proceedings of the First International Conference on Audio- and Video-based Biometric Person Authentication ABVPA. (1997)
9. Dupont, S., Luettin, J.: Audio-visual speech modelling for continuous speech recognition. IEEE Transactions on Multimedia **2** (2000) 141–151
10. Reynolds, D.A., Quatieri, T.F., Dunn, R.B.: Speaker verification using adapted gaussian mixture models. Digital Signal Processing **10** (2000)
11. Varga, A., Steeneken, H., Tomlinson, M., Jones, D.: The noisex-92 study on the effect of additive noise on automatic speech recognition. Technical report, DRA Speech Research Unit (1992)