Phone Recognition Analysis for Trajectory HMM

Le Zhang, Steve Renals

The Centre for Speech Technology Research
University of Edinburgh, Edinburgh EH8 9LW, UK
{Zhang.Le,S.Renals}@ed.ac.uk

Abstract

The trajectory HMM has been shown to be useful for model-based speech synthesis where a smoothed trajectory is generated using temporal constraints imposed by dynamic features. To evaluate the performance of such model on an ASR task, we present a trajectory decoder based on tree search with delayed path merging. Experiment on a speaker-dependent phone recognition task using the MOCHA-TIMIT database shows that the MLE-trained trajectory model, while retaining attractive properties of being a proper generative model, tends to favour over-smoothed trajectory among competing hypothesises, and does not perform better than a conventional HMM. We use this to build an argument that models giving better fit on training data may suffer a reduction of discrimination by being too faithful to training data. This partially explains why alternative acoustic models that try to explicitly model temporal constraints do not achieve significant improvements in ASR.

Index Terms: trajectory HMM, acoustic models, MOCHA-TIMIT

1. Introduction

For decades, the mainstream acoustic models in ASR have been dominated by the Hidden Markov Model (HMM) and its variants. The HMM performs surprisingly well considering various assumptions it makes, namely piecewise stationarity within states, the framewise independence assumption on state output given current state, and simple geometric duration distribution [1]. Although none of these assumptions holds for real speech, the independence assumption, which makes it difficult for the HMM to model temporal correlation within speech data, is regarded by many to be the major drawback of the use of HMMs in speech recognition.

A number of alternative models have been designed to explicitly model temporal correlations in acoustic features, such as the Segmental Models [2], Hidden Dynamic Models [3], and Linear Dynamic Models [4]. When viewed as generative models, those models have a better modelling power for modelling acoustic trajectories, and hence produce a higher likelihood for the training data. However, in practice the added complexity and computational cost of these models are not strongly justified by their performance over the conventional HMMs. It is therefore interesting to ask why models that give a better fit to acoustic data do not produce lower speech recognition error rates compared with simple HMMs for which almost every assumption conflicts with real speech observations. Unfortunately, fully answering this question requires a better understanding of acoustic variability than we currently have [5], mainly due to the highly complex nature of human speech production process.

As a first step in providing some insights on modelling speech dynamics, we study the behaviour of a recently proposed trajectory HMM that handles temporal acoustic correlation through dynamic features [6]. The main difficulty in applying the trajectory model to ASR is that the search space is an order of magnitude larger than a conventional HMM decoding network with the same structure, as the model imposes no conditional independence assumption on the state output. In this paper we present a trajectory decoder based on tree search with delayed path-merging, and analyse its performance on an English speaker-dependent phone recognition task.

The rest of this paper is organised as follows. Section 2 gives a brief introduction to trajectory HMM. In section 3 we describe in detail a trajectory decoder based on tree search with delayed path merging. Decoding analysis on an English phone recognition task is given in section 4. We then conclude with a discussion.

2. Trajectory HMM

Enhancing ASR performance through the use of dynamic features has been a standard practice since 1980s. However, training a conventional HMM directly on the augmented feature vectors will result in an inconsistent model, as the use of dynamic features effectively introduces inter-frame dependence that an HMM can not handle. Recently a theoretical framework called the trajectory HMM has been set up to explicitly model the constraints imposed by dynamic features [6].

For a given HMM state sequence $\mathbf{q} = \{q_1, q_2, \ldots, q_T\}$ corresponding to an utterance with T frames, the likelihood function of the trajectory model is defined as a function of $\mathbf{c} = [\mathbf{c}_1{}', \mathbf{c}_2{}', \ldots, \mathbf{c}_T{}']'$, the sequence of static feature vectors, rather than as a function of $\mathbf{o} = [\mathbf{o}_1{}', \mathbf{o}_2{}', \ldots, \mathbf{o}_T{}']'$, the sequence of augmented feature vectors, which is just a linear transformation of $\mathbf{c}$. Without imposing any conditional independence assumption the likelihood function of observing $\mathbf{c}$ given state sequence $\mathbf{q}$, and the model parameters λ, can be obtained by performing a per-utterance normalisation of the original HMM likelihood function $p(\mathbf{o} \mid \mathbf{q}, \lambda)$:

$$p(\mathbf{c} \mid \mathbf{q}, \lambda) = \frac{1}{\mathbf{Z_q}} p(\mathbf{o} \mid \mathbf{q}, \lambda) \tag{1}$$

where $\mathbf{Z_q}$ is a normalisation term that depends on $\mathbf{q}$:

$$\mathbf{Z_q} = \int p(\mathbf{o} \mid \mathbf{q}, \lambda) d\mathbf{c} \tag{2}$$

Assuming the static acoustic feature has a dimension of M, typically 13 for MFCC-based front-end, the normalised p.d.f. $p(\mathbf{c} \mid \mathbf{q}, \lambda)$ over an utterance of T frames is a TM-dimtionsonal

Gaussian:

$$\boxed{\begin{aligned} &P(\mathbf{c} \mid \mathbf{q}, \lambda) = \mathcal{N}(\mathbf{c} \mid \overline{\mathbf{c}}_\mathbf{q}, \mathbf{P}_\mathbf{q}, \lambda) \\ &\text{satisfying:} \\ &\mathbf{R}_\mathbf{q} = \mathbf{W}' {\boldsymbol{\Sigma}_\mathbf{q}}^{-1} \mathbf{W} \\ &\mathbf{r}_\mathbf{q} = \mathbf{W}' {\boldsymbol{\Sigma}_\mathbf{q}}^{-1} \boldsymbol{\mu}_q \\ &\mathbf{P}_\mathbf{q} = \mathbf{R}_\mathbf{q}^{-1} \\ &\mathbf{R}_\mathbf{q} \overline{\mathbf{c}}_\mathbf{q} = \mathbf{r}_\mathbf{q} \end{aligned}} \tag{3}$$

here $\overline{\mathbf{c}}_\mathbf{q}$ and $\mathbf{P}_\mathbf{q}$ are the TM by 1 mean vector and TM by TM full covariance matrix depending on the state sequence $\mathbf{q}$. $\mu_\mathbf{q}$ and $\boldsymbol{\Sigma}_\mathbf{q}$ are the mean vector and covariance matrix of the corresponding conventional HMM. $\mathbf{W}$ is a deterministic $3TM$ by TM matrix to transform the static feature vector sequence $\mathbf{c}$ into $\mathbf{o}$, the sequence of the augmented feature vectors:

$$\mathbf{o} = \mathbf{W}\mathbf{c} \tag{4}$$

The trajectory HMM can be trained using maximum likelihood estimation with viterbi path approximation. The reader is referred to [6] for more details.

Compared to a conventional HMM, the output of a trajectory HMM satisfies the constraints imposed by the dynamic features, resulting in a smoothed trajectory that varies within a single HMM state. The trajectory HMM has been successfully used as a generative model in HMM-based speech synthesis [7]. Promising results on ASR have also been obtained using an N-best list re-scoring paradigm [6]. A more principled way of using a trajectory model is to apply a "trajectory decoder" directly on the acoustic input, which will be described in the next section.

3. Decoding

Decoding with a trajectory model is much more difficult than decoding using a conventional HMM. Firstly, the state output at time t, $\mathbf{c}_t$, depends on the whole state sequence $\mathbf{q}$, making the usual viterbi algorithm inapplicable. Secondly, as the number of hypothesis grows exponentially to the length of the utterance, we must resort to some approximations to make the search tractable. Thirdly, the formula of trajectory likelihood (3) involves high dimensional matrix manipulation, and is more expensive to evaluate than conventional HMM likelihood. The rest of this section will present a trajectory decoder based on tree search, which can be seen as a variant of the D-frame delayed viterbi algorithm [8].

3.1. Decoding as a tree search with delayed path merging

The decoding problem in ASR can be phrased naturally as a tree search, where nodes at the t-th level of the search tree represents valid HMM states at time t. Starting from an empty root node at time 0, we gradually extend the leaf nodes of the tree using nodes subject to language model constraints. At time T the path with the highest score will be picked up as the decoding result. Figure 1 illustrates a search tree with only two HMM states.

It is easy to see that the tree search procedure outlined above is infeasible to implement as the number of paths grows exponentially as t increases. For a conventional HMM, this exponential growth in path numbers can be avoided by assuming that nodes at the t-th level of the tree only depend on nodes at the $(t-1)$-th level (viterbi assumption). Therefore two paths are equivalent if their last two nodes have the same state. We can generalise this idea further for trajectory HMM by assuming two paths are equivalent if they share the same fixed-length history. More specifically, for given integers $D \geq 1$, $L \geq 1$. two paths $\mathbf{q}_1^{t+L}$ and $\mathbf{p}_1^{t+L}$ are considered to be equal if $\mathbf{q}_{t-D}^{t+L} = \{q_{t-D}, \ldots, q_{t+L}\} = \{p_{t-D}, \ldots, p_{t+L}\} = \mathbf{p}_{t-D}^{t+L}$. Here D and L are the number of frames we need to look back/ahead in order to compute the likelihood score at time t. In other words, whenever we see two equivalent paths we only need to retain the one with the higher score and discard the other. We refer the process of retaining the highest-score path out of its equivalent class $\Psi(\mathbf{q}_1^{t+L}) = \mathbf{q}_{t-D}^{t+L}$ to as path merging. After extending all the nodes at frame $t-1$ to t, we perform path merging for all paths within a window $[t-D, t+L]$, which effectively discards half of the hypothesis at frame t. The usual beam pruning can then be applied to further reduce the hypothesis set. This process is illustrated in figure 1. The main decoding algorithm is outlined below:

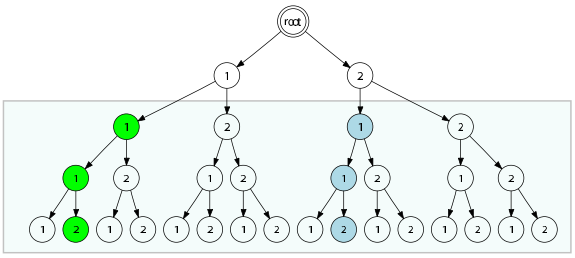

Figure 1: *Illustration of path merging for a search tree of a two-state HMM. Two paths* $1 \to 1 \to 1 \to 2$ *and* $2 \to 1 \to 1 \to 2$ *will be merged as they have the same partial path within the shaded window spanning 3 frames (*$D = 1, L = 1$*). Only the one with a higher partial likelihood score will be retained.*

Algorithm 1 Trajectory Decoding Algorithm

1: $t = 0$: Initialise search tree with an empty root node
2: **for** $t = 1$ to T **do**
3: Extend all nodes at time $t-1$ to time t subject to LM constraints
4: Compute partial trajectory likelihood for all nodes at time t
5: Perform path merging for all paths within a window $[t-D, t+L]$
6: Return the highest-score path at time T

3.2. Recursive likelihood calculation

In order to apply the tree search algorithm, which operates in a time-synchronous fashion, we can write the trajectory likelihood $p(\mathbf{c} \mid \mathbf{q}, \lambda)$ in a recursive form described in [8]:

$$p(\mathbf{c} \mid \mathbf{q}, \lambda) = \prod_{t=1}^{T} \frac{1}{\mathbf{Z}^{(t)}_{\mathbf{q}_1^{t+L}}} p(\mathbf{o}_t \mid q_t, \lambda) \tag{5}$$

where $p(\mathbf{o}_t \mid q_t, \lambda)$ is just the conventional HMM state output function at time t. $\mathbf{Z}^{(t)}_{\mathbf{q}_1^{t+L}}$ is the normalisation term at time t and

depends on the partial state sequence up to time $t+L$:

$$\mathbf{Z}^{(t)}_{\mathbf{q}_1^{t+L}} = \frac{\sqrt{(2\pi)^M}|\mathbf{U}^{(t,t)}_{\mathbf{q}_1^{t+L}}|^{-1}}{\sqrt{(2\pi)^{3M}|\mathbf{\Sigma}_{\mathbf{q}_t}|}} \cdot \exp\left\{-\frac{1}{2}\left(\boldsymbol{\mu}_{\mathbf{q_t}}{}'\Sigma^{-1}_{\mathbf{q}_t}\boldsymbol{\mu}_{\mathbf{q}_t} - \left[\mathbf{g}^{(t)}_{\mathbf{q}_1^{t+L}}\right]' \mathbf{g}^{(t)}_{\mathbf{q}_1^{t+L}}\right)\right\} \quad (6)$$

where $\mathbf{U}^{(t,t)}_{\mathbf{q}_1^{t+L}}$ denotes the t-th diagonal element of the matrix $\mathbf{U}_{\mathbf{q}_1^{t+L}}$, which is the upper triangular matrix of the Cholesky decomposition of the precision matrix $\mathbf{R_q} = \mathbf{U_q}'\mathbf{U_q}$, and $\mathbf{g}^{(t)}_{\mathbf{q}_1^{t+L}}$ is the t-th element of the vector $\mathbf{g_q}$, which is the solution of $\mathbf{U_q}'\mathbf{g_q} = \mathbf{r_q}$.

4. Analysis of decoding result

To analyse the performance of the proposed decoding algorithm, we ran an English phone recognition task using the speech data from the MOCHA-TIMIT database [1], which contains one male speaker (msak0) and one female speaker (fsew0) with 460 TIMIT utterances for each. The first 400 sentences were used for training a monophone trajectory model, and the remaining 60 sentences were used for testing. We use the standard 12 MFCC features plus the log energy, and their delta coefficients. The vocabulary consists of 45 phones, each of which is modelled by a 3-state left-to-right HMM with no skip. The output of HMM state is modelled by a single Gaussian with diagonal covariance. A phone loop grammar was used for decoding.

We started from a conventional HMM trained using HTK. Then the model parameters were updated using the HMM segmentation. After that, the optimal state boundaries were obtained by running the decoder in alignment mode with a large delay ($D = 6$). Then the model parameters were updated again based on the new state boundary information. This process was iterated a few times until the trajectory likelihood on the training data stabilised. For decoding, we used a merging window of 5 frames ($D = 2, L = 2$) with no pruning. The number of active nodes during decoding is of the order of 10^7 under this setting. We note that it is possible to get better result than what is presented here by using larger delays ($D \geq 3$), although heavy pruning had to be employed so that the decoder can run on a machine with 2G RAM. Since the primary focus of this paper is error analysis, we did not include results from larger D, which can be biased due to search errors caused by pruning. Table 1 gives the phone error rate.

Table 1: *Phone error rates on MOCHA data*

	HMM	trajectory
fsew0	50.56%	54.79%
msak0	52.68%	58.96%

The phone error rates of the trajectory HMM (54.79% and 58.96%) are higher than a conventional HMM (50.56% and 52.68%). To help categorise the errors, we now introduce some notation to define the concept of phone level matching alignment. Let t_{w_i} denote the start frame of the i-th phone in a transcription, then the duration and the central position of the i-th phone can be written as:

$$d(w_i) = t_{w_{i+1}} - t_{w_i} \quad (7)$$
$$c(w_i) = t_{w_i} + d(w_i)/2 \quad (8)$$

We define two phones w_i, w_j have a matching alignment if:

$$\begin{cases} |d(w_i) - d(w_j)| < \delta \\ |c(w_i) - c(w_j)| < \epsilon \end{cases} \quad (9)$$

where δ and ϵ are small integers, and are set to 2 and 3 in this experiment. Also we define $f(w_i)$ to be the per-frame likelihood of the i-th phone w_i averaged over its duration $d(w_i)$. Using the above definitions, we are able to classify phone recognition errors into three groups:

1. modelling error: A phone w_{rec} in the recognised transcription receives a much higher score than the correct phone w_{ref} with the matching alignment (as defined by (9)) in the reference transcription:
$$f(w_{rec}) - f(w_{ref}) > \Delta \quad (10)$$
where Δ is a positive number ($\Delta = 1.0$ in this experiment).
2. confusion error: A phone w_{rec} in the recognised transcription receives a score higher than the correct phone w_{ref} with the matching alignment in the reference transcription, but the difference in score is relatively small:
$$f(w_{rec}) - f(w_{ref}) < \Delta \quad (11)$$
3. insert/delete error: Because the model is capable of generating smoothed trajectory, sometimes a phone trajectory is smoothed excessively to give a good fit to data, which is not the case in the reference transcription where the correct alignment corresponds to two phones w_{ref1}, w_{ref2}.
$$f(w_{rec}) - f(w_{ref1}\ w_{ref2}) > 0 \quad (12)$$
It is possible to find an over-smoothed phone spanning over more than two phones, although we find two phones are most common.

Examples of all three kinds of errors are illustrated in figure 2, where smoothed trajectories are generated for both decoded transcription (above) and reference transcription (below) for one utterance in testing set. The first shaded area highlights a confusion error where both phone `p` and phone `y` have a good fit to the data. The second shaded area shows an insert/delete error: phone `y` is smoothed excessively to give a better fit to the data than the correct phone sequence `ii-@`, which explains why it is picked up by the decoder. A modelling error can be seen in the third shaded area where the wrong phone `jh` has a much better fit to the data while the correct phone `s` is not.

The distribution of the three error groups for the baseline HMM and the trajectory HMM is given in table 2. The majority of errors of both models comes from modelling error (53.72% and 62.42%), which is due to the mismatch between the learned model and data. This type of error is difficult to get rid of unless a more complex model, such as a Gaussian mixture, is used to enhance the modelling power. The trajectory HMM, while in general has a better fit to the speech data, actually committed 8.7% more modelling error compared to the baseline HMM. This suggests that the trajectory HMM may suffer a degradation of discrimination by favouring over-smoothed phone trajectory that fits data

[1] http://www.cstr.ed.ac.uk/research/projects/artic/mocha.html.

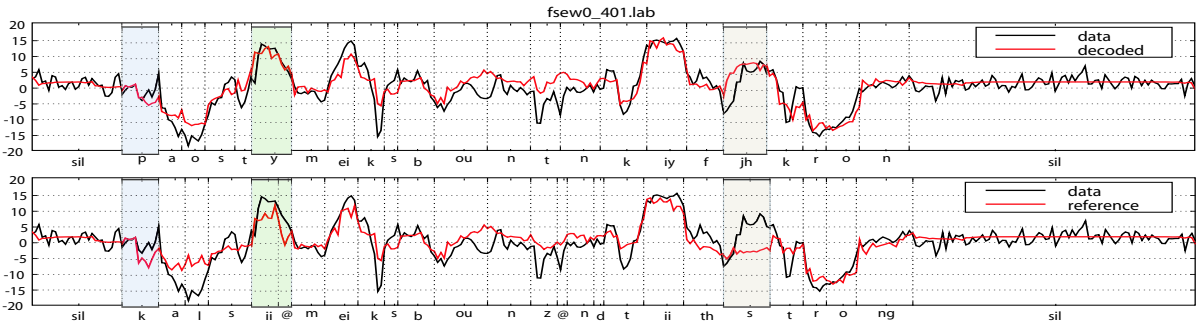

Figure 2: *Smoothed mean trajectories for the first MFCC coefficient generated from transcription returned by the decoder (above) and the reference transcription (below) for one utterance in the test set. The shaded areas correspond to a confusion error, an insert/delete error and a modelling error, respectively.*

well among competing hypothesises. Both models have a similar number of insert/delete errors (28.41% and 26.39%), which can be reduced by applying some sorts of penalty measure to control the length of phone alignment. For confusion errors, trajectory HMM makes 6.7% fewer errors than the conventional HMM (11.19% to 17.88%), which is still a large portion. Training them discriminatively may provide better discrimination between acoustically similar phones in this case.

Table 2: *Distribution of different error groups*

	modelling	confusion	insert/delete
HMM	53.72%	17.88%	28.41%
trajectory	62.42%	11.19%	26.39%

5. Discussion

The decoding result of the trajectory HMM brings out some interesting findings: Firstly, the fact that the trajectory HMM can generate smoothed trajectory for speech data does not necessarily mean a lower recognition error rate. In contrast, we observed that under certain conditions it may be difficult for a model being too faithful to the data to have good discrimination between competing hypothesis. This partially explains why previously proposed alternative acoustic models that explicitly account for temporal correlation in speech, such as Segmental Models and Linear Dynamic Models, are not vastly superior to HMMs in practice. Secondly, like MLE-trained HMMs, the MLE-trained trajectory HMM also suffers from weak disambiguation between acoustically similar phones. Training the model discriminatively may help improve the ASR performance, albeit more computational expensive.

It is worth mentioning that we do observe a reduced error rate on the ATR data as used in [6] via N-best re-scoring, but it does not happen on the MOCHA data when direct decoding is employed. We note the following factors may contribute to the diverse results: 1. our baseline system has a higher error rate than [6]'s 19.7%, which suggests the MOCHA data we used is more noisy; 2. the search space of a full decoder is much larger than that of N-best re-scoring, and a short delay (D=2) has to be used to make decoding tractable, which compromises the performance.

In closing this paper, we emphasis that when seeking alteiternative acoustic models for speech recognition, one should pay more attention on the discriminative power rather than the fitness between learned model and training data. The increase of likelihood on training data does not always lead to improved ASR performance, as is the case in this paper. Our future work will include methods to regularise the over-smoothness of the output trajectory in decoding and applying discriminative training to the model.

6. Acknowledgements

The authors would like to thank Keiichi Tokuda and Heiga Zen of Nagoya Institute of Technology for helpful discussions.

7. References

[1] W. J. Holmes and M. J. Russell, "Probabilistic-trajectory segmental HMMs," *Computer Speech & Language*, vol. 13, no. 1, pp. 3–37, January 1999.

[2] Mari Ostendorf, Vassilios V. Digalakis, and Owen A. Kimball, "From HMM's to segment models: a unified view of stochastic modeling for speech recognition," *IEEE Trans. on SAP*, vol. 4, no. 5, pp. 360–378, 1996.

[3] Hywel B. Richards and John S. Bridle, "The HDM: A segmental hidden dynamic model of coarticulation," in *Proc. of ICASSP 1999*, Phoenix, Arizona, USA, May 1999.

[4] Antti-Veikko I. Rosti, *Linear Gaussian Models for Speech Recognition*, Ph.D. thesis, University of Cambridge, 2004.

[5] Sorin Dusan and Larry R. Rabiner, "On integrating insights from human speech perception into automatic speech recognition," in *Proceedings of INTERSPEECH 2005*, Lisbon, 2005.

[6] Keiichi Tokuda, Heiga Zen, and Tadashi Kitamura, "Reformulating the HMM as a trajectory model," in *Proc. of Beyond HMM*, December 2004.

[7] Keiichi Tokuda, Takayoshi Yoshimura, Takao Kobayashi Takashi Masuko, and Tadashi Kitamura, "Speech parameter generation algorithms for HMM-based speech synthesis," in *Proc. of ICASSP 2000*, Istanbul, Turkey, June 2000.

[8] Heiga Zen, Keiichi Tokuda, and Tadashi Kitamura, "A viterbi algorithm for a trajectory model derived from HMM with explicit relationship between static and dynamic features," in *Proc. of ICASSP 2004*, Montreal, May 2004, pp. 837–840.