COARTICULATION MODELING BY EMBEDDING A TARGET-DIRECTED HIDDEN TRAJECTORY MODEL INTO HMM – MAP DECODING AND EVALUATION

Frank Seide, Jian-Lai Zhou, and Li Deng[*]

Microsoft Research Asia, 5F Beijing Sigma Center, No. 49 Zhichun Rd., 100080 Beijing, P.R.C.
[*]Microsoft Research, One Microsoft Way, Redmond, WA 98052, U.S.A.
{fseide,jlzhou,deng}@microsoft.com

ABSTRACT

The Hidden Dynamic Model (HDM) has been an attractive acoustic modeling approach because it provides a computational model for coarticulation and the dynamics of human speech. However, the lack of a direct decoding algorithm has been a barrier to research progress on HDM.

We have developed a new HDM-based acoustic model, the *Hidden-Trajectory HMM* (HTHMM), which combines the state/mixture topology of a traditional monophone HMM with a target-directed hidden-trajectory model (a special form of HDM) for coarticulation modeling. Because the classical Viterbi algorithm is not admissible, we have developed a novel MAP decoding algorithm for HTHMM that correctly takes the hidden continuous trajectory into account.

This paper introduces our new HTHMM decoder that allows for the first time to evaluate an HDM-type model by direct decoding instead of N-best rescoring. Using direct decoding, we demonstrate that the coarticulatory mechanism of our HTHMM matches traditional context-dependent modeling (enumeration of model parameters): The *context-independent* HTHMM has slightly better accuracy than a crossword-triphone HMM on the Aurora2 task.

The decoder also enables us to include state-boundary optimization into the HDM/HTHMM training procedure. This paper presents the detailed decoding algorithm and evaluation results, while in [1] we present the HTHMM model itself and parameter training.

1. INTRODUCTION

Speech recognition technology has achieved significant progress with the introduction of the Hidden Markov Model (HMM). However, satisfactory accuracies are not yet achieved for spontaneous speech, due to poor modeling of coarticulation, especially for highly varying speaking rates. To overcome the current limitations, we believe it is necessary to incorporate knowledge of structural properties of human speech dynamics into the mathematical representation, in particular the basic target-directed dynamic properties of speech production.

The basis of this paper is our novel Hidden-Trajectory HMM (HTHMM) [1]. HTHMM draws heavily on previous work on Hidden Dynamic Models (HDM) [2, 3], but drops the segmental aspects in favor of a multi-state/frame-based mixture HMM architecture.

A serious issue of previous HDM work has been the lack of a direct decoding algorithm. Most HDM work used rescoring of HMM-generated N-best lists (N=5 or N=100). Although N-best rescoring is resonable in some degree, it uses a reduced search space produced by another modeling mechanism – We get no evidence what would happen without the "help" of that other model (ROVER effect). We doubt that this approach can assess the modeling ability of an HDM (or HTHMM) independently of the HMM[1].

Although the focus of this paper is on the decoding algorithm, we have realized that the decoder has been a key factor during model development. The direct-decoding results have provided the best evidence on how to improve the model, leading to the HTHMM in its present form.

In brief, the HTHMM, which we describe in detail in [1], uses a target-directed trajectory function to represent hidden dynamic variables, characterized by a *deterministic* dynamic system (contrasting the usual stochastic dynamic system, e.g. [2]). Continuity constraints across adjacent phonemes model long-span coarticulation effects. Piece-wise linear mapping is used to modify the HMM means depending on the context-dependent hidden trajectory. Other than in [2], mixture components are independent across frames. This results in a two-layer model structure containing two kinds of hidden states: the discrete HMM state and the continuous hidden dynamics variable.

Our new decoder has served the main goal of this initial study: to compare the HTHMM's capability of modeling coarticulation – using only context-independent (CI) phoneme units – with that of a traditional context-dependent (CD) HMM. This first study was done on a small-vocabulary task (Aurora2 TI-Digits) to keep training times low and eliminate dependence on the language model.

This paper is organized as follows. Section 2 briefly describes the HTHMM structure, while section 3 presents our novel HTHMM decoding algorithm. Section 4 describes the experimental setup and results. Section 5 concludes the paper.

2. HIDDEN-TRAJECTORY HMM (HTHMM)

The coarticulation model underlying our new HTHMM makes two model assumptions: First, speech is assumed to be *target-directed*. Each phoneme has an associated target articulator position, but the target is usually not reached (phonetic reduction). Secondly, we assume articulators succumb to physiological constraints that can be described by a linear filter applied to articulator-position related parameters, in our case the *vocal-tract resonances* (VTR) [2].

[1]Often, even the language model used in N-best generation is omitted in HDM rescoring, adding to the ROVER effect.

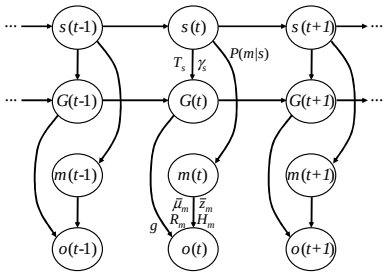

Fig. 1. HTHMM dependency network. HMM state $s(t)$ controls the development of trajectory state $G(t)$ (that depends on its previous value and includes trajectory value $g(t)$) and predicts the mixture component $m(t)$. Observation vector $o(t)$ depends on both.

Our new HTHMM realizes this coarticulation model as a hidden trajectory model (a particular form of HDM) embedded in an HMM. It exposes the hidden VTR parameters underlying the speech-production process; imposes constraints on their trajectories through the *trajectory state equation*; and provides a context/time-dependent *dynamic correction term* for modifying the HMM mixture means. Hence, we call our model *Hidden-Trajectory HMM* (HTHMM). Fig. 1 illustrates the stochastic dependences.

2.1. Trajectory State Equation

The predicted trajectory, named $g(t)$, is described as a second-order discrete-time critically-damped unity-gain low-pass filter:

$$g(t) \;=\; 2\gamma_{s(t)}g(t-1) - \gamma_{s(t)}^2 g(t-2) + (1-\gamma_{s(t)})^2 T_{s(t)} \quad (1)$$

which can be rewritten in canonical form:

$$G(t) \;=\; \Phi_{s(t)} \cdot G(t-1) + U_{s(t)} \quad (2)$$

$$\Phi_s \;=\; \begin{pmatrix} 2\gamma_s & -\gamma_s^2 \\ 1 & 0 \end{pmatrix} \quad (3)$$

$$G(t) \;=\; \begin{pmatrix} g(t) \\ g(t-1) \end{pmatrix} \quad (4)$$

$$U_s \;=\; (1-\gamma_s)^2 \cdot \begin{pmatrix} T_s \\ 0 \end{pmatrix} \quad (5)$$

where $G(t)$ is the (augmented) continuous state, $s(t)$ the HMM state at time t, $T_{s(t)}$ the target vector associated with the corresponding phoneme, and $\gamma_{s(t)}$ the system-dynamics parameter.

The predicted trajectory $g(t)$ contains several inaccuracies and deviates from the "true" trajectory called $z(t)$. $z(t)$ is modeled by a Gaussian distribution with mean $g(t)$ and covariance matrix Q:

$$p(z|g) \;=\; \mathcal{N}(z; g; Q) \quad (6)$$

2.2. Dynamic Mean-Correction Term

The non-linear relationship between the trajectory value z and the mixture means is developed into a Taylor series w.r.t. mixture m's expected trajectory value $\bar{z}_m$ and cut off after the first-order term. This yields a state emission PDF of the following form:

$$p(o|s,g) \;=\; \sum_m P(m|s) \int_z \mathcal{N}(o; \bar{\mu}_m + H_m \cdot (z-\bar{z}_m); R_m) \cdot \mathcal{N}(z; g; Q)\; \mathrm{d}z \quad (7)$$

$$\;=\; \sum_m P(m|s) \cdot \mathcal{N}(o; \bar{\mu}_m + H_m \cdot (g-\bar{z}_m); R'_m) \quad (8)$$

with $\bar{\mu}_m$ being the context-independent HMM mixture mean that is modified by the context-dependent trajectory value z through H_m. Silence, whose model parameters do not depend on z, is implemented by setting $H_m = 0$ (model degenerates to an HMM).

3. MAP DECODING WITH HDM

A MAP decoding algorithm for an HDM-type model like the HTHMM must correctly takes the hidden trajectory state variable $G(t)$ into account. The traditional Viterbi decoder is not applicable because $G(t)$ is continuous. Our solution is to discretize $G(t)$ to make dynamic programming applicable again. The resulting new decoding algorithm is described in the following.

As usual, we state the decoding problem as finding the word sequence $\hat{W}$ that most likely generated our acoustic observation O (maximum-a-posteriori decoder)[2]:

$$\hat{W} \;=\; \operatorname*{argmax}_W P(W|O) \;=\; \operatorname*{argmax}_W p(O|W) \cdot P(W)$$
$$\;\approx\; \operatorname*{argmax}_W \operatorname*{argmax}_S p(O|S) \cdot P(S|W) \cdot P(W) \quad (9)$$

where S denotes an HMM state sequence (path), $p(O|S)$ the acoustic model, $P(S|W)$ the path's state-transition probability, and $P(W)$ the language model (LM) or grammar.

As usual, we assume independence of individual frames for $p(O|S)$:

$$p(O|S) \;=\; \prod_{t=1}^{T} p(o(t)|s(t), g(t)) \quad (10)$$

This expression differs from the traditional HMM by the presence of the additional dependence on $g(t)$, which is fully determined by the state sequence $s(t)$ and Eq. (1).

$P(S|W)$ and $P(W)$ are the same as for the traditional HMM. We represent both by weighted finite-state transducers[3] [4]:

For the LM $P(W)$, the quantity h shall denote the LM state. A word sequence W corresponds to a sequence of LM states in the LM transducer. $P(W)$ is computed as the product of transition probabilities $P(h|h')$ along this state sequence. For example, in the simplest case of an M-gram language model, h would represent sequences of $M-1$ words, while a more compact network can be achieved by exploiting pruning and backing-off properties. LM state $h = 0$ denotes the sentence beginning, and h_{term} the end.

The lexicon – usually organized as a tree – is represented by $P(S|W)$, which is composed of state transition probabilities $P(s|s')$ from lexicon state s' to state s. State $s = 0$ shall represent the non-emitting start state (root of the tree) and s_h the terminal state of the last word associated with LM state h.

[2]Although in this paper we only apply the decoder to a small-vocabulary task, we will formulate the decoding equations for the general case of large-vocabulary recognition with a language model.

[3]In our algorithm, the transducers for $P(S|W)$ and $P(W)$ are composed on the fly during decoding. Currently, the composition is suboptimal: no optimization is performed after composition (especially determinization, which has the important effect of language-model factorization. Unigram factorization can, however, easily be integrated in $P(S|W)$).

3.1. Recursive formulation of MAP criterion

The total search space is a composition of the HMM, LM, and (continuous) trajectory state spaces. Every partial-path hypothesis at time t depends on the joint state $(s(t), h(t), G(t))$. In a dynamic-programming formulation, only partial-path hypotheses with the same joint state can be recombined, in order to guarantee finding the globally optimal path.

Extending the notation in [5], we define the following quantities:

> $Q_{G,h}(t,s) :=$ probability of best path up to time t that ends in state s of the lexical tree with the trajectory state G and the LM state h.
>
> $H_G(h;t) :=$ probability that the acoustic observation vectors $o(1)...o(t)$ are generated by a word/state sequence that ends with the trajectory state G and in LM state h at time t.

It can be shown that with these, the MAP criterion in Eq. (9) can be rewritten in a recursive dynamic-programming like form:

$$\max_W \max_S p(O|S) \cdot P(S|W) \cdot P(W)$$

$$= \max_{h'} P(h_{\text{term}}|h') \cdot \max_G \Big\{ H_G(h';T) \Big\} \quad (11)$$

$$Q_{G,h}(t,s) = \max_{g,g'} \Bigg\{ \delta_{G,\binom{g}{g'}} \cdot p(o(t)|s,g) \quad (12)$$

$$\cdot \max_{G'} \Big\{ \delta_{G,\Phi_s G'+U_s}$$

$$\cdot \max_{s'} \big\{ P(s|s') \cdot Q_{G',h}(t-1,s') \big\} \Big\} \Bigg\}$$

$$Q_{G,h}(0,s) = 0 \quad (13)$$

$$Q_{G,h}(t-1,0) = H_G(h;t-1) \quad (14)$$

$$\delta_{x,y} = \begin{cases} 1 & \text{for } x = y \\ 0 & \text{otherwise} \end{cases} \quad (15)$$

$$H_G(h;t) = \max_{h'} \Big\{ P(h|h') \cdot Q_{G,h'}(t,s_h) \Big\} \quad (16)$$

$$H_G(h;0) = 0 \quad (17)$$

$$H_G(0;0) = 1 \quad (18)$$

with $t > 0$, $s > 0$, and $h > 0$. The word-level recombination equations (16-18) do not differ from the traditional HMM except for the additional dependence on G.

3.2. Linear complexity through quantization of G

The above formulation is guaranteed to find the most likely path given the HTHMM model. However, because in general different state sequences lead to different trajectory states G, recombination is unlikely. Being a continuous variable, G is "too precise" – even a tiny difference of G will prevent recombination, although it may have no practical impact on the decision for the best path. With this, computational complexity grows exponentially in time.

To achieve complexity linear in time as for the traditional Viterbi HMM decoder, G needs to be replaced by a discrete variable. We introduce the heuristic of quantizing G w.r.t. recombination. The idea is that if the trajectory state is similar, the scores of succeeding paths are also likely to be similar due to the target-directed nature of speech, and that a local decision, although sub-optimal, would not affect the global decision for the word sequence.

Although based on the same idea, our quantization approach is different from the stack-based path-merging strategy presented in [6], where paths are recombined if their two expected trajectory values z differ by less than a threshold. That method is less efficient because it involves searching the stack for the most similar hypothesis (including computing distances of trajectory values), the results are somewhat unpredictable in that they depend on order of processing the hypotheses, and an absolute per-state stack-size limit is used to control recombination, rather than path likelihoods.

We define the discrete quantity $c(G)$ which denotes the class of trajectory states that are equal after quantization. With this, the above equations are modified by replacing $Q_{G,h}$ by $Q_{c(G),h}$, H_G by $H_{c(G)}$, and the $\delta_{x,y}$ expressions by $\delta_{c(x),c(y)}$, to yield the final decoding equations.

3.3. Generalizing to similar types of models

The formalism above can be generalized to similar types of models. E.g., we have modified it for segmental models by modifying Eq. (10) and redefining G as $(g(t_s), t_s)$ with segment start time t_s (resembling a time-conditioned structure [5]). Additional layers of hidden variables can be incorporated by extending G.

4. EVALUATION ON SMALL VOCABULARY

In this initial study, we have evaluated our system on the small-vocabulary task of digit-string recognition: the clean portion of the Aurora2/TI-DIGITs database (training set: 4h; test set: 2h). The system is gender independent and uses the standard HTK feature configuration from the Aurora2 distribution. As for the default Aurora2 setup, the whole-word baseline uses word models with 16 emitting states. The monophone, triphone, and HTHMM systems use a 20-phoneme subset of the SAMPA phone set (3 states each).

The dictionary consists of twelve entries including `OH` (transcribed as `/oU/`) and (except for the whole-word system) contained two transcriptions of `ZERO` (`/zIro/` and `/zIroU/`). The latter was necessary because otherwise `ZERO` pronounced the latter way would regularly be recognized as `ZERO OH` generating a large number of insertions which dominated the error rates. The language model is a simple word-loop grammar with word-insertion penalty hand-tuned w.r.t. test-set word error rate.

4.1. HMM Baseline

The HMM baseline was built using HTK 3.0. CART decision-tree based clustering was used for state tying and crossword-triphone generalization. We found that for the digit-recognition task, the CART threshold had to be reoptimized. The final crossword-triphone system had 188 tied states.

4.2. HTHMM Training

We used the HTHMM Viterbi training described in [1]. Forced alignment was done with our decoder. The HTHMM has the same topology as the HMM, which allowed us to initialize the model parameters from the baseline HMM monophone models: HMM mixture means, variances, and weights were copied into the HTHMM, transform matrices H were set to 0. The dynamics parameters γ_s and T_s were initialized by values used in previous work [3], and a meaningful initial value of Q, which we found to be not critical, was guessed. The initial EM iterations used a state-level segmentation that was also generated using the HMM models.

Table 1. Comparison of traditional HMM system with HTHMM.

Id	System	WER [%]				rel.
	#mix:	8	16	32	64	imp.
baseline: traditional HMM						
B1	monophone	0.87	0.69	0.61	**0.52**	ref
B2	triphone	0.56	0.49	**0.40**	0.42	23%
B3	whole-word	0.51	0.36	**0.32**	0.32	39%
proposed HTHMM model						
M1	baseline mapping	0.52	0.43	**0.37**	0.40	29%
M2	Δg in mapping	0.43	0.36	**0.35**	0.36	33%
H1	$H = 0$	1.00	0.79	0.68	**0.53**	-2%

4.3. HTHMM Decoding

Decoding was done using the algorithm described in section 3. For $c(G)$, we used a simple discretization of G by quantizing each of the six components of G linearly into several bits. Because of the low error rates and small search space, we have not yet evaluated the relationship between accuracy and G quantization.

4.4. Results

The goal of this study was to achieve the error rates of an HMM crossword-triphone system with an HTHMM using only context-independent models. Models with 8, 16, 32, and 64 mixture components per state were compared w.r.t. word error rate (WER). Because different models have different parameter numbers, the intention is to compare the systems by their "best-possible" result, i.e. for the mixture number that yields the respective lowest error rate. Except for the monophone setup, the table shows that all models have reached saturation at 32 mixtures.

The baseline monophone HMM system (experiment B1) has a best WER of 0.52% for 64 mixture components per state. Triphone HMMs are 23% relatively better (0.40%, B2). The whole-word baseline B3 reaches 0.32%, but whole-word models do not generalize to phoneme-based large-vocabulary systems (our ultimate target), so we do not aim at reaching their performance with our HTHMM system. For comparison, one of the best WERs for a whole-word based systems (0.24%) is reported in [7].

The HTHMM model (experiment M1) achieves slightly better WERs than the HMM triphone system (0.37% vs. 0.40%). This has been achieved by using only context-independent units: context dependence was modeled entirely through the trajectory-based mean-correction term. The model is about 20% smaller than a triphone system with the same mixture number.

Experiment M2 shows the results for a variation of our model in which we incorporate the first-order derivative of $g(t)$ into the mapping. This was done by replacing $g(t)$ by $\begin{pmatrix} g(t) \\ \Delta g(t) \end{pmatrix}$ (and $z(t)$ accordingly) in Eqs. (6, 8). The number of columns of H_m doubles (roughly 1.6-times overall parameter increase). We observe an obvious improvement for both 8 and 16 mixtures, but the improvement of the best WER is small.

To our knowledge, this is the world's first HDM-based speech recognizer that reaches (slightly outperforms) HMM triphone accuracy with context-independent units only, using direct one-pass decoding rather than rescoring of HMM-generated N-best lists.

In experiment H1, we wanted to know the meaning of the context-independent mixture means $\bar{\mu}_m$. We used the models trained for M1, but set all H transforms to 0 (without further EM iterations). One can see that the system still works reasonably well and nearly achieves the WER of the HMM monophone baseline. This confirms experimentally our characterization of the HTHMM as a monophone HMM with an HDM-based correction term. Most important, this experiment, together with the fact that the HTHMM reaches crossword-triphone performance with context-independent units, demonstrates that the correction term indeed fulfills its intended purpose of modeling coarticulation.

5. CONCLUSIONS AND FUTURE WORK

We have developed a novel MAP decoding algorithm to evaluate our new HTHMM model that provides a computational model of coarticulation and dynamics of human speech. The ability of direct decoding rather than N-best rescoring allowed us to obtain results free of bias from another model and was a key factor in the HTHMM model development.

With the CI HTHMM (a compact model with no CD parameters), we obtained performance improvement over the CD HMM (crossword triphone system). This provides evidence that the coarticulatory mechanism represented by the HTHMM via the model structure matches the traditional context-dependent modeling approach based on enumeration of model parameters.

To our knowledge, this is the first study to evaluate an HDM-based model using one-pass direct decoding, and we hope to remove a major obstacle in the research of HDM-based acoustic models. We are currently working on large-vocabulary tasks, including modeling, training, and analysis and optimization of search-space, and hope to demonstrate that equally excellent performance of the new HTHMM approach can be established despite the weaker phonotactic constraints and more confusable acoustic space.

6. ACKNOWLEDGEMENTS

The authors wish to thank Dr. Asela Gunawardana for sharing his insight on weighted finite-state transducers, and Drs. X.D. Huang, E. Chang, and A. Acero, and other EARS-project researchers at MSR for valuable discussions and encouragement of this work.

7. REFERENCES

[1] J.-L. Zhou, F. Seide, and L. Deng. Coarticulation modeling by embedding a target-directed hidden trajectory model into HMM – Model and training. *Submitted to ICASSP'03*.

[2] J. Ma and L. Deng. Target-directed mixture linear dynamic models for spontaneous speech recognition. *IEEE Trans. Speech and Audio Processing*, to appear in 2002.

[3] J. Bridle *et al*. The WS98 final report on the dynamic model. `http://www.clsp.jhu.edu/ws98/projects/dynamic/presentations/finalhtml/index.html`, Johns Hopkins University, 1998.

[4] M. Mohri. Finite-state transducers in language and speech processing. *Computational Linguistics*, 23, 1997.

[5] S. Ortmanns, H. Ney, F. Seide, and I. Lindam. A comparison of the time conditioned and word conditioned search techniques for large-vocabulary speech recognition. *Proc. ICSLP'96*, Vol. 2, pp. 1017-1020, Philadelphia, 1996.

[6] Jeff Ma and Li Deng. A path-stack algorithm for optimizing dynamic regimes in a statistical hidden dynamic model of speech. *Computer Speech and Language*, 2000, pp. 1-14.

[7] L. Welling *et al*.. Connected digit recognition using statistical template matching *Proc. Eurospeech'95*, pp. 1483-1486, Madrid, 1995.