Intermediate-State HMMs to Capture Continuously-Changing Signal Features

Gustav Eje Henter[1] *and W. Bastiaan Kleijn*[1,2]

[1]Sound and Image Processing Laboratory, KTH–Royal Institute of Technology, Stockholm, Sweden
[2]School of Engineering and Computer Science, Victoria University of Wellington, New Zealand

gustav.henter@ee.kth.se, bastiaan.kleijn@ecs.vuw.ac.nz

Abstract

Traditional discrete-state HMMs are not well suited for describing steadily evolving, path-following natural processes like motion capture data or speech. HMMs cannot represent incremental progress between behaviors, and sequences sampled from the models have unnatural segment durations, unsmooth transitions, and excessive rapid variation. We propose to address these problems by permitting the state variable to occupy positions between the discrete states, and present a concrete left-right model incorporating this idea. We call this *intermediate-state HMMs*. The state evolution remains Markovian. We describe training using the generalized EM-algorithm and present associated update formulas. An experiment shows that the intermediate-state model is capable of gradual transitions, with more natural durations and less noise in sampled sequences compared to a conventional HMM.
Index Terms: Markov models, HMMs, speech synthesis

1. Introduction

Hidden Markov models (HMMs) [1] are a central modeling tool of modern signal processing, both for discriminative and generative tasks. In the speech field, HMMs are ubiquitous, e.g., [2, 3]. An important reason why HMMs have become popular is that they are relatively fast to use and train also for large databases, without making overly simplistic assumptions of short memory or linearity.

However, discrete-state HMMs are more accurate for some processes than for others, as is seen by sampling from the models. In particular, several assumptions made in constructing traditional HMMs are not satisfied by natural processes such as speech and motion capture data.

Generally, HMMs have problems representing steadily-evolving processes where the mean, variance, and mode change gradually over time, so that the process follows a smooth path through the output space. Three prominent problem areas are segment durations, which are often unnatural, transitions, which are not gradual, and variation between frames, which is very rapid and frequently too large. These shortcomings are very noticeable in output generated by sampling from speech HMMs.

In practice, the above problems are often addressed by increasing the number of HMM states, adding dynamic features (not straightforward to do in a mathematically consistent manner [4]), and not actually sampling from models during synthesis, but only generating the most probable parameters [5]—effectively hiding model flaws rather than fixing them. Even when results thus obtained are acceptable, better model classes could provide greater insight into the studied processes, and act as stepping stones to further increased performance.

In this paper, we formulate new models that are better suited for describing steadily-evolving processes, using only a few parameters more than conventional HMMs. The main contribution is to introduce the idea of an intermediate state: By allowing the process to be in a position in between discrete HMM states, gradual transitions and reasonable duration distributions are made possible. The state evolution remains Markovian, enabling simple and efficient training. Since more systematic variation is explained by these models, the amount of unnatural, rapid random variation is reduced. Our results confirm these improvements over traditional HMMs.

The dense or continuous state spaces offered by intermediate states have connections to dynamical models. However, previous dynamical models of speech are often restricted by being linear [6] or non-probabilistic [7]. Our proposal is also related to hidden semi-Markov models (HSMMs). These let the time spent in the current state—a progress indicator similar to intermediate state positions—influence transition probabilities to get natural durations. However, HSMMs do not utilize their extra information to also provide gradual transitions. Because HSMMs have an unbounded state-space, efficient training is complicated and only possible in certain cases [8].

The remainder of the text is laid out as follows: section 2 describes the general advantages and shortcomings of using HMMs to model speech. Section 3 introduces the idea of intermediate states, and section 4 discusses how the resulting models can be trained efficiently. Experiments confirming better modeling of speech durations and transitions, with less inter-frame random variation, are presented in section 5, while section 6 concludes.

This research is supported by the LISTA project. The project LISTA acknowledges the financial support of the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under FET-Open grant number: 256230.

2. Background

A discrete-state HMM is a Markov chain where the state variable $S_t \in \{1, \ldots, N\}$ at t is not directly observable. Instead, one observes a random vector $\boldsymbol{X}_t \in \mathbb{R}^K$ whose distribution depends on S_t. In speech, S_t may be seen as an index determining the current speech sound, and $\boldsymbol{X}_t$ represents its spectral properties (e.g., MFCCs) and their variation. We assume Gaussian output distributions, a common choice in speech modeling.

2.1. Advantages and Disadvantages of HMMs

HMMs have numerous appealing properties: They are probabilistic, so they treat uncertainty in a systematic manner. They are generative, so they can be applied for both classification and synthesis. They strike an attractive balance between expressiveness and efficiency: Simpler models make strong assumptions (e.g., linearity) that limit their accuracy. More complicated models cannot attain their full potential, as they cannot be trained on large databases due to superlinear computational demands.

Despite their advantages, conventional HMMs are not appropriate in all contexts. By design, HMMs describe features that switch instantaneously and memorylessly between distinct regimes of white noise behavior. This is not a good fit for steadily-evolving real-world processes, e.g., speech or motion capture data. By sampling from simple left-right HMMs trained on such gradually changing data, it is obvious that the random samples are not like the training examples at all. Three major aspects differ:

1. Durations are incorrectly modeled;
2. Gradual transitions are not well described;
3. The output has a large amount of uncorrelated random variation.

These differences tend to be most problematic in synthesis applications. For recognition tasks, models are typically both trained and tested on similar signals, and only the description accuracy inside that data space matters.

2.2. HMM shortcomings explained

In an HMM, the total time spent in a certain hidden state n before moving on to the next state is the *state* or *epoch duration* D_n. Since the Markov chain is memoryless, D_n follows a geometric distribution. The most probable duration is then invariably a single frame (the shortest possible), and duration means and variances are tied together.

Speech features tend to evolve steadily from one sound to the next, so very short and very long segment durations are uncommon. The wide range of durations produced by HMMs sounds unnatural in comparison. Moreover, the HMM state evolves in discrete steps. Output statistics (e.g., mode) remain constant between state switches, making HMM output stepwise and unsmooth.

Any variation the model cannot explain, such as gradual transitions, is attributed to independent Gaussian noise in the HMM features. Unexplained systematic variation produces correlations among the residuals (estimated deviations from the mean) of the fitted HMM. Samples from HMMs preserve the general deviation magnitude but have independent deviations between frames, leading to rapid variation in the output, compared to the training data.

3. Intermediate states

HMMs generally fail to recognize that a process can be in between two values or behaviors. The main idea in this paper is to consider the effects of permitting the *state variable* of the process to be in-between two integer HMM states. This allows gradual changes in output distribution. Incremental state-space motion is also made possible, enabling more natural durations. Finally, as gradual transitions can follow a smooth mean-value contour better, more variation can be explained, reducing output variances and mitigating also the third major HMM issue.

A defining advantage of the proposal is that the state evolution remains Markovian. This permits efficient training and sampling based on established HMM algorithms.

3.1. Intermediate state preliminaries

A natural intermediate-state generalization is to replace the discrete state $S_t \in \{1, \ldots, N\}$ by a real variable $I_t \in [1,\, N]$. We think of I_t as a point on the I-axis representing progress through the sound. Such a model, with a continuous-valued hidden-state variable sampled in time, constitutes a nonlinear dynamical system.

Another option is to let I_t take values on a finite, discrete space $\widehat{I} \subset [1,\, N]$ with significantly more than N points. This paper uses such a construction, though with an eye towards continuous state spaces in future work.

In either case, parameters shall remain associated with *integer* state-space positions n (like in regular HMMs), acting as templates for local process behavior. Therefore the greater-cardinality state space does not require additional parameters to be introduced.

In a discrete HMM, S_t follows a Markovian random walk on $\{1, \ldots, N\}$, where the next-step distribution may depend on the current state. We similarly let the intermediate state I_t perform a random walk on $\widehat{I}$, but with fractional steps $\Delta I_{t+1}{=}I_{t+1}{-}I_t$ to allow incremental progress. We initialize with $I_0{=}1$, terminate when $I_{t+1}{\geq}N$, and require $\Delta I_t{\geq}0$ to get a left-to-right process.

For this paper, we opt for a uniformly discretized state space $I_t \in \widehat{I} = \{1,\, 1+\delta,\, 1+2\delta, \ldots, N\}$ and a simple next-step distribution $\Delta I_{t+1} \in \{\delta,\, 2\delta,\, 4\delta\}$. The longest and shortest possible durations then differ by a factor four. We set the resolution δ such that the number of medium-length steps ($\Delta I_t = 2\delta$) from $I_t = 1$ to N approximately equals the mean training sequence length, so that, on average, there are two fractional states per frame.

3.2. Intermediate state parameters

The parameters tied to an HMM state n act as templates for the local process properties when $S_t = n$. For an

intermediate state I_t, properties of the process are in-between these templates as well. To define intermediate parameters we use an N-vector $\boldsymbol{w}\,(i) \geq \mathbf{0}$ of *weight functions* such that the influence of template n when $I_t = i$ is weighted by $w_n\,(i)$. We require $\sum_{n=1}^{N} w_n\,(i) = 1\,\forall i$.

In this work we only consider fixed weight functions satisfying $w_n\,(i = n) = 1$ and $w_n\,(i = n + r) = 0$ for $|r| \geq 1$. At most two $\boldsymbol{w}$-components may then be nonzero for any i. For the experiments, we further restrict ourselves to triangular w-functions; smoother behavior than piecewise linear is possible by using smoother w.

Each template is associated with a mean vector $\boldsymbol{\mu}_n$ and a covariance matrix $\boldsymbol{\Sigma}_n = \operatorname{diag}(\boldsymbol{\sigma}_n)^2$. (We assume independent components.) For positions i between templates we let $\boldsymbol{w}\,(i)$ define intermediate output parameters $\boldsymbol{\mu}\,(i) = \sum_{n=1}^{N} w_n\,(i)\,\boldsymbol{\mu}_n$ and $\boldsymbol{\sigma}\,(i) = \sum_{n=1}^{N} w_n\,(i)\,\boldsymbol{\sigma}_n$. This linear interpolation ensures that the output mode, mean, and variance all vary continuously in $\boldsymbol{w}$ between the extremes defined by the templates.

For dynamics we parameterize the ΔI_{t+1}-distribution using $P\,(\Delta I_{t+1} = \delta) = p_-$ and $P\,(\Delta I_{t+1} = 4\delta) = p_+$. This yields two dynamics parameters per template, one more than in regular left-right HMMs. Intermediate parameters are found by interpolating p_- and p_+ using $\boldsymbol{w}\,(i)$. These dynamics parameters govern the state variable I_t evolution speed, indirectly determining phone durations.

4. Training intermediate-state HMMs

Probabilistic sequence models are first trained on example data to estimate parameters. Thereafter one may sample from the model for generative tasks, or for discriminative tasks compute the log-probability of some data, or its most likely (Viterbi) corresponding state sequence.

In the case of continuous state spaces, it is straightforward to sample data from intermediate-state models, but training and other tasks necessitate approximations. EM-based ML-parameter estimation is possible in theory, but the hidden-state distribution inferred from the training data $\{\underline{\boldsymbol{x}}\}$ in the E-step is a general continuous distribution with no simple numerical representation.

One solution is to discretize the I-axis, and then train the model as a discrete HMM with parameters tied to the N templates. This offers guaranteed ascent, and thus convergence, with linear complexity in the amount of data and adjustable approximation accuracy. The idea can also be used to estimate log-probabilities and Viterbi sequences.

As our concrete model earlier has a discrete state space, it can be trained as a tied HMM without approximation.

4.1. Updating template output parameters

Generalized EM update formulas are derived by optimizing $Q\left(\boldsymbol{\theta}',\,\boldsymbol{\theta}\right) = \mathbb{E}\left(\ln P\left(\{\underline{\boldsymbol{x}},\,\underline{\boldsymbol{I}}\} \mid \boldsymbol{\theta}'\right) \mid \underline{\boldsymbol{x}},\,\boldsymbol{\theta}\right)$. For hidden-state models, Q separates into a part for the output parameters and one for the dynamics. With independent components, the output part further breaks down into one function for each dimension. These have the form

$$Q_k\left(\boldsymbol{\mu}'_k,\boldsymbol{\sigma}'_k\right)=\text{const.}-\sum_{m=1}^{M}\sum_{t=1}^{T_m}\sum_{g=1}^{G}\gamma_{gt}^{(m)}\ln(\boldsymbol{\sigma}'_k\boldsymbol{w}(\widehat{I}_g)) - \tfrac{1}{2}\sum_{m=1}^{M}\sum_{t=1}^{T_m}\sum_{g=1}^{G}\gamma_{gt}^{(m)}\left(\frac{x_{kt}^{(m)}-\boldsymbol{\mu}'_k\boldsymbol{w}(\widehat{I}_g)}{\boldsymbol{\sigma}'_k\boldsymbol{w}(\widehat{I}_g)}\right)^2, \quad (1)$$

where $\gamma_{gt}^{(m)} = P(I_t = \widehat{I}_g \mid \underline{\boldsymbol{x}}^{(m)})$ are computed by the E-step, the row vectors $\boldsymbol{\mu}'_k$ and $\boldsymbol{\sigma}'_k$ represent hypothetical new parameters in dimension k, and g is an index into $\widehat{I}$.

It is not feasible to optimize Q_k analytically, but it is possible to find the optimal mean parameters under fixed standard deviations by solving the $N \times N$ linear system $\boldsymbol{U}^{(k)}\boldsymbol{\mu}_k^{(\text{new})} = \boldsymbol{b}^{(k)}$ defined through

$$\left(\boldsymbol{U}^{(k)}\right)_{no}=\sum_{g=1}^{G}\frac{w_{ng}w_{og}}{\sigma_{kg}^2}\overline{\gamma}_g \quad (2)$$

$$\left(\boldsymbol{b}^{(k)}\right)_{n}=\sum_{g=1}^{G}\frac{w_{ng}}{\sigma_{kg}^2}\sum_{m=1}^{M}\sum_{t=1}^{T_m}\gamma_{gt}^{(m)}x_{kt}^{(m)}. \quad (3)$$

Here $w_{ng} = w_n(\widehat{I}_g)$ is weight n at $\widehat{I}_g$, $\sigma_{kg}^2 = \sigma_k^2(\widehat{I}_g)$ is the local variance, and $\overline{\gamma}_g = \sum_{m=1}^{M}\sum_{t=1}^{T_m}\gamma_{gt}^{(m)}$.

Standard deviations are difficult to solve for even if means are fixed. One can use Newton's method to find an update step $\boldsymbol{\sigma}_k^{(\delta)}$ through $\boldsymbol{H}^{(k)}\boldsymbol{\sigma}_k^{(\delta)} = -\boldsymbol{\nabla}^{(k)}$, where

$$\left(\boldsymbol{H}^{(k)}\right)_{no}=\sum_{g=1}^{G}\frac{w_{ng}w_{og}}{\sigma_{kg}^2}\left(\overline{\gamma}_g-\sum_{m=1}^{M}\sum_{t=1}^{T_m}\frac{3\gamma_{gt}^{(m)}}{\sigma_{kg}^2}\left(x_{kt}^{(m)}-\mu_{kg}\right)^2\right)$$

$$\left(\boldsymbol{\nabla}^{(k)}\right)_{n}=\sum_{g=1}^{G}\frac{w_{ng}}{\sigma_{kg}}\left(\left(\sum_{m=1}^{M}\sum_{t=1}^{T_m}\frac{\gamma_{gt}^{(m)}}{\sigma_{kg}^2}\left(x_{kt}^{(m)}-\mu_{kg}\right)^2\right)-\overline{\gamma}_g\right)$$

are the Hessian and the gradient in $\boldsymbol{\sigma}'_k$ evaluated at the current parameter estimates. Gradient descent can be used where $\boldsymbol{H}^{(k)}$ is indefinite or $\boldsymbol{\sigma}_k^{(\delta)}$ does not increase Q_k.

4.2. Updating state evolution parameters

The Q-function for the state evolution parameters is

$$Q_\xi\left(\boldsymbol{p}'_-,\boldsymbol{p}'_+\right)=\sum_{g=1}^{G}\left(\overline{\xi}_{g,g+1}\ln\left(\boldsymbol{p}'_-\boldsymbol{w}(\widehat{I}_g)\right)+\overline{\xi}_{g,g+4}\ln\left(\boldsymbol{p}'_+\boldsymbol{w}(\widehat{I}_g)\right)\right) + \sum_{g=1}^{G}\overline{\xi}_{g,g+2}\ln\left(1-\boldsymbol{p}'_-\boldsymbol{w}(\widehat{I}_g)-\boldsymbol{p}'_+\boldsymbol{w}(\widehat{I}_g)\right), \quad (4)$$

where $\overline{\xi}_{gh}=\sum_{m=1}^{M}\sum_{t=0}^{T_m}P(I_t = \widehat{I}_g, I_{t+1} = \widehat{I}_h \mid \underline{\boldsymbol{x}}^{(m)})$ from the E-step, and the $\boldsymbol{p}'$s are row vectors. Again, Newton's method (local quadratic approximation) is used to update the parameters. For $\boldsymbol{p}'_-$ the relevant quantities are

$$\left(\boldsymbol{H}^{-}\right)_{no}=-\sum_{g=1}^{G}w_{ng}w_{og}\left(\frac{\overline{\xi}_{g,g+1}}{(\boldsymbol{p}_-\boldsymbol{w}(\widehat{I}_g))^2}+\frac{\overline{\xi}_{g,g+2}}{(1-\boldsymbol{p}'_-\boldsymbol{w}(\widehat{I}_g)-\boldsymbol{p}'_+\boldsymbol{w}(\widehat{I}_g))^2}\right)$$

$$\left(\boldsymbol{\nabla}^{-}\right)_{n}=\sum_{g=1}^{G}w_{ng}\left(\frac{\overline{\xi}_{g,g+1}}{\boldsymbol{p}_-\boldsymbol{w}(\widehat{I}_g)}-\frac{\overline{\xi}_{g,g+2}}{1-\boldsymbol{p}'_-\boldsymbol{w}(\widehat{I}_g)-\boldsymbol{p}'_+\boldsymbol{w}(\widehat{I}_g)}\right);$$

the expressions for updating $\boldsymbol{p}_+$ for fixed $\boldsymbol{p}_-$ are similar.

5. Experiments

We now present experimental results showing that the proposed models address the issues listed in section 2.1.

5.1. Setup

The experiments used eight examples of the utterance "titta bilen" (Swedish for "look, the car") from speaker Olov in the CAREGIVER corpus [9]. Utterances were processed by the STRAIGHT system [10], and the resulting filter and aperiodicity spectra converted to 40 MFCCs each. With log pitch this yielded a $K = 81$-element fea-

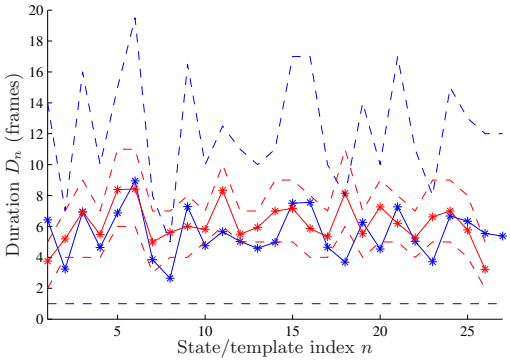

Figure 1: *State duration distributions of the intermediate-state model (red) and the discrete HMM (blue). The mean is solid while* 0.1 *and* 0.9 *quantiles are dashed.*

ture vector, which was sampled at 200 frames per second. Synthesis was performed by reversing the mappings.

We trained two models on the data: one intermediate-state model as above with $N = 26$ templates[1] and step length $\delta = \frac{1}{12}$, and a standard discrete finite-duration left-right HMM (not HSMM) with Gaussian outputs. The HMM was created with $N + 1$ states, to ensure it benefited from the advantage of having more parameters.

5.2. Results

To evaluate the models as stochastic descriptions of speech, we sampled 1000 state-space trajectories from each. Figure 1 graphs the resulting state durations. HMMs are known to achieve reasonable mean durations, but exhibit wide duration variation. The time spent near each state in the intermediate-state model has a similar mean profile, but spans a more natural range.

We also computed the average energy (variance) of the Gaussian random deviations from the mean along trajectories. The intermediate-state model had a lower average noise variance in the pitch feature contour, 0.16 compared to 0.18, meaning that more training data variation was explained. However, the residual energy is substantial also for intermediate-state HMMs. This is likely due to systematic variation, for instance pitch contours with different offsets, manifesting itself as correlated residuals. This energy cannot be eliminated even by following the mean contour perfectly.

Figure 2 shows short-time spectrogram representations of sequences sampled from each model, below a training example. It is evident that intermediate-state models reproduce gradual transitions in intensity and formants better than conventional HMMs, and with durations noticeably more similar to the reference utterance.

6. Conclusions and future work

We have described a number of shortcomings of conventional discrete-state HMMs for modeling path-following

[1]This yields about six frames per state/template and training example, and around two states per phone, plus some extra states for silences.

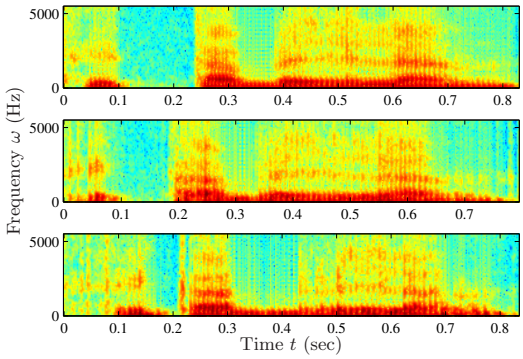

Figure 2: *Sample spectra of reference (top), intermediate-state model (middle), and discrete HMM (bottom).*

processes such as motion capture data or speech. We proposed to solve the problems by allowing intermediate states in the HMM, and described a concrete mathematical model embodying the idea. An experiment confirmed that, versus conventional HMMs, the new method produced more natural durations, was capable of gradual transitions between behaviors, and exhibited reduced unnatural between-frame random variation.

A number of aspects of our approach suggest themselves as worthy targets for future research. As our proposal cannot replicate the correlated residuals common in real-world processes, we are investigating improvements that directly address this issue. We are also considering alternative parameter estimation procedures.

7. References

[1] L. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc IEEE*, vol. 77, no. 2, pp. 257–286, Feb. 1989.

[2] P. Woodland, C. Leggetter, J. Odell, V. Valtchev, and S. Young, "The 1994 HTK large vocabulary speech recognition system," in *Proc ICASSP'95*, May 1995, pp. 73–76.

[3] H. Zen, T. Nose, J. Yamagishi, S. Sako, T. Masuko, A. W. Black, and K. Tokuda, "The HMM-based speech synthesis system version 2.0," in *Proc ISCA SSW6*, vol. 6, Aug. 2007, pp. 294–299.

[4] H. Zen, K. Tokuda, and T. Kitamura, "Reformulating the HMM as a trajectory model by imposing explicit relationships between static and dynamic feature vector sequences," *Comput Speech Lang*, vol. 21, no. 1, pp. 153–173, 2007.

[5] K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, and T. Kitamura, "Speech parameter generation algorithms for HMM-based speech synthesis," in *Proc ICASSP'00*, Jun. 2000, pp. 1315–1318.

[6] V. Digalakis, J. Rohlicek, and M. Ostendorf, "A dynamical system approach to continuous speech recognition," in *Proc ICASSP'91*, 1991, pp. 289–292.

[7] H. Richards and J. Bridle, "The HDM: a segmental hidden dynamic model of coarticulation," in *Proc ICASSP'99*, 1999, pp. 357–360.

[8] D. Tweed, R. Fisher, J. Bins, and T. List, "Efficient hidden semi-Markov model inference for structured video sequences," in *Proc VSPETS'05*, 2005, pp. 247–254.

[9] T. Altosaar, L. ten Bosch, G. Aimetti, C. Koniaris, K. Demuynck, and H. van den Heuvel, "A speech corpus for modeling language acquisition: CAREGIVER," in *Proc LREC 2010*, vol. 7, 2010, pp. 1062–1068.

[10] H. Kawahara, "STRAIGHT, exploitation of the other aspect of VOCODER: Perceptually isomorphic decomposition of speech sounds," *Acoust Sci & Tech*, vol. 27, no. 6, pp. 349–353, 2006.