Learning Structured Models for Phone Recognition

Slav Petrov **Adam Pauls** **Dan Klein**
Computer Science Department, EECS Divison
University of California at Berkeley
Berkeley, CA, 94720, USA
{petrov,adpauls,klein}@cs.berkeley.edu

Abstract

We present a maximally streamlined approach to learning HMM-based acoustic models for automatic speech recognition. In our approach, an initial monophone HMM is iteratively refined using a split-merge EM procedure which makes no assumptions about subphone structure or context-dependent structure, and which uses only a single Gaussian per HMM state. Despite the much simplified training process, our acoustic model achieves state-of-the-art results on phone classification (where it outperforms almost all other methods) and competitive performance on phone recognition (where it outperforms standard CD triphone / subphone / GMM approaches). We also present an analysis of what is and is not learned by our system.

1 Introduction

Continuous density hidden Markov models (HMMs) underlie most automatic speech recognition (ASR) systems in some form. While the basic algorithms for HMM learning and inference are quite general, acoustic models of speech standardly employ rich speech-specific structures to improve performance. For example, it is well known that a *monophone* HMM with one state per phone is too coarse an approximation to the true articulatory and acoustic process. The HMM state space is therefore refined in several ways. To model phone-internal dynamics, phones are split into *beginning*, *middle*, and *end* subphones (Jelinek, 1976). To model cross-phone coarticulation, the states of the HMM are refined by splitting the phones into context-dependent *triphones*. These states are then re-clustered (Odell, 1995) and the parameters of their observation distributions are tied back together (Young and Woodland, 1994). Finally, to model complex emission densities, states emit mixtures of multivariate Gaussians. This standard structure is shown schematically in Figure 1. While this rich structure is phonetically well-motivated and empirically successful, so much structural bias may be unnecessary, or even harmful. For example in the domain of syntactic parsing with probabilistic context-free grammars (PCFGs), a surprising recent result is that automatically induced grammar refinements can outperform sophisticated methods which exploit substantial manually articulated structure (Petrov et al., 2006).

In this paper, we consider a much more automatic, data-driven approach to learning HMM structure for acoustic modeling, analagous to the approach taken by Petrov et al. (2006) for learning PCFGs. We start with a minimal monophone HMM in which there is a single state for each (context-independent) phone. Moreover, the emission model for each state is a single multivariate Gaussian (over the standard MFCC acoustic features). We then iteratively refine this minimal HMM through state splitting, adding complexity as needed. States in the refined HMMs are always substates of the original HMM and are therefore each identified with a unique base phone. States are split, estimated, and (perhaps) merged, based on a likelihood criterion. Our model never allows explicit Gaussian mixtures, though substates may develop similar distributions and thereby emulate such mixtures.

In principle, discarding the traditional structure can either help or hurt the model. Incorrect prior splits can needlessly fragment training data and incorrect prior tying can limit the model's expressivity. On the other hand, correct assumptions can increase the efficiency of the learner. Empirically,

Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pp. 897–905, Prague, June 2007.

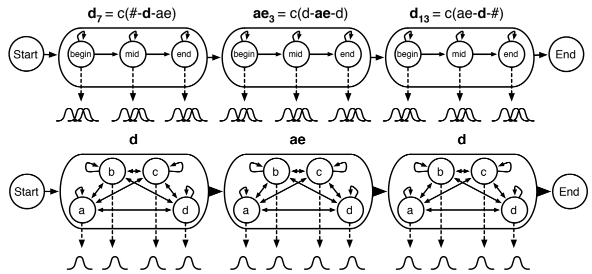

Figure 1: Comparison of the standard model to our model (here shown with $k = 4$ subphones per phone) for the word *dad*. The dependence of subphones across phones in our model is not shown, while the context clustering in the standard model is shown only schematically.

we show that our automatic approach outperforms classic systems on the task of phone recognition on the TIMIT data set. In particular, it outperforms standard state-tied triphone models like Young and Woodland (1994), achieving a phone error rate of 26.4% versus 27.7%. In addition, our approach gives state-of-the-art performance on the task of phone classification on the TIMIT data set, suggesting that our learned structure is particularly effective at modeling phone-internal structure. Indeed, our error rate of 21.4% is outperformed only by the recent structured margin approach of Sha and Saul (2006). It remains to be seen whether these positive results on acoustic modeling will facilitate better word recognition rates in a large vocabulary speech recognition system.

We also consider the structures learned by the model. Subphone structure is learned, similar to, but richer than, standard begin-middle-end structures. Cross-phone coarticulation is also learned, with classic phonological classes often emerging naturally.

Many aspects of this work are intended to simplify rather than further articulate the acoustic process. It should therefore be clear that the basic techniques of splitting, merging, and learning using EM are not in themselves new for ASR. Nor is the basic latent induction method new (Matsuzaki et al., 2005; Petrov et al., 2006). What is novel in this paper is (1) the construction of an automatic system for acoustic modeling, with substantially streamlined structure, (2) the investigation of variational inference for such a task, (3) the analysis of the kinds of structures learned by such a system, and (4) the empirical demonstration that such a system is not only competitive with the traditional approach, but can indeed outperform even very recent work on some preliminary measures.

2 Learning

In the following, we propose a greatly simplified model that does not impose any manually specified structural constraints. Instead of specifying structure *a priori*, we use the Expectation-Maximization (EM) algorithm for HMMs (Baum-Welch) to automatically induce the structure in a way that maximizes data likelihood.

In general, our training data consists of sets of acoustic observation sequences and phone level transcriptions $\mathbf{r}$ which specify a sequence of phones from a set of phones Y, but does not label each time frame with a phone. We refer to an observation sequence as $\mathbf{x} = x_1, \ldots, x_T$ where $x_i \in \mathbb{R}^{39}$ are standard MFCC features (Davis and Mermelstein, 1980). We wish to induce an HMM over a set of states S for which we also have a function $\pi : S \rightarrow Y$ that maps every state in S to a phone in Y. Note that in the usual formulation of the EM algorithm for HMMs, one is interested in learning HMM parameters θ that maximize the likelihood of the observations $\mathrm{P}(\mathbf{x}|\theta)$; in contrast, we aim to maximize the joint probability of our observations and phone transcriptions $\mathrm{P}(\mathbf{x}, \mathbf{r}|\theta)$ or observations and phone sequences $\mathrm{P}(\mathbf{x}, \mathbf{y}|\theta)$ (see below). We now describe this relatively straightforward modification of the EM algorithm.

2.1 The Hand-Aligned Case

For clarity of exposition we first consider a simplified scenario in which we are given hand-aligned phone labels $\mathbf{y} = y_1, \ldots, y_T$ for each time t, as is the case for the TIMIT dataset. Our procedure does not require such extensive annotation of the training data and in fact gives better performance when the exact transition point between phones are not prespecified but learned.

We define forward and backward probabilities (Rabiner, 1989) in the following way: the forward probability is the probability of observing the sequence $x_1, \ldots, x_t$ with transcription $y_1, \ldots, y_t$ and

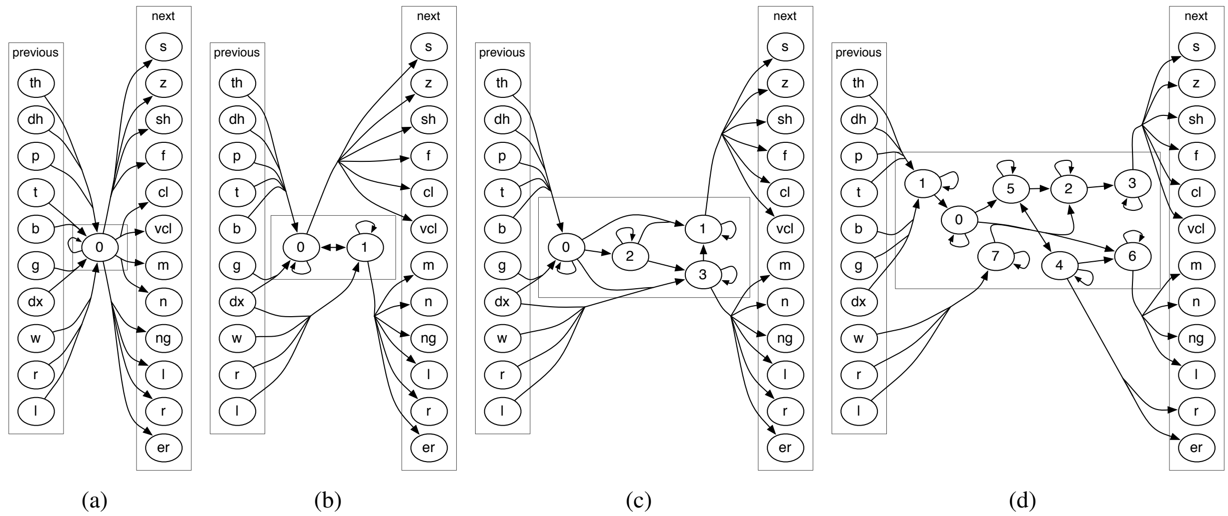

Figure 2: Iterative refinement of the /ih/ phone with $1, 2, 4, 8$ substates.

ending in state s at time t:

$$\alpha_t(s) = \mathrm{P}(x_1, \ldots, x_t, y_1, \ldots y_t, s_t = s|\lambda),$$

and the backward probability is the probability of observing the sequence $x_{t+1}, \ldots, x_T$ with transcription $y_{t+1}, \ldots, y_T$, given that we start in state s at time t:

$$\beta_t(s) = \mathrm{P}(x_{t+1}, \ldots, x_T, y_{t+1}, \ldots, y_T|s_t = s, \lambda),$$

where λ are the model parameters. As usual, we parameterize our HMMs with $a_{ss'}$, the probability of transitioning from state s to s', and $b_s(x) \sim \mathcal{N}(\mu_s, \Sigma_s)$, the probability emitting the observation x when in state s.

These probabilities can be computed using the standard forward and backward recursions (Rabiner, 1989), except that at each time t, we only consider states s_t for which $\pi(s_t) = y_t$, because we have hand-aligned labels for the observations. These quantities also allow us to compute the posterior counts necessary for the E-step of the EM algorithm.

2.2 Splitting

One way of inducing arbitrary structural annotations would be to split each HMM state in into m substates, and re-estimate the parameters for the split HMM using EM. This approach has two major drawbacks: for larger m it is likely to converge to poor local optima, and it allocates substates uniformly across all states, regardless of how much annotation is required for good performance.

To avoid these problems, we apply a hierarchical parameter estimation strategy similar in spirit to the work of Sankar (1998) and Ueda et al. (2000), but here applied to HMMs rather than to GMMs. Beginning with the baseline model, where each state corresponds to one phone, we repeatedly split and re-train the HMM. This strategy ensures that each split HMM is initialized "close" to some reasonable maximum.

Concretely, each state s in the HMM is split in two new states s_1, s_2 with $\pi(s_1) = \pi(s_2) = \pi(s)$. We initialize EM with the parameters of the previous HMM, splitting every previous state s in two and adding a small amount of randomness $\epsilon \leq 1\%$ to its transition and emission probabilities to break symmetry:

$$a_{s_1 s'} \propto a_{ss'} + \epsilon,$$

$$b_{s_1}(o) \sim \mathcal{N}(\mu_s + \epsilon, \Sigma_s),$$

and similarly for s_2. The incoming transitions are split evenly.

We then apply the EM algorithm described above to re-estimate these parameters before performing subsequent split operations.

2.3 Merging

Since adding substates divides HMM statistics into many bins, the HMM parameters are effectively estimated from less data, which can lead to overfitting. Therefore, it would be to our advantage to split substates only where needed, rather than splitting them all.

We realize this goal by merging back those splits $s \rightarrow s_1 s_2$ for which, if the split were reversed, the loss in data likelihood would be smallest. We approximate the loss in data likelihood for a merge $s_1 s_2 \rightarrow s$ with the following likelihood ratio (Petrov et al., 2006):

$$\Delta(s_1\ s_2 \rightarrow s) = \prod_{sequences} \prod_t \frac{\mathrm{P}^t(\mathbf{x}, \mathbf{y})}{\mathrm{P}(\mathbf{x}, \mathbf{y})}.$$

Here $\mathrm{P}(\mathbf{x}, \mathbf{y})$ is the joint likelihood of an emission sequence $\mathbf{x}$ and associated state sequence $\mathbf{y}$. This quantity can be recovered from the forward and backward probabilities using

$$\mathrm{P}(\mathbf{x}, \mathbf{y}) = \sum_{s:\pi(s)=y_t} \alpha_t(s) \cdot \beta_t(s).$$

$\mathrm{P}^t(\mathbf{x}, \mathbf{y})$ is an approximation to the same joint likelihood where states s_1 and s_2 are merged. We approximate the true loss by only considering merging states s_1 and s_2 at time t, a value which can be efficiently computed from the forward and backward probabilities. The forward score for the merged state s at time t is just the sum of the two split scores:

$$\hat{\alpha}_t(s) = \alpha_t(s_1) + \alpha_t(s_2),$$

while the backward score is a weighted sum of the split scores:

$$\hat{\beta}_t(s) = p_1 \beta_t(s_1) + p_2 \beta_t(s_2),$$

where p_1 and p_2 are the relative (posterior) frequencies of the states s_1 and s_2.

Thus, the likelihood after merging s_1 and s_2 at time t can be computed from these merged forward and backward scores as:

$$P^t(\mathbf{x}, \mathbf{y}) = \hat{\alpha}_t(s) \cdot \hat{\beta}_t(s) + \sum_{s'} \alpha_t(s') \cdot \beta_t(s')$$

where the second sum is over the other substates of x_t, i.e. $\{s' : \pi(s') = x_t, s' \notin \{s_1, s_2\}\}$. This expression is an approximation because it neglects interactions between instances of the same states at multiple places in the same sequence. In particular, since phones frequently occur with multiple consecutive repetitions, this criterion may vastly overestimate the actual likelihood loss. As such, we also implemented the exact criterion, that is, for each split, we formed a new HMM with s_1 and s_2 merged and calculated the total data likelihood. This method is much more computationally expensive, requiring a full forward-backward pass through the data for each potential merge, and was not found to produce noticeably better performance. Therefore, all experiments use the approximate criterion.

2.4 The Automatically-Aligned Case

It is straightforward to generalize the hand-aligned case to the case where the phone transcription is known, but no frame level labeling is available. The main difference is that the phone boundaries are not known in advance, which means that there is now additional uncertainty over the phone states. The forward and backward recursions must thus be expanded to consider all state sequences that yield the given phone transcription. We can accomplish this with standard Baum-Welch training.

3 Inference

An HMM over refined subphone states $s \in S$ naturally gives posterior distributions $\mathrm{P}(\mathbf{s}|\mathbf{x})$ over *sequences* of states $\mathbf{s}$. We would ideally like to extract the transcription $\mathbf{r}$ of underlying phones which is most probable according to this posterior[1]. The transcription is two stages removed from $\mathbf{s}$. First, it collapses the distinctions between states s which correspond to the same phone $y = \pi(s)$. Second, it collapses the distinctions between where phone transitions exactly occur. Viterbi state sequences can easily be extracted using the basic Viterbi algorithm. On the other hand, finding the best phone sequence or transcription is intractable.

As a compromise, we extract the phone sequence (not transcription) which has highest probability in a variational approximation to the true distribution (Jordan et al., 1999). Let the true posterior distribution over phone sequences be $\mathrm{P}(\mathbf{y}|\mathbf{x})$. We form an approximation $\mathrm{Q}(\mathbf{y}) \approx \mathrm{P}(\mathbf{y}|\mathbf{x})$, where Q is an approximation specific to the sequence $\mathbf{x}$ and factor-

[1] Remember that by "transcription" we mean a sequence of phones with duplicates removed.

izes as:

$$Q(\mathbf{y}) = \prod_t q(t, x_t, y_{t+1}).$$

We would like to fit the values q, one for each time step and state-state pair, so as to make Q as close to P as possible:

$$\min_q KL(P(\mathbf{y}|\mathbf{x})||Q(\mathbf{y})).$$

The solution can be found analytically using Lagrange multipliers:

$$q(t, y, y') = \frac{P(Y_t = y, Y_{t+1} = y'|\mathbf{x})}{P(Y_t = y|\mathbf{x})}.$$

where we have made the position-specific random variables Y_t explicit for clarity. This approximation depends only on our ability to calculate posteriors over phones or phone-phone pairs at individual positions t, which is easy to obtain from the state posteriors, for example:

$$P(Y_t = y, Y_{t+1} = y'|\mathbf{x}) = \frac{\sum_{s:\pi(s)=y} \sum_{s':\pi(s')=y'} \alpha_t(s) a_{ss'} b_{s'}(x_t) \beta_{t+1}(s')}{P(\mathbf{x})}$$

Finding the Viterbi *phone* sequence in the approximate distribution Q, can be done with the Forward-Backward algorithm over the lattice of q values.

4 Experiments

We tested our model on the TIMIT database, using the standard setups for phone recognition and phone classification. We partitioned the TIMIT data into training, development, and (core) test sets according to standard practice (Lee and Hon, 1989; Gunawardana et al., 2005; Sha and Saul, 2006). In particular, we excluded all *sa* sentences and mapped the 61 phonetic labels in TIMIT down to 48 classes before training our HMMs. At evaluation, these 48 classes were further mapped down to 39 classes, again in the standard way.

MFCC coefficients were extracted from the TIMIT source as in Sha and Saul (2006), including delta and delta-delta components. For all experiments, our system and all baselines we implemented used *full covariance* when parameterizing emission models.[2] All Gaussians were endowed with weak inverse Wishart priors with zero mean and identity covariance.[3]

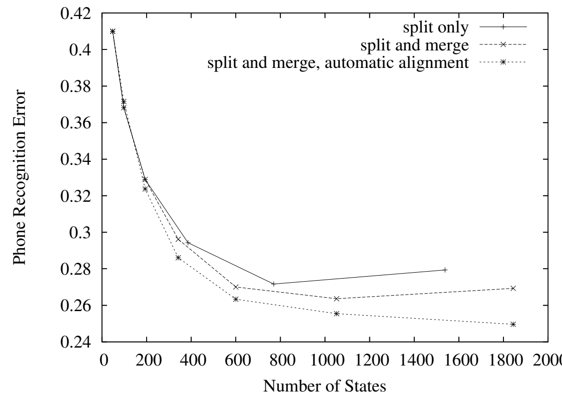

Figure 3: Phone recognition error for models of increasing size

4.1 Phone Recognition

In the task of phone recognition, we fit an HMM whose output, with subsequent states collapsed, corresponds to the training transcriptions. In the TIMIT data set, each frame is manually phone-annotated, so the only uncertainty in the basic setup is the identity of the (sub)states at each frame.

We therefore began with a single state for each phone, in a fully connected HMM (except for special treatment of dedicated start and end states). We incrementally trained our model as described in Section 2, with up to 6 split-merge rounds. We found that reversing 25% of the splits yielded good overall performance while maintaining compactness of the model.

We decoded using the variational decoder described in Section 3. The output was then scored against the reference phone transcription using the standard string edit distance.

During both training and decoding, we used "flattened" emission probabilities by exponentiating to some $0 < \gamma < 1$. We found the best setting for γ to be 0.2, as determined by tuning on the development set. This flattening compensates for the non-

[2]Most of our findings also hold for diagonal covariance Gaussians, albeit the final error rates are 2-3% higher.

[3]Following previous work with PCFGs (Petrov et al., 2006), we experimented with smoothing the substates towards each other to prevent overfitting, but we were unable to achieve any performance gains.

Method	Error Rate
State-Tied Triphone HMM (Young and Woodland, 1994)	27.7%[1]
Gender Dependent Triphone HMM (Lamel and Gauvain, 1993)	27.1%[1]
This Paper	**26.4%**
Bayesian Triphone HMM (Ming and Smith, 1998)	25.6%
Heterogeneous classifiers (Halberstadt and Glass, 1998)	24.4%

Table 1: Phone recognition error rates on the TIMIT core test from Glass (2003).
[1]These results are on a slightly easier test set.

independence of the frames, partially due to overlapping source samples and partially due to other unmodeled correlations.

Figure 3 shows the recognition error as the model grows in size. In addition to the basic setup described so far (*split and merge*), we also show a model in which merging was not performed (*split only*). As can be seen, the merging phase not only decreases the number of HMM states at each round, but also improves phone recognition error at each round.

We also compared our hierarchical *split only* model with a model where we directly split all states into 2^k substates, so that these models had the same number of states as a a hierarchical model after k split and merge cycles. While for small k, the difference was negligible, we found that the error increased by 1% absolute for $k = 5$. This trend is to be expected, as the possible interactions between the substates grows with the number of substates.

Also shown in Figure 3, and perhaps unsurprising, is that the error rate can be further reduced by allowing the phone boundaries to drift from the manual alignments provided in the TIMIT training data. The *split and merge, automatic alignment* line shows the result of allowing the EM fitting phase to reposition each phone boundary, giving absolute improvements of up to 0.6%.

We investigated how much improvement in accuracy one can gain by computing the variational approximation introduced in Section 3 versus extracting the Viterbi state sequence and projecting that sequence to its phone transcription. The gap varies,

Method	Error Rate
GMM Baseline (Sha and Saul, 2006)	26.0%
HMM Baseline (Gunawardana et al., 2005)	25.1%
SVM (Clarkson and Moreno, 1999)	22.4%
Hidden CRF (Gunawardana et al., 2005)	21.7%
This Paper	**21.4%**
Large Margin GMM (Sha and Saul, 2006)	21.1%

Table 2: Phone classification error rates on the TIMIT core test.

but on a model with roughly 1000 states (5 split-merge rounds), the variational decoder decreases error from 26.5% to 25.6%. The gain in accuracy comes at a cost in time: we must run a (possibly pruned) Forward-Backward pass over the full state space S, then another over the smaller phone space Y. In our experiments, the cost of variational decoding was a factor of about 3, which may or may not justify a relative error reduction of around 4%.

The performance of our best model (split and merge, automatic alignment, and variational decoding) on the test set is 26.4%. A comparison of our performance with other methods in the literature is shown in Table 1. Despite our structural simplicity, we outperform state-tied triphone systems like Young and Woodland (1994), a standard baseline for this task, by nearly 2% absolute. However, we fall short of the best current systems.

4.2 Phone Classification

Phone classification is the fairly constrained task of classifying in isolation a sequence of frames which is known to span exactly one phone. In order to quantify how much of our gains over the triphone baseline stem from modeling context-dependencies and how much from modeling the inner structure of the phones, we fit separate HMM models for each phone, using the same split and merge procedure as above (though in this case only manual alignments are reasonable because we test on manual segmentations). For each test frame sequence, we compute the likelihood of the sequence from the forward probabilities of each individual phone HMM. The phone giving highest likelihood to the input was selected. The error rate is a simple fraction of test phones classified correctly.

Table 2 shows a comparison of our performance with that of some other methods in the literature. A minimal comparison is to a GMM with the same number of mixtures per phone as our model's maxi-

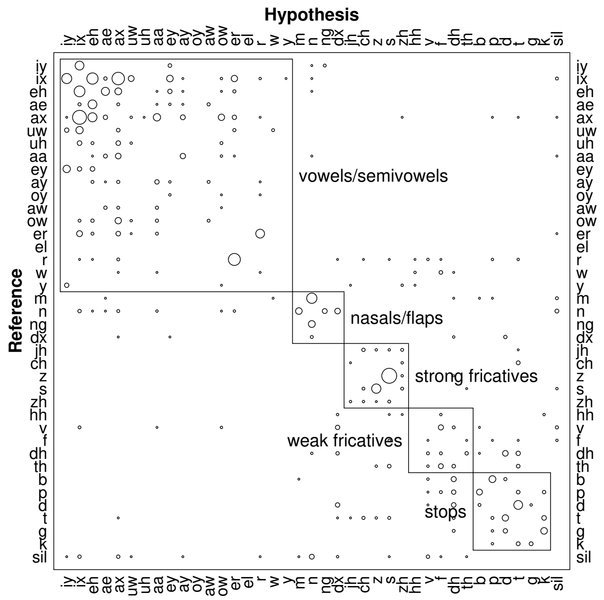

Figure 4: Phone confusion matrix. 76% of the substitutions fall within the shown classes.

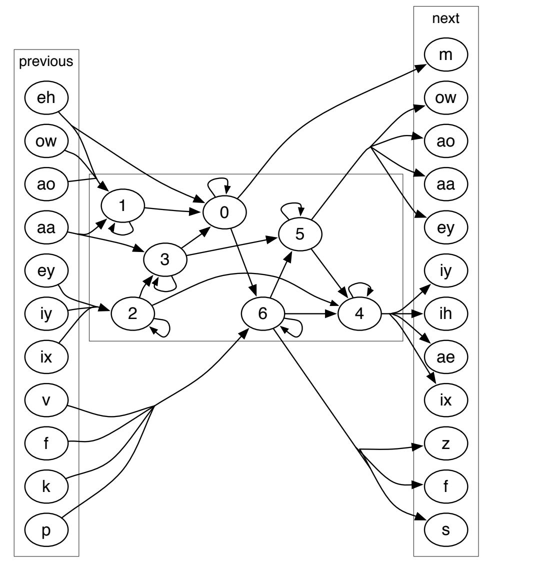

Figure 5: Phone contexts and subphone structure. The /l/ phone after 3 split-merge iterations is shown.

mum substates per phone. While these models have the same number of total Gaussians, in our model the Gaussians are correlated temporally, while in the GMM they are independent. Enforcing begin-middle-end HMM structure (see *HMM Baseline*) increases accuracy somewhat, but our more general model clearly makes better use of the available parameters than those baselines.

Indeed, our best model achieves a surprising performance of 21.4%, greatly outperforming other generative methods and achieving performance competitive with state-of-the-art discriminative methods. Only the recent structured margin approach of Sha and Saul (2006) gives a better performance than our model. The strength of our system on the classification task suggests that perhaps it is modeling phone-internal structure more effectively than cross-phone context.

5 Analysis

While the overall phone recognition and classification numbers suggest that our system is broadly comparable to and perhaps in certain ways superior to classical approaches, it is illuminating to investigate what is and is not learned by the model.

Figure 4 gives a confusion matrix over the substitution errors made by our model. The majority of the confusions are within natural classes. Some particularly frequent and reasonable confusions arise between the consonantal /r/ and the vocalic /er/ (the same confusion arises between /l/ and /el/, but the standard evaluation already collapses this distinction), the reduced vowels /ax/ and /ix/, the voiced and voiceless alveolar sibilants /z/ and /s/, and the voiced and voiceless stop pairs. Other vocalic confusions are generally between vowels and their corresponding reduced forms. Overall, 76% of the substitutions are within the broad classes shown in the figure.

We can also examine the substructure learned for the various phones. Figure 2 shows the evolution of the phone /ih/ from a single state to 8 substates during split/merge (no merges were chosen for this phone), using hand-alignment of phones to frames. These figures were simplified from the complete state transition matrices as follows: (1) adjacent phones' substates are collapsed, (2) adjacent phones are selected based on frequency and inbound probability (and forced to be the same across figures), (3) infrequent arcs are suppressed. In the first split, (b), a sonorant / non-sonorant distinction is learned over adjacent phones, along with a state chain which captures basic duration (a self-looping state gives an exponential model of duration; the sum of two such states is more expressive). Note that the natural classes interact with the chain in a way which allows duration to depend on context. In further refinements, more structure is added, including a two-track path in (d) where one track captures the distinct effects on higher formants of r-coloring and nasalization. Figure 5 shows the corresponding diagram for /l/, where some merging has also occurred. Different natural classes emerge in this case, with, for example, preceding states partitioned into front/high vowels vs. rounded vowels vs. other vowels vs. consonants. Following states show a front/back distinction and a consonant distinction, and the phone /m/ is treated specially, largely because the /lm/ sequence tends to shorten the /l/ substantially. Note again how context, internal structure, and duration are simultaneously modeled. Of course, it should be emphasized that *post hoc* analysis of such structure is a simplification and prone to seeing what one expects; we present these examples to illustrate the broad kinds of patterns which are detected.

As a final illustration of the nature of the learned models, Table 3 shows the number of substates allocated to each phone by the split/merge process (the maximum is 32 for this stage) for the case of hand-aligned (left) as well as automatically-aligned (right) phone boundaries. Interestingly, in the hand-aligned case, the vowels absorb most of the complexity since many consonantal cues are heavily evidenced on adjacent vowels. However, in the automatically-aligned case, many vowel frames with substantial consontant coloring are re-allocated to those adjacent consonants, giving more complex consonants, but comparatively less complex vowels.

6 Conclusions

We have presented a minimalist, automatic approach for building an accurate acoustic model for phonetic classification and recognition. Our model does not require any *a priori* phonetic bias or manual specification of structure, but rather induces the structure in an automatic and streamlined fashion. Starting from a minimal monophone HMM, we automatically learn models that achieve highly competitive performance. On the TIMIT phone recognition task our model clearly outperforms standard state-tied triphone models like Young and Woodland (1994). For phone classification, our model

Vowels		
aa	31	32
ae	32	17
ah	31	8
ao	32	23
aw	18	6
ax	18	3
ay	32	28
eh	32	16
el	6	4
en	4	3
er	32	31
ey	32	30
ih	32	11
ix	31	16
iy	31	32
ow	26	10
oy	4	4
uh	5	2
uw	21	8
Consonants		
b	2	32
ch	13	30
d	2	14
dh	6	31
dx	2	3
f	32	32
g	2	15
hh	3	5
jh	3	16
k	30	32
l	25	32
m	25	25
n	29	32
ng	3	4
p	5	24
r	32	32
s	32	32
sh	30	32
t	24	32
th	8	11
v	23	11
w	10	21
y	3	7
z	31	32
zh	2	2
Other		
epi	2	4
sil	32	32
vcl	29	30
cl	31	32

Table 3: Number of substates allocated per phone. The left column gives the number of substates allocated when training on manually aligned training sequences, while the right column gives the number allocated when we automatically determine phone boundaries.

achieves performance competitive with the state-of-the-art discriminative methods (Sha and Saul, 2006), despite being generative in nature. This result together with our analysis of the context-dependencies and substructures that are being learned, suggests that our model is particularly well suited for modeling phone-internal structure. It does, of course remain to be seen if and how these benefits can be scaled to larger systems.

References

P. Clarkson and P. Moreno. 1999. On the use of Support Vector Machines for phonetic classification. In *ICASSP '99*.

S. B. Davis and P. Mermelstein. 1980. Comparison of parametric representation for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 28(4).

J. Glass. 2003. A probabilistic framework for segment-based speech recognition. *Computer Speech and Language*, 17(2).

A. Gunawardana, M. Mahajan, A. Acero, and J. Platt. 2005. Hidden Conditional Random Fields for phone recognition. In *Eurospeech '05*.

A. K. Halberstadt and J. R. Glass. 1998. Heterogeneous measurements and multiple classifiers for speech recognition. In *ICSLP '98*.

F. Jelinek. 1976. Continuous speech recognition by statistical methods. *Proceedings of the IEEE*.

M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul. 1999. An introduction to variational methods for graphical models. *Learning in Graphical Models*.

L. Lamel and J. Gauvain. 1993. Cross-lingual experiments with phone recognition. In *ICASSP '93*.

K. F. Lee and H. W. Hon. 1989. Speaker-independent phone recognition using Hidden Markov Models. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 37(11).

T. Matsuzaki, Y. Miyao, and J. Tsujii. 2005. Probabilistic CFG with latent annotations. In *ACL '05*.

J. Ming and F.J. Smith. 1998. Improved phone recognition using Bayesian triphone models. In *ICASSP '98*.

J. J. Odell. 1995. *The Use of Context in Large Vocabulary Speech Recognition*. Ph.D. thesis, University of Cambridge.

S. Petrov, L. Barrett, R. Thibaux, and D. Klein. 2006. Learning accurate, compact, and interpretable tree annotation. In *COLING-ACL '06*.

L. Rabiner. 1989. A Tutorial on hidden Markov models and selected applications in speech recognition. In *IEEE*.

A. Sankar. 1998. Experiments with a Gaussian merging-splitting algorithm for HMM training for speech recognition. In *DARPA Speech Recognition Workshop '98*.

F. Sha and L. K. Saul. 2006. Large margin Gaussian mixture modeling for phonetic classification and recognition. In *ICASSP '06*.

N. Ueda, R. Nakano, Z. Ghahramani, and G. E. Hinton. 2000. Split and Merge EM algorithm for mixture models. *Neural Computation*, 12(9).

S. J. Young and P. C. Woodland. 1994. State clustering in HMM-based continuous speech recognition. *Computer Speech and Language*, 8(4).