RETHINKING DERIVED ACOUSTIC FEATURES IN SPEECH RECOGNITIO N

Article · November 2001

Source: CiteSeer

1 author:

Kevin S. Van Horn
Adobe Inc.
17 PUBLICATIONS **322** CITATIONS

SEE PROFILE

RETHINKING DERIVED ACOUSTIC FEATURES IN SPEECH RECOGNITION

Kevin S. Van Horn

North Dakota State University, Dept. of Computer Science, Fargo, ND 58105
email: Kevin.VanHorn@ndsu.nodak.edu

ABSTRACT

We present a new acoustic model for speech recognition that explicitly accounts for information omitted from current acoustic models: the definitions of derived acoustic features such as estimated derivatives. We incorporate this information using the method of maximum entropy. We find that, compared to the corresponding HMM, our model cuts an already low error rate about in half for a simple task. We also examine the consequences of ignoring the origin of derived features in CDHMM systems, showing that such an omission in the *acoustic* model can severely distort the effective *language* model.

1. INTRODUCTION

Current acoustic models for speech recognition fail to incorporate some significant, relevant information: the definition of derived acoustic features in terms of the base features. Estimated time derivatives of base features have proven useful as additional, derived features [1], with $x_{i,k}[t]$ (the estimated k-th derivative of feature i at time t) defined to be some linear function of the values $x_i[u]$ (base feature i at time u), $t - w \leq u \leq t + w$, for some w. Current practice is to simply use the combined feature set as the observables for a hidden Markov (or other) model, feigning ignorance of the origin of the derived features. In particular, *HMMs using derived features fail to assign zero probability to sequences of feature vectors in which the derived feature values do not match their definitions in terms of the base feature values.*

Current acoustic models are all *directed graphical models* [2, 3]: they decompose a joint distribution into a (partially ordered) product of conditional distributions for each individual model variable. Directed graphical models are incapable of accounting for the origin of derived features without making the derived features irrelevant to recognition [4].

We present an acoustic model that accounts for derived features using the method of maximum entropy [5]. We also investigate the effects of ignoring the origin of derived features; surprisingly, such an omission in the *acoustic* model can severely distort the effective *language* model.

Research supported in part by NSF EPSCoR.

2. A MAXIMUM-ENTROPY ACOUSTIC MODEL

We define the following model variables and notation:

- U is the utterance (sequence of words spoken).
- τ is the duration of the utterance.
- A *substate* is analogous to a (state, mixture component) pair for a CDHMM.
- q is the sequence of substates, q_t the substate at time t, q_0 an entry state, and $q_{\tau+1}$ an exit state.
- $\mathsf{a}(q, U)$ is 1 if (q_t, q_{t+1}) is an allowed substate transition for utterance U for all t, and 0 otherwise.
- x is the collection of all *base* acoustic feature values $x_i[t]$, $1 \leq t \leq \tau$.
- (b) is defined to be 1 if b is true, 0 otherwise.
- An *occupancy class* is a set of substates for which we pool occurrence statistics.
- A *transition class* is a set of substate transitions for which we pool occurrence statistics.
- A *mean-variance class* is a set of substates for which we pool feature statistics.

We wish to construct a joint distribution over U, q, and x satisfying a particular set of constraints on expected values $E[f_j(U, q, x)] = F_j$. If these constraints are all the information our distribution should express, then the principle of maximum entropy prescribes that we choose the distribution that maximizes the entropy (a measure of how spread-out and uncertain a distribution is) subject to the constraints.

To retain the separation between language model and acoustic model, one constraint is that $P(U) = \mathsf{LM}(U)$, a given language model. The remaining constraints use the same statistics gathered in CDHMM training:

- The expected number of transitions from a transition class T.
- The expected number of occurrences of a substate from occupancy class S.
- The expected sum of $x_{i,k}[t]$ over all time steps t at which the substate q_t is in mean-variance class G.
- The expected sum squared deviation of $x_{i,k}[t]$ from a fixed central value over all t for which $q_t \in G$.

The target values for the chosen statistics may be thought of as indirect parameters of the model. The occupancy, transition, and mean-variance classes implement parameter tying.

Let θ be the collection of all parameters of the acoustic model. The maximum-entropy distribution we seek has the form $p(U, q, x \mid \theta) = P(U)p(q, x \mid U, \theta)$, where

$$p(q, x \mid U, \theta) = \frac{\mathsf{a}(q, U)}{Z(U \mid \theta)} \cdot \exp\left(\sum_j \theta_j f_j(x, q)\right). \quad (1)$$

$Z(U \mid \theta)$ is the normalization constant for utterance U. Each index j corresponds to one of the statistics mentioned above. If j corresponds to a transition class T, then f_j counts the number of transitions that occur from T,

$$f_j(x, q) = \sum_{t=0}^{\tau}((q_t, q_{t+1}) \in T),$$

and we define $\alpha[T] = \theta_j$. If j corresponds to an occupancy class S, then f_j counts the number of occurrences of substates from S,

$$f_j(x, q) = \sum_{t=1}^{\tau}(q_t \in S),$$

and we define $\beta[S] = \theta_j$. If j corresponds to the mean for the k-th derivative of feature i in mean-variance class G, ($k = 0$ means the base feature itself), then

$$f_j(x, q) = \sum_{t=1}^{\tau}(q_t \in G)x_{i,k}[t],$$

and we define $\nu[i, k, G] = \theta_j$. If j corresponds to the mean squared deviation for i, k, and G, then

$$f_j(x, q) = \sum_{t=1}^{\tau}(q_t \in G)\left(x_{i,k}[t] - \mu[i, k, G]\right)^2,$$

where $\mu[i, k, G]$ is a constant chosen for convenience, and we define $\lambda[i, k, G] = \theta_j$.

Note that the above distribution is defined over only the collection of *base* feature values x. When we write $x_{i,k}[t]$ in the definition of some of the functions f_j above, this is just a shorthand for a linear combination of values $x_i[u]$, $t - w \le u \le t + w$.

Suppose that for each mean-variance class G there is a corresponding set of occupancy classes S that partition G. (This is the equivalent of tying mixture-component weights at an equal or finer granularity than Gaussian parameters, for a CDHMM.) Then we can change $\mu[i, k, G]$ arbitrarily without changing the resulting probability distribution by making compensating changes to $\nu[i, k, G]$ and those $\beta[S]$ for which $S \subseteq G$. This follows from

$$\lambda(x - \mu_0)^2 + \nu_0 x = \lambda(x - \mu_1)^2 + \nu_1 x + c,$$

where $\nu_1 = \nu_0 + 2\lambda(\mu_1 - \mu_0)$ and $c = \lambda(\mu_0^2 - \mu_1^2)$. For example, we may set all $\mu[i, k, G]$ to zero, or, by appropriate choice of $\mu[i, k, G]$, we can force $\nu[i, k, G]$ to zero.

We have shown [4] that the mean-variance constraints for estimated derivatives introduce temporal dependencies by limiting the variability of the (estimated) time derivatives. They also produce nonstationary mean feature trajectories within a substate. This we accomplish with little (a fraction of a percent) or no increase in the number of parameters relative to the corresponding CDHMM.

3. CORRECTING CDHMMS

Another approach is to take the joint pmf/pdf for a CDHMM and substitute in the definitions of the derived features. This gives one the scoring function $\xi(U, q, x)$ that HMM-based decoders use, as a function of the *base* feature values only. This scoring function doesn't sum/integrate to 1 when we integrate over only the base features, but we can normalize it to produce a valid pmf/pdf $\tilde{p}(U, q, x) = \xi(U, q, x)/\tilde{Z}$. The normalizer $\tilde{Z}$ has no effect on recognition, so $\tilde{p}(U, q, x)$ defines the effective distribution used in decoding.

It is straightforward to show that

$$\tilde{p}(U, q, x) = \mathsf{LM}(U)\frac{Z(U \mid \theta)}{\tilde{Z}}p(q, x \mid U, \theta),$$

where $p(q, x \mid U, \theta)$ is defined by (1) and we define:

- $\alpha[T] = \log a$, where a is the state transition probability corresponding to substate transitions in T.
- $\mu[i, k, G]$ is the indicated tied Gaussian mean.
- $\nu[i, k, G] = 0$.
- $\lambda[i, k, G] = -\frac{1}{2}\sigma^{-2}$, where σ^2 is the indicated tied Gaussian variance.
- $\beta[S] = \log w - \frac{1}{2}\sum_{i,k}\log(2\pi\sigma_{i,k}^2)$, where w is the tied mixture component weight and the $\sigma_{i,k}^2$ values the tied Gaussian variances for S.

In other words, conventional HMM-based systems using estimated derivatives as derived features are effectively using a joint distribution that is an instance of our maxent model.

This has two important consequences:

- The effective language model is $\mathsf{LM}(U)Z(U \mid \theta)/\tilde{Z}$. That is, a defect in conventional HMM-based *acoustic* models—failure to account for the origin of derived features—results in a distortion of the effective *language* model.

- Standard HMM training algorithms are incorrect for obtaining locally maximum-likelihood parameter estimates, as they are based on an incorrect likelihood formula that omits a factor of $Z(U \mid \theta)^{-1}$ for each training utterance U.

4. EXPERIMENTS

To gain further understanding of the model and its relation to HMMs, we ran some experiments using the ISOLET corpus [6]. Each utterance in this corpus is a single letter of the English alphabet. The corpus contains 7800 utterances: two for each letter of the alphabet, for each of 150 speakers.

4.1. Parameter Training

Maximum-likelihood training for our model requires computing the normalization constant $Z(U \mid \theta)$ for each training utterance U, as well as the expectations $E[f_j(x,q) \mid U,\theta]$ (for computing the gradient). These tasks become straightforward when the number of possible substate sequences is small enough that one can feasibly enumerate them all. For these initial experiments, we therefore restricted our attention to a simplified case in which we impose the following restrictions:

- There are only a finite, not too large, number of possible substate sequences q for each utterance. In other words, we have a topology of parallel state chains where every state except the entry state has only one possible next state, and only one substate per state.

- Our training instances include both the sequence of feature vectors x and the substate sequence q. (This considerably simplifies and speeds up the optimization procedure.)

(We are investigating a training algorithm that will remove these restrictions by combining dynamic programming with heuristics for pruning and merging sub-solutions.)

To satisfy these requirements, we first used the ISIP speech-recognition system [7], prototype version 5.8, to extract acoustic features and train a simple 15-state, single-Gaussian, left-to-right HMM for each letter. The acoustic features were 12 MFCC coefficients and energy, plus estimated first and second derivatives using a window delta of two. We used the ISIP decoder with the models thus trained to do forced alignments of each of the utterances in the corpus, obtaining 300 allowed state sequences per letter. We made no further use of these HMMs.

We then computed eight sets of maximum-likelihood parameter estimates using the restricted topology: four different definitions of the estimated derivatives, and maxent vs. naïve HMM training. This involved 90 ($15 \cdot 3 \cdot 2$) ν or λ parameters per letter. We set the parameters corresponding to transitions out of the entry state to values forcing equal probability for each allowed substate sequence. We use δ to indicate which definition we chose for the derived features: $\delta = 1$ or 2 indicates estimating first derivatives with a least-squares fit of the $2\delta + 1$ points centered at the current time, and obtaining second derivatives by two applications of the same. $\delta = L$ indicates use of $x[t] - x[t-1]$ for the first and $x[t+1] - 2x[t] + x[t-1]$ for the second derivative. $\delta = R$ indicates use of $x[t+1] - x[t]$ for the first derivative. Details of the procedure to find the maximum-likelihood parameters are in our technical report [4].

Table 1. Effective language model ($\log_{10}$ prob)

A	0.0	R	−49.8	V	−80.0	F	−112.3
E	−7.8	I	−55.6	P	−82.0	X	−147.4
O	−18.6	U	−56.6	K	−88.8	S	−149.9
Z	−20.6	L	−58.9	Q	−91.1	C	−169.7
G	−26.7	T	−72.1	M	−94.9	W	−236.2
D	−27.8	N	−72.5	Y	−102.9		
B	−34.3	H	−77.0	J	−104.2		

4.2. Results

We first investigated the distortion of the effective language model from ignoring the origin of derived features when using HMMs. We took the HMM parameters with $\delta = 2$ and a uniform nominal language model $\mathsf{LM}(U)$, and computed the effective language model $\mathsf{LM}(U)Z(U \mid \theta)/\tilde{Z}$. We computed $Z(U \mid \theta)$ by summing over the 300 allowed state sequences and analytically integrating over the 13τ base feature values for each state sequence.

Table 1 shows the results. We see that the distortion of the language model due to ignoring the origin of derived features can be quite severe. We find the amount of language model distortion surprisingly high—if typical, it leads one to wonder how language models for HMM-based systems manage to be effective at all. Part of the answer is that the language model scaling commonly used in HMM-based systems partially suppresses this language model distortion, as it scales the $\log \mathsf{LM}(U)$ term but not the distorting $\log Z(U \mid \theta)$ term.

Table 2. Likelihoods for varying δ, per time step

δ	relative likelihood
2	1.0
1	28.2
L	356.2
R	352.0

Table 3. Recognition performance

model	δ	errs	UER %	ALPP
HMM	2	20	0.256	-0.114
ME	2	10	0.128	-0.021
HMM	1	32	0.410	-0.157
ME	1	9	0.115	-0.011
HMM	L	53	0.679	-0.275
ME	L	29	0.372	-0.042
HMM	R	63	0.808	-0.286
ME	R	29	0.372	-0.048

We next compared the likelihoods of the parameters obtained for the four maxent models, to test their fit to the data. Table 2 gives the relative likelihoods, per time step.

Finally, we compared actual recognition performance for the HMM versus maxent parameters, with varying δ values. We used two measures of performance: the utterance error rate (UER) and the average log posterior probability (ALPP) of the correct letter. We ran into a problem here arising from our simplistic handling of substate sequences: we could not properly train parameters to handle substate sequences that do not appear in the training set. In fact, we found that parameters trained for only the substate sequences appearing in a training set could be invalid for other sequences appearing in a test set (the pdf is non-normalizable). This problem will disappear when we apply more sophisticated training algorithms that either don't require the substate sequences for training, or can do non-linearly constrained optimization. To get some notion of relative recognition performance in spite of this problem, we used the same set of data for both training and testing. We realize that doing so biases the results toward lower error rates. However, each of the models being compared has the same number of parameters, so the performance of the models *relative to each other* should be unbiased.

Table 3 gives the recognition results. All of the error rates are quite low. The very low error rates appear to be due to the limited number of possible state sequences, rather than the use of the same set for testing as for training. The HMM trained with the ISIP system for forced alignment achieved an error rate of 5.9% on a separate test set. In some early experiments we took those parameters and limited the allowed state sequences either to just those sequences appearing in the training set (as produced by forced alignment), or to all state sequences appearing in either the training or test set. In the latter case we also got very low error rates, whereas in the former case the error rate shot up to around 30%, as none of the available state sequences could provide a good match.

Note that $\delta = 1, 2$ performs better than $\delta = L, R$, in contradiction to the likelihood results. Apparently, the former have more discriminatory power, whereas the latter fit the data better. (Other results [4] indicate that $\delta = L, R$ better models temporal dependencies.) It may be possible to combine the benefits of both without increasing the number of parameters by pooling their respective statistics.

In comparing the HMM performance with the maxent model performance, we see a relative reduction in the error rate ranging from 45% to 72%, depending on how the derived features are computed. Of course, this is a very simple problem with a restricted number of possible state sequences and a very low initial error rate; nevertheless, these results suggest that the maximum entropy approach has promise, and may yield substantial improvements in recognition accuracy.

5. OPEN SCIENCE

Upon request, we will furnish all software and full instructions for reproducing the experimental results of this paper.

6. REFERENCES

[1] S. Furui, "Speaker-independent isolated word recognition using dynamic features of speech spectrum," *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. ASSP-34, no. 1, pp. 52–59, Feb. 1986.

[2] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann, San Mateo, CA, USA, 1988.

[3] P. Smyth, "Belief networks, hidden Markov models, and Markov random fields: A unifying view," *Pattern Recognition Letters*, 1998.

[4] K. S. Van Horn, "A maximum-entropy acoustic model for speech recognition," Tech. Rep., Dept. of Computer Science, North Dakota State University, Fargo, ND, Oct. 2001.

[5] E. T. Jaynes, "Information theory and statistical mechanics I," *Physical Review*, vol. 106, pp. 620–630, 1957.

[6] R. Cole and Y. Muthusamy, "The ISOLET spoken letter database," Tech. Rep. CSE 90-004, Dept. of Computer Science and Engineering, Oregon Graduate Institute, Beaverton, OR, Nov. 1994.

[7] N. Deshmukh, A. Ganapathiraju, and J. Picone, "Hierarchical search for large vocabulary conversational speech recognition," *IEEE Signal Processing Magazine*, vol. 16, no. 5, pp. 84–107, Sept. 1999, Software available at `http://www.isip.msstate.edu/-projects/speech`.