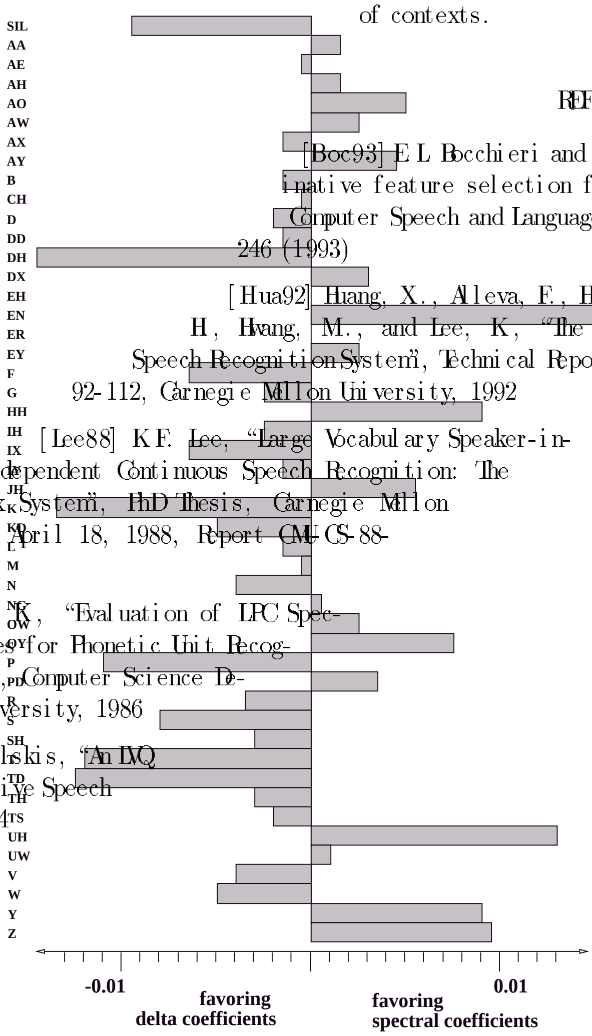
phoneme tend more to favoring delta-coefficients, although one might expect that these phonemes' acoustics are rather static and less context-dependent. The explanation for this observation is the fact that delta coefficients do model the dynamics of a signal but not necessarily the context. A stable context-independent signal like silence has very stable delta coefficients.

delta-coefficients of features, like e.g. delta-spectral-coefficients, delta-delta-spectral-coefficients, power, delta-power and delta-delta-power. Certainly, one may expect different results for other pairs of features and delta-feature. Experiments with non-generalized triphones including cross-word triphones will give also more information about the dependence of the streamweights on the different types of contexts.



4. FUTURE WORK

So far we have only performed experiments with
...o streams. We believe that the proposed ap-
... will be even more fruitful for systems with

REFERENCES

[Boc93] E. L. Bocchieri and J. G. Wilpon, "Discriminative feature selection for speech recognition", Computer Speech and Language, Vol. 7, pp. 229-246 (1993)

[Hua92] Huang, X., Alleva, F., Hayamizu, S., Hon H., Hwang, M., and Lee, K., "The SPHINX-II Speech Recognition System", Technical Report CS-92-112, Carnegie Mellon University, 1992

[Lee88] K.F. Lee, "Large Vocabulary Speaker-independent Continuous Speech Recognition: The ...inx System", PhD Thesis, Carnegie Mellon ...ty, April 18, 1988, Report CMU-CS-88-

...no, K., "Evaluation of LPC Spec-... ...sures for Phonetic Unit Recog-... ...ort, Computer Science De-... University, 1986

...ebelskis, "An LVQ ... ...aptive Speech ... -444

...ch

$\alpha_i(B)$ after iteration $k+n$ to be approximately $\alpha_i(B)^t - n \cdot \lambda(d^* LP_t(\alpha, B)/d^*\alpha_i(B))$ (if no sigmoid is applied). We have found that the differences from iteration to iteration are in fact so small that this approximation is valid, which suggested a second solution to the above mentioned problem, namely to run simply one or two iterations with a large stepsize, or alternatively to use a cross validation mechanism to decide what number of iterations (i.e. what stepsize λ) is best.

3. EXPERIMENTS

ave performed experiments on the English
e Registration Task (CR) [Wo92] and
nagement Task (RM), using the
92] of the JANUS Speech to
[Wai91]. The recog-
l probabilities for
ned 50-cluster
probability
300 context
1000
m

path, C, did not get the highest probability of all states, then there is some other state B that has the highest probability, and we will adjust the parameters to increase the probability of C and to decrease the probability of B. (If the optimal path already has the highest probability, no training will happen at all.) Now, let $c_i(t)$ be the contribution of the stream i to the score for the correct state C at time t, and let $LP_t(\alpha, C) := -\log P(x_t|C)$, so $LP_t(\alpha, C) = \sum_{i=1}^{n} c_i(t) \cdot \alpha_i(C)$. Let $b_i(t)$ be the contribution of the stream i to the score for the best state B at time t (the state with the highest probability). This means that $\sum_{i=1}^{n} c_i(t) \cdot \alpha_i(C) \leq \sum_{i=1}^{n} b_i(t) \cdot \alpha_i(B)$. The goal of the training procedure is to modify $\alpha_i(B)$ and $\alpha_i(C)$ such that $LP_t(\alpha, C)$ decreases and $LP_t(\alpha, B)$ increases. For that, we need to compute the derivative of $LP_t(\alpha, S)$ with respect to $\alpha_i(S)$. The update rule will then be:

$$\alpha_j(B)^{updated} = \alpha_j(B) + \lambda \cdot \frac{d\ LP_t(\alpha,\ B)}{d\ \alpha_j(B)} \quad (3)$$

$$\alpha_j(C)^{updated} = \alpha_j(C) - \lambda \cdot \frac{d\ LP_t(\alpha,\ C)}{d\ \alpha_j(C)} \quad (4)$$

We can easily see, that in the general case the updated system will produce a higher probability for the correct Viterbi-path (or for some given labels). Note that the partial derivative $\left(\frac{\partial LP_t(\alpha,S)}{\partial \alpha_i(S)}\right)$ will not yield the correct result since the $\alpha_i(S)$ and $\alpha_j(S)$ are not independent from each other because of the above mentioned summation constraint. So any gradient descent step must result in a set of $\alpha_i(S)$ which meet this constraint. [illegible] step from $A(S) = \langle \alpha_1(S), \ldots \alpha_n(S) \rangle$ [illegible] $\langle \alpha'_1(S) = \alpha_1(S) + \delta_1, \ldots \alpha'_n(S) = \ldots \rangle$ could thus be defined as $\delta_j := \epsilon$ and [illegible] for $i \neq j$. This step definition [illegible] $\alpha_j(S)$ the most, while all the other $\alpha_i(S)$ are [illegible] meets the summation constraint and to [illegible] values unchanged. Other step [illegible]

definition of a directional derivative. Now we are numerically computing the derivative of a function, whose domain is a manifold in parameter space, only for values on a hyperplane of the feature space. With this step definition, the d^* will be used [illegible] restricted to the hyperplane of the feature space which meets the summation constraint:

$$\frac{d^* LP_t(\alpha,\ B)}{d^* \alpha_j(B)} := \lim_{\epsilon \to 0} \frac{LP_t(\alpha',\ B) - LP_t(\alpha,\ B)}{\epsilon}$$

$$= \lim_{\epsilon \to 0} \frac{1}{\epsilon} \left(\sum_{i=1}^{n} \alpha'_i(B) b_i - \sum_{i=1}^{n} \alpha_i(B) b_i \right)$$

$$= b_j + \frac{\sum_{i \neq j} \alpha_i(B) b_i}{\alpha_j(B) - 1} \quad (5)$$

Here, we have ignored that the actual size of the infinitesimal step is somewhat greater than ϵ, resulting in a somewhat greater denominator. But since we will use a stepsize λ for performing the gradient descent, λ will subsume this difference.

For a simple two-feature system eq. (5) results in:

$$\frac{d^* LP_t(\alpha,\ B)}{d^* \alpha_1(B)} = b_1 + \frac{\alpha_2(B) b_2}{\alpha_1(B) - 1} = b_1 - b_2 \quad (6)$$

2. IMPLEMENTATION ISSUES

We have to consider the possibility of one feature always being the best for some model. In this case its feature weight would rise with every update. Nothing would stop it from growing greater than 1 and thus pushing the other feature weights below 0. This, obviously, is an unwanted effect. The solution to this problem could be to set some floor and ceiling value to keep the weights in these bounds. [illegible] be increased most, by increasing all $\alpha_i(S)$ [illegible] is to apply a sigmoid to [illegible] according to [illegible] don't want because we would like to dis[illegible] different features. [illegible] one feature F_a is [illegible] feature F_b in [illegible] more important [illegible] should ex[illegible]

LEARNING STATE-DEPENDENT STREAM WEIGHTS FOR MULTI-CODEBOOK HMM SPEECH RECOGNITION SYSTEMS

I. Rogina, A. Waibel
University of Karlsruhe, Postfach 6980, 76128 Karlsruhe, Germany
Carnegie Mellon University, Pittsburgh, PA 15213-3890, USA

ABSTRACT

Many speech recognition systems [Lee88], [Shi85], [Ma92], use multiple information streams to compute HMM output probabilities (e.g. systems based on semicontinuous or discrete HMMs use one codebook for cepstral coefficients, and another one for delta cepstral coefficients). The final score a weighted sum of the contributions of every These weights can be found empirically ly the same set of weights is used for ev- c model. There is reason to believe that eatures which are more important for odels than for others. Especially, the beginning and ending seg- more context dependent that case the proba- h recognizer should stream -spectrum than 5], [Boc93], coefficients nd more

1. TRAINING

which uses n information t, and for a given HMM ream i will compute $P_i(x_t|S)$, where x_t . The overall prob- is then $P(x_t|S) =$ s a weighting factor for lled codebook expo- and $\sum \alpha_i = 1$ obability. We HMM state eam i for stream ility

(2)