ADVANCES IN CONFIDENCE MEASURES FOR LARGE VOCABULARY

A.Wendemuth, G. Rose and J.G.A. Dolfing

Philips Research Laboratories
Weißhausstraße 2
D-52066 Aachen, Germany
Email: {wendemu,rose,dolfing}@pfa.research.philips.com

ABSTRACT

This paper adresses the correct choice and combination of confidence measures in large vocabulary speech recognition tasks. We classify single words within continuous as well as large vocabulary utterances into two categories: utterances within the vocabulary which are recognized correctly, and other utterances, namely misrecognized utterances or (less frequent) out-of-vocabulary (OOV).

To this end, we investigate the classification error rate (CER) of several classes of confidence measures and transformations. In particular, we employed data-independent and data-dependent measures. The transformations we investigated include mapping to single confidence measures and linear combinations of these measures. These combinations are computed by means of neural networks trained with Bayes-optimal, and with Gardner-Derrida-optimal criteria.

Compared to a recognition system without confidence measures, the selection of (various combinations of) confidence measures, the selection of suitable neural network architectures and training methods, continuously improves the CER.

1 Introduction

We address the problem of correct choice and combination of confidence measures in speaker-dependent large vocabulary recognition based on hidden Markov models. Our motivation is manifold: e.g. when the speech input is recognized reliably with the help of suitable confidence measures, the need to verify a speaker's input in a dialog structure diminishes. Therefore, the aim of this work is to judge the word recognition result and to determine whether we have to 'accept' or 'reject' this result. This decision is based on speaker-independent and speaker-dependent confidence measures. We do not apply elaborate garbage models but investigate the performance of several classes of confidence measures and transformations. We investigate novel combinations of data-dependent confidence measures leading to a very effective and efficient classifier.

A number of confidence measure realizations related to the acoustic model, the search process and the language model exist in the literature. Examples of confidence measures applied to the acoustic model are [3, 11], to the decoding process [5], and to language model and word graphs [8, 10, 14]. It is possible to combine several confidence measures of the same and/or neighboring word hypotheses to solve the decision problem as demonstrated by [4, 8, 11, 12]. However, complex combination strategies do not significantly outperform simpler linear feature combinations [8].

In Section 2 and 3, we introduce the procedure to arrive at the best classification given the model parameters. Section 4 and 5 introduce the experimental setup and results, respectively. Finally, we draw conclusions in Section 6.

2 Best classification with given model parameters

We address the following question: after selecting the set of raw input parameters X (see following sections), can we define a classifier for utterance verification $f(X)$ and a threshold τ such that the condition $f(X) \leq \tau$ will classify into class $c = 0$ (rejection), and otherwise $c = 1$ (acceptance)?

We shall treat this problem in the framework of probability density functions $P(.)$ and conditional probability density functions $P(.|.)$, where it is understood that these functions are not known to us but that our aim is to reproduce them, using the samples at our disposal. It is clear that the decision boundary $f(X) = \tau$ will ideally, after Bayes' decision rule, have to be equal to the Bayes posterior decision boundary $P(c = 1|X) = P(c = 0|X) = 0.5$, with the Bayes posterior probability $P(C|X)$ of class C given the observation X. We take into account the possible presence of outliers and misclassifications in our training set and will therefore experiment in Section 3 with a careful adjustment of the decision *boundary* $f(X) = \tau$.

In order to deal directly with the functional forms of $f(.)$, we adopt a vector notation. A particular sample from the set of raw input parameters X will be the vector $\underline{\mathbf{X}}_{\text{raw}}$. For a linear functional form of $f(.)$, we can first of all include the threshold in $f(.)$ simply by augmenting $\underline{\mathbf{X}}_{\text{raw}}$ with a constant 1 to give $\underline{\mathbf{X}} = (\underline{\mathbf{X}}_{\text{raw}}, 1)$. The decision boundary $f(X) = \tau$ is then equivalent to $a \overset{def}{=} \underline{\mathbf{J}} \cdot \underline{\mathbf{X}} \overset{!}{=} 0$, where we have to find the components of $\underline{\mathbf{J}}$. Note that in this formulation we do not attempt to model $P(C|X)$, but just the Bayes posterior decision boundary following from $P(C|X)$.

The following shows under which conditions the Bayes posterior distribution can be modelled as a function of a. Some of the outline follows [2]. Starting with Bayes theorem, it can be seen as follows that the Bayes posterior can be written in the sigmoid form

$$y = P(c=1|X) = g(a') \overset{def}{=} \frac{1}{1+e^{-a'}} \tag{1}$$

with

$$a' = \ln \frac{p(X|c=1)P(c=1)}{p(X|c=0)P(c=0)} \tag{2}$$

We now assume that the class-conditional densities $p(X|C)$ are members of the *exponential family* of distributions with common non-linear dependence of the exponents on $\underline{\mathbf{X}}_{\text{raw}}$ and individual linear dependence on $\underline{\mathbf{X}}_{\text{raw}}$. Bernoulli and Gaussian distributions are special cases of members of this family. Inserting the functional form of these class-conditional densities into (2), we indeed obtain $a' = \underline{\mathbf{J}} \cdot \underline{\mathbf{X}}_{\text{raw}} - \tau = a$.

We have therefore established that the use of a sigmoid form (1), with $a = \underline{\mathbf{J}} \cdot \underline{\mathbf{X}}$, always applies given the stated functional form of the class-conditional densities. Since the latter is only a very mild restriction, our *ansatz* is correct under rather general conditions. However, we shall later see that fine-tuning of the result can lead to better generalization which can be interpreted as an artefact of these assumptions only applying approximately in our test cases.

Since we are only interested in classification, we may apply directly from (1) the decision boundary $a = 0$, i.e., we never actually need to compute the posterior probability. Note however that this computation can become useful if training and test scenario have different *known* distributions $P(C)$ and $P(x|C)$ which can then be taken care of very simply by multiplications.

Having established the functional form of a Bayes posterior distribution, we now look at a suitable error function that will be minimized. Following standard arguments [2], for binary classifications we minimize over all samples i the *Cross Entropy* [7]

$$E = -\sum_i \{c_i \log(y_i) + (1-c_i)\log(1-y_i)\}. \tag{3}$$

We find a $\underline{\mathbf{J}}$ that minimizes (3) if we apply a stochastic sequence of additive modifications $\delta\underline{\mathbf{J}}$. To this end, we choose a constant η and, at each step, we choose randomly an input i and update $\underline{\mathbf{J}}$ along the negative gradient of E with respect to $\underline{\mathbf{J}}$,

$$(\delta\underline{\mathbf{J}})(i) = -\eta \frac{\partial E}{\partial a_i} \nabla \underline{\mathrm{J}}(a_i) = \eta \underline{\mathbf{X}}_i \left(c_i - \frac{1}{1+e^{-a_i}}\right) \tag{4}$$

This defines our learning rule for a Neural Network with one layer and sigmoid output function (1). Note that the term in parentheses lies in the range $(-1, 1)$. In the case of complete misclassification it approaches the values ± 1 which makes (4) exactly equivalent to conventional Perceptron learning [9]. Equating (4) to 0 is a fixed-point equation for $\underline{\mathbf{J}}$ which however cannot be solved analytically, justifying the Neural Network approach.

3 Fine-tuning the result

Having trained the network in this Bayes-optimal sense (with η decreasing over time) still leaves us with the problems of outliers or misclassified data in our training samples. Our assumptions for validity of the functional form (1) may also lead to non-optimality of the result obtained so far.

How can these problems be tackled? Although the cross entropy error has the pleasing property of estimating small probabilities much better than a LMS error function, which is favorable in the case of outliers, other choices of error functions such as regularized or marginalized ones [2] can be considered. This is outside the scope of this paper.

Instead, we fine-tuned our result for $\underline{\mathbf{J}}$ at the decision *boundary*. To this end, an algorithm developed by one of the authors [13] was used to include further data into the set of correctly classified patterns. The *Gardner–Derrida* error function in [13], measuring the number of correctly classified data, is maximized. By doing so, outliers or originally misclassified data are ignored for the calculation of $\underline{\mathbf{J}}$. This results in a shift of the decision boundary, together with a higher number of correctly classified data, and improved classification ability in the test sets (Section 5.2).

4 Experimental setup

In the following, we will present and compare results obtained both for small vocabulary and large vocabulary tasks. The experimental setup for both scenarios is described. Data for the small vocabulary task are taken from [6] and are used here for comparison.

The employed database for small vocabulary command-and-control contains single word utterances by 50 individuals (25 male, 25 female) who each spoke four to six utterances of 10 given words plus a number of additional out-of-vocabulary (OOV) utterances. The development data model 500 words with hidden Markov models each trained with only two additional utterances. The number of states of a word model equals about 0.8 times the number of observed frames and each state contains only one density. The acoustic preprocessing employs a frame-shift of 20ms and computes 20 cepstral features, including derivatives, for every feature vector. The evaluation data contains a total of 3345 utterances, 2861 utterances to test the word models and 484 OOV utterances evenly distributed over all speakers.

For the large vocabulary task we employed the male part of the evaluation set NAB'94 which contains 10 male speakers who each spoke 15-17 whole sentences composed out of a 64k vocabulary. The training of the triphone models was carried out gender dependently on the WSJ0+1 corpus. The Philips system for large vocabulary continuous speech recognition used here is described in [1].

The classification error rate (CER), which is the number of correctly tagged words divided by the total number of words, is used to compare results.

In our experiments, we employ five basic confidence measures for the small vocabulary task, and eight (three additional) for the large vocabulary task. They are described in the following. Each confidence measure is computed at the end of a word hypothesis with loglikelihood l_w at time t_{end} for a word starting at t_{start}. The 'two-best' measure contains the log–likelihood difference between the best and second best hypothesis at time t while the 'n-avg-best' measure contains the difference between the best and the average loglikelihood of the N-best hypotheses. The measure 'n-best-states' is computed as the difference of the loglikelihood of the word hypothesis and the sum of the best state hypotheses over the interval $[t_{\text{start}}, t_{\text{end}}]$. The 'avg-acoustics' divides $l_w/(t_{\text{end}} - t_{\text{start}} + 1)$. The 'speaking-rate' divides the number of speech frames of the word hypothesis by the number of states in the word model.

The measure 'word-end-frequency' is the frequency of occurences of the desired word or its homophones in the list of all word ends. The 'active-state-count' is the number of remaining active states at the word end time after pruning. The 'lm-score' is simply the bigram language model score for the word and its predecessor. The measure 'word-graph' was computed off-line using the full wordgraph. It is the "posterior word hypothesis probability" which is calculated for each word hypothesis ("edge" in the word graph) within a time interval given a sequence of acoustic feature vectors (for more details see [14]).

For each utterance of the development and evaluation data, we compute a vector with confidence measures. Because the confidence measures obtained from the development data partially exhibit a behavior completely different from the measures computed on the evaluation data, we split the set of vectors of confidence measures randomly in two parts. For the small vocabulary task, we split a set of 3345 vectors into one part containing 1672 vectors (used to train the confidence classifier) and a second part of 1673 vectors used for testing. For the large vocabulary task, we split a set of 3637 vectors into one part containing 1818 vectors (used to train the confidence classifier) and a second part of 1819 vectors used for testing.

All results given in this paper are on the test part of the two databases.

Besides a speaker-independent setup, we can use a speaker-dependent setup. Instead of the decision problem $f(X) \leq \tau$ with a fixed threshold τ for all speakers i, we employ one threshold for all data but first subtract a speaker-specific offset $\mathcal{O}_i$. The decision problem is then $(f(X) - \mathcal{O}_i) \leq \tau$. This approach is investigated in Section 5.3.

Proper classification of the vector of confidence measures probably cannot be done linearly. Therefore, we took the five best single confidence measures $\underline{\mathbf{X}}_5$ and appended to $\underline{\mathbf{X}}_5$ the 15 2nd-order components $(x_1^2, x_1x_2, x_1x_3, \ldots, x_5^2)$. This leads to a 20 dimensional vector $\underline{\mathbf{X}}_{20}$ which can be treated with standard scalar multiplications.

5 Experiments

In the initial, speaker-dependent recognition system without any confidence measures, the classification error rate equals the word error rate of 16.7% in the small vocabulary task and 14.9% in the large vocabulary task. We compute an optimal threshold on the training set and apply that threshold to the test set.

5.1 Speaker-independent confidence measures

We investigate the tagging accuracy of the eight individual confidence measures in a speaker-independent setting.

Table 1: The classification error rate [%] for individual confidence measures.

Confidence measure	Error rate	
	Small Voc.	**Large Voc.**
two-best	10.2	.
n-avg-best	9.8	12.2
n-best-state	12.2	12.5
avg-acoustic	12.4	12.4
speaking-rate	15.1	12.9
word-graph	.	11.4
word-end-frequency	.	24.8
lm-score	.	12.4
active-state-count	.	12.6

One striking feature in the large vocabulary task is the fact that the best single confidence measure "word-graph" is already very efficient with 11.4% CER and exceeds the other single confidence measures by at least 0.8% absolute in CER. This can be explained since the "word-graph' is the only measure which uses the history, including the language model information, in a manner more sophisticated than the "lm-score".

5.2 Confidence measure combination

In a follow-up experiment, we try to combine the confidence measures such that the resulting classification error rate is lower than that of the individual confidence measures. The improvement is measured against the best single CER, this is 9.8% for small vocabulary and 11.4% for large vocabulary. We classify both $\underline{\mathbf{X}}_5$, $\underline{\mathbf{X}}_8$ and $\underline{\mathbf{X}}_{20}$ with the one-layer perceptron J as explained in Section 2.

Table 2: The classification error rate [%] for combined confidence measures.

Combination	Error rate			
	Small Voc.		**Large Voc.**	
Bayes (d=5), $\underline{\mathbf{J}}$	8.4	(-14.3%)	11.0	(-3.5%)
Bayes (d=8), $\underline{\mathbf{J}}$	.		10.9	(-4.4%)
Bayes (d=20), $\underline{\mathbf{J}}$	8.5	(-13.3%)	11.3	(-1.8%)

It can be immediately seen that the improvement in CER is much better for the small vocabulary task. This can be traced to two effects. The task for large vocabulary is certainly more difficult. The other effect is that the best single confidence measure "word-graph" is already very efficient, as explained above in Sec. 5.1.
A further improvement is therefore more difficult.

5.3 Data-dependent confidence measures and combination

First, we investigate the effect of personalizing the 'avg-acoustic' and 'speaking-rate' measures. For the 'avg-acoustic' measure, we subtracted a speaker-specific offset $\mathcal{O}_i^{aa}$, as explained in Section 4. $\mathcal{O}_i^{aa}$ contains the average value of 'avg-acoustic' on all training utterances of speaker i. In the case of the speaking rate, we determine the offset $\mathcal{O}_i^{sp}$ similar to the 'avg-acoustic' measure. We compared minimum, maximum and mean functions to obtain speaker-dependent offsets and found the best performance for taking the mean ‘avg-acoustics’ and the maximum ‘speaking rate’. The results are presented in Table 3 while combinations of confidence measures are presented in Table 4.

Table 3: The classification error rate [%] for single, individual confidence measures which are speaker dependent

Confidence measure	**Error rate**	
	Small Voc.	**Large Voc.**
avg-acoustic	12.4 → 11.1 (-10.5%)	12.4 → 12.4 (0.0%)
speaking-rate	15.1 → 15.1 (0.0%)	12.9 → 12.4 (- 3.9%)

Second, we replace two confidence measures in the speaker-independent measure vector $\underline{\mathbf{X}}_8 = (x_1, \ldots, x_8)$ to obtain a new feature vector $\underline{\mathbf{X}}_8^{'}$ where the $x_4^{'} = x_4 - \mathcal{O}_i^{aa}$ and $x_5^{'} = x_5 - \mathcal{O}_i^{sp}$ as explained above. Acting on these vectors which contain our raw confidence measures, we now use the neural network trained with Bayes only, and trained with Bayes and Gardner-Derrida (GD) error functions, to find the best technique for confidence measure combination. The classification results with the word-specific feature vector $\underline{\mathbf{X}}_8^{'}$ and $\underline{\mathbf{X}}_{20}^{'}$ are given in Table 4. Improvements are again given against the best single confidence measure.

Table 4: The classification error rate [%] for combined confidence measures including spaeker-specific confidence measures.

Combination	**Error rate**	
	Small Voc.	Large Voc.
Bayes	7.0 (-28.6%)	10.9 (-4.4%)
Bayes + GD	6.7 (-31.6%)	10.7 (-6.1%)

As stated in Section 3, the fine-tuning shifts the decision boundary. We indeed see that, for both small and large vocabulary, this fine-tuning leads to improved classification error rate .

6 Conclusion

Overall, the single confidence measures as well as the combined measures improve the classification error rate. Compared to a recognition system without confidence measures, we have improved the classification error rate from 16.7% to 6.7% (-60% relative) for small vocabulary tasks, and from 14.9% to 10.7% (-28% relative) for large vocabulary tasks. Compared to the single best confidence measure, the improvement is -31.6% for small vocabulary tasks, and -6.1% for large vocabulary tasks. The application of a Bayes one-layer perceptron, Bayes plus data-dependent measures, and Bayes plus *Gardner–Derrida* plus data-dependent confidence measures continuously improves the classification error rate. Comparing small and large vocabulary tasks, we find that the improvements for small vocabulary tasks are larger. This can be explained by two effects: a) The task for large vocabulary is certainly more difficult. b) The best single large vocabulary confidence measure “word-graph” is already very efficient since it uses the word history, a further improvement is therefore more difficult.

7 REFERENCES

1. P. Beyerlein, M. Ullrich, and P. Wilcox. Modelling and decoding of crossword context dependent phones in the Philips large vocabulary continuous speech recognition system. In *Proc. EUROSPEECH*, volume 3, pages 1183–1187, August 1997.
2. C. Bishop. *Neural Networks for pattern recog.* Oxford, 1995.
3. H. Bourlard, B. D’hoore, and J.M. Boite. Optimizing recognition and rejection performance in wordspotting systems. In *Proc. ICASSP*, volume 1, pages 373–376, May 1994.
4. J. Caminero, C. de la Torre, L. Villarrubia, C. Martín, and L. Hernandez. On-line garbage modelling with discriminant analysis for utterance verification. In *Proc. ICSLP*, volume 4, pages 2111–2114, Philadelphia, PA, October 1996.
5. Stephen Cox and Richard C. Rose. Confidence measures for the switchboard database. In *Proc. ICASSP*, volume I, pages 511–514, Atlanta, GA, May 1996.
6. J.G.A. Dolfing and A. Wendemuth. Combinations of confidence measures in isolated word recognition. In *Proc. ICSLP*, volume 1, December 1998.
7. J. Hopfield. Learning algorithms and probability distributions in feed-forward and feed-back neural networks. *Proc. Nat. Ac. Sciences*, 84:8429, 1987.
8. Thomas Kemp and Thomas Schaaf. Estimating confidence using word lattices. In *Proc. EUROSPEECH*, volume 2, pages 827–830, Rhodes, Greece, September 1997. ESCA.
9. F. Rosenblatt. *Principles of Neurodynamics – Perceptrons and the theory of brain*. Spartan, Washington D.C., 1961.
10. Bernhard Rueber. Obtaining confidence measures from sentence probabilities. In *Proc. EUROSPEECH*, volume 2, pages 739–742, Rhodes, Greece, September 1997.
11. Thomas Schaaf and Thomas Kemp. Confidence measures for spontaneous speech recognition. In *Proc. ICASSP*, volume II, pages 875–878, Munich, Germany, April 1997.
12. M. Weintraub, F. Beaufays, Z. Rivlin, Y. Konig, and A. Stolcke. Neural-network based measures of confidence for word recognition. In *Proc. ICASSP*, volume II, pages 887–890, Munich, Germany, April 1997.
13. A. Wendemuth. Learning the unlearnable. *J. Phys. A*, 28:5423, 1995.
14. F. Wessel, K. Macherey, and R. Schlueter. Using word probabilities as confidence measures. In *Proc. ICASSP*, volume 1, pages 225–228, May 1998.