COMBINATION OF CONFIDENCE MEASURES IN ISOLATED WORD RECOGNITION

J.G.A. Dolfing and A.Wendemuth

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

ABSTRACT

In the context of command-and-control applications, we exploit confidence measures in order to classify single-word utterances into two categories: utterances within the vocabulary which are recognized correctly, and other utterances, namely out-of-vocabulary (OOV) or misrecognized utterances.

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, LDA-transformed measures, and other 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. Additionally, we found that a linear perceptron generalizes better than a non-linear backpropagation network.

1 Introduction

In this paper, we address the problem of confidence estimation for isolated, speaker-dependent word recognition based on hidden Markov models. With an increasing number of users of command-and-control applications with speech input, the need for reliable speech recognition also increases. When the speech input is recognized reliably, 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, speaker-dependent, and word-specific 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.

In the literature, we find a number of confidence measure realizations related to the acoustic model, the search process and the language model. Examples of confidence measures applied to the acoustic model are [2, 10], to the decoding process [4], and to language model and word graphs [6, 9, 13]. It is possible to combine several confidence measures of the same and/or neighboring word hypotheses to solve the decision problem as demonstrated by [3, 6, 10, 11]. However, complex combination strategies do not significantly outperform simpler linear feature combinations [6].

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* $f(X) = \tau$.

A general nonlinear function can be realized by a Multilayer Neural Net architecture which in principle is known [7] to be able to model arbitrary functional forms. However, it is also known [1] that more detailed functional approximation may lead to a loss in generalization ability. In order to check these competing effects for the specific problems discussed here, we performed experiments with multilayer networks in Section 5.4. We indeed found that the training data could be matched excellently, albeit only with a loss of generalization ability. Therefore, we have indications that linear functions f indeed outperform nonlinear ones.

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 $a \stackrel{def}{=\!=} \underline{\mathbf{J}} \cdot \underline{\mathbf{X}} \stackrel{!}{=\!=} 0$, 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 discussion is stimulated by [1]. Let us first of all show under which conditions the Bayes posterior ditribution can be modelled as a function of a. Using Bayes theorem, it can be seen as follows that the Bayes posterior can be written in the sigmoid form

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

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 dependance of the exponents on $\underline{\mathbf{X}}_{\text{raw}}$ and individual linear dependance 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* priors $P(C)$ 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 [1], for binary classifications we minimize over all samples i the *Cross Entropy* [5]

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

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 $\underline{\mathbf{J}}$,

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

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 [8]. Note that equating (4) to 0 is a fixed-point equation for $\underline{\mathbf{J}}$ which however cannot be solved analytically, which justifies 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 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 [1] 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 [12] was used to include further data into the set of correctly classified patterns. The *Gardner–Derrida* error function in [12], measuring the number of correctly classified data, is maximized. By doing so, outliers or orignally 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

The employed database 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. 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.

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 3345 vectors of confidence measures randomly in two parts. One part contains 1672 vectors and is used to train the confidence classifier. The other part of 1673 vectors is used for testing and for all test results given in this paper.

In our experiments, we employ five basic confidence measures.

Each confidence measure is computed at the end of a word hypothesis with loglikelihood l_w at time t_{end} while the word started at t_{start}. The 'two-best' measure contains the loglikelihood 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.

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

Proper classification of the vector of confidence measures $\underline{\mathbf{X}}_5 = (x_1, \ldots, x_5)$ probably cannot be done linearly. Therefore, we optionally append 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%. 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 five individual confidence measures in a speaker-independent setting. For historical reasons, the classification error rate of 10.2% for the 'two-best' confidence measure serves as a baseline classification error rate for the other experiments. This means that the single confidence measure 'n-avg-best' already yields a small improvement of 3.9% (rel.) against the baseline. However, the other confidence measures yield a higher classification error rate .

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

Confidence measure	Error rate
two-best	10.2
n-avg-best	9.8
n-best-state	12.2
avg-acoustic	12.4
speaking-rate	15.1

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 compared to the CER=10.2% of the single 'two-best' measure. To this end, we employ linear discriminant analysis (LDA). The LDA transform matrix is a linear transformation, estimated on the 1672 training vectors, and applied in this experiment to the test data either for full transformation or for projection to the eigenvector with highest eigenvalue (marked "1st ev." in tables). We estimate two LDA matrices with dimension 5x5 and 20x20 for the original vectors of confidence measures $\underline{\mathbf{X}}_5$ and the extended vectors $\underline{\mathbf{X}}_{20}$. Additionally, we classify both $\underline{\mathbf{X}}_5$ 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	
LDA (d=5), 1st ev.	10.4	(+ 2.0%)
LDA (d=20), 1st ev.	9.0	(-11.8%)
Bayes (d=6), $\underline{\mathbf{J}}$	8.4	(-17.6%)
Bayes (d=21), $\underline{\mathbf{J}}$	8.5	(-16.7%)

Although the LDA and the perceptron both employ a vector multiplication to classify the input, the LDA improves the classification error rate by 11.8% (rel.) to 9.0% while the perceptron improves the classification error rate by 17.6% (rel.) to 8.4%.

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 speaker-specific and word-specific offsets $\mathcal{O}_i^{aa}$ and $\mathcal{O}_{i,w_j}^{aa}$, respectively, as explained in Section 4. While $\mathcal{O}_i^{aa}$ contains the average value of 'avg-acoustic' on all training utterances of speaker i, $\mathcal{O}_{i,w_j}^{aa}$ contains the average value of 'avg-acoustic' for only the two training utterances of word j of speaker i. In the case of the speaking rate, we determine the offsets $\mathcal{O}_i^{sp}$ and $\mathcal{O}_{i,w_j}^{sp}$ similar to the 'avg-acoustic' measure. We compared minimum, maximum and mean functions to obtain word-dependent and 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 and word dependent, respectively.

Confidence measure	Error rate		
	Speaker indep.	Speaker dependent	Word dependent
avg-acoustic	12.4	11.1 (-10.5%)	10.0 (-19.4%)
speaking-rate	15.1	15.1 (- 0.0%)	14.2 (- 6.0%)

Second, we replace two confidence measures in the speaker-independent measure vector $\underline{\mathbf{X}}_5 = (x_1, \ldots, x_5)$ to obtain a

new feature vector $\underline{\mathbf{X}}_5^{'}$ where the $x_4^{'} = x_4 - \mathcal{O}_{i,w_j}^{aa}$ and $x_5^{'} = x_5 - \mathcal{O}_{i,w_j}^{sp}$ are word-specific as explained above. The same procedure is applied to $\underline{\mathbf{X}}_{20}$ to obtain $\underline{\mathbf{X}}_{20}^{'}$. Acting on these vectors which contain our raw confidence measures, we now use projection on the LDA's first eigenvector, 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 $\underline{\mathbf{X}}_5^{'}$ and $\underline{\mathbf{X}}_{20}^{'}$ are given in Table 4.

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

Combination	Error rate	
	'linear'd=5	'nonlinear'd=20
LDA (d), 1st ev.	8.2	7.5
Bayes (d+1), $\underline{\mathbf{J}}$	7.0	7.3
(Bayes + GD) (d+1), $\underline{\mathbf{J}}$	6.7	6.6

As stated in Section 3, the fine-tuning shifts the decision boundary. In this case, this shift can be measured as the normalized overlap. Using $\underline{\mathbf{X}}_5^{'}$, we obtain $(\underline{\mathbf{J}}_{Bayes} \cdot \underline{\mathbf{J}}_{Bayes+GD})/(|\underline{\mathbf{J}}_{Bayes}||\underline{\mathbf{J}}_{Bayes+GD}|) = 0.988$. The result with $\underline{\mathbf{X}}_{20}^{'}$ is 0.856. We can compare this to the error rate results given in Table 4: the greater improvement in error rate in the 'nonlinear' case corresponds to a greater shift in the decision boundary. This lies in the nature of the *Gardner–Derrida* error function that we optimized (Section 3): its ability to enlarge the number of correctly classified patterns increases with the number of dimensions of the problem [12].

5.4 Nonlinear combination

So far, we solved the problem $f(X) \leq \tau$ with linear classifiers based on linear discriminant analysis and a one-layer perceptron. Due to the distribution of the confidence measures, a non-linear classifier such as a backpropagation neural network might be able to achieve a better classification. We used a 3-layer network with layout 6-30-1 with LDA preprocessing. This is possible since we use a full rank LDA transformation which only linearly transforms the input data and therefore does not have any effect on subsequent manipulations other than the desired one of speedup. In particular, this can be seen from the 6-dimensional test set error rate which is identically 6.7% for (Bayes + GD) (Table 4) and LDA + (Bayes + GD) (Table 5). Table 5 shows that the achieved classification error rate of the Backpropagation network on the training set is excellent but the classification error rate on the test set is worse than the linear Bayes classifier.

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 6.6% (-60% relative). Compared to the baseline system with the single 'two-best' confidence measure, we have improved the classification error rate from 10.2% to 6.6% (-35% relative). The application of linear discriminant analysis, 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. Additionally, we found that the results of our one-layer Bayesian perceptron generalize better compared to a non-linear backpropagation network.

Table 5: The classification error rate [%] for $\underline{\mathbf{X}}_5^{'}$ with multilayer architecture. Backpropagation = BP.

Combination	Error rate	
	Training set	Test set
LDA + (Bayes + GD)	6.6	6.7
LDA + BP 10,000 steps	5.2	7.5
LDA + BP 100,000 steps	0.9	10.2
LDA + BP 1,000,000 steps	0.8	12.1

7 REFERENCES

1. C. Bishop. *Neural Networks for pattern recog.* Oxford, 1995.
2. 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.
3. 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.
4. Stephen Cox and Richard C. Rose. Confidence measures for the switchboard database. In *Proc. ICASSP*, volume I, pages 511–514, Atlanta, GA, May 1996.
5. J. Hopfield. Learning algorithms and probability distributions in feed-forward and feed-back neural networks. *Proc. Nat. Ac. Sciences*, 84:8429, 1987.
6. Thomas Kemp and Thomas Schaaf. Estimating confidence using word lattices. In *Proc. EUROSPEECH*, volume 2, pages 827–830, Rhodes, Greece, September 1997. ESCA.
7. R. Lippmann. An introduction to computing with neural nets. *IEEE ASSP*, 4:4, 1987.
8. F. Rosenblatt. *Principles of Neurodynamics – Perceptrons and the theory of brain.* Spartan, Washington D.C., 1961.
9. Bernhard Rueber. Obtaining confidence measures from sentence probabilities. In *Proc. EUROSPEECH*, volume 2, pages 739–742, Rhodes, Greece, September 1997.
10. Thomas Schaaf and Thomas Kemp. Confidence measures for spontaneous speech recognition. In *Proc. ICASSP*, volume II, pages 875–878, Munich, Germany, April 1997.
11. 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.
12. A. Wendemuth. Learning the unlearnable. *J. Phys. A*, 28:5423, 1995.
13. F. Wessel, K. Macherey, and R. Schlueter. Using word probabilities as confidence measures. In *Proc. ICASSP*, volume 1, pages 225–228, May 1998.