INTERNATIONAL COMPUTER SCIENCE INSTITUTE

1947 Center Street • Suite 600 • Berkeley, California 94704 • 1-415-642-4274 • FAX 1-415-643-7684

PROBABILITY ESTIMATION BY FEED-FORWARD NETWORKS IN CONTINUOUS SPEECH RECOGNITION

Steve Renals, Nelson Morgan and Hervé Bourlard*

TR-91-030

28 August 1991

Abstract

We review the use of feed-forward networks as estimators of probability densities in hidden Markov modelling. In this paper we are mostly concerned with radial basis functions (RBF) networks. We note the isomorphism of RBF networks to tied mixture density estimators; additionally we note that RBF networks are trained to estimate posteriors rather than the likelihoods estimated by tied mixture density estimators. We show how the neural network training should be modified to resolve this mismatch. We also discuss problems with discriminative training, particularly the problem of dealing with unlabelled training data and the mismatch between model and data priors.

*L&H Speechproducts, Ieper, B-8900, Belgium

INTRODUCTION

In continuous speech recognition we wish to estimate $P(\mathbf{W}_1^W|\mathbf{X}_1^T, \mathbf{M})$, the posterior probability of a word sequence $\mathbf{W}_1^W = \mathbf{w}_1, ..., \mathbf{w}_W$ given the acoustic evidence $\mathbf{X}_1^T = \mathbf{x}_1, ..., \mathbf{x}_T$ and the parameters of the models used Θ. This probability cannot be estimated directly; however we may re-express it using Bayes' rule:

$$P(\mathbf{W}_1^W|\mathbf{X}_1^T, \Theta) = \frac{P(\mathbf{X}_1^T|\mathbf{W}_1^W, \Theta)P(\mathbf{W}_1^W|\Theta)}{P(\mathbf{X}_1^T|\Theta)} = \frac{P(\mathbf{X}_1^T|\mathbf{W}_1^W, \Theta)P(\mathbf{W}_1^W|\Theta)}{\sum_{\mathbf{W}'} P(\mathbf{X}_1^T|\mathbf{W}', \Theta)P(\mathbf{W}'|\Theta)}. \quad (1)$$

Equation (1) separates the problem into two components: acoustic modelling and language modelling. The language model is used to estimate the prior probability of a word sequence $P(\mathbf{W}_1^W|\Theta)$. The acoustic model is used to estimate the likelihood of the acoustic evidence given the word sequence $P(\mathbf{X}_1^T|\mathbf{W}_1^W, \Theta)$. The normalising denominator of (1) is constant at recognition time; however, during training it is not constant, as the parameters of the models are changing.

Each unit of speech is modelled by a hidden Markov model. A typical unit is the phone; word models consist of concatenations of phone HMMs, according to a phone-structured lexicon. A HMM is defined by a set of states q_l, a topology specifying allowed transitions between states and a set of local probability density functions (PDFs) $P(\mathbf{x}_t, q(t)|q(t-1), \mathbf{X}_1^{t-1})$. Making the further assumptions that the output at time t is independent of previous outputs and depends only on the current state, we may separate the local probabilities into state transition probabilities $p(q(t)|q(t-1))$ and output PDFs $P(\mathbf{x}_t|q(t))$. A set of initial state probabilities must also be specified.

The transition probabilities and the parameters of the output PDFs are frequently estimated using a maximum likelihood training procedure, the forward-backward algorithm (see e.g. [2]). This procedure is optimal if the true model is in the space of models being searched[1]. However, this is not the case for speech recognition. What is desired is not the best possible model of each class, but the best set of models for discrimination between classes. Thus, discriminative training would seem to be preferable to maximum likelihood training. In terms of (1), this means that the best acoustic model would be achieved by maximising the likelihood of the correct model, whilst simultaneously minimising the likelihoods of the competing models.

In practice, a full maximum likelihood procedure is rarely used for either recognition or training. Instead, the Viterbi criterion is used. Here, the maximisation of $P(\mathbf{X}_1^T|\mathbf{W}_1^W, \Theta)$ which should be computed over all allowable state sequences is replaced by an approximation that considers only the most probable state sequence. This computation may be efficiently performed using a dynamic programming algorithm. When used at recognition time this is referred to as Viterbi decoding.

[1] And if some other conditions are satisfied [11].

We have used discriminatively trained classifiers to estimate the output PDFs [5, 14, 17]. It may be shown that a "1-from–n" classifier trained using a relative entropy (or a least mean squares) objective function outputs the posterior probabilities, $P(q_l|\mathbf{x})$, of each class given the input data [6]. However, the likelihoods $P(\mathbf{x}|q_l)$ are required; the prior probabilities, $p(q_l)$ are given by the allowable sentence models constructed from the basic HMMs using a phone-structured lexicon and the language model. Likelihood estimates may be obtained simply by dividing the output posteriors by the relative frequencies of each class[2].

The classifiers we have used are layered, feed-forward networks: multi-layer perceptrons (MLPs) and radial basis function (RBF) networks. MLPs consist of layers of units that define a hyperplane over the space of the previous layer, followed by a "soft" transfer function (typically a sigmoid). The outputs of such hidden units may be considered as the probabilities of certain "facts" about the previous layer. An RBF network generally has a single hidden layer, whose units may be regarded as computing local (or approximately local) densities, rather than global decision surfaces. The resultant posteriors are obtained by output units that combine these local densities.

In this paper, we are mainly concerned with RBF networks. An isomorphism to tied mixture density modelling has been pointed out. We also remark on a mismatch between the posteriors estimated by discriminatively trained RBF networks and the likelihoods estimated in tied mixture density modelling. This mismatch is resolved by redefining the transfer function of the output units of the RBF network to implement Bayes' rule, relating the posterior to the likelihood. The issue of a mismatch between discriminative and maximum likelihood training is important and has implications regarding our current approach to HMM probability estimation. We survey this problem and discuss some possible solutions.

TIED MIXTURE HMM

Tied mixture density (or semi-continuous) HMMs have proven to be powerful PDF estimators in continuous speech recognition [13, 3]. This method may be regarded as intermediate between discrete vector-quantised methods and separate continuous PDF estimates for each state. If a unified formalism for both discrete and continuous HMMs is adopted, then tied mixture density modelling may be regarded as an interpolation between discrete and continuous modelling [3]. Essentially, tied mixture modelling has a single "codebook" of Gaussians shared by all output PDFs. Each of these PDFs has its own set of mixture coefficients used to combine the individual Gaussians. If $f_k(\mathbf{x}|q_k)$ is the output PDF of state q_k, and $N_j(\mathbf{x}|\mu_j, \Sigma_j)$ are the component

[2]These are the estimates of $p(q_l)$ implicitly used during classifier training.

Gaussians, then:

$$f_k(\mathbf{x}|q_k,\Theta) = \sum_j a_{kj} N_j(\mathbf{x}|\mu_j,\Sigma_j) \tag{2}$$

$$\sum_j a_{kj} = 1 \qquad 0 \le a_{kj} \le 1\,,$$

where a_{kj} is an element of the matrix of mixture coefficients (which may be interpreted as the prior probability $p(\mu_j,\Sigma_j|q_k)$) defining how much component density $N_j(\mathbf{x}|\mu_j,\Sigma_j)$ contributes to output PDF $f_k(\mathbf{x}|q_k,\Theta)$.

RADIAL BASIS FUNCTIONS

The radial basis functions (RBF) network was originally introduced as a means of function interpolation [16, 10]. A set of K approximating functions, $f_k(\mathbf{x})$ is constructed from a set of J basis functions $\phi(\mathbf{x})$:

$$f_k(\mathbf{x}) = \sum_{j=1}^{J} a_{kj}\phi_j(\mathbf{x}) \qquad 1 \le k \le K \tag{3}$$

This equation defines a RBF network with J RBFs (hidden units) and K outputs. The output units here are linear, with weights a_{kj}. The RBFs are typically Gaussians, with means μ_j and covariance matrices Σ_j:

$$\phi_j(\mathbf{x}) = R\exp\left(-\frac{1}{2}(\mathbf{x}-\mu_j)^T\Sigma_j^{-1}(\mathbf{x}-\mu_j)\right)\,, \tag{4}$$

where R is a normalising constant. The covariance matrix is frequently assumed to be diagonal[3].

Such a network has been used for HMM output probability estimation in continuous speech recognition [17] and an isomorphism to tied-mixture HMMs was noted. However, there is a mismatch between the posterior probabilities estimated by the network and the likelihoods required for the HMM decoding. Previously this was resolved by dividing the outputs by the relative frequencies of each state. It would be desirable, though, to retain the isomorphism to tied mixtures: specifically we wish to interpret the hidden-to-output weights of an RBF network as the mixture coefficients of a tied mixture likelihood function. This can be achieved by defining the transfer units of the output units to implement Bayes' rule, which relates the posterior $g_k(\mathbf{x})$ to the likelihood $f_k(\mathbf{x})$:

$$g_k(\mathbf{x}) \;=\; \frac{f_k(\mathbf{x})p(q_k)}{\sum_{l=1}^{K} f_l(\mathbf{x})p(q_l)}\,. \tag{5}$$

[3]This is often reasonable for speech applications, since mel or PLP cepstral coefficients are orthogonal.

Such a transfer function ensures the output units sum to 1; if $f_k(\mathbf{x})$ is guaranteed non-negative, then the outputs are formally probabilities. The output of such a network is a probability distribution and we are using '1-from-K' training: thus the relative entropy E is simply:

$$E = -\log g_c(\mathbf{x}), \tag{6}$$

where q_c is the desired output class (HMM distribution). Bridle has demonstrated that minimising this error function is equivalent to maximising the mutual information between the acoustic evidence and HMM state sequence [9].

If we wish to interpret the weights as mixture coefficients, then we must ensure that they are non-negative and sum to 1. This may be achieved using a normalised exponential (softmax) transformation:

$$a_{kj} = \frac{\exp(w_{kj})}{\sum_h \exp(w_{kh})}. \tag{7}$$

The mixture coefficients a_{kj} are used to compute the likelihood estimates, but it is the derived variables w_{kj} that are used in the unconstrained optimisation.

Training

Steepest descent training specifies that:

$$\frac{\partial w_{kj}}{\partial t} = -\frac{\partial E}{\partial w_{kj}}. \tag{8}$$

Here E is the relative entropy objective function (6). We may decompose the right hand side of this by a careful application of the chain rule of differentiation:

$$\frac{\partial E}{\partial w_{kj}} = \sum_{l=1}^{K} \frac{\partial E}{\partial g_l(\mathbf{x})} \frac{\partial g_l(\mathbf{x})}{\partial f_k(\mathbf{x})} \sum_{h=1}^{J} \frac{\partial f_k(\mathbf{x})}{\partial a_{kh}} \frac{\partial a_{kh}}{\partial w_{kj}}. \tag{9}$$

We may write down expressions for each of these partials (where δ_{ab} is the Kronecker delta and q_c is the desired state):

$$\frac{\partial E}{\partial g_l(\mathbf{x})} = -\frac{\delta_{cl}}{g_c} \tag{10}$$

$$\begin{aligned} \frac{\partial g_l(\mathbf{x})}{\partial f_k(\mathbf{x})} &= \frac{\delta_{kl} p(q_k)}{\sum_{i=1}^{K} f_i(\mathbf{x}) p(q_i)} - \frac{p(q_k) f_l(\mathbf{x}) p(q_l)}{(\sum_{i=1}^{K} f_i(\mathbf{x}) p(q_i))^2} \\ &= \frac{p(q_k)}{\sum_{i=1}^{K} f_i(\mathbf{x}) p(q_i)} (\delta_{kl} - g_l) \\ &= \frac{g_k(\mathbf{x})}{f_k(\mathbf{x})} (\delta_{kl} - g_l) \end{aligned} \tag{11}$$

$$\frac{\partial f_k(\mathbf{x})}{\partial a_{kh}} = \phi_h(\mathbf{x}) \tag{12}$$

$$\frac{\partial a_{kh}}{\partial w_{kj}} = a_{kh}(\delta_{hj} - a_{kj}). \tag{13}$$

Substituting (10), (11), (12) and (13) into (9) we obtain:

$$\begin{aligned}(14)\qquad \frac{\partial E}{\partial w_{kj}} &= -\frac{1}{g_c(\mathbf{x})}\frac{g_k(\mathbf{x})}{f_k(\mathbf{x})}\left(\delta_{kc} - g_c(\mathbf{x})\right) a_{kj}\left(\phi_j(\mathbf{x}) - \sum_{h=1}^{J}\phi_h(\mathbf{x})a_{kh}\right) \\ &= \frac{1}{f_k(\mathbf{x})}\left(g_k(\mathbf{x}) - \delta_{kc}\right) a_{kj}\left(\phi_j(\mathbf{x}) - \sum_{h=1}^{J}\phi_h(\mathbf{x})a_{kh}\right) \\ &= \frac{1}{f_k(\mathbf{x})}\left(g_k(\mathbf{x}) - \delta_{kc}\right) a_{kj}\left(\phi_j(\mathbf{x}) - f_k(\mathbf{x})\right)\ .\end{aligned}$$

The expression is simpler if we ignore the constraints on the weights (i.e. if $w_{kj} = a_{kj}$), although $f(\mathbf{x})$ is no longer guaranteed to be a PDF:

$$(15)\qquad \frac{\partial E}{\partial w_{kj}} = \frac{1}{f_k(\mathbf{x})}\left(g_k(\mathbf{x}) - \delta_{kc}\right)\phi_j(\mathbf{x})\,.$$

The only difference between this gradient and the one obtained using a sigmoid output transfer function with a relative entropy objective function is the $1/f_k(\mathbf{x})$ factor, which may be regarded as a 'dimensional artifact'.

The required gradient is simpler if we construct the network to estimate log likelihoods, replacing $f_k(\mathbf{x})$ with $z_k(\mathbf{x}) = \log f_k(\mathbf{x})$:

$$(16)\qquad z_k(\mathbf{x}) = \sum_j w_{kj}\phi_j(\mathbf{x})$$

$$(17)\qquad g_k(\mathbf{x}) = \frac{p(q_k)\exp(z_k(\mathbf{x}))}{\sum_l p(q_l)\exp(z_l(\mathbf{x}))}\,.$$

Since this is in the log domain, no constraints on the weights are required. The new gradient we need is:

$$(18)\qquad \frac{\partial g_l(\mathbf{x})}{\partial f_k(\mathbf{x})} = g_l\left(\delta_{kl} - g_k\right)\ .$$

Thus the gradient of the error is:

$$(19)\qquad \frac{\partial E}{\partial w_{kj}} = \left(g_k(\mathbf{x}) - \delta_{ck}\right)\phi_j(\mathbf{x})\,.$$

Since we are in log domain, the "$1/f_k(\mathbf{x})$" factor is additive and thus disappears from the gradient. This network is similar to Bridle's softmax, except here uniform priors are not assumed; the gradient is of identical form, though. However in this case the weights do not have a simple relationship with the mixture coefficients obtained in tied mixture density modelling: thus we use the likelihood estimation of (3) and (5).

We may also train the means and variances of the RBFs by back-propagation of error; alternatively they can be trained by some self-organising process. The relevant

partials for gradient descent training are (assuming a diagonal covariance matrix with diagonal elements σ_{ji}):

$$\frac{\partial \phi_j(\mathbf{x})}{\partial \mu_{ji}} = \frac{\phi_j(\mathbf{x})(x_i - \mu_{ji})}{\sigma_{ji}} \tag{20}$$

$$\frac{\partial \phi_j(\mathbf{x})}{\partial \sigma_{ji}} = \frac{-\phi_j(\mathbf{x})(x_i - \mu_{ji})^2}{2\sigma_{ji}^2} \tag{21}$$

If the determinant of the covariance matrix $\det(\Sigma_j)$ is used as a scale factor for $\phi_j(\mathbf{x})$, then (4) becomes:

$$\phi_j(\mathbf{x}) = \frac{R}{\det(\Sigma_j)^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x} - \mu_j)^T \Sigma_j^{-1} (\mathbf{x} - \mu_j)\right), \tag{22}$$

and (21) becomes:

$$\frac{\partial \phi_j(\mathbf{x})}{\partial \sigma_{ji}} = \frac{-\phi_j(\mathbf{x})}{2\sigma_{ji}} \left(\frac{(x_i - \mu_{ji})^2}{\sigma_{ji}} + 1\right) \tag{23}$$

These expressions, used with the back-propagation algorithm, enable us to adapt the means and covariances in a discriminative fashion.

GLOBAL OPTIMISATION

The above methods for HMM probability density estimation involve only a local optimisation of parameters. In speech recognition training we typically have a small amount of labelled training data (used for model bootstrapping) and a large amount of unlabelled training data. (Here labelled training data refers to speech labelled and time-aligned at a phone level; unlabelled training data refers to speech for which only the (non-time-aligned) word sequence is available.) The local optimisation we have used has involved an initial maximum likelihood (or Viterbi) training to generate a prototype segmentation of the unlabelled data. These labels are then used as the targets for neural network training (performed on a framewise basis). This is a local training, since only the most likely path given the initial parameter estimation is considered.

One approach to a global optimisation method is analogous to segmental k-means training. In this method after an initial network training on labelled data and Viterbi segmentation, the targets used in training the unlabelled data are updated by performing a Viterbi segmentation after each epoch of discriminative training. Such an approach has been referred to as embedded MLP [5] or connectionist Viterbi training [12]. It should be noted that the transition probabilities are still optimised by a maximum likelihood criterion (or the Viterbi approximation to it). It may be proved that performing a Viterbi segmentation using posterior local probabilities will also

result in a global optimisation [6]: however, there is a mismatch between model and data priors here (see next section).

It is possible to attempt a global optimisation in which all the parameters of the HMM are optimised simultaneously according to some discriminative criterion. Such an approach was first proposed by Bahl et al. [1] who presented a training scheme for continuous HMMs in which the mutual information between the acoustic evidence and the word sequence was maximised using gradient descent. More recently, Bridle introduced the "alphanet" representation [8] of HMMs, in which the computation of the HMM "forward" probabilities $\alpha_{jt} = P(X_1^t, q(t) = j)$ is performed by the forward dynamics of a recurrent network. Alphanets may be discriminatively trained by minimising a relative entropy objective function. This function has similar form to (6) (i.e. the negative log of the posterior of the correct output): however here we are looking at the global posterior probability of the word sequence given the acoustic evidence $P(\mathbf{W}_1^W|\mathbf{X}_1^T, \Theta)$ (1), rather than the local posterior of a state given the one frame of acoustic evidence. From (1), this posterior is the ratio of the likelihood of the correct model to the sum of the likelihoods of all models. For continuous speech, a model here refers to a sentence model; thus the numerator is the quantity computed by the forward-backward algorithm in training mode (when the word sequence is constrained to be the correct word sequence, so only time-warping variations are considered). The denominator involves a sum over all possible models: this is equivalent to the sum computed if the forward-backward algorithm were to be run at recognition time (with the only constraints over the word sequence provided by the language model). Computation of this quantity would be prohibitive for both training and recognition. A simpler quantity to compute is just the sum over all possible phoneme sequences (unconstrained by language model). This is not desirable as it assumes uniform priors rather than those specified by the language model.

Initial work in using global optimisation methods for continuous speech recognition has been performed by Bridle [7] and Bengio [4]; both of these involved training the parameters of the HMM by a maximum likelihood process, using the "alphanets" method to optimise the input parameters via some (linear or non-linear) transform.

PROBLEMS WITH DISCRIMINATIVE TRAINING

It has been shown, both theoretically and in practice, that the training and recognition procedures used with standard HMMs remain valid for posterior probabilities [6]. Why then do we replace these posterior probabilities with likelihoods?

The answer to this problem lies in a mismatch between the prior probabilities given by the training data and those imposed by the topology of the HMMs. Choosing the HMM topology also amounts to fixing the priors. For instance, if classes q_k represent phones, prior probabilities $p(q_k)$ are fixed when word models are defined as particular sequences of phone models. This discussion can be extended to different levels of processing: if q_k represents sub-phonemic states and recognition is constrained by a

language model, prior probabilities q_k are fixed by (and can be calculated from) the phone models, word models and the language model. Ideally, the topologies of these models would be inferred directly from the training data, by using a discriminative criterion which implicitly contains the priors. Here, at least in theory, it would be possible to start from fully-connected models and to determine their topology according to the priors observed on the training data. Unfortunately this results in a huge number of parameters that would require an unrealistic amount of training data to estimate them significantly. This problem has also been raised in the context of language modelling [15].

Since the ideal theoretical solution is not accessible in practice, it is usually better to dispose of the poor estimate of the priors obtained using the training data, replacing them with "prior" phonological or syntactic knowledge.

A second problem arises from a mismatch between the maximum likelihood and discriminant criteria. As is well known, if the models are correct, then maximum likelihood training is optimal. In speech recognition, we use discriminative training because it is known that the models being used are incorrect. The use of unlabelled data highlights a contradiction in our current training methodology. To give unlabelled data the labels that discriminative training requires, the current best model estimates are used. Thus discriminative training is employed because of a belief that the models are incorrect, yet the labels used by the discriminative training assume model correctness.

It maybe that this mismatch is responsible for the lack of robustness of discriminative training (compared with pure maximum likelihood training) in vocabulary independent speech recognition tasks [15]. The assumption of model correctness used to generate the labels may have the effect of further embedding specifics of the training data into the final models.

CONCLUSION

We have a defined a feed-forward network that estimates Gaussian mixture densities using a discriminative training criterion. Additionally we have discussed a mismatch between maximum likelihood and discriminative training that is inherent in many discriminative training schemes. We are currently performing speech recognition experiments using the RBF networks and training procedure described above.

ACKNOWLEDGEMENTS

Thanks to David MacKay and Richard Durbin for good discussions.

REFERENCES

[1] Lalit R. Bahl, Peter F. Brown, Peter V. de Souza, and Robert L. Mercer. Maximum mutual information estimation of hidden Markov model parameters for speech recognition. In *Proceedings IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 49–52, Tokyo, 1986.

[2] Lalit R. Bahl, Frederick Jelinek, and Robert L. Mercer. A maximum likelihood approach to continuous speech recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-5:179–190, 1983.

[3] Jerome R. Bellegarda and David Nahamoo. Tied mixture continuous parameter modeling for continuous speech recognition. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 38:2033–2045, 1990.

[4] Yoshua Bengio, Renato de Mori, Giovammi Flammia, and Ralf Kompe. Global optimization of a neural network - hidden Markov model hybrid. Technical Report TR-SOCS-90.22, McGill University School of Computer Science, 1990.

[5] H. Bourlard and N. Morgan. A continuous speech recognition system embedding MLP into HMM. In David S. Touretzky, editor, *Advances in Neural Information Processing Systems*, volume 2, pages 413–416. Morgan Kaufmann, San Mateo CA, 1990.

[6] H. Bourlard and C. J. Wellekens. Links between Markov models and multilayer perceptrons. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-12:1167–1178, 1990.

[7] J. S. Bridle and L. Dodd. An alphanet aproach to optimising input transformations for continuous speech recognition. In *Proceedings IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 277–280, Toronto, 1991.

[8] John S. Bridle. Alpha-nets: a recurrent neural network architecture with a hidden Markov model interpretation. *Speech Communication*, 9:83–92, 1990.

[9] John S. Bridle. Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimation of parameters. In David S. Touretzky, editor, *Advances in Neural Information Processing Systems*, volume 2, pages 211–217. Morgan Kaufmann, San Mateo CA, 1990.

[10] D. S. Broomhead and David Lowe. Multi-variable functional interpolation and adaptive networks. *Complex Systems*, 2:321–355, 1988.

[11] Peter F. Brown. *The Acoustic-Modelling Problem in Automatic Speech Recognition.* PhD thesis, School of Computer Science, Carnegie Mellon University, 1987.

[12] Michael A. Franzini, Kai-Fu Lee, and Alex Waibel. Connectionist Viterbi training: a new hybrid method for continuous speech recognition. In *Proceedings IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 425–428, Albuquerque, 1990.

[13] X. D. Huang and M. A. Jack. Semi-continuous hidden Markov models for speech signals. *Computer Speech and Language*, 3:239–251, 1989.

[14] N. Morgan, H. Hermansky, H. Bourlard, C. Wooters, and P. Kohn. Continuous speech recognition using PLP analysis with multi-layer perceptrons. In *Proceedings IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 49–52, Toronto, 1991.

[15] Douglas B. Paul, James K. Baker, and Janet M. Baker. On the interaction between true source, training and testing language models. In *Proceedings IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 569–572, Toronto, 1991.

[16] M. J. D. Powell. Radial basis functions for multi-variable interpolation: a review. Technical Report DAMPT/NA12, Dept. of Applied Mathematics and Theoretical Physics, University of Cambridge, 1985.

[17] Steve Renals, David McKelvie, and Fergus McInnes. A comparative study of continuous speech recognition using neural networks and hidden Markov models. In *Proceedings IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 369–372, Toronto, 1991.