A NEURAL NETWORK FOR CLASSIFICATION WITH INCOMPLETE DATA: APPLICATION TO ROBUST ASR

Andrew C. Morris[1], *Ljubomir Josifovski*[3], *Hervé Bourlard*[1,2], *Martin Cooke*[3], *Phil Green*[3]

[1]Dalle Molle Institute for Perceptual Artificial Intelligence (IDIAP)
[2]Swiss Federal Institute of Technology (EPFL), Lausanne, CH
[3]Sheffield University, Dept. of Computer Science, UK

ABSTRACT

There are many situations in data classification where the data vector to be classified is partially corrupted, or otherwise incomplete. In this case the optimal estimate for each class probability output, for any given set of missing data components, can be obtained by calculating its expected value. However, this means that classifiers whose expected outputs do not have a closed form expression in terms of the original function parameters, such as the commonly used multi-layer perceptron (MLP), cannot be used for classification with missing data. No classifier can compete with the performance of an MLP on complete data unless it is discriminatively trained. In this paper we present a particular form of RBF classifier which can be discriminatively trained and whose expected outputs are a simple function of the original classifier parameters, even though the output unit function is non-linear. This provides us with an incomplete data classifier network (IDCN) which combines the discriminative classification performance normally associated with artificial neural networks, with the ability to deal gracefully with missing data. We describe two ways in which this IDCN can be applied to robust automatic speech recognition (ASR), depending on whether or not the position of missing data is known. We compare the performance of one of these models with an existing system for ASR with missing data.

Keywords: missing features, robust recognition, neural networks

1. INTRODUCTION

In any realistic automatic recognition task it is common that part of the input feature vector x to be classified is corrupted by some kind of noise process, and the recognition performance of a system which is not trained to expect this kind of noise will degrade dramatically as the noise level increases. In many cases this problem can be reduced by applying some kind of noise removal or data enhancement process. But there are also many situations in which some feature components are irretrievable. The approach taken in this case depends on to what extent it is possible to identify which features have been corrupted.

If the position of missing features is given, then the estimate for the posterior probability for each class, which is best in the sense that it gives the maximum probability of correct classification, can be obtained as the expected value of the classifier output for that class, conditioned by any available constraints on the missing data [11]. The main problem with this approach is that for most classifiers, the expected value of the class probability outputs cannot be obtained as a simple closed form expression from the classifier parameters.

If the position of missing data is not known, one successful approach [12,13] has been to train a separate classifier for each possible position of missing data and then to combine the posteriors for one class as a weighted sum over all classifiers. Even with equal weights this approach shows some robustness to missing data, because “uncertain” classifiers tend to contribute equal and therefore small probabilities to each class. The problem with this approach is that the number of different possible positions of missing data is generally far too large to allow training of a separate classifier for each position.

In this paper we present a particular form of RBF classifier in which the output layer uses Bayes’ Rule to directly transform pooled mixture likelihoods from the RBF layer into a-posteriori class probabilities [2,3,8]. Even though the output units are non-linear, the expected outputs of this classifier, for any given missing data components, are a simple function of the original classifier parameters. The use of closely related RBF networks for recognition with missing data is not new [1], but to the authors’ knowledge the particular form of incomplete data classification network (IDCN) described here has not been used before in either of the techniques presented in this article.

In Section 2 we present the IDCN architecture, and describe how it can be applied, either as a new kind of HMM/ANN hybrid if the position of missing data is known, or else as a new form of multistream HMM/ANN. In Section 3 we discuss various ways in which the IDCN can be trained, and give full details of the way it was trained for the ASR tests which are presented in Section 5. Section 4 shows how network outputs (class posterior probabilities) are calculated when some of the input features are missing. Section 5 describes tests made for an HMM/IDCN system (position of missing data given), and compares results to those from a previous HMM based system using likelihood based missing feature theory, and *identical* missing data masks. Finally in Section 6 these results are briefly discussed and new ways forward are suggested.

2. IDCN ARCHITECTURE

The network has one input, one hidden and one output layer, as shown in Fig.1. Each RBF unit y_j in the hidden layer uses a

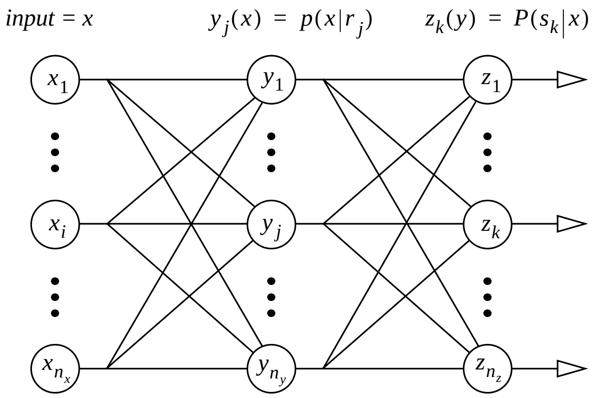

Figure 1: *RBF network used here for classification with incomplete data. The output layer uses Bayes' Rule to directly transform pooled mixture likelihoods from the RBF layer into a-posteriori class probabilities.*

diagonal covariance Gaussian $y_j(x)$ to model the probability density $p(x|r_j)$ for input vector x having been generated by this Gaussian, while each output unit uses a function $z_k(x)$ to model the posterior probability that x is from output class k. If r_j denotes that x was generated by Gaussian j, and s_k that x is from class k, then:

$$y_j(x) = p(x|r_j) = N(x, \mu_j, v_j) \quad (1)$$

$$z_k(x) = P(s_k|x) = \frac{p(x, s_k)}{p(x)} = \frac{net_k}{p(x)} \quad (2)$$

where

$$net_k = \sum_j P(r_j, s_k)p(x|r_j) = \sum_j w_{jk}y_j \quad (3)$$

$$p(x) = \sum_{j,k} p(x, r_j, s_k) = \sum_k net_k \quad (4)$$

Although the above structure of the IDCN does not change, the way in which it is applied depends on whether the position of missing input data is known.

2.1 Position of missing data given

The IDCN can be used as a front end to a conventional HMM based ASR system, whereby the likelihoods which are normally calculated from the Gaussian mixture models for each hidden state are replaced, during decoding, by scaled likelihoods from the IDCN. This comprises a form of HMM/ANN based ASR system [3] suitable for use with missing data, and is the approach tested in this paper.

2.2 Position of missing data unknown

In principle a single IDCN can be used to replace the 2^d different ANN experts which are normally required [13] to cover all possible selections of missing features from a d dimensional feature vector. Provided that the combined features input to each expert are merely concatenated (i.e. no compression, orthogonalisation, or whatever is applied), the marginal posteriors for each position of missing features can be computed directly from the IDCN parameters, and then simply combined in a linearly weighted sum [12] or geometrically weighted product [5].

3. IDCN TRAINING

Classifier parameters to be trained are the mean and variance vectors in Eq.(1) for each Gaussian RBF unit, and the output layer weights, w, in Eq.(3).

In order for the performance of this classifier to compete with that of the MLP, it is essential that all parameters are trained together, and with a discriminative objective function. Unsupervised discriminative training is also possible, using minimum classification error techniques [9]. However, in this article we take the simpler approach of training by supervised gradient descent. During training the softmax function is used to constrain the weights $w_{jk} = P(r_j, s_k)$ to lie in $[0, 1]$, and sum to one.

$$w_{jk} = e^{\alpha_{jk}} / \sum_{l,m} e^{\alpha_{lm}} \quad (5)$$

This gives the full set of parameters to be trained as $(\mu_{ij}, v_{ij}, \alpha_{jk})$, for $i = 1 \ldots n_x$, $j = 1 \ldots n_y$, $k = 1 \ldots n_z$.

3.1 Parameter initialisation

Any hill climbing procedure can encounter problems with local minima, so that system performance may be very sensitive to the initial parameter values used. The following two methods were tested for initialising the RBF layer parameters (Gaussian means, variances and priors $P(r_j)$):

- Randomly assign each data point to an RBF centre, followed by k-means clustering and likelihood maximisation by Expectation Maximisation (EM).
- Use HTK 1.5 [17] to train a set of pooled Gaussians, using the Baum-Welch forward-backward training algorithm, with embedded realignment.

HTK also trains mix weights $P(r_j|s_k)$ for each of the hidden states as specified by whatever HMM structure is to be used in recognition. Whichever of the above methods was used, the trained HMM model was also used to provide a training data segmentation, from which we can estimate $P(s_k)$. Once the Gaussian parameters were initialised, two methods were tested for initialising the weights w, using the given segmentation:

- Use HMM trained mix weights $P(r_j|s_k)$ only:

$$w_{jk} = P(r_j, s_k) = P(r_j|s_k)P(s_k) \quad (6)$$

- Also use HMM trained Gaussians [10]:

$$P(r_j) = \sum_k P(r_j, s_k) \quad (7)$$

$$w_{jk} = P(r_j) \sum_{x_i \in s_k} y_j(x_i) / \sum_{i=1}^{N} y_j(x_i) \quad (8)$$

Of these different RBF layer and output layer initialisation methods, the best results *by far* were obtained using RBFs trained using HTK, and output weights trained using Eq.(8).

Auxiliary parameters α were then initialised as:

$$\alpha_{jk} = \log(w_{jk}) \tag{9}$$

3.2 Error gradient calculation

Whichever error function E is used, the derivatives of E with respect to each of the model parameters were obtained by the usual “error back propagation” (EBP) approach, first calculating the “delta” values for each output unit [10]:

$$\delta_k = \frac{\partial E}{\partial net_k} = \sum_l \frac{\partial E}{\partial z_l} \frac{(\delta_{kl} - z_l)}{p(x)} \tag{10}$$

$$\frac{\partial E}{\partial \mu_{ij}} = \frac{(x_i - \mu_{ij})}{v_{ij}} y_j \sum_k w_{jk} \delta_k \tag{11}$$

$$\frac{\partial E}{\partial v_{ij}} = \left(\frac{(x_i - \mu_{ij})^2}{v_{ij}} - 1 \right) \frac{y_j}{2v_{ij}} \sum_k w_{jk} \delta_k \tag{12}$$

$$\frac{\partial E}{\partial \alpha_{jk}} = w_{jk}(1 - w_{jk}) y_j \delta_k \tag{13}$$

If τ_l is the target posterior for class l, then for three common error functions we have $\frac{\partial E}{\partial z_l}$ as:

$$z_l - \tau_l \qquad \text{: mean square error} \tag{14}$$

$$-\tau_l / z_l \qquad \text{: cross-entropy} \tag{15}$$

$$-\tau_l \qquad \text{: correlation} \tag{16}$$

Best results here used the cross-entropy objective.

3.3 Gradient descent iteration

A constant “momentum” factor $\theta = 1$ was used, and an adaptable learning rate, φ [4]. With $g = \nabla E$, $g_0 = 0$, $dw_0 = 0$, $\varphi_0 = 1$, we have $\varphi_{t+1} = \varphi_t(1 - 0.5\widehat{dw}_t \cdot \hat{g}_t)$, $dw_{t+1} = \varphi_{t+1}(\theta dw_t - \hat{g}_t)$, $w_{t+1} = w_t + dw_{t+1}$ (where $\hat{g}$ is a unit vector). Training continued until the correct state classification rate on the cross-validation set stopped increasing.

The gradient with respect to *all* IDCN parameters was evaluated, and all parameters updated, using all frames from a fixed number of utterances selected at random from the full training set. We found that very small samples led rapidly to one or more RBFs developing zero priors, from which they could not escape. As a compromise between processing speed and performance level at convergence, we settled on samples of 100 utterances.

It was found that further training of the RBF parameters by EM, after gradient descent training had converged, followed by application of Eq.(8), resulted in a rapid increase in data likelihood. However, this was inevitably accompanied by a dramatic fall in classification accuracy, so this technique was not used.

4. RECOGNITION WITH MISSING DATA

If the position of missing features is given, and the present and missing components of the feature vector are denoted (x_p, x_m), then the estimate for $P(s_k|x)$ which results in the highest probability of correct classification is given by the expected value of the classifier output function, conditioned on x_p and any knowledge κ_m which may constrain missing data values [11]. For the present classifier this leads to the following missing data posterior probability estimates [10].

If nothing is known about the missing data then:

$$\hat{z}_k(x) = E[P(s_k|x)|x_p] \propto \sum_j w_{jk} y_j(x_p) \tag{17}$$

If each missing feature has a limited range of possible values (as is the case for filterbank features, which are bounded below by zero and above by their observed value):

$$\hat{z}_k(x) = E[P(s_k|x)|x_p, x_m \in r_m]$$

$$\propto \sum_j w_{jk} y_j(x_p) \int_{r_m} y_j(x_m) dx_m \tag{18}$$

In Eqs. (17) and (18) $y_j(x_p)$ is the marginal diagonal Gaussian over the indicated x components. Posteriors $\hat{z}_k(x)$ are obtained by scaling the above values to sum to one across all classes.

It should be noted that it is only due to the consistent probabilistic interpretation of each stage of processing by this network that it is so simple to obtain the marginal posteriors in this way directly from the full system parameters.

5. RECOGNITION TESTS

The training and test data is the adult male section of the TIDIGITs digit sequences database. This has a total of 4235 utterances (77 each from 55 men from 21 dialect regions of the USA). A subset of the training set was selected for cross-validation. So as to test for speaker independence, this has all 77 examples from each of 5 speakers (ff,eh,gt,cr,gj) from 5 of the most typical dialect regions. Acoustic vectors use 24 mel scaled HTK filterbank features, with first different coefficients, using 25 ms windows at 10 ms centres. Posteriors from the IDCN are converted to log scaled likelihood (LSL) features (by dividing by their class priors $P(s_k)$, then taking the logarithm) then passed to a normal HMM decoder in place of log state likelihoods [3]. Figure 2 compares recognition accuracy results for this HMM/IDCN system (posteriors based missing-data recognition) with those for the likelihood based system previously reported in [15], using *identical* missing-data masks.

6. SUMMARY AND CONCLUSION

We have shown that an RBF network, in which the output layer uses Bayes’ Rule to directly transform pooled mixture likelihoods from the RBF layer into a-posteriori class probabilities, is a suitable candidate network for classification with missing data. This is because it can be discriminatively trained, and the expected values of its posterior class probability

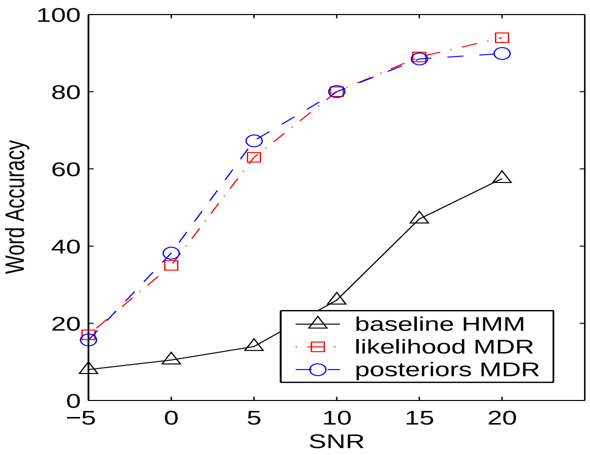

Figure 2: *Recognition accuracy (for TIDigits) against SNR (Noisex factory noise) for a standard HMM with no missing-data recognition (MDR), for a previously reported likelihood based MDR system, and for IDCN posteriors based MDR.*

outputs can readily be evaluated as a simple function of the original model parameters. We have further shown how this network can be incorporated into two different approaches to robust ASR. For the case where the position of missing data is known we have integrated the IDCN into an HMM system and compared its performance with a previous likelihood based implementation of the same missing feature theory. In these initial tests the performance of the two systems was very similar. In some ways this is not surprising, because both of these posteriors and likelihood based missing-data recognition systems make use of the same pool of Gaussians. However, it was expected that the posteriors based system would show some advantage due to the advantage of discriminative training. Severe problems were encountered with local minima during IDCN training by gradient descent. It is possible that the performance of the IDCN could be improved by use of a more effective discriminative training procedure, such as MCE [9] and/or boosting [14].

When the position of missing data is not known, the IDCN offers a new approach to multi-stream processing which should permit large numbers of feature streams to be combined with greatly reduced effort, but this approach remains to be tested.

ACKNOWLEDGEMENTS

This work was supported by the EC/OFES (European Community / Swiss Federal Office for Education and Science) RESPITE project (REcognition of Speech by Partial Information TEchniques). Ljubomir Josiofvski's doctoral work was also supported by Motorola.

REFERENCES

1. Ahmed, S. & Tresp, V. (1993) “Some solutions to the missing feature problem in vision”, in Advances in Neural Information Processing Systems 5, Morgan Kauffman, San Mateo, pp. 393-400.
2. Bishop, C. (1995) Neural Networks for Pattern Recognition, Clarendon Press, Oxford.
3. Bourlard, H. & Morgan, N. (1993) Connectionist Speech Recognition, Kluwer Academic Publishers, Boston.
4. Chan, L.W. & Fallside, F. (1987) “An adaptive training algorithm for back-propagation networks”, Computer Speech and Language 2, pp.205-218.
5. Hagen, A. & Morris, A.C. (in press) “Comparison of HMM experts with MLP experts in the full combination multi-band approach to robust ASR”, Proc. ICSLP-2000.
6. Hagen, A., Morris, A.C. & Bourlard, H. (1998) “Sub-band based speech recognition in noisy conditions: The Full-Combination approach”, Research Report IDIAP-RR 98-15.
7. Hermansky, H., Ellis, D. & Sharma, S. (2000) “Tandem connectionist feature stream extraction for conventional HMM systems”, Proc. ICASSP-2000, pp.1635-1638..
8. Lippmann, R. P. & Carlson, B. A. (1997) “Using missing feature theory to actively select features for robust speech recognition with interruptions, filtering and noise”, Proc. Eurospeech’97, pp. 37-40
9. McDermott, E. & Katagiri, S. (1994) “Prototype-based minimum classification error / generalised probabilistic descent training for various speech units”, Computer Speech and Language, No.8, pp.351-368.
10. Morris, A.C. (2000) “A neural network for classification with missing data”, Research Report IDIAP RR 00-23.
11. Morris, A.C., Cooke, M. & Green, P. (1998) “Some solutions to the missing feature problem in data classification, with application to noise robust ASR”, Proc. ICASSP'98, pp.737-740.
12. Morris, A.C., Hagen, A. & Bourlard, H. (1999) “The full-combination subbands approach to noise robust HMM/ANN based ASR”, Proc. Eurospeech’99, pp.599-602.
13. Morris, A.C., Hagen, A., Glotin, H. & Bourlard, H. (in press) “Multi-stream adaptive evidence combination for noise robust ASR”, Speech Communication.
14. Schwenk, H. (1999) “Using boosting to improve a hybrid HMM/neural network speech recogniser”, Proc. ICASSP’99. pp.1009-1012.
15. Vizinho, A., Green, P., Cooke, M. & Josifovski, L. (1999) “Missing data theory, spectral subtraction and signal-to-noise estimation for robust ASR: An integrated study”, Proc. Eurospeech’99, pp.2407-2410.
16. White, H. (1989) “Learning in artificial neural networks: A statistical perspective”, Neural Computing, 1, pp.425-464.
17. Young, S.J. & Woodland, P.C. (1993) HTK Version 1.5: User, Reference and Programmer Manual, Cambridge University Engineering Dept., Speech Group.