WRITER ADAPTATION OF A HMM HANDWRITING RECOGNITION SYSTEM

Andrew Senior and Krishna Nathan

IBM T. J. Watson Research Center, Yorktown Heights, NY 10598, USA.
{aws,ksn}@watson.ibm.com

ABSTRACT

This paper describes a scheme to adapt the parameters of a tied-mixture, hidden Markov model, on-line handwriting recognition system to improve performance on new writers' handwriting. The means and variances of the distributions are adapted using the Maximum Likelihood Linear Regression technique [1,2]. Experiments are performed with a number of new writers in both supervised and unsupervised modes. Adaptation on data quantities as small as 5 words is found to result in models with 6% lower error rate than the writer independent model.

1. INTRODUCTION

In on-line handwriting recognition, the goal is to recognize what has been written with an electronic stylus. The stylus returns a sequence of coordinates, recording the trajectory of the pen while in contact with a tablet surface. Because individual handwriting varies tremendously from person to person, a system trained to recognize one person's writing will generally perform poorly when another person writes, and it is difficult to make a system that will perform well on every person's handwriting.

It has been found that relatively small quantities of data (a few hundred words [4]) can be used to train a model to one writer's style, but in many situations it is impractical to collect even this amount data from a user, and it is rarely practical to obtain data with a correct transcription as would be ideal. However, it is observed that, examining just a few words of a person's handwriting we can immediately get a good idea of their writing style and know what shapes they use for most of their letters. This means that it might be possible to use just a few words of data to characterize a person's writing and to *adapt* a general model to perform better for this individual.

A number of techniques involving transformations or maximum *a posteriori* statistics have been proposed for adapting hidden Markov model speech recognition systems to an individual speaker, by changing the parameters of a speaker independent model which has been trained on a large pool of speakers. In this paper we investigate one method which has performed well in speech tasks. Since our aim is to achieve adaptation with small quantities of data, we use a transformation based approach.

The handwriting recognition model used for these experiments is a writer independent system trained on data from over 100 writers. The system is a continuous density Gaussian mixture hidden Markov model system, with the distribution for each state being a mixture from a common pool of 750 Gaussians with diagonal covariance matrices [3]. The words written on the system are represented by a sequence of vectors called *frames* each of which encodes the shape of a short section of pen trajectory.

2. MAXIMUM LIKELIHOOD LINEAR REGRESSION

The frames of handwritten data for an individual writer will have a different probability distribution to the frames derived from data collected from a large pool of writers. Some shapes will occur far more frequently for the writer, some shapes will never occur, and new shapes, not seen in the writing of the pool of writers, may be observed. Maximum likelihood linear regression (MLLR [2]) seeks to apply a transformation to the model to achieve the maximum likelihood match between the probability distribution of a Writer Independent model, and the frames of data seen in a small adaptation sample of a new writer's handwriting. There is no guarantee that any limited class of transforms will perform such a mapping well, but a linear transformation is applied as an approximation. When more data are available, different transforms can be estimated for different areas of the feature space resulting in a piecewise-linear transformation that gives a better approximation to the writer's distribution; goodness-

of-fit being judged based on the likelihood of the adaptation data given the new model parameters.

To adapt the mean, μ, of a Gaussian distribution, a linear transform, W is applied:

$$\hat{\mu} = W\mu \quad (1)$$

The transform is estimated as the maximum likelihood transform using the following auxiliary equation:

$$\mathcal{Q}(\mathcal{M}, \hat{\mathcal{M}}) = K' - \frac{1}{2}\mathcal{L}(\mathbf{o}|\mathcal{M}) \sum_{\mathbf{m}} \sum_{\tau} \mathbf{P}_\tau(\mathbf{m}|\mathcal{M}, \mathbf{o}) \quad (2)$$
$$\left(K_m + \log(\prod_i \sigma_i^2) + \sum_i \frac{(o_i(\tau) - \hat{\mu}_{m_i})^2}{2\sigma_i^2} \right),$$

where the K are constants not depending on the means. The derivation of the transformation is given in detail elsewhere [2].

A separate transform can also be estimated for the variances [1]. Since the model used here has diagonal covariance matrices, each dimension's variance is scaled separately:

$$\hat{\sigma}_i^2 = c_i \sigma_i^2 \quad (3)$$
$$\mathcal{Q}(\mathcal{M}, \hat{\mathcal{M}}) = K_1 - \quad (4)$$
$$\frac{1}{2}\mathcal{L}(\mathbf{o}|\mathcal{M}) \sum_{\mathbf{m}} \sum_{\tau} \mathbf{P}_\tau(\mathbf{m}|\mathcal{M}, \mathbf{o})$$
$$\left(K_m + \log(\prod_i c_i \sigma_i^2) + \sum_i \frac{(o_i(\tau) - \mu_{m_i})^2}{2 c_i \sigma_i^2} \right).$$

Where $P_\tau(m|\mathcal{M}, \mathbf{o})$ is the probability of Gaussian m at time τ, given the alignment of the adaptation sequence $\mathbf{o}$ with model $\mathcal{M}$ and $\mathcal{L}(\mathbf{o}|\mathcal{M})$ is the likelihood of the observation sequence given the model. This results in a re-estimation equation:

$$c_i = \frac{\sum_m \sum_\tau P_\tau(m|\mathcal{M}, \mathbf{o}) \frac{(o_i(\tau) - \mu_{m_i})^2}{\sigma_i^2}}{\sum_m \sum_\tau P_\tau(m|\mathcal{M}, \mathbf{o})} \quad (5)$$

The variance adaptation above is guaranteed to increase the likelihood only when the means are held constant. Thus the means and the variances need to be adapted separately, with realignment taking place in between. This can either be done by alternating means and variance adaptation, or by adapting the means for several iterations and then adapting the variances. In this paper both techniques have been tried.

2.1. Multiple transforms

The transforms can be estimated on pools of Gaussians — from a single transform on the whole set to a separate transform for each Gaussian, which is equivalent to the standard re-estimation formulae. To decide which Gaussians to transform together, they are clustered together in a binary tree, built top-down. The pool of 750 Gaussians is iteratively bisected until each leaf represents a single Gaussian, the other nodes represent clusters of the Gaussians. With a particular adaptation dataset, the frames are aligned to the Gaussians in the pool, and the numbers of observations of each Gaussian, and of each cluster, are calculated. A threshold is chosen empirically, as the minimum number of observations required to estimate a transform.

Two methods of choosing which Gaussians should be transformed together have been tested, using the following criteria:

- Descend the tree until a leaf is reached, or either child has insufficient data to estimate a transform. Estimate a transform for each of the nodes so reached, and apply it to all the Gaussians in that node.
- For each Gaussian, ascend the tree until a node with enough data is reached. Apply the transformation for that node to the Gaussian.

The difference between the two methods can be seen in figure 1. If a threshold of 100 observations is used, using the first method, a transform will be estimated using all 200 observations under node **a**, and applied to all Gaussians in node **a**. Using the second method, there are enough observations under node **b**, so a transform is based on these 150 observations and applied to all the Gaussians under node **b**. A separate transform is estimated based on all the 200 observations under node **a**, but only applied to those Gaussians under node **c**. In practice the latter method has been found to perform better than the former, though in both cases the thresholds need to be chosen carefully to achieve the best performance, and the thresholds are different for the two methods.

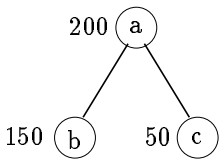

Figure 1: Part of a tree showing two nodes **b**, **c** and their common parent **a**, with the observation counts.

3. EXPERIMENTS

The experiments described below all use the same tied-mixture, writer independent hidden Markov model recognition system. Twelve writers with a variety of writing styles are tested. For each writer, up to 200 words of adaptation data are available, and around 500 words of test data are used to evaluate the system performance. Testing is carried out with a 23,000 word dictionary and less than one percent of the test set is out of vocabulary.

The initial performance of the writer-independent model varies from writer to writer. The average word error rate is 28.7%, but several writers have around 10% errors, and one has 60% errors. Experimental results are shown averaged across all the writers. For a given quantity of data, adaptation performance is found to vary depending on the actual training examples used, so the results presented here are averaged across five different samples for each experiment. Results are shown for adaptation sets containing 5 to 200 words, though the length and legibility of the words varies considerably.

3.1. Supervised vs. unsupervised

Most of the experiments have been conducted with correctly labelled adaptation data, but in a practical application it would be useful to be able to carry out adaptation when the only data available are not labelled. (Since labelled data can only be gathered by the labour-intensive operations of writing out stimuli as training data or correcting machine transcriptions). Since the labels of the data are unknown, they can only be inferred using the recognition system. To adapt in this *unsupervised* mode, the writer-independent recognizer is run on the training data and its answers are used as the labels for adaptation. The labelling is done either with a dictionary or with a character n-gram model. Though the word accuracy of the n-gram model is lower, the character accuracy (estimated using string alignment) is about the same as when using the dictionary, and allows better handling of training sets with many words not in the vocabulary.

4. RESULTS AND CONCLUSIONS

Figure 2 shows the effect of varying the number of iterations of alignment and adaptation. The adaptation improves the model so that successive iterations give better alignments, resulting in better adaptation. It can be seen that more iterations give better performance, with diminishing returns.

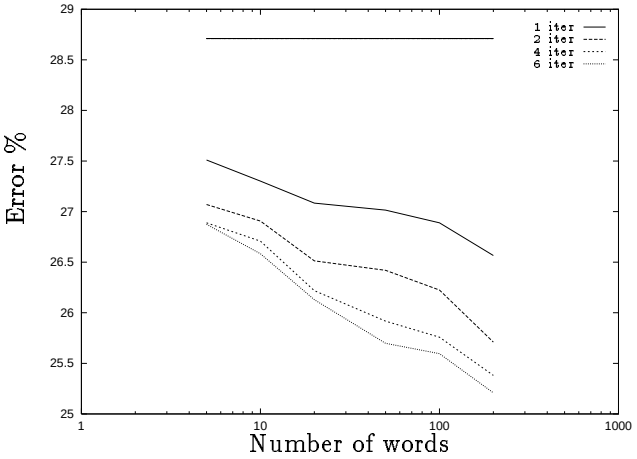

Figure 2: This graph shows the effect of adapting for one or more iterations and compares results with the error rate obtained when using the writer independent model without adaptation (the horizontal line).

Figure 3 shows the effect of adapting the variances as well as the distribution means. Adapting the variances outperforms means-only adaptation only for some writers, and when there is enough data available. For the adaptation set sizes examined here, it does not give an improvement on average, using either method.

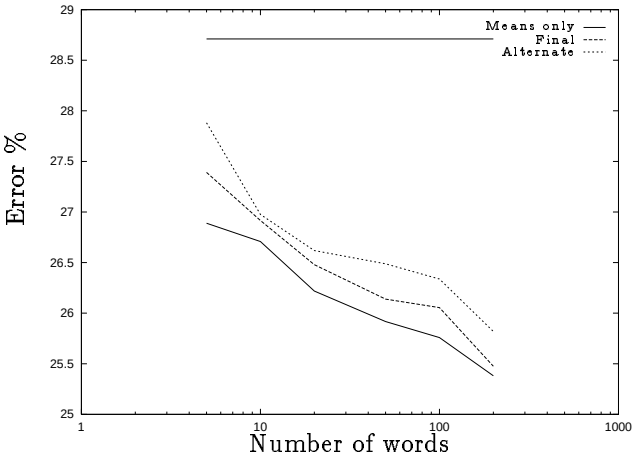

Figure 3: This graph shows the difference in performance when adapting means only (four iterations) or means and variances (a fifth iteration for adapting the variances only, or eight iterations alternating means and variances). The unadapted model performance is also shown.

Figure 4 shows three curves for adaptation with the

correct labels and with labels generated by the recognizer with and without a dictionary. Unsupervised adaptation is a little worse than supervised adaptation but still improves performance significantly Using a non-word model instead of a dictionary is found to increase the accuracy.

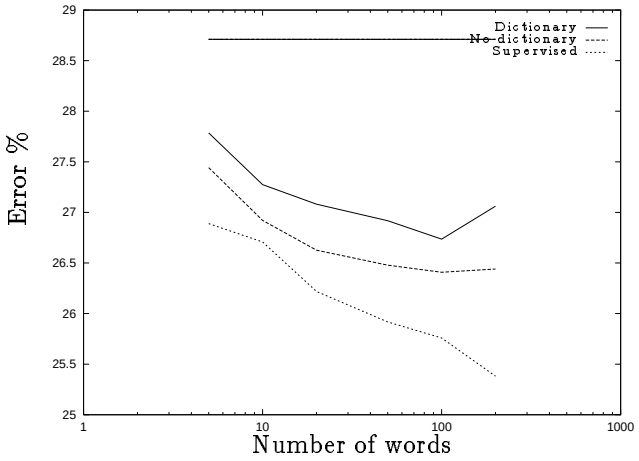

Figure 4: A comparison of errors obtained with supervised (using correctly labelled data) or unsupervised (data labelled using a recognizer with or without a dictionary) adaptation. The unadapted model performance is also shown.

5. CONCLUSIONS

It has been shown that the MLLR framework can be successfully applied to a tied-mixture HMM used for handwriting recognition. This adaptation technique gives significantly improved performance even when very small amounts of unlabelled training data are available and the performance increase becomes greater when more data are available or the data are well-labelled. A 9% reduction in error rate is achieved on only twenty words of training data (figure 2).

It is seen that multiple iterations of re-alignment and adaptation improve the performance. So far, adaptation of variances has given no further improvement in recognition accuracy with the data quantities being examined, though further adjustment of the thresholds for multiple transforms may yield such an improvement, and for larger quantities of data this is expected.

References

[1] M.J.F. Gales and P.C. Woodland. Variance compensation within the MLLR framework. Technical report, Cambridge University Engineering Department, 1996.

[2] C.J. Leggetter and P. C. Woodland. Speaker adaptation of HMMs using linear regression. Technical Report CUED/F-INFENG/TR 181, Cambridge University Engineering Department, June 1994.

[3] Krishna S. Nathan, Homayoon S. M. Beigi, Jayashree Subrahmonia, Gregory J. Clary, and Hiroshi Maruyama. On-line handwriting recognition using continuous parameter hidden Markov models. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 5, pages 121–4, 1993.

[4] Jayashree Subrahmonia, Krishna S. Nathan, and Michael P. Perrone. Writer dependent recognition of on-line unconstrained handwriting. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 6, pages 3479–3482, 1996.