Discriminative Training for HMM-Based Offline Handwritten Character Recognition

Roongroj Nopsuwanchai
Computer Laboratory
University of Cambridge
Cambridge, CB3 0FD, UK
rn225@cam.ac.uk

Dan Povey
Department of Engineering
University of Cambridge
Cambridge, CB2 1PZ, UK
dp10006@eng.cam.ac.uk

Abstract

In this paper we report the use of discriminative training and other techniques to improve performance in a HMM-based isolated handwritten character recognition system. The discriminative training is Maximum Mutual Information (MMI) training; we also improve results by using composite images which are the concatenation of the raw images, rotated and polar transformed versions of them; and we describe a technique called block-based Principal Component Analysis (PCA). For effective discriminative training we need to increase the size of our training database, which we do by eroding and dilating the images to give a three-fold increase in training data. Although these techniques are tested using isolated Thai characters, both MMI and block-based PCA are applicable to the more difficult task of cursive handwriting recognition.

1. Introduction

In this paper we report experiments on a HMM-based system for Thai character recognition. Although HMMs are not always the method of choice for isolated character recognition, they are generally used for cursive character recognition and the techniques we report here should be applicable in that case. We apply discriminative training using the Maximum Mutual Information (MMI) criterion, based on a previously described implementation with the use of lattices. Such an implementation is intended for continuous speech recognition and also applicable to cursive handwriting recognition. We describe a novel approach to PCA, which we call block-based PCA. This applies PCA to small overlapping sections of the vertical frame independently.

We also obtain considerable improvements by using "composite images," in which the image is concatenated with a rotated and polar transformed version of itself.

Section 2 explains the MMI criterion and describes the techniques used to optimise it; Section 3 describes how composite images are obtained from the raw images; Section 4 describes block-based PCA; Section 5 describes the training data and how it was obtained; Section 6 describes the baseline system and experimental conditions; Section 7 gives experimental results; and conclusions are presented in Section 8.

2. Maximum Mutual Information training of HMM parameters

In Maximum Likelihood (ML) training, we maximise the likelihood of the training data given the training transcriptions, i.e.

$$\mathcal{F}_{\mathrm{ML}}(\lambda) = \sum_{r=1}^{R} \log p_\lambda\left(\mathcal{O}_r | s_r\right), \tag{1}$$

where λ is the HMM parameters, $\mathcal{O}_r$ is the observed data for the r'th training file, and s_r is the correct transcription of the r'th file (just a single character in this case).

The Maximum Mutual Information (MMI) criterion is the posterior probability of the correct transcription, so:

$$\mathcal{F}_{\mathrm{MMI}}(\lambda) = \sum_{r=1}^{R} \log \frac{p_\lambda\left(\mathcal{O}_r | s_r\right)^\kappa P(s_r)^\kappa}{\sum_s p_\lambda\left(\mathcal{O}_r | s\right)^\kappa P(s)^\kappa}. \tag{2}$$

The summation $\sum_s$ in the denominator is a summation over all possible sentences (characters in this case), or in practice only the most likely ones. The scale κ, which will typically be less than one (e.g. in the range $\frac{1}{10}$ to $\frac{1}{20}$), is a scale on the log likelihoods which enables more sentences to compete with the correct sentence. $P(s)$ are the language model probabilities, which are not discriminatively trained. Experiments in large vocabulary speech recognition [5] have shown that scaling probabilities is essential for good test set performance. In experiments reported here we use κ=0.1.

Our implementation uses the Extended Baum-Welch (EB) formulae to optimise the HMM parameters, and represents the most likely competing sentences by "lattices" which in this case just include the 5 most likely characters derived from N-best recognition using a baseline HMM set. See [4] and [5] for a more complete description of the optimisation methods used to train the HMM parameters.

Discriminative training is more sensitive to the amount of training data than is ML training, as is clear from the results presented in [5] and in this paper. We augment our training data by eroding and dilating each training image to give a threefold increase in training data, and this improves discriminative training results. Dilation and erosion are two basic operators for image morphology which are used to respectively enlarge and erode away the boundaries of regions of foreground (black) pixels. We used a kernel with 3×3 pixels with origin in the center for the dilation operation and the kernel with 2×2 pixels with top-left origin for the erosion operation.

In [1], discriminative training (MCE in that case) was also applied successfully to character recognition. More improvement was obtained (33% relative difference for the best system reported there, relative to 25% in our case) but this is expected since there are about 1000 training examples per character, whereas in this case only 60 are available which are augmented to 180 by adding an eroded and dilated version of each image (although this is not the same as 180 independent examples).

3. Composite images

We obtain a large relative improvement of up to 33% depending on experimental conditions, by using a composite image. The image is concatenated with a 90° rotation of itself, and a polar transformed version of itself.

The polar transformation, similar to the log-polar transform [2] widely used in computer vision research, is a conformal mapping from points in image $f(x, y)$ to points in the polar image $g(r, \theta)$. We adapt this by defining an 'origin' $O = (o_x, o_y)$ given by the centroid $(o_x = \bar{x}, o_y = \bar{y})$ of the image. Defining d as the maximum distance between O and all pixels in f, the mapping is described by

$$r = \frac{\sqrt{(x - o_x)^2 + (y - o_y)^2}}{d} \quad (3)$$

$$\theta = \arctan\left(\frac{y - o_y}{x - o_x}\right) \quad (4)$$

The maps are then normalised to the size of 64×64 pixels. An example of a polar transformed image is given in Fig. 1(b) and a rotated image in Fig. 1(c). The original image and the two transformed images give similar recognition results individually, but when concatenated into a composite image as in Fig. 1(d) they give substantial improvements.

As can be seen from the results in Table 1, the composite image technique gives an improvement in recognition results ranging from about 11% to 33% depending on the feature vectors used.

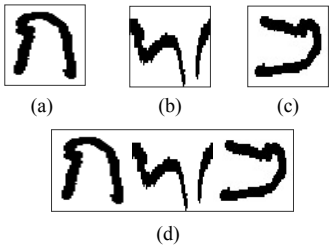


Figure 1. Samples of transformed character images: (a) size-normalised image, (b) polar transformed image, (c) 90° rotated image and (d) composite image.

4. Feature vectors and block-based PCA

In our experimental setup the raw images to be processed into features are size-normalised to give images of a constant height H (64 pixels in this case). The baseline method for processing these images into a sequence of feature vectors is to use each vertical column of pixels as the feature vector (Fig. 2(a)).

The simplest application of PCA is to project the first d principal components of the H-dimensional pixel vector into a d-dimensional vector, which is referred to as standard PCA technique. In [3], higher-resolution images were available so average blackness values for cells of pixels were input into their PCA rather than the single pixels.

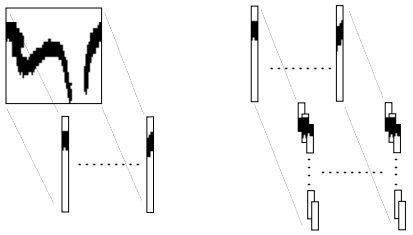
In block-based PCA we start with a block of the image, which will typically be a tall $w \times H$ block of pixels with a small width w (e.g. $1 \ldots 4$). This is divided vertically into overlapping sub-blocks of height h (e.g. 16), with successive sub-blocks moving by a vertical offset o_v (e.g. 8) until they cover the entire vertical height (Fig. 2(b)). For example, with sub-blocks of height 16 separated by an offset of 8 we would need 7 sub-blocks to cover a block of height 64.

PCA is then applied to each of the $w \times h$ sub-blocks to reduce its dimension to a small dimension d'. The vectors of size d' from the sub-blocks are concatenated to form a feature vector of size d. In our implementation each sub-block has its own matrix to extract the principal components, calculated from the covariance of that sub-block in the training data.

In all experiments reported here, the successive feature vectors are obtained by shifting the horizontal position by 1 each time and repeating the analysis. For generality, we can

express this as a horizontal offset o_h, with $o_h = 1$ in this case.

To summarise: a $w \times H$ block of the image is broken to $w \times h$ subblocks separated by a vertical offset o_v, and each sub-block is projected by PCA to d' dimensions. These are concatenated for all the $1 + \frac{H-h}{o_v}$ subblocks to make a final vector of size d.

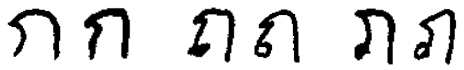


(a) (b)

Figure 2. Difference between (a) vertical columns of pixels where normal PCA was applied, and (b) sub-blocks of the vertical columns where block-based PCA was applied.

5. Data and recognition task

The Thai written language consists of isolated characters that cannot be written as a cursive script, and Thai characters are difficult to recognise as there are 77 of them including tone marks and numbers, with some characters looking quite similar (Fig. 3).



(a) (b) (c)

Figure 3. Samples of three different Thai characters with unconstrained writing style: Character (a) *gor-gai*, (b) *tor-thung* and (c) *por-sam-pao*.

At present there is no publicly available database of Thai handwriting. We established a database of Thai characters with unconstrained writing style collected from twenty native writers who were instructed to write characters in a specially prepared form. Writers were instructed to write in an unconstrained style, resulting in a wide variety of styles (see Figure 4).

The database has 120 samples from each character for each of the 20 writers. Currently it is divided into training and testing databases of equal size, by including half

Figure 4. Example of handwritten characters.

the samples from each writer in the training database and half in the testing database. The character images are extracted from the boxes in the form by Connected Component Analysis (with manual intervention for where the characters overlap the margins of the boxes), and the aspect ratios and sizes of the resulting images are normalised to give bi-level images of 64×64 pixels.

64 of the 77 characters are so-called "baseline" characters (analogous to letters and numbers) and 13 are "non-baseline" characters which appear above and below characters (these are tone marks, above and below vowels).

6. Experimental conditions

HMMs are trained and tested using the HTK toolkit (a toolkit primarily used for speech recognition), with additional software which we use to prepare the feature vector sequences from the images and to train models using MMI.

The testing setup is based on the notion that one would have prior information (from the positions of the characters) which characters are baseline characters and which appear above and below the baseline characters. The 63 baseline characters are recognised with a single word net with equal "language model" probabilities, and the 11 non-baseline characters have their own word net again with equal probabilities.

The HMMs use a left-to-right topology, in which each state has a transition to itself and the next state. HMMs for each character have 50 states, which has been found to be approximately the optimal number of states for the raw image and close to the optimal number for the composite image. HMM states have diagonal Gaussian probability density functions (pdfs), with no parameter tying.

For MMI training we use the probability scale κ=0.1, and the constant E which controls speed of optimisation in the Extended Baum-Welch (EB) equations is set to 1 (see [5] or [4] for an explanation of the setup for optimisation). The lattices required to encode the most likely competing hy-

potheses are generated from the output of N-best recognition using the 5 most likely outputs from recognition of the training file; training using these lattices this gives almost identical recognition results to training using a lattice consisting of all characters. The N-best recognition is repeated on each iteration of training. MMI training starts from an ML-trained HMM. Results are given after 10 iterations of EB optimisation.

7. Experimental results

7.1 Feature extraction and image concatenation

Initially, experiments were performed using ML estimation only to investigate feature extraction methods. Experiments are performed using both raw images, and composite images obtained as described in Section 3.

Feature	Block width (w)	Feature vector dimension (d)	Recognition rates (%)	
			Raw Image	Composite Image
Pixels	1	64	85.63	87.22
	4	256	85.39	87.81
PCA	1	42	81.01	86.92
	4	42	81.61	87.77
Block-based PCA (h=16, o_v=8)	1	42	85.47	89.36
	4	42	84.92	88.23

Table 1. Recognition results on different features with ML trained system.

Results are given in Table 1. In all cases composite images give better results than raw images, the relative improvement varying between 11% and 33%. In addition block-based PCA consistently gives better results than standard PCA. Standard PCA baselines are also given for a block with a width of 4, for comparability with block-based PCA.

Comparing block-based with standard PCA, for $w = 1$ there is an improvement of 23% or 19% with the raw and composite images respectively, and with $w = 4$ an improvement of 19% or 3.8%. The PCA baseline for $w = 4$ consists of doing PCA on all of the pixel values in the 64×4 block.

There is a difference between the raw and composite images in the relative advantage of using raw pixels or PCA. PCA usually gives a degradation for the raw image but an improvement for the composite image. This is probably related to the fact that we used a constant number of HMM states for both types of image, while the composite image is 3 times more complex. PCA may be more suitable where there are relatively few states relative to the complexity of the image. Although these results do not find a consistent benefit from PCA relative to pixels, block-based PCA does give better results than PCA in all cases.

7.2 MMI training

Feature	Number of training samples / character	Recognition rates (%)	
		ML	MMI
Block-based PCA (w = 4, d = 42)	60	88.23	90.35
	180	88.54	91.38

Table 2. Recognition results on testing set of MMI training.

Table 2 gives ML and MMI test set results for the block-based PCA features. Results are given both for the basic training set (60 samples per character) and the augmented training set (180 samples per character) obtained by eroding and dilating each training image. The extra data only slightly improves ML results (2.6% relative) but gives a considerable improvement for MMI (11% relative). Thus, augmenting the training data makes much more difference to the discriminatively trained result than to the ML-trained result. The improvement from MMI compared with the ML baseline is 18% relative with the original data and 25% relative with the augmented data. The improvement gained from MMI is predicted to be even greater than 25% for a system with a more typical quantity of training data.

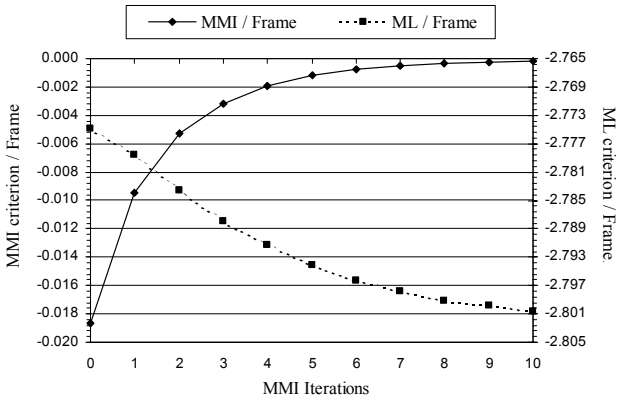


Figure 5. MMI and ML criterion per frame versus the number of iterations of the experiment in Table 2 (180 training samples).

Figure 5 shows the MMI and ML criteria varying with iteration of training, and demonstrates that MMI optimisation is proceeding smoothly.

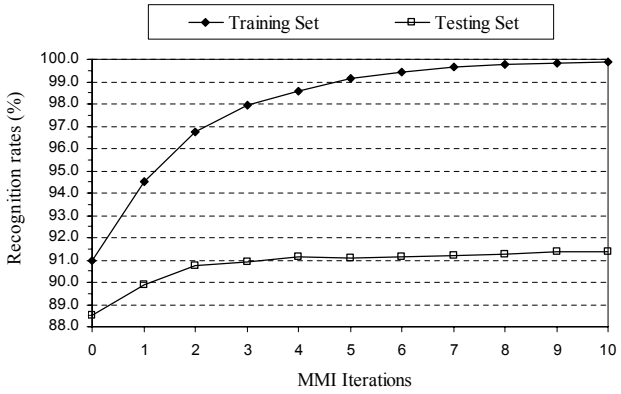


Figure 6. Recognition rates on training and testing sets versus the number of MMI iterations of the experiment in Table 2 (180 training samples).

Figure 6 shows recognition results varying with training iteration. As expected, the training set results rise quickly to nearly 100% correct recognition while the test set shows a more modest improvement.

Results in Table 2 are only given for the block-based PCA features with $w = 4$. MMI training was not as successful with all of the other features. With pixel-based features the EB training algorithm failed to converge and training and test accuracy fell, probably because our optimisation approach is not well suited to work with binary pixels (indeed, continuous Gaussian HMMs would not be expected to work well with such features). Block-based PCA using single columns of pixels ($w = 1$) was the best ML result (89.36% for the composite image, as shown in Table 1), but the testing accuracy of the MMI-trained system trained using the non-augmented training set peaked on the 3rd iteration of training at 90.23% (only 8.9% relative improvement), and dropped after that. Training set accuracy improved faster as MMI training proceeded than for the wider block size, which may indicate that the narrower-based features are more susceptible to overtraining. This problem may be due to the very small size of our training set, but in any case it still holds true that block-based PCA always outperforms standard PCA and that MMI always gives at least some improvement for non-binary features.

8. Conclusions

We have demonstrated improvements over a basic HMM-based isolated character recognition system by applying three techniques: MMI, composite images and block-based PCA.

MMI is a technique used for speech recognition and we have demonstrated its utility for written character recognition. This implementation of MMI can also be used for cursive handwriting recognition which is analogous to continuous speech recognition. We have improved results by about 25% (relative) compared with ML, using training data augmented by erosion and dilation of images.

Composite images are a concatenation of an image with a 90° rotated image and a polar representation of the image. These improve results by 11% to 33% (relative) depending on which other techniques are used.

Block-based PCA is an alternative way to apply PCA, applying to small vertical sections separately rather than a single large vertical section. This improves results by between 4% and 25% (relative) compared with standard PCA.

Acknowledgements

Mr. Nopsuwanchai is partially funded for his PhD study by the Cambridge Thai Foundation and the Cambridge Overseas Trust. The authors would like to thank Thai students from the University of Cambridge who contributed their handwriting samples.

References

[1] A. Biem. Minimum classification error training of hidden markov model for handwriting recognition. In *Proc. ICASSP'01*, volume 3, pages 1529–1532, 2001.

[2] M. Tistarelli and G. Sandini. On the advantage of polar and log-polar mapping for direct estimation of time-to-impact from optical flow. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 15(4):401–410, 1993.

[3] A. Vinciarelli and S. Bengio. Offline cursive word recognition using continuous density hmms trained with pca or ica features. In *Proc. ICPR'02*, volume 3, pages 81–84, 2002.

[4] P. Woodland and D. Povey. Large scale discriminative training for speech recognition. In *Proc. ISCA ITRW ASR2000*, pages 7–16, 2002.

[5] P. Woodland and D. Povey. Large scale discriminative training of hidden markov models for speech recognition. *Computer Speech and Language*, 16(1):401–410, 2002.