In proceedings of the 31st Asilomar conference on Signals, Systems and Computers, 1997, pp 306–310.

A Hidden Markov Model Fingerprint Classifier

Andrew Senior

IBM T.J.Watson Research Center,
P.O. Box 704, Yorktown Heights,
NY 10598-0218, USA.
E-mail: `aws@watson.ibm.com`

Abstract

Fingerprint classification is an important indexing method for any fingerprint database or recognition system. Fingerprints are classified based on overall characteristics. This paper describes a novel method of classification using hidden Markov models to recognize the ridge structure of the print. The paper also describes a method for achieving any level of accuracy required by the system, by sacrificing the efficiency of the classifier. Results are presented on a NIST fingerprint database.

1 Introduction

The classification of fingerprints has long been an important part of any fingerprinting system. A partition of fingerprints into groups of broadly similar patterns allows filing and retrieval of large databases of fingerprints for quick reference. Currently interest in fingerprint classification is stimulated by its use in automatic fingerprint identification systems (AFIS). Classification is used in an AFIS to reduce the size of the search space to fingerprints of the same class before attempting identification. The most widely-used system of fingerprint classification is the Henry system and its variants [2]. Examples from the five of the main classes of the Henry system are shown in figure 1.

This paper describes a novel, hidden Markov model approach to fingerprint classification. The system described has been designed to operate on both rolled and 'dab' fingerprints, where some of the structural information used by other systems (such as the delta position) may not be available in the fingerprint image. The system described has been tested on the NIST-4 [6] database of fingerprint images and results are presented. Further, a method of measuring the efficiency of a classification algorithm is described, along with a method for achieving arbitrary accuracy by trading off classifier efficiency which will enable an imperfect classifier to be used in a real-world system.

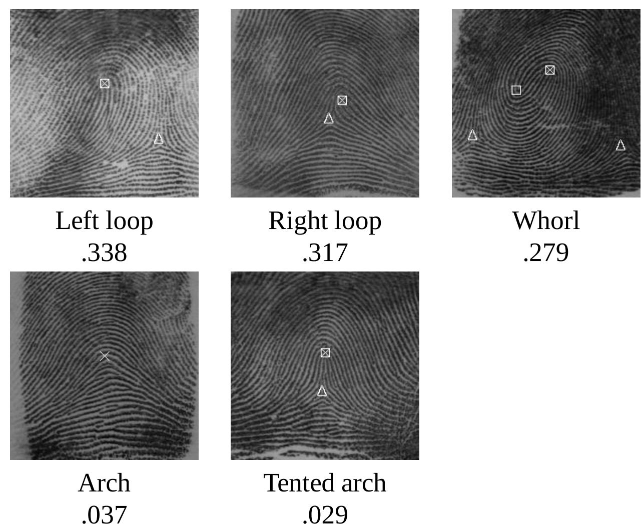

Figure 1. Five fingerprint categories, with their frequencies of occurrence.

2 Feature extraction

The system deals with fingerprint images stored as arrays of grey levels and obtained with a scanner or camera device — either from an inked fingerprint on paper, or as a 'live-scan' directly from the finger. For much of the work in this paper, the NIST-4 [6] database of rolled fingerprint images has been used, since this provides a large number (4000) of fingerprints with associated class labels.

The features provided to the recognizer are based on the characteristics of the intersections of ridges with a set of fiducial lines that are lain across the fingerprint image. To find the ridge locations, a number of image processing techniques are used. These have previously been described elsewhere [4] and are not detailed here. The basic steps are:

1. smoothing;
2. finding the predominant direction in each of an array of blocks covering the image;
3. segmenting the image into the area of the print (*foreground*) and the unwanted background, based on the strength of directionality found in each block;
4. applying directional filters to highlight the ridges and detect pixels that are parts of ridges;
5. thinning the ridge image so that each ridge is left represented by an eight-connected, one-pixel-wide line termed the *skeleton*.

Given the skeleton image of the ridges, parallel fiducial lines are lain across the image and each one followed in turn. For each intersection of a fiducial line with a ridge, a

Figure 2. A sample fingerprint showing horizontal fiducial lines.

feature is generated. Each feature consists of a number of measurements:

1. the distance since the last intersection;
2. the angle of intersection;
3. the change in angle since the last intersection;
4. the curvature of the ridge at the intersection.

The angle features (2) can be seen to contain similar information to the direction field calculated in the preprocessing stages of this system and used by other systems as the feature set for classification [1]. However, this representation allows a higher resolution representation of the fingerprints, and allows more information to be represented (*e.g.* ridge spacing and curvature). Further measurements could also be taken at each point.

The measurements of each feature are termed a frame and the frames, R_{i_k} for the ith fiducial line are collectively termed a 'row', R_i, whose ordering is preserved. For each orientation ϕ of fiducial lines, a separate representation $\mathcal{R}^\phi = \{R_i, \forall i\}$ of the print is obtained. Typically, in this research, only horizontal and vertical lines have been used, but other angles may allow further information to be captured.

3 Hidden Markov models

Hidden Markov models are a form of stochastic finite state automaton well suited to pattern recognition and successfully applied to speech recognition [3, 8]. They are appropriate to the problem posed here because of their ability to classify patterns based on a large quantity of features, whose number is unknown and which have certain types of underlying structure, especially if that structure results in stationarity of the feature distributions over some spatial or temporal period. (This is found in fingerprints, where ridge orientations, spacings and curvatures are for the most part only slowly varying.)

Typically HMMs are one-dimensional structures suitable for analyzing temporal data. Here, the data are two dimensional, but the process of feature extraction can also be described as a one dimensional array of one-dimensional processes. Each row is a one-dimensional process, and the ensemble of rows is itself a one-dimensional process. Thus we can apply a 'two dimensional hidden Markov model' which consists of a nesting of row models within whole-print models as shown in figure 3.

3.1 Row modelling

To simplify the analysis of the model, first consider a row model modelling a single row of fingerprint data. Each row model M_j, consists of a number of states, which model the small, stationary regions in a row. Any row R_i, is assumed to have been generated by the row automaton transiting from state to state, producing the frames in the observed order at each transition, with the kth state being S_{ij_k}. The state transitions are controlled by probabilities $P(S_{ij_k}|S_{ij_{k-1}})$ trained with certain constraints: the state must monotonically increase $S_{ij_k} > S_{ij_{k'}}$ for $k > k'$ and it may be possible to skip states at the edge. because of the nature of the printing process whereby, especially for dabs, it is to be expected that edge regions of the fingerprint will be missing but the central regions will always be present, only states at the edge of the print may be skipped. This effectively constrains the initial state distribution $P(S_{ij_0})$.

The frames are modelled with mixtures of diagonal covariance, multivariate Gaussian distributions. Thus for any frame R_{i_k}, it is possible to calculate the likelihood

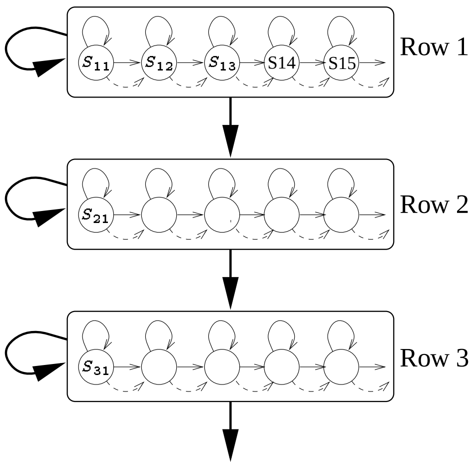

Figure 3. A schematic of the two-dimensional structure of the HMM, showing three row models of five states each forming a global model.

$P(R_{i_k}|S_{ij_k})$ of it occurring in any state S_{ij_k}. With these likelihoods, for any row model, the likelihood of any row can be calculated as a product of frame likelihoods and transition probabilities for an alignment between the features and states:

$$P(R_i|M_j) = \max_{S_{ij}} P(R_{i_0}|S_{ij_0})P(S_{ij_0}) \prod_k P(R_{i_k}|S_{ij_k})P(S_{ij_k}|S_{ij_{k-1}}) \qquad (1)$$

The models are initialized by using an equal-length alignment with the frames evenly distributed across the states of the model. After estimating the initial parameter values, Viterbi alignment is used to find the maximum likelihood alignment of frames with states, which is used for retraining.

3.2 Global model

The global model is the same as a row model, except that its states are row models, and its frames are whole rows. Thus for each model c:

$$P(\mathcal{R}|\mathrm{Model}_c) = \max_{S'} P(R_0|S'_0)P(S'_0) \prod_k P(R_k|S'_k)P(S'_k|S'_{k-1}), \qquad (2)$$

and S' is an alignment specifying which rows of data are aligned with each row model.

3.3 Postprocessing

For each orientation of fiducial lines, a separate classifier can be made. Since the errors of the different classifiers will be different, a combination of their scores may yield a better accuracy. So far little effort has been expended on investigating this combination of classifiers, and only simple, equal-weight mixing of the classifiers for horizontal and vertical features ($\mathcal{R}^h$ and $\mathcal{R}^v$) has been tested. Denoting by Model^h_c, Model^v_c the class c models trained with vertical and horizontal features respectively:

$$P(\mathcal{R}^h, \mathcal{R}^v|\mathrm{C}lass_c) \approx P(\mathcal{R}^h|\mathrm{Model}^h_c)P(\mathcal{R}^v|\mathrm{Model}^v_c) \qquad (3)$$

Because the classes that are used are not equal in frequency of occurrence, calculating the posterior probability of a class given the data requires the product of the data likelihood given the class and the prior probability of the class:

$$P(\mathrm{C}lass_c|\mathcal{R}^h, \mathcal{R}^v) \propto P(\mathcal{R}^h, \mathcal{R}^v|\mathrm{C}lass_c)P(\mathrm{C}lass_c) \qquad (4)$$

The class priors have been estimated by Wilson *et al.* [7] on 222 million fingerprints. (The proportions are .037, .338, .317, .029, .279 for arch, left loop, right loop, tented arch and whorl respectively). Further, since the NIST-4 database (and likewise the test set used here) has equal numbers of prints from each class, to obtain a good estimate of the true test-condition accuracy, the results can be weighted according to the true frequencies of occurrence. Otherwise, a classifier good at recognizing arches, which are rare, would appear better on this test set than on a representative test set where this ability is rarely called upon. Weighting the answers gives better estimates of the error rates than simply throwing away test data to conform to the observed frequencies [7].

Some of the experimental results combine the hidden Markov model classifiers with a decision tree classifier [5].

4 Classifier efficiency

Since the purpose of a fingerprint classifier is to partition large fingerprint databases, in addition to the classification accuracy — the proportion of classifications that give the correct class — the classification efficiency must also be considered. The classification efficiency can be considered as a measure of reduction of search space. In practice, the proportion of the database to be searched will vary with each query, so over a test set, the average efficiency can be calculated as:

$$\frac{\text{Number of matches required with no classifier}}{\text{Number of matches required when classifier is used}}, \qquad (5)$$

where an exhaustive 1:many match against a database of N prints is counted as N matches.

If a perfect classifier is used to classify M prints before matching against a database of N prints, any of the MP_i test prints in class i (which occurs with probability P_i) need only be tested against the NP_i database prints of class i. Thus the total number of matches required is now $\sum_i NMP_i^2$ instead of NM. The efficiency of a perfect classifier using these classes is thus $\frac{1}{\sum_i P_i^2}$. Using the five NIST-4 classes, this gives an efficiency of 3.39. Merging arch and tented arch classes only reduces this to 3.37, since this distinction so rarely needs to be made. As can be seen the imbalance of the class priors makes the efficiency significantly lower than would be obtained with five equally frequent classes (*i.e.* 5).

This efficiency measure permits the evaluation of rejection and backing-off strategies. It is clear that accuracy can be traded off for efficiency – searching more than just the top one class will give higher accuracy but lower efficiency.

Previous classification works have quoted error rates that would be unacceptable in real-world systems. If a reliable measure of confidence in the classifier's answer is available, it is possible to devise methods to adjust the reliance on the classifier answer, when that classification is uncertain, and thus reducing the number of errors made. Some classifiers [1] have used a rejection mechanism, which improves the accuracy at the cost of not pruning the search with those prints that are rejected.

4.1 Backing-off

This section proposes a more complex scheme to allow graceful and efficient '*backing-off*' of classifier reliance based on a likelihood ratio confidence measure. It is clear that if the likelihoods for the top two classes are very different, then the classifier can be considered to be more 'confident' in its answer than if the two likelihoods are similar (when it would only take a small perturbation to change the ranks of the answers). Thus, the likelihood ratio of the top two answers is examined and if less than an empirically determined threshold, then the top choice is deemed to be not confident and the top two classes are returned as the answer (increasing the proportion of the database subsequently searched by the 1:many matcher). Similarly, the likelihood ratio of the second and third choices is compared to a threshold (in this work, a single threshold is used for all comparisons) to allow backing off to three classes. Repeating the procedure, if all the likelihoods are similar, the classifier will return a "don't know" answer, and all classes must be searched. This is equivalent to the simple rejection option allowed by some classifiers.

The efficiency of the classifier when allowing backing-off is now:

$$\frac{MN}{\sum_{m=1}^{M} \pi_m N} \tag{6}$$

where π_m is the proportion of the database searched for query print m, and $\pi_m \geq P(C_i)$ if print m is in class i.

Adjusting the likelihood ratio threshold allows arbitrary accuracy to be obtained. A large threshold would give 100% accuracy but efficiency of only one. A threshold of zero would give the 'raw' top-one accuracy and maximum efficiency (3.37 for the 4 class problem). Adjusting the threshold on a separate cross-validation set allows us to set the overall classifier accuracy to that deemed necessary for the whole system.

5 Results

The system has been trained and tested on the NIST-4 database of fingerprint images. The test set was a random sample of 542 prints from the 4000 available. The class labels given by the database were used, but since the efficiency is hardly affected, the classifier was only trained to distinguish four classes, treating arch and tented arch as identical.

Experiment	Error (%)
Vertical (v) features only	23.2
v & h classifiers	18.4
v & h with priors and weighting	11.8
ditto with decision tree	10.0

Table 1. Error rates, testing a 2DHMM classifier. The classifier used here has 6 rows of 9 states each and uses a shared pool of 80 Gaussians with 4 dimensional features.

A model with more parameters (8 states, 8 rows and a pool of 200 Gaussians) achieves an error rate of 8.4% on the same test set, but takes more time for train and for classification. The PCASYS [1] classifier was tested on the same test set and gave a 9.9% error rate, though it had been trained on a different training set, with more but perhaps less well-matched fingerprint images.

Experiment	Error (%)	Efficiency
v & h, priors and weighting	0.6	2.0
Ditto with decision tree	0.8	2.3

Table 2. Efficiencies for the 6/9/80 models with rejection thresholds set to give < 1% error rate.

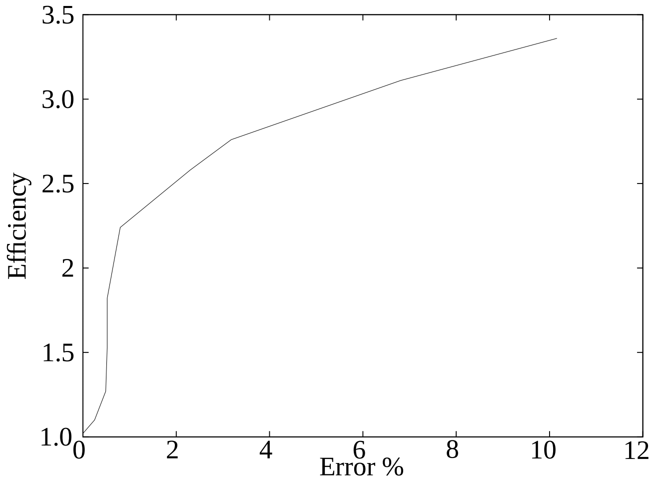

Figure 4. A graph of efficiency against error rate, testing with a variety of thresholds.

6 Conclusions

This paper has proposed a new method for fingerprint classification that does not rely on core and delta information, and has been designed to work on both dabs and rolled prints. Existing fingerprint classifier accuracies fall short of what is required to make a significant contribution to an AFI system. This paper has proposed a method for comparing the efficiencies of different classification schemes and describes a system for achieving an arbitrary degree of accuracy from a classification system while evaluating the effect of the trade off. By this means, current fingerprint classifiers can be rendered of use in an AFI system. The new classification method can achieve filtering of a factor of 2.3, with an error rate of only 0.8%.

References

[1] G. T. Candela, P. J. Grother, C. I. Watson, R. A. Wilkinson, and C. L. Wilson. PCASYS — a pattern-level classification automation system for fingerprints. Technical Report NISTIR 5647, NIST, April 1995.

[2] Federal Bureau of Investigations. *The Science of Fingerprints (Classification and Uses)*. US Department of Justice, Superintendant of Documents, U.S. Government Printing office, Washington D.C. 20402, 12–84 edition, 1984.

[3] L. R. Rabiner and B. H. Juang. An introduction to hidden Markov models. *IEEE ASSP magazine*, 3(1):4–16, January 1986.

[4] N. K. Ratha, K. Karu, S. Chen, and A. K. Jain. A real-time matching system for large fingerprint databases. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(8):799–813, August 1996.

[5] A. Senior. A decision tree fingerprint classifier. in preparation.

[6] C. Watson and C. Wilson. NIST special database 4: Fingerprint database. Technical report, National Institute of Standards and Technology, March 1992.

[7] C. L. Wilson, G. T. Candela, and W. C. I. Neural network fingerprint classification. *Journal for Artificial Neural Networks*, 1(2):203–228, 1994.

[8] P. C. Woodland, J. J. Odell, V. V. Valtchev, and S. J. Young. Large vocabulary continuous speech recognition using HTK. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 2, pages 125–128, Apr. 1994.