ELSEVIER

Pattern Recognition Letters 22 (2001) 209–214

Pattern Recognition Letters

www.elsevier.nl/locate/patrec

Unconstrained Farsi handwritten word recognition using fuzzy vector quantization and hidden Markov models

M. Dehghan [a,b], K. Faez [a], M. Ahmadi [b,*], M. Shridhar [c]

[a] *Electrical Engineering Department, Amirkabir University of Technology, Tehran, Iran*

[b] *Electrical and Computer Engineering Department, University of Windsor, Windsor, Ont., Canada N9C 3P4*

[c] *Electrical and Computer Engineering Department, University of Michigan, Dearborn, MI 48128, USA*

Received 13 January 2000; received in revised form 29 August 2000

Abstract

An unconstrained Farsi handwritten word recognition system based on fuzzy vector quantization (FVQ) and hidden Markov model (HMM) for reading city names in postal addresses is presented. Preprocessing techniques including binarization, noise removal, slope correction and baseline estimation are described. Each word image is represented by its contour information. The histogram of chain code slopes of the image strips (frames), scanned from right to left by a sliding window, is used as feature vectors. Fuzzy c-means (FCM) clustering is used for generating a fuzzy codebook. A separate HMM is trained by modified Baum–Welch algorithm for each city name. A test image is recognized by finding the best match (likelihood) between the image and all of the HMM word models using forward algorithm. Experimental results show the advantages of using FVQ/HMM recognizer engine instead of conventional discrete HMMs. © 2001 Elsevier Science B.V. All rights reserved.

Keywords: Farsi handwritten word; Fuzzy vector quantization; Hidden Markov model

1. Introduction

Several approaches have been described for handwritten word recognition (Chen et al., 1995; Guillevic and Suen, 1998; Kim and Govindaraju, 1997). One of the most promising methods is based on hidden Markov models (HMM). Both discrete and continuous HMMs have been successfully applied to handwritten word recognition applications (Chen et al., 1995; Guillevic and Suen, 1998). However, discrete HMM is more attractive because of its low computation cost (Rabiner, 1989). Nevertheless, discrete HMM inherently suffers from some problems as outlined below (Rabiner, 1989; Tsuboka and Nakahashi, 1995):

- Quantization error caused by quantizing of input vectors into a limited set of cluster centers. Assigning an input vector to the closest cluster according to the Euclidean distance causes information loss that in turn leads to degradation of recognition performance.
- Lack of sufficient training data causes poor estimation of HMM parameters, especially the observation symbol probabilities. Although a parameter smoothing algorithm (Rabiner, 1989) can reduce this problem.
- Finally, calculation of symbol probability in a given state is based on the frequency of

* Corresponding author. Fax: +1-519-971-3695.
E-mail address: ahmadi@uwindsor.ca (M. Ahmadi).

0167-8655/01/$ - see front matter
PII: S0167-8655(00)00090-8

occurrence of symbols (cluster centers) in the observation sequences. Since it is normally unlikely that a symbol will reoccur exactly in the same form in the recognition phase, a loss of performance will result. It would be required to take into account all of the clusters with different membership values associated to them based on similarity to the observation symbol.

All of these problems can be reduced by using a fuzzy vector quantization (FVQ) instead of the conventional crisp VQ. HMM based on FVQ (FVQ/HMM) was proposed by Tseng et al. (1987) as an extension to the conventional discrete HMM. Then, Tsuboka and Nakahashi (1995) have resolved a mathematical inconsistency which existed in the FVQ/HMM method by introducing two new types (multiplication and addition type) of FVQ/HMMs. In this paper, the multiplication type FVQ/HMM is used for unconstrained Farsi handwritten word recognition system.

2. System description

The proposed system is designed for reading the city names from address field. This application falls within the limited lexicon category. The lexicon size is limited (198 city names) or can be pruned by using additional information like Zip codes. The block diagram of the system is illustrated in Fig. 1. It is necessary to mention that Farsi (which is very similar to Arabic) script is inherently cursive both in handwritten and printed forms and is written horizontally from right to left. The reader is referred to Amin (1998) for further details on Farsi/Arabic script characteristics and also the state of the art of Arabic character recognition.

2.1. Preprocessing

The images of city names were extracted from postal envelopes scanned with 300-dpi resolution and 256 gray levels. These were then binarized. From the horizontal projection histogram of word image and rotated images of the word by $\pm\theta$ (5°), the base line was estimated and then, the image was de-skewed based on the estimated slope of the baseline (as shown in Fig. 2(a)). The stroke width of the word image was estimated using the method proposed in (Kim and Govindaraju, 1997).

2.2. Contour representation

A variant of stroke width compensation algorithm proposed by Hu et al. (1996) is applied to the word image to make the image stroke width at least three pixels wide in order to ensure proper contour extraction. Then, the image is enlarged with some blank rows so that the baseline of word is located in the middle of the image. This increases the discriminative power of feature extraction process for the upper and lower parts of the words. Later, the chain code of the image contour is derived and the coordinates and chain code direction of each border pixel are stored in a proper data structure.

2.3. Frame generation

In this phase, the word image is converted to an appropriate sequential form suitable for HMM recognition engine. The area of the image, obtained from contour representation phase, is first divided into a set of vertical fixed-width frames (strips) from right to left (Fig. 2(b)). The width of a frame is set to twice the average stroke width of the word image extracted in the preprocessing phase, and there is a 50% overlap between two consecutive frames.

2.4. Feature extraction

For feature extraction, each frame is divided into five equal-sized zones as shown in Fig. 2. In each zone, a local histogram of the contour chain codes is calculated. The feature vector is composed of these local histograms. Since the contour direction is quantized to one of four possible values (0°, 45°, 90°, and 135° with respect to horizontal), a histogram in each zone has four components. Therefore, each frame is represented as a 20-dimensional feature vector. In order to reduce the sensitivity of features to stroke width distortion, each component in feature vector is normalized to the height of the zone. To reduce the sensitivity of

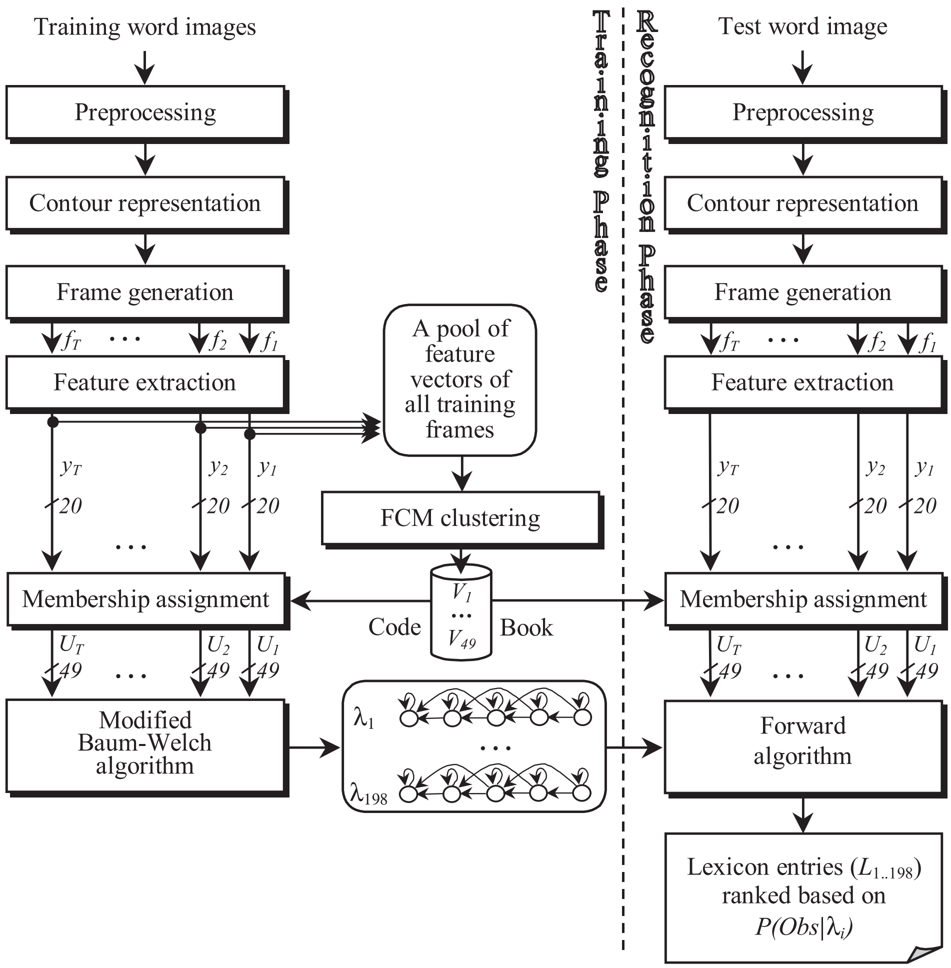

Fig. 1. The block diagram of proposed system.

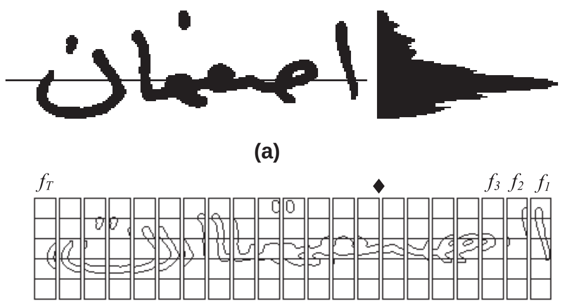

Fig. 2. An example image during processing steps. (a) An image after preprocessing and baseline estimation. (b) The image after contour representation, frame generation.

the features to the errors in baseline estimation, the border zones are fuzzified. In this way, zones are overlapped by 20% and the contour pixels in the overlapping regions have different contributions on the calculation of chain code histogram based on the membership function shown in Fig. 3.

3. Fuzzy vector quantization

A set of feature vectors extracted ($\boldsymbol{Y} = \{y_1, y_2, \ldots, y_n\}$) from more than 400,000 word strips (frames) were gathered to generate a fuzzy code

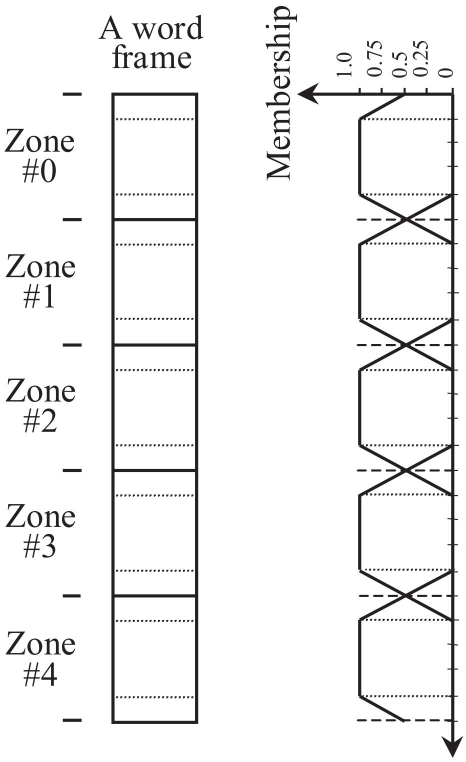

Fig. 3. The membership function used in chain-code histogram feature extraction.

book. The Fuzzy c-means (FCM) algorithm (Bezdek et al., 1984) is used to segment this vector space into c fuzzy partitions represented by a set of cluster centers ($\boldsymbol{V} = \{V_1, V_2, \ldots, V_c\}$) with the following constraints:

$$\sum_{i=1}^{c} u_{ik} = 1, \quad u_{ik} \in [0,1], \quad 0 < \sum_{k=1}^{n} u_{ik} < n; \quad 1 \leqslant k \leqslant n, \; 1 \leqslant i \leqslant c, \tag{1}$$

where u_{ik} is defined as the membership degree of the kth feature vector in the ith partition. After generating the fuzzy code book, a given feature vector ($\boldsymbol{y}_i$) is mapped into a membership vector $\boldsymbol{U}_i = [u_{i1}, u_{i2}, \ldots, u_{ic}]$. Thus, each word image, represented by a sequence of feature vectors extracted from word frames, is now mapped into an observation sequence of membership vectors (instead of an observation sequence of single values in the case of conventional VQ/HMM). Therefore, the forward algorithm and Baum–Welch re-estimation algorithm (Rabiner, 1989) must be modified to take into account this fuzzy observation sequences.

4. Hidden Markov model

In this method, each word class (city name) is modeled by a single right–left multiplication-type FVQ/HMM (Tsuboka and Nakahashi, 1995). All necessary notations and the modified version of forward and Baum–Welch algorithms are described in detail as below (Rabiner, 1989; Tsuboka and Nakahashi, 1995):

- The set of K observation sequences (training samples) for each word class:

  $$Y = \left\{Y^{(k)}\right\}, \quad 1 \leqslant k \leqslant K, \tag{2}$$

  where

  $$Y^{(k)} = \left\{y_t^{(k)}\right\}, \quad 1 \leqslant t \leqslant T_k \tag{3}$$

  and $y_t^{(k)}$ is the observation vector at frame t in the kth training sample and T_k is the number of frames in the kth training sample.
- The number of states (N) which is set for each class, is proportional to the average numbers of frames in training samples in that class:

  $$N = \eta\left(\frac{1}{K}\sum_{k=1}^{K} T_k\right), \tag{4}$$

  where η is a constant and chosen to be equal to 0.66. We denote the states as $\boldsymbol{S} = \{S_1, S_2, \ldots, S_N\}$ and the state at time t as q_t.
- The number of distinct observation symbols per state (M), which is set equal to 49 in this case. We denote the individual symbols as

  $$V = \{V_m\}, \quad 1 \leqslant m \leqslant M \tag{5}$$

  which are fuzzy cluster centers obtained by fuzzy clustering algorithm.
- The initial state distribution $\Pi = \{\pi_i\}$, $1 \leqslant i \leqslant N$, where

  $$\pi_i = P[q_1 = S_i] = \begin{cases} 0, & i \neq 1, \\ 1, & i = 1. \end{cases} \tag{6}$$

- The last state distribution $\Gamma = \{\gamma_i\},\ 1 \leqslant i \leqslant N$, where

$$\gamma_i = P[q_T = S_i] = \begin{cases} 0, & i \neq N, \\ 1, & i = N. \end{cases} \tag{7}$$

- The state transition probability distribution $\boldsymbol{A} = \{a_{ij}\}$, where

$$a_{ij} = P[q_{t+1} = S_j \mid q_t = S_i], \quad 1 \leqslant i,\ j \leqslant N \tag{8}$$

and

$$a_{ij} = 0 \quad \text{if } (j < i) \text{ or } (j > i + \Delta). \tag{9}$$

The maximum number of forward jumps in each state (Δ) is chosen experimentally to be between 2 and 4 for each class during training.

- The observation symbol probability distribution $\boldsymbol{B} = \{b_j(m)\}$, where

$$b_j(m) = P[V_m \text{ at } t \mid q_t = S_j], \quad 1 \leqslant j \leqslant N,\ 1 \leqslant m \leqslant M. \tag{10}$$

- The occurrence degree of a given observation vector at state i, $\omega_i(y_t^{(k)})$, where

$$\omega_i\left(y_t^{(k)}\right) = \prod_{m=1}^{M} b_i(m)^{u_{tm}^{(k)}}, \quad 1 \leqslant i \leqslant N,\ 1 \leqslant t \leqslant T_k,\ 1 \leqslant k \leqslant K, \tag{11}$$

where $u_{tm}^{(k)}$ is the membership degree of observation vector $y_t^{(k)}$ belonging to cluster m, and $\sum_{m=1}^{M} u_{tm}^{(k)} = 1$.

The forward variable for a given word sample k is calculated as follows:

$$\alpha_t^{(k)}(j) = \begin{cases} \pi_j \omega_j(y_t^{(k)}), & t = 1, \\ \left[\sum_{i=1}^{N} \alpha_{t-1}^{(k)}(i) \cdot a_{ij} \right] \\ \quad \times \omega_j(y_t^{(k)}), & t = 2, \ldots, T_k, \end{cases} \tag{12}$$

for $1 \leqslant j \leqslant N$.

Similarly, the backward variable for a given word sample k is calculated as

$$\beta_t^{(k)}(j) = \begin{cases} \gamma_j, & t = T_k, \\ \sum_{i=1}^{N} a_{ji} \omega_j(y_t^{(k)}) \beta_{t+1}^{(k)}(i), \\ \quad t = T_k - 1, \ldots, 1, \end{cases} \tag{13}$$

for $1 \leqslant j \leqslant N$.

The observation probability is calculated as

$$P_K = P(Y^{(k)} \mid \lambda) = \sum_{i=1}^{N} \alpha_{T_k}^{(k)}(i) \cdot \gamma_i. \tag{14}$$

During training phase, each HMM is trained independently by the modified Baum–Welch algorithm. The re-estimation formulas of Baum–Welch algorithm are modified as follows:

$$\bar{a}_{ij} = \frac{\sum_{k=1}^{K} (1/P_k) \sum_{t=1}^{T_k - 1} \alpha_t^{(k)}(i) \cdot a_{ij} \cdot \omega_j\left(y_{t+1}^{(k)}\right) \cdot \beta_{t+1}^{(k)}(j)}{\sum_{k=1}^{K} (1/P_k) \sum_{t=1}^{T_k - 1} \alpha_t^{(k)}(i) \cdot \beta_t^{(k)}(i)}, \quad 1 \leqslant i,\ j \leqslant N, \tag{15}$$

$$\bar{b}_i(m) = \frac{\sum_{k=1}^{K} (1/P_k) \sum_{t=1}^{T_k} \alpha_t^{(k)}(i) \cdot \beta_t^{(k)}(i) \cdot u_{tm}^{(k)}}{\sum_{k=1}^{K} (1/P_k) \sum_{t=1}^{T_k} \alpha_t^{(k)}(i) \cdot \beta_t^{(k)}(i)}, \quad 1 \leqslant i \leqslant N,\ 1 \leqslant m \leqslant M. \tag{16}$$

The distributions Π and Γ are not re-estimated since they are predefined in right–left HMMs as shown in Eqs. (6) and (7).

5. Experimental results

A database consisting of about 17,000 images of 198 city names is used for developing Farsi handwritten word recognition. After applying preprocessing steps including binarization, noise removal, slope correction and baseline estimation, each word image is represented by its contour information. The chain-code histogram of frames, scanned from right to left by a sliding window over word image area, is used as the 20-dimensional feature vector. A fuzzy codebook consisting of 49 fuzzy code words is constructed using FUZZY clustering method from a pool of about 400,000 feature vectors extracted from word images in the training data set. By using this codebook, each word is represented as a sequence of membership vectors. For each city name, a separate right–left FVQ/HMM is trained by modified Baum–Welch algorithm (Eqs. (15) and (16)). In recognition phase, the probability of generating the test image by each HMM is computed by forward algorithm (Eq. (14)) and a sorted list of candidate classes is obtained. The performance of the proposed FVQ/

Table 1
Recognition rate of proposed FVQ/HMM method compare to other discrete HMM methods

Method	Recognition rate				
	Top-1	Top-2	Top-5	Top-10	Top-20
DHMM	32.04	44.72	69.47	86.41	93.63
DHMM + smoothing	65.05	76.09	86.08	90.83	95.00
FVQ/HMM	67.18	77.35	87.55	92.06	96.50

Table 2
Comparisons to other word recognition systems in the literature

Method	Lexicon size	Top-1	Top-2	Top-20
CDVDHMM, Chen et al. (1995)	271	67.0	75.5	88.3
Proposed system	198	67.2	77.4	96.5
DP, Kim and Govindaraju (1997)	100	84.6	91.2	99.0
KNN + HMM, Guillevic and Suen (1998)	30	86.7	94.6	99.9

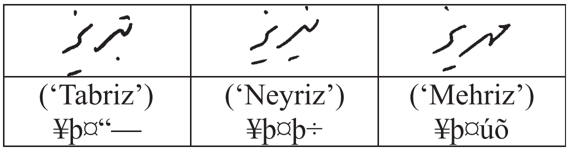

Fig. 4. An example of similar lexicon words.

HMM word recognition system is compared with conventional discrete HMM with/without smoothing and with some other Latin word recognition systems in Tables 1 and 2, respectively.

By examining the misclassified word images, the main causes of error are categorized below:

- Existence of some very similar lexicon entries (city names), which might easily be confused with each other even by a human observer without using any other contextual information. Fig. 4 shows such a confusing case.
- Incorrect baseline estimation, which affects the extracted feature vectors, and consequently decreases recognition rate considerably.

Such problems are normally resolved using other contextual information that is normally available in recognition applications.

References

Amin, A., 1998. Off-line Arabic character recognition: the state of the art. Pattern Recognition 31 (5), 517–530.

Bezdek, J., Ehrlich, R., Full, W., 1984. FCM: the fuzzy c-means clustering algorithm. Comput. Geosci. 10 (2&3), 191–203.

Chen, M.Y., Kundu, A., Srihari, S.N., 1995. Variable duration hidden Markov and morphological segmentation for handwritten word recognition. IEEE Trans. Image Process. 4 (12), 1675–1688.

Guillevic, D., Suen, C.Y., 1998. HMM-KNN word recognition engine for bank cheque processing. In: Proc. of International Conference on Pattern Recognition, Vol. 2. Brisebane, Australia, August 1998, pp. 1526–1529.

Hu, J., Yu, D., Yan, H., 1996. Algorithm for stroke width compensation of handwritten characters. Electronics Lett. 32 (24), 2221–2222.

Kim, G., Govindaraju, V., 1997. A Lexicon driven approach to handwritten word recognition for real-time applications. IEEE Trans. Pattern Anal. Machine Intell. 19 (4), 366–379.

Rabiner, L.R., 1989. A tutorial on hidden Markov models and selected applications in speech recognition. Proc. IEEE 77 (2), 257–286.

Tseng, H.P., Sabin, M.J., Lee, E.A., 1987. Fuzzy vector quantization applied to hidden Markov modeling. In: Proc. ICASSP. pp. 641–644.

Tsuboka, E., Nakahashi, J., 1995. Mathematical considerations and improvement on the fuzzy vector quantization-based hidden Markov model. Electronics Comm. Jpn., Part 3 78 (11), 9–22.