Optimization of DHMM Based on Chaotic Migration-Based GA for Chinese Signature Verification

Zhenhua Wu

College of Computer Science, Nanchang Hangkong University, Nanchang 330063 , P. R. China

Abstract

In this paper, Genetic Algorithm (GA) is used to train the parameters of Discrete Hidden Markov Model (DHMM). To overcome the premature convergence in GA, a chaotic migration strategy is introduced to the pseudo parallel genetic algorithm to increase the diversity of population. Because the GA's evolution speed is very slow, the Baum-Welch is applied to the GA. A floating matrix encoding mechanism is used for reflecting internal relations consisted in parameters of DHMM. This encoding method reduces the searching range of solutions space and increases the searching efficiency further. By using GA, the number of states can be adjusted dynamically. At last, the proposed method is used for signature verification. The promising experiment result indicates that the chaotic migration-based GA can optimize DHMM effectively.

Keywords: Genetic algorithm, Hidden markov model, Chaos, Migration strategy

1. Introduction

By the supporting of mature mathematics theory and successful application in automatic speech recognition, hidden markov model (HMM) as a classic statistic pattern recognition method has become increasingly popular in the last decade and has been applied to more application system such as handwriting recognition [1], face recognition [2] and on-line signature verification [3]. In our previous work [4], we applied discrete hidden markov model (DHMM) to off-line Chinese signature verification and gained a promising result.

As a complex model, there are many parameters, such as number of states, size of symbols and the topology of states etc to be decided before using classic Baum-Welch algorithm to train the DHMM. The Baum-Welch algorithm, which is a Maximum Likelihood method, is exploring for only one local maxima in practice. This method can not recover from the local maxima to obtain the global maxima or other more optimized local maxima, which causes the final model depends on the initial model much. If the data for training is not enough, there maybe many zero probability among parameters of DHMM using Baum-Welch algorithm to train the DHMM. Furthermore, the fixed number of states is not propitious for generating a more fitting model for each class.

Genetic Algorithm (GA) is a robust general-purpose optimization technique which evolves a population of solutions [5]. GA mimics nature evolution and performs global searching within the defined searching space. There are some literatures using GA to optimize HMM. By using GA for HMM parameters optimization, Sun et al [6] got better recognition rate than using Baum-Welch algorithm. He et al [7] used GA to train HMM to recognize isolated English word. In their method, the fitness function is not the likelihood of the model but the minimum classifier error rate. Chou et al. [8] presented a GA-HMM in which the hmm parameters are trained using GA. Because the initial population is generated using a random method, the GA-HMM needs great number of evolution generation. In addition, the number of state and the size of symbols must be pre-determined. So in literature [9], they optimized the structure and parameters of the HMM simultaneously for TIMIT corpus recognition, and got a better recognition rate. In literature [10], the authors used GA to evolve the structure of HMM, and applied Baum-Welch algorithm to train parameters of HMM having determinate structure. This method used Baum-Welch for training essentially. GA was just used for optimize the logic structure of the model. All these methods have some disadvantages, such as great number of evolution generation, converging to a local optimal solution etc. Based on the fact that increasing the diversity of population is help for recovering from the local optimal solution, Liu et al [11] presented a chaotic anneal genetic algorithm to train HMM parameters for hand gesture recognition. In this paper, a chaotic migration strategy is used to alter standard genetic algorithm for improving diversity of the population and increasing the possibility of converging to the global optimization solution. Then the altered GA is used to train the DHMM parameters for Chinese signature verification.

2. DHMM for Signature Verification

Usually, a HMM is characterized by the following:

0) N, the number of the states in the model. We denote the individual state $\theta=\{\theta_1, \theta_2, \cdots, \theta_N\}$ and the state in the time t q_t
0) M, the number of distinct observation symbols per state. We denote the individual symbol $V=\{V_1, V_2, \cdots, V_M\}$and the observation symbol in the time t O_t.
0) A, the state transition probability matrix. $A=(a_{ij})_{N\times N}$, where $a_{ij}=P(q_{t+1}=\theta_j \mid q_t=\theta_i)$, $1\leqslant i, j \leqslant N$.
0) B, the observation symbol probability matrix. $B=(b_j(k))_{N\times M}$, where $b_j(k)=P(O_t=V_k \mid q_t=\theta_j)$, $1\leqslant j\leqslant N$ and $1\leqslant k\leqslant M$.
0) Π, the initial state probability vector. $\Pi=(\pi_1, \pi_2, \cdots, \pi_N)$, where $\pi_i=P(q_1=\theta_i)$, $1\leqslant i\leqslant N$.

It can be seen from above discussion that the parameters N and M are implied by A and B. So, for convenience, a HMM is often represented by the compact notation $\lambda=(A, B, \Pi)$. According to the definition of A and B, the state transition probability and observation symbol probability must satisfy the following condition:

$$\sum_{j=1}^{N} a_{ij} = 1,\ 1 \le i \le N \text{ and} \tag{1}$$

$$\sum_{k=1}^{M} b_j(k) = 1,\ 1 \le j \le N .$$

Given a observation sequence $\boldsymbol{O}=O_1O_2\cdots O_T$ and a model $\lambda=(A, B, \Pi)$, we can use forward-back algorithm to calculate $P(\boldsymbol{O}|\lambda)$, the probability of observation sequence $\boldsymbol{O}$, given the model. On the contrary, using Baum-Welch algorithm the parameters of a model λ can be adjusted to augment $P(\boldsymbol{O}|\lambda)$, given the observation sequence $\boldsymbol{O}$.

3. Pseudo parallel GA for optimizing DHMM

As a global searching method, GA provids a chance which allows the searching process to escape from local maxima and to obtain a global maxima or at least other more optimized local maxima. By the evolution of population, GA can avoid too many zero probability of observation symbol brought by Baum-Welch for lacking of training data. In addition, through the use of good genetic operators, GA can improve the adverse effects of depending on initial model of Baum-Welch training method remarkably. But, if the initializing population is completely random process, GA needs a large number of evolving iterations to converge. So, we divide the initial population into two parts initialized in two different methods. The first one initialized with Baum-Welch, second with random generator. Then the global population is divided into some subpopulations. Each subpopulation evolves using different system parameters. To avoid the model "overfit" to training data, our training data are divided into two parts. One used for Baum-Welch to train the first initial parts, the other used for evaluating the fitness. Figure 1 shows the whole procedure of using GA to optimize DHMM.

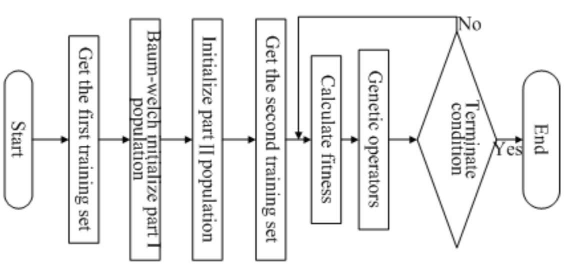

Fig. 1: The procedure using GA to optimize DHMM.

3.1. Encoding mechanism and fitness evaluation

It can be seen from the description of section 2 that there exist internal relations among parameters of HMM (represented by equation 1). The floating string encoding mechanism used by Chou [8] cannot reflect these relations. Furthermore, using this encoding method the number of states must be fixed. So, we use floating matrix encoding mechanism to represent the solution. Combined with fitting genetic operators, this encoding method is more fitting for optimizing DHMM. The past research indicates that compared with A and B, Π has low influence for model's performance. In view of this, a solution is encoded using real-valued and directly represented by 2 matrices $\lambda=(A, B)$.

Let n denote the size of the second training set for each class; O_i is the observation sequence of training sample i in training set; l_i denote the length of observation sequence O_i; $\lambda\mu$ represent the solution μ. The fitness evaluation function used in GA can be mathematically formalized by

$$E_\mu = 100 \times \exp\left(\sum_{i=1}^{n} \frac{\log(P(O_i \mid \lambda_\mu))}{l_i} \Big/ n\right). \tag{2}$$

3.2. Genetic operators

Two basic types of operators: crossover and mutation are used to create new individual based on existing individuals in the population. They provide the basic search mechanism of the GA.

When the training data is not enough, using Baum-Welch to train the model often bring too many zero probability of observation symbol for each state. One solution usually used to resolve this problem is combining two models λ1 and λ2 into one model λ. This idea can be mathematically formulated as following:

$$\lambda=\varepsilon\lambda_1+(1-\varepsilon)\lambda_2,\ \varepsilon \in [0, 1]. \quad (3)$$

Here, ε represents the weighting of the parameters of model λ_1, and $(1-\varepsilon)$ represents the weighting of the parameters of model λ_2. It is consistent with the floating arithmetic crossover operator developed by Michalewicz [5] when $\varepsilon \in (0, 1)$. If $\varepsilon \in (-1, 0)$ and λ_2 is better than λ_1 in terms of fitness value, it is consistent with the floating heuristic crossover operator. A key issue using this method is to determine the optimal value of ε. Because GA is a global search method, we randomly select the value of ε, namely, ε=U(0,1). U(0,1) is uniform distribution probability. The corresponding part between λ_1 and λ_2, such as A_1 and A_2, B_1^1 and B_2^1, and B_1^2 and B_2^2, are calculated using formula 3 to obtain the new model $\lambda=(A, B^1, B^2)$. Because the N_1 may be not equal to N_2, we adjust dimension of the matrices of the model which is worse than the other's in terms of fitness. The adjustment process is shown in figure 2. All the 3 matrices must be normalized to satisfy the condition denoted by formula 1. It is worth further exploration whether the A should be normalized to maintain the model structure unchanged. In our experiment, we normalize A to keep the model structure as left-right structure.

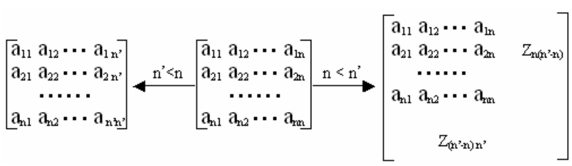

Fig. 2: Matrix adjusting process. All elements of Matrix Z is zero.

Mutation operator is very important for the searching process of the GA to escape from local maxima and to obtain global maxima or at least other more optimized local maxima. Floating uniform mutation method is used to finish this task. After mutation, the same normalization process for satisfying the condition denoted by formula 1 must be executed.

The selection of individuals to produce successive generations plays an extremely important role in a genetic algorithm. Normalized geometric ranking [5] method is a common selection operator. When all individuals have been sorted in descending order according to fitness value, the probability of selecting individual i is calculated as following:

$$P_i=q'(1-q)^{i-1}. \quad (4)$$

Here, q is the probability of selecting the best individual, which is a system parameter of the GA; $q'=q/(1-(1-q)^n)$, n is the size of population. The individuals of the next generation are selected from the previous generation and the new set of individuals created with crossover and mutation operators. To avoid the premature convergence, we divide the next generation into two parts. The first part is selected using normalized geometric ranking method discussed above. The second part is randomly selected according to uniform probability distribution. Furthermore, to ensure the fitness of the best individual in every population is a rising value, the best individual of one population must be selected to be one individual of the next generation.

3.3. Chaotic migration strategy

In parallel genetic algorithm, there are m subpopulations. Each subpopulation is evolved using different system options, such as different selection options, different crossover fraction, different mutation options etc. Migration provides a chance of information exchanging between different subpopulations. Solution's migration improves the diversity of the global population. A classic migration strategy is that the best solution of the nth subpopulation is copied to the (n+1)th subpopulation. The direction of migration may be forward or both (the nth subpopulation migrates into the (n-1)th subpopulation and the (n+1)th subpopulation). This method copies the best solution of the nth subpopulation to the next, but the best solution usually is not help for recovering from the local maxima. So, in our method the chaotic migration strategy is used.

Suppose there are m subpopulations: g_t^1, g_t^2, ⋯, g_t^i, ⋯, g_t^m. The size of each subpopulation is s_t^1, s_t^2, ⋯, s_t^i ⋯, s_t^m. Give each solution a series number. Such as, in the ith subpopulation $g_t^i = \{x_t^i(1), x_t^i(2), \cdots, x_t^i(s_t^i)\}$ the series number of each solution is $L^i = \{l^i \mid l^i=1, 2, \cdots, s^i\}$. All these subpopulations are combined to a global population: $G_t = \{ g_t^1, g_t^2, \cdots, g_t^i, \cdots, g_t^m \}$. Each solution in the G_t has a series number: $L = \{l \mid l = 1, 2, \cdots, S\}$. Here, S is the size of the G_t: $S = \sum_m s^i$. Each solution's two series number has the following relationship: $1 = 1^{i} + \sum_{j=1}^{i-1} s^j$.

After the series number of each solution is decided, we can apply chaotic migration strategy to these subpopulations. The Logistic map Chaos model is used to generate the chaotic series: $\Omega = \{\omega^1, \omega^2, \cdots, \omega^k, \cdots, \omega^m\}$. Here, m is the number of the subpopulations. Then the real series is mapped to integral series: $\Theta = \{\upsilon^1, \upsilon^2, \cdots, \upsilon^k, \cdots, \upsilon^m\}$, according to the following formula.

$$\upsilon^k = \sum_{j=1}^{k-1} s^j + \left\lfloor (\omega^k \cdot (\sum_{j=1}^{k} s^j - \sum_{j=1}^{k-1} s^j)) \right\rfloor . \quad \textbf{(5)}$$

Here, "$\lfloor \ \rfloor$" is a function of round down to the nearest integer. The migration process can be described as follow.

Step 0. Let k=2, the (k-1)th solution and kth solution are exchanged, i.e. $x_t(v^{(k-1)}) \leftarrow\rightarrow x_t(v^{(k)})$;

Step 0. Let k=k+1, exchange two solutions, i.e. $x_t(v^{(k-1)}) \leftarrow\rightarrow x_t(v^{(k)})$;

Step 0. If k<n go to step 2, else end migration process.

4. Signature verification experiment

4.1. Using DHMM for off-line signature verification

In our previous paper [4], firstly a signature image is thinned to gain the skeleton of the signature images binary scanned. Then a segmentation algorithm is used to gain a set of segmentation of signature. At last a six dimensions feature vector is extracted from each segmentation. The grouping vector quantization method brings little change for the notation of a HMM. The six-dimension feature vector is divided into two groups according to the physical significance. Two groups are quantized respectively. The observation symbol probability is divided into two parts. So, the notation λ of a HMM change to λ=(A, B1, B2, Π). Furthermore, we use the left-right structure model. Figure 3 shows the structure of this mode. There is no in-depth discussion for how to determine the appropriate number of states for each writer and the number of observation symbols in paper [4].

In this paper, the experiment data, feature extraction, vector quantization and the logic structure of model are the same with these in our previous paper [4]. A set of signatures coming from 32 writers makes up the signature database. There are 18 genuine samples, 8 simple forgeries and 5 simulated forgeries and 124 random forgeries for each writer. The genuine samples are divided into two parts. One used for training stage include 12 genuine samples. The other 6 genuine samples, all 8 simple forgeries, 5 simulated forgeries and 124 random forgeries consist of the test data for each writer. According to description above, the training data includes two parts. The first part is consisted by 6 genuine samples, and the second part is consisted by the other 6 genuine samples.

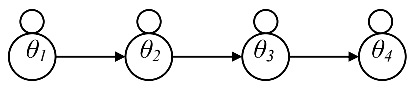

Fig. 3: A four states left-right model.

4.2. Experiment Results

Because GA and Baum-Welch can make up the shortness of each other, they are combined to training DHMM. GA provides a chance which allows the searching process to escape from local maxima and to obtain a global maximum or at least other more optimized local maxima. Using Baum-Welch to generate the parts of initial solutions decreases the evolution generation of GA. In the first experiment, the performances of GA in two situations that one is the initial solutions are generated randomly and the other is parts of the initial solutions are generated using Baum-Welch are compared. Figure 4 shows the comparison result. It can be seen from this figure, not only the initial population of using Baum-Welch is better than that of no Baum-Welch, but also the velocity of convergence is quicker when using Baum-Welch.

In second experiment, the no optimization model and the model optimized with chaotic migration-based GA are used for the first writer's signature verification respectively. The system parameters used in GA are given in table 1. Verification results are shown in figure 5. The six samples pointed in box are genuine sample the others are forgeries. It can be seen from figure 5, the model optimized with GA is more propitious for forgeries identifying. There are only 3 forgeries whose verification value is bigger than 0.3 in optimized model. The corresponding number is 7 in no optimization model. Furthermore, majority forgeries' verification value is smaller than 0.1 in optimized model. In no optimization model there are many forgeries whose verification value is bigger than 0.1.

To compare the chaotic migration to the best solution migration (BSM), we carry out 5 verifications using DHMM optimized with chaotic migration-based GA and the DHMM optimized with BSM-based GA respectively for verifying all 32 writers' test signatures (each writer has 143 test signatures) in the third experiment. Table 2 gives the result of this experiment. Both HMM and GA have heuristic property, so the result of chaotic migration-based GA isn't always better than the BSM-based GA. But the average value indicates the chaotic migration-based GA improves the correct rate of verification. This can be more clearly seen from figure 6. One hundred verifications for all 32 writers' test signatures (each writer has 143 test signatures) are carried out using two kind of GA respectively in forth experiment. Figure 6 shows the comparison of two sets of average error rate.

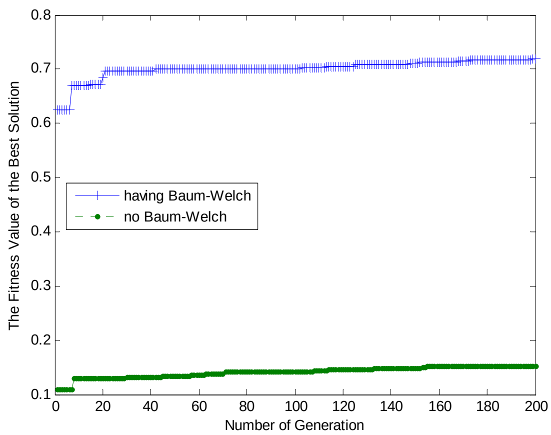

Fig. 4: The comparison of GA's performances in two different situations.

Using floating matrix encoding mechanism provides a chance for adjusting the number of states dynamically. Comparing to using fixed number of states, adapting number of states is more propitious to create unique signature model for each writer. Table 3 gives the comparison of number of states. Baum-Welch is used repeatedly to train 40 models for each writer. The state numbers of the 40 models consist of 3 to 10. For each number, five models are trained using Baum-Welch. The model which fitness value is maximal is selected to count the number of states which consist the values of the row of "NO GA". The values of "NO GA" only include two numbers: 9 and 10. In "Having GA" the values are a few more scattered.

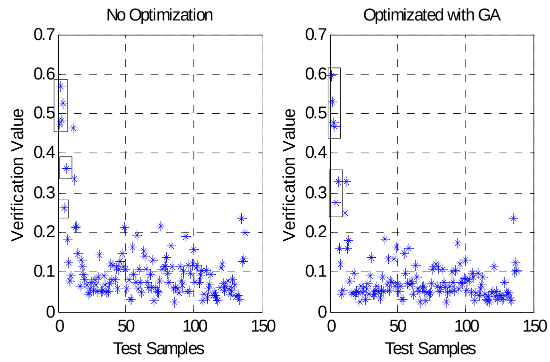

Fig. 5: The comparison of verification result between having GA and no GA.

Parameter	Value
Population size	100
Subpopulation size	20
Iteration	300
Selection Probability	0.06~0.08
crossover fraction	0.6~0.8
mutation probability	0.02~0.06

Table 1. The system parameter of the GA.

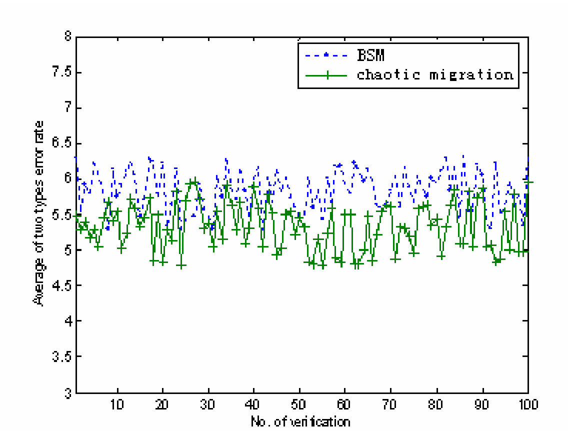

Fig. 6: Comparison of two sets of average error rate.

5. Conclusions

The experiment results indicate chaotic migration-based GA is effective for optimizing DHMM. Chaotic migration-based GA has a higher probability in finding global maxima or at least local maxima with better performance. But if the initial solutions are generated randomly, GA needs a substantial number of evolution generations. GA and Baum-Welch can make up the shortness of each other. Combining GA and Baum-Welch to train DHMM can get more optimization model.

Although we applied two kinds of initial and selection mechanism, a larger mutation probability and the chaotic migration strategy, the premature convergence isn't completely eliminated. Furthermore, GA brings a tremendous amount of calculation. How to solve these problems will be the main work in future.

References

[1] A.W. Senior and A. J. Robinson, An Off-line Cursive Handwriting Recognition System, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3):309-321, 1998.

[2] X.J. Liu, D. F. Wan and L. F. Zhang, An Approach for Face Recognition Based on Singular Value Decomposition and Hidden Markov Model, *Chinese Journal of Computers*, 26(3):340-344, 2003.

[3] R.S. Kashi, J.Hu, W.L. Nelson and W. Turin, On-line Handwritten Signature Verification Using Hidden Markov Model Features, *In Proceedings of the Fourth International Conference on Document Analysis and Recognition, Los Alamitos*, *USA:IEEE Computer Society*, 1: 253-257, 1997.

[4] X.S. Chen, Z.H. Wu and D. J. Xiao, Off-line Chinese Signature Verification Based on Segmentation and HMM, *ACTA AUTOMATICA SINICA*, 33(2):205-210, 2007.

[5] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs. 3rd ed*, Springer-Verlag, Berlin Heidelberg New York , 1996.

[6] F. Sun and G.R. Hu, Speech Recognition Based on Genetic Algorithm for Training HMM, *IEE Electronic Letters*, 34(6):1563-1564, 1998.

[7] Q. H. He, G. Wei and L. W. Jin, A Genetic Algorithm Based MCE for HMM Training, *Journal of Circuits and Systems*, 4(4):46-50, 1999.

[8] C. W. Chau, S. Kong, C. K. Diu and W. R. Fahrner, Optimization of HMM by a Genetic Algorithm, *In proceedings of the IEEE International Conference on acoustics*, *Speech, and Signal Processing*, pp.1727-1730, 1997.

[9] S. Kong, C. W. Chau and K. F. Man, Optimisation of HMM Topology and Its Model Parameters by Genetic Algorithms, *Pattern Recognition*, 34(2):509-522, 2001.

[10] K. J. Won, P. B. Adam and K. Anders, Evolving the Structure of hidden Markov Models, *IEEE Transactions on Evolutionary Computation*, 10(1):39-49, 2006.

[11] J. H. Liu, J. S. Cheng and J. P. Chen, Optimization of HMM Parameters Based on Chaos and Genetic Algorithm for Hand Gesture Recognition, *Journal of Systems Engineering and Electronics*, 13(4):79-84, 2002.

		1	2	3	4	5	Average
BSM (%)	FRR	8.33	5.73	6.25	8.33	4.17	6.56
	FAR	5.63	6.48	6.34	5.5	6.55	6.1
Chaotic migration (%)	FRR	5.21	6.77	6.25	4.17	5.73	5.63
	FAR	5.52	4.68	5.38	5.79	5.47	5.37

Table 2. The verification results compare between BSM and chaotic migration. Where, FRR is fault-rejected rate, FAR is fault-accepted rate.

No. of Writer		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Number of states	No GA	10	9	9	10	9	10	10	9	9	10	10	10	10	10	10	9
	Having GA	10	9	8	10	5	8	9	10	6	3	10	7	10	7	6	7
No. of Writer		17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
Number of states	No GA	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10
	Having GA	10	9	6	5	10	5	10	9	10	3	9	6	6	8	7	4

Table 3. The comparison of number of states between having GA and no GA.