Vol 16 No 6, June 2007
1009-1963/2007/16(06)/1619-05

Chinese Physics

© 2007 Chin. Phys. Soc.
and IOP Publishing Ltd

Discrete channel modelling based on genetic algorithm and simulated annealing for training hidden Markov model*

Zhao Zhi-Jin(赵知劲)†, Zheng Shi-Lian(郑仕链),
Xu Chun-Yun(徐春云), and Kong Xian-Zheng(孔宪正)

Telecommunication School, Hangzhou Dianzi University, Hangzhou 310018, *China*

(Received 27 August 2006; revised manuscript received 27 October 2006)

Hidden Markov models (HMMs) have been used to model burst error sources of wireless channels. This paper proposes a hybrid method of using genetic algorithm (GA) and simulated annealing (SA) to train HMM for discrete channel modelling. The proposed method is compared with pure GA, and experimental results show that the HMMs trained by the hybrid method can better describe the error sequences due to SA's ability of facilitating hill-climbing at the later stage of the search. The burst error statistics of the HMMs trained by the proposed method and the corresponding error sequences are also presented to validate the proposed method.

Keywords: hidden Markov model, discrete channel model, genetic algorithm, simulated annealing
PACC: 0650

1. Introduction

Hidden Markov models (HMMs)[1] have been used for modelling of various systems: speech and image recognition, telecommunications, and automatic control. The study of the HMM for real communication channels error statistics was initiated by Gilbert.[2] One of the primary limitations of Gilbert's model is that it lacks the ability to describe more complex bit error state transitions due to its simple geometric distribution. Fritchman[3] improved Gilbert's model by proposing a state partitioned model in which there is more than one good state and errors occurring only in one bad state with probability of one. However, the Fritchman model assumes that the intervals between consecutive errors are independent and identically distributed, and it is not always true for experimental data.[4] A solution for this problem is to use a channel model with more than one good state and bad state, as a real communication model can be modelled by an HMM. Several works have been devoted to model the burst errors in radio channels by using HMM.[5−7]

One of the basic problems of HMMs is to determine a method to adjust the model parameters to maximize the probability of the observation sequence given the model. The Baum–Welch algorithm[1] is the classical iterative method used to estimate and train the HMM model to best describe the training observation sequences. However, it starts from an initial guess and cannot achieve global optimization. The genetic algorithm (GA)[8] mimics natural evolution and performs global searching within the defined searching space, and it has been widely used in information, computer and communications field and chaotic system.[9] Using GA to train HMM can get better results than the Baum–Welch algorithm.[10−12] Although there is performance improvement when applying GA to train HMM, GA still has some limitations. The GA is quite poor at hill-climbing and also has the problem of premature convergence.[13] In order to improve GA's ability of hill-climbing, simulated annealing (SA)[13,14] is embedded in it and the resulting hybrid method is used to train HMM in this paper.

The rest of the paper is organized as follows. In Section 2, discrete channel modelling with HMMs is discussed. Section 3 focuses on the detailed steps of the hybrid method of using GA and SA to train HMM. Some experimental results are shown in Section 4. Finally, concluding remarks are offered in Section 5.

*Project supported by Pre-Research Foundation of Electronics Science Research Institute (Grant No 41101040102).
†E-mail: zhaozj03@hdu.edu.cn

2. HMMs for discrete channel modelling

The discrete channel model[15] can be represented as shown in Fig.1, where the input sequence at point A and the output sequence at point B are discrete symbol vectors which are denoted as $\boldsymbol{X} = [X_1X_2\ldots X_k\ldots]$ and $\boldsymbol{Y} = [Y_1Y_2\ldots Y_k\ldots]$ respectively.

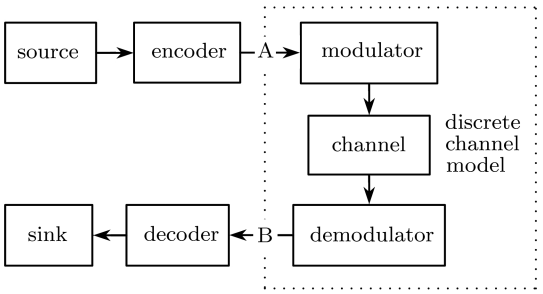


Fig.1. Elements of a discrete channel model.

Between the input and the output, the perturbations due to the physical channel, nonideal modulator and demodulator can be considered as an introduction of an error sequence $\boldsymbol{O} = [O_1O_2\ldots O_k\ldots]$. As a consequence the output sequence can be represented as $\boldsymbol{Y} = \boldsymbol{X} \oplus \boldsymbol{O}$, where $\boldsymbol{X}, \boldsymbol{O}$ and $\boldsymbol{Y}$ are considered as elements of a vector space over Galois field GF(q) and $\oplus$ as vector addition over the same field.[3] In this framework the channel characterization problem becomes that of statistically describing the error sequence generated by noise source. HMMs have been widely used to model such error sources. In those models the observed sequence of errors is assumed to be a function of underlying Markov chain which describes a sequence of hidden channel states. In general, HMM is characterized by the following elements:

(1) Set of HMM states $Q = \{q_1, q_2, \ldots, q_N\}$, where N is the number of states in the model. The state at time t is denoted as S_t.

(2) Set $V = \{v_1, v_2\ldots, v_M\}$ of M symbols which can be observed. In consideration of binary channels, $M = 2$ and $V = \{v_1, v_2\} = \{0, 1\}$, where 0 indicates correct reception of a symbol and 1 indicates an incorrect reception. An observation sequence (error sequence) is denoted as $\boldsymbol{O} = O_1O_2\ldots O_T$, where each observation $O_t(1 \le t \le T)$ is one of the symbols from V and T is the number of observations in the sequence.

(3) State transition probability matrix $A = \{a_{ij}\}$, where $a_{ij} = P(S_{t+1} = q_j|S_t = q_i),\ 1 \le i,\ j \le N$.

(4) Observation probability matrix $B = \{b_{jk}\}$, where $b_{jk} = P(O_t = v_k|S_t = q_j), 1 \le j \le N,\ 1 \le k \le M$.

(5) Initial state probability vector $\Pi = \{\pi_1, \pi_2, \ldots,\ \pi_N\}$, where $\pi_i = P(S_1 = q_i),\ 1 \le i \le N$.

Elements in A, B and Π must satisfy the following two conditions:

$$0 \le a_{ij} \le 1,\quad 0 \le b_{jk} \le 1,\quad 0 \le \pi_i \le 1,\quad \forall i, j, k, \tag{1}$$

$$\sum_{j=1}^{N} a_{ij} = 1(1 \le i \le N),\quad \sum_{k=1}^{M} b_{jk} = 1(1 \le j \le N),\quad \sum_{i=1}^{N} \pi_i = 1. \tag{2}$$

For convenience, the compact notation $\lambda = \{A, B, \Pi\}$ is used to indicate the complete parameters of HMM.

Training problem is one of the three basic problems for HMMs: how can the HMM parameters $\lambda = \{A, B, \Pi\}$ be adjusted to maximize $P(\boldsymbol{O}|\lambda)$ the probability of the observed sequence given the model? A hybrid algorithm using GA and SA is used to solve this problem in this paper.

Given appropriate values of λ, the HMM can be used as a generator to produce an error sequence $\boldsymbol{O} = O_1O_2\ldots O_T$ as follows:[1]

(1) Set $t = 1$ and choose an initial state $S_1 = q_i$ according to the initial state distribution Π.

(2) Choose $O_t = v_k$ according to the observation probability distribution at state q_i, i.e., b_{ik}.

(3) Transit to a new state $S_{t+1} = q_j$ according to the transition probability distribution of state q_i, i.e., a_{ij}.

(4) Set $t = t + 1$, and go back to 2 if $t < T$; otherwise, terminate the procedure.

3. A hybrid method for training HMM

A GA is applied to search for the best HMM, and a SA is also embedded in the GA search in order to improve GA's ability of hill-climbing and converge to the globally optimal solution or a better locally optimal solution. The detailed implementation of the proposed method is described as follows.

3.1. Genetic algorithm

The fundamental of the GA structure is the encoding mechanism for representing the optimization

problem's variables. A HMM with N states has $N^2 + N + NM$ parameters to be trained. All of the parameters are encoded into a string of real numbers which range from 0 to 1 to construct a chromosome as shown in Fig.2.

a_{11}	a_{12}	$\cdots$	a_{1N}	a_{21}	$\cdots$	a_{2N}	$\cdots$	a_{NN}	b_{11}	$\cdots$	b_{1M}	b_{21}	$\cdots$	b_{NM}	π_1	π_2	$\cdots$	π_N

Fig.2. The structure of chromosome.

The initial population is randomly generated in which every individual has the structure as shown in Fig.2. Note that the parameters of each chromosome must be normalized to satisfy Eq.(2).

The objective function $P(\boldsymbol{O}|\lambda)$ provides the mechanism for evaluating each chromosome. However, the values of $P(\boldsymbol{O}|\lambda)$ vary largely in a population and a few chromosomes with relatively high values cause the proportionate selection scheme to allocate a large number of offspring to these chromosomes and they take over the population, causing premature convergence. To overcome this problem, a scaling mechanism is proposed as follows:

$$f = \lg[P(\boldsymbol{O}|\lambda)], \tag{3}$$

$$f' = f - \min\{f\} + 1, \tag{4}$$

where f' is the finally used fitness function, and $P(\boldsymbol{O}|\lambda)$ is calculated by forward procedure.[1]

Selection models nature's survival-of-the-fittest mechanism. Fitter solutions have more chances to survive while weaker ones eliminate generally. Crossover and mutation are two traditional genetic operators of GA. They simulate natural genetic recombination and variation respectively. The roulette wheel selection scheme [8] and the two-point crossover scheme [8] are used in the proposed method. To carry out the mutation operation, a parameter is randomly chosen and replaced by a real number which ranges from 0 to 1 if the condition of mutation is positive. Note that after crossover and mutation, the parameters in chromosome are changed and must be normalized to satisfy Eq.(2).

After selection, crossover and mutation, the algorithm creates the subsequent generation from the strings of the current population. This generational cycle is terminated when the number of generation reaches the maximum generation.

3.2. Embedded simulated annealing algorithm

After crossover, SA is applied to decide whether to accept the new individuals or not. Suppose $f_{1\text{old}}, f_{2\text{old}}, f_{1\text{new}}$, and $f_{2\text{new}}$ are the log likelihood probabilities of the old chromosomes $I_{1\text{old}}$, $I_{2\text{old}}$, and the new chromosomes after crossover $I_{1\text{new}}$ and $I_{2\text{new}}$ respectively, then $I_{1\text{old}}(I_{2\text{old}})$ is replaced with $I_{1\text{new}}(I_{2\text{new}})$ based on the following acceptance probability:

$$P = \begin{cases} 1, & (\Delta f < 0), \\ \exp\left(\dfrac{\Delta f}{T_g}\right), & (\Delta f \geq 0), \end{cases} \tag{5}$$

where

$$\Delta f = \frac{f_{1\text{old}} + f_{2\text{old}} + f_{1\text{new}} + f_{2\text{new}}}{4} - f_{1\text{new}}(f_{2\text{new}}),$$

and T_g is a control parameter which is generally called temperature, where g is the number of the generation. Note that the log likelihood probabilities are calculated by using Eq.(3). In the proposed method, T_g is decreased as $T_g = T_0(\alpha^{g-1})$, where $\alpha \in (0, 1)$ and T_0 is the initial temperature. Similarly, SA is also applied to accept the new individual after mutation. The acceptance probability has the same format as Eq.(5). The only difference is that $\Delta f = f_{\text{old}} - f_{\text{new}}$, where f_{old} is the log likelihood probability of the chromosome before mutation and f_{new} after mutation.

It is worthwhile pointing out that as the generation increases, T_g decreases and hence the probability of accepting the new individual which has less log likelihood probability than the desired decrease. This means that at the early stage of the genetic algorithm when g is small, the proposed method accepts most new individuals in order to enlarge the search space, while when g gets close to the maximum generation, the proposed method tends to accept the 'better' individuals and discard most 'worse' individuals after crossover or mutation. So the proposed method not

only preserves the advantages of GA but also strengthens the ability of hill-climbing at the later stages of the search.

4. Experimental results

The results trained by the hybrid method are compared with the results of pure GA. The HMM has $N = 8$ states and $M = 2$ to represent binary values. The GA uses a 30 chromosome population with a crossover rate of 0.8 and a mutation rate of 0.05. The maximum generation is set 3000. Note that different crossover rate or mutation rate may lead to different training results. High crossover rate can help the algorithm converge to the expected optimal region quickly, but may result in premature convergence. The higher the mutation rate, the more variable the individuals and the less stable the algorithm. The crossover rate is normally between 0.4 and 0.9, and the mutation rate normally ranges from 0.001 to 0.1.[16] The initial temperature of the SA T_0 must be large enough to ensure that the algorithm converges to a better solution,[17] and it is set 100 in the simulation. Another parameter of the embedded SA is set relatively high ($\alpha = 0.995$) to ensure that at the early stage of the search, the SA can accept most new individuals to enlarge the searching space. Four error sequences are used and 50 experiments are conducted to each error sequence by using the hybrid method and pure GA respectively. The average log likelihood probabilities $\bar{P} = \left\{ \sum_{i=1}^{50} \lg[P(\boldsymbol{O}_k|\lambda_i)] \right\} /50$, the maximum log likelihood probabilities $P_{\max} = \max_{1\le i\le 50} \{\lg[P(\boldsymbol{O}_k|\lambda_i)]\}$ and the minimum log likelihood probabilities $P_{\min} = \min_{1\le i\le 50} \{\lg[P(\boldsymbol{O}_k|\lambda_i)]\}$ of the 50 experiments are shown in Table 1, where $\boldsymbol{O}_k$ denotes the kth error sequence and λ_i denotes the HMM trained in the ith experiment which has the maximum log likelihood probability when the algorithm is terminated, $1 \le k \le 4, 1 \le i \le 50$. Table 1 illustrates that the HMMs trained by the hybrid method have higher average log likelihood probabilities than HMMs trained by pure GA and hence they can better describe the error sequences and obtain more accurate models for discrete channels.

Table 1. Experimental results of the hybrid method and pure GA.

error	$\bar{P}$		$P_{\max}$		$P_{\min}$	
sequence	pure GA	hybrid	pure GA	hybrid	pure GA	hybrid
$\boldsymbol{O}_1$	−33.7550	−31.2426	−33.3346	−29.2991	−34.2747	−32.6349
$\boldsymbol{O}_2$	−52.0830	−49.3836	−51.3035	−47.0644	−53.1278	−50.6780
$\boldsymbol{O}_3$	−87.0068	−84.4051	−86.7821	−81.6327	−87.2206	−86.0729
$\boldsymbol{O}_4$	−143.3300	−140.0541	−142.1388	−137.4340	−144.3852	−142.0406

In order to evaluate the impact of the crossover rate to the training result, Fig.3 shows the average log likelihood probabilities trained by the hybrid method and pure GA when different value of crossover rate is set. The other parameters of the algorithm are kept steady as above. The error sequence used here is $\boldsymbol{O}_1$. Slight difference of the average log likelihood probability is obtained when the crossover rate is changed. So it seems that both the hybrid method and pure GA are robust to the changing crossover rate.

Figure 4 shows the burst error statistics of the HMMs trained by the hybrid method and the corresponding error sequences. How to simulate the burst error statistics of the HMM please refer to Ref.[15]. Close similarities between the histograms of the length of error bursts indicate that the HMMs trained by the proposed method can well describe the burst error statistics of the channels.

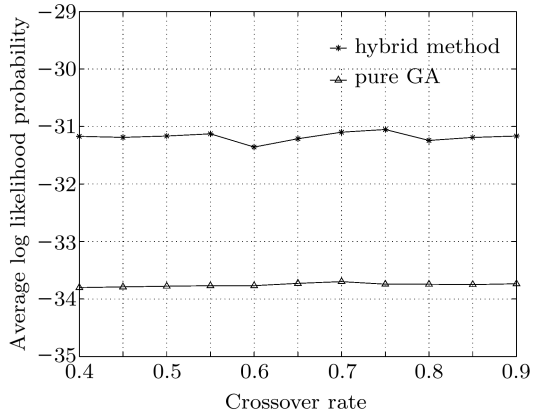


Fig.3. Average log likelihood probability with different crossover rate.

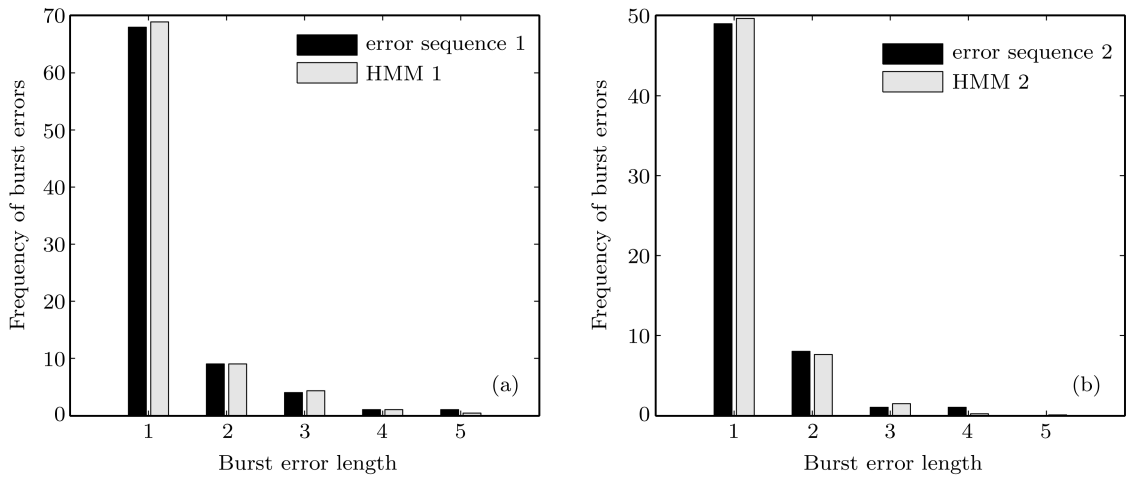


Fig.4. Burst error statistics of the HMMs versus the error sequences.

5. Conclusions

The method of hybridizing GA with SA to train HMM for discrete channel modelling is presented. The hybrid method is compared with pure GA and experimental results show that the HMMs trained by the proposed method have higher average log likelihood probabilities and thus they can better describe the error sequences due to SA's ability of facilitating hill-climbing in the later stage of the search. Close continuity between the burst error length statistics of the HMMs trained by the hybrid method and the corresponding error sequences indicates that the HMMs trained by the proposed method can well describe the burst error statistics of the channels.

References

[1] Rabiner L R 1989 *Proceedings of the IEEE* **77** 257
[2] Gilbert E N 1960 *B.S.T.J.* **39** 1253
[3] Fritchman B D 1967 *IEEE Trans. Inform. Theory* **IT-13** 221
[4] Turin W and Sondhi M M 1993 *IEEE J. Sel. Areas in Commun.* **11** 340
[5] Umbert A and Diaz P 2000 *IEEE 52nd Vehicular Technology Conference* **5** 2173
[6] Vinicius M, Pinto E L and Grivet M 2004 *IEEE International Conference on Communications* **1** 567
[7] Umbert A and Diaz P 2004 *IEEE 15th International Symposium on PIMRC* **4** 2998
[8] Srinivas M and Patnaik L M 1994 *IEEE Computer* **27** 17
[9] Dai D, Ma X K, Li F C and You Y 2002 *Acta Phys. Sin.* **51** 2459 (in Chinese)
[10] Chau C W, Kwong S, Diu C K and Fahrner W R 1997 *Proc. ICASSP*-97 **3** 1727
[11] Sun F and Hu G 1998 *Electron. Lett.* **34** 1563
[12] Rondeau T W, Rieser C J, Gallagher T M and Bostian C W 2004 *IEEE Proc. IMS* **2** 739
[13] Adler D 1993 *IEEE International Conference on Neural Networks* 1104
[14] Li S Y, Du Z H, Wu M Y, Zhu J and Li S L 2001 *Acta Phys. Sin.* **50** 1260 (in Chinese)
[15] Tranter W H, Shanmugan K S, Rappaport T S and Kosbar K L 2004 *Principles of communication systems simulation with wireless applications* p460
[16] Wang X P and Cao L M 2004 *Genetic algorithm: theory, application and software realization* p47 (in Chinese)
[17] Yi H 2006 *Research on the network load balancing algorithm based on simulated annealing genetic algorithm* p43 (in Chinese)