

OPTIMIZATION OF HMM BY A GENETIC ALGORITHM

C.W. Chau*, S. Kwong*, C.K. Diu**, W.R. Fahrner***

*Department of Computer Science, City University of Hong Kong, Hong Kong

**Department of Applied Computing, Open Learning Institute, Hong Kong

***Department of Electrical Engineering, FernUniversität, Germany

ABSTRACT

Hidden Markov Model (HMM) is a natural and highly robust statistical methodology for automatic speech recognition. It is also being tested and proved considerably in a wide range of applications. The model parameters of the HMM are essence in describing the behavior of the utterance of the speech segments. Many successful heuristic algorithms are developed to optimize the model parameters in order to best describe the trained observation sequences. However, all these methodologies are exploring for only one local maxima in practice. No one methodology can recovering from the local maxima to obtain the global maxima or other more optimized local maxima. In this paper, a stochastic search method called Genetic Algorithm (GA) is presented for HMM training. GA mimics natural evolution and perform global searching within the defined searching space. Experimental results showed that using GA for HMM training (GA-HMM training) has a better performance than using other heuristic algorithms.

1. INTRODUCTION

The Hidden Markov Model (HMM) [1,2] model parameters are the most important data in a HMM based speech recognizer because the speech segment can be characterized and represented by the HMM model parameters, thus, it directly affects the system's recognition accuracy. For artificial speech, the model parameters of the HMM are known in advance and can be used for recognition, however, in the case of natural human speech, these parameters can only be estimated. Many successful heuristic algorithms such as the forward-backward method [3] and the gradient method [4] are developed to optimize the model parameters to best describe the training observation sequences. However, all these methodologies are starting from an initial guess and iteratively converge to a local maxima. A stochastic search method called Genetic Algorithm (GA) can be used for HMM training. Unlike the hillclimbing method

used by many heuristic algorithms, GA mimics natural evolution and performs global searching within the defined searching space. Figure 1 describes the structure of Genetic Algorithm. As shown in the figure, GA is manipulated on a population of individuals or genetic representation of solutions. An encoding mechanism is then employed to encode the potential solutions. After initialization, the fitness values of the solutions are evaluated and act as index to describe how best are the solutions to the problem. Each pass of the while-loop in the figure mimic one natural evolution. The selection leads fitter solutions to have more chance of survival and reproduction or more its fitter genes survive in subsequent generations. On the other hand, weaker solutions die out generally. Crossover and mutation are two traditional genetic operators of GA. They simulate natural genetic recombination and variation respectively. Crossover exchanges portions of encoded solutions (genes) to generate new solutions, hopefully more optimized solutions. Mutation randomly alter portions of encoded solutions. Mutation plays an important role in GA which can regain missed information that are not generated during initialization of solutions and provide global search capability to GA. The final step in each evolution cycle is the evaluation of fitness values which will be used as references for the selection in the next generation.

```
Program Genetic Algorithm ()
{
    initialize population;
    evaluate population;
    while (termination conditions not reached)
    {
        select solutions for next population;
        perform crossover and mutation
        evaluate population;
    }
}
```

Figure 1. Structure of Genetic Algorithm [5].

The proposed GA-HMM training provided a mechanism which allows the searching process to escape from the local maxima and to obtain a global maxima or at least other more optimized local maxima. In fact, GA can be applied to various optimization or combination problems, provided that appropriate encoding mechanism is given.

2. ENCODING MECHANISM

In this paper, the configuration of the HMM used is a five states left-right model [3] and a set of 256 observation symbols. As shown in figure 2, this model has the desirable property that it can readily model speech signal whose properties change over time in a successive manner.

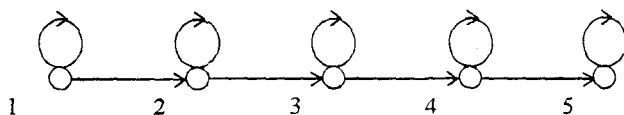


Figure 2. A five states left-right model.

The model parameters of HMM consisted of two matrices: **A** and **B**. Matrix **A** is a 5-by-5 transition probability distribution matrix, its element at row *i*, column *j* is the probability $a_{i,j}$ of transition from current state *i* to next state *j* and must satisfy the following condition:

$$1 = \sum_{j=1}^5 a_{i,j} \quad \text{where } i = 1, \dots, 5 \quad (1)$$

Matrix **B** is a 5-by-256 observation symbol probability distribution matrix, its element at row *i*, column *k* is the probability $b_{i,k}$ of observation symbol with index *k* emitted by current state *i* and must satisfy the condition:

$$1 = \sum_{k=1}^{256} b_{i,k} \quad \text{where } i = 1, \dots, 5 \quad (2)$$

In GA-HMM training, the model is encoded into a string of real numbers. As shown in figure 3, this string acts as a chromosome and is composed of two parts: **A'** and **B'**. These two parts: **A'** and **B'** are formed by concatenate the rows of matrices **A** and **B** respectively. Due to the configuration of the model, some transitions between states do not exist so that the corresponding elements in matrix **A** are constantly zero and these elements will not be encoded as genes of the

chromosomes. Therefore, a total of 9 elements (genes) of matrix **A** are coded as a chromosome.

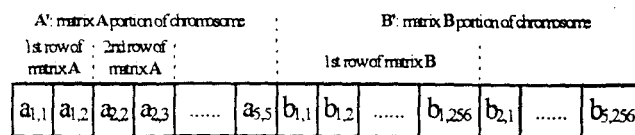


Figure 3. Genetic representation of HMM model.

3. SELECTION MECHANISM

GA-HMM training uses the roulette wheel selection scheme [5] as its selection mechanism. In the selection, each solution is allocated a sector of the roulette wheel with the angle subtended by sector at the center of the wheel, which is equal to 2π multiplies by the fitness value of the solution. A solution is selected as an offspring if a randomly-generated number in the range 0 to 2π is falls into the sector corresponding to the solution. The algorithm selects solutions in this manner until the entire population of next generation has been produced.

4. Crossover AND MUTATION

One point and two points crossover are used in the GA-HMM training. As shown in figure 4, two parent chromosomes are randomly chosen from the population and a random number between 0 and 1 is generated to compare with the crossover rate. If the random number is greater than the crossover rate, then the two parents act as offspring and no crossover occurred. Otherwise, one cross point and two cross points will be randomly selected in the parts **A'** and **B'** respectively. The portions of the chromosomes between the selected cross points are exchanged to generate new chromosomes. The mutation in GA-HMM training alter three real numbers at a time. Mutation randomly choose one chromosome from the population as the parent and similar to the crossover operator. It uses a randomly generated number and mutation rate to determine if mutation can be carried out or not. If the decision is positive, then three probabilities are randomly selected for the parent, one probability is selected for the part **A'** and two probabilities are selected for the part **B'**. Then the three selected probabilities are replaced by three random generated real numbers. After crossover or mutation, the probabilities in chromosome are changed and must be normalized to satisfy Equations (1) & (2).

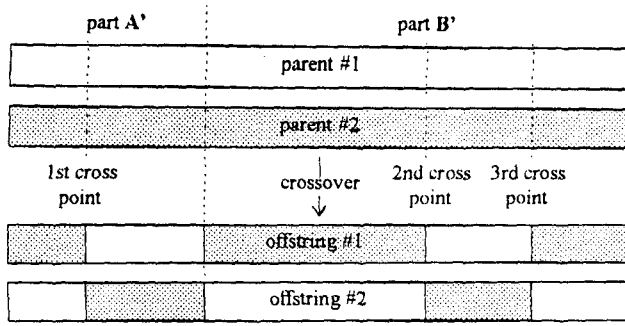


Figure 4. Three points crossover.

5. FITNESS VALUE EVALUATION

The average probability p_n of the HMM solution λ_n generates the training observation sequences O_1, \dots, O_M must be calculated as references for the fitness evaluation:

$$p_n = \left(\sum_{i=1}^M p(O_i | \lambda_n) \right) / M \quad (3)$$

where $p(O_i | \lambda_n)$ is calculated by forward procedure [1]. Then the fitness value f_n of solution λ_n will be calculated as follows:

$$f_n = p_n / \sum_{i=1}^N p_i \quad (4)$$

where N is number of solutions in the population.

6. EXPERIMENTAL RESULTS

The results of GA-HMM training was compared to the results trained by forward-backward procedure described by Rabiner and Juang [1]. The initial model parameters in each experiment were generated by an uniform random number generator and is normalized to satisfy Equations (1) & (2). and 10 experiments are conducted. In each experiment, two HMMs are trained with same 10 observation training sequences using GA-HMM training and forward-backward procedure respectively. In GA-HMM training the following control parameters are used:

population size	30
crossover rate	0.01
mutation rate	0.0001

Each results of HMM is consisted of two values: p_{same} - average log probability of the HMM generated by the 10 training observation sequences of the same HMMs and $p_{different}$ - average log probability of the HMM generated by the 90 training observation sequences of the other HMMs. In the experiment, the HMM training using forward-backward procedure will terminated after 200 turns or the increase of the average log probability p_{same} less than 0.00001 and GA-HMM training will be terminated after 20000 generations.

Experimen	#1	#2	#3	#4	#5
Genetic Algorithm					
P_{same}	-4.9473	-3.5693	-3.2932	-3.0982	-4.2345
$P_{different}$	-7.4982	-8.9727	-8.6473	-8.5291	-9.1483
Forward-Backward Procedure					
P_{same}	-4.7359	-4.2125	-4.9843	-4.3908	-4.3876
$P_{different}$	-7.2714	-8.6137	-7.5914	-7.7634	-7.1007

Experimen	#6	#7	#8	#9	#10
Genetic Algorithm					
P_{same}	-3.3281	-4.1869	-4.2322	-4.3872	-3.1539
$P_{different}$	-7.5581	-7.6257	-8.6274	-8.7812	-8.3641
Forward-Backward Procedure					
P_{same}	-4.9811	-4.3481	-4.0567	-4.4860	-4.9251
$P_{different}$	-7.3825	-7.7351	-7.9328	-7.7514	-8.2254

Table 1. Experimental results : p_{same} and $p_{different}$.

As shown in Table 1, the HMMs trained by GA-HMM training have higher average log probabilities p_{same} than the HMMs trained by forward-backward procedure except experiment #1 and #8. This means the HMMs trained by GA-HMM training can be better describe the training observation sequences than the HMM models trained by forward-backward procedure. The exceptional cases of experiment #1 and #8 are due to the random search property of GA that the global maxima is not encountered in the 20000 generations. The average log probabilities $p_{different}$ of the HMMs trained by GA have lower values are because of the models are optimized to the training observation sequences but not the training observation sequences of other models.

7. CONCLUSION

The above results indicated that Genetic Algorithm has a higher probability in finding the global maxima or at least a local maxima with better performance to generate training observation sequences than forward-backward

algorithm. This is due to GA's probabilistic acceptance of a system change, GA can escape from local maxima to search other distant points to search possible global maxima.

Genetic Algorithms allow optimization of a 'black box' problem for which problem itself need not be analyzed, and the only requirement is to encode the chromosome properly and evaluates the fitness of the solution. Thus, it can optimized in a wide range of systems include the systems which can only be solved by heuristic algorithms. However, GA requires more computation efforts to mimic natural evolution and fitness evaluations. Special purpose hardware is one of solutions to improve searching time. Besides, parallel implementation of GA [6]-[8] practicable method to reduce searching time such that the searching time of GA can compare with heuristic algorithms.

8. REFERENCE

- [1] Lawrence Rabiner and Biing-Hwang Juang, *"Fundamentals of Speech Recognition"*, Prentic Hall International Editions.
- [2] L. E. Baum, T. Petrie, G. Soules and N. Weiss, "A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains", *Ann. Math. Stat.*, Vol 41, pp.164-171, 1970.
- [3] F. Jelinek, "Continuous speech recognition by statistical methods", *Proc. IEEE*, Vol. 64, No. 4, pp.532-556, April 1976.
- [4] S. E. Levinson, L. R. Rabiner and M. M. Sondhi, "An introduction to the Application of the Theory of Probabilistic Functions of a Markov Process to Automatic Speech Recognition", *The Bell System Technical Journal*, pp. 1035-1074, April 1983.
- [5] M. Srinivas and Lalit M. Patnaik, "Genetic Algorithms : A Survey", *IEEE Computer*, pp. 17-25, June 1994.
- [6] T. C. Fogarty and R. Huang., "Implementing the genetic algorithm on transputer based parallel processing systems", *Parallel Problem Solving from Nature*, pages 145-149. Spring-Verlag, 1991.
- [7] J. J. Grefenstette., "Parallel adaptive algorithms for function optimization", Technical Report CS-81-19, Nashville: Vanderbilt University. Computer Science Dept., 1981.
- [8] M. Gorges-Schleuter, "Asparagos: A asynchronous parallel genetic optimization strategy", *Proc. of the 3rd Int. Conf. on Genetic Algorithms and their Applications*, pages 422-427, 1989.