Genetic Algorithms Principles Towards Hidden Markov Model

Nabil M. Hewahi
Department of Computer Science
Islamic University of Gaza, Palestine (Palestinian Territories)
nhewahi@iugaza.edu.ps

Abstract

In this paper we propose a general approach based on Genetic Algorithms (GAs) to evolve Hidden Markov Models (HMM). The problem appears when experts assign probability values for HMM, they use only some limited inputs. The assigned probability values might not be accurate to serve in other cases related to the same domain. We introduce an approach based on GAs to find out the suitable probability values for the HMM to be mostly correct in more cases than what have been used to assign the probability values.

Keywords: Hidden Markov Model; Genetic Algorithms

1. Introduction

Hidden Markov Model (HMM) is a statistical model based on probabilities used in various applications such as cryptanalysis, machine translation, speech and hand recognition, natural language processing, gene prediction and bioinformatics [1][5][7-12]. A Markov model is a probabilistic process over a finite set, {S1, ..., Sk}, usually called its states. Each state-transition generates a character from the alphabet of the process. In a regular Markov model, the state is directly visible to the observer, and therefore the state transition probabilities are the only parameters. In a hidden Markov model, the state is not directly visible, but variables influenced by the state are visible. Each state has a probability distribution over the possible output tokens. Therefore, the sequence of tokens generated by a HMM gives some information about the sequence of states. There are three problems associated with HMM, evaluation, decoding, and learning problems. The evaluation problem, given the parameters of the model, compute the probability of a particular output sequence, and the probabilities of the hidden state values given that output sequence. The evaluation problem can be solved by forward-backward algorithm. The decoding problem, given the parameters of the model, find the most likely sequence of hidden states that could have generated a given output sequence. This is solved by Viterbi algorithm. The learning problem, given an output sequence or a set of such sequences, find the most likely set of state transition and output probabilities. This problem is solved by the Baum-Welch algorithm [11].

Hewahi [4] presented a modified version of Censored Production Rule (CPR) called Modified Censored Production Rules (MCPR). CPR is proposed by Michalski and Winston [6] to capture real time situations. MCPR can fit with hidden Markov model and present a scheme to compute the certainty values of the obtained conclusions out of the induced rules. To compute the certainty values for the rule actions (conclusions), the approach exploited only the probability values associated with the hidden Markov model without using any of the other well known certainty computation approaches. Hewahi [3] also proposed an intelligent networking management system based on the induced MCPRs extracted from a networking structure based on HMM. The advantage of using this technique is that MCPRs are very useful in real time applications and can be adapted over time based on the obtained experience of the networking working process.

Let us consider the HMM presented in Figure 1.

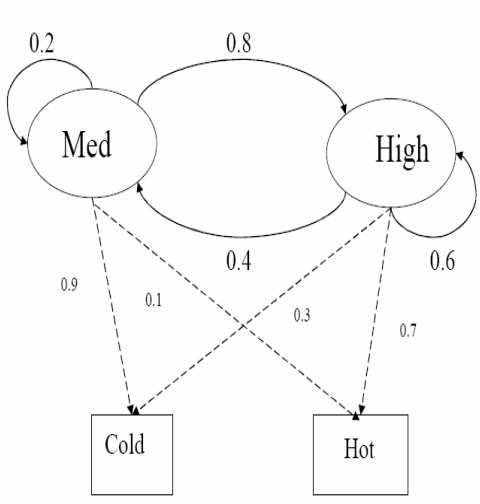

Figure 1. HMM to describe a relation between the states Med. and High with the observations (invisible states) cold and hot.

The summary of Figure 1 says:
P(cold | med) = 0.9, P(hot | med) = 0.1, P(cold | high) = 0.3, P(hot | high) = 0.7,
P(med |med) = 0.2, P(high | med) = 0.8, P(high | high) = 0.6, P(med | high) = 0.4

The HMM in Figure 1can be represented in a rule structure form as shown in [3].
If Med Then Cold 0.9
If Med Then Hot 0.1
If High then cold 0.3
If High then Hot 0.7
If Med then Med 0.2
If Med then High 0.8
If High then High 0.6
If High then Med 0.4

The initial probability values in HMM are usually assigned based on a survey on several related inputs. For example in general a commonly used related words could be car-type, country made and model. These words may appear with various probability values in several related texts, therefore a general probability based on all inputs are assigned to HMM. The question is that whether the adopted probability expresses the reality. To get the optimum probability values that might make HMM represents the real situation, we shall use the general approach of Genetic Algorithms (GAs). We choose GAs because they are very good algorithms in evolving new solution and are easy to implement. GAs are inspired from biology and based on natural genetics and survival of the fittest, it is used in searching, optimization, and machine learning. GA proved very high capability in many research areas and in many NP-problems. GAs use what so called genetic operators to modify or update the population (set of individuals/ chromosomes that represent the problem solutions). Some of the well known genetic operators are reproduction, crossover and mutation [2]. In this paper we shall propose a general guide line that will be of great benefit to discover the proper HMM probability values. This will be done by having several HMM within a population.

2. The proposed approach

Our main concern is to find somehow a mechanism that ensures the values assigned as probability values in HMM to the far extent are correct. To do this we follow up the following procedure:

a) Set the size of the population (initial population) to be N for example.
b) Divide your domain inputs into two partitions say each partition contains 10 chromosomes where each chromosome is representing one HMM. Each HMM is formed out of say 5 input text documents related to the problem domain. This means each partition needs 50 text documents related to the domain. One partition is used to represent the initial population where the second partition is used for fitness function evaluation for each HMM in the population and called evaluation population.. We give the first and second partition to an expert who is going to decide the values of probability values for the state relations for each HMM.
c) Formulize every HMM in both the partitions in a chromosome / individual form.
d) Compute fitness function for every chromosome in the initial population existing in the first partition.
e) Apply genetic operators on the population (in the first partition).
f) Get the offspring out of the previous step and compute the fitness function for each child in the offspring after formulating them to chromosomes.
g) Modify the population by removing very low fitness values chromosomes with chromosomes obtained from the crossover applied in step e, which have higher fitness values. The size of the population does not change.
h) Repeat steps from e to g until the best chromosome is obtained.

Figure 2 represents the general steps required for the proposed approach. We now clarify certain points such as chromosome formulation, computing fitness function and applying the genetic operators.

A. Formulating the Chromosome

Let us again consider Figure 1, we formulate the chromosome of the HMM given in Figure 1 as below:

Med-Med:02 Med-High:0.8 High-High:0.6 High-Med:0.4 Med-Cold:0.9
Med-Hot:0.1 High-Cold:0.3 High-Hot:0.7

The chromosome contains 8 genes, each is represented by the relation between two states accompanied with a probability value. The genes should be formed in this way because this is important in the crossover operation as to be explained later. The most important thing is that each two genes has the probability summation of 1.0. For example Med-Med:02 and Med-High:08 have the summation of 1.0. Similarly High-High:0.6 and High-Med:0.4 have the summation of 1.0. Each two genes with summation of 1.0 should be neighbors.

B. Genetic Operators

We use two main genetic operators in our approach, crossover and mutation.

1) Crossover

In this genetic operator, we choose two chromosomes at random and apply crossover between them. Figure 3 shows the proposed crossover. We choose a crossing cut site at random. It is to be noted that the crossing cut site should be even number. We should have two crossing cut sites. If we make crossing cut site odd number, the resultant child will not have a correct value of probability. The incorrect crossover is shown in Figure 4.

2) Mutation

Mutation is applied to a randomly chosen chromosome from the population. The proposed mutation is changing the probability value of a certain relation by increasing or decreasing it be a specific number. Here also a care while doing mutation should be followed. Changing the probability in one relation without changing the other related probability will spoil the values.

Figure 5 illustrates an example of mutation process. In Figure 5, Med-Cold:0.9 and Med-Hot:0.1 before mutation and become Med-Cold:0.7 and Med-Hot:0.3 after mutation. This is done by decreasing 0.2 from Med-Cold probability and adding 0.2 to Med-Hot probability.

3) Evaluating the HMMs

This is done by having a fitness function that can measure the strength of the chromosome. Our proposed function must be able to scale how much general can the HMM serve in its domain. To do this we follow the following steps:

a) Each HMM represented in a chromosome form and within the population has it fitness value computed by comparing its relation probability values with the one in the evaluation partition. This is done as below:

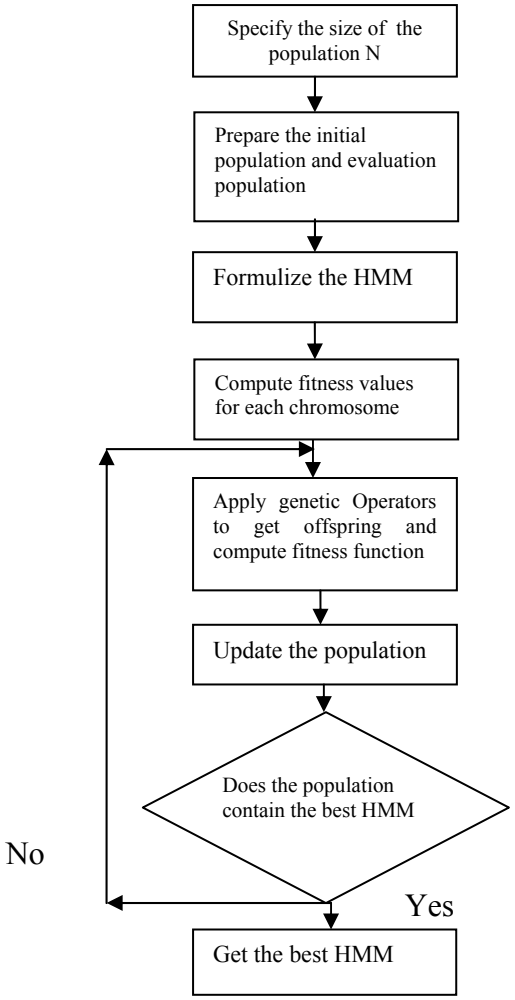

Figure 2. The general structure of the proposed approach.

Let us assume that the chromosome in the population as shown in Figure 6. Also let us consider that the other five chromosomes in the evaluation partition are as in Figure 6. The fitness value for the ith chromosome in the population is computed by the following formula:

$$\text{fitness}_i = 1 / \sum_{j=1}^{m} \text{compare}(i, j)$$

where m is the number of chromosomes in the evaluation partition and compare function is performed by computing the difference between the ith chromosome in the population (first partition) with each chromosome in the evaluation partition. This is done by comparing the

difference in the probability value for each related pair of relations. For example comparing the chromosome given in Figure 6 with the first chromosome in the evaluation partition, the difference between the relation Med-Med and Med-High as a pair is 0.0 and the difference between the relation High-High and High-Med as a pair is 0.1. Similarly the difference between the relation Med-Cold and Med-Hot as a pair is 0.1 and the difference between the relation High-Cold and High-Hot as a pair is 0.2. We sum all these differences to get the value of compare(i,j), the sum value is 0+0.1+0.1+0.2 = 0.4. Using the same approach we compute the compare function with the other four chromosomes and we get values 0.4, 0.5,0.4 and 0.6. Now we sum the five values 0.4 + 0.4 + 0.5+ 0.4 +0.6 = 2.3. The fitness value is then 1/ 2.3 = 0.434. The highest is the fitness value, the better is the performance of the chromosome.

b) Every n iterations choose some of the initial population at random and add it to the evaluation partition. This is important to make the new generated chromosomes not to forget the initial data formed the initial population specially after making changes on the chromosomes based on the genetic operators. This way, the finally selected chromosome will be the closest to represent the initial and evaluating partitions. The more chromosomes we have in the evaluation partition the better is the result.

3. Discussion

Our assumption in this research is that we can get data to be used to establish the population and evaluation partition. Such kind of systems can be adaptive overtime. If periodically the system collects a new evaluation partition (data) from the domain related data over the internet, the system can continuously improve its HMM to be useful for more cases within the domain. The only problem in this case is that every time the new obtained evaluation partition is gathered, an expert is needed to assign probability values to it. However, automatic mechanism can be adopted to assign the probability values.

4. Conclusion

In this paper we presented a general concept that can be followed to find out the optimum HMM to be used in serving a certain domain area. The proposed approach is based on GAs to evolve new HMM, the evolving means finding the appropriate probability values for the relations between the states and not evolving HMM structures. We proposed a carful mechanism for applying the genetic operators, crossover and mutation. The whole procedure is done because in various domain cases the constructed HMM based on an expert person can't cover the real situation especially if the expertise depend on a small set of inputs to assign the probability values between the relations. The proposed approach evaluates each HMM after formulating it in a chromosome form by calculating a fitness function which depends on a compare function that compares the HMM in the population with those existing in the evaluation partition. The highest is the fitness value of the chromosome; the better is its performance. To keep the chosen chromosome the optimum one, a portion from the initial population chosen at random is added to the evaluation partition. This is done to avoid any changes in the chromosomes (through the genetic operators) that might fall after a while to be adequate with the initial used data to build the initial population. Some of the future directions are 1. Developing and implementing this approach and apply it to various domains 2. apply GA to evolve complete HMM structures and not only evolving the probability values.

Chromosome 1: Med-Med:0.2 Med-High:0.8 High-High:0.6 High-Med:0.4 Med-Cold:0.9 Med-Hot:0.1 High-Cold:0.3 High-Hot:0.7

Chromosome 1: Med-Med:0.3 Med-High:0.7 High-High:0.8 High-Med:0.2 Med-Cold:0.4 Med-Hot:0.6 High-Cold:0.4 High-Hot:0.6

After Crossover

Child 1: Med-Med:0.2 Med-High:0.8 High-High:0.8 High-Med:.0.2 Med-Cold:0.9 Med-Hot:0.1 High-Cold:0.3 High-Hot:0.7

Child 2 : Med-Med:0.2 Med-High:0.7 High-High:0.6 High-Med:0.4 Med-Cold:0.4 Med-Hot:0.6 High-Cold:0.4 High-Hot:0.6

Figure 3. The crossover operation between two HMM chromosomes

Chromosome 1: Med-Med:0.2 Med-High:0.8 High-High:0.6 High-Med:0.4 Med-Cold:0.9 Med-Hot:0.1 High-Cold:0.3 High-Hot:0.7

Chromosome 1: Med-Med:0.3 Med-High:0.7 High-High:0.8 High-Med:0.2 Med-Cold:0.4 Med-Hot:0.6 High-Cold:0.4 High-Hot:0.6

After Crossover

Child 1: Med-Med:0.2 Med-High:0.8 High-High:0.8 High-Med:0.4 Med-Cold:0.9 Med-Hot:0.1 High-Cold:0.3 High-Hot:0.7

Child 2 : Med-Med:0.2 Med-High:0.8 High-High:0.6 High-Med:0.2 Med-Cold:0.9 Med-Hot:0.1 High-Cold:0.3 High-Hot:0.7

Figure 4. Incorrect crossover operation. The High-High and High-Med probability values summation should be 1.

Chromosome : Med-Med:0.2 Med-High:0.8 High-High:0.6 High-Med:0.4 **Med-Cold:0.9 Med-Hot:0.1** High-Cold:0.3 High-Hot:0.7

After mutation

Med-Med:0.2 Med-High:0.8 High-High:0.6 High-Med:0.4 **Med-Cold:0.7 Med-Hot:0.3** High-Cold:0.3 High-Hot:0.7

Figure 5. Mutation process. This is happened by decreasing 0.2 from Med-Cold probability and adding 0.2 to Med-Hot.

The chromosome from the population:

Med-Med:0.2 Med-High:0.8 High-High:0.6 High-Med:0.4 Med-Cold:0.9 Med-Hot:0.1 High-Cold:0.3 High-Hot:0.7

The five chromosome existing in the evaluation partition:

Med-Med:0.2 Med-High:0.8 High-High:0.5 High-Med:0.5 Med-Cold:0.8 Med-Hot:0.2 High-Cold:0.5 High-Hot:0.5

Med-Med:0.1 Med-High:0.9 High-High:0.5 High-Med:0.5Med-Cold:0.7 Med-Hot:0.3 High-Cold:0.3 High-Hot:0.7

Med-Med:0.3 Med-High:0.7 High-High:0.4 High-Med:0.6 Med-Cold:0.6 Med-Hot:0.4 High-Cold:0.5 High-Hot:0.5

Med-Med:0.3 Med-High:0.7 High-High:0.4 High-Med:0.6 Med-Cold:0.9 Med-Hot:0.1 High-Cold:0.4 High-Hot:0.6

Med-Med:0.2 Med-High:0.8 High-High:0.5 High-Med:0.5 Med-Cold:0.5 Med-Hot:0.5 High-Cold:0.4 High-Hot:0.6

The Fitness value of the chromosome in the population will be 0.434

Figure 6. One chromosome from the population and the five chromosomes existing in the evaluation partition.

References

[1] L. Allison, L. Stern, T. Edgoose & T. I. Dix," Sequence complexity for biological sequence analysis", Computers and Chemistry 24(1), pp.43-55, Jan. 2000

[2] D. Goldenberg, " Genetic Algorithms in Search, Optimization, and Machine Learning", Addison-Wesley Longman Publishing, 1989.

[3] N. Hewahi, "An Intelligent Approach For Network Management Based on Hidden Markov Model", Proceedings of the International Arab Conference for IT, ACIT'10, Libya, Dec.14-16, 2010.

[4] N. Hewahi, "Hidden Markov Model for Censored Production Rules", Proceedings of the International Conference of Information Technology, ICIT'09, Jordan, May 3-5, 2009.

[5] J. Li, A. Najmi, R. M. Gray, Image classification by a two dimensional hidden Markov model, IEEE Transactions on Signal Processing, 48(2),pp. 517-33, February 2000.

[6] R. Michalski, P. Winston, "Variable precision logic", Artificial Intelligence, 29, pp. 121-145, 1986.

[7] Lior Pachter and Bernd Sturmfels. "Algebraic Statistics for Computational Biology". Cambridge University Press, ISBN 0-521-85700-7, 2005.

[8] B. Pardo and W. Birmingham "Modeling form for on-line following of musical performances", AAAI-05 Proc., July 2005.

[9] L. Rabiner, "A tutorial on hidden Markov model and selected applications on speech recognition", Proceedings of IEEE, vol.77, no.2, 1989.

[10] K. Seymore, A. McCallum, and R. Rosenfeld, "Learning hidden Markov model structure for information extraction", AAAI 99 Workshop on Machine Learning for Information Extraction, 1999.

[11] L. Stern, L. Allison, R. Coppel, and T. Dix, "Discovering patterns in Plasmodium falciparum genomic DNA", Molecular and Biochemical Parasitology, **118**(2) pp.175-186, 2001.

[12] http://en.wikipedia.org/wiki/Markov-model