Parameter Estimation of Hidden Markov Models (HMM) using Go With the Winner Algorithms

Ravindra Nath
Department of Computer Science & Engineering
UIET, CSJM University, Kanpur-208024,

Dr. Renu Jain
Department of Computer Science & Engineering
UIET, CSJM University, Kanpur-208024, India

ABSTRACT
Hidden Markov model (HMM) is a stochastic method which has been used in various application like speech processing, signal processing and character recognition. It has three main problems. Third problem of HMM is the one in which we optimize the model parameters so as to describe how a given observation sequence comes about. The observation sequence is used to adjust the model parameters is called training sequence since it is used to train the HMM. One of the conventional methods that are applied in setting HMM model parameters values is Baum Welch algorithm. So in this paper Go With the Winner (GWW) method is used to train the HMM Parameters. We have already done experiment of same set of data using Baum Welch, Metropolis, Simulated Annealing and Genetic algorithm. The experimental results show that GWW is found to reach maxima in less number of transactions and the value of P(O|λ) is also much higher in comparison to Metropolis, Simulated Annealing and Genetic algorithm.

Keywords: Hidden Markov Models (HMM); Go With the Winner (GWW), Genetic Algorithm(GA), Baum-Welch method (BW).

1. INTRODUCTION
Some random search methods can be used to estimate HMM parameters. In this paper, four random search techniques are used and the performance of these method Compared with Go with the winner algorithm. These methods are Metropolis, Simulated Annealing,, Genetic Algorithm and one of the traditional method ie. Baum Welch algorithm. These algorithms are used to estimate HMM parameters. The estimation of good model parameters affects the performance to search global maxima or minima so that values of these parameters are needed to be altered and the value of P(O| λ) estimated, such that the value of P(O| λ) get maximum in less number of state transactions. Hidden Markov Models (HMM) have many applications in signal processing, pattern recognition, and speech recognition[1] etc. Basic component of HMM is considered in speech recognition systems.

2. HIDDEN MARKOV MODELS (HMM)
HMM are probabilistic models useful for modeling stochastic sequence with underlying finite state structure. Stochastic sequences are called observation sequences $O = o_1\ o_2$ ……o_T, where T is the length of the sequence. HMM with n states (S_1,S_2 …. S_n) can be characterized by a set of parameters: $\lambda = \{A, B, \pi\}$ where π is the initial distribution probability that describes the probability distribution of the observation symbol in the initial moment, and $\sum_{i=1}^{n} \pi_i = 1$ and π_i >=0.A is the state transition probability matrix $\{a_{ij} \mid i,j=1,2,3 \ldots n\}$, a_{ij} is the probability of transition from state i to state j, and $\sum_{j=1}^{n} a_{ij} = 1$ and a_{ij} >=0.

B is the observation matrix { b_{ik} | i =1,2,3 ……n, k=1,2……m} where n is the number of the states and m is the number of observation symbols. $\sum_{k=1}^{m} b_{ik} = 1$, b_{ik} >=0, b_{ik} is the probability of observation symbol with index k emitted by the current state i.

The main problems of HMM are: **evaluation**, **decoding**, and **Learning** problems.

Evaluation problem

Given the HMM $\lambda = \{A, B, \pi\}$ and the observation sequence O=o1 o2 ... oT, the probability that model λ has generated sequence O is calculated.

Often this problem is solved by The Forward Backward Algorithm (Rabiner,1989) (Rabiner,1993).

Decoding problem

Given the HMM $\lambda = \{A, B, \pi\}$ and the observation sequence O=o1 o2 ... oT, calculate the most likely sequence of hidden states that produced this observation sequence O.

Learning problem

Given some training observation sequences O = o1 o2 ... oT and general structure of HMM (numbers of hidden and visible states), determine HMM parameters $\lambda = \{\pi, A, B\}$ that best fit training data.

The most common solution for this problem is Baum-Welch algorithm [1] which is considered the traditional method for training HMM.

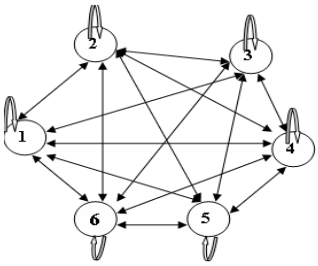

Figure 1: Six states HMM models

In this paper, a six state HMM model is used as shown in Figure 1. This figure shows that states are six and we can move from one state to another including the same state. Here if we are consider the example of six dies each state can emits six symbols are number (1,2,3,4,5,6). So The transition matrix *A* is a 6 x 6 matrix and the observation matrix *B* is of size 6 x 6. According to this configuration (HMM model).

3. HMM TRAINING BY GWW ALGORITHM

3.1 Data training

Every set of A, B, x **is** modeled by the forward-backward algorithm or baum-welch algorithm. this algorithms is used to find maxima using a data set of casino, calculated drom the observation sequence O and HMM $\lambda = \{A, B, \pi\}$.

Observation sequence $O = O_1\ O_2\ \ldots\ldots\ldots O_t$ is denoted as state sequence probability, is defined as forward variable. state s_i at time t with λ and α as t(i) as defined in equation no(1).this sequence state probability is denoted as:

$$\mathbf{P}(O|\ \lambda) = \sum_{i=1}^{N} \alpha_t(i)\beta_t(i) = \sum_{i=1}^{N} \alpha_t(i) \quad \text{----------eq-1}$$

3.2 GO WITH WINNER ALGORITHM (GWW)

Algorithms 0 : Start at root, repeatedly choose a child at random until reaching a leaf, then stop[12].

Algorithms 1 : "Go with the winners" Repeat the following procedure, starting at stage 0 with R particles (solution ie value of P(O| λ) according to HMM) at the root. At stage i each of the R particles is at some vertex at depth i. if all the particles are at leaves, then stop. Otherwise some particles $j_1 \ldots\ldots j_m$ are at non leaves: spread the remaining R-m particles evenly among the position each of which gets assigned an extra particles moves from its current vertex to a child chosen at random.

Dataset Description and formulation to implement Go With the Winner Algorithms

We have assumed a Casino problem and modeled it as discrete HMM learning problem. We assume that in a casino there are six dies (N = 6), some of which may be loaded dies, each die consists of six faces (M = 6). Output in each state is the symbol from the set {1, 2, 3…..6}. It can switch from a one die to another die (dies can loaded or fair) or to same die with some probability a_{ij}. Switch between dies is a Hidden Markov process. Each state of Markov process is an example of a HMM. In above scenario the values of HMM parameters will be as follows:

N = 6 (denotes the number of dies taken)
M = 6 (denote the number of faces of a die)
T = 50 length of observation sequence i.e. dies have been thrown 50 times. Experiments perform for 10 observation sequences.
We assume initial probability distribution as follows:
π[N] = {(1.0/6.0), (1.0/6.0), (1.0/6.0), (1.0/6.0), (1.0/6.0), (1.0/6.0)}
Initial probability of each die is assumed to be same.

By taking different probability distributions, different sets for symbol emitting probabilities and state transition probabilities are arranged. For example:
Set I:

A[N][N]=(1/6),(1/6),(1/6),(1/6),(1/6),(1/6),(1/6),(1/6),(1/6),(1/6),(1/6),(1/6),(1/6),(1/12),(3/6),(1/12),(1/12),(1/12),(1/6),(1/6),(1/6),(1/6),(1/6),(1/6),(1/6),(1/3),(1/12),(1/6),(1/6),(1/6),(1/6),(1/6),(1/6),(1/6),(1/6),(1/6),(1/6),(1/6)

B[N][M]=(1/6),(1/6)

(a) According to GWW algorithm[13] at initial we choose the set A (state transition probabilities) and set B (symbol emitting probabilities) and set the leaf value and generate R solutions (ie. Value of P(O| λ)) at root, separate (m) leaves and (R-m) non leaves values.
(b) algorithm applied in two ways,

(i) Sorted all non leaf solution in descending order and choose first k=20 values for again generating L= 5 leaves for each solution.
(ii) Randomly choose first K=20 values for repeatedly generating L=5 leaves for each solution.

(c) set new leave values of P(O| λ) by increasing 1000 in each iteration.
(d) and check that if all solutions are at leave value then stop.
(e) Otherwise repeat step (b) to (d) until get maximum value of P(O| λ).
The Description of algorithm using Dataset is as follows.
(a)

A[N][N]=(1/6),(1/6),(1/6),(1/6),(1/6),(1/6),(1/6),(1/6),(1/6),(1/6),(1/6),(1/6),(1/6),(1/12),(3/6),(1/12),(1/12),(1/12),(1/6),(1/6),(1/6),(1/6),(1/6),(1/6),(1/6),(1/3),(1/12),(1/6),(1/6),(1/6),(1/6),(1/6),(1/6),(1/6),(1/6),(1/6),(1/6),(1/6)

B[N][M]= (1/6), (1/6)

We assume the First Leaf value = 1.000000e-34
(b)
(i) Sorted non leaves values

0	9.621437e-33	NON LEAVES
1	5.791460e-33	NON LEAVES
2	5.439813e-33	NON LEAVES
3	3.847358e-33	NON LEAVES
4	3.833270e-33	NON LEAVES
5	2.515955e-33	NON LEAVES
6	2.060615e-33	NON LEAVES
7	1.835383e-33	NON LEAVES
8	1.740814e-33	NON LEAVES
9	1.510107e-33	NON LEAVES
10	1.460473e-33	NON LEAVES ……. So on

(b)
(ii) unsorted non leaves values.

1	2.064851e-34	NON LEAVES
2	5.035246e-34	NON LEAVES
3	4.774281e-34	NON LEAVES
4	2.375590e-34	NON LEAVES
5	1.306842e-34	NON LEAVES
6	3.705399e-34	NON LEAVES
7	5.636091e-33	NON LEAVES
8	3.705399e-34	NON LEAVES
9	5.636091e-33	NON LEAVES
10	1.533966e-34	NON LEAVES ….. so on

(c) Set the new leaf value by multiplying 1000 to previous value.
New leaves value=1.000000e-31
(d) check that all new generated value are leaf then stop

1	9.648747e-37	LEAVES
2	3.426626e-35	LEAVES
3	1.352878e-35	LEAVES
4	5.043870e-29	NON LEAVES
5	2.722681e-29	NON LEAVES

Process will continue until all the non leaves values will be leaf value.

4. RESULTS AND DISCUSSIONS

In this paper, we have applied GWW algorithm to the casino problem for solving HMM learning problem. We implemented it for 10 different observation sequences. For each observation sequence, a solution of 100 solutions was taken, where each non leaf solution is generated entries (keeping initial probability same) denoting state transition probabilities and symbol emission probabilities. After setting new leaf value new solutions are generated from previous non leaf solution. For each solution of this population, the value of $P(O|\lambda)$ is calculated. A population of 500 solutions is chosen randomly from this new population and the rest of the solutions are discarded. The same process is continued until all solutions are leaves. For each iteration, the highest value of $P(O|\lambda)$ is recorded. For all the ten observation sequences given below, the GWW algorithm was applied and very encouraging results were found.

For each observation sequence, it was observed that the value of highest $P(O|\lambda)$ calculated by GWW is much higher in comparison to the value of highest $P(O|\lambda)$ calculated using GWW algorithm Simulated Annealing and Metropolis algorithms. Figure 1, Figure2 and Figure 3 show comparison graphs for three observation sequences. It can been seen from the graphs that in case of GWW algorithm, Simulated annealing and Metropolis algorithms, the value of $P(O|\lambda)$ oscillates within a very small range but the values obtained from GWW cover a wider range. It can be concluded from the results that other algorithms are able to explore a very small part of a large search space but, GWW algorithm is able to explore a long range of search space and hence giving a much higher value for local maxima. It may be because GWW imbibes many levels of randomization while performing Different level of GWW algorithm.

Observation No.	***Sequence Number***
1	*{3,2,6,5,1,2,4,5,6,3,6,6,6,4,6,3,1,6,1,6,6,6,3,1,6,2,3,2,6,4,5,5,2,3,6,2,6,6,6,6,6,6,2,5,1,5,1,6,3,1}*
2	*{3,6,6,1,6,3,6,6,6,4,6,6,2,3,2,5,3,4,4,1,3,6,6,1,6,6,1,1,6,3,2,5,2,5,6,2,4,6,2,2,5,5,2,6,5,2,5,2,2,6}*
3	*{3,3,3,3,3,3,3,5,5,5,5,5,5,5,6,6,6,6,6,6,6,6,3,3,3,3,3,3,2,2,2,2,2,2,2,2,6,6,6,6,6,6,2,2,2,2,2,2,2,1}*
4	*{2,2,2,5,5,5,4,4,1,6,6,6,5,6,6,5,6,3,5,6,4,3,2,4,3,6,4,1,3,1,5,1,3,4,6,5,1,4,6,3,6,5,3,4,1,1,1,2,6,4}*
5	*{6,5,1,1,6,6,4,5,3,1,3,2,6,5,1,2,4,5,6,3,6,6,6,4,6,1,6,3,6,6,6,3,1,6,2,3,2,6,4,5,5,2,3,2,6,4,5,5,2,3}*
6	*{1,5,4,6,3,2,2,4,4,3,1,3,2,2,5,3,3,2,6,5,4,1,2,5,3,5,4,1,2,3,3,2,3,4,5,5,3,2,4,2,3,6,3,6,3,4,1,4,2,6}*
7	*{5,4,2,1,4,4,1,5,6,1,3,2,2,5,1,2,1,3,5,1,1,1,6,4,5,4,3,3,3,6,2,3,2,6,6,3,1,6,5,4,1,2,6,1,3,2,2,2,5,1}*
8	*{6,4,5,4,6,5,3,2,3,3,3,4,3,6,5,6,4,3,6,6,4,4,6,3,6,4,4,5,1,3,2,1,2,6,3,6,3,3,5,2,5,6,5,2,1,6,5,5,2,4}*
9	*{5,4,6,4,4,4,1,2,1,2,5,4,2,2,6,2,1,6,1,6,1,1,4,2,6,1,6,3,4,6,6,2,1,3,6,2,4,3,3,4,3,5,6,4,5,2,5,3,3,5}*
10	*{2,6,2,2,6,5,2,6,2,5,1,3,1,4,1,5,4,2,2,1,5,4,6,6,6,1,4,5,1,2,3,3,4,3,6,2,5,6,3,4,4,3,6,2,5,3,3,6,6,3}*

Table: 1

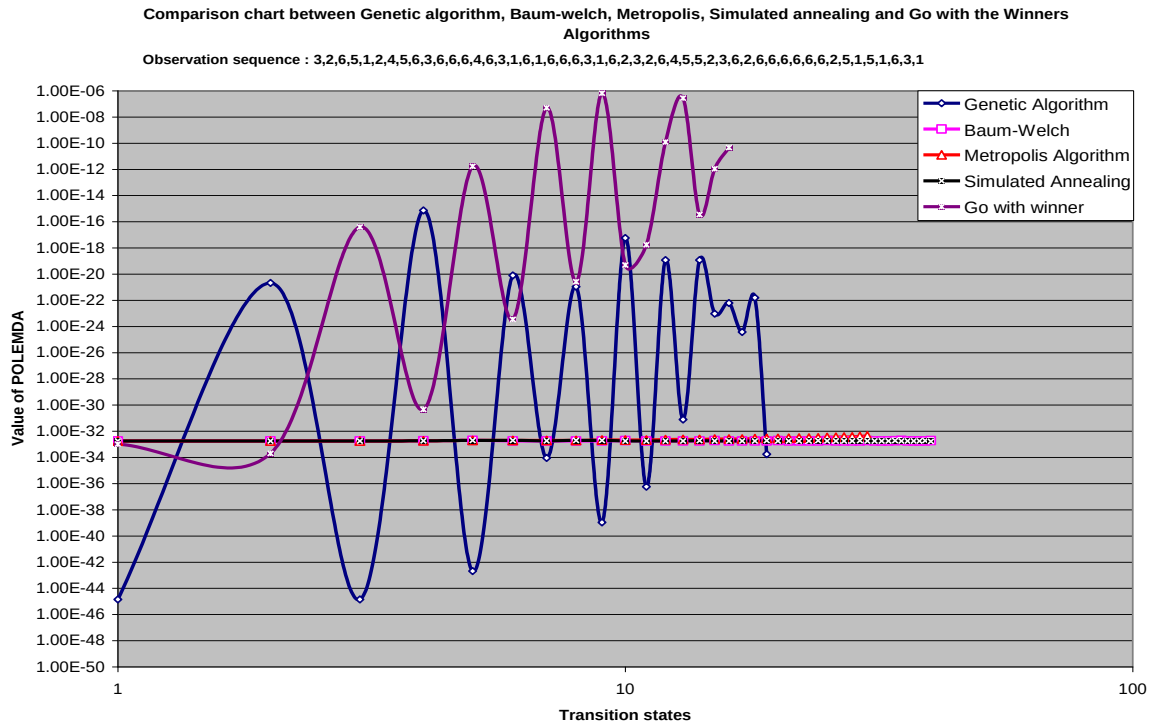

Figure 2: Comparison graph for the observation sequence1

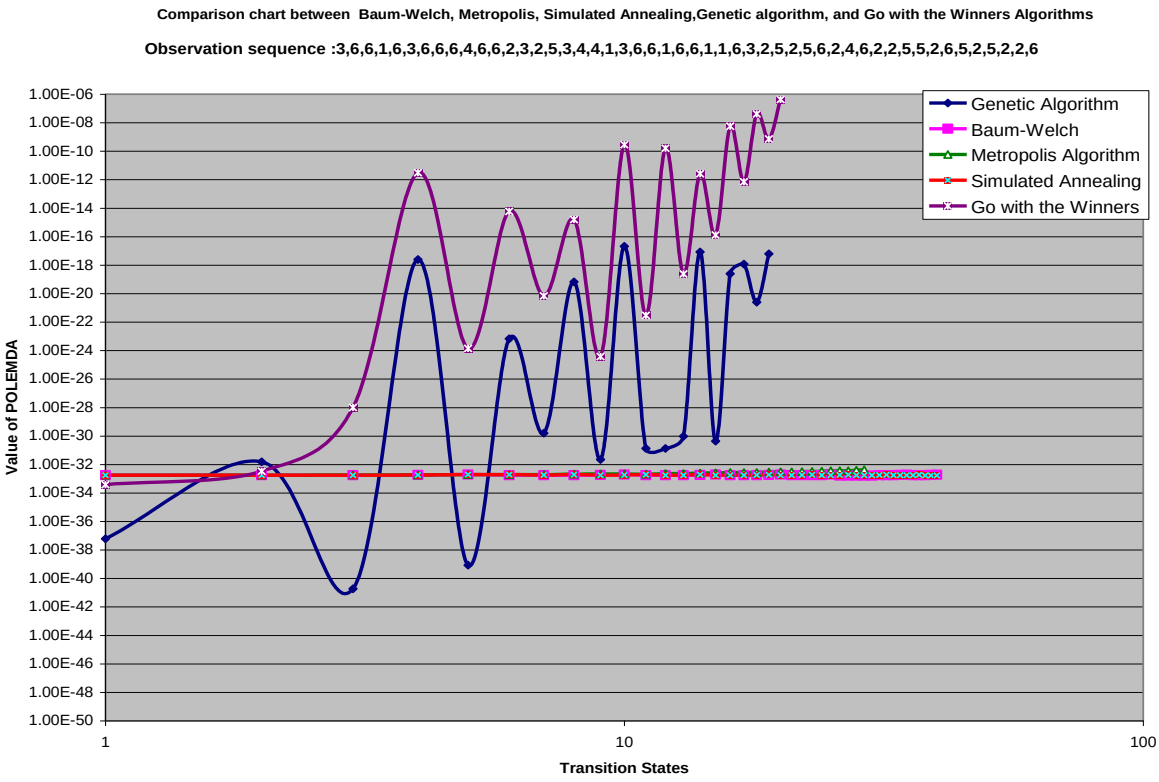

Figure 3: Comparison graph for the observation sequence2

In both the graphs shown above, X-axis represents the number of new generated model and Y-axis represents the value of P (O | λ). We observe that the value of P (O | λ) varies for each new generated model. It can be observed from the experimental data that, for new model is generated by GWW algorithm to achieve a local maxima/global maxima in comparison to different solution used. Table1 shows how the new model and corresponding P (O | λ) values are generated for each algorithm.

This table represents three observation sequences and maximum value of P(O | λ) for a particular observation sequence. As we have seen that in case of comparison of different algorithm as we have done in the previous papers, maximum value of P(O | λ) is 1.43 E-13 in the, 4.91E-10 and

1.50E-08 in the algorithm GWW and so on. And we also observe that max value of $P(O \mid \lambda)$ get in GWW algorithm.

S.No.	Baum-Welch	Metropolis	Simulated Anneling	Genetic Algorithms	Go With the Winner	Observation Sequence
1	1.79E-33	1.79E-33	1.79E-33	1.40E-20	1.43E-13	3,6,6,1,6,3,6,6,6,4,6,6,2,3,2,5,3,4,4,1,3,6,6,1,6,6,1,1,6,3,2,5,2,5,6,2,4,6,2,2,5,,5,2,6,5,2,5,2,2,6
2	1.79E-33	1.79E-33	1.79E-33	7.55E-15	4.91E-10	3,2,6,5,1,2,4,5,6,3,6,6,6,4,6,3,1,6,1,6,6,6,3,1,6,2,3,2,6,4,5,5,2,3,6,2,6,6,6,6,6,6,2,5,1,5,1,6,3,1
3	1.80E-33	1.80E-33	1.80E-33	2.73E-22	1.50E-08	3,3,3,3,3,3,3,5,5,5,5,5,5,5,6,6,6,6,6,6,6,6,3,3,3,3,3,3,2,2,2,2,2,2,2,2,6,6,6,6,6,6,2,2,2,2,2,2,2,1

Table 2

5. CONCLUSION

In this paper, we have modelled the HMM learning problem as a graph searching problem and tried to solve it using Go with the winner algorithms. We have implemented Go with the winner taking a fictitious data set for a casino. We have done the analysis for 10 observation sequences of length 50. It was found that for same observation sequences, the maxima obtained through Go with the winner is much higher than the maxima obtained from other randomized algorithms and the number of iterations required are also quite less in number.

It can be concluded from the experimental results that there is a high probability of getting better solutions of HMM learning problem using GWW algorithm. We are in the process of implementing the Gww algorithm for other HMM learning problems also. We are also working towards the implementation of other randomized search algorithms and compare the results with them.

6. ACKNOWLEDGMENT

The authors of the paper are highly grateful to Professor Somenath Biswas, Department of Computer Science and Engineering, IIT Kanpur, for motivating us to work in this interesting field of randomized algorithms and giving us direction to model the HMM learning problem as a discrete optimization problem and use randomized search algorithms to solve it.

7. REFERENCES

[1] Rabiner L.R. "A tutorial on HMM and Selected Applications in Speech Recognition", Proceedings of the IEEE, Vol. 77, NO. 2, P267-296.

[2] I. Wegener, Randomized Search Heuristics as an Alternative to Exact Optimization, Technical report , University of Dortmund, Dept of the Computer Science, February 2004.

[3] J. Kleinberg and E. Tardos, Algorithm Design , Cornell University Spring 2004.

[4] R. Durbin, R. Eddy and A. Krogh , Graeme Mitchison , Biological Sequence Analysis. Cambridge University Press 2005.

[5] Muthaiah Venkatachalam, Ismail Syed, "Web Site Optimization Through Web Log Analysis".

[6] Morteza ShahRam, "Image Processing and Reconstruction HMM-Based Pattern Detection" Project Report for EE 262: 2002.

[7] The Metropolis Algorithm, "Statistical Systems and Simulated Annealing".

[8] E. Rich & K.Knight "Artificial Intelligence" TMH Edition 1991.

[9] C. M. Coleman "Investigation of Simulated Annealing, Ant-Colony Optimization, and genetic Algorithms for self – Structuring Antennas" Vol.52 No.4, April 2004.

[10] Ravindra Nath, and Renu Jain "Using Randomized Search Algorithms to Estimate HMM Learning Parameters" IEEE International Advanced computing Conference (IACC-2009).

[11] A.H. Mantawy, L. Abdul Mazid, Z. Selim "Integrating genetic Algorithms, Tabu search and Simulated Annealing for the unit commitment problem", IEEE Transaction on power system, Vol.14, No. 3 August 1999.

[12] Mark Pirlot, "general local search method", European journal of operational research -92 (1996) 493-511.

[13] David Aldous, Umesh Vazirani,"Go With the Winner Algorithms".

[14] Tassos Dimitriou, Russell Impagliazzo," Towards an analysis of local optimization algorithms"

[15] Anastasia Rita Widiarti and Phalita Nari wastu "Javanese

[16] Character Recognition using HMM ", word Academy of science, Engineering and Technology 57 2009