State Complexity of Hidden Markov Model

Jamil Ahmed
Department of Computer Science
University of Western Ontario
London, Ontario, Canada N6A 5B7
jahmed6@uwo.ca

Stephen M. Watt
Department of Computer Science
University of Western Ontario
London, Ontario, Canada N6A 5B7
watt@uwo.ca

Sherjil Ahmed
Department of Software Engineering
Venture Chest
Shahra-e-Faisal, Karachi, Pakistan
sherjil.ahmed@venturechest.com

***Abstract*—A Classification problem can be viewed as a problem of assigning a category or class to a given input. The Hidden Markov model is a well known stochastic model that is used to solve classification problems and has been widely exploited in diverse computing applications ranging from speech, acoustics, gesture recognition to part-of-speech tagging, cryptography to Google page rank and the list goes on. State Complexity of Deterministic Finite automata is now a well established research area. State complexity of Deterministic Finite Automata defines the total number of states in the minimal Deterministic Finite Automata. State complexity, if known, of a given automata helps to realize how expensive the application would be that will exploit that automata. Similarly, if known, the state complexity of the Hidden Markov Model will help to know the complexity of a computing application that exploits that Hidden Markov Model. In this paper, we have explored several, yet unpublished, important facts about the Hidden Markov Model including the state complexity of Hidden Markov Model and the diagram of 2nd order Hidden Markov Model (Fig. 3). Our discussion of the Hidden Markov Model is unique in the sense that we present a complete diagram of the 1st order Hidden Markov Model (Fig. 2) with all estimated parameters for a given input sequence. We explicitly define a generalized rule to give "Dimension of Transition probability matrix of HMM" which is also not available in the literature yet. We present a generalized rule to draw the Mth order Hidden Markov Model diagram for M greater than 1. We present the generalized state complexity of the Mth Order HMM, the state complexity of the diagram for the "Training of Mth Order HMM" and also present the diagram for second order Hidden Markov Model.**

Keywords – State complexity; Hidden Markov Model.

I. INTRODUCTION

The Hidden Markov Model (HMM) has states, input symbols and transitions much like a deterministic finite automata (DFA). Transitions between states of HMM are labeled with the probabilities, unlike DFA, defining how likely that transition is to take place. States of HMM actually represent the classes or categories that we intend to assign to the symbols of input sequence. How it works is that first we define the classes or categories that will be represented by the states of HMM. In order to assign categories or states to input symbols, we process the input sequence on the HMM. The sequence of states assigned to as input sequence is called the output sequence. Many recent techniques exploit the Hidden Markov Model as in [10] [11][12].

Section II defines the HMM as well as the related terminology. State Complexity of Deterministic Finite automata is now a well established research area [8]. In Section III, we define the state complexity of the 1st order HMM. In Section IV, we define the rule for drawing the Mth order HMM diagram for M>1. In Section V, we define the transition probability matrix dimensions of Mth-Order HMM. In Section VI, we define the state complexity of the Mth-Order HMM for M>1. In Section VII, we define the 2nd order HMM Example. In Section VIII, the diagram of the 2nd Order HMM is presented. Section IX concludes the paper.

II. HIDDEN MARKOV MODEL

We present an overview and terminologies for the HMM.

A. *Formal Definition*

A HMM is a tuple (S, ∑, ∏, A,B)

- A set of states: $S=\{S_1,S_2,\dots S_m\}$
- A set of input symbols: $\sum = \{O_1,O_2\dots .O_k\}$
- Initial state probability: $\prod = \sum_{i=1}^{m} \prod = 1$
- Transition probability matrix: $A=\{a_{ij}\}$
- Emission probability matrix: $B=\{b_{ij}\}$

In order to show the working of the HMM, usually, we add two additional states "Start" and "End" states. Dimensions of the transition probability matrix and emission probability matrix, above, are for the 1st order HMM and are contingent to the order of the HMM. Transition probability is termed as "$P(t_i \mid t_{i-1})$" as mentioned in Section V.

B. *Training of HMM*

Processing of the input sequence on a given HMM develops a HMM training diagram, which we call "HMM with all estimated parameters". Those parameters include transition/emission probabilities. One such but partial HMM training diagram is given in [1]. "HMM with all estimated parameters" is also called "training of HMM" [9]. This "training of HMM" gives all possible sequences of states or categories which can be assigned to the input sequence. All these possible sequences of states or categories are also called hidden states sequence because these sequences are not known unless we train the original HMM for a given input sequence. Each hidden state sequence of states or categories assigned to the input sequence is also called the output sequence. Brute Force expansion of the HMM is usually intractable for most real world classification problems, as the number of possible hidden state sequences is extremely high and scales exponentially with the length of input sequence.

ISBN: 978-1-61208-202-8

All output sequences that are assigned to a given input sequence have probability of likelihood. This probability defines how likely the output sequence is as an appropriate assignment for the input sequence. These probabilities of likelihood are calculated with the help of transition probabilities and emission probabilities of the original HMM. For a given input sequence, HMM chooses the state sequence that maximizes in the following formula.

P(input symbol/state) * P(state/previous 'M' states)

In the above formula, Transition probability = P(state/previous 'M' states) and Emission probability = P(input symbol/state) . Transition probabilities (TPs) of each state of HMM either depend on one previous state or more than one previous state. If they depend on one previous state (i.e., M=1), this HMM is called a 1^{st} order HMM. If all TPs of states of HMM depend on two previous states (i.e., M=2), this HMM is called a 2^{nd} order HMM and so on.

C. 1^{st} Order HMM Example

Below, we have given an example HMM of two states, cold and hot. Major components of the HMM are also mentioned in the section ahead as well as the dimensions of the transition probability matrix.

We simulate a real world phenomena in Fig. 1 related with the 'weather of a day'. We suppose we can have either a Hot or Cold day. We know the probability (transition probability) of the next day is either cold or hot depending upon the weather of the previous day. This 1^{st} Order HMM represents how many ice creams servings, (e.g., 8, 7, 6 etc), a person is likely (emission probability) to eat on a given day depending upon the "weather of that day". Further we show the training diagram of the 1^{st} order HMM diagram in Section F.

We can see in Fig. 1 that each state of 1^{st} Order HMM directly corresponds to one single category, e.g., Cold or Hot in this example. This is also shown in [4].

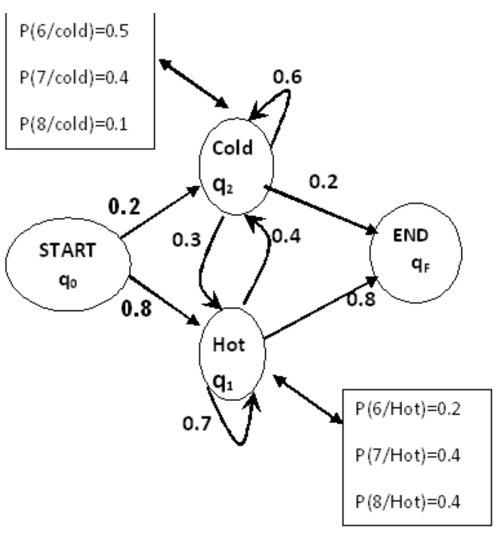

Figure 1. 1^{st} Order Hidden Markov Model

D. Components of our Example HMM

Components of our HMM given in the previous Section are shown in the tabular form below.

TABLE I. Components of Hidden Markov Model

Q-Set of states	Q= {Hot, Cold} ={q_1,q_2}
Transition probability Matrix [2-dimensional 2*2]	a_{11} = 0.7 , a_{12}=0.3, a_{21}= 0.4 , a_{22}= 0.6
Vocabulary of Inputs	V= {6,7,8,9}
Input Observation under consideration for training of this HMM	(8 7 6), i.e., O_1 =8 ,O_2= 7, O_3=6
Emission Probability: $b_1(6)$=P[6/Hot],$b_1(7)$=P[7/Hot], $b_1(8)$=P[8/Hot],$b_2(6)$=P[6/Cold], $b_2(7)$=P[7/Cold],$b_2(8)$=P[8/ Cold]	$b_1(6)$=0.2 , $b_1(7)$=0.4, $b_1(8)$=0.4,$b_2(6)$=0.5, $b_2(7)$=0.4, $b_2(8)$=0.1
Start state and End state	q_0 , q_F
Transition probability from start state to q_1 and q_2	a_{o1} = 0.8 , a_{02}=0.2
Transition probability to End state from q_1 and q_2	a_{1F} = 0.8 , a_{2F}=0.2

E. Dimension of Transition probability matrix of 1^{st} Order HMM

Dimensions of the transition probability matrix are based on the simple principle, i.e., Transition probability matrix shows probabilities of all the transitions from each state to all other states as well as to itself.

If 'S' is the number of states and 'M' is the order of HMM then the Transition probability matrix of the HMM is 'M+1' dimensional such as S^i *….* S^{M+1}. The superscript on 'S' represents the dimensions, i.e., how many times 'S' should appear.

In case S=3 and M=1, i.e., the 1^{st} order HMM then the transition probability matrix is 2 dimensional, i.e., 3*3. So, if S=3, there are three states say Q={q_1,q_2,q_3}, the two dimensional (3*3) matrix will be as follows:

$$a_{jk} = \text{probability of transition from state j to k}, \quad \forall\ j,k \in Q$$

F. 1st Order HMM Training Diagram

We show a training of the HMM of the 1^{st} Order for the input sequence "8 7 6". We process this input sequence on the HMM and try to find the most likely sequence of "weather of days" corresponding to the given "number of ice creams" a person eats each consecutive day.

The terminologies we have used are such that the equation "[CHH] $X_3(1)$=0.0032 * 0.14 = 0.000448" from Fig. 2 represents that this probability is the likelihood for assigning state sequence [CHH] to the input sequence "8 7 6" and it is finally calculated in the 3^{rd} column for State q_1(HOT) by multiplying two other probabilities [CH] $X_2(1)$ and P(H/H) * P(6/H) because of "(1)" of Section H.

 ISBN: 978-1-61208-202-8

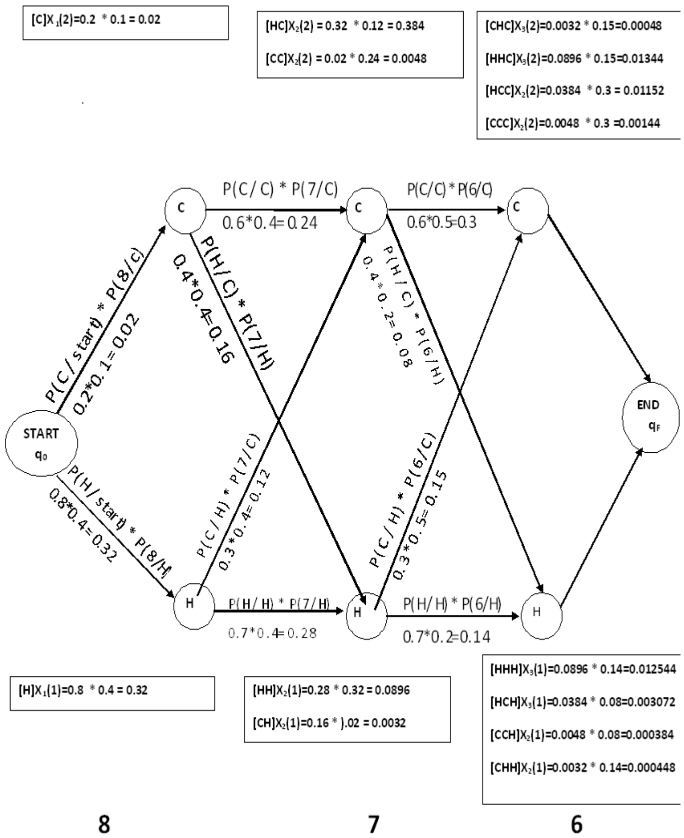

Figure 2. 1st Order HMM Training diagram

[CHH] is one possible output state sequence of the "weather of days" corresponding to the given input sequence "8 7 6" of the "number of ice creams" a person eat each consecutive day.

G. All Possible Output Observations

This table shows all possible output observations that can be assigned to the input observation along with their likelihood for the given input observation.

Table II. Output Observation Probabilities

Output Observation	Probability
H H H	0.012544
C C C	0.00144
H C C	0.01152
H H C	0.01344
C H H	0.000448
C C H	0.000384
H C H	0.003072
C H C	0.00048

We conclude that "H H C" is the most likely output observation for the input sequence "8 7 6" because its probability is found to be highest, i.e., 0.01344 using "(2)"of Section H.

H. Training of 1st Order HMM

Transition probabilities (TPs) of each state of the HMM either depend on one previous state or more than one previous state. If they depend on one previous state (M=1), the HMM is called 1st order HMM. If all TPs of states of HMM depend on two previous states (M=2) then, the HMM is called 2nd order HMM and so on. We gave a HMM in Section C and then trained that model for the input sequence "8 7 6" in Section F. The trained model shows all calculated parameters. Transition probabilities of the 1st order model (i.e., M=1) are represented as below:

$$P(q_i \mid q_{i-1}) \qquad \forall\ q_1 \ldots q_S$$

For our example model we have given these probabilistic parameters along with the transitions of Fig. 1 in Section C. The likelihood of each possible output observation is calculated from the following formula.

$$[\,\Pi\, P(t_i \mid t_{i-1}) P(w \mid t_i)] \tag{1}$$

The most likely output observation is calculated from the following formula.

$$\underset{q_1 \ldots q_S}{\text{Argmax}} \quad [\,\Pi\, P(q_i \mid q_{i-1}) P(w \mid q_i)] \tag{2}$$

The 1st Order HMM training diagram shows all possible output observations and their likelihood probability estimates by a brute force approach. The viterbi algorithm calculates only the most likely output observation sequences. For each input observation, we calculate its likelihood for both states, i.e., for both cold and hot. So, for three input observations we have shown three columns. Each column has state q_1(cold) and q_2(Hot).

III. STATE COMPLEXITY OF 1ST ORDER HMM

Training of the HMM is a state diagram that shows how a given input observation sequence is processed by the underlying HMM. This Training model gives us all possible output observations or all possible classifications for a given input observation sequence.

If 'H' is a 1st order HMM, 'S' is the number of states of 'H'. 'X' is an input observation sequence. 'L' is the training diagram of 'H' for 'X'. 'K' is the number of observations in 'X'. 'M' is the total number of hidden states of 'L' which are trained for 'X'.

M= total number of hidden states of 'L'

i.e., M= state complexity of 'L'

Then, M = S * K

So, we can see the state complexity of the training diagram of the 1st order HMM depends on S (number of states of underlying HMM) and K (number of input observation of input sequence) such that for each input observation ('K' in total), we calculate its likelihood for all states ('S' in total). As we can see in Fig. 2, for each of the three input observations, we have to find the likelihood for both states, i.e., (K=3 and S=2, so M=2*3=6). There are 6 likelihoods calculated in total represented by 6 states of Fig. 2.

Two additional ('start' and 'end') states are also added to complete the Fig. 2. This state complexity is of the brute Force expansion, which is usually intractable for most real world classification problems, as the number of possible hidden states sequences is extremely high and scales exponentially with the length of the input sequence.

IV. RULE TO DRAW MTH ORDER HMM DIAGRAM FOR M>1

First of all, we have to define 'S' basic categories or states we intend to assign to the input symbols. In the case of the Mth order HMM, each of these 'S' categories will depend on 'M' previous categories.

Then, we will find all possible combinations for 'M' previous categories. The M^{th} order HMM diagram will have two kinds of states.

(1) The 'S' original states (or categories) that we intend to be ultimately assigned to input symbols.

(2) The states that represent all possible combinations of 'S' states of length 'M'. (i.e., S^M).

Each of these 'S' distinct states and S^M distinct combinations of states will be combined to form M^{th} order HMM as shown in Fig. 3 for M=2.

Example1: if we have two basic categories for "Weather of the Day", i.e., Cold (C) or Hot (H), i.e., S=2 and M=2. The total possibilities for 'M' previous categories are as follows:

Table III. Possible Previous States for M=2

Possibility for M=2 previous state	Remarks
YZ= HH, CH, HC, CC	Last category is Z, second last is Y. Each of the Y and Z represent either H or C

Since, the total possibilities of M previous categories are $S^M=2^2=4$.

Example2: if we have two basic categories for "Weather of the Day", i.e., Cold (C) or Hot (H), i.e., S=2 and M=3. The total possibilities for 'M' previous categories are as follows:

Table IV. Possible Previous States for M=3

Possibility for M=3 previous state	Remarks
XYZ=HHH,HHC,HCH,CHH, HHC, HCC, CHC,CCC	Last category is Z, second last is Y and third last is X Each of the X, Y and Z represent either H or C

Since, the total possibilities of M previous categories are $S^M=2^3=8$.

V. Transition probability matrix Dimensions of M^{TH} Order HMM

The M^{th} order HMM means that the probability of transiting to each state depends on last 'M' states.

If 'S' is the number of states (categories) and 'M' is the order of HMM, then the Transition probability matrix of the HMM is 'M+1' dimensional such as $S^i * \ldots * S^{M+1}$. The superscript on S represents the dimensions, i.e., how many times S should appear.

In case S=2 and M=2, i.e., 2^{nd} order HMM then the transition probability matrix is 3 dimensional, i.e., 2*2*2.

Further, $A=\{a_{ijk}\}$ such that $\sum a_{ijk} = 1$

Transition a_{ijk} represents the transition probability to state 'k' depending upon the last two categories j & i respectively as shown in Fig. 3.

VI. State Complexity of M^{TH} Order HMM for M>1

Theorem:
State complexity of M^{th} Order HMM = $S+S^M$ and
State complexity of M^{th} Order HMM training diagram= N*K. $\forall$ M>1.

Proof:

Let 'S' represent the total number of basic categories that we intend to assign to input symbols and M is the order of the HMM. Let 'N' be the state complexity of the M^{th} order HMM. Recall our basic idea of the higher order HMM, that each state (or category) depends on previous 'M' states. The M^{th} order training diagram of the HMM will have two kinds of states.

(1) The 'S' original states (or categories) that we intend to assign to input symbols.

(2) The states that represent all possible combinations of 'S' states of length 'M'. (i.e., S^M)

Therefore, combining both the above kinds of states for the M^{th} order HMM will give us a total number of states or state complexity of the M^{th} Order HMM as below:

N=State complexity of M^{th} Order HMM = $S+S^M$.

For completeness, we add '2' as well for the "start" and "end" state.

$$N= 2+S+S^M \quad (3)$$

State complexity of its training model will depend on the number of observations of the input sequence similar to what is discussed in Section F.

Let 'X' be an input sequence. Let 'K' be the number of input observations of 'X'. Let 'P' be the total number of states of training diagram of the M^{th} order HMM for 'X'. Then,

$$P=N*K \quad (4)$$

Equations "(3)" and "(4)" above prove the theorem.

All possible combinations of states, i.e., S^M is calculated in [7] as well. We have now given a generalized formula "(4)" for the state complexity of the higher order HMM diagram.

Indeed, $P(t_3|t_2,t_1)$ represents the transition probability of the 2^{nd} order model because it says the probability of "t_3"depends on the last two (i.e., M=2) states "t_2" and "t_1". Similarly, we know that $P(t_2|t_1)$ is the transition probability of the 1^{st} order model because it says the probability of "t_2" depends on the last single (i.e., M=1) state "t_1" and this is mentioned in [2] as well.

Law and Chan [4] also defined that, for the M^{th} order HMM, all possible combinations of states should be S^M but do not explain how each of these S^M states will be connected and any higher order HMM diagram is also not illustrated.

VII. 2^{ND} Order HMM Example

For M=2, transition probabilities should depend on the last two states. We represent each transition probability by 'a_{ijk}'.

This means that a_{ijk} = transition probability to state k depending on the last two states j & i respectively.

$$a_{ijk} = P(q_k \mid q_j,q_i)$$

In the 2^{nd} order HMM, the transition probability matrix is three dimensional, i.e. a_{ijk}, as the probability of transiting to a new state depends on the last two previous states as mentioned in [5] as well. Hence the dimension of the transition probability matrix is S*S*S, where 'S' is the number of states.

Let's suppose for our example diagram of 2nd order HMM, the number of states, i.e., S=2. The transition probability for each state will depend on all possible "pairs of states". If we have S=2 states then the total number of pairs of states = $S^2 = 4$

 ISBN: 978-1-61208-202-8

The dimension of the transition probability matrix is S*S*S, i.e., 2*2*2 , which means, in total, 8 transitions represented in general by a_{ijk}.

Each a_{ijk} = the transition probability to state 'k' such that last two states are j & i respectively.

We write the following formula using "(3)".

Total number of states of the 2nd order HMM = $2+S+S^2$

$$= 2+2+2^2 = 8$$

VIII. DIAGRAM OF 2ND ORDER HMM

We draw in this section the 2nd Order HMM diagram of the example discussed in Section VII. Let's suppose we have S=2 states. The first state is called 'Cold' represented by C_1 and the second state is 'Hot' represented by H_2. The diagram of this example second order HMM is given in Fig. 3 with two additional start and end states. The transition probability matrix will contain 8 transition probabilities, i.e., $a_{ijk}=\{a_{111},a_{112},a_{121},a_{122},a_{211},a_{212},a_{221},a_{222}\}$. The other transition probabilities in the diagram q_{ik} and q_{ijk} are just for completion of the diagram and for the calculation of actual transition probabilities, a_{ijk}.

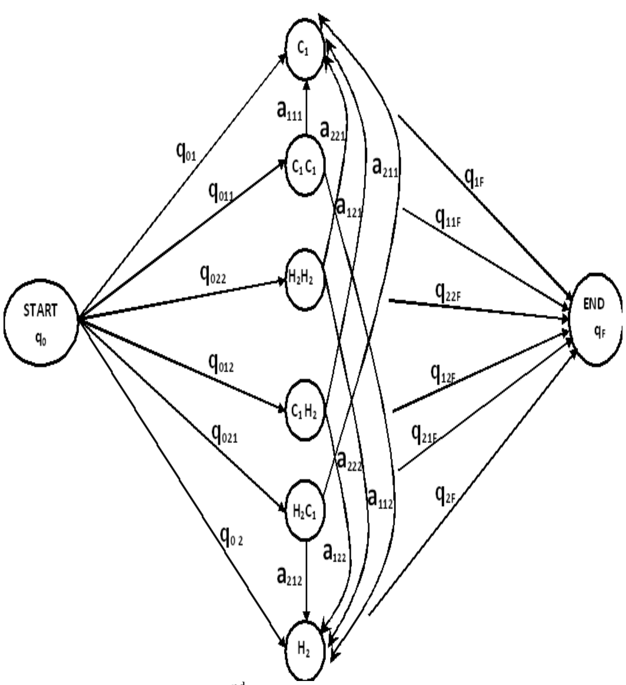

Figure 3. 2nd Order Hidden Markov Model diagram

We describe the transition probabilities, a_{ijk}, for the Fig. 3 in Table V.

Table V. Transition probability Description for 2nd Order Hidden Markov Model diagram

Transition Probability	Transition From State	Transition To State	Description
a_{111}	C_1C_1	C_1	The transition represents the probability of occurrence of day C_1 based on the last two days C_1 & C_1 respectively.
a_{112}	C_1 C_1	H_2	The transition represents the probability of occurrence of day H_2 based on the last two days C_1 & C_1 respectively.
a_{121}	C_1 H_2	C_1	The transition represents the probability of occurrence of day C_1 based on the last two days H_2 & C_1 respectively.
a_{122}	C_1 H_2	H_2	The transition represents the probability of occurrence of day H_2 based on the last two days H_2 & C_1 respectively.
a_{211}	H_2 C_1	C_1	The transition represents the probability of occurrence of day C_1 based on the last two days C_1 & H_2 respectively.
a_{212}	H_2 C_1	H_2	The transition represents the probability of occurrence of day H_2 based on the last two days C_1 & H_2 respectively.
a_{221}	H_2 H_2	C_1	The transition represents the probability of occurrence of day C_1 based on the last two days H_2 & H_2 respectively.
a_{222}	H_2 H_2	H_2	The transition represents the probability of occurrence of day H_2 based on the last two days H_2 & H_2 respectively.

IX. CONCLUSION AND FUTURE WORK

In this paper, we presented the state complexity of the Mth Order HMM. We also presented the state complexity of the diagram for "Training of Mth Order HMM". We explicitly defined HMM of different orders along with its training in a clear cut way. We presented a complete diagram of the "1st order training HMM" in which all possible sequences of categories are mentioned along with their probabilities of assigning these output observations to input observations. We explicitly defined a generalized rule to give the "dimension of Transition probability matrix of HMM", which is also not available in the literature yet. We defined a generalized rule for drawing the Mth order HMM diagram for M>1 and also drew the HMM diagram for M=2, which is not available in the literature yet. Our study of the state complexity of HMM will thus help researchers to analyze the complexity of implementation of their proposed HMM based scheme. This paper has introduced the terminology, i.e., "State complexity of HMM" that will lead researchers to explore the state complexity of different kinds of HMM and similar stochastic models. Diagrams for Mth Order HMM for M>2 can also be explored. Since the complexity of computing applications that exploit HMM depends on the complexity of HMM used. Therefore, our analysis of state complexity of HMM will help to analyze the complexity of those computing applications that exploit HMM.

ACKNOWLEDGMENT

We acknowledge the suggestions of the Late Prof. Dr. Sheng Yu and Prof. Bob Mercer for this paper. This work is funded by the Higher Education Commission, Pakistan and the University of Western Ontario, Canada.

 ISBN: 978-1-61208-202-8

REFERENCES

[1] Daniel Jurafsky and James H. Martin, “Speech and Language Processing an Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition”, 2nd Edition 2008, pp. 146- 148, 177.

[2] Scott M. Thede and Mary P. Harper, “A Second-Order Hidden Markov Model for Part-of-Speech Tagging”, In Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics on Computational Linguistics (ACL '99). Association for Computational Linguistics, Stroudsburg, PA, USA, 1999, pp. 175-182, doi:10.3115/1034678.1034712.

[3] Thorsten Brants, “TnT: A Statistical Part-of-Speech Tagger”. In Proceedings of the Sixth Conference on Applied Natural Language Processing (ANLC '00). Association for Computational Linguistics, Stroudsburg, PA, USA, 2000, pp. 224-231, doi:10.3115/974147.974178.

[4] Hubert Hin-Cheung Law and Chorkin Chan, “N-th Order Ergodic Multigram HMM for Modeling of Languages Without Marked Word Boundaries”. In Proceedings of the 16th Conference on Computational Linguistics - Volume 1 (COLING '96), Vol. 1. Association for Computational Linguistics, Stroudsburg, PA, USA, 1996, pp. 204-209, doi:10.3115/992628.992666.

[5] Zaid Md Abdul Wahab Sheikh and Felipe Sanchez-Martinez, “A Trigram POS Tagger for Apertium Free/Open-Source Machine Translation Platform”, In Proceedings of the First International Workshop on Free/Open-Source Rule-Based Machine Translation, Alacant, Spain, November, 2009, pp. 67-74

[6] Fahim Muhammad, Hasan, Naushad Uzaman, and Mumit Kahn “Comparison of Different POS Tagging Techniques (N-gram, HMM and Brill’s Tagger) for Bangla”, Bangladesh, 2006, pp. 31-37

[7] Mohammed Albared, Nazlia Omar, and Mohd. Juzaiddin Ab Aziz, “Developing a Competitive HMM Arabic POS Tagger Using Small Training Corpora”, In Proceedings of the Third International Conference on Intelligent Information and Database Systems - Volume Part I (ACIIDS'11), Ngoc Thanh Nguyen, Chong-Gun Kim, and Adam Janiak (Eds.), Vol. Part I. Springer-Verlag, Berlin, Heidelberg, 2011, pp. 288-296.

[8] Sheng Yu, “State Complexity of Regular Languages”, Journal of Automata, Languages and Combinatorics”, Volume 6 Issue 2, May 2001, pp. 221-234.

[9] Greg Kochanski, “Markov Model, Hidden and Otherwise”, UTC, March, 2005, pp. 1-11

[10] Shaojun Zhao, “Named Entity Recognition in Biomedical Texts Using an HMM Model”. In *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications* (JNLPBA '04), Association for Computational Linguistics, Stroudsburg, PA, USA, 2004, pp. 84-87.

[11] Szymon Jaroszewicz, “Interactive HMM Construction Based on Interesting Sequences”. In Proc. of Local Patterns to Global Models (LeGo'08) Workshop at the 12th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD'08), Antwerp, Belgium, 2008, pp. 82-91.

[12] Yu-Shu Chen and Yi-Ming Chen, “Combining Incremental Hidden Markov Model and Adaboost Algorithm for Anomaly Intrusion Detection”, In Proceedings of the ACM SIGKDD Workshop on CyberSecurity and Intelligence Informatics (CSI-KDD '09), ACM, New York, NY, USA, 2009, pp. 3-9, doi:10.1145/1599272.1599276.

[13] Rahul Khanna and Huaping Liu, “Distributed and Control Theoretic Approach to Intrusion Detection”. In Proceedings of the 2007 International Conference on Wireless Communications and Mobile Computing (IWCMC '07). ACM, New York, NY, USA, 2007, pp. 115-120, doi:10.1145/1280940.1280965

[14] Mohammed J. Zaki, Christopher D. Carothers, and Boleslaw K. Szymanski, “VOGUE: A Variable Order Hidden Markov Model With Duration Based on Frequent Sequence Mining”, Journal ACM Transactions on Knowledge Discovery from Data, Volume 4, Issue 1, Article 5, January 2010, 31 pages, doi:10.1145/1644873.1644878