A SURVEY OF MARKOV CHAIN MODELS IN LINGUISTICS APPLICATIONS

Fawaz S. Al-Anzi and Dia AbuZeina

Department of Computer Engineering, Kuwait University, Kuwait City, Kuwait
fawaz.alanzi@ku.edu.kw, abuzeina@ku.edu.kw

ABSTRACT

Markov chain theory is an important tool in applied probability that is quite useful in modeling real-world computing applications. For a long time, researchers have used Markov chains for data modeling in a wide range of applications that belong to different fields such as computational linguistics, image processing, communications, bioinformatics, finance systems, etc. This paper explores the Markov chain theory and its extension hidden Markov models (HMM) in natural language processing (NLP) applications. This paper also presents some aspects related to Markov chains and HMM such as creating transition matrices, calculating data sequence probabilities, and extracting the hidden states.

KEYWORDS

Markov chains, Hidden Markov Models, computational linguistics, pattern recognition, statistical

1. INTRODUCTION

Markov chains theory is increasingly being adopted in real-world computing applications since it provides a convenient way for modeling temporal, time-series data. At each clock tick, the system moves into a new state that can be the same as the previous one. A Markov chain model is a mathematical tool that capture the patterns dependencies in pattern recognition systems. For this reason, Markov chain theory is appropriate in natural language processing (NLP) where it is naturally characterized by dependencies between patterns such as characters or words.

Markov chains are directed graphs (a graphical model) that are generally used with relatively long data sequences for data-mining tasks. Such tasks include prediction, classification, clustering, pattern discovery, software testing, multimedia analysis, networks, etc. Reference [1] indicated that there are two reasons of Markov chains popularity; very rich in mathematical structure and work well in practice for several important applications. Hidden Markov models (HMM) is an extension of Markov chains that used to find the hidden system’s states based on the observations.

In order to facilitate the research in this direction, this paper provides a survey of this so popular data modeling technique. However, because of the wide range of the research domains that use this technique, We specifically focus on the linguistics related applications. Reference [2] list some domains that utilize Markov chains theory which include: physics, chemistry, testing, speech recognition, information sciences, queueing theory, internet applications, statistics, economics and finance, social sciences, mathematical biology, genetics, games, music, baseball.
Markov text generators, bioinformatics. Reference [3] lists the five greatest applications of Markov chains that include Scherr’s application to computer performance evaluation, Brin and Page’s application to PageRank and Web Search, Baum’s application to HMM, Shannon’s application to information theory, and Markov’s application to Eugen Onegin.

This paper is organized as follows. The next section presents a background of Markov chains theory. Section 3 highlights the main concepts of HMM followed by a literature review of Markov chains and HMM in section 4. Finally, we conclude in section 5.

2. MARKOV CHAINS

Markov chains are quite useful in modeling computational linguistics. A Markov chain is a memoryless stochastic model that describes the behaviour of an integer-valued random process. The behaviour is the simple form of dependency in which the next state (or event) depends only on the current state. According to [4], a random process is said to be Markov if the future of the process, given the present, is independent of the past. To describe the transitions between states, a transition diagram is used to describe the model and the probabilities of going from one state to another. For example, Figure 1 shows a Markov chain diagram with three states (Easy, Ok, and Hard) that belong to exam cases (i.e., states). In the figure, each arc represents the probability value for transition from one state to another.

![Markov Chain Diagram](Image)

Figure 1. A Simple Markov chain with three states

The Markov chain diagrams are generally represented using state transition matrices that denote the transition probabilities from one state to another. Hence, a state transition matrix is created using the entire states in the system. For example, if a particular textual application has a training data that contains N states (e.g., the size of lexicon), then the state transition matrix is described by a matrix A= {aij} of size N*N. In matrix A, the element aij denote the transition probability from a state i to a state j. Table 1 shows how the state transition matrix used to characterize the Markov diagram shown in Figure 1. That is, the matrix carries the state transitions probabilities between the involved states (Easy, Ok, and Hard). For illustration, the P(E|H) denote to the probability of the next exam to be Easy given that the previous exam was Hard.

<table>
<thead>
<tr>
<th>Previous Exam</th>
<th>Next Exam</th>
<th>State</th>
<th>Easy (E)</th>
<th>Ok (O)</th>
<th>Hard (H)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Easy (E)</td>
<td></td>
<td>Easy (E)</td>
<td>P(E</td>
<td>E)</td>
<td>P(O</td>
</tr>
<tr>
<td>Ok (O)</td>
<td></td>
<td>Ok (O)</td>
<td>P(E</td>
<td>O)</td>
<td>P(O</td>
</tr>
<tr>
<td>Hard (H)</td>
<td></td>
<td>Hard (H)</td>
<td>P(E</td>
<td>H)</td>
<td>P(O</td>
</tr>
</tbody>
</table>

In Table 1, the sum of the probability values at each row is 1 as the sum of the probabilities coming out of each node should be 1. Hence, P(E|E)+P(O|E)+P(H|E) equal 1. Markov chain is a worthy topic that has many details. For examples, it contains discrete-time, continuous-time,
time-reversed, reversible, and irreducible Markov chains. The case shown in Figure 1 is irreducible case, also called ergodic, where it is possible to go from every state to every state.

To illustrate a simple Markov chain data model, a small data set contains two English sentences used to create a transition matrix based on the neighbouring characters sequences. The sentences are inspirational English quotes picked from [5]:

1. Power perceived is power achieved.
2. If you come to a fork in the road, take it.

Figure 2 shows the transition matrix of these quotes by counting the total number of occurrences of the adjacent two character sequences. It is a 19 × 19 matrix where the value 19 is the total number of unique characters appeared in the sentences (i.e. the two quotes). In this example, creating transition matrix is case insensitive where D is same as d, as an example. In addition, a space between two words discarded and not considered in the transition matrix. Figure 2 also shows that the maximum number in the matrix’s entries is 3 (a highlighted underlined value) which means that moving from character e to r (e→r) is the most frequently sequence appeared in this small corpus. The words that contains this sequence are:{ Power (two times) and perceived}.

Based on the information provided in the transition matrix shown in Figure 2. It is possible to answer some questions related to the given data collection. Among inquires, what is the total number of the two characters sequences appeared in the given data set? What are the two characters sequences that did not appear in the data collection? What is the least frequently two characters sequences in the data set? Accordingly, Markov chains are used as prediction systems such as weather forecasting. Therefore, it is possible to predict the tomorrow’s weather according to the today’s weather. For example, if we have two states (Sunny, Rainy), and the requirement is to find the probability P(Sunny|Rainy), Markov chains make it possible based on the information provided in the probability transition matrix. Another example of the using Markov chains is in the banking industry. A big portfolio of banks is based on loans. Therefore, Markov chains are used to classify loans to different states such as Good, Risky, and Bad loans.

For simplicity, the information presented in Figure 2 shows the transition matrix based on total number of occurrences. Figure 3 shows the same information but using probabilities instead of
the number of occurrences. That is, it contains the probability of moving from one character to another. As previously indicated, the sum of entries at each row is equal 1. In Figure 3, any matrix entry that has 0 means that there is no transition at that case. Similarly, if the matrix entry is 1, it means that there is only one possible output of that state. For example, the character “o” comes after “y”, and this is the only possible arc of the state “y”.

![Figure 3. A probability transition matrix of two characters sequences](image)

3. HIDDEN MARKOV MODELS

Hidden Markov models (HMM) is an extension to Markov chains models as both used for temporal data modeling. However, the difference is that the states in Markov chain models are directly observed while they are hidden in the case of HMM. We explain the concept of HMM based on Figure 1 that shows a three exam’s states Markov diagram. As a very simple example, supposed that a student’s parents want to know the levels (i.e the difficulty) of their son’s exams, naturally, it is possible to recognize the exam as Easy or Ok if the son feels Fine. Similarly, it is possible to recognize the exam as Hard if the son looks Scared. From the parents’ point of view, the required states (i.e. Easy, Ok, or Hard) are hidden. However, they directly observe the student’s reaction or feeling. Hence, the parents might use the observed reactions as an indication to know the hidden states. HMM is described using three matrices: the initial probability matrix, the observation probability matrix, and the state transition matrix. Figure 4 shows a HMM diagram that shows the states and the observations. In the figure, each arc represents the probability between the states and between the states and the observations.

![Figure 4. A HMM diagram with the transition and the observation arcs](image)
Based on the information provided in the matrices, either Baum-Welch (also called any path) or Viterbi (also called best path) algorithms used to find the probability scores during recognition phase. Figure 5 shows the trellis diagram foreground states HMM. While Baum-Welch algorithm is used to compute the recognition probability of a sequence, Viterbi is used to find the best-state sequence associated with the given observation, this process is also known as back-tracking. Hence, after computing the observations sequence probability and finding the maximum-likelihood procedure (supposed the star in Figure 5), the Viterbi algorithm leads the process back to identify the states (sources) from which the observations sequence have been emitted. In Figure 5, the maximum probabilities supposed to be achieved at the states shown using the dotted lines: Ok, Easy, Hard, respectively.

<table>
<thead>
<tr>
<th>States</th>
<th>t=1</th>
<th>t=2</th>
<th>...</th>
<th>t=n</th>
</tr>
</thead>
<tbody>
<tr>
<td>Easy</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ok</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hard</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Observations → Observation 1  Observation 2  ...  Observation n
Fine or Scared  Fine or Scared  ...  Fine or Scared

![Trellis diagram of three states HMM](image)

**4. LINGUISTIC APPLICATIONS**

In the literature, there are quite many works on modeling content dependencies for linguistics applications. Markov chain models and HMMs are of great interest to linguistic scholar who primarily work on data sequences. Even though this study focuses on linguistic applications, however, Markov chains used to model a variety of phenomena in different fields. The following are some of studies employed Markov chains. We intentionally ignored the references as the literature has too many studies employed Markov chains:

- Authorship attribution
- Speech emotion recognition
- Part-of-speech tagging
- Machine translation
- Text classification
- Text summarization
- Optical character recognition (OCR)
- Named entity recognition
- Question answering
- Authorship attribution
- etc.

For the reader who interested in NLP, Reference [6] is a good reference as it demonstrates a thorough study of NLP (Almost) from Scratch.
4.1. Markov chains based research


4.2. Hidden Markov models based research

Linguistic HMM based research has been for long an active research area due to the rapid development in NLP applications. The literature has many studies as follows. Reference [26] proposed to extract acronyms and their meaning from unstructured text as a stochastic process using HMM. Reference [27] proposed a morphological segmentation method with HMM method for Mongolian. Reference [28] employed HMM for Arabic handwritten word recognition based on HMM. Reference [29] presented a scheme for off-line recognition of large-set handwritten characters in the framework of the first-order HMMs. Reference [30] proposed the use of hybrid HMM/Artificial Neural Network (ANN) models for recognizing unconstrained offline handwritten texts. Reference [31] used HMMs for recognizing Farsi handwritten words.


5. CONCLUSIONS

This work demonstrates the potential and the size of Markov chains research. The study reveals that the Markov chain and HMM is of high important for linguistic applications. Similarly, Markov chains are also widely used in many other applications. For future work, it worthy to explore the power of Markov chain in new linguistic and scientific directions with more details.

ACKNOWLEDGEMENTS

This work is supported by Kuwait Foundation of Advancement of Science (KFAS), Research Grant Number P11418EO01 and Kuwait University Research Administration Research Project Number EO06/12.

REFERENCES


