Industrial Engineering Letters
ISSN 2224-6096 (print) ISSN 2225-0581 (online)
Vol 2, No.5, 2012

MARKOV MODEL: Analyzing its behavior for Uncertainty conditions

Lalitha.R.V.S
Sri Sai Aditya Institute of Science and Technology,Surampalem
e-mail: rvslalitha@gmail.com

Sri Divya.R
Sri Sai Aditya Institute of Science and Technology,Surampalem
e-mail: rayavarapudivya@gmail.com

Abstract: Markov model is used to analyze the dynamic behavior of the system in predicting the next state with the previous state. The process of attempting to guess the next character reveals information about the password strategy. In this paper, we give fuzzy inferences about the guessing passwords, by examining with the previous state and computing the possible outcomes of probability of each character. For some problems there cannot be complete solutions. For such problems Fuzzy inferences allow us to evaluate sub expressions. In the present paper, we discuss how to trace out some uncertainty conditions and analyze their behavior using fuzzy inference system and finally test the system for finding steady state behavior in guessing the characters in the password.

Keywords: Markov model, Fuzzy sets, Transition matrix, Membership functions, Fuzzy Logic

1. **Introduction**: To understand the nature of the threat to password-based systems, we have to have information about the guessable passwords. Markov model [1] gives a representation of guessable passwords by creating a transition matrix using prefixes. If the users are assigned passwords that contain eight randomly selected printable characters, then remembering password is difficult, so password cracking is very difficult. The techniques used to eliminate guessable passwords are 1. *User Education*-users are given the guidelines for selecting hard passwords. 2. *Computer generated passwords*- If the passwords are random, difficult to remember.3--*Reactive password checking*—system periodically runs a cracker to find guessable passwords for cancelling passwords guessed by the user.4. *Proactive password checking*—user is allowed to select his or her own password.

While guessing passwords guessing entire string is difficult. But when a subset of string is guessed we can predict the next occurrence by storing its previous state. The next character to be arrived depends only on the previous state but not independent. This concept is useful in playing cards, rolling dice etc.

In this paper, we give mathematical representation for few mathematical models for the computation of next possible state based on the current state. In password checking, guessable password is compared with the dictionary of passwords stored. When the probability of the certain word is reached, we can forget the occurrence of the particular character in the next subsequent states. For that, we explain some of the mathematical representations for computing various combinations of subsets of the string.

The idea of fuzzy sets was born in July 1964. The concept of grade of membership is the backbone of fuzzy set theory. Fuzzy set is a set with smooth boundary. Fuzzy sets are used where to generalize query or to deal with partial membership. Partial membership occurs when either a consequent or antecedent have partial satisfaction. Fuzzy rules are helpful to give analysis for sub expressions. By taking conclusion, by combining all the outcomes of the fuzzy sub expressions we will obtain a solution which covers almost all the possible entities. In section 2 we discussed the basic principle of Markov model. In section 3 we discussed the concepts of uncertainty conditions by taking an example. In section 4, we discussed the behavioral study of Fuzzy inferences for some of the states can be deduced using inference logic of Fuzzy theory. The inference logic helps us to retrieve information either by comparative analysis or by probabilistic approach. We consider the probabilistic approach for evaluating subsets. In section 5, the significance of triangular membership function, operating between lower and upper bounds is discussed. Finally, in section 6, we have given mathematical representation, for some of the strings guessed, and

computed their probabilities. In section [7] we gave a few samples for procurement of probabilities and to have some history of previous states.

2. Principle of Markov model:- The problems associated with the above approaches are 1. *Space* and 2. *Time.* Markov model helps in preparing list of guessable passwords, which are to be rejected by the system.

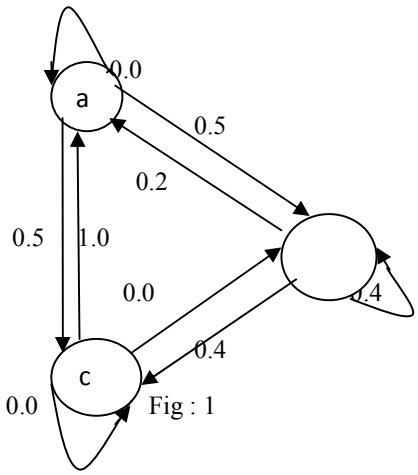

Fig : 1

M={ m, A, T, k}

M= {3, {a, b, c}, T, 1}

$$T=\begin{bmatrix} 0.0 & 0.5 & 0.5 \\ 0.2 & 0.4 & 0.4 \\ 1.0 & 0.0 & 0.0 \end{bmatrix}$$

A Markov model is a quadruple [m, A, T, k], where m is the number of states in the model, A is the state space, T is the matrix of transition probabilities, and k is the order of the model.

Markov model is applicable if each row of transition matrix sums to 1.

The sum of row of elements of row 1 =0.0+0.5+0.5=1.0
The sum of elements of row 2 =0.2+0.2+0.6=1.0
The sum of elements of row 3 =1.0+0.0+0.0=1.0
The Markov model is not applicable, if the elements of any row is less than or greater than 1.

3. **Analyzing the behavior for some uncertainty conditions:-**

In the Fig 2, entire information is sinked into the node ‘b’, as there is no outgoing state. No outgoing state identifies that there will be no occurrence of next character. If we analyze the behavior of state b we can derive the previous state analysis that is reaching b.

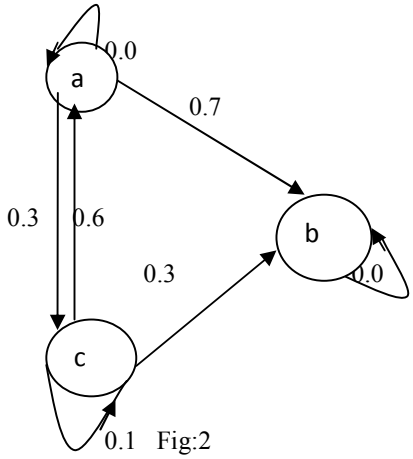

Fig:2

The transition matrix for Fig.2 is

$$T=\begin{bmatrix} 0.0 & 0.7 & 0.3 \\ 0.0 & 0.0 & 0.0 \\ 0.6 & 0.3 & 0.1 \end{bmatrix}$$

By this, we can conclude that the guessable passwords for this transition matrix can be ab, or cb, or ccb, or cccb or cccb or accb or aca etc. Which shows the position of 'b' is always last for most of the substrings and the position of 'a' can be either at the beginning or at the last.

The probable occurrences of the next character for 'a' can be a, b or c .The probable occurrences of the next character for 'c' is a or b. And the probable occurrences of the next character for 'b' is nothing.

4.Fuzzy Inferences:- We can apply fuzzy inferences logical reasoning, for the states that output to the same state. To know how to convert the crisp set into a fuzzy one and fuzzy into a crisp one, observe the following rules for the example states taken as below:

Let x is a password submitted by the user.

The fuzzy rule for the fig.1 can be written as

Rule 1: if x is a then y is b with 0.7 probability

Rule 2: if x is c then y is b with 0.3 probability

Obviously, multiple inputs produce the same output upto some extent. Fuzzy reasoning is used to generalize logical reasoning called *modus ponens*. This can be referred generalization, in the sense whatever the inputs submitted we are obtaining the state 'b' for some probabilities.

Fuzzy rule based inference:

As the probabilities of getting b for a and c are different, we can assure that we can predict b after a or c for 0.3 probability also. This type of inferences can by represented using membership functions.

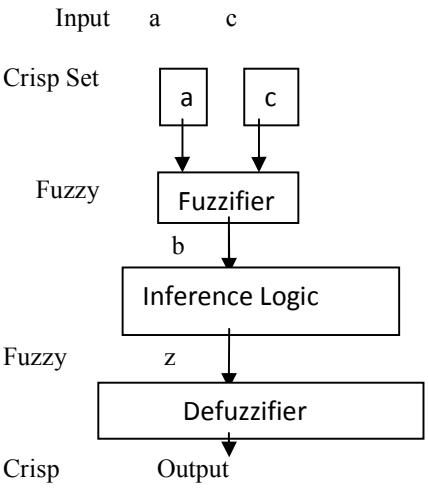

Fig:3

5. Representation of Fig 3, using Triangular membership function:

This function is designed, to distinguish the lower and upper bounds of two inputs to be considered for producing the same output. A fuzzy is defined by a function that maps objects in a domain of concern to their membership value in the set. Such a function is called membership function, denoted by Greek symbol μ for recognition and consistency. The membership [4] of fuzzy set A is defined as μ_a and the membership value of x in A is $\mu_a(x)$.

Let A= {a, b, c}

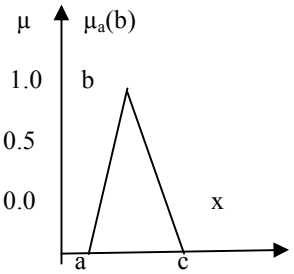

Fig:4

In the above fig , a with 0.7 probability and c with 0.3 probability outputs to c. The next occurrence of character b appears only with the probabilities of a 0.7 and b 0.3. Beyond that checking for the occurrence of character b is not required. If there is a process with 4 elements, we can go for a trapezoidal membership function. These membership functions allow us to give information about the range of values to be allowed for computation. To obtain the region of information that is common to both the states, we adopt scaling and clipping methods, to deduce inference logic. The following two diagrams, gives information about the inferred conclusion with a matching degree 0.3.

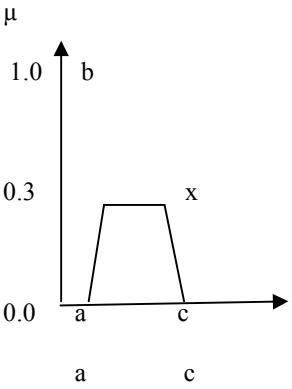

Fig. 5 inferred conclusion using clipping method

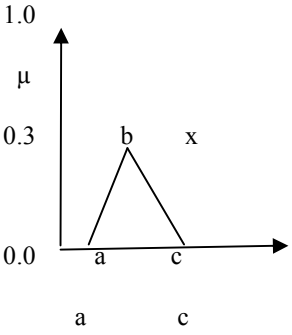

Fig. 6 inferred conclusion using scaling method

6. Computation of no. possible occurrences of each character [2]:-

To evaluate the behavior of the steady state vector,

Consider Markov chain with N states. A state vector for a Markov chain is a row vector $X=[x_1,x_2..x_n]$

Our example Markov process is with 3 states (a,b,c).

Row vector 1=a_{1a}, a_{1b}, a_{1c}

Row vector 2 =b_{2a}, b_{2b}, b_{2c}

Row vector 3=c_{3a}, c_{3b}, c_{3c}

a_{1a} is the probability that the probability of a in first state so that it gets the next character as a in the next state and so on.

Let X= [a b c]

Let T is n x n matrix, k is the k^{th} power of matrix, then[8],

$X=\lim_{k\to\infty} T^k$

Let X_0 denote the initial state of the Markov chain. In general,
$Xn = X_{n-1}T$

Let Identity matrix $I=\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix}$

And Transition matrix $T=\begin{bmatrix}0 & .7 & .3\\ 0 & 0 & 0\\ .6 & .3 & .1\end{bmatrix}$

x^0 is the matrix obtained initially by multiplying identity matrix with the transition matrix for finding the initial state[3].
Let us find probabilities for some example strings.
Probability of obtaining only 'b' s in the string.

$x^1 = x^0 T$

$$[0 \quad 1 \quad 0]\begin{bmatrix}0 & .7 & .3\\ 0 & 0 & 0\\ .6 & .3 & .1\end{bmatrix} = [0 \quad 0 \quad 0]$$

Probability of obtaining only 'a' s in the string

$$[1 \quad 0 \quad 0]\begin{bmatrix}0 & .7 & .3\\ 0 & 0 & 0\\ .6 & .3 & .1\end{bmatrix} = [0.6 \quad 0 \quad 0]$$

Probability of obtaining only 'c's in the string[6]:

$$[0 \quad 0 \quad 1]\begin{bmatrix}0 & .7 & .3\\ 0 & 0 & 0\\ .6 & .3 & .1\end{bmatrix} = [0 \quad 0 \quad .4]$$

Finding probability of two character word (bi-gram) in the password-

Probability of getting ab in the string

$$[1 \quad 1 \quad 0]\begin{bmatrix}0 & .7 & .3\\ 0 & 0 & 0\\ .6 & .3 & .1\end{bmatrix} = [.6 \; 1 \; 0]$$

Finding probability of three character word(trigram)[5] in the password-

Probability of getting cab in the string

$$[1 \quad 1 \quad 1]\begin{bmatrix}0 & .7 & .3\\ 0 & 0 & 0\\ .6 & .3 & .1\end{bmatrix} = [.6 \; 1 \; .4]$$

and so on.

The strings are checked for probabilities taking initial state as 'a'. If we change the initial state, we go for any combination of strings, like ba, cab etc.

7. Uncertainity conditions:

Set s is the set of acceptable strings. And w1, w2, w3 ... are the probable strings/substrings in the set.

$S=\{ w_1,w_2 \ldots w_n\}$

Let N_{ij} be the no. of transitions made by a string from state i to state j.

Total no. of transitions(N_{ij}) from state i to state j depends on the probability of the particular character that can have transition from state i to state j.

N_i actual no. of transitions made.

Then transition probability(tp) from state i to state j is N_{ij} / N_i

Then emission probability can be computed as

Probability of state I - N_{ij} / N_i

For eg. Emission probability (ep) for state a is

$0.7 – N_{ab}/N_a$

As the emission probability increases, the probability of guessing password character decreases.

8. Conclusion:- The probabilities computed for each character from transition matrix and the probabilities derived from the vector, both become same. With the above computation analysis, we can restrict our guess in predicting the next character up to some extent.

9. Future scope: This analysis can be extended to second order Markov model. By giving analysis of more and more sub expressions of fuzzy analysis, leads to comprehensive study of the predicting passwords.

References:

1. "Network Security Essentials " Applications and Standards by William Stallings, Pearson Education
2. http://ceee.rice.edu/Books/LA/smarkov/index.html
3. "Fuzzy markov modeling in automatic control of complex dynamic systems" V.Arkov, Institute of Mechanics, Russian Academy of Science, Ufa, Russia G.G.Kukikov, T.V. Breikin, Ufa State Aviation, Technical University, Ufa, Russia
4. FUZZY LOGIC - INTELLEGENCE ,CONTROL, AND INFORMATION by JOHN YEN, REGA LANGARI
5. "Introduction to Information Retrieval - Search Engines" *Michael Hawthornthwaite works at Acid Computer Services (Manchester) who specialize in web design, web development and bespoke software development*
6. http://en.wikipedia.org/wiki/Examples_of_Markov_chains
7. http://en.wikipedia.org/wiki/Fuzzy_logic
8. http://en.wikipedia.org/wiki/Markov_chain
9. §8.3–Regular Markov Chains
   Tom Lewis
   Winter Term 2008
10. http://www.cs.uiowa.edu/~hviet/papers/A%20Fuzzy%20Synset-based%20Hidden%20Markov%20Model%20for%20Automatic%20Text%20Segmentation.pdf