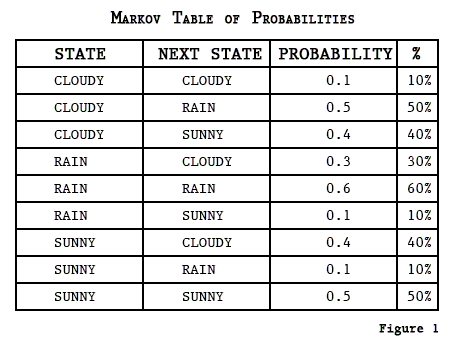
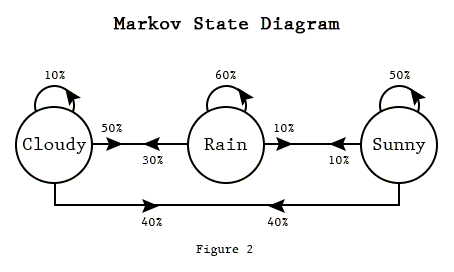
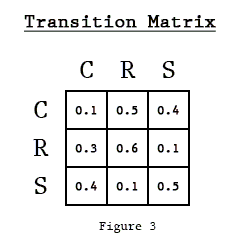
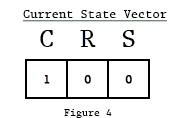
[**Markov Chains - Explained**](http://techeffigytutorials.blogspot.com/2015/01/markov-chains-explained.html)

Markov Chains is a probabilistic process, that relies on the current state to predict the next state. For Markov chains to be effective the current state has to be dependent on the previous state in some way; For instance, from experience we know that if it looks cloudy outside, the next state we expect is rain. We can also say that when the rain starts to subside into cloudiness, the next state will most likely be sunny. Not every process has the Markov Property, such as the Lottery, this weeks winning numbers have no dependence to the previous weeks winning numbers.  
  
Usually when we have data, we calculate the probability of a state by counting the amount of times it occurs within a total of all states, we then end up with 0.5(50%) cloudy, 0.3(30%) rain, 0.2(20%) Sunny days of a year; Containing no information about when they occur relating to each other. To get the dependent probabilities, we count the number of times states occur relating to a state, and when we do that for all the states we then end up with a Markov Table :

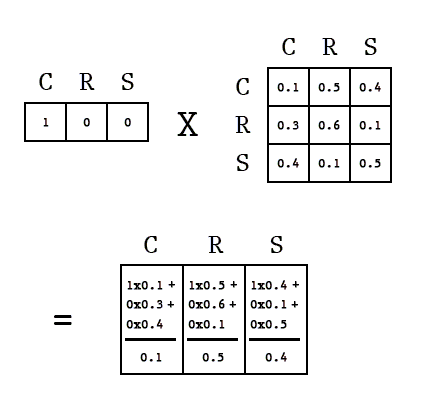
[](http://3.bp.blogspot.com/-VmxKc2OH_Q4/VK71D4Dok6I/AAAAAAAAABA/HWeLyc7O8fY/s1600/markovtbl.png)

To generate a simple prediction of events, we can say today is Cloudy, and from the table above we can determine that the next state will most likely be Rain with 50%, then we go to the Rain state, and the following day it will probably still be raining with 60%. Markov Chains can be represented as a state diagram, or a matrix called a Transition Matrix:

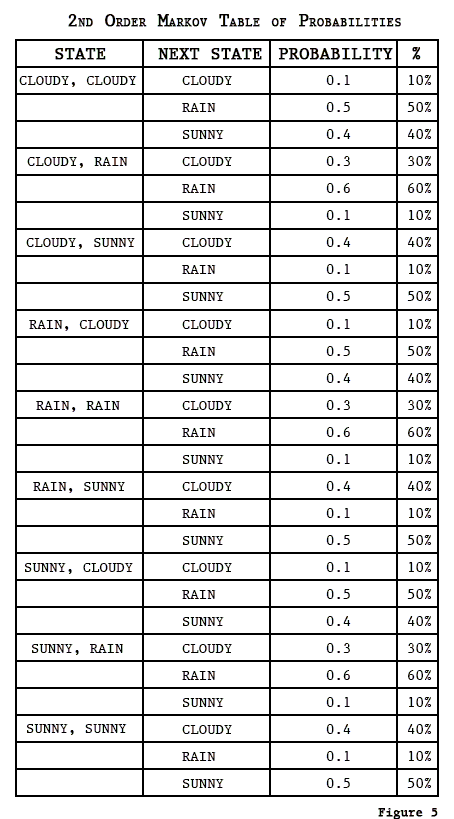
[](http://4.bp.blogspot.com/-u9XslQrACb8/VK71Cym3zQI/AAAAAAAAAA0/DAxkKTcCKvc/s1600/markovdiag.png)

[](http://1.bp.blogspot.com/-zHCmumGgQwU/VK71DvSnvAI/AAAAAAAAAA8/fgB0UCQYlQk/s1600/markovmx.png)[](http://1.bp.blogspot.com/-7g3vpKGtG1c/VK71C6qK8TI/AAAAAAAAAAw/bBgECJt8D3g/s1600/markovec.png)

The Transition Matrix transitions from Row to Column as in the Markov Table. We can do calculations with a Transition Matrix utilizing a State Vector(vector of our current conditions) to give us the probabilities of the next states. The above figure is set to 1(100%) cloudy for the current state, to calculate the probabilities of the next state we multiply the Current State Vector with the Matrix in figure 3:  
  
State2 = Vector\*Matrix  
=[(1 \*0.1 + 0\*0.3 + 0\*0.4); (1\*0.5 + 0\*0.6 + 0\*0.1); (1\*0.4 + 0\*0.1 + 0\*0.5)]  
= [C;R;S] = [0.1; 0.5; 0.4]

[](http://3.bp.blogspot.com/-vQwPkdqNTt4/VK71EPa6ffI/AAAAAAAAABE/5-HQ_SHcIFs/s1600/matrixmath.gif)

No surprise, most likely rain. Now we can calculate the probabilities of the state after that using the resultant vector and multiplying it with the matrix in figure 3:  
  
State3 =State2\*Matrix  
=[(0.1\*0.1 + 0.5\*0.3 + 0.4\*0.4); (0.1\*0.5 + 0.5\*0.6 + 0.4\*0.1); (0.1\*0.4 + 0.5\*0.1 + 0.4\*0.5)]  
= C;R;S = [0.32; 0.39; 0.29]  
Rain most likely again, and we can calculate the probabilities of the state after that by multiplying with figure 2 again. That is the method of the Markov Chain of probabilities.  
  
The Markov Chains that I have been working with are called 1st order Markov Chains, they only deal with 1 state to predict the next. In the above example, when it transitions from cloudy to rain, it then absorbs into the rain state, never leaving leaving that state. The reason is because the Transition Table only holds information of the last state, we don't know if it was sunny or raining before it was cloudy. you could have a 2nd order Markov Chain that would take the last two states and get the probability of the next states. All that is required is grouping the last two states into 1 state as in the example Table Below:

[](http://4.bp.blogspot.com/-3zMER5OIPmo/VLStLb19_PI/AAAAAAAAAHE/CUB2Q1V-kUY/s1600/2ndordermarkov.png)

When programming Markov Chains most developers use the table method, linking a list of states to its list of next state probabilities. Markov Chain text generator, trained on text, basically the states are words, and each word is linked to a list of words that have appeared after it in the training text.  
  
Here`s a quote from my 3rd Order Markov Chain Text Generator trained on the Bible "Then Joshua said to the olive-tree, Be king over us."

**Mathematical Modeling with Markov Chains and Stochastic Methods** By [**Lillian Pierson**](http://www.dummies.com/?s=&a=lillian-pierson)

A *stochastic model* is a tool that you can use to estimate probable outcomes when one or more model variables is changed randomly. A Markov chain — also called a *discreet time Markov chain* — is a stochastic process that acts as a mathematical method to chain together a series of randomly generated variables representing the present state in order to model how changes in those present state variables affect future states.

Imagine that you love to travel but that you travel only to places that are a) a tropical paradise, b) ultramodern cities, or c) mountainous in their majesty. When choosing where to travel next, you always make your decisions according to the following rules:

* You travel exactly once every two months.
* If you travel somewhere tropical today, next you will travel to an ultramodern city (with a probability of 7/10) or to a place in the mountains (with a probability of 3/10), but you will not travel to another tropical paradise next.

ADVERTISING

* If you travel to an ultramodern city today, you will travel next to a tropical paradise or a mountainous region with equal probability, but definitely not to another ultramodern city.
* If you travel to the mountains today, you will travel next to a tropical paradise (with probability of 7/10) or an ultramodern city (with a ­probability of 2/10) or a different mountainous region (with a prob­ability of 1/10).

Because your choice on where to travel tomorrow depends solely on where you travel today and not where you’ve traveled in the past, you can use a special kind of statistical model known as a Markov chain to model your destination decision making. What’s more, you could use this model to generate statistics to predict how many of your future vacation days you will spend traveling to a tropical paradise, a mountainous majesty, or an ultramodern city.

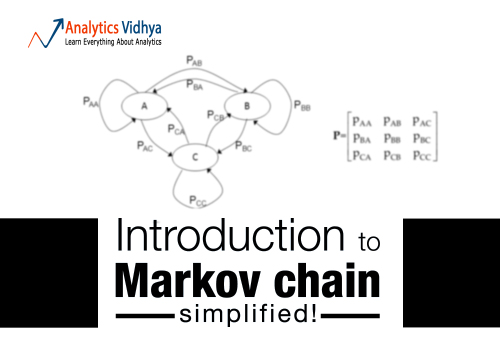
Looking a little closer at what’s going on here, the above-described scenario represents both a stochastic model and a Markov chain method. The model includes one or more random variables and shows how changes in these variables affect the predicted outcomes. In Markov methods, future states must depend on the value of the present state and be conditionally independent from all past states.

You can use Markov chains as a data science tool by building a model that generates predictive estimates for the value of future data points based on what you know about the value of the current data points in a dataset. To predict future states based solely on what’s happening in the current state of a system, use Markov chains.

Markov chains are extremely useful in modeling a variety of real-world processes. They’re commonly used in stock-market exchange models, in financial asset-pricing models, in speech-to-text recognition systems, in webpage search and rank systems, in thermodynamic systems, in gene-regulation systems, in state-estimation models, for pattern recognition, and for population modeling.

An important method in Markov chains is in Markov chain Monte Carlo (MCMC) processes. A Markov chain will eventually reach a steady state — a long-term set of probabilities for the chain’s states. You can use this characteristic to derive probability distributions and then sample from those distributions by using Monte Carlo sampling to generate long-term estimates of future states.

Markov chain is a simple concept which can explain most complicated real time processes.Speech recognition, Text identifiers, Path recognition and many other Artificial intelligence tools use this simple principle called Markov chain in some form. In this article we will illustrate how easy it is to understand this concept.

[](https://www.analyticsvidhya.com/wp-content/uploads/2014/07/Introduction-to-Markov-chain-simplified.jpg)Markov chain is based on a principle of “memorylessness”. In other words the next state of the process only depends on the previous state and not the sequence of states. This simple assumption makes the calculation of conditional probability easy and enables this algorithm to be applied in number of scenarios. In this article we will restrict ourself to simple Markov chain. In real life problems we generally use Latent Markov model, which is a much evolved version of Markov chain. We will also talk about a simple application of Markov chain in the next article.

**A simple business case**

Coke and Pepsi are the only companies in country X. A soda company wants to tie up with one of these competitor. They hire a market research company to find which of the brand will have a higher market share after 1 month. Currently, Pepsi owns 55% and Coke owns 45% of market share. Following are the conclusions drawn out by the market research company:

P(P->P) : Probability of a customer staying with the brand Pepsi over a month = 0.7

P(P->C) : Probability of a customer switching from Pepsi to Coke over a month = 0.3

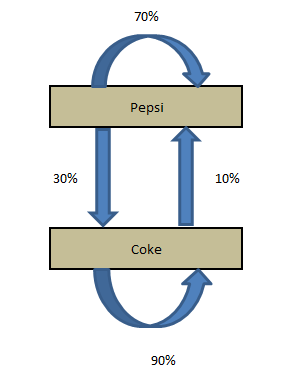
P(C->C) : Probability of a customer staying with the brand Coke over a month = 0.9

P(C->P) : Probability of a customer switching from Coke to Pepsi over a month = 0.1

We can clearly see customer tend to stick with Coke but Coke currently has a lower wallet share. Hence, we cannot be sure on the recommendation without making some transition calculations.

**Transition diagram**

The four statements made by the research company can be structured in a simple transition diagram.

[](https://www.analyticsvidhya.com/blog/wp-content/uploads/2014/07/transition.png)

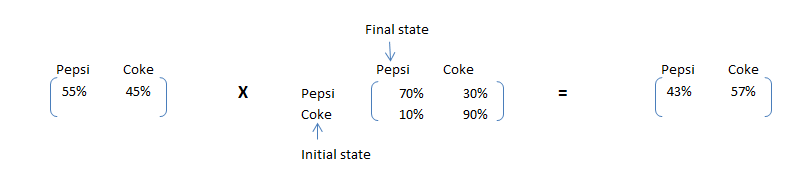
The diagram simply shows the transitions and the current market share. Now, if we want to calculate the market share after a month, we need to do following calculations :

Market share (t+1) of Pepsi = Current market Share of Pepsi \* P(P->P) + Current market Share of Coke \* P(C->P)

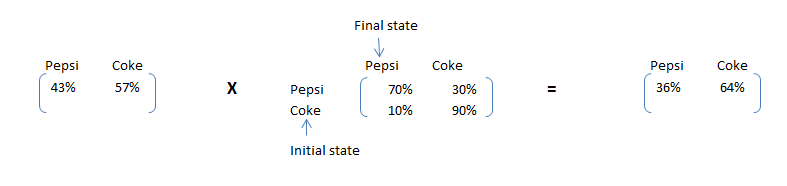
Market share (t+1) of Coke = Current market Share of Coke \* P(C->C) + Current market Share of Pepsi \* P(P->C)

These calculations can be simply done by looking at the following matrix multiplication :

Current State X Transition Matrix = Final State

[](https://www.analyticsvidhya.com/blog/wp-content/uploads/2014/07/trans1.png)

As we can see clearly see that Pepsi, although has a higher market share now, will have a lower market share after one month. This simple calculation is called Markov chain. If the transition matrix does not change with time, we can predict the market share at any future time point. Let’s make the same calculation for 2 months later.

[](https://www.analyticsvidhya.com/blog/wp-content/uploads/2014/07/trans2.png)

**Steady state Calculations**

Furthermore to the business case in hand, the soda company wants to size the gap in market share of the company Coke and Pepsi in a long run. This will help them frame the right costing strategy while pitching to Coke.The share of Pepsi will keep on going down till a point the number of customer leaving Pepsi and number of customers adapting Pepsi is same. Hence, we need to satisfy following conditions to find the steady state proportions:

Pepsi MS \* 30% = Coke MS \* 10%  ……………………………………………..1

Pepsi MS + Coke MS = 100% ……………………………………………………2

4 \* Pepsi MS = 100% => Pepsi MS = 25% and Coke MS = 75%

Let’s formulate an algorithm to find the steady state. After steady state, multiplication of Initial state with transition matrix will give initial state itself. Hence, the matrix which can satisfy following condition will be the final proportions:

**Initial state X Transition Matrix = Initial state**

By solving for above equation, we can find the steady state matrix. The solution will be same as [25%,75%]. In this article we introduced you to Markov chain equations and terminology. We also looked at how simple equations can be scaled using Matrix multiplication. We will use these terminologies and framework to solve a real life example in the next article.

**Business Use Case:** “Krazy Bank”, deals with both asset and liability products in retail bank industry. A big portfolio of the bank is based on loans. These loans make the majority of the total revenue earned by the bank. Hence, it is very essential for the bank to find the proportion of loans which have a high propensity to be paid in full and those which will finally become Bad loans. “Krazy Bank” has hired you as a consultant to come up with these scoping numbers.

All the loans, which have been issued by “Krazy Bank” can be classified into four categories :

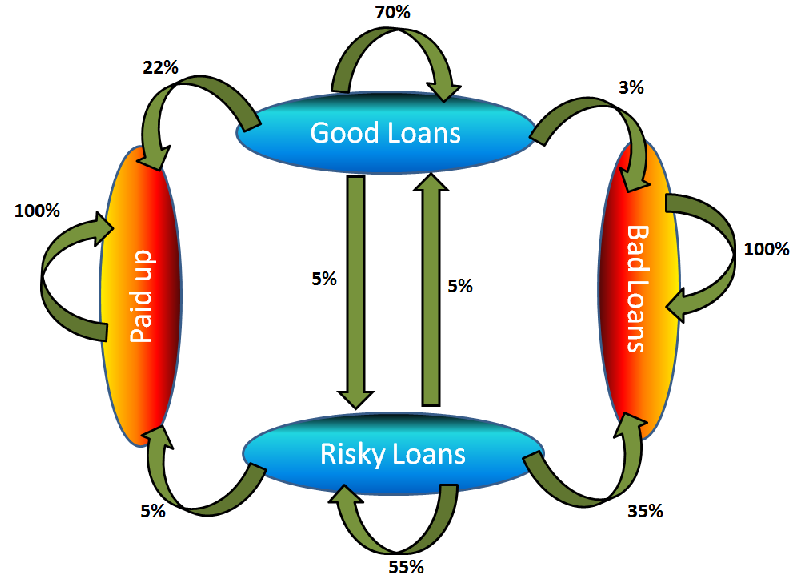
1. **Good Loans** : These are the loans which are in progress but are given to low risk customers. We expect most of these loans will be paid up in full with time.
2. **Risky loans :** These are also the loans which are in progress but are given to medium or high risk customers. We expect a good number of these customers will default.
3. **Bad loans :** The customer to whom these loans were given have already defaulted.
4. **Paid up loans :** These loans have already been paid in full.

**Short Note on Absorbing nodes**

Absorbing nodes in a Markov chain are the possible end states. All nodes in Markov chain have an array of transitional probability to all other nodes and themselves. But, absorbing nodes have no transitional probability to any other node. Hence, if any individual lands up to this state, he will stick to this node for ever. Let’s take a simple example. We are making a Markov chain for a bill which is being passed in parliament house. It has a sequence of steps to follow, but the end states are always either it becomes a law or it is scrapped. These two are said to be absorbing nodes. For the loans example, bad loans and paid up loans are end states and hence absorbing nodes.

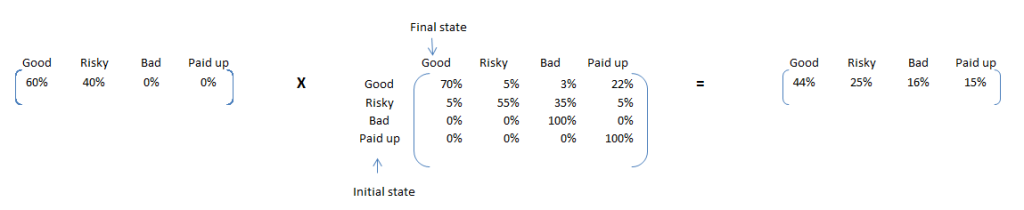
**Transition diagram**

You have done a thorough research on past trends of loan cycle and based on past trends here is the Markov chain you observed (for a period of 1 year):

[](https://www.analyticsvidhya.com/blog/wp-content/uploads/2014/07/cycle.png)

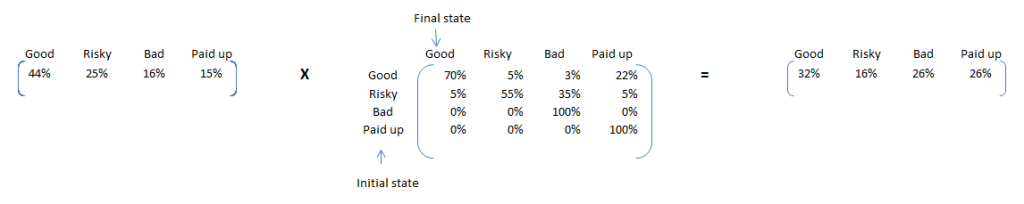
Explaining absorbing nodes becomes simple from this diagram. As you can see, Paid up and bad loans transition only to themselves. Hence, whatever path a process takes, if it lands up to one of these two states, it will stay there for ever. In other words, a bad loan cannot become paid up, risky or good loan ever after. Same is true with paid up loans.

**Transition Calcs:** Once, we have  1 year transition probability, we can convert prediction algorithm to simple matrix multiplication. Currently the portfolio has 60% Good and 40% Risky loans. How many of these loans will be finally paid up in full? Using 1 year transition probability, we can estimate the number of loans falling into each of the four bucket 1 year down the line.

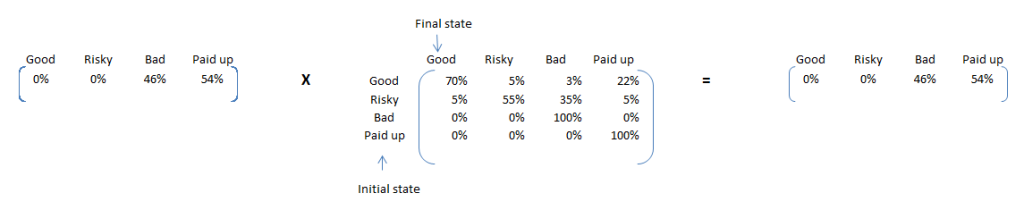
[](https://www.analyticsvidhya.com/blog/wp-content/uploads/2014/07/trans3.png)

Here are some interesting insights from this calculation. We can expect 15% of the loans to be paid up in this year and 16% being ending up as bad loans. As the % of bad loans seem to be on a higher side, it will be beneficial to identify these loans and make adequate interventions. Pin pointing to these 15% is not possible using simple Markov chain, but is possible using a Latent Markov model.

Now, to make a prediction for 2 years, we can use the same transition matrix. This time the initial proportions will the final proportions of last calculation. Transition probability generally do not change much. This is because, it is based on several time points in past.

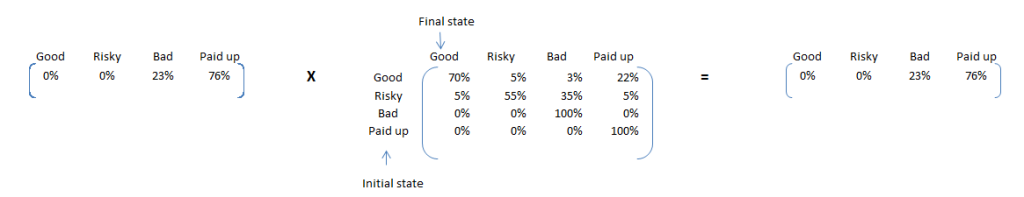
[](https://www.analyticsvidhya.com/blog/wp-content/uploads/2014/07/trans4.png)

 If we keep on repeating this exercise, we see the proportion matrix converges. Following is the converged matrix. Note that, multiplying it with transition matrix makes no change to proportions.

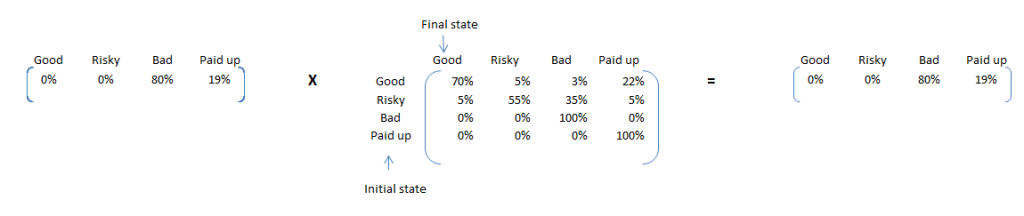
[](https://www.analyticsvidhya.com/blog/wp-content/uploads/2014/07/trans5.png)

These are the proportion numbers we were looking for. 54% of the current loans will be paid up in full but 46% will default. Hence doing this simple exercise lead us to such an important conclusion that the current portfolio exposes bank to a very high risk.

We have already seen the stationary point proportions for the portfolio. Something which will be of interest to us next is that what proportion of Good loans land up being paid up in full. For this we can start with an initial proportion split of Good – 100% and rest – 0%. The final converged matrix is as follows:

[](https://www.analyticsvidhya.com/blog/wp-content/uploads/2014/07/trans6.png)

Here are some interesting insights. If the entire portfolio was built of good loans, only 23% of loans would have defaulted against 46% for current portfolio. Hence, we will expect a very high proportion of risky loans will show default. We can find this using a simple transition calculation using Risky – 100% and rest – 0%. Following is the final converged matrix :

[](https://www.analyticsvidhya.com/blog/wp-content/uploads/2014/07/trans7.png)

80% of such loans will default. Hence, our classification of Risky and Good separates out propensity to default pretty nicely. In this article, we saw how Markov chain can be used to find out multiple insights and make good predictions on an overall level. There are many other processes which can be explained using Markov chain. In such cases, Markov chain algorithm will give you number of insights and will serve as a very handy forecasting tool. However, Markov chain can only make forecast on segment level and not make prediction on customer level. In case you need to make customer level forecast, you need a Latent Markov model and not a simple Markov model.

**How to interpret hidden state in Latent Markov Model** Simple Markov model cannot be used for customer level predictions, because it does not take into account any covariates for predictions. Latent Markov model is a modified version of the same Markov chain formulation, which can be leveraged for customer level predictions. “Latent” in this name is a representation of “Hidden states”. In this article, our focus will not be on how to formulate a Latent Markov model but simply on what do these hidden state actually mean. This is a concept which I have found quite ambiguous in the web world and too much statistics to understand this simple concept. In this article, I will try to illustrate physical interpretation of this concept “Hidden state” using a simple example.

[](https://www.analyticsvidhya.com/blog/wp-content/uploads/2014/07/Markov-chain-banner.png)

**Case Background**

A prisoner was trying to escape. He was told that he will be sent a help from outside the prison, the first day when it rains. But, he was caught having a fight with his cellmate and sentenced for stay in a dark cell for a day. He is good with probabilities and will like to make inference about the weather outside. In case he gets a probability more than 50% of the day being rainy, he will make a move else will not attract attention unnecessarily. The only clue he gets in the dark cell is the accessories, which the policeman carries while coming to the cell. Given that the policeman carries Food plate wrapped in polythene 25% of times, Food plate in packed container 25% times and open food plate 50% of times; what is the probability that it will rain the same day when the prisoner is in the dark cell?

**Using case to build analogies**

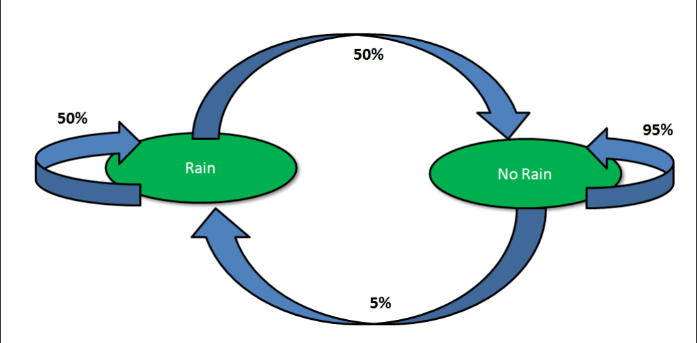
In this case we have two key events. First event is “what accessories does the policeman carry” and second event is that “it will rain on the day when the prisoner is in the dark cell”.

What accessories does the policeman carry : ***Observation or Ownership***

it will rain on the day when the prisoner is in the dark cell : ***Hidden state***

Hidden state and Ownership are commonly used terms in LMM model. The observation is something the prisoner can see and accurately determine at any point of time. But the event of raining the day when he is in dark cell is something which he can only infer and not state with 100% accuracy.

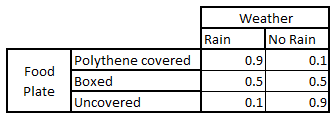
**Calculations:** aving understood the concept of hidden states, let’s crunch some numbers to come up with the final probability of it raining on the day prisoner is in the dark cell. Prisoner being anxious for last few days about the weather was noting the weather for last few months. Based on these sequence, he has make a Markov chain for the weather next day given the weather of that day. Following is how the chain looks like :

[](https://www.analyticsvidhya.com/blog/wp-content/uploads/2014/07/Markov-chain.png)

The prisoner knows that it didn’t rain yesterday (Obviously, otherwise he would not have been in jail anymore). If he uses the Markov chain directly, he can conclude with some accuracy whether it will rain today or not. Following is the formulation for such a calculation :

P(Rain today/No Rain yesterday)= 5%

Hence, the chances seem really low that it is raining out today. Now, let’s bring in some amount of information on the observation or ownership. Using some good judgement, the prisoner already knows the following conditional probability Matrix :

[](https://www.analyticsvidhya.com/blog/wp-content/uploads/2014/07/prob_grid1.png)

Let’s take one cell to clarify the grid. The chances are 90% that it is raining today if we already know that the policeman is carrying the food plate with a polythene without taking into account the weather of last day. The prisoner is keenly waiting for the policeman to come and give the final clue to determine the final set of probability. The policeman actually brings in food with a polythene. Before making calculations, let’s first decide the set of events.

A : It will rain today

B: It did not rain yesterday

C: The  policeman brings in food with a polythene

What we want to calculate is ***P(A/B,C)***? Now let’s look at the set of probabilities we know :

P(A/B) = 5%         P(C/A) = 90%      P(C) = 25%

We now will convert the expression P(A/B,C) into these know 3 parameters.

P(A/B,C) = P(A,B/C)/P(B/C) = P(A,B/C)/P(B) {Using Markov first order principle} …………………………1

P(A,B/C) = P(A,B,C)/P(C) = P(C/A,B)\*P(A,B)/P(C) = P(C/A)\*P(A,B)/P(C) {Using Markov first order principle}

=> P(A,B/C) = P(C/A) \* P(A/B)\*P(B)/P(C)

Substituting this in equation 1,

***P(A/B,C) = P(C/A) \* P(A/B) / P(C) = 90%\*5%/25% = 18%***

**Final inferences**

***P(It will rain today/no rain yesterday,policeman brings in food with a polythene) = 18%***

As you can see, this probability is between 5% and 90% as estimated  separately by the two clues we have for prediction. Combination of both the clues reveals a more accurate prediction of the event in focus. Because this probability is less than 50%, the prisoner will not take a chance expecting a rain today.**U**Using Markov chain simplifications , observations and Markov chain transition probability we were able to find out the hidden state for the day when prisoner was in the dark cell. The scope of this article was restricted to understanding hidden states and not framework of Latent Markov model. In some of the future article we will also touch up on formulation of Latent Markov model and its applications.

**SOURCE:** Tavish Srivastava <https://www.analyticsvidhya.com/blog/2014/07/interpret-hidden-state-latent-markov-model/>