Understanding Markov Chains: Predicting the Unpredictable

1. Introduction

Have you ever wondered how your phone manages to predict the next word you are about to type, or how Google seems to ‘know’ what your next search is going to be, after only typing a couple of words? A lot of things in life seem to be completely random, for example, weather changes, the arrangement of a deck of cards, even text prediction on mobile phones seems to be impossible to guess. After all, human language is random, isn’t it? What if I were to say that there was a way to predict the next state (e.g. whether the weather is rainy or sunny) based solely on the current state? This method is called a ‘Markov chain’, and in order to explain what this is, we must first explore how they were born out of a feud between two Russian mathematicians: Pavel Nekrasov and Andrey Markov. A feud, which later was recognised as the origin of a mathematical breakthrough.

2. The Russian Feud

The 1905 Russian Revolution was a nationwide uprising against Tsar Nicholas II that deeply divided the country, even splitting mathematicians Andrey Markov and Pavel Nekrasov. Nekrasov was a deeply religious man, who believed that maths could be used to explain spiritual concepts such as free will and the will of God. However, Markov, also known as ‘Andrey the Furious’, was outraged at this idea and publicly criticised Nekrasov’s work. *[1] It is important to understand here that, for 200 years, probability theory had relied upon the key assumption that you need independence to observe the Law of Large Numbers*. Nekrasov even took this one step further, claiming that if you can see the Law of Large Numbers, then you can infer that the underlying events must be independent, meaning that the events do not affect one another. *[1] As you might expect, Markov once again disagreed with this and became determined to prove that dependent events could also follow the Law of Large Numbers. Here is where the maths really begins…

*The Law of Large Numbers states that if you repeat an experiment independently a large number of times and average the results, what you obtain should be close to the expected value.

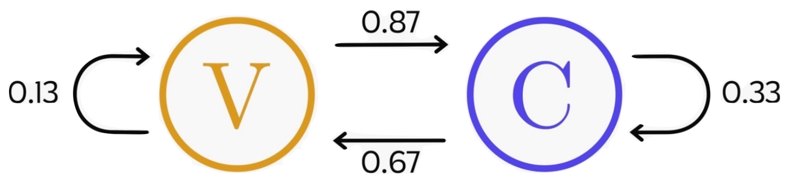
3. Markov’s Breakthrough

Markov realised that in text each letter is dependent on the letter before it, and he called these letters ‘states’. In an attempt to prove this dependency, Markov decided to analyse the vowel-consonant patterns in Alexander Pushkin’s poem ‘Eugene Onegin’. After removing all punctuation and spaces, he combined the letters into one long string: 43% were vowels, and 57% were consonants. Markov then broke this string into overlapping pairs so that he could analyse the different combinations and calculate their probabilities. The four possible combinations were vowel-vowel, consonant-vowel, vowel-consonant and consonant-consonant. *[1]

If these letters were independent, the probability of a vowel-vowel pairing would be $0.43 \times 0.43 = 0.185$. However, this was of course not the case. When Markov counted the exact number of vowel-vowel pairings, they accounted for only 6% of the total, showing that the letters were dependent. Markov now just had to prove that these dependent letters still followed the Law of Large Numbers. In doing so, he created a ‘prediction machine’*, which shows the probability of the next letter being a vowel or a consonant given the current letter. For example, if the current letter is a vowel, then the probability of the next letter being a consonant is 0.87.

As more and more letters are predicted using this machine, a pattern begins to emerge. The probabilities converge to fixed values - 43% vowels and 57% consonants. Markov had done it - he had proved the Law of Large Numbers without independence, shattering Nekrasov’s argument.*[1] To end his paper, he wrote, “thus free will is not necessary to probability”.*[3] This finale to the Russian feud introduced one of the most useful ideas in statistics. Did you guess it? Markov’s ‘predictive machine’ was later recognised as the ‘Markov chain’.

*[1]*Figure 1: screenshot from *The Strange Math That Predicts (Almost) Anything* (Veritasium, 2025, 7:42)



4. The Markov Chain

The Oxford reference defines a Markov chain as “a sequence of discrete random variables such that each member of the sequence is probabilistically dependent only on its predecessor” (Oxford University Press, 2023). *4 Let’s break that down, and since we Brits just love the weather, we’ll use that as our example.

*2

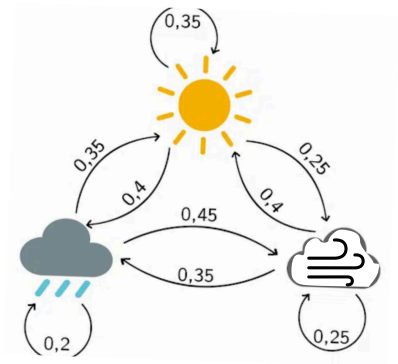


Figure 2: Adapted from *Markov Models: Get Ready* (Loth, 2023)

Here is a simple Markov chain that connects three dependent events. Let’s say that on any given day, the weather will be sunny, rainy, or windy (it cannot be multiple, even though that’s the reality living in England!). Now let’s have a look at how we can predict the weather for tomorrow given the weather today. To use the correct terminology, the weighted arrows (meaning the arrows have a value attached) originate from the current state* and point towards the future state. Each arrow is called a transition, from one state to the other. This leads us to the first property of a Markov chain, known as the ‘Markov property’: the future state depends only on the current state and not on any previous states. We can write this mathematically:

$P(X_{n+1} = x \mid X_1 = x_1 , X_2 = x_2 , \ldots, X_n = x_n)$ -> Here we can ignore all the values leading up to $X_n = x_n$, therefore,
$P(X_{n+1} = x \mid X_n = x_n)$

For example, if it was rainy two days ago, but sunny today, the probability of it being sunny tomorrow is 0.35. Now that we’ve covered what a basic Markov chain is, and what a very simple one looks like, we can hopefully link this back to our good friend the Law of Large Numbers. If a computer were to predict the weather for 100,000 days using this Markov chain, and we then counted how many of those days were rainy, windy, or sunny, the proportions of each type of weather would begin to stabilise. Over time, these proportions would converge to fixed values, representing the long-term behaviour of the system.

*By ‘state’ I am referring to a day that is sunny, rainy, or windy.

5. Transition Matrices

When we start to introduce more variables into a Markov chain, its basic diagram begins to look very complicated and difficult to work with. To represent a Markov chain more efficiently, we can use a transition matrix. To help us better understand what a transition matrix is and how it is used by computers, we can use our trusty Markov chain for weather:

$$\begin{bmatrix} 0.35 & 0.4 & 0.25 \\ 0.35 & 0.2 & 0.45 \\ 0.4 & 0.35 & 0.25 \end{bmatrix}$$

Here is a transition matrix for the same Markov diagram we used above. It shows exactly the same information, but in a way that computers can easily process. Each row represents the current state, and each column represents the next state. For example, the top middle value (0.4) represents the transition probability from sunny to rainy.

Now comes the interesting part: how do we actually use these transition matrices? We can answer this by asking another question: what is the probability of reaching state j from state i after exactly n steps? *5 To find the probability of going from one state to another in two steps, we must consider all possible intermediate states and add their probabilities together. Matrix multiplication performs this process automatically, and repeating it allows us to calculate probabilities over many steps. Therefore, instead of calculating each step individually, we can use powers of the transition matrix to perform this over many steps. The probability is given by:

$P_{ij}(n) = A^n_{ij}$

6. Application of Markov Chains

Markov chains are used in many surprising ways, such as in text generation. Firstly, your original text must be converted into a Markov chain. This is done by recording which words are most likely to come after other words, and assigning the transition between two different words a probability. Essentially, every word is 'connected' to possible next words with varying probabilities. This process is very similar to what Markov did when he analysed the poem 'Eugene Onegin'; whilst he used vowels and consonants to generate a pattern, we use word patterns to do the same thing. These sentences often sound like the original text that was used because they are following the same pattern of words. A common example to prove this is using a piece of Shakespearean text. If you entered the play 'Macbeth' (which is my personal favourite), it might have 'shall' 65% likely to follow 'thou' along with 'is' at 20%, 'not' at 10%, and so on. *[6] Over time, these patterns start to repeat, meaning that the style will stay consistent.

However, there are two clear limitations to using Markov chains in text generation. Firstly, the sentences will not always be grammatically correct, as the next word is selected based on probabilities, not what would make the most sense. Secondly, if a large number of sentences are generated, the frequencies will begin to stabilise and the same pattern will repeat. This is exactly what we saw earlier due to the Law of Large Numbers. Nonetheless, this is a very interesting and exciting use of Markov chains, and this has paved the way for more complete text generators that are now used in AI systems such as ChatGPT.

7. Conclusion

We really should be thanking Nekrasov for infuriating Markov and sparking a mathematical breakthrough, even if it didn't seem that important at the time. I hope that I have clearly explained what a Markov chain is, and that I have sparked your interest in how else Markov chains are used in the real world. However, there are quite a few more details that are yet to have concrete answers. For example, if randomness is seen as a state of unpredictability, then how are we able to model it and predict the outcome of a random event? Surely this means that the event was never random in the first place? Maybe you have the answer, and if you do, please be sure to share it with the rest of us. Thank you for taking the time to read my essay, and I hope you found it thought-provoking.

Referencing:

*[1] : Veritasium (2025) *The Strange Math That Predicts (Almost) Anything*. Available at: https://www.youtube.com/watch?v=KZeIEiBrT_w (Accessed: 06/04/2026)

*[2] : Nathan Loth (2023) *Markov Models: Get Ready*. Available at: https://liora.io/en/markov-models-get-ready (Accessed: 06/04/2026)

*[3] : A. A. Markov (1906). Extension of the law of large numbers to quantities, depending on each other. *Journal Électronique d'Histoire des Probabilités et de la Statistique.* Available at: ve42.co/lawlargenum (Accessed: 06/04/2026)

*[4] : Oxford University Press (2023). *Markov Chain* . Available at: www.oxfordreference.com (Accessed: 06/04/2026)

*[5] : Normalized Nerd (2020) *Markov Chains: n-step Transition Matrix* | *Part 3*. Available at: https://www.youtube.com/watch?v=Zo3ieESzr4E (Accessed: 06/04/2026).

*[6] : Gregory Pernicano (2021) *Text Generation with Markov Chains: An Introduction to using Markovify.* Available at: referencing Text Generation with Markov Chains: An Introduction to using Markovify: https://medium.com/data-science/text-generation-with-markov-chains-an-introduction-to-using-markovify-742e6680dc33 (Accessed: 06/04/2026)