Markov Chains and Substitution Ciphers

Tesleem Mutairu

August 8, 2024

1 Introduction to Markov Chains

1.1 Definition and Basic Concepts

A Markov chain is a stochastic process that satisfies the Markov property: the probability of transitioning to any particular state depends solely on the current state and not on the sequence of events that preceded it.

Formally, let $\{X_n : n \geq 0\}$ be a sequence of random variables taking values in a countable state space S. The process is a Markov chain if for any states $i_0, \ldots, i_n, j \in S$ and any $n \geq 0$:

$$P(X_{n+1} = j \mid X_n = i_n, X_{n-1} = i_{n-1}, \ldots, X_0 = i_0) = P(X_{n+1} = j \mid X_n = i_n)$$

This conditional probability is denoted as p_{ij} and is called the transition probability from state i to state j.

1.2 Transition Matrix

For a Markov chain with finite state space $S = \{1, 2, \ldots, m\}$, the transition matrix P is:

$$P = [P_{ij}]_{i,j=1}^{m}$$

where $P_{ij} = P(X_{n+1} = j \mid X_n = i)$ for all $n \geq 0$.

The matrix P is a square matrix of size $m \times m$. Each element P_{ij} represents the probability of transitioning from state i to state j. The notation $[P_{ij}]_{i,j=1}^{m}$ is a concise way to represent the entire matrix, indicating that i and j both range from 1 to m.

Theorem 2.2 (Properties of Transition Matrices): Let P be the transition matrix of a Markov chain. Then:

1. P^n is also a transition matrix for all $n \geq 1$.
2. The (i, j)-th entry of P^n, denoted $p_{ij}^{(n)}$, gives the probability of moving from state i to state j in exactly n steps.

Properties of the transition matrix:

1. $P_{ij} \geq 0$ for all i, j.
2. $\sum_{j=1}^{m} P_{ij} = 1$ for all i.

Example 1: Consider a weather model with three states: Sunny (S), Rainy (R), and Cloudy (C). A possible transition matrix could be:

$$P = \left[\begin{array}{c|ccc} & \text{Sunny} & \text{Rainy} & \text{Cloudy} \\ \hline \text{Sunny} & 0.7 & 0.2 & 0.1 \\ \text{Rainy} & 0.3 & 0.5 & 0.2 \\ \text{Cloudy} & 0.2 & 0.3 & 0.5 \end{array}\right]$$

Here, $P_{12} = 0.2$ means there's a 20% chance of transitioning from Sunny to Rainy.

1.3 Chapman-Kolmogorov Equations

1.3.1 What are Chapman-Kolmogorov Equations?

The Chapman-Kolmogorov equations are a set of identities that describe the relationship between transition probabilities in a Markov chain over different time intervals.

The Chapman-Kolmogorov equations provide a method to calculate multi-step transition probabilities.

For a Markov chain with transition matrix P, the Chapman-Kolmogorov equations state that the probability of moving from state i to state j in $m+n$ steps is equal to the sum of the probabilities of moving from i to any intermediate state k in m steps, and then from k to j in n steps. Mathematically, this is expressed as:

$$P_{ij}^{(m+n)} = \sum_{k=1}^{m} P_{ik}^{(m)} P_{kj}^{(n)}$$

where $P_{ij}^{(m)}$ represents the probability of transitioning from state i to state j in m steps.

The term $P_{ij}^{(m+n)}$ represents the probability of transitioning from state i to state j in $m+n$ steps. This means that you first take m steps from state i and then take n more steps to reach state j.

The summation $\sum_{k=1}^{m} P_{ik}^{(m)} P_{kj}^{(n)}$ involves summing over all possible intermediate states k. The idea is that to move from state i to state j in $m+n$ steps, you could first move from i to some intermediate state k in m steps (represented by $P_{ik}^{(m)}$), and then move from k to j in n more steps (represented by $P_{kj}^{(n)}$).

These equations are crucial because they allow us to compute multi-step transition probabilities by considering all possible intermediate states, effectively breaking down a complex transition into simpler ones.

1.3.2 Real-life Illustration of Chapman-Kolmogorov Equations

Let's consider a real-life scenario to illustrate the Chapman-Kolmogorov equations: a simple model of a person's daily commute between home (H), work (W), and a coffee shop (C).

Suppose we have the following one-step transition matrix:

$$P = \begin{bmatrix} 0.7 & 0.2 & 0.1 \\ 0.4 & 0.5 & 0.1 \\ 0.3 & 0.3 & 0.4 \end{bmatrix} = \left[\begin{array}{c|ccc} & H & W & C \\ \hline H & 0.7 & 0.2 & 0.1 \\ W & 0.4 & 0.5 & 0.1 \\ C & 0.3 & 0.3 & 0.4 \end{array}\right]$$

Now, let's say we want to know the probability of being at work two steps (days) from now, given that we're currently at home. We can use the Chapman-Kolmogorov equations to calculate this:

$$P_{HW}^{(2)} = P_{HH}^{(1)}P_{HW}^{(1)} + P_{HW}^{(1)}P_{WW}^{(1)} + P_{HC}^{(1)}P_{CW}^{(1)} = (0.7\times0.2)+(0.2\times0.5)+(0.1\times0.3) = 0.14+0.10+0.03 = 0.27$$

This means there's a 27% chance of being at work two days from now, given that we're currently at home. The Chapman-Kolmogorov equations allowed us to consider all possible paths: staying at home then going to work, going to work and staying there, or going to the coffee shop and then to work.

2 Advanced Concepts in Markov Chains

2.1 Stationary Distribution

A probability vector $\pi = (\pi_1, \ldots, \pi_m)$ is a stationary distribution if:

$$\pi P = \pi$$

This means that if the chain starts in the stationary distribution, it will remain in that distribution after any number of steps.

Theorem 2.4 (Existence and Uniqueness of Stationary Distribution): For an irreducible and aperiodic Markov chain, there exists a unique stationary distribution π, and for any initial distribution μ:

$$\lim_{n\to\infty} \mu P^n = \pi$$

The notation $\lim_{n\to\infty}$ means that we are interested in what happens as n becomes very large — essentially, as the number of steps in the Markov chain goes to infinity.

As the number of steps increases, the probability distribution of a Markov chain's states will eventually stabilize and converge to a fixed distribution, regardless of the initial starting point.

Example 2:

Solved Example: Weather Markov Chain

Let's consider a simple weather model as a Markov chain. Suppose we have three states: Sunny (S), Cloudy (C), and Rainy (R). The transition matrix is given by:

$$P = \begin{bmatrix} 0.7 & 0.2 & 0.1 \\ 0.3 & 0.4 & 0.3 \\ 0.2 & 0.3 & 0.5 \end{bmatrix} =$$

	S	C	R
S	0.7	0.2	0.1
C	0.3	0.4	0.3
R	0.2	0.3	0.5

Problem: Find the stationary distribution of this Markov chain.

Solution: To find the stationary distribution $\pi = (\pi_S, \pi_C, \pi_R)$, we need to solve the equation $\pi = \pi P$ along with the constraint that $\pi_S + \pi_C + \pi_R = 1$.

This gives us the system of equations:

$$\pi_S = 0.7\pi_S + 0.3\pi_C + 0.2\pi_R$$

$$\pi_C = 0.2\pi_S + 0.4\pi_C + 0.3\pi_R$$

$$\pi_R = 0.1\pi_S + 0.3\pi_C + 0.5\pi_R$$
$$\pi_S + \pi_C + \pi_R = 1$$

Solving this system (using substitution or matrix methods), we get:

$$\pi_S \approx 0.4545, \quad \pi_C \approx 0.3182, \quad \pi_R \approx 0.2273$$

Interpretation: In the long run, it will be Sunny about 45.45% of the time, Cloudy 31.82% of the time, and Rainy 22.73% of the time, regardless of the initial weather condition.

Verification: We can verify this by multiplying π by P:

$$[0.4545 \quad 0.3182 \quad 0.2273] \cdot \begin{bmatrix} 0.7 & 0.2 & 0.1 \\ 0.3 & 0.4 & 0.3 \\ 0.2 & 0.3 & 0.5 \end{bmatrix} \approx [0.4545 \quad 0.3182 \quad 0.2273]$$

This example illustrates how Markov chains can model real-world phenomena and how the stationary distribution provides long-term predictions.

3 Ergodicity

A Markov chain is ergodic if it is both irreducible (it's possible to get from any state to any other state) and aperiodic (the chain doesn't get stuck in cycles).

For an ergodic Markov chain, the long-term probabilities converge to the stationary distribution regardless of the starting state:

$$\lim_{n\to\infty} P_{ij}^n = \pi_j$$

The expression

$$\lim_{n\to\infty} P_{ij}^n = \pi_j$$

states that as the number of steps n approaches infinity, the probability that the Markov chain is in state j (starting from any initial state i) converges to π_j. This means that in the long run, the Markov chain "forgets" its initial state and the probability of being in any particular state j is given by the stationary distribution π_j. This result holds under certain conditions, such as the Markov chain being irreducible and aperiodic.

4 Mean First Passage Time

The mean first passage time m_{ij} is the expected number of steps to reach state j for the first time, starting from state i. It can be calculated using the following equation:

$$m_{ij} = 1 + \sum_{k\neq j} P_{ik} m_{kj}$$

where:

m_{kj} : This is the expected number of steps required to reach state j starting from state k.

The summation

$$\sum_{k\neq j} P_{ik} m_{kj}$$

takes into account all the possible states k that the Markov chain could transition to from state i (excluding state j), weighted by the probability of moving to each state k from state i.

5 Substitution Ciphers

5.1 Definition and Types

A substitution cipher is a method of encryption where units of plaintext are replaced with ciphertext according to a fixed system. The "units" may be single letters, pairs of letters, triplets of letters, mixtures of the above, and so forth.

Types of substitution ciphers include:

1. Simple substitution: Each letter is replaced by another letter.
2. Homophonic substitution: Each letter can be replaced by multiple symbols.
3. Polyalphabetic substitution: Multiple substitution alphabets are used.
4. Polygraphic substitution: Groups of letters are encoded together.

5.2 Mathematical Formulation

Let P be the plaintext alphabet and C be the ciphertext alphabet. A simple substitution cipher can be defined as a bijective function $f : P \to C$.

Encryption Process:

$$\text{Plaintext message } m = m_1 m_2 \dots m_n$$

This represents the original message that you want to encrypt, where $m_1, m_2, \dots, m_n$ are the individual characters of the message.

Encryption:

$$c = f(m) = f(m_1)f(m_2)\dots f(m_n)$$

The encryption is done by applying the function f to each character of the plaintext message m. Each character m_i in the plaintext is substituted with the corresponding character $f(m_i)$ in the ciphertext. The result is the ciphertext c, which is the encoded version of the original message.

Decryption Process:

$$\text{Inverse function } f^{-1} : C \to P$$

Since f is bijective, it has an inverse function f^{-1} that maps each character from the ciphertext alphabet C back to its corresponding character in the plaintext alphabet P.

Decryption: To decrypt the ciphertext c and recover the original message m, you apply the inverse function f^{-1} to each character of the ciphertext. This reverses the encryption process, yielding the original plaintext message.

5.3 Theorem 3.1 (Permutation Group of Substitution Ciphers)

The set of all possible simple substitution ciphers on an alphabet of size n forms a symmetric group S_n of order $n!$.

Proof:

1. **Closure:** The composition of two substitutions is a substitution.
2. **Associativity:** Function composition is associative.

3. **Identity:** The identity permutation (leaving the alphabet unchanged) exists.

4. **Inverse:** Each substitution has an inverse (the decryption function).

These properties define a group. The order is $n!$ because there are $n!$ ways to permute n distinct objects.

5.4 Cryptanalysis: Frequency Analysis

Let $f_p(x)$ be the frequency of letter x in the plaintext language, and $f_c(y)$ be the frequency of letter y in the ciphertext. We aim to find a mapping that minimizes:

$$D = \sum_{x,y \in A} |f_p(x) - f_c(y)|$$

Components:

$f_p(x)$: This represents the frequency of the letter x in the plaintext language.

For example, in English, certain letters like 'E', 'T', and 'A' occur more frequently than others like 'Q' or 'Z'. These frequencies are well known and can be used as a reference.

$f_c(y)$: This represents the frequency of the letter y in the ciphertext.

Since the ciphertext is a result of substituting plaintext letters, the frequency of each letter in the ciphertext will correspond to the frequency of its corresponding plaintext letter.

A : This is the set of all possible letters in the alphabet (e.g., A-Z for English).

Objective: The goal of frequency analysis in cryptanalysis is to find a mapping from the ciphertext letters y back to the plaintext letters x that likely minimizes the difference between the known plaintext frequencies $f_p(x)$ and the observed ciphertext frequencies $f_c(y)$.

Expression Explained:

$$D = \sum_{x,y \in A} |f_p(x) - f_c(y)|$$

D is a measure of how different the frequency distribution of the letters in the ciphertext is from the expected frequency distribution in the plaintext.

The sum $\sum_{x,y \in A}$ runs over all possible pairs of plaintext letters x and ciphertext letters y.

$$|f_p(x) - f_c(y)|$$

represents the absolute difference between the frequency of a plaintext letter x and the frequency of a ciphertext letter

Interpretation: Minimizing D: The objective is to find a mapping between the ciphertext and plaintext letters that minimizes the total difference D. A smaller D means that the frequency distribution of the letters in the ciphertext closely matches the known

distribution in the plaintext, which suggests that the mapping (or substitution) is likely correct.

Cryptanalysis Strategy: In practice, cryptanalysts compare the frequency of letters in the ciphertext with typical frequencies in the plaintext language. By matching the most frequent letters in the ciphertext to the most frequent letters in the plaintext, they can start guessing the substitution pattern. For example, if 'E' is the most common letter in English, and 'X' is the most common letter in the ciphertext, it's likely that 'X' corresponds to 'E'.

5.5 Index of Coincidence (IC)

The Index of Coincidence measures the unevenness of letter frequencies:

$$IC = \frac{\sum_{i=1}^{m} n_i(n_i - 1)}{n(n-1)}$$

Where n is the text length and n_i is the count of the i-th letter.

Explanation:

$$\text{Numerator } \sum_{i=1}^{m} n_i(n_i - 1) :$$

This summation calculates the sum of the products $n_i(n_i - 1)$ for each letter i in the alphabet. $n_i(n_i - 1)$ is related to the number of ways to choose 2 occurrences of the i-th letter from its total occurrences in the text. This helps measure how often pairs of identical letters occur.

$$\text{Denominator } n(n-1) :$$

This is the total number of ways to pick 2 characters from the entire text. It normalizes the numerator by considering the total text length, making the IC a relative measure.

Interpretation:

- **High IC:** A high Index of Coincidence indicates that the text has an uneven distribution of letters, meaning some letters appear more frequently than others. This is typical of natural language texts (e.g., English) because certain letters like 'E', 'T', and 'A' are more common.
- **Low IC:** A low Index of Coincidence suggests a more uniform distribution of letters, which might occur in ciphertexts generated by substitution ciphers where the frequency distribution of the letters is altered.

Typical Values:

- In English plaintext, the IC is typically around 0.068.
- For a random text or text encrypted with a simple substitution cipher, the IC is lower, around 0.038.

6 Connection between Markov Chains and Substitution Ciphers

6.1 Language Modeling with Markov Chains

We can model language as a Markov chain where states represent letters. The transition matrix $P = [P_{ij}]$ gives the probability of letter j following letter i. This means that each entry P_{ij} in the matrix represents how likely it is for the letter i to be followed by the letter j.

Example 3: Consider a simplified English model with only four letters: A, E, T, S. A possible transition matrix might be:

$$P = \begin{bmatrix} 0.3 & 0.4 & 0.2 & 0.1 \\ 0.2 & 0.1 & 0.5 & 0.2 \\ 0.3 & 0.3 & 0.1 & 0.3 \\ 0.2 & 0.3 & 0.4 & 0.1 \end{bmatrix}$$

This matrix captures common letter patterns in English, like "EA" and "ST". For example, the probability of 'E' following 'A' is 0.4, while the probability of 'S' following 'A' is 0.1.

6.2 Using Markov Chains to Decrypt Substitution Ciphers

The connection between Markov chains and language modeling can be exploited to break substitution ciphers. Recall that a substitution cipher is defined by a bijective function $f : A \to A$, where A is the alphabet. For a plaintext $p = p_1 p_2 \dots p_n$, the ciphertext $c = c_1 c_2 \dots c_n$ is:

$$c_i = f(p_i) \quad \text{for } i = 1, \dots, n$$

This means that each letter in the plaintext is replaced by another letter in the ciphertext according to the function f.

To use Markov chains to decrypt a substitution cipher, we need to find the substitution mapping f that best aligns the ciphertext transitions with the expected transitions in the target language.

Let's define the following:

- P_L: The transition matrix for the language model (e.g., English)
- P_C: The observed transition matrix in the ciphertext

Our goal is to find a permutation matrix Q that represents the substitution key, such that:

$$QP_CQ^T \approx P_L$$

Where Q^T is the transpose of Q. The permutation matrix Q maps the ciphertext transitions to the expected transitions in the target language. Essentially, we want to adjust P_C using Q so that it resembles P_L as closely as possible.

We can formulate this as an optimization problem:

$$\min_Q \|QP_CQ^T - P_L\|_F$$

Where $\|\cdot\|_F$ is the Frobenius norm, which measures the difference between the two matrices. The Frobenius norm is like a way to measure the "distance" between two matrices, summing up the squares of the differences of their corresponding entries.

By minimizing this distance, we can find the substitution key Q that best aligns the ciphertext transitions with the language model transitions. This allows us to recover the original plaintext.

Example 4.2: Suppose we observe in a ciphertext that the bigram "XY" appears frequently. If our English language model shows that "th" is a common bigram, we might hypothesize that $X \to t$ and $Y \to h$ in our substitution key. This insight can guide the optimization process to find the correct key.

The use of Markov chain models, combined with optimization techniques like the Hill Climbing algorithm, provides a powerful approach for breaking substitution ciphers by exploiting the statistical properties of natural languages.

7 Hill Climbing Algorithm for Breaking Ciphers

The hill climbing algorithm described here is a heuristic method used to break substitution ciphers. The algorithm iteratively improves a candidate decryption key by making small changes and selecting the one that yields the best decryption result, as measured by a fitness function.

7.1 Steps of the Hill Climbing Algorithm

Initialization: Start with a random key k: The key k is a random permutation of the alphabet. This key represents the initial guess for how the letters in the ciphertext map to letters in the plaintext.

Iteration:

1. (a) Generate neighbor keys $N(k)$: Neighbor keys are generated by making small changes to the current key k. Typically, this involves swapping pairs of letters in k. For example, if the current key maps 'A' to 'X' and 'B' to 'Y', a neighbor key might swap these so that 'A' maps to 'Y' and 'B' to 'X'.

2. (b) For each k' in $N(k)$, compute the fitness $F(k')$: The fitness function $F(k')$ evaluates how "good" a particular key is at decrypting the ciphertext. The fitness is based on how likely the decrypted text is to resemble the target language, according to a language model.

3. (c) Update the key if a better one is found.

4. (d) Termination condition: If no neighbor key has a higher fitness than the current key, the algorithm can either terminate (indicating a local maximum has been reached) or jump to a new random key and start the process again. This jump can help escape local maxima and potentially find a better solution.

The fitness function F can be based on the likelihood of the decrypted text under the language model:

$$F(k) = \sum_{i=1}^{n-1} \log(P_{w_i w_{i+1}})$$

Where w_i is the i-th letter of the decrypted text using key k.

8 Historical Context and Real-World Applications

Substitution ciphers have a rich history dating back to ancient civilizations. One of the most famous examples is the Caesar cipher, used by Julius Caesar for military communications. In this cipher, each letter in the plaintext is replaced by a letter some fixed number of positions down the alphabet.

In modern times, substitution ciphers are still used in various contexts:

1. Educational tools: They are often used to introduce students to the concepts of cryptography.
2. Puzzle games: Many word puzzles and games use simple substitution ciphers.
3. Steganography: In some steganographic techniques, substitution ciphers are used to hide messages within seemingly innocuous text or images.

While simple substitution ciphers are no longer considered secure for sensitive communications, their study remains crucial for understanding the foundations of cryptography and for developing more advanced encryption methods.

9 Relevance of Markov Chains to Substitution Ciphers

Markov chains are relevant to substitution ciphers in several ways:

1. **Language Modeling:** Markov chains can model the transition probabilities between letters or words in a language, which is useful for both creating and breaking ciphers.
2. **Cryptanalysis:** The n-gram frequency analysis used in breaking substitution ciphers can be modeled as a Markov process.
3. **Key Generation:** Markov chains can be used to generate pseudo-random sequences for key generation in more advanced substitution ciphers.

9.1 Theorem 3 (Markov Chain Language Model)

Let L be a language modeled as a Markov chain with states representing letters and transition probabilities P_{ij} representing the probability of letter j following letter i. The probability of a word $w = w_1 w_2 \dots w_n$ in L is given by:

$$P(w) = P(w_1) \cdot P(w_2 \mid w_1) \cdot P(w_3 \mid w_2) \cdots P(w_n \mid w_{n-1})$$

Initial Probability $P(w_1)$: This is the probability of the first letter w_1 occurring. It is often determined based on the letter's frequency in the language.

Conditional Probabilities $P(w_{i+1} \mid w_i)$: For each subsequent letter w_{i+1}, this is the probability of w_{i+1} following w_i. This captures the likelihood of letter sequences, such as bigrams (pairs of letters) or trigrams (triplets of letters).

The product of these probabilities provides the overall probability of the entire sequence of letters w.

This theorem forms the basis for using Markov chains in language modeling and cryptanalysis.

10 Language Modeling with Markov Chains

10.1 n-gram Markov Models

Natural languages exhibit statistical regularities that can be modeled using Markov chains. An n-gram Markov model captures the probability of a letter occurring given the $n-1$ preceding letters.

Definition 4.1: For an n-gram Markov model of a language, the state space S consists of all possible $(n-1)$-grams, and the transition probabilities represent the likelihood of one letter following a given $(n-1)$-gram.

Formally, for a sequence of letters $L_1, L_2, \ldots, L_m$, an n-gram Markov model assumes:

$$P(L_i \mid L_1, \ldots, L_{i-1}) = P(L_i \mid L_{i-n+1}, \ldots, L_{i-1})$$

This assumption allows us to approximate the probability of a sequence of letters using only local context.

Example 4.1: Consider a bigram (2-gram) model for English. The transition probability $P(\text{"}h\text{"} \mid \text{"}t\text{"})$ represents the likelihood of "h" following "t". This captures common patterns like "th" in English.

10.2 Consistency of n-gram Models

As we increase the order of the n-gram model, we capture more context and potentially improve our approximation of the language.

Theorem 4.1 (Consistency of n-gram Models): As n increases and given sufficient data, the n-gram model becomes a more accurate representation of the language, converging to the true distribution in the limit.

Proof (Intuitive Explanation):

- Let P_n be the probability distribution defined by the n-gram model, and P be the true distribution of the language.
- As n increases, P_n captures more context and thus more accurately reflects P.
- With infinite data, we can estimate arbitrarily long contexts, allowing P_n to approach P.

This approximation can be achieved through iterative methods like the Hill Climbing algorithm discussed earlier.

Example 4.2: Suppose we observe in a ciphertext that the bigram "XY" appears frequently. If our English language model shows that "th" is a common bigram, we might hypothesize that X $\rightarrow$ t and Y $\rightarrow$ h in our substitution key.

By leveraging the statistical properties captured by Markov models, we can develop sophisticated methods for breaking substitution ciphers that go beyond simple frequency analysis.

11 Conclusion

The application of Markov chains to the cryptanalysis of substitution ciphers represents a powerful fusion of probability theory, information theory, and linguistics. We have explored the theoretical foundations of Markov chains, their application in language modeling, and their implementation in various algorithms for cipher breaking.

As we look to the future, the interplay between Markov models and emerging technologies like quantum computing and advanced machine learning promises to push the boundaries of what is possible in cryptanalysis and related fields.

The study of Markov chains in the context of substitution ciphers offers a rich tapestry of mathematical theory and practical application, standing as a testament to the power of probabilistic methods in unraveling complex patterns and extracting meaning from apparent randomness.