Final Project: Text Decryption Using MCMC

Name: Ruizhong Miao
NetID: rmiao3

Abstract:

In this final project, we demonstrate how we can use the technique of Markov Chain Monte Carlo (MCMC) to solve a text decryption problem, specifically substitution cipher problem. We also model the English text as a Markov chain. We incorporate “global move” and “local move” in our MCMC so that it is less likely to be stuck in a local optimum. We also try two types of decryption problem, namely homogeneous text decryption and heterogeneous text decryption. The result shows that our method works well in both cases.

1. Introduction:

In cryptography, one commonly used way of encoding text is the substitution cipher. The substitution cipher works by replacing each unit of plaintext by a cipher according to a fixed system. The encrypted text will be written in the new symbols, and therefore keeps the information classified. In this project, our goal is to recover the original text by finding the one-to-one mapping between the old and the new text symbols.

In practice, the substitution cipher is created by mapping one English letter to another. For example, if we map each letter to the letter following it in alphabetical order, and map the letter “z” to “a”, then the word “apple” will be written as “bqqmf”.

One way of solving the problem is to come up with a mapping and see if the resulting text makes sense. However, the number of all such possible mappings is $26! \approx 4\times10^{26}$, which is so large that enumerating all of them is unrealistic.

Fortunately, we can harness the power of Markov Chain Monte Carlo (MCMC) method. The MCMC is a class of algorithms for sampling from a probability distribution. In MCMC, we construct a Markov chain that has the desired probability distribution as its invariant distribution. In practice, the most commonly used algorithm for MCMC is perhaps the Metropolis-Hastings algorithm, in which at each step of the Markov chain, we propose a new state, and accept it with the probability described by the following formula:

$$A(x'|x) = \min\left(1, \frac{P(x')g(x|x')}{P(x)g(x'|x)}\right)$$

where x' is the proposed state, and x is the current state; $P(x)$ is the target distribution, and $g(x'|x)$ is the probability of proposing x' when we are at x. $A(x'|x)$

is the acceptance probability of x' when we are at x. Whether we accept x' or stay at x, the next state will be included in our final samples.

The thus sampled points will have distribution $P(x)$ given that the above Markov chain has a unique invariant distribution.

2. Method:

2.1 The First Markov Chain: Modeling English Text

We first model English text as a Markov chain with the state space being the letters. We also add whitespace to the state space to represent anything that is not an English letter. Therefore, the word "apple" can be considered as the following chain:

$$a \to p \to p \to l \to e$$

The probability of this chain is $\pi(a)P(p|a)P(p|p)P(l|p)P(e|l)$. We assume $\pi(X)$ to be equal for all 26 letters. Then, once we know the transition probabilities, we know, for a given piece of text, its probability.

The transition probabilities can be calculated from any unencrypted text. For example, if we want to estimate the probability $P(p|a)$, and we have a large volume of text, we can walk through the whole text and count how many times "a" is followed by "p". Then we divide this quantity by the number of times "a" appears. Then the ratio will be our estimate of $P(p|a)$. In practice, we add one to the count of each transition, so for any piece of text there is a non-zero probability.

2.2 The Second Markov Chain: Sampling Mappings Using MCMC

The second Markov chain is a chain of mappings from the ciphers to the original letters. This is the Markov chain we use in MCMC iteration, while the Markov chain in section 2.1 is only used for calculating likelihoods. Under each mapping, the encrypted text will be mapped back to its original text and the corresponding likelihood will be calculated.

Suppose at one step, the current mapping is M_0, and under M_0, the decrypted text has probability P_0. Then suppose the proposed mapping is M_1, and the probability of the corresponding decrypted text is P_1. Then, we accept M_1 as our new state with probability

$$A(M_1|M_0) = \min\left(1, \frac{P_1 g(M_0|M_1)}{P_0 g(M_1|M_0)}\right)$$

There are still two components that haven't been specified, which are $g(M_0|M_1)$ and $g(M_1|M_0)$. We implement two kind of proposals in which $g(M_1|M_0) = g(M_0|M_1)$, so the acceptance probability becomes

$$A(M_1|M_0) = \min\left(1, \frac{P_1}{P_0}\right)$$

The resulting Markov chain will have the following property: the higher the probability of the decrypted text is, the higher the probability of the corresponding mapping in the invariant distribution is. During the process of MCMC, we record the mapping that has the highest likelihood, and we output this mapping after MCMC as the final result.

2.3 Proposing New States:

The two types of proposals are:

> (1) Local Move: Given the current mapping M_0, we randomly select two letters (possibly whitespace) and switch their images in the mapping, and the new mapping is our proposed mapping M_1.
> (b) Global Move: Proposing a completely new random mapping, which can be done by proposing a permutation of the letters and whitespace.

The idea comes from Chen, etc.[1]. The motivation of using these two kinds of proposals is to reduce the probability of being stuck in a local extremum. We also define a parameter α, which is the probability we take a global move at each step. Correspondingly, $(1 - \alpha)$ is the probability we take a local move. In practice, we find that the best results are often obtained when α is around 0.02.

3. Results:

3.1 Homogeneous Text Decryption:

First, in order to estimate the transition probabilities, we download *The Adventures of Sherlock Holmes* from Project Gutenberg. The transition probability is estimated from this book.

We use a heat map to visualize the transition matrix in Figure 1.

[1] J. Zhang and Y. Chen, *Monte Carlo Algorithms for Identifying Densely Connected Subgraphs*, Journal of Computational and Graphical Statistics (2015)

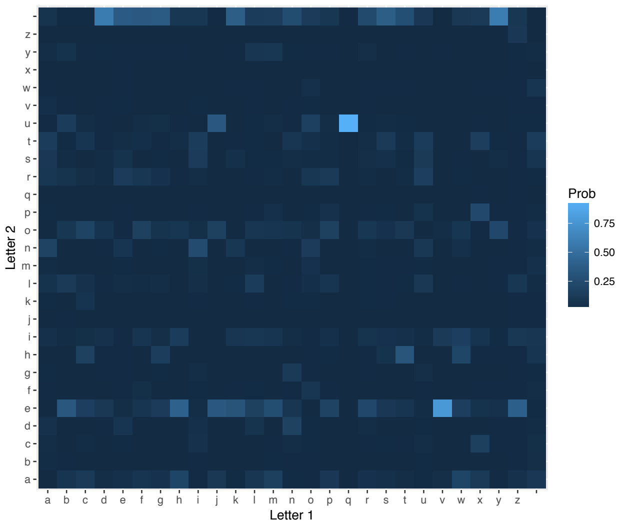

Figure 1: Transition matrix estimated from *The Adventures of Sherlock Holmes*

The highest transition probability is $P(u|q)$, which means the letter "u" follows the letter "q". This can result from the high frequency of words like "question" or "quit" etc.

Next, we pick one sentence from *The Adventures of Sherlock Holmes* and use it as an example:

IT IS AN OLD MAXIM OF MINE THAT WHEN YOU HAVE EXCLUDED THE IMPOSSIBLE WHATEVER REMAINS HOWEVER IMPROBABLE MUST BE THE TRUTH

Then, we use a random mapping to map each letter to a new one, the resulting encrypted text will be:

VM VS JR QYN LJBVL QE LVRI MWJM PWIR UQX WJGI IBDYXNIN MWI VLOQSSVHYI PWJMIGIZ ZILJVRS WQPIGIZ VLOZQHJHYI LXSM HI MWI MZXMW

At a glance, it will be very difficult for people to make sense of it. However, this problem is doable with the help of MCMC. We call this task homogeneous text decryption because the transition matrix is estimated from the same text where the encrypted text came from.

We feed the encrypted text to the MCMC iteration. The result will look like:

- Iteration: 1000:
  AT AS ON UKD WOMAW UG WANE TLOT BLEN QUI LOVE EMYKIDED TLE AWHUSSACKE BLOTEVER REWOANS LUBEVER AWHRUCOCKE WIST CE TLE TRITL
- Iteration: 2000:

  OT OS IN ULD WIMOW UP WONE THIT GHEN QUA HIVE EMYLADED THE OWBUSSOCLE GHITEVER REWIONS HUGEVER OWBRUCICLE WAST CE THE TRATH

- Iteration: 3000:
  IT IS ON ULD MOFIM UY MINE THOT GHEN QUA HOVE EFPLADED THE IMBUSSICLE GHOTEVER REMOINS HUGEVER IMBRUCOCLE MAST CE THE TRATH
- Iteration: 4000:
  IT IS AN OLY MADIM OK MINE THAT GHEN BOU HAVE EDFLUYEY THE IMPOSSICLE GHATEVER REMAINS HOGEVER IMPROCACLE MUST CE THE TRUTH
- Iteration: 5000:
  IT IS AN OLD MAYIM OK MINE THAT WHEN FOU HAVE EYBLUDED THE IMPOSSICLE WHATEVER REMAINS HOWEVER IMPROCACLE MUST CE THE TRUTH
- Iteration: 6000:
  IT IS AN OLD MAXIM OF MINE THAT WHEN YOU HAVE EXCLUDED THE IMPOSSIBLE WHATEVER REMAINS HOWEVER IMPROBABLE MUST BE THE TRUTH
- Iteration: 7000:
  IT IS AN OLD MAXIM OF MINE THAT WHEN YOU HAVE EXCLUDED THE IMPOSSIBLE WHATEVER REMAINS HOWEVER IMPROBABLE MUST BE THE TRUTH
- Iteration: 8000:
  IT IS AN OLD MAXIM OF MINE THAT WHEN YOU HAVE EXCLUDED THE IMPOSSIBLE WHATEVER REMAINS HOWEVER IMPROBABLE MUST BE THE TRUTH
- Iteration: 9000:
  IT IS AN OLD MAXIM OF MINE THAT WHEN YOU HAVE EXCLUDED THE IMPOSSIBLE WHATEVER REMAINS HOWEVER IMPROBABLE MUST BE THE TRUTH
- Iteration: 10000:
  IT IS AN OLD MAXIM OF MINE THAT WHEN YOU HAVE EXCLUDED THE IMPOSSIBLE WHATEVER REMAINS HOWEVER IMPROBABLE MUST BE THE TRUTH

We can see that after the 4000th iteration, the text begins to make sense, and we are able to recover the original text after the 6000th iteration. In practice, many trials may be needed before getting ideal results.

3.2 Heterogeneous Text Decryption:

In section 3.1, we successfully decrypted a piece of text using the transition matrix that is estimated from the same source text. However, different source text may possess different characteristics. For example, the words set of a computer manual is often different from that of a news article. In the case of text decryption, most likely we don't know where the encrypted text came from. Therefore, we need to estimate our transition matrix from another source text.

In order to investigate the impact this kind of situation can have on the result, we also download *War and Peace* (WaP), *The Adventures of Tom Sawyer* (Tom), *Adventures of Huckleberry Finn* (Huckleberry), and *The End of the Middle Ages* (Middle) from Project Gutenberg.

Figure 2 shows the transition matrices estimated from these books respectively. From the figure, we can see that the four transition matrices share some common structure. For example, they all have high probability for $P(u|q)$, $P(e|v)$, and $P(" "|y)$.

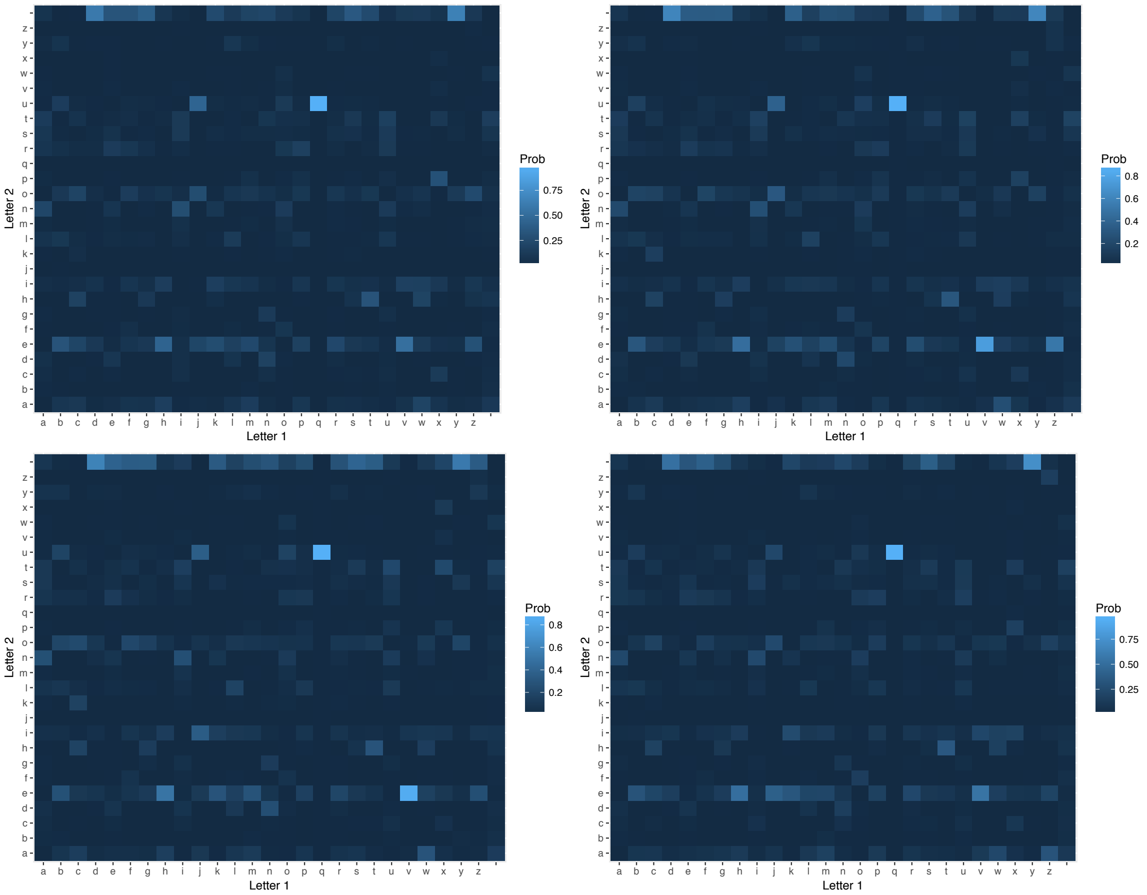

Figure 2: Transition matrices estimated from *War and Peace* (Top left), *The Adventures of Tom Sawyer* (Top right), *Adventures of Huckleberry Finn* (Bottom left), and *The End of the Middle Ages* (Bottom right).

If we treat these transition matrices as vectors, we can calculate their correlation coefficients as a measure of similarity. Figure 3 shows the pairwise correlation between these matrices.

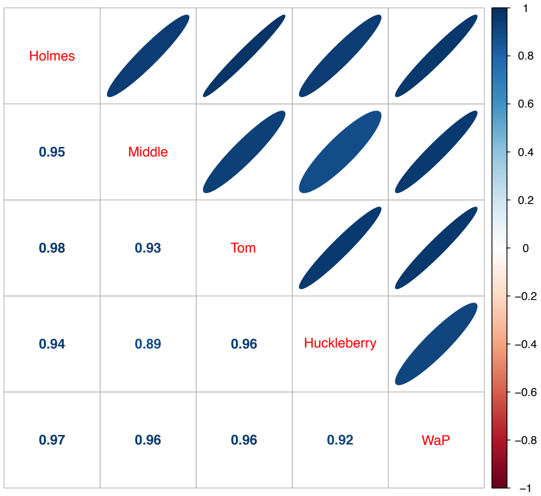

Figure 3: Correlation plot of the transition matrices

From Figure 3, we can see that except for the correlation between “Middle” and “Huckleberry”, all of the correlations are greater than 0.9, which indicates the similarity between these matrices.

For “Holmes”, the matrix with which it has the smallest correlation is “Huckleberry”. For the previous example, we use “Huckleberry” as our transition matrix, and we feed the same piece of text to our machinery of MCMC. The result will be:

- Iteration: 10000:
  AS AG IN OLY WIMAW OU WANE STIS FTEN COD TIVE EMBLDYEY STE AWHOGGAPLE FTISEVER REWIANG TOFEVER AWHROPIPLE WDGS PE STE SRDST
- Iteration: 20000:
  AT AG IN OLD RICAR OF RANE THIT WHEN BOU HIVE ECKLUDED THE ARYOGGAPLE WHITEVES SERIANG HOWEVES ARYSOPIPLE RUGT PE THE TSUTH
- Iteration: 30000:
  IT IG AN ORD MALIM OF MINE THAT WHEN BOU HAVE ELPRUDED THE IMYOGGICRE WHATEVES SEMAING HOWEVES IMYSOCACRE MUGT CE THE TSUTH
- Iteration: 40000:
  IT IS AN OLD MAYIM OF MINE THAT WHEN GOU HAVE EYBLUDED THE IMPOSSICLE WHATEVER REMAINS HOWEVER IMPROCACLE MUST CE THE TRUTH
- Iteration: 50000:
  IT IS AN OLD MAYIM OF MINE THAT WHEN GOU HAVE EYBLUDED THE IMPOSSICLE WHATEVER REMAINS HOWEVER IMPROCACLE MUST CE THE TRUTH

We can see that although we are using different source text to estimate our transition matrix, the decrypted text is still able to make sense after a number of iterations. The reason may be that different source text may give different transition matrices, but they are highly correlated, and therefore have similar entries, so for a give piece of encrypted text, their decryption results are similar.

5. Conclusion:

In this project, we have demonstrated that the substitution cipher problem can be solved by modeling English text as a Markov chain and Markov chain Monte Carlo sampling. We also found that transition matrices estimated from different source text are highly correlated, and therefore give similar decryption result. In reality, we usually don't know where the encrypted text came from, but we can estimate our transition matrix from any available source text, and this should give satisfactory result.

6. Reference:

1. Diaconis, P. *The Markov Chain Monte Carlo Revolution.* Retrieved from https://math.uchicago.edu/~shmuel/Network-course-readings/MCMCRev.pdf

2. Connor, S. *Simulation and Solving Substitution Codes.* Retrieved from http://www-users.york.ac.uk/~sbc502/decode.pdf

3. Andrew. *Text Decryption Using MCMC*. Retrieved from R-bloggers: https://www.r-bloggers.com/text-decryption-using-mcmc/

4. J. Zhang and Y. Chen, *Monte Carlo Algorithms for Identifying Densely Connected Subgraphs*, Journal of Computational and Graphical Statistics (2015): http://www.tandfonline.com/doi/abs/10.1080/10618600.2014.930040