On the Difficulty of Breaking Substitution Ciphers

Phil Wertheimer

October 14, 2021

Abstract

We analyze different methods of attacking substitution ciphers using m-gram frequency analysis. Our study includes both numerical simulations of the English language and theoretical analysis of random alphabets. We present several results in the direction of explaining why the 2-gram method performs the best in breaking the substitution ciphers.

Summary

1 Introduction

1.1 Cryptographic Background

The substitution cipher is one of the most basic forms of encryption, achieved by applying a permutation to the letters of an alphabet. For English this amounts to a permutation in S_{26}[1] which may have the following effect:

$$\begin{gathered} a \mapsto k \\ b \mapsto y \\ \dots \\ z \mapsto e \end{gathered}$$

Denoting the alphabet by $\aleph$, the encryption key is the map $\sigma : \aleph \to \aleph$ used to scramble the letters. The decryption key is the inverse map σ^{-1}. The fact that the key space S_N has $N!$ elements makes a brute force attack computationally infeasible. For example, the number of permutations of the English alphabet is $|S_{26}| = 26! > 10^{27}$. However, the structure of the English language can be exploited to find the correct key quite easily; because of this, these cryptosystems are insecure and should be avoided in practice.

The standard approach [2] to breaking substitution ciphers uses frequency analysis: one compares the frequencies of symbols occurring in the *ciphertext* (encrypted text) with the empirical frequencies of English letters (or whichever language is being used). For instance, if the symbol "k" appears most in the ciphertext, one would guess that the *plaintext* (unencrypted) symbol corresponding to "k" is "e", the letter that occurs most frequently in the English language.

This method can be enhanced by considering frequencies of *digrams*, or two letter strings. For instance, the letters "s" and "r" are both relatively common in English, but the digram "sr" is relatively uncommon. In general, one can consider *m-grams* - strings of m letters - to take advantage of the structure and patterns in English, or whichever language is being used.

An interesting implementation of this strategy was first introduced by Marc Coram and Phil Beineke in the Stanford statistical consulting service [5], and later studied more systematically by Connor [3]. As outlined by Diaconis [5], digram frequency analysis can be used as the basis for a Markov Chain Monte Carlo technique to decrypt substitution ciphers. The method is empirically effective, but in general the algorithm's success is dependent on the length of the text sample, and the question of how much text is needed is unresolved.

[1] There are 26 letters in the English alphabet. If we consider the alphabet with spaces, punctuation, or special characters, then the space of permutations will be larger.

[2] For a discussion of attacks on substitution ciphers and other basic ciphers, see [17]

1.2 A Brief Summary of Main Results

Our analysis of decrypting substitution ciphers using m-gram frequencies can be broken into the following three questions:

1. Is it possible to decode the text given arbitrarily long text and infinite computing time?
2. How much text is needed for reliable decoding?
3. For which m is the algorithm most effective?

Intuitively, the answer to question 1 should be yes: as the length of text grows, one expects the distribution of m-gram frequencies to approach the ”true” distribution[3]. Proposition 1.1 gives a rigorous ”yes” to question 1 by proving that asymptotically, distinct symbols can be distinguished by the algorithm and that distinguishing different pairs of symbols occurs independently.

Questions 2 and 3 are more difficult. In order to gain some insight on these questions, we introduce the concept of a random alphabet. This construction provides methods for generating alphabets with random letter frequencies, and for producing random strings from these alphabets. Section 3 analyzes random alphabets for $m = 1$; Section 4 does the same for $m = 2$. With this construction, our main results give evidence that the algorithm is much more effective for $m = 2$ than $m = 1$, which is supported by the experiments in Section 1.6. The experiments perform MCMC on varying lengths of text sampled from the novel War & Peace by Leo Tolstoy, and record performance using metrics such as the number of decoded symbols, and whether or not the algorithm visited the plaintext.

Before proceeding to the random model, we define the notion of a scoring function and Section 2 proves that the highest scoring permutation for English is the transposition (xj). Intuitively, this says that of all the permutations that can be applied to an English text, the result of applying the transposition (xj) is a string that can most easily be deciphered using frequency analysis.

Section 3 contains a series of results that show that asymptotically, the highest scoring non-identity permutation is a transposition τ^* and that the score of τ^* is $1/N^5$ close to that of the identity. Intuitively, this means that for $m = 1$, the high-scoring non-identity permutations are very hard to distinguish from the identity, thus it is harder for the algorithm to find the identity (global max).

Section 4 explores the random model for $m = 2$. Theorem 4.1 proves that for $m = 2$, transpositions are on the order of $1/N$ far from the identity, which implies that decoding is much more reliable in this case. This provides a partial answer to question 3: that the algorithm for $m = 2$ is more effective than for $m = 1$.

Our last main result is Theorem 4.3, which address question 2 and proves that if the length of text is at least on the order of $N \ln N$, then the identity has non-vanishing Gibbs measure. This gives strong evidence that the algorithm will reliably reach the identity (or at least, a permutation that is very close to it).

1.3 Markov Chain Monte Carlo

Monte Carlo methods utilize repeated random sampling to obtain numerical results. In other words, they use randomness to calculate deterministic quantities. This often requires sampling from com-

[3] Of course there is no ”true” distribution of m-gram frequencies for a language. In practice, one takes a large corpus of text (for example, millions of web pages found by scraping the internet) and uses the distribution found therein.

plicated, high-dimensional distributions, which presents an issue for implementation. Markov Chain Monte Carlo (MCMC) provides a way to handle this issue.

Given a state space X, let π be a probability distribution on X from which we wish to sample (often referred to as the *target* distribution). In practice, efficient computation of π is not possible e.g. π is high-dimensional. MCMC proceeds by defining a sequence (X_i) such that

$$\lim_{n\to\infty} \mathbb{P}(X_n \in A) = \int_A \pi(x)\ dx \tag{1}$$

In other words, for large n, the value X_n is an approximate sample from π. The MCMC method we will use is the Metropolis-Hastings algorithm, first introduced in 1953 by Metropolis et al. [11] then generalized by Hastings in 1970 [9]. The Metropolis algorithm creates a Markov Chain (X_i) whose stationary distribution is π, and therefore which satisfies equation (1). Here is the method [10]:

1. Start with a *symmetric* transition matrix Ψ[4] i.e. $\Psi(x,y) = \Psi(y,x)\ \forall x,y \in X$
2. When at state $x \in X$, a candidate move is generated from the distribution $\Psi(x,\cdot)$
3. Suppose the proposed state is $y \in X$. Then with probability $\alpha(x,y)$ the move is accepted and the chain moves to y; equivalently, the move is "censored" with probability $1-\alpha(x,y)$, in which case the chain remains at x for another step

It is easily shown that for $\alpha = \min\{\frac{\pi(y)}{\pi(x)}, 1\}$, the distribution π is stationary for the chain. Observe that in order to apply this method, we need only be able to compute ratios $\pi(y)/\pi(x)$. This extremely useful feature of the algorithm allows us to bypass computation of π, which may involve an unwieldy normalization constant. Moreover, if there exists an easily computable function f which is proportional to π, then we may use f in place of π in the algorithm. For example, f could be an un-normalized version of π. We will herein refer to such an f as a *likelihood* function or a *scoring* function.

1.4 Decrypting Substitution Ciphers with MCMC

Denote the plaintext by M and the ciphertext by C. Note that if M and C contain n characters, then $M, C \in \aleph^n$. For a given C, the decryption method introduced by Diaconis [5] uses the Metropolis-Hastings algorithm to find a permutation $\sigma \in S_N$ for which it is likely that $M = \sigma(C)$. Here, the state space is $X = S_N$ and our target distribution π gives the probability $\pi(\sigma)$ that σ^{-1} was used to encrypt M. Note that π depends on M and n, but we will omit these from the notation since we will not work with π directly.

Our scoring function will be denoted $H_m(\sigma)$ and is given by multiplying the empirical frequencies of all m-grams appearing in $\sigma(C)$.

The algorithm works by performing a modified random walk on S_{26} (for English; or S_N for an alphabet with N symbols). Denote by S_{26}^2 (S_N^2, respectively) the set of *transpositions*: permutations for which only two elements are swapped. Note that $|S_N^2| = \binom{N}{2}$. The random walk proceeds by proposing a uniformly chosen transposition from S_{26}^2 to apply to the current state. This can be

[4] Section 1.4 explains our choice of Ψ

summarized with the transition matrix Ψ given by

$$\Psi(\sigma_1,\sigma_2) = \begin{cases} \dfrac{1}{\binom{26}{2}} & \exists \tau \in S_{26}^2 \; s.t. \; \sigma_1 = \tau\sigma_2 \\ 0 & otherwise \end{cases}$$

The steps of the random walk may only differ by a transposition to ensure that the walk does not roam the state space too wildly. As explained above, if the proposed transposition produces a permutation $\tau\sigma$ whose score is higher, we accept the transposition and the chain proceeds to $\tau\sigma$. If not, we flip a biased coin to decide whether to accept the move; the bias is given by the ratio $H(\tau\sigma)/H(\sigma)$. This ensures that the algorithm doesn't get stuck at local maxima of H. Therefore the Markov Chain used by the algorithm has transition matrix P defined by:

$$P(\sigma_1,\sigma_2) = \begin{cases} \dfrac{1}{\binom{N}{2}} \cdot \min\left\{\dfrac{H(\sigma_2)}{H(\sigma_1)}, 1\right\}, & \exists \tau \in S_{26}^2 \; s.t. \; \sigma_2 = \tau\sigma_1 \\ 1 - \displaystyle\sum_{\sigma \neq \sigma_1} \Psi(\sigma_1,\sigma) \cdot \min\left\{\dfrac{H(\sigma)}{H(\sigma_1)}, 1\right\}, & \sigma_1 = \sigma_2 \\ 0, & otherwise \end{cases}$$

We now verify that this Markov Chain satisfies the necessary conditions of the Metropolis-Hastings algorithm. First, note that Ψ is clearly symmetric because

$$\forall \sigma_1, \sigma_2 \in S_N, \forall \tau \in S_N^2, \sigma_1 = \tau\sigma_2 \Leftrightarrow \sigma_2 = \tau\sigma_1$$

Therefore if $\exists \tau \in S_N^2$ with $\sigma_1 = \tau\sigma_2$, we have

$$\Psi(\sigma_1,\sigma_2) = \Psi(\sigma_2,\sigma_1) = \frac{1}{\binom{N}{2}}$$

and if σ_1, σ_2 do not differ by a transposition, we have

$$\Psi(\sigma_1,\sigma_2) = \Psi(\sigma_2,\sigma_1) = 0$$

Next, in order to ensure that (1) holds, we need to verify that the Markov chain is *irreducible* ($\forall x, y \in X, \exists t \; s.t. \; P^t(x,y) > 0$) and *aperiodic* (every state $x \in X$ satisfies gcd $\{t > 0 : \mathbb{P}(X_t = x : X_0 = x) > 0\} = 1$)[5]. Irreducibility follows immediately from the fact that any permutation can be written as a product of transpositions[6]. Thus $\forall \sigma_1, \sigma_2 \in S_N$, there exists $\tau_1, \ldots, \tau_k$ such that

$$\sigma_2 = (\sigma_2\sigma_1^{-1})\sigma_1 = (\tau_1 \cdots \tau_k)\sigma_1$$

Therefore it is possible to transition to σ_2 from σ_1 in k steps. Finally, the chain is clearly aperiodic because

$$P(\sigma,\sigma) > 0 \; \forall \sigma \in S_N$$

[5] The greatest common divisor (gcd) of a set of integers is the largest positive integer that divides each of the integers.

[6] This is a well known fact about permutation groups. See, for example, Dummit & Foote [6]

1.5 Scoring Functions

For an alphabet $\aleph = \{\alpha_1, \dots, \alpha_N\}$, let P_m denote the probability measure on $\aleph^m$ induced by empirical m-gram frequencies. For example, $P_1(\alpha_1)$ gives the empirical frequency of the symbol α_1 and $P_2(\alpha_1\alpha_2)$ gives the empirical frequency of the digram $\alpha_1\alpha_2$. Note that the measures P_m depend on the alphabet $\aleph$ and, by definition, must be manually computed. One popular method to compute the distributions P_m is by using a web crawler to scour billions of pages on the internet, and count the frequencies of each symbol. More generally, one can perform a similar analysis on any large (and hopefully, representative of the underlying language) corpus of text. For this paper, we take P_m from Wikipedia; see Section 2 for the exact distribution for the English language.

Given an n-symbol string $s = s_1 \cdots s_n$, define the m-gram scoring function

$$H_m(s) = \sum_{i=1}^{n-m+1} \ln P_m(s_i, \dots, s_{i+(m-1)})$$

for $n \in \mathbb{Z}^+$. The 1-gram scoring function H_1 is based purely on symbol frequencies; Diaconis used the digram scoring function H_2[7].

Our first result shows that this is the "correct" scoring function in the following sense: a random string generated one symbol via repeated sampling from P_1 will tend to have a higher score than a string generated by any other process.

Proposition 1.1. *Let $m, N \in \mathbb{Z}^+$ and let $\aleph = \{\alpha_1, \dots, \alpha_N\}$ be an alphabet with N symbols. Suppose $s = s_1 \cdots s_n$ is generated by sampling from P_1 with replacement n times. Then for any $\sigma \in S_N$,*

$$\lim_{n\to\infty} \mathbb{P}(H_m(s) > H_m(\sigma(s))) = 1$$

where $\sigma(s)$ is the result of applying σ to s symbol-wise i.e. $\sigma(s) = \sigma(s_1)\cdots\sigma(s_n)$.

Proof. We first prove the result for H_1. Write

$$H_1(s) = \sum_{i=1}^{N} N_i \ln P_1(\alpha_i)$$

where N_i is the count of the symbol α_i in the string s. The Law of Large Numbers implies that as $n \to \infty$

$$N_i/n \xrightarrow{d} P_1(\alpha_i) \tag{2}$$

Therefore

$$\lim_{n\to\infty} \mathbb{P}(H_1(s) > H_1(\sigma(s))) = \lim_{n\to\infty} \mathbb{P}\left(\sum_{i=1}^{N} N_i \ln P_1(\alpha_i) > \sum_{i=1}^{N} N_i \ln P_1(\sigma(\alpha_i))\right) = 1$$

[7] The original scoring function used by Coram & Beineke is a product of frequencies; we've computed this function on the logarithmic scale for easier calculation and to avoid rounding errors.

where the last equality uses (2) and the fact that by Gibbs Inequality[8]

$$\sum_{i=1}^{N} P_1(\alpha_i)\ln P_1(\alpha_i) > \sum_{i=1}^{N} P_1(\alpha_i)\ln P_1(\sigma(\alpha_i)).$$

For $m > 1$, let $I_N = \{1, \dots, N\}$ and let $I_N^m = \underbrace{I_N \times \cdots \times I_N}_{m \text{ times}}$. Write

$$H_m(s) = \sum_{\vec{i}=(i_1,\dots,i_m)\in I_N^m} N_{\vec{i}} \ln P_m(\vec{i})$$

where $N_{\vec{i}}$ is the count of the m-gram $\alpha_{i_1}\cdots\alpha_{i_m}$ in the string s. Note that many of the N^m terms in the sum will have $N_{\vec{i}} = 0$ since s has length n and so contains only $n-(m-1)$ m-grams.

The Law of Large Numbers implies that as $n \to \infty$

$$\frac{N_{\vec{i}}}{n-m+1} \xrightarrow{d} P_m(\vec{i}) \tag{3}$$

Therefore the same argument as above yields

$$\begin{aligned}\lim_{n\to\infty} \mathbb{P}(H_m(s) > H_m(\sigma(s))) &= \lim_{n\to\infty} \mathbb{P}\left(\sum_{\vec{i}\in I_N^m} N_{\vec{i}}\ln P_m(\vec{i}) > \sum_{\vec{i}\in I_N^m} N_{\vec{i}}\ln P_m(\sigma(\vec{i}))\right) \\ &= \mathbb{P}\left(\sum_{\vec{i}\in I_N^m} P_m(\vec{i})\ln P_m(\vec{i}) > \sum_{\vec{i}\in I_N^m} P_m(\vec{i})\ln P_m(\sigma(\vec{i}))\right) = 1. \quad \blacksquare\end{aligned}$$

Although the states of the MCMC random walk are permutations, the functions H_m assign a score to a string. However, each state $\sigma \in S_N$ corresponds to a string - namely $\sigma(s)$, where s is the plaintext.

In our analysis, we want to consider the score of a permutation without an underlying string. The intuition for doing this is that some permutations of the alphabet typically lead to an easier decryption, and we would like to think of these permuatations as having a high score. For example, one would guess that the permutation (xj) which only transposes two symbols "x" and "j" would lead to extremely decipherable text, because these symbols are very rare in English. In other words, applying (xj) to an English string will likely produce a string that one can easily decipher. In fact, we will prove in Section 2 that the transposition (xj) is indeed the "easiest to decipher".

To make this notion more precise, define the (string-agnostic) scoring function

$$\tilde{H}_m(s) = \sum_{\vec{i}\in I_N^m} P_m(\vec{i})\ln P_m(\vec{i}) \tag{4}$$

[8] The Gibbs Inequality[7] says that if $P = \{p_1, \dots, p_n\}$ is a probability distribution, then for any other probability distribution $Q = \{q_1, \dots, q_n\}$, the following holds:

$$-\sum_{i=1}^{n} p_i \log p_i \le -\sum_{i=1}^{n} p_i \log q_i$$

The Law of Large Numbers motivates this definition as in the proof of Proposition (1.1). One may notice that $\tilde{H}_m$ is in fact equal to the negative entropy of the distribution P_m. In sections 3 and 4, we use $\tilde{H}_m$ to compare the theoretical effectiveness of MCMC decryption for $m = 1$ vs. $m = 2$.

1.6 Empirical Analysis

For $m = 1$, the maximal value of H_1 is achieved by the permutation which orders the ciphertext symbols with the ordering as in English. Therefore, Monte Carlo is not needed; to decrypt, one needs only count the frequencies of each symbol occurring in the ciphertext and compare this to the frequencies of English letters. The following string gives the empirical ordering of the letters of the English alphabet along with the space symbol [18]:

$$\text{`` etaonihsrdlumcwfgypbvkxjzq''} \tag{5}$$

Thus, the space symbol occurs more frequently than any letter, and the most common letters are (in order) "e", "t", "a", etc. We will refer to such strings as *frequency strings*. So, for example, if the frequency string of a given ciphertext is

"gtkeazuqhspvicm dflnywrobx"

then H_1 is maximized by $\sigma^* \in S_{27}$ defined by

$$g \mapsto \text{[space symbol]}$$

$$t \mapsto e$$

$$\dots$$

$$x \mapsto q$$

Note that we included the space symbol in the alphabet so our state space is now S_{27}. Because the space symbol occurs most frequently, it is the easiest to decipher. Therefore its inclusion in the alphabet does not generally affect the success of the algorithm; we include it as a matter of preference and to be consistent with Diaconis [5].

Now, because we can always find a maximal permutation (under the measure induced by H_1) by counting frequencies, the success of the method is completely determined by the frequency string of the plaintext. In particular, if a given plaintext M has the frequency string given in (5), then $(\sigma^*)^{-1}(C) = M$ for any ciphertext C which was obtained using a substitution cipher on M. In general, the number of coincidences between the two frequency strings gives the number of correctly decoded symbols.

In practice, if all except, say, 4 or 5 symbols are correctly decoded, a human can likely look at the resulting text and decipher it. But if there are many incorrectly decoded symbols, there is little to no hope of recovering the plaintext M. Thus, the success of decryption when $m = 1$ is completely tied to the frequency string of the plaintext.

Python code[9] was written to obtain data about frequency strings. For various values of L, we took random samples of text of length L[10] from the (famously long) novel War & Peace by Leo Tolstoy[16]

[9] See Section 5.2

[10] For context, the typical length of an English word is 4.7 [12], and a printed novel typically contains 300 words per page. Therefore, one page corresponds to roughly 1,400 characters.

and computed their frequency strings. We then counted the number of matches with (5) and divided by 27 to get a proportion κ, which we may view as the *correctness* of the string. For each L, this was performed 1000 times to obtain average values κ_{avg}. Multiplying this proportion by 27 gives the number of symbols that will be correctly decoded.

Table 1: Correctness of Frequency Strings for Random Strings from War & Peace

L	κ_{avg}	$27 \cdot \kappa_{avg}$
100	0.16	4
1,000	0.31	8
10,000	0.47	13
100,000	0.66	18
1,000,000[11]	0.82	22

As expected, the correctness grows with L. However, even for large L, we are only able to correctly decode 22 of 27 symbols. For smaller L, decryption is near hopeless. It is clear that this method is unreliable.

One weakness of frequency strings is that they are very linked to the author and text source. For example, the protagonist of War & Peace is named Rostov, so the text contains a large number of the symbol "v" relative to most English text. This observation leads to a potential method of establishing authorship of a text source, a topic which we will not explore further in this text. The interested reader may consult, for instance, Chen et. al [2]. We remark, however, that the full text of War & Peace does have frequency string equal to that in (5).

The performance above is consistent with the theoretical analysis in Section 3, which will indicate that L need be on the order of 10,000,000 to consistently produce values of κ near 1.0.

We next examine the case $m = 2$. Python code was written to implement the Metropolis algorithm outlined in the previous section. We took portions of text from War & Peace [16] of varying lengths L, sanitized them by removing all non-alphabetic characters (except the space symbol), applied a random substitution cipher to the plaintext, and ran the algorithm. We also varied the number of steps the algorithm could take. The table below is based off the maximally scoring permutation σ^* visited by the algorithm during the allotted steps. Note: this was used instead of the value of the state where it ended since the chain could easily visit a maximizing state but then leave. In practice, one could have the program remember the top 5 or 10 permutations, then have a human look them over.

The following table summarizes the results of the experiments. The columns are defined as follows:

- L: length of random string to sample
- Step Limit: the number of iterations of MCMC to run in each experiment
- Unique Letters: the average number of unique English letters appearing in the random string, over the 1,000 experiments
- Decoded Letters: the average number of letters correctly decoded by σ^*, over the 1,000 experiments

[11] The number of characters in War and Peace, after sanitized by removing all non-alphabetic characters except the space symbol, is 3,068,166. Therefore samples of this size account for nearly 1/3 of the entire novel.

- Decoded Symbols: the average number of symbols (out of L) correctly decoded by σ^*, over the 1,000 experiments. Note that we would expect (and do see) this to be higher than "Decoded Letters" because the algorithm will likely correctly decode symbols that appear more frequently
- Plaintext Visited: the number of experiments (out of 1,000) during which the plaintext was visited
- Plaintext Best: the number of experiments (out of 1,000) during which the identity map was the highest scoring state

Table 2: Performance of MCMC Algorithm using Digram Scoring Function

L	Step Limit	Unique Letters	Decoded Letters	Decoded Symbols	Plaintext Visited	Plaintext Best
100	2,000	20.6	5.7	44.0	1	0
100	10,000	20.7	6.7	49.0	0	0
500	2,000	24.2	16.7	398	21	20
500	10,000	24.2	18.1	418.0	52	43
500	30,000	24.2	18.5	426	47	41
1,000	2,000	25.1	21.0	897	111	106
1,000	10,000	25.1	21.5	914	135	121
1,000	30,000	25.1	21.6	915	138	122
5,000	2,000	26.8	25.8	4869	694	687
5,000	10,000	26.8	26.3	4950	834	820
5,000	30,000	26.8	26.1	4923	807	782
10,000	2,000	27	26.5	9880	851	850
10,000	10,000	27	26.8	9928	959	959

Clearly, the MCMC algorithm greatly outperforms the naive $m = 1$ method. Furthermore, significantly less text is needed for high rates of success. Both Connor [3] and Chen & Rosenthal [1] found that with a ciphertext length of 2,000 characters, the algorithm had a greater than 99% success rate after approximately 10,000 iterations. We obtained similar results with our code, and will show that these results are consistent with the theory for random alphabets.

1.6.1 The Dictionary Method

Because the MCMC process is tied to the underlying language of the text, it is reasonable to wonder if we can use knowledge about the language to improve the algorithm further. We modified the above algorithm to impose penalties (rewards) for states that contained a lower (higher) number of correct 2 and 3 letter words - the "Dictionary Method".

To perform this, we compute a multiplier at each step of the random walk. This multiplier is computed as

$$MULTIPLIER = 1 + (\beta) * [n_{incorrect_words}(curr) - n_{incorrect_words}(proposed)]$$

where β is a factor that determines the severity of the reward.[12]

[12] Determining an appropriate value of β was a manual process; we experimented with $\beta = B{*}0.01$ for $B = 1, \ldots, 10$.

We then multiply the acceptance probability α by the multiplier. Because the random walk will likely start in a state where the corresponding text is complete gibberish, we allow the algorithm to run for a while before using the method. For Step Limit of $2,000$, we started imposing the method after $1,000$ steps; for Step Limits of $10,000$ and $30,000$ we started after $5,000$ steps.

This method greatly improved the algorithm, as shown in the following table.

Table 3: Performance of MCMC Algorithm using Digram Scoring Function with Dictionary Method

L	Step Limit	Unique Letters	Decoded Letters	Decoded Symbols	Plaintext Visited	Plaintext Best
100	2,000	20.6	6.1	48.1	1	0
100	10,000	20.7	7.2	52.3	2	0
500	2,000	24.2	20.7	502	134	22
500	10,000	24.2	23.1	523	171	40
500	30,000	24.2	18.5	426	175	42
1,000	2,000	25.1	22.8	897	205	108
1,000	10,000	25.1	22.9	956	421	115
1,000	30,000	25.1	23.1	971	460	112
5,000	2,000	26.8	25.8	4869	694	659
5,000	10,000	26.8	26.5	8950	904	810
5,000	30,000	26.8	26.6	9023	912	775
10,000	2,000	27	26.7	9970	972	867
10,000	10,000	27	26.9	9998	1000	940

2 The best scoring permutation for English

The main results of the thesis pertain to random alphabets where asymptotic expansions in N (the number of letters) could be employed. The analysis of English (or any other specific language) is more difficult. In the present section we present one rigorous result for English. Namely we show that the highest scoring non-identity permutation is the transposition (xj). The proof relies on both analytic estimates, some of which will be used extensively elsewhere in our work (see, in particular, inequality (7)) as well as a brute force analysis of a small number of cases. In the case of random alphabets the brute force analysis is replaced by probabilistic estimates (large deviations). It is possible that some other results proven in this work for random alphabets could be extended to English, however, a number of cases which needs to be considered by hand is much larger, so optimized computer code may be required.

To obtain results for English, we used symbol frequencies provided by Wikipedia [18]. We list these frequencies below for reference:

For $\sigma \in S_N$ we seek to bound the difference $\tilde{H}(id) - \tilde{H}(\sigma)$. Proposition 1.1 states that

$$\tilde{H}(id) \geq \tilde{H}(\sigma) \quad \forall \sigma \in S_N$$

For low values of β, the severity was too low and the performance of the algorithm was very similar to that of Table 2. For higher values of β, too much influence was given to the dictionary words, which caused performance to degrade. Ultimately we settled on the value $\beta = 0.05$, which is what was used to generate the results in Table 3.

Table 4: Relative Frequency of English Letters

a	.08167	j	.00153	s	.06327
b	.01492	k	.00772	t	.09056
c	.02782	l	.04025	u	.02758
d	.04253	m	.02406	v	.00978
e	.12702	n	.06749	w	.02360
f	.02228	o	.07507	x	.00150
g	.02015	p	.01929	y	.01974
h	.06094	q	.00095	z	.00074
i	.06966	r	.05987		

so that the difference is positive. Note that a high value of the difference indicates that σ has a low score, since its score is far from that of the identity, which is maximal.

Let N_i denote the count of symbol i. By the Law of Large Numbers,

$$\frac{N_i}{n} \to P_1(\alpha_i)$$

Writing $p_i = P_1(\alpha_i)$, we may express the difference as

$$\tilde{H}_1(id) - \tilde{H}_1(\sigma) = -\sum_{i=1}^{N} p_i \ln \frac{p_{\sigma(i)}}{p_i} = -\sum_{i=1}^{N} p_i \ln \left(1 - \frac{p_i - p_{\sigma(i)}}{p_i}\right) \tag{6}$$

We break the sum into three components, based on the value of

$$a_i := \frac{p_i - p_{\sigma(i)}}{p_i} = 1 - \frac{p_{\sigma(i)}}{p_i}$$

Define the following sets:

$$I_1 = \{i : a_i \in (-1, 0)\} = \left\{i : 1 < \frac{p_{\sigma(i)}}{p_i} < 2\right\}$$

$$I_2 = \{i : a_i \in (0, 1)\} = \left\{i : 0 < \frac{p_{\sigma(i)}}{p_i} < 1\right\}$$

$$I_3 = \{i : a_i < -1\} = \left\{i : \frac{p_{\sigma(i)}}{p_i} > 2\right\}$$

We will refer to the terms in I_j as being of Type j. Observe that the Type 1 and Type 2 terms are those for which the Maclaurin series of $\ln(1 - x)$ converges as

$$-\sum_{i\in I_1} p_i \ln \left(1 - \frac{p_i - p_{\sigma(i)}}{p_i}\right) = -\sum_{i\in I_2} p_i \ln \left(1 - \frac{p_i - p_{\sigma(i)}}{p_i}\right) = \sum_{i} p_i \sum_{n=1}^{\infty} \frac{a_i^n}{n}$$

For the Type 1 sum, we use the following Lemma:

Lemma 2.1. *For $x \in (0,1)$,*

$$\frac{x - \ln(1+x)}{x^2} > 1 - \ln 2$$

Proof. Denoting the left hand side by $f(x)$, we have $f'(x) < 0$ for all x, so that f is decreasing everywhere. This means for $x < 1$, $f(x) > f(1) = 1 - \ln 2$. ■

For the Type 1 terms, $-a_i \in (0,1)$ so we obtain the bound

$$\begin{aligned}
-\sum_{i \in I_1} p_i \ln\left(1 - \frac{p_i - p_{\sigma(i)}}{p_i}\right) &= -\sum_{i \in I_1} p_i \ln\left(1 + \frac{p_{\sigma(i)} - p_i}{p_i}\right) \\
&> \sum_{i \in I_1} p_i \left[(1 - \ln 2)\left(\frac{p_{\sigma(i)} - p_i}{p_i}\right)^2 - \frac{p_{\sigma(i)} - p_i}{p_i}\right] \\
&= \sum_{i \in I_1} (1 - \ln 2)\frac{(p_i - p_{\sigma(i)})^2}{p_i} + (p_i - p_{\sigma(i)})
\end{aligned}$$

For the Type 2 sum, all terms in the Maclaurin series are positive, so we have the lower bound

$$\begin{aligned}
-\sum_{i \in I_2} p_i \ln\left(1 - \frac{p_i - p_{\sigma(i)}}{p_i}\right) &> \sum_{i \in I_2} p_i \left[a_i + \frac{a_i^2}{2}\right] \\
&= \sum_{i \in I_2} (p_i - p_{\sigma_i}) + \frac{(p_i - p_{\sigma(i)})^2}{2p_i} \\
&> \sum_{i \in I_2} (p_i - p_{\sigma_i}) + (1 - \ln 2)\frac{(p_i - p_{\sigma(i)})^2}{p_i}
\end{aligned}$$

We rewrite the Type 3 sum as

$$-\sum_{i \in I_3} p_i \ln\left(1 + \frac{p_{\sigma(i)} - p_i}{p_i}\right)$$

For $x > 1$, observe that the function $f(x) = x - \ln(1+x)$ satisfies

$$f'(x) = 1 - \frac{1}{1+x} > 0$$

so that for $x > 1$ we have

$$f(x) > f(1) = 1 - \ln 2$$

Applying this fact with

$$x = \frac{p_{\sigma(i)} - p_i}{p_i} > 1$$

yields

$$\begin{aligned}-\sum_{i\in I_3} p_i \ln\left(1+\frac{p_{\sigma(i)}-p_i}{p_i}\right) &> \sum_{i\in I_3} p_i\left(2-\ln 2+\frac{p_i-p_{\sigma(i)}}{p_i}\right)\\ &= \sum_{i\in I_3}(2-\ln 2)p_i - p_{\sigma(i)}\\ &= \sum_{i\in I_3}(1-\ln 2)p_i\end{aligned}$$

Combining the above results, we obtain

$$\tilde{H}_1(id)-\tilde{H}_1(\sigma) > (1-\ln 2)\left[\sum_{I_1\cup I_2}\frac{(p_i-p_{\sigma(i)})^2}{p_i}+\sum_{I_3}p_i\right] \tag{7}$$

With this bound, we can prove the following Theorem.

Theorem 2.2. *For the English language,*

$$\tilde{H}_1(xj) = \max_{\sigma\in S_{26}\setminus\{id\}} \tilde{H}_1(\sigma)$$

In other words, the highest scoring permutation for $\tilde{H}_1$ is the transposition $\sigma = (xj)$.

Proof. We start by computing $\tilde{H}_1(\tau)$ for each transposition $\tau \in S_{26}$, and immediately find that the (approximate) value of $5.9\cdot 10^{-7}$ corresponding to (xj) is minimal. Now, suppose that $\tilde{H}_1(\sigma) < \tilde{H}_1(xj)$. From the above bounds, we have

$$\tilde{H}_1(\sigma) > (1-\ln 2)(|I_1\cup I_2|\delta_{\min}+|I_3|p_{\min}) \tag{8}$$

where

$$\delta_{\min} = \min_{i,j=1,\dots,26}\frac{(p_i-p_j)^2}{p_i}$$

and

$$p_{\min} = \min_i p_i$$

A quick computation produces

$$(1-\ln 2)\delta_{\min} > 1.8\cdot 10^{-7}$$

and

$$(1-\ln 2)p_{\min} = (1-\ln 2)p_z \approx 2\cdot 10^{-4}$$

Therefore, σ cannot contain any Type 3 terms. If σ contains L terms, we may then rewrite (8) as

$$\tilde{H}_1(\sigma) > 1.8\cdot 10^{-7}\cdot L$$

Because $\tilde{H}_1(\sigma) = 5.9\cdot 10^{-7}$, this immediately implies that $L < 4$. Our first computation was to verify that $\tilde{H}_1(xj)$ is minimal for σ with $L = 2$, so it remains to check for $L = 3$ i.e. for 3-cycles.

This could be done easily enough using brute force, or one can use the following simple argument. In order to have $\tilde{H}_1(\sigma) < \tilde{H}_1(xj) = 5.94 \cdot 10^{-7}$, at least one δ term in $\tilde{H}_1(\sigma)$ must satisfy

$$(1 - \ln 2)\delta < \frac{\tilde{H}_1(\sigma)}{3} < 2 \cdot 10^{-7}$$

Since δ_{jx} is the only δ satisfying this bound, we must have $\sigma = (x\ j\ \beta)$ or $\sigma = (j\ x\ \beta)$ for some letter β. Now, the other two terms are either $\{\delta_{j\beta}, \delta_{\beta x}\}$ or $\{\delta_{x\beta}, \delta_{\beta j}\}$ and their terms must sum to less than $\tilde{H}_1(xj) - \delta_{jx} \approx 3.96 \cdot 10^{-7}$, which means one must be less than $2 \cdot 10^{-7}$. Since δ_{jx} is the only such δ and it is already used, this is impossible. ■

3 The Random Model

A higher "correctness" in the ordering of symbol frequencies that occurs in the ciphertext should lead to a higher probability of decryption. In an extreme case, if the original message was written in such a way that the symbol with the highest "natural" frequency occurs the most, followed by the second, etc., then a simple frequency count will decrypt the message. In general it is intuitively clear that more correct orderings will lead to a higher probability of decryption.

We examine this phenomenon by sampling from random alphabets and analyzing how large a sample is needed until we would expect the symbols in the sample to occur in their natural ordering with high probability. If the natural frequencies of symbols are very close, we would expect that a longer sample string is needed to differentiate the symbols in the correct order.

The main result of this section is Corollary 3.15 which states that the probability that a substitution cipher applied to a string of length n sampled from a random alphabet with N letters will be correctly decoded using the single frequency method has a non trivial limit if n scales as N^5. In other words, a huge amount of text is needed for correct decoding.

In order to prove our main result we obtain some auxiliary results of independent interest. To randomly generate the N frequencies, let $\xi_1, \ldots, \xi_N$ be i.i.d., each uniform on $[0, 1]$. We refer to the event $\{|\xi_i - \xi_j| < \epsilon\}$ as an $\epsilon-$**encounter**. When ϵ is clear from context we simply refer to an "encounter". The random variable E will denote the number of encounters. Note that E depends on N and ϵ, but we omit these from the notation since N, ϵ will usually be clear from context.

For fixed values of N, the distribution of E is quite complicated, but asymptotics are attainable. Namely, in this section we shall see that for an appropriate choice of ϵ, E converges in distribution to a Poisson random variable.

3.1 The Encounter Process

A *point process* on a locally compact topological space S is a random variable whose value is a locally finite point configuration in S. Let $\mathcal{F}$ be a σ-field on S. Given a σ-finite measure ν on $\mathcal{F}$, a *Poisson Process* N is characterized by the following two properties (here $N(A)$ is the random variable denoting the number of points lying in the set A)[13]:

1. For each $F \in \mathcal{F}$ with $\nu(F) < \infty$, $\mathbb{P}\{N(F) = k\} = \frac{\nu(F)^k}{k!} e^{-\nu(F)}$

2. $N(F_1), \ldots, N(F_m)$ are independent for each $m \in \mathcal{N}$ and pairwise disjoint sets $F_1, \ldots, F_m \in \mathcal{F}$ with $\nu(F_j) < \infty, j = 1, \ldots, m$

In the case S is a subset of $\mathbb{R}^d$ for some d and ν has a density ρ with respect to the Lebesgue measure we call ρ the **intensity** of the Poisson process.

Let $\xi_1, \ldots, \xi_N$ be i.i.d. with $\xi_1 \sim U[0,1]$. Define a point process D_N whose points are the $\binom{N}{2}$ distances $|\xi_i - \xi_j|$ for $1 \leq i < j \leq N$. In other words, D_N is a point process whose points are the distances between the ξ_i. Note that the distances also take values in $[0,1]$.

3.1.1 The Probability of No Encounters

It will be useful to obtain the probability of no encounters occurring. In this case, each of the N uniformly distributed i.i.d.'s is at least ϵ away from all others. To compute this probability, assume $\xi_1 \leq \ldots \leq \xi_N$. Multiplying by $N!$ to account for the possible orderings of the ξ_i gives

$$P(E=0) = N! \int\limits_{0}^{1-(N-1)\epsilon} \int\limits_{x_1+\epsilon}^{1-(N-2)\epsilon} \cdots \int\limits_{x_{N-2}+\epsilon}^{1-\epsilon} \int\limits_{x_{N-1}+\epsilon}^{1} dx_N dx_{N-1} \ldots dx_2 dx_1 \tag{9}$$

The innermost $N-1$ integrals are all of the same form. We will use induction to write a closed-form solution for these integrals.

Lemma 3.1. *For $k \in \{0, 1, \ldots, N\}$,*

$$\int\limits_{x_{N-k-1}+\epsilon}^{1-k\epsilon} \frac{(1-k\epsilon - x_{N-k})^k}{k!} \, dx_{N-k} = \frac{(1-(k+1)\epsilon - x_{N-k-1})^{k+1}}{(k+1)!}$$

Proof. For $k=0$, the integral clearly evaluates to $1-\epsilon-x_{N-1}$. For $k>0$, use the substitution $y = 1-k\epsilon - x_{N-k}$ to obtain

$$\int\limits_{x_{N-k-1}+\epsilon}^{1-k\epsilon} \frac{(1-k\epsilon - x_{N-k})^k}{k!} \, dx_{N-k} = \int\limits_{1-(k+1)\epsilon - x_{N-k-1}}^{0} \frac{-y^k}{k!} \, dy = \frac{(1-(k+1)\epsilon - x_{N-k-1})^{k+1}}{(k+1)!}$$

■

Lemma 3.2. *The probability of no encounters is*

$$P(E=0) = (1-(N-1)\epsilon)^N$$

Proof. Using (9) and applying Lemma 3.1 $N-1$ times we get

$$P(E=0) = N! \int\limits_{0}^{1-(N-1)\epsilon} \frac{(1-(N-1)\epsilon - x_1)^{N-1}}{(N-1)!} \, dx_1$$

With the substitution $y = 1-(N-1)\epsilon - x_1$ we obtain

$$P(E=0) = N! \int\limits_0^{1-(N-1)\epsilon} \frac{(1-(N-1)\epsilon - x_1)^{N-1}}{(N-1)!}\, dx_1 = N! \int\limits_{1-(N-1)\epsilon}^{0} \frac{-y^{N-1}}{(N-1)!}\, dy$$

$$= N! \cdot \frac{-1}{(N-1)!} \cdot \frac{y^N}{N}\bigg|_{1-(N-1)\epsilon}^{0} = (1-(N-1)\epsilon)^N \qquad \blacksquare$$

3.1.2 Expected Minimum Distance

Since

$$P(E=0) = P(\min_{i\neq j} |\xi_i - \xi_j| > \epsilon),$$

we can use Lemma 3.2 to compute the expected value of the random variable

$$\Delta = \min_{i\neq j} |\xi_i - \xi_j|$$

Δ is a continuous, non-negative random variable whose maximum value is $1/(N-1)$ (corresponding to the case where all ξ_i are evenly spaced). Therefore, for all N,

$$E[\Delta] = \int\limits_0^{1/(N-1)} P(\Delta > \epsilon)\, d\epsilon = \int\limits_0^{1/(N-1)} (1-(N-1)\epsilon)^N\, d\epsilon = \frac{-(1-(N-1)\epsilon)^{N+1}}{(N+1)(N-1)}\bigg|_0^{1/(N-1)} = \frac{1}{N^2-1}$$

Thus, we expect the minimum distance $\min_{i\neq j} |\xi_i - \xi_j|$ to be roughly $1/N^2$. It is therefore reasonable to take

$$\epsilon = C/N^2 \tag{10}$$

for some $C \in \mathbb{R}^+$. In this case, Lemma 3.2 gives

$$\lim_{N\to\infty} P(E=0) = \lim_{N\to\infty} \left(1 - \frac{C(N-1)}{N^2}\right)^N = e^{-C} \tag{11}$$

Going forward, we may write ϵ_N to emphasize that ϵ depends on N.

3.1.3 Limiting Distribution of Number of Encounters

In this section we suppose that ϵ is given by (10). We now compute

$$\lim_{N\to\infty} P(E=k)$$

for $k > 0$. Let us first fix some notation.

- Let E_k denote the event $\{E = k\}$ i.e. there are precisely k encounters
- Let T denote the event that there is at least one variable involved in multiple encounters i.e. there exists ξ_j with $|\xi_j - \xi_{i_1}| < \epsilon, |\xi_j - \xi_{i_2}| < \epsilon$, $j \neq i_1 \neq i_2 \neq j$
- Let $A_k = E_k \cap T^c$

We have

$$A_k \subset E_k = (E_k \cap T^c) \cup (E_k \cap T) = A_k \cup (E_k \cap T)$$

and since the union is disjoint,

$$P(A_k) \leq P(E_k) = P(A_k) + P(E_k \cap T) \leq P(A_k) + P(T) \tag{12}$$

We will compute $\lim_{N\to\infty} P(E_k)$ by showing that $P(T) \to 0$, then computing $\lim_{N\to\infty} P(A_k)$ and applying the Squeeze Theorem.

Lemma 3.3. $P(T) \to 0$

Proof. Define $T_{ijk} = \{|\xi_i - \xi_j| < \epsilon, |\xi_j - \xi_k| < \epsilon\}$. That is, T_{ijk} is the event that ξ_j is involved in two encounters (with ξ_i and ξ_k). Observe that $T \subseteq \bigcup_{i,j,k} T_{ijk}$. The event T_{ijk} can be realized as fixing ξ_j then generating ξ_i and ξ_k, each lying inside $(\xi_j - \epsilon, \xi_j + \epsilon)$. Thus $\mathbb{P}(T_{ijk}) \leq (2\epsilon)^2$. Since there are $\binom{N}{3} < N^3$ such events T_{ijk}, we have

$$\mathbb{P}(T) \leq \mathbb{P}(\bigcup_{i,j,k} T_{ijk}) \leq \sum_{i,j,k} \mathbb{P}(T_{ijk}) < N^3(2\epsilon)^2 = \frac{4C^2N^3}{N^4} \to 0$$

■

We will now prove that A_k has a Poisson limiting distribution i.e.

$$\lim_{N\to\infty} P(A_k) = \frac{C^k e^{-C}}{k!}$$

By reindexing the ξ_i, we may realize the event A_k in the following manner: first sample $\xi_1, \dots, \xi_{N-2k}$ so that there are no encounters among any of the $(N - 2k)$ points. Then place each of the k pairs involved in encounters; since each pair must only be involved in one encounter, both of its points must be $\epsilon-$far from all previous points. It will be useful to establish some terminology.

In the following, d denotes the usual Euclidean distance.

Definition 3.4. Call a pair (ξ', ξ'') *good* with respect to a finite set Σ if

1. $d(\xi', \Sigma) \geq \epsilon$
2. $d(\xi'', \Sigma) \geq \epsilon$
3. $|\xi' - \xi''| < \epsilon$

Thus a pair is good with respect to Σ iff adding it to Σ creates exactly one new encounter - specifically, between its two elements.

Denote by Σ_k the configuration of the first k points in our set.

Let G_j denote the event in which $(\xi_{N-2k+2j-1}, \xi_{N-2k+2j})$ is a good pair with respect to $\Sigma_{2N-2k+2j}$ and let F_{N-2k} denote the event in which there are no encounters among any of the $\xi_1, \dots, \xi_{N-2k}$. Then by the discussion above Definition 3.4, we have

$$P(A_k) = \mathcal{C} \cdot P\left(F_{N-2k} \cap \left(\bigcap_{j=1}^{k} G_j\right)\right) = \mathcal{C} \cdot P\left(\bigcap_{j=1}^{k} G_j \;\middle|\; F_{N-2k}\right) \cdot P(F_{N-2k}) \tag{13}$$

where $\mathcal{C}$ is a combinatorial factor which we will determine later. From Definition 3.4, G_j can be written as the intersection of three events; denote the third of these by B_j. Then

$$P(G_j) \leq P(B_j) \leq 2\epsilon$$

The events $\{B_j\}_{j=1}^k$ are pairwise independent, and each is independent of F_{N-2k}. Therefore

$$P\left(\bigcap_{j=1}^{k} G_j \;\middle|\; F_{N-2k}\right) = P\left(\bigcap_{j=1}^{k} B_j \;\middle|\; F_{N-2k}\right) = P\left(\bigcap_{j=1}^{k} B_j\right) = \prod_{j=1}^{k} P(B_j) \leq (2\epsilon)^k \tag{14}$$

To get a lower bound, define

$$C_j = \left\{ d\left(\xi_{N-2k+2j-1}, \left[\Sigma_{N-2k+2j} \bigcup \{0,1\}\right]\right) \geq 2\epsilon \right\}$$

Note that if B_j and C_j happen then the first two conditions of Definition 3.4 are satisfied for $(\xi_{N-2k+2j-1}, \xi_{N-2k+2j})$ and $\Sigma_{N-2k+2j}$. Therefore

$$P(G_j|\Sigma_{N-2k+2j}) \geq P(C_j \cap B_j|\Sigma_{N-2k+2j}) = P(B_j|Cj\Sigma_{N-2k+2j}) \cdot P(C_j|\Sigma_{N-2k+2j}) = 2\epsilon \cdot P(C_j\Sigma_{N-2k+2j})$$

The last equality holds because $\xi_i \sim U[0, L]$ and $\xi_{N-2k+2j-1}$ is $\epsilon-$far from the boundary $\{0, 1\}$. To compute $P(C_j)$, denote by S^δ the $\delta-$neighborhood of a set S i.e.

$$S^\delta = \bigcup_{s \in S} (s - \delta, s + \delta)$$

Since C_j is the event in which $\xi_{N-2k+2j-1}$ misses all prior points (consisting of the $N - 2k$ isolated points and the previous $j - 1$ pairs) by at least 2ϵ, and is at least ϵ from the boundary,

$$P(C_j|\Sigma_{N-2k+2j}) = 1 - \mu\left(\left[\Sigma_{N-2k+2j} \bigcup \{0,1\}\right]^\epsilon\right) \tag{15}$$

Note that for any finite set Σ

$$\mu(\Sigma^\epsilon) \leq 2\epsilon \text{Cardinality}(\Sigma) \tag{16}$$

By (15) and (16),

$$1 - 2\epsilon(N + 2) \leq P(C_j|\Sigma_{2N-2j+2j}) \leq 1 \tag{17}$$

and since $\epsilon = CL/N^2$, the Squeeze Theorem implies

$$\lim_{N\to\infty} P(C_j|\Sigma_{2N-2k+2j}) = 1$$

uniformly over configurations $\Sigma_{N-2k+2j}$. We thus have our desired lower bound for the probability of good pairs

$$P\left(\bigcap_{j=1}^{k} G_j|F_{N-2k}\right) = \prod_{j=1}^{k} P(G_j|F_{N-2k}) \geq \prod_{j=1}^{k}[2\epsilon P(C_j|F_{N-2k}, C_1, \cdots C_{j-1})]$$

Since

$$P(C_j|F_{N-2k}, C_1, \cdots C_{j-1}) = \frac{E(P(C_j|F_{N-2k}, C_1, \cdots C_{j-1})1_{F_{N-2k}\cap C_1 \ldots C_{j-1}})}{P(F_{N-2k}\cap C_1 \cap \cdots C_{j-1})}$$

(17) implies that

$$P(C_j|F_{N-2k}, C_1, \ldots C_{j-1}) \geq (1 - 2\epsilon(N+2))$$

and so

$$P\left(\bigcap_{j=1}^{k} G_j|F_{N-2k}\right) \geq (2\epsilon(1-2\epsilon(N+2)))^k \tag{18}$$

The last thing to consider is the combinatorial factor $\mathcal{C}$. We do not distinguish between the orders of encounters and which nodes are involved in each; therefore the combinatorial factor is equal to the number of ways of choosing the k pairs:

$$\mathcal{C} = \frac{1}{k!}\binom{N}{2, 2, \ldots, 2, N-2k} = \frac{1}{k!}\cdot\frac{N!}{2^k(N-2k)!} \tag{19}$$

Now we can prove the main result.

Theorem 3.5.

$$\lim_{N\to\infty} P(E_k) = \frac{C^k e^{-C}}{k!}$$

Thus the limiting distribution of the number of encounters E is Poisson with parameter $C = \epsilon N^2/L$.

Proof. Lemma 3.3 and (12) immediately imply

$$\lim_{N\to\infty} P(E_k) = \lim_{N\to\infty} P(A_k)$$

By (13),

$$\lim_{N\to\infty} P(A_k) = \lim_{N\to\infty} \mathcal{C}\cdot P\left(\bigcap_{j=1}^{k} G_j \,\middle|\, F_{N-2k}\right)\cdot P(F_{N-2k}).$$

From (19) and the fact that $\lim_{N\to\infty} \epsilon_N (N-2k)^2 = \lim_{N\to\infty} \epsilon_N N^2 = C$,

$$\lim_{n\to\infty} \frac{\mathcal{C}}{N^{2k}} = \frac{1}{k! \cdot 2^k} \tag{20}$$

Next, Lemma 3.2 gives

$$\lim_{N\to\infty} P(F_{N-2k}) = \lim_{N\to\infty} e^{-C_N} = e^{C}, \tag{21}$$

where $C_N = (N-2k)^2 \epsilon_N$. Combining (14) and (18) gives

$$(1-2\epsilon(N+2))^k (2\epsilon)^k \leq P\left(\bigcap_{j=1}^{k} G_j \;\middle|\; F_{N-2k} \right) \leq (2\epsilon)^k \tag{22}$$

Finally, combining (20), (21) and (22) we obtain

$$\lim_{N\to\infty} P(A_k) = \lim_{N\to\infty} \frac{(2\epsilon)^k N^{2k}}{2^k \cdot k!} \cdot e^{-C} = \frac{C^k e^{-C}}{k!}$$

■

3.1.4 Limiting Distribution of The Encounter Process

As before $\epsilon > 0$ will be fixed. We also fix a large number M. For an interval $I = (a, b)$, define an ***I*-encounter** (or an encounter "in" I) to be the event $\{a < |\xi_i - \xi_j| < b\}$. Let

$$I_1 = (a_1, b_1), \ldots I_m = (a_m, b_m)$$

be disjoint intervals contained in $[0, M\epsilon]$. Set $I_0 = [0, M] \setminus \bigcup_{j=1}^{m} I_j$. Given positive numbers $k_1, \ldots, k_m$ we now compute the limiting probability that there are exactly k_j encounters in $\epsilon I_j = (\epsilon a_j, \epsilon b_j)$ for $j = 1, \ldots, m$ and no other encounters in $[0, \epsilon M]$. Let $\bar{k} = \sum_{i=1}^{m} k_i$.

Using the same logic as in the previous section, by reindexing the ξ_i, we may realize this by first sampling $\xi_1, \ldots, \xi_{N-2\bar{k}}$ so that $\{\xi_i\}_{i=1}^{N-2\bar{k}}$ is an $\epsilon M-$isolated set. Then for $j = 1, \ldots, m$ place each of the k_j pairs involved in I_j-encounters in such a way that no other encounters are created. To account for the varying encounter distances, we will need to expand on our previous notation.

Definition 3.6. Call (ξ', ξ'') a **good pair** with respect to a finite set Σ and an interval I if

1. $d(\xi', \Sigma) \geq \epsilon M$
2. $d(\xi'', \Sigma) \geq \epsilon M$
3. $\dfrac{|\xi' - \xi''|}{\epsilon} \in I$

Thus a pair is good iff adding it to Σ creates exactly one ϵI-encounter and does not create any other $\epsilon M-$encounters.

Let F_{N-2k} denote the event in which there are no ϵM-encounters among any of the $\xi_1, \ldots, \xi_{N-2\bar{k}}$ and let G_j denote the event that $(\xi_{N-2\bar{k}+2j-1}, \xi_{n-2\bar{k}+2j})$ is a good pair with respect to $\Sigma_{N-2\bar{k}+2j}$ and $I_{i(j)}$ where $i(j)$ is defined by the condition

$$\sum_{s=1}^{i-1} k_s < j \leq \sum_{s=1}^{i} k_s$$

Then we have

$$P(A_k) = \mathcal{C} \cdot P\left(\bigcap_{i=1}^{\bar{k}} G_j | F_{N-2\bar{k}}\right) \cdot P(F_{N-2\bar{k}}) \tag{23}$$

where $\mathcal{C}$ is a combinatorial factor which we will determine later. From Definition 3.6, G_j can be written as the intersection of three events; denote the third of these by B_j. Then

$$P(G_j) \leq P(B_j) \leq 2\epsilon(b_{i(j)} - a_{i(j)})$$

The events $\{B_j\}$ are pairwise independent, and each is independent of $F_{n-2\bar{k}}$. Therefore

$$P\left(\bigcap_{j=1}^{\bar{k}} G_j | F_{n-2\bar{k}}\right) \leq \prod_{j=1}^{\bar{k}} P(B_j | F_{n-2\bar{k}}) = \prod_{j=1}^{\bar{k}} P(B_j) \leq \prod_{i=1}^{m} [2\epsilon(b_i - a_i)]^{k_i} \tag{24}$$

To get a lower bound, define

$$C_j = \left\{ d\left(\xi_{N-2\bar{k}+2j-1}, \left[\Sigma_{N-2\bar{k}+2j} \bigcup \{0,1\}\right]\right) \geq 2\epsilon M \right\}$$

Then

$$P(G_j) \geq P(C_j \cap B_j | F_{N-2\bar{k}}) = P(B_j | C_j, F_{N-2\bar{k}}) P(C_j | F_{N-2\bar{k}}) = [2\epsilon(b_{i(j)} - a_{i(j)})] \cdot P(C_j | F_{N-2\bar{k}})$$

The last equality holds because $\xi_s \sim U[0,1]$ and on C_j, $\xi_{N-2\bar{k}+2j}$ is ϵb_i-far from the boundary $\{0,1\}$. Therefore

$$P(C_j | \Sigma_{N-2\bar{k}+2j}) = 1 - \mu\left(\left[\Sigma_{N-2\bar{k}+2j} \bigcup \{0,1\}\right]^{2\epsilon M}\right) \geq 1 - 2\epsilon N M \tag{25}$$

Since $\epsilon = C/N^2$, the Squeeze Theorem implies

$$\lim_{N\to\infty} P(C_j | \Sigma_{N-2\bar{k}+2j}) = 1 \tag{26}$$

Next

$$P\left(\bigcap_{j=1}^{\bar{k}} C_j | F_{N-2\bar{k}}\right) = \prod_{j=1}^{k_i} P(C_j | F_{N-2\bar{k}} C_1 \cdots C_{j-1})$$

Since

$$P(C_j|F_{N-2\bar{k}}C_1\cdots C_{j-1}) = \frac{E(P(C_j|F_{N-2\bar{k}}, C_1\cdots C_{j-1})1_{F_{N-2\bar{k}}\cap C_1\cdots C_{j-1}})}{P(F_{N-2\bar{k}}\cap C_1\cdots C_{j-1})}$$

(17) implies that

$$P(C_j|F_{N-2\bar{k}}C_1\cdots C_{j-1}) \geq (1-2\epsilon M(N+2))$$

and so

$$P(C_1\cdots C_{\bar{k}}|F_{N-2\bar{k}}) \geq (1-2\epsilon M(N+2))^{\bar{k}}$$

Next

$$P(G_1\cdots G_{\bar{k}}|F_{N-2\bar{k}}) = P(C_1\cdots C_{\bar{k}}|F_{N-2\bar{k}})\cdot P(B_1\cdots B_{\bar{k}}|F_{N-2\bar{k}}C_1\cdots C_{\bar{k}})$$

Since

$$P(B_j|C_1\cdots C_{\bar{k}}B_1\cdots B_{j-1}F_{N-2\bar{k}}) = P(B_j|C_j) = 2\epsilon(b_{i(j)}-a_{i(j)})$$

we obtain

$$P(G_1\cdots G_{\bar{k}}|F_{N-2\bar{k}}) \geq (1-2M(N+2)\epsilon)^{\bar{k}}\prod_{i=1}^{m}(2\epsilon(b_i-a_i))^{k_j} \tag{27}$$

The last thing to consider is the combinatorial factor $\mathcal{C}$. We must first choose the $2k_j$ points to be involved in the I_j encounters ($j=1,\cdots,m$). This can be done in

$$\begin{pmatrix} N \\ 2k_1, 2k_2, \ldots, 2k_m \end{pmatrix} = \frac{N!}{(N-2\bar{k})!(2k_1)!\cdots(2k_m)!}$$

ways. Next, we must choose the pairs of points to be involved in the encounters. For each j, this gives us a factor of

$$\frac{1}{k_j!}\begin{pmatrix} 2k_j \\ 2,2,\ldots,2 \end{pmatrix} = \frac{(2k_j)!}{2^{k_j}(k_j)!}$$

ways. Therefore

$$\mathcal{C} = \frac{N!}{(N-2\bar{k})!(2k_1)!\cdots(2k_m)!}\cdot\prod_{j=1}^{m}\frac{(2k_j)!}{2^{k_j}(k_j)!} = \frac{N!}{(N-2\bar{k})!\cdot 2^{\bar{k}}\cdot(k_1)!\cdots(k_m)!} \tag{28}$$

We are now ready to prove the main result about the Encounter Process. Set $k_0 = 0$.

Theorem 3.7. *Let $E(I_j)$ denote the number of encounters in I_j. Then*

$$\lim_{N\to\infty} P(E(I_j) = k_j\ \forall j = 0,\ldots,m) = \prod_{j=0}^{m}\frac{(C|I_j|)^{k_j}}{(k_j)!}e^{-C|I_j|}$$

In other words, if $\epsilon_N = C/N^2$ then the Encounter Process converges weakly to a Poisson Process with parameter

$$C = \epsilon_N N^2$$

Proof. By (23),

$$\lim_{N\to\infty} P(A_k) = \lim_{n\to\infty} \mathcal{C} \cdot P\left(\bigcap_{j=1}^{\bar{k}} G_j | F_{N-2\bar{k}}\right) \cdot P(F_{N-2\bar{k}})$$

It follows from (28) and Lemma 3.2 that

$$\lim_{N\to\infty} \frac{\mathcal{C}}{N^{2\bar{k}}} = \frac{1}{((k_1)! \cdots (k_m)!) \cdot 2^{\bar{k}}} \tag{29}$$

and

$$\lim_{N\to\infty} P(F_{N-2k}) = \lim_{N\to\infty} e^{-C_N M} = e^{-CM}, \tag{30}$$

where $C_N = \epsilon_N (N - 2\bar{k})^2$. Combining (24) and (27) gives

$$(1 - 2\epsilon M(N+2))^{\bar{k}} \prod_{i=1}^{m} [2\epsilon(b_i - a_i)]^{k_i} \leq P\left(\bigcap_{j=1}^{\bar{k}} G_j | F_{N-2\bar{k}}\right) \leq \prod_{i=1}^{m} [2\epsilon(b_i - a_i)]^{k_i}$$

Hence

$$\lim_{N\to\infty} \frac{P\left(\bigcap_{j=1}^{\bar{k}} G_j | F_{N-2\bar{k}}\right)}{\epsilon_N^{\bar{k}}} = \prod_{i=1}^{m} [2(b_i - a_i)]^{k_i} \tag{31}$$

Finally, combining (29), (30) and (31) gives

$$\begin{aligned}\lim_{N\to\infty} P(E(I_j) = k_j \ \forall j = 1, \ldots, m) &= \lim_{N\to\infty} (N^{2\bar{k}} \epsilon_N^{\bar{k}}) \frac{1}{2^{\bar{k}} \cdot (k_1)! \cdots (k_m)!} \cdot \prod_{i=1}^{m} [2\epsilon(b_i - a_i)]^{k_i} \epsilon^{CM} \\ &= \frac{C^{\bar{k}}}{(k_1)! \cdots (k_m)!} \cdot \prod_{i=1}^{m} (b_i - a_i)^{k_i} \cdot e^{-CM}\end{aligned}$$

Recalling that $k_0 = 0$ we obtain

$$\begin{aligned}\lim_{N\to\infty} P(E(I_j) = k_j \ \forall j = 1, \ldots, m) &= \frac{C^{\bar{k}}}{(k_1)! \cdots (k_m)!} \cdot \prod_{i=1}^{m} (b_i - a_i)^{k_i} \cdot e^{-CM} \\ &= e^{-C|I_0|} \prod_{j=1}^{m} \frac{(C(b_j - a_j))^{k_j}}{(k_j)!} e^{-C(b_j - a_j)} \\ &= \prod_{j=0}^{m} \frac{(C|I_j|)^{k_j}}{(k_j)!} e^{-C|I_j|}\end{aligned}$$

■

3.1.5 Extended encounter process

Here we put $\epsilon = 1/N^2$, thus $C = 1$. Fix a large L and let $\{I_i\}$ a partition of $[0, L]$, J_j be a partition of $[0, 1]$.

Theorem 3.8. *Let $E(I_i, J_j)$ denote the number of encounters in I_i such that the corresponding frequencies are in J_j. Then*

$$\lim_{N\to\infty} P(E(I_i, J_j) = k_{ij}\ \forall i, j) = \prod_{i,j} \frac{(C|I_j|J_j|)^{k_{ij}}}{(k_{ij})!} e^{-C|I_i||J_j|}$$

Proof. Let N_j be the number of frequencies in J_j. Let $\mathcal{A}$ be the event of interest, that is the event that $E(I_i, J_j) = k_{ij}$ for each j, and $\mathcal{B}$ be the event that

$$N_j \in \left[|J_j|N - N^{2/3}, |J_j|N + N^{2/3}\right] \quad \forall j \in 1, \dots m.$$

Due to Chernoff bounds, $\mathbb{P}(\mathcal{B}) \to 1$ as $N \to \infty$. Thus it remains to show that

$$\mathbb{P}(\mathcal{A}|\mathcal{B}) \to \prod_{ij} \frac{(C|I_i|J_j|)^{k_{ij}}}{(k_{ij})!} e^{-C|I_j||J_j|} \text{ as } N \to \infty.$$

To this end it suffices to show that for each $n_1, \dots n_m$ satisfying $n_j \in \left[|J_j|N - N^{2/3}, |J_j|N + N^{2/3}\right]$ we have

$$\mathbb{P}(E(I_i, J_j) = k_{ij}\ \ \forall i, j|N_j = n_j) \to \prod_{i,j} \frac{(C|I_j|J_j|)^{k_{ij}}}{(k_{ij})!} e^{-C|I_i||J_j|} \quad \text{as} \quad N \to \infty. \tag{32}$$

Note that conditioned on $N_1 = n_1, \dots N_m = n_m$ the numbers of encounters in $J_1, \dots, J_m$ are independent. Therefore (32) follows from Theorem 3.7 applied to each interval I_j separately. ■

3.2 Symbol Frequencies in Random Alphabets

With the results of §3.1, we can turn to our primary objective of analyzing frequencies of symbols drawn from random alphabets. We will describe the procedure for generating the frequencies, then apply our results about the Encounter Process to draw conclusions about the likelihood of symbol orderings.

Consider an alphabet of N symbols which we can think of as the integers $1, \dots, N$. Let $\xi_1, \dots, \xi_N$ be i.i.d $\sim U[0, 1]$ and set the frequency of the i-th symbol to be

$$f_i = \xi_i / \sum_{i=1}^{N} \xi_i$$

Since $\mathbb{E}[\sum_{i=1}^{N} \xi_i] = N/2$ we expect f_i to be roughly $2\xi_i/N$.

We assume that the consecutive letters are chosen independently so that the j-th letter is chosen with frequency f_j.

We consider sampling finite strings from this alphabet; we denote a string of n symbols by $s_1 \cdots s_n \in \{1, \ldots, N\}^n$. Let $F_i(n)$ denote the count of symbol i in a string of length n. We are interested in the differences $F_i(n) - F_j(n)$ for $i \neq j$. Denoting this variable by X_{ij}^n, we have

$$X_{ij}^n = F_i(n) - F_j(n) = \sum_{r=1}^{n} Y_{ij}^r$$

where

$$Y_{ij}^s = \begin{cases} 1 & s_r = i \\ -1 & s_r = j \\ 0 & \text{else} \end{cases}$$

Fixing i and j, the Y_{ij}^s are i.i.d. with $\mu_{ij} = \mathbb{E}[Y_{ij}] = f_i - f_j$ and $\mathbb{E}[Y_{ij}^2] = f_i + f_j$ so that

$$\sigma_{ij}^2 = (f_i + f_j) - (f_i - f_j)^2 < \infty$$

We can therefore apply the Central Limit Theorem to conclude that

$$X_{ij}^n \approx \sum_{s=1}^{n} Y_{ij}(s) \xrightarrow{d} \mathcal{N}(n\mu, n\sigma^2),$$

that is

$$\frac{X_{ij}^n - n(f_i - f_j)}{\sqrt{n}\sigma_{ij}} \Rightarrow \mathcal{N}(0,1) \tag{33}$$

provided that n is large (compared to N). In the next section we describe the precise scaling at which the probability of the correct decoding changes.

3.3 The Probability of Incorrect Orders

If $f_i > f_j$ then we would expect to see more of character i than character j if we sample a sufficiently long random string. Define a "mistake" or incorrect order to be an event of the form $\{f_i > f_j\} \cap \{F_j > F_i\}$. The following theorem gives a bound on the probability of an incorrect ordering.

Theorem 3.9. *Suppose that $f_i - f_j > L/N^3$. Then*

$$\mathbb{P}(F_j(n) > F_i(n)) \leq e^{-3L^2 n/\pi^2 N^5}$$

Proof. Since

$$\mathbb{P}(F_j(n) > F_i(n)) = \mathbb{P}(X_{ji}^n > 0), \tag{34}$$

we can apply Markov's inequality to conclude that for all $s > 0$,

$$\mathbb{P}(X_{ji}^n > 0) = \mathbb{P}(e^{sX_{ji}^n} > 1) \leq \mathbb{E}[e^{sX_{ji}^n}] \tag{35}$$

Since $X_{ji}^n = \sum_{m=1}^n Y_{ji}^m$, where the Y_{ji}^m are i.i.d, we have

$$\mathbb{E}[e^{sX_{ji}^n}] = \mathbb{E}[e^{s\sum_{m=1}^n Y_{ji}^m}] = \mathbb{E}\left[\prod_{m=1}^n e^{sY_{ji}^m}\right] = \prod_{m=1}^n \mathbb{E}[e^{sY_{ji}^m}] \tag{36}$$

Assuming $s < 1$, using the Taylor expansion of e^s we obtain

$$\begin{aligned}
\mathbb{E}[e^{sY_{ji}^m}] &= 1 - f_i - f_j + e^s f_j + e^{-s} f_i \\
&= (1 - f_i - f_j) + s(f_j - f_i) + f_j \sum_{n=2}^{\infty} s^n/n! + f_i \sum_{n=2}^{\infty} (-s)^n/n! \\
&< 1 - sL/N^3 + (f_j + f_i) \sum_{n=1}^{\infty} \frac{s^{2n}}{(2n)!} + (f_j - f_i) \sum_{n=1}^{\infty} \frac{s^{2n+1}}{(2n+1)!} \\
&< 1 - sL/N^3 + \frac{2}{N} \sum_{n=1}^{\infty} \frac{s^2}{n^2} \\
&= 1 - sL/N^3 + 2(\pi s)^2/6N
\end{aligned}$$

For $s = 3L/2\pi^2 N^2$, this gives

$$1 - sL/N^3 + 2\pi^2 s^2/6N = 1 - 3L^2/\pi^2 K N^5$$

It now follows that

$$\mathbb{E}[e^{sX_{ji}^n}] < 1 - sL/N^3 + 2K\pi^2 s^2/6N = (1 - 3L^2/\pi^2 N^5)^n < e^{-3L^2 n/\pi^2 N^5}$$

■

The above Theorem indicates that string lengths should be taken on the order of N^5 in order to make the probability of a mistake negligible. We will make this more precise using the Encounter process D_N. The idea is as follows: consider (inhomogenously) thinning the Encounter Process so that a point - which represents a symbol pair - is retained if the pair is decoded incorrectly. Then the number of incorrectly decoded symbol pairs is precisely the number of points in the thinned process. Our goal is to compute the probability that the number of points in this limiting thinned process is 0.

Then, using our knowledge of the distribution of the Encounter process, we can give results about the probability of incorrect decodings by analyzing the thinned process. Rather than considering the Encounter Process on $[0, 1]$ directly, it will be convenient to consider an equivalent process on $[0, 1] \times [0, 1]$ defined as follows: a point $|\xi_i - \xi_j|$ of the Encounter Process is represented $(\xi_j, \xi_i - \xi_j)$.

Theorem 3.10. *Fix $\tau = n/N^5$. Consider two symbols ξ_i^N, ξ_j^N corresponding to a point from the Encounter Process. More precisely, assume that $\xi_i^N > \xi_j^N$ and let $u_N = \xi_j^N$ and $\ell_N = \xi_i - \xi_j$. Assume also that $\ell_N \leq \frac{L}{N^2}$, that $\xi_j^N > N^{-0.1}$ and that*

$$\lim_{N\to\infty} \frac{1}{N} \sum_j \xi_j^N = \frac{1}{2}, \quad \lim_{N\to\infty} u_N = u, \quad \lim_{N\to\infty} \ell_N N^2 = \ell$$

Then the probability $p_N(u,\ell,\tau)$ of incorrectly decoding the order of the symbols i and j converges

$$\lim_{N\to\infty} p_N(u,\ell,\tau) = H(u,\ell,\tau)$$

where

1. *H is a decreasing function of τ*
2. *$\lim_{\tau\to\infty} H(u,\ell,\tau) = 0$*

and the convergence is uniform provided that (u,ℓ) and u^{-1} vary over a compact set.

The explicit formula for H will be given at the end of the proof. See (40).

Proof. As above, let $f_i = \dfrac{\xi_i}{\sum_{k=1}^N \xi_k}$. Define the following random parameters:

$$\delta_{ij} = \xi_i - \xi_j, \quad T_N = \sum_{k=1}^N \xi_k, \quad \Delta_{ij} = f_i - f_j = \delta_{ij}/T_N, \quad L_{ij} = N^3\Delta_{ij}, \quad U_{ij} = \xi_j$$

We want to compute $\lim_{n\to\infty} \mathbb{P}(X_{ji}^{n(N)} > 0)$. Recall that $X_{ji}^n = \sum_{r=1}^n Y_{ji}^r$ where the Y_{ji}^r are i.i.d. (so we'll drop the superscript), with mean $f_j - f_i$. Define

$$W_{ji} = Y_{ji} - (f_j - f_i) = Y_{ji} + \Delta_{ij}$$

so that W_{ji} has mean 0. Then for $S_n = \frac{1}{n}\sum_{i=1}^n W_{ji}$, the Berry-Esseen Theorem [14] states that there is a constant $C < 1$ such that for all x, n,

$$\left|\mathbb{P}\left(\frac{S_n\sqrt{n}}{\sigma} \leq x\right) - \mathbb{P}(\mathcal{Z} \leq x)\right| \leq \frac{C\rho}{\sigma^3\sqrt{n}} \tag{37}$$

Here $\mathcal{Z} \sim \mathcal{N}(0,1), \rho = \mathbb{E}[|W_{ji}|^3] = (1-2f_j)\Delta_{ij}^3 + 3(f_i+f_j+1)\Delta_{ij}^2 + (f_i+f_j)$ and

$$\begin{aligned}\sigma^2 = \mathbb{E}[W_{ji}^2] &= \mathbb{E}[(Y_{ji}+\Delta_{ij})^2] \\ &= \mathbb{E}[Y_{ji}^2] + 2\Delta_{ij}\mathbb{E}[Y_{ji}] + \Delta_{ij}^2 \\ &= f_i + f_j + 3\Delta_{ij}^2\end{aligned}$$

Keeping in mind that f_i, f_j, and σ all depend on N, it follows that for all n,

$$\left|\mathbb{P}\left(\frac{S_n\sqrt{n}}{\sigma} > \frac{\sqrt{n}\Delta_{ij}}{\sigma_N}\right) - \mathbb{P}\left(\mathcal{Z} > \frac{\sqrt{n}\Delta_{ij}}{\sigma_N}\right)\right| < \frac{\rho}{2\sigma_N^3\sqrt{n}} \tag{38}$$

Since

$$S_n = \frac{1}{n}\sum_{i=1}^n W_{ji} = \frac{1}{n}\sum_{i=1}^n (Y_{ji}+\Delta_{ij}) = \frac{X_{ji}^n + n\Delta_{ij}}{n},$$

we have

$$\mathbb{P}\left(\frac{S_n\sqrt{n}}{\sigma_N} > \frac{\sqrt{n}\Delta_{ij}}{\sigma_N}\right) = \mathbb{P}\left(\left(\frac{X_{ji}^n + n\Delta_{ij}}{n}\right)\frac{\sqrt{n}}{\sigma_N} > \frac{\sqrt{n}\Delta_{ij}}{\sigma_N}\right) = \mathbb{P}(X_{ji}^n > 0)$$

Denote the right-hand side of (38) by $\epsilon_{n,N}$. Then we may write

$$\mathbb{P}(X_{ji}^{n} > 0) = \mathbb{P}\left(\mathcal{Z} > \frac{\sqrt{n}\Delta_{ij}}{\sigma_n}\right) \pm \epsilon_{n,N} \tag{39}$$

Now,

$$\sigma_N = \sqrt{f_i + f_j + 3\Delta_{ij}^2} = \sqrt{\frac{\xi_i + \xi_j}{T_N} + 3\Delta_{ij}^2} = \sqrt{\frac{2U_{ij} + \Delta_{ij}}{T_N} + 3\Delta_{ij}^2}$$

Therefore

$$\frac{\sqrt{n}\Delta_{ij}}{\sigma_N} = \frac{\sqrt{\tau N^5}\Delta_{ij}}{\sqrt{\frac{2U_{ij}+\Delta_{ij}}{T_N} + 3\Delta_{ij}^2}} = \frac{\sqrt{\tau}L_{ij}}{\sqrt{(2U_{ij} + \Delta_{ij})/(T_N/N) + 3N\Delta_{ij}^2}} \overset{N\to\infty}{\to} \frac{\sqrt{\tau}\ell}{\sqrt{u}}$$

Convergence follows because by assumptions of the theorem

$$\Delta_{ij} \to 0, \quad U_{ij} \to u, \quad L_{ij} = \frac{N^2\delta_{ij}}{T_N/N} \to 2\ell, \quad \frac{T_N}{N} \to \frac{1}{2}$$

We next show that $\lim_{n\to\infty} \epsilon_{n(N),N} = 0$. Write $\epsilon_{n,N} = a(n,N)/b(n,N)$, where

$$a(n,N) = (1 - 2f_j)\Delta_{ij}^3 + 3(f_i + f_j + 1)\Delta_{ij}^2 + (f_i + f_j)$$

and

$$b(n,N) = 2(f_i + f_j) + 3\Delta_{ij}^2)^{3/2} \cdot \sqrt{n}$$

By assumptions of the theorem, for large N, $f_k^N = \dfrac{\xi_k}{T_N} < \dfrac{4}{N}$, whence $\lim\limits_{N\to\infty} f_k^N = 0$. Therefore

$$\lim_{n\to\infty} a(n,N) < \lim_{n\to\infty} [\Delta_{ij}^3 + 9\Delta_{ij}^2 + (f_i + f_j)] = 0$$

On the other hand, $b(n,N) \geq \dfrac{c}{N}\sqrt{\tau}N^{5/2}$ so that $\lim\limits_{N\to\infty} b(n,N) = \infty$.

Therefore $\lim\limits_{n\to\infty} \epsilon_{n,N} = \lim\limits_{n\to\infty} \dfrac{a(n,N)}{b(n,N)} = 0$. Now (39) implies the statement of the theorem with

$$H(u,\ell,\tau) = \lim_{N\to\infty} \mathbb{P}(X_{ji}^{n} > 0) = \mathbb{P}\left(\mathcal{Z} > \frac{\sqrt{\tau}\ell}{\sqrt{u}}\right) = \int_{\frac{\sqrt{\tau}\ell}{\sqrt{u}}}^{\infty} e^{-s^2/2}\, ds. \tag{40}$$

This function clearly satisfies the desired properties. ■

Next we would like to generalize this result to the case of finitely many (mutually disjoint) pairs of symbols.

Theorem 3.11. *Fix $\tau = n/N^5$. Consider k distinct pairs of symbols $(\xi_1, \xi_2), \dots, (\xi_{2k-1}, \xi_{2k})$ from the Encounter Process. For each i, assume $\xi_{2i-1} > \xi_{2i}$. As before let $u_{i,N} = \xi_{2i}^n$ be the symbol with lower frequency in each pair, and let $\ell_{i,N} = \xi_{2i-1} - \xi_{2i}$ be the difference. Suppose that*

$$\lim_{N\to\infty} \frac{1}{N} \sum_j \xi_j^N = \frac{1}{2}, \quad \lim_{N\to\infty} u_{i,N} = u_i, \quad \lim_{N\to\infty} \ell_{i,N} N^2 = \ell_i$$

Divide $\{1, \dots, k\} = \mathfrak{C} \cup \mathfrak{I}$. Then the probability $p_k(\vec{u}, \vec{\ell}, \mathfrak{C})$ of correctly decoding the orders of all symbols in $\mathfrak{C}$ and incorrectly decoding the orders of all symbols in $\mathfrak{I}$ converges

$$\lim_{N\to\infty} p_k(\vec{u}, \vec{\ell}, \mathfrak{C}) = H_k(\vec{u}, \vec{\ell}, \tau, \mathfrak{C})$$

where

$$H_k(\vec{u}, \vec{\ell}, \tau, \mathfrak{C}) = \left[\prod_{i\in\mathfrak{C}} (1 - H(u_i, \ell_i, \tau))\right] \left[\prod_{i\in\mathfrak{I}} H(u_i, \ell_i, \tau)\right]$$

The convergence is uniform when $\vec{u}, \vec{\ell}$ vary over a compact set such that $\min_i u_i > N^{-0.1}$.

In other words, the order of a pair labeled (ℓ, u) for the Extended Encounter Process is decoded correctly with probability $1 - H(u, \ell, \tau)$ and the decoding of different pairs is asymptotically independent.

Proof. For $i = 1, \dots, k$, let X_i^n denote the random variable $F_{2i}(n) - F_{2i-1}(n)$ i.e. X_i^n is the difference in observed frequencies between the less and more probable symbols in pair i for a string of length n. Thus $X_i^n > 0$ represents an incorrect decoding of symbol pair i. Define the random vector $\vec{X_k^n} = (X_1^n, \dots, X_k^n)$. Then the correct decoding of all symbols in $\mathfrak{C}$ corresponds precisely to the event that $\vec{X_k^n}$ lies inside the hyper-quadrant of $\mathbb{R}^k$ defined by

$$Q := \{X_i > 0 \text{ iff } i \in \mathfrak{C}\}$$

Let $\vec{e}_1, \dots \vec{e}_k$ be the standard basis of $\mathbb{R}^k$ i.e. $\vec{e}_m$ is the vector of all zeros except for a 1 in the m-th position. Then define

$$Y_t = \begin{cases} \vec{e}_m & s_t = i_{2m-1} \\ -\vec{e}_m & s_t = i_{2m} \\ 0 & \text{else} \end{cases}$$

Observe that $\vec{X_k^n} = \sum_{t=1}^n Y_t$. Indeed, we may think of the index t as corresponding to a letter in the random string, and Y_t denoting which symbol pair was affected by this letter, by recording a $+1$ (-1) in the $m - th$ component for the lower (higher) frequency symbol in pair m. Therefore the sum of the Y_t represents a sort of “scorecard” for keeping track of the occurrences amongst the various pairs of symbols, which is precisely $\vec{X_k^n}$.

Let μ_k denote the mean of Y_t and note that

$$\mu_k = \mathbb{E}[Y_t] = (f_2 - f_1, f_4 - f_3, \dots, f_{2k} - f_{2k-1})$$

Define $W_t = Y_t - \mathbb{E}[Y_t]$ so that W_t has mean 0, and let $W = \sum_{t=1}^n W_t, \tilde{W} = W/N^2$. Then the multidimensional Berry-Esseen Theorem [14] states that there exists a constant C such that $\forall U \subset \mathbb{R}^k$ convex,

$$\left|\mathbb{P}\left(\tilde{W} \in U\right) - \mathbb{P}(\mathcal{Z} \in U)\right| < Ck^{1/4}\gamma$$

where $\mathcal{Z} \sim \mathcal{N}(0, \Sigma)$, Σ is the covariance of $\tilde{W}$, and

$$\gamma = \frac{1}{N^6} \sum_{t=1}^{n} \mathbb{E}[||\Sigma^{-1/2} W_t||_2^3]$$

Since $X_k = N^2 \tilde{W} + n\mu_k$, denoting $Q_k = Q - \dfrac{n\mu_k}{N^2}$ we get

$$\mathbb{P}(\tilde{W} \in Q_k) = \mathbb{P}\left(N^2 \tilde{W} \in N^2 \left(Q - \frac{n\mu_k}{N^2}\right)\right) = \mathbb{P}(N^2 \tilde{W} + n\mu_k \in Q) = \mathbb{P}(\vec{X_k^n} \in Q)$$

where the second equality uses that $N^2 Q = Q$.

Since Q_k is open and convex, we can apply the Berry-Esseen Theorem to get that

$$\mathbb{P}(\vec{X_k^n} \in Q) = \mathbb{P}(\mathcal{Z} \in Q_k) \pm C k^{1/4} \gamma \tag{41}$$

Observe that W_{t1} and W_{t2} are independent for $t_1 \neq t_2$, since the symbols appearing in position t_1 and t_2 are drawn independently. Recall that W_t (and hence, each $W_t(j)$) has mean 0. Denote the mean of $Y_t(j)$ by $\mu_j = f_{2j-1} - f_{2j}$. Then

$$\mathrm{Var}(W_t(j)) = \mathbb{E}[W_t(j)^2] = \mathbb{E}[(Y_t(j) - \mu_j)^2] = \mathbb{E}[Y_t(j)^2] - 2\mu_j \mathbb{E}[Y_t(j)] + \mu_j^2 = f_{2j} + f_{2j-1} - \mu_j^2 = 2f_{2j} + \mu_j - \mu_j^2.$$

Hence

$$\mathrm{Var}(W(j)) = n\mathrm{Var}(W_1(j)) = n\left[2f_{2j} + \mu_j - \mu_j^2\right].$$

Next

$$\begin{aligned} Cov(W_t(i), W_t(j)) &= Cov(Y_t(i), Y_t(j)) = E(Y_t(i)Y_t(j)) - E(Y_t(i))E(Y_t(j)) \\ &= -E(Y_t(i))E(Y_t(j)) = -[f_{2i-1} - f_{2i}][f_{2j-1} - f_{2j}] \end{aligned}$$

Using the assumption of the theorem we obtain

$$\lim_{N\to\infty} \mathrm{Var}(W(j)/N^2) = \lim_{N\to\infty} \frac{2nf_{2j}}{N^4} = \lim_{N\to\infty} \frac{2\tau N^5 \xi_{2j}}{T_N N^4} = \lim_{N\to\infty} \frac{2\tau \xi_{2j}}{(T_N/N)} = 4u_j \tau$$

Likewise for large N

$$\mathrm{Cov}\left(\frac{W(i)}{N^2}, \frac{W(j)}{N^2}\right) \sim \frac{n\mathrm{Cov}(W(i), W(j))}{N^4} = -\frac{\tau N^5 [f_{2i-1} - f_{2i}][f_{2j-1} - f_{2j}]}{N^4} \sim \frac{\ell_i \ell_j}{(T_N/N)^2 N^3} \sim \frac{4\ell_i \ell_j}{N^3}$$

Hence $\lim\limits_{N\to\infty} \Sigma_N = 4\tau \mathrm{diag}(u_1, \ldots, u_k)$.

Since the set of invertible matrices is open, $||\Sigma_N^{-1}|| \leq C$ for large N. Therefore

$$\gamma \leq \frac{C}{N^6} \sum_{t=1}^{n} ||W_t||_2^3 \leq \frac{cn}{N^6}$$

and so $\lim\limits_{N\to\infty} \gamma_N = 0$. Next

$$\lim_{N\to\infty} \frac{n\mu_k(j)}{N^2} = -\lim_{N\to\infty} \frac{\tau N^5 \ell_{j,N} N^{-2}}{T_N N^2} = -\frac{\tau \ell_j}{\lim\limits_{N\to\infty} (T_N/N)} = -2\tau \ell_j$$

Combining the above estimates we conclude that $\lim_{N\to\infty} \mathbb{P}(\vec{X_k^n} \in Q) = \mathbb{P}(\mathcal{Z} \in \mathcal{Q})$ where $\mathcal{Z}$ is the normal vector with zero mean and covariance matrix $4\tau\mathrm{diag}(u_1, \dots, u_k)$ and $\mathcal{Q} = Q - \mu$ where μ is the vector with components $-2\tau(\ell_1, \dots, \ell_k)$.

Since the covariance matrix of $\mathcal{Z}$ is diagonal, its components are independent, whence

$$\mathbb{P}(\mathcal{Z} \in \mathcal{Q}) = \prod_{j=1}^{k} \mathbb{P}\left(Z_j \star \frac{\sqrt{\tau}\ell_j}{\sqrt{u_j}}\right)$$

where $\star$ means $\geq$ if $j \in \mathfrak{I}$ and $\star$ means $\leq$ if $j \in \mathfrak{C}$. This completes the proof. ■

We next compute the distribution of the thinned point process, which we call the Retained Process. This process is obtained by applying thinning to an Extended Encounter Process of §3.1.5, where a point (corresponding to the distance between a pair of symbol frequencies) is retained iff its corresponding pair of symbols is decoded incorrectly for the text of length $n = \tau N^5$.

Theorem 3.12. *For an Encounter Process, the associated retained process approaches as $N \to \infty$ a Poisson process on $[0, \infty] \times [0, 1]$ with measure ν which has density $H(u, \ell, \tau)$ with respect to the Lebesgue measure.*

The proof of Theorem 3.12 will rely on the following fact[13].

Proposition 3.13. *Consider a Poisson process on a space S with measure μ. Suppose that each point ξ in our process is retained with probability $p(s)$ independently of the other points. Then the set of retained points forms a Poisson process with measure ν which has density p with respect to μ.*

Proof. of Theorem 3.10. Fix a large number L and a small number δ. By Theorem 3.6 the number of points in $[0, L] \times [\delta, 1]$ asymptotically forms a Poisson process with intensity 1. By Theorem 3.9 a point (ℓ, u) is retained with probability $H(u, \ell, \tau)$ independently of the other points. Hence by Proposition 3.11 the restriction of the retained process on $[0, L] \times [\delta, 1]$ is the Poisson process with the measure described in the statement of the theorem. It remains to show that for each ϵ we can choose δ and L so that the probability to have a retained point such that either $\ell > L$ or $u < \delta$ is smaller than ϵ. Let $\mathcal{R}_L$ be the number of retained points with $\ell > L$. By Markov's inequality it suffices to show that $\mathbb{E}(\mathcal{R}_L)$ can be made as small as we wish by taking L large. Let $\mathcal{E}_{m,L}$ be the number of points in the encounter process with $\ell \in [Lm, L(m+1)]$ and $\mathcal{R}_{m,L}$ be the number of the retained points in the same interval.

Lemma 3.14. $\mathbb{E}(\mathcal{E}_{m,L}) \leq L$.

Proof. Write $\mathcal{E}_{m,L} = \sum_{i \neq j} E_{ij}$, where

$$E_{ij} = \begin{cases} 1, & N^2(\xi_i - \xi_j) \in [Lm, L(m+1)] \\ 0, & \text{else} \end{cases}$$

In other words, E_{ij} indicates whether the encounter between ξ_i and ξ_j contributes to $\mathcal{E}_{m,L}$. Then we have

[13] Proposition 3.13 embodies what is often referred to as the "thinning" property of Poisson processes. For a proof of this Proposition and related discussion see [4].

$$\mathbb{E}[\mathcal{E}_{m,L}] = \binom{N}{2} \cdot \mathbb{E}[\mathcal{E}_{1,2}] = \binom{N}{2} \cdot \mathbb{P}\left(N^2(\xi_1 - \xi_2) \in [Lm, L(m+1)]\right)$$

$\xi_1 - \xi_2$ has a triangular distribution with CDF $F(x) = 2x - x^2$. Therefore

$$\mathbb{P}(\xi_1 - \xi_2 \in [Lm/N^2, L(m+1)/N^2]) = F\left(\frac{L(m+1)}{N^2}\right) - F\left(\frac{Lm}{N^2}\right) = \frac{2L}{N^2} - \frac{2L^2m - L^2}{N^4}.$$

Thus $\mathbb{E}[\mathcal{E}_{m,L}] = \binom{N}{2} \cdot \left[\frac{2L}{N^2} - \frac{2L^2m - L^2}{N^4}\right] \leq L.$ ■

By Theorem 3.9 and Lemma 3.14

$$\mathbb{E}(\mathcal{R}_{m,L}) \leq \mathbb{E}(\mathcal{E}_{m,L})e^{-3(mL)^2\tau/\pi^2} \leq Le^{-3(mL)^2\tau/\pi^2}.$$

Thus

$$\mathbb{E}(\mathcal{R}_L) = \sum_{m=1}^{\infty} \mathbb{E}(\mathcal{R}_{m,L}) \leq L \sum_{m=1}^{\infty} e^{-3(mL)^2\tau/\pi^2} \leq L \sum_{m=1}^{\infty} e^{-3mL\tau/\pi^2} = \frac{Le^{-3L\tau/\pi^2}}{1 - e^{-3L\tau/\pi^2}}$$

Since the last expression tends to 0 as $L \to \infty$ we can take L so large that $\mathbb{E}(\mathcal{R}_L) \leq \epsilon/2$. Hence by Markov inequality $\mathbb{P}(\mathcal{R}_L \geq 1) \leq \epsilon/2$.

Next by Theorem 3.8 the probability that the number of points in the encounter process with $u < \delta$ is non-zero converges as $N \to \infty$ to $1 - e^{-\delta}$ which can be made smaller that $\epsilon/2$ by taking δ small enough.

Combing the above estimates proves the theorem. ■

Note that the total number of points in the Extended Encounter process is infinite since Lesbegue$[0,\infty] \times [0,1] = \infty$ the total number of points for the Retained process is finite. Namely denote

$$\Lambda(\tau) = \nu([0,\infty] \times [0,1]) = \int_0^1 \int_0^L H(u,\ell,\tau)\, d\ell\, du = \frac{1}{\sqrt{2\pi}} \int_0^1 \int_0^L \int_{\ell\sqrt{\tau}/\sqrt{u}}^{\infty} e^{-s^2/2}\, ds\, d\ell\, du.$$

Then the total number of points in the retained process has Poisson distribution with parameter $\Lambda(\tau)$. Since the retained process contains pairs whose order is not decoded correctly, the whole cipher is correctly decoded iff $R = 0$. We thus obtain

Corollary 3.15. *In the random frequency model the probability that the naive guess leads to the correct order for the text with $n(N) = \tau N^5$ symbols converges as $N \to \infty$ to $e^{-\Lambda(\tau)}$.*

3.4 The Best Scoring Permutation

We will prove that with high probability, the best scoring permutation in the random model is a transposition. To that end, fix τ^* to be the highest scoring transposition. In this subsection we shall show that with probability close to 1 τ^* is the permutation whose score is closest to that of the identity. That is,

$$\tilde{H}_1(id) - \tilde{H}_1(\tau^*) = \min_{\sigma \in S_N} \{\tilde{H}_1(id) - \tilde{H}_1(\sigma)\} \tag{42}$$

For convenience define $D_1(\sigma) = \tilde{H}_1(id) - \tilde{H}_1(\sigma)$. Recall from (6) that

$$D_1(\sigma) = \sum_i p_i \ln \frac{p_i}{p_{\sigma(i)}}$$

and also

$$I_1 = \left\{ i : 0 < \frac{p_{\sigma(i)}}{p_i} < 1 \right\}, \quad I_2 = \left\{ i : 1 < \frac{p_{\sigma(i)}}{p_i} < 2 \right\}, \quad I_3 = \left\{ i : \frac{p_{\sigma(i)}}{p_i} > 2 \right\}$$

Every transposition contains a Type 1 term and either a Type 2 or Type 3 term. By (7), we have for any σ

$$D_1(\sigma) > (1 - \ln 2) \left[\sum_{I_1 \cup I_2} \frac{(p_i - p_{\sigma(i)})^2}{p_i} + \sum_{I_3} p_i \right] \tag{43}$$

Our strategy is to first obtain an estimate for $D_1(\tau^*)$, and then prove that with high probability this estimate is minimal among all $D_1(\sigma)$.

Lemma 3.16. $\forall \epsilon > 0, \exists K$ *such that*

$$\mathbb{P}\left(D_1(\tau^*) \leq \frac{K}{N^5} \right) > 1 - \epsilon$$

Proof. Recall that $S = \sum_{j=1}^{N} \xi_j$. S has an Irwin-Hall distribution[14] and has mean $N/2$ and variance $N/12$. Applying Chevyshev's inequality then gives

$$\mathbb{P}\left(S < \frac{N}{10} \right) = \mathbb{P}(S - \mathbb{E}[S] < -0.4N) \leq \frac{Var(S)}{(0.4N)^2} = \frac{1}{1.92N} \tag{44}$$

Next we show that if κ is large then with high probability there exists a pair ξ_1, ξ_2 such that $\xi_1 > 0.1$ and $|\xi_1 - \xi_2| < \kappa/N^2$. Indeed if such a pair exists and $S \geq 0.1N$ then letting σ be the transposition of symbols corresponding to ξ_1 and ξ_2 and supposing that $\xi_2 > \xi_1$ we get

$$D(\sigma) = \frac{\xi_2 - \xi_1}{S} \ln\left(1 + \frac{\xi_2 - \xi_1}{\xi_1} \right) \leq \frac{(\xi_2 - \xi_1)^2}{S\xi_1} \leq \frac{100\kappa^2}{N^5}$$

where in the first inequality we used that $\ln(1 + t) < t$ for $t > 0$. Thus we get the claim with $K = 100\kappa^2$. It remains to show that the pair described above exists with probability close to 1. Let

$$E = \{\text{There are no pairs } \xi_1, \xi_2 : (\xi_1 - \xi_2) < \frac{\kappa}{N^2}; \xi_1, \xi_2 > 0.1\}$$

[14] The Irwin-Hall distribution describes the sum of uniformly distributed random variables. See [8]

Let M be the number of points from $\{\xi_1, \dots, \xi_N\}$ such that $\xi_j > 0.1$. Then

$$\begin{aligned}\mathbb{P}(E) &= \mathbb{P}(E|M \geq N/2) \cdot \mathbb{P}(M \geq N/2) + \mathbb{P}(E|M < N/2) \cdot \mathbb{P}(M < N/2) \\ &\leq \mathbb{P}(E|M \geq N/2) + \mathbb{P}(M < N/2)\end{aligned}$$

Note that M has a binomial distribution with parameters $(N, 0.9)$. Accordingly $\mathbb{E}[M] = 0.9N$, $Var(M) = 0.09N$. Hence by Chebyshev's inequality,

$$\mathbb{P}(M < N/2) = \mathbb{P}(M - \mathbb{E}[M] < 0.4N) \leq \frac{Var(M)}{0.16N^2} = \frac{9}{16N} \to 0 \text{ as } N \to \infty \tag{45}$$

On the other hand $\mathbb{P}(E|M \geq N/2) \cdot \mathbb{P}(M \geq N/2) \leq \mathbb{P}(A_N)$, where A_N is the event that among the first $N/2$ points in $[0.1, 1]$ there are no κ/N^2 encounters. Denote $L = N/2$. Rescaling the interval $[0.1, 1]$ to the unit size we see that $\mathbb{P}(A_N) = \mathbb{P}(B_L)$ where B_L is the event that among L points on $[0, 1]$ there are no $\frac{5\kappa}{18L^2}$ encounters. Applying Theorem 3.5 we conclude that

$$\lim_{N\to\infty} \mathbb{P}(A_N) = \lim_{L\to\infty} \mathbb{P}(B_L) = e^{-5\kappa/18} \tag{46}$$

Fix $\epsilon > 0$. Then we could choose κ such that $e^{-5\kappa/18} < \frac{\epsilon}{10}$, so (46) tells us that for large N, $\mathbb{P}(A_N) < \frac{\epsilon}{5}$. Combining this with (44) and (45) gives the result. ■

Now we must show that with high probability, $D_1(\sigma)$ is much larger than $1/N^5$ for all other permutations moving more than 2 symbols. We begin by showing that the highest scoring permutation cannot be a product of disjoint cycles, and therefore is itself a cycle.

Lemma 3.17. *Let σ_1, σ_2 be disjoint cycles. Then*

$$D_1(\sigma_1\sigma_2) > D_1(\sigma_1)$$

Equivalently, $\tilde{H}_1(\sigma_1) > \tilde{H}_1(\sigma_1\sigma_2)$

Proof. Let $D_{KL}(P||Q)$ denote the relative entropy from Q to P[15]. We have

$$D_1(\sigma_1) = \sum_i p_i \ln \frac{p_i}{p_{\sigma_1(i)}} = -\sum_i p_i \ln \frac{p_{\sigma_1(i)}}{p_i} = D_{KL}(P||\sigma_1(P))$$

and

$$D_1(\sigma_1\sigma_2) = \sum_i p_i \ln \frac{p_i}{p_{\sigma_1\sigma_2(i)}} = -\sum_i p_i \ln \frac{p_{\sigma_1\sigma_2(i)}}{p_i}$$

Since σ_1 and σ_2 are disjoint, the last sum can be broken up as

$$-\sum_{i:\sigma_1(i)\neq i} \ln \frac{p_{\sigma_1(i)}}{p_i} - \sum_{i:\sigma_2(i)\neq i} \ln \frac{p_{\sigma_2(i)}}{p_i} = D_{KL}(P||\sigma_1(P)) + D_{KL}(P||\sigma_2(P)) \geq D_{KL}(P||\sigma_1(P))$$

since relative entropy is non-negative. ■

[15] D_{KL} represents Kullback–Leibler divergence, commonly referred to as relative entropy, and is a measure of how one probability distribution is different from a second, reference probability distribution. See [15].

It remains to prove that for all cycles σ of length $\geq 3, D_1(\sigma) \gg N^{-5}$ (with high probability).

Lemma 3.18. *For all cycles σ of length $\geq$ 3, we have that for sufficiently large N*

$$\mathbb{P}(D_1(\sigma) > 1/N^{4.5}) > 1 - \frac{3}{(1-\ln 2)\sqrt{N}}$$

Proof. Let $\alpha = (1-\ln 2)^{-1}$ and $S = \sum_{i=1}^{N} \xi_i$. If $D_1(\sigma) < \frac{1}{N^5}$ and σ contains any Type 3 terms, then (43) immediately implies that

$$\min_{i \in I_3} p_i < \frac{\alpha}{N^5}$$

By definition of the frequencies p_i, we have

$$\mathbb{P}\left(\min_{i\in I_3} p_i < \frac{\alpha}{N^{4.5}}\right) < \mathbb{P}\left(\exists i, p_i < \frac{\alpha}{N^{4.5}}\right) = \mathbb{P}\left(\exists i, \frac{\xi_i}{S} < \frac{\alpha}{N^{4.5}}\right) = \mathbb{P}\left(\exists i, \xi_i < \frac{\alpha S}{N^{4.5}}\right)$$

$$\leq \mathbb{P}\left(\min_i \xi_i < \frac{\alpha}{N^{3.5}}\right) \leq N\mathbb{P}\left(\xi_1 < \frac{\alpha}{N^{3.5}}\right) < \frac{1}{N^{2.5}}$$

Therefore, with probability greater than $1 - N^{-2.5}$, any transposition σ with score below $N^{-4.5}$ only contains Type 1 and Type 2 terms. From (43), the contribution to $D_1(\sigma)$ for each of these terms is at least

$$(1-\ln 2)\frac{(p_i - p_{\sigma(i)})^2}{p_i}$$

If there are K such terms (i.e. σ is a K-cycle), then there must exist i with

$$\alpha\frac{(p_i - p_{\sigma(i)})^2}{p_i} < \frac{1}{N^{4.5}} \quad \text{and} \quad \alpha\frac{(p_{\sigma(i)} - p_{\sigma^2(i)})^2}{p_{\sigma(i)}} < \frac{1}{N^{4.5}}$$

For this term, we must have

$$(\xi_i - \xi_{\sigma(i)})^2 < \frac{\xi_i S}{\alpha N^{4.5}} < \frac{1}{\alpha N^{3.5}}$$

since ξ_1 and $S < N$. Taking roots, this implies $\xi_i - \xi_{\sigma(i)} < \frac{1}{\sqrt{\alpha} N^{7/4}}$. By Lemma 3.3, the probability that a single point is involved in two encounters of distance $\epsilon = \alpha{-}1/2N^{-7/4}$ is less than

$$N^3\epsilon^2 = \frac{1}{\alpha\sqrt{N}} \tag{47}$$

Therefore, denoting by T the event that σ contains a Type 3 term, we have

$$\begin{aligned}\mathbb{P}(D_1(\sigma) < 1/N^{4.5}) &= \mathbb{P}(D_1(\sigma) < 1/N^{4.5} \mid T) \cdot \mathbb{P}(T) + \mathbb{P}(D_1(\sigma) < 1/N^{4.5} \mid T^c) \cdot (1 - \mathbb{P}(T)) \\ &< \mathbb{P}(T) + \mathbb{P}(D_1(\sigma) < 1/N^{4.5} \mid T^c) \\ &< \frac{1}{N^{2.5}} + \frac{2}{\alpha\sqrt{N}} < \frac{3}{\alpha\sqrt{N}}\end{aligned}$$

■

We now can prove our main result.

Theorem 3.19. *For the random model with N symbols, let A_N be the event that the highest scoring permutation is a transposition. Then* $\lim_{N\to\infty} \mathbb{P}(A_N) = 1$.

Proof. Let τ^* denote the highest scoring permutation and let σ^* be the highest scoring transposition changing at least three symbols. Fix ϵ and choose K so large that $\mathbb{P}(D_1(\tau^*) \geq K/N^5) < \epsilon$ for all sufficiently large N . Then A_N is equivalent to the event that $D_1(\tau^*) < D_1(\sigma^*)$. Hence Lemmas 3.16 and 3.18 give

$$\mathbb{P}(A_N^c) \leq \mathbb{P}\left(D_1(\tau^*) > \frac{K}{N^5}\right) + \mathbb{P}\left(D_1(\sigma^*) < \frac{K}{N^{4.5}}\right) \leq \epsilon + \frac{3}{\alpha\sqrt{N}}$$

It follows that $\liminf_{N\to\infty} \mathbb{P}(A_N) \geq 1 - \epsilon$. Since ϵ is arbitrary the result follows.

■

4 The Random Model with Digram Scoring Function

4.1 Generating the frequencies

To determine m-gram frequencies in English, for example, one would analyze a large corpus of text (e.g. a collection of large novels, or by using a web crawler and analyzing millions of web pages) and estimate the frequencies from it. Since a random alphabet does not correspond to any real language, we will determine the symbol frequencies theoretically rather than empirically. To do so, first generate N probability vectors, each with N components. That is, for $i = 1, \ldots, N$, generate $\{p_{ij}\}_{j=1}^N$ so that $\sum_j p_{ij} = 1$. This can be done by sampling i.i.d. random variables $\{\xi_{ij}\}_{j=1}^N$, each uniform on $[0, 1]$ and setting

$$p_{ij} = \frac{\xi_{ij}}{\sum_j \xi_{ij}}$$

We view the $N \times N$ stochastic matrix $\Pi = (p_{ij})$ as the transition matrix for a Markov chain whose realization can be thought of as randomly generating a string one symbol at a time, where subsequent symbols are chosen based on digram probabilities relative to the previous symbol. In other words, from state i (which corresponds to symbol s_i), the next state (symbol) is chosen from the distribution $(p_{ij})_{j=1}^N$. It is then natural to define the individual symbol frequencies (π_i) to be the eigenvector of

Π. In other words, the symbol frequencies correspond to the stationary distribution of the Markov chain.

In this setting, the symbol s_i corresponds to the Markov chain visiting state i and the digram $s_i s_j$ corresponds to the chain consecutively visiting states i and j. Alternatively, the digram represents a transition from state i to state j.

Remark: It may seem odd that we first generate digram frequencies and then derive individual symbol frequencies from them, and not vice versa. However, this approach is natural if one considers that digram frequencies contain structure of the underlying language being used and are not based on individual letter frequencies per se. For this reason, given $\{p_i\}_{i=1}^N$, there is no obvious logical way to define $\{p_{ij}\}_{i,j=1}^N$. For instance, one seemingly natural option would be to define $p_{ij} = p_i p_j$, but in English this would correspond to asserting that "QR" has a greater frequency than "QU" simply because the frequency of "R" is greater than that of "U".

The main results of this Section are Theorem 4.1, which says that the gap in score between the identity and the best scoring non-trivial permutation is of order $1/N$, and Theorem 4.4 which indicates that for texts of length of order $N \ln N$ the Gibbs measure assigns sizable weight to a correct decoding, making it plausible that this text can be decoded using the Gibbs sampler algorithm.

4.2 The Best Scoring Permutation

In Lemma 3.16, we showed that the probability of the event

$$\min_{\tau \in S_N^2} |\tilde{H}_1(\tau) - \tilde{H}_1(id)| \leq \frac{K}{N^5}$$

can be made arbitrarily close to 1, by taking K large. Because the score of such τ is very close to the (optimal) score of the identity, this proves that in the single frequency (i.e. $1-$gram) model of Section 3, it is very hard to distinguish these transpositions as sub-optimal. This provides evidence to the fact that the single frequency model is weak.

In this section, we prove an analogous result for the digram model, showing that with high probability, transpositions τ satisfy, for some $C > 0$,

$$|\tilde{H}_2(\tau) - \tilde{H}_2(id)| \geq \frac{C}{N}$$

This provides additional evidence that the digram model outperforms the individual frequency model.

Theorem 4.1. *With high probability, there exists $C > 0$ such that for each transposition τ*

$$|\tilde{H}_2(\tau) - \tilde{H}_2(id)| \geq \frac{C}{N}.$$

The proof of Theorem 4.1 will rely on two lemmas. In what follows, let $S_i = \sum_\ell \xi_{i\ell}$.

Lemma 4.2. $\mathbb{P}\left(\exists i : \frac{S_i}{N} \notin [0.49, 0.51]\right) \leq e^{-C_1 N}$ *for some $C_1 > 0$*

Proof. Large Deviation bound (Chernoff bound) ∎

Lemma 4.3. *There is a constant $b > 0$ such that for all i,* $\mathbb{P}\left(\pi_i \leq \frac{1}{100N}\right) \leq e^{-bN}$

Proof. Being the stationary distribution of the Markov Chain, π is an eigenvector of the transition matrix (digram frequency matrix) P. In particular,

$$\pi P^2 = \pi$$

Writing this equation term-wise gives

$$\pi_i = \sum_{j,k} \pi_j p_{jk} p_{ki} = \sum_{j,k} \frac{p_{ij}\xi_{jk}\xi_{ki}}{S_j S_k}$$

Note that the random variables $\{\xi_{ik}\xi_{kj}\}_k$ are independent. Therefore, applying a Chernoff bound[16] gives

$$\begin{aligned}\mathbb{P}\left(\sum_k \xi_{ik}\xi_{kj} < \frac{N}{100}\right) &< \min_{t>0} e^{tN/100} \prod_{k=1}^{N} \mathbb{E}[e^{-t\xi_{ik}\xi_{kj}}] \\ &= \min_{t>0} e^{tN/100} \left(\mathbb{E}[e^{-t\xi_{ik}\xi_{kj}}]\right)^N \\ &= \min_{t>0} \left(e^{t/100} \cdot \mathbb{E}[e^{-t\xi_{ik}\xi_{kj}}]\right)^N\end{aligned}$$

Write $\phi(t) = e^{t/100} \cdot \mathbb{E}[e^{-t\xi_{ik}\xi_{kj}}]$. Note that $\phi(0) = 1$, and that

$$\begin{aligned}\phi'(t) &= e^{t/100} \cdot \frac{d}{dt}\mathbb{E}[e^{-t\xi_{ik}\xi_{kj}}] + \frac{e^{t/100}}{100} \cdot \mathbb{E}[e^{-t\xi_{ik}\xi_{kj}}] \\ &= e^{t/100} \cdot \mathbb{E}[\frac{d}{dt}e^{-t\xi_{ik}\xi_{kj}}] + \frac{e^{t/100}}{100} \cdot \mathbb{E}[e^{-t\xi_{ik}\xi_{kj}}] \\ &= e^{t/100} \cdot \mathbb{E}[-\xi_{ik}\xi_{kj}e^{-t\xi_{ik}\xi_{kj}}] + \frac{e^{t/100}}{100} \cdot \mathbb{E}[e^{-t\xi_{ik}\xi_{kj}}]\end{aligned}$$

so that

$$\phi'(0) = \mathbb{E}[-\xi_{ik}\xi_{kj}] + \frac{1}{100} = \frac{1}{100} - \frac{1}{4} < 0$$

Since $\phi(0) = 1$ and $\phi'(0) < 1$, there exists $t_0 > 0$ with $\phi(t_0) < 1$. Write $\phi(t_0) = e^{-\ln\phi(t_0)} = e^{-b}$, where $b > 0$. Then

$$\mathbb{P}\left(\sum_k \xi_{ik}\xi_{kj} < \frac{N}{100}\right) < e^{-bN}$$

[16] The Chernoff Bound for a random variable X is attained by applying Markov's Inequality (a.k.a Chebyshev's Inequality)[14] to e^{tX}. Explicitly, we have for all $t > 0$,

$$\mathbb{P}(X \geq a) \leq \frac{\mathbb{E}[e^{tX}]}{e^{ta}}$$

For the likely case that $\sum_k \xi_{ik}\xi_{kj} > N/100$, we have

$$\frac{\sum_k \xi_{ik}\xi_{kj}}{S_k S_i} > \frac{1}{100N}$$

since $S_i, S_k < N$. Therefore in this case

$$\pi_i = \sum_{j,k} \pi_j p_{jk} p_{ki} = \sum_{j,k} \pi_j \frac{\xi_{jk}\xi_{ki}}{S_k S_j} = \sum_i \pi_i \left(\sum_k \frac{\xi_{jk}\xi_{ki}}{S_k S_j} \right) > \sum_i \pi_i \left(\frac{1}{100N} \right) = \frac{1}{100N}$$

Putting this all together gives

$$\begin{aligned}\mathbb{P}\left(\pi_i \le \frac{1}{100N}\right) &= \mathbb{P}\left(\pi_i \le \frac{1}{100N} \middle| \sum_k \xi_{ik}\xi_{kj} < \frac{N}{100}\right) \cdot \mathbb{P}\left(\sum_k \xi_{ik}\xi_{kj} < \frac{N}{100}\right) \\ &\quad + \mathbb{P}\left(\pi_i \le \frac{1}{100N} \middle| \sum_k \xi_{ik}\xi_{kj} > \frac{N}{100}\right) \cdot \mathbb{P}\left(\sum_k \xi_{ik}\xi_{kj} > \frac{N}{100}\right) \\ &< e^{-bN}\end{aligned}$$

■

We may now prove Theorem 4.1.

Proof. of Theorem 4.1. Let N_{ij} denote the count of the digram $s_i s_j$. By the Law of Large Numbers, and the fact that π is the stationary distribution of the Markov Chain, we have

$$\frac{N_{ij}}{n} \to \mathfrak{p}_{ij} := \pi_i p_{ij}.$$

Then for any transposition $\tau = (ij)$,

$$\begin{aligned}|\tilde{H}_2(id) - \tilde{H}_2(\tau)| &= \sum_{k=1}^{N} \pi_i p_{ik} \ln \frac{\pi_i p_{ik}}{\pi_j p_{jk}} + \pi_j p_{jk} \ln \frac{\pi_j p_{jk}}{\pi_i p_{ik}} + \pi_k p_{ki} \ln \frac{\pi_k p_{ki}}{\pi_k p_{kj}} + \pi_k p_{kj} \ln \frac{\pi_k p_{kj}}{\pi_k p_{ki}} \\ &= -\sum_{k=1}^{N} w_k \ln \frac{x_k}{w_k} - \sum_{k=1}^{N} x_k \ln \frac{w_k}{x_k} - \sum_{k=1}^{N} y_k \ln \frac{z_k}{y_k} - \sum_{k=1}^{N} z_k \ln \frac{y_k}{z_k}\end{aligned}$$

Where for convenience we have defined

$$w_k = \pi_i p_{ik}, \quad x_k = \pi_j p_{jk}, \quad y_k = \pi_k p_{ki}, \quad z_k = \pi_k p_{kj}.$$

Note that each sum has the same form as the sum in Section 2.1. Thus we may apply (7) to get:

$$-\sum_{k=1}^{N} w_k \ln \frac{x_k}{w_k} > (1 - \ln 2) \left[\sum_{I_1 \cup I_2} \frac{(w_k - x_k)^2}{w_k} + \sum_{I_3} w_k \right]$$

$$-\sum_{k=1}^{N} x_k \ln \frac{w_k}{x_k} > (1-\ln 2)\left[\sum_{I_1 \cup I_2} \frac{(x_k - w_k)^2}{x_k} + \sum_{I_3} x_k\right]$$
$$-\sum_{k=1}^{N} y_k \ln \frac{z_k}{y_k} > (1-\ln 2)\left[\sum_{I_1 \cup I_2} \frac{(y_k - z_k)^2}{y_k} + \sum_{I_3} y_k\right]$$
$$-\sum_{k=1}^{N} z_k \ln \frac{y_k}{z_k} > (1-\ln 2)\left[\sum_{I_1 \cup I_2} \frac{(z_k - y_k)^2}{z_k} + \sum_{I_3} z_k\right] \tag{48}$$

Note that there is a slight abuse of notation here as the sets I_1, I_2, I_3 are not the same in each sum; however our strategy is to consider these sums independently. Observe that all 8 sums contained in brackets are positive. Therefore it suffices to prove the lower bound for (48).

We have
$$\frac{(z_k - y_k)^2}{z_k} = \frac{(\pi_k p_{kj} - \pi_k p_{ki})^2}{\pi_k p_{kj}} = \pi_k \cdot \frac{(p_{kj} - p_{ki})^2}{p_{kj}} = \frac{\pi_k}{S_k} \frac{(\xi_{kj} - \xi_{ki})^2}{\xi_{kj}}$$

By Lemma 4.2, and Lemma 4.3,
$$\frac{\pi_k}{S_k} > \frac{1}{51N^2}$$
with probability greater than $(1-e^{-C_1 N})(1-e^{-bN})$. Therefore it suffices to obtain a lower bound for
$$\sum_{I_1 \cup I_2} \frac{(\xi_{ki} - \xi_{kj})^2}{\xi_{ki}} > \sum_{I_1 \cup I_2} (\xi_{ki} - \xi_{kj})^2$$

Call the index k *good* if $k \in I_1 \cup I_2$ and $|\xi_{ki} - \xi_{kj}| > 0.1$. Then

$$\mathbb{P}(k \text{ is good}) = \mathbb{P}\left(|\xi_{ki} - \xi_{kj}| > 0.1 \middle| \frac{\xi_{ki}}{\xi_{kj}} \in (0,1)\right) \cdot \mathbb{P}\left(\frac{\xi_{ki}}{\xi_{kj}} \in (0,1)\right) \tag{49}$$
$$+ \mathbb{P}\left(|\xi_{ki} - \xi_{kj}| > 0.1 \middle| \frac{\xi_{ki}}{\xi_{kj}} \in (1,2)\right) \cdot \mathbb{P}\left(\frac{\xi_{ki}}{\xi_{kj}} \in (1,2)\right) \tag{50}$$
$$\tag{51}$$

We can compute the exact value of this probability by viewing the situation geometrically. Because $\xi_{ki}, \xi_{kj} \sim U[0,1]$, the set of all ξ_{ki}, ξ_{kj} that contribute to the probability (49) corresponds to the region bounded by the lines $y = 0, x = 1$, and $y = x - 0.1$. This region is a triangle with area $\frac{1}{2}(0.9)(0.9) = \frac{81}{200}$.

Similarly, the set of all ξ_{ki}, ξ_{kj} that contribute to the probability (50) corresponds to the region bounded by the lines $y = 2x, y = 1$, and $y = x + 0.1$. This region is also a triangle and has area $\frac{1}{2}(0.4)(0.4) = \frac{18}{100}$.

Therefore the probability that an index is good is precisely $\frac{117}{200} > 0.5$. Because indices are good independently of other indices being good, a Chernoff bound gives
$$\mathbb{P}\left(\#\{k : k \text{ is good}\} > \frac{N}{2}\right) > (1 - e^{-C_2 N})$$

for some $C_2 > 0$. Putting this all together gives, with probability greater than $(1 - e^{-C_1 N})(1 - e^{-C_2 N})(1 - e^{-bN})$,

$$-\sum_{k=1}^{N} z_k \ln \frac{y_k}{z_k} > \frac{N}{2}(1 - \ln 2)\frac{1}{51N^2}$$

■

4.3 Gibbs Measure

We now answer the final original question regarding the Gibbs measure of the permutation corresponding to the decryption key. As above, let $\xi_{ij} \sim U[0,1]$ and $S_i = \sum_{j=1}^{N} \xi_{ij}$ with $p_{ij} = \frac{\xi_{ij}}{S_i}$. Furthermore, define the random variables:

- $\gamma_{ij} = \pi_i p_{ij}$ (which we can think of as the "digram frequency" in the random model)
- $r_{ij} = \frac{\gamma_{ij}}{\gamma_{\sigma(i)\sigma(j)}}$

Let S_N denote the permutation group on N elements, and $\mathcal{S}_k \subset S_N$ be the permutations moving exactly k elements.

The Gibbs measure on S_N is defined by

$$\mathcal{P}(\sigma) = \frac{e^{\beta \tilde{H}_2(\sigma)}}{z_\beta}$$

where $\tilde{H}_2(\sigma) = \sum_{i,j} r_{ij} \ln r_{\sigma(i)\sigma(j)}$.

Note that β usually represents the inverse temperature of the system[17], but here it represents the length of the string.

Define a related energy function by

$$\mathcal{H}(\sigma) = \tilde{H}_2(\sigma) - \tilde{H}_2(id)$$

and denote its induced Gibbs measure by $\mathcal{P}_\beta$. Thus

$$\mathcal{P}_\beta = \frac{e^{\beta \mathcal{H}(\sigma)}}{z^*_\beta}$$

Note that $z^*_\beta = z_\beta * e^{-\beta \tilde{H}_2(id)}$, and that $\mathcal{P}_\beta(id) = 1/z^*_\beta$.

Theorem 4.4. *There exist C_1, C_2 such that $\forall \beta \geq C_1 N \ln N$,*

$$\lim_{N\to\infty} (\mathbb{P}(\mathcal{P}_\beta(id) \geq C_2) = 1$$

[17] For a full explanation of the Ising model, see [10]

The theorem states that if string length is taken to be on the order of $N \ln N$, then the Gibbs measure of the identity is bounded below by some constant. Namely, it is non-trivial and non-vanishing, and thus we'd expect a random walk using Metropolis-Hastings and this Gibbs measure to eventually visit the identity, or a permutation quite close to it.

Definition 4.5. We say that $\mathcal{H}$ is *c-monotone* if $\forall k, \forall \sigma \in \mathcal{S}_k, \mathcal{H}(\sigma) \leq -\frac{ck}{N}$.

Lemma 4.6. *$\forall C_1 > 1, \exists C_2 > 1, C_3 \in (0,1)$ such that if $\mathcal{H}$ is C_1-monotone then $\forall \beta \geq C_2 N \ln N$ we have $\mathcal{P}_\beta(id) \geq C_3$.*

Proof. Recall that $\mathcal{P}_\beta(id) = 1/z_\beta^*$. By hypothesis we have

$$z_\beta^* = \sum_{\sigma \in S_N} e^{\beta \mathcal{H}(\sigma)} = \sum_{k=0}^{N} \sum_{\sigma \in \mathcal{S}_k} e^{\beta \mathcal{H}(\sigma)} \leq \sum_{k=0}^{N} |\mathcal{S}_k| e^{-C_1 C_2 k \ln N}$$

Since $|\mathcal{S}_k| \leq \binom{N}{k} \cdot k! \leq N^k = e^{k \ln N}$, we have

$$z_\beta^* \leq \sum_{k=0}^{N} e^{\ln N [-C_1 C_2 + 1] k} = \sum_{k=0}^{N} N^{(1-C_1C_2)k} = \frac{1 - N^{(1-C_1C_2)(N+1)}}{1 - N^{(1-C_1C_2)}}$$

Choosing C_2 large enough, we can force the geometric sum to lie in $(1, 1+\epsilon)$ for any $\epsilon > 0$. ∎

Recall that

$$-\mathcal{H}(\sigma) \geq [1 - \ln 2] \left(\sum_{(i,j) \in I_1 \cup I_2} \frac{(\gamma_{ij} - \gamma_{\sigma(i)\sigma(j)})^2}{\gamma_{ij}} + \sum_{(i,j) \in I_3} \gamma_{ij} \right)$$

where

$$I_1 = \{r_{ij} \in [1,2]\}$$
$$I_2 = \{r_{ij} < 1\}$$
$$I_3 = \{r_{ij} > 2\}$$

In view of Lemma 4.6, Theorem 4.4 follows from

Proposition 4.7. *If C_1 is sufficiently small then*

$$\mathbb{P}_N \left(\textit{Random alphabet satisfies } \mathcal{H}(\sigma) \geq -\frac{C_1 k}{N} \; \forall \sigma \in \mathcal{S}_k \right) \to 1$$

as $N \to \infty$

Let $\sigma \in S_N$. Define the following sets

$$\Gamma_i^+ = \left\{ j : \frac{\xi_{ij}}{\xi_{\sigma(i)\sigma(j)}} \geq 3 \right\}$$

$$\Gamma_i^- = \left\{ j : \frac{\xi_{ij}}{\xi_{\sigma(i)\sigma(j)}} \leq \frac{1}{3} \right\}$$

Definition 4.8. We say that i is *C_5-good with respect to σ* if $\sigma(i) \neq i$ and

$$\sum_{j \in \Gamma_i^+} \xi_{ij} \geq C_5 N$$

$$\sum_{j \in \Gamma_i^-} \xi_{\sigma(i)\sigma(j)} \geq C_5 N$$

Definition 4.9. We say that a random alphabet is *C_5-typical* if

1. $\forall i, \pi_i > 0.01/N$
2. $\forall i, S_i \in [0.49N, 0.51N]$
3. $\forall \sigma \in \mathcal{S}_k$, at least $\left[\dfrac{k}{100}\right] + 1$ indices i are C_5-good w.r.t. σ.

Proposition 4.7 then follows from the following two Lemmas.

Lemma 4.10. *If an alphabet is C_5-typical then $\forall k, \forall \sigma \in \mathcal{S}_k, \mathcal{H}(\sigma) \leq -\dfrac{C_1 k}{N}$ for some C_1 (depending on C_5).*

Lemma 4.11. *If C_5 is small enough then*

$$\mathbb{P}_N(\textit{Random alphabet is } C_5 - \textit{typical}) \to 1$$

as $N \to \infty$.

Proof of Lemma 4.10: Let i be a C_5-good index. We will estimate the contribution of the terms starting with i. There are two cases to consider.

1. $\pi_i \leq \pi_{\sigma(i)}$. Then for all $j \in \Gamma_i^-$ we have $(ij) \in I_2$. To see this, observe that

$$r_{ij} \leq \frac{\xi_{ij}}{\xi_{\sigma(i)\sigma(j)}} \cdot \frac{\sum_{j=1}^N \xi_{\sigma(i)\sigma(j)}}{\sum_{j=1}^N \xi_{ij}} \leq \frac{1}{3} \cdot \frac{\sum_{j=1}^N \xi_{\sigma(i)\sigma(j)}}{\sum_{j=1}^N \xi_{ij}} \leq \frac{1}{3} \cdot \frac{0.51N}{0.49N} < 1$$

   Moreover $\gamma_{ij} - \gamma_{\sigma(i)\sigma(j)} \geq \dfrac{\gamma_{\sigma(i)\sigma(j)}}{2}$, whence

$$\sum_{j \in \Gamma_i^-} \frac{(\gamma_{ij} - \gamma_{\sigma(i)\sigma(j)})^2}{\gamma_{ij}} \geq \sum_{j \in \Gamma_i^-} \frac{\gamma^2_{\sigma(i)\sigma(j)}}{4\gamma_{ij}} \geq \sum_{j \in \Gamma_i^-} \frac{\gamma_{\sigma(i)\sigma(j)}}{2} \geq \frac{C_5 N \pi_i}{2S_i} \geq \frac{C_5}{102N}$$

2. $\pi_i \geq \pi_{\sigma(j)}$. Then for all $j \in \Gamma_i^+$ we have $(ij) \in I_3$ and

$$\sum_{j \in \Gamma_i^+} \gamma_{ij} \geq \frac{1}{100NS_i} \sum_{j \in \Gamma_i^+} \xi_{ij} \geq \frac{C_5}{102N}$$

Summing the estimates of both cases over all good i we obtain the result. ■

Proof of Lemma 4.11: Property 1 of typical alphabets holds with high probability by Lemma 4.3 and property 2 holds with high probability due to Lemma 4.2. We need to show that property 3 is unlikely to fail. Let $\sigma \in S_N$. We say that a set I is *representative* if $\forall i \in I, \sigma(i) \neq i$. Note that $\forall \sigma \in \mathcal{S}_k$, there exists a representative set of size $\left[\frac{k}{2}\right]$ as shown by the greedy algorithm.

Choose a representative set of size $m = [k/2]$. We note that $\forall i \in I, \mathbb{P}(i \text{ is not good}) \leq e^{-C_6 N}$ due to a large deviation bound. If I is representative, the events that $i_1, i_2, \ldots, i_m$ are good are independent since the ξ's involved are independent. Let $r = \left[\frac{k}{100}\right] + 1$. Then

$$\mathbb{P}(\text{at least } r \text{ indices are bad for } \sigma) \leq m^r e^{-C_6 N r}$$

and

$$\mathbb{P}(\exists \sigma \in \mathcal{S}_k) \leq N^k m^r e^{-C_6 N r} \leq N^{k+r} e^{-C_6 N r} = e^{((k+r)\ln N - C_6 N)}$$

Summing over k we obtain the result. ■

5 Code

All code was written and executed using Python 3.9 on a 2018 Macbook Pro with 16 GB of 2400 MHz DDR4 memory and a 2.6 GHz 6-Core Intel Core i7 processor. The repository is available on Github at https://github.com/pwerth/python-metropolis-crypto.

5.1 Metropolis Algorithm

By far the most computationally expensive (and time consuming) step of our Metropolis-Hastings algorithm occurs when we must decide whether or not to accept a proposed transition - in particular, it is the computation of the acceptance threshold α. Recall that α is defined as

$$\alpha = \frac{\pi(\tau\sigma)}{\pi(\sigma)}$$

where π is the target distribution of the MCMC, σ is the current state, and $\tau\sigma$ is the proposed state (so τ is a transposition). Denoting the plaintext $s = s_1 \cdots s_n$, we have

$$\begin{aligned} \alpha &= \frac{\pi(\tau\sigma)}{\pi(\sigma)} \\ &= \prod_{k=1}^{n-1} \frac{P_2(\tau\sigma(s_k), \tau\sigma(s_{k+1}))}{P_2(\sigma(s_k), \sigma(s_{k+1}))} \qquad (52) \\ &= \prod_{x,y \in \aleph} \frac{P_2(x,y)^{N_{\tau\sigma}(x,y)}}{P_2(x,y)^{N_\sigma(x,y)}} \qquad (53) \end{aligned}$$

where we define $N_\sigma(x,y)$ as the number of times the digram xy appears in $\sigma(s)$. Equivalently, it is the number of times the digram $\sigma^{-1}(x)\sigma^{-1}(y)$ appears in the plaintext. Note that because a string of length n contains $n-1$ digrams, computation of the product (52) requires $2(n-1)$ multiplications.

However, many of these are unnecessary because, since only two symbols are being swapped, most terms will remain the same in both the numerator and denominator, and hence will cancel each other out. Taking advantage of this fact will greatly improve the efficiency of the algorithm.

If $\tau = (ij)$ transposes the symbols i and j, we can rewrite the acceptance threshold α as

$$\prod_{x \in \aleph} \frac{P_2(x,i)^{N_{\tau\sigma}(x,i)} \cdot P_2(x,j)^{N_{\tau\sigma}(x,j)} \cdot P_2(i,x)^{N_{\tau\sigma}(i,x)} \cdot P_2(j,x)^{N_{\tau\sigma}(j,x)}}{P_2(x,i)^{N_{\sigma}(x,i)} \cdot P_2(x,j)^{N_{\sigma}(x,j)} \cdot P_2(i,x)^{N_{\sigma}(i,x)} \cdot P_2(j,x)^{N_{\sigma}(j,x)}} \tag{54}$$

Because of the large number of terms, many being quite small, we instead computed the logarithm of the product. For the code, this mitigated rounding error, but it conveniently also had benefits for some of the theoretical results (e.g. by allowing us to use inequalities and identities involving logarithms). Thus, instead of computing α, we compute

$$\begin{aligned}
\log(\alpha) &= \log(\pi(\tau\sigma)) - \log(\pi(\sigma)) \\
&= \sum_{x \in \aleph} [N_{\tau\sigma}(x,i)\log(P_2(x,i)) + N_{\tau\sigma}(x,j)\log(P_2(x,j)) + \\
&\qquad\qquad N_{\tau\sigma}(i,x)\log(P_2(i,x)) + N_{\tau\sigma}(j,x)\log(P_2(j,x))] \\
&- \sum_{x \in \aleph} [N_{\sigma}(x,i)\log(P_2(x,i)) + N_{\sigma}(x,j)\log(P_2(x,j)) + \\
&\qquad\qquad N_{\sigma}(i,x)\log(P_2(i,x)) + N_{\sigma}(j,x)\log(P_2(j,x))] \\
&= \sum_{x \in \aleph} [(N_{\tau\sigma}(x,i) - N_{\sigma}(x,i))\log P_2(x,i) + (N_{\tau\sigma}(x,j) - N_{\sigma}(x,j))\log P_2(x,j) + \\
&\qquad (N_{\tau\sigma}(i,x) - N_{\sigma}(i,x))\log P_2(i,x) + (N_{\tau\sigma}(j,x) - N_{\sigma}(j,x))\log P_2(j,x)]
\end{aligned} \tag{55}$$

Note that this sum contains $4N$ terms. Since the values of P_2 are precomputed and stored in memory, lookup occurs in constant time. Therefore if we can do the same for N_σ, then we can compute α using (on the order of) $4N$ steps. This is not only better than $2(n-1)$ because $4N << 2(n-1)$ in practice, but also because it does not scale with n. Therefore, asymptotically, we can run the algorithm just as quickly on longer portions of text than on short ones. For our simulations, $N = 27$ (26 English letters and the space symbol) but n may be as large as $100,000$ (see Tables 2 and 3).

Another benefit of this approach is that the program does not need to apply any permutations to the original string; it just needs to keep track of the counts of each digram. This leads us to the following:

Definition 5.1. Let $\aleph$ be an alphabet containing N symbols. Given a string $s \in \aleph^n$ and a permutation $\sigma \in S_N$, the *character transition matrix* (or more simply, *transition matrix*) of the string $\sigma(s)$ is the $N \times N$ matrix $T_{\sigma(s)}$ whose (i,j)-th entry is $N_\sigma(i,j)$.

In other words, $T_{\sigma(s)}(i,j)$ records the number of times the digram ij appears in the string $\sigma(s)$. Note: we will abbreviate $T_{\sigma(s)}$ to T_σ when the underlying string is clear from context.

Remark. Suppose that *Eng* represented the collection of all English text that was ever written. In this case, $T_{id(Eng)} = T_{Eng}$ gives the empirical frequencies of English language digrams. In particular, we would hope to have $T_{Eng}(i,j) = P_2(i,j)$. Of course, it is impossible to obtain such a string Eng, so we may think of the matrix $(P_2(i,j))$ as an approximation to the theoretical T_{Eng}.

For two $n \times n$ matrices A, B, let $A : B$ denote the sum of all entries in the element-wise product of A and B. That is,

$$A : B = \sum_{i=1}^{n} \sum_{j=1}^{n} A(i,j) \cdot B(i,j)$$

With this notation, we may now express (55) as

$$\log \alpha = \log(\pi(\tau\sigma)) - \log(\pi(\sigma)) = (T_{\tau\sigma} - T_\sigma) : (\log P_2) \tag{56}$$

where $\log P_2$ is the matrix whose (i,j)-th entry is $\log P_2(i,j)$.

Note that because most symbols are unaffected by the transposition τ, the vast majority of entries in $T_{\tau\sigma} - T_\sigma$ will be zero. In particular, if $\tau = (ij)$ then only the rows and columns containing i or j may be non-zero.

Let $N \in \mathbb{N}$. We denote by R_{ij} the matrix obtained from the identity matrix $I_{N\times N}$ by swapping rows i and j. Note that for any $N \times N$ matrix M, left multiplication by R_{ij} swaps rows i and j of M, while right multiplication by R_{ij} swaps the columns i and j of M. The following proposition shows us how to compute $T_{\tau\sigma}$ from T_σ.

Proposition 5.2. *Let $\tau = (ij)$ be a transposition of the i-th and j-th symbols of the alphabet, and let $\sigma \in S_N$. Then the transition matrix of the permutation $\tau\sigma$ is*

$$T_{\tau\sigma} = R_{ij} T_\sigma R_{ij}$$

In other words, $T_{\tau\sigma}$ is obtained from T_σ by first swapping the rows corresponding to symbols i and j, and then swapping the columns corresponding to these symbols.[18]

Proof. When going from T_σ to $T_{\tau\sigma}$ we must consider how to express the digram counts $N_{\tau\sigma}$ in terms of N_σ. This can be derived by considering the relationship between $T_\sigma, T_{\tau\sigma}$ and the underlying string s.

There are 9 cases to consider for a digram $d = s_1 s_2$. In the following table, x and y denote any character that is neither i nor j, and $(T_\sigma R_{ij})(d)$ denotes the entry of the matrix $T_\sigma R_{ij}$ corresponding to N_d.

[18] There is a slight abuse of notation in the statement of Proposition 5.2. In the initial description of R_{ij}, i, j are integer indices corresponding to matrix rows/columns, whereas in the description $\tau = (ij)$ they are symbols in the English alphabet. What is technically meant by R_{ij} is the matrix that swaps the rows or columns corresponding to symbols i and j.

Digram type	$(R_{ij}T_\sigma)(d)$	$(R_{ij}T_\sigma R_{ij})(d)$
ix	jx	jx
jx	ix	ix
xi	xi	xj
xj	xj	xi
ii	ji	jj
ij	jj	ji
ji	ii	ij
jj	ij	ii
xy	xy	xy

Observe that the digram types in third column of the table are indeed the results of applying $\tau = (ij)$ to the digram types in the first column. Thus, because each cell of the matrix T_σ corresponds to a digram with one of the above forms, this proves that $T_{\tau\sigma}$ has the desired form. ■

With the above Proposition, our final form of (56) is

$$\alpha = \log(\pi(\tau\sigma)) - \log(\pi(\sigma)) = (R_{ij}T_\sigma R_{ij} - T_\sigma) : \log P_2$$

Matrix operations are quite fast on a computer, and our program was written to take full advantage of this by using Python's Numpy package.

6 Conclusion

We have presented a theoretical framework for constructing random alphabets and sampling strings from them. In doing so, we defined the Encounter Process, a point process whose limiting distribution is proved to be Poisson. We used this process and large deviations theory to provide strong evidence - which is supported by simulations - that using the digram scoring function H_2 significantly outperforms the single frequency scoring function H_1 in MCMC decryption. We also proved that for reliable decryption using $m = 1$, one needs a massive amount of sample text (on the order of N^5, where N is the size of the alphabet).

A natural continuation of this work is to consider scoring functions for $m \geq 3$, which provide challenges both from a computational standpoint (computation of H_m incurs precision and rounding errors due to very small values of P_m), and a theoretical standpoint (bounds such as (7) become much more complicated with more terms).

Furthermore, although introduced in the context of MCMC decryption, the Encounter Process is interesting in its own right, and could lead to further research in other applications.

References

[1] Jian Chen and Jeffrey S. Rosenthal. "Decrypting Classical Cipher Text Using Markov Chain Monte Carlo". In: Statistics and Computing 22.1 (2012), pp. 397–413.

[2] Zhili Chen et al. More than Word Frequencies: Authorship Attribution via Natural Frequency Zoned Word Distr
URL: https://arxiv.org/pdf/1208.3001.pdf.

[3] Stephen Connor. "Simulation and Solving Substitution Codes". In: Master's Thesis, Department of Statistics, Uni
46.1 (2003), pp. 179–205.

[4] Daryl J Daley and D. Vere-Jones. An Introduction to the Theory of Point Processes. Springer, 2003, p. 24. ISBN: 9780387213378.

[5] Persi Diaconis. "The Markov Chain Monte Carlo Revolution". In: Bulletin of the American Mathematical Society
46.1 (2009), pp. 179–205.

[6] David S. Dummit and Richard M. Foote. Abstract Algebra. John Wiley & Sons, 2004. ISBN: 0471433349.

[7] Charles M. Goldie and Richard G.E. Pinch. Communication Theory. London Mathematical Society Student Texts. Cambridge University Press, 1991. ISBN: 0521404568.

[8] Philip Hall. "The Distribution of Means for Samples of Size N Drawn from a Population in which the Variate Takes Values Between 0 and 1, All Such Values Being Equally Probable". In: Biometrika 19 (1927), pp. 240–245.

[9] W.K. Hastings. "Monte Carlo Sampling Methods Using Markov Chains and Their Applications". In: Biometrika 57.1 (1970), pp. 97–109.

[10] David A. Levin, Yuval Peres, and Elizabeth L. Wilmer. Markov Chains and Mixing Times. American Mathematical Soc., 2008. ISBN: 0821847392.

[11] Nicholas Metropolis et al. "Equation of State Calculations by Fast Computing Machines". In: The Journal of Chemical Physics 21.1 (1953), pp. 1087–1092.

[12] Peter Norvig. English Letter Frequency Counts: Mayzner Revisited. URL: http://norvig.com/mayzner.html.

[13] R.-D. Reiss. A Course on Point Processes. Springer-Verlag, 1993. ISBN: 0387979247.

[14] Albert N. Shiryaev. Probability. Graduate Texts in Mathematics. Springer-Verlag, 1980, p. 342. ISBN: 9780387945491.

[15] Joy A. Thomas Thomas M. Cover. Elements of Information Theory. John Wiley Sons, Inc, 2005, pp. 12–29. ISBN: 9780471241959.

[16] Leo Tolstoy. War and Peace. The Russian Messenger, 1869. ISBN: 0140447938.

[17] Wade Trappe and Larry Washington. Introduction to Cryptography with Coding Theory. Pearson, 2005. ISBN: 0131862391.

[18] Wikipedia. Letter Frequency. URL: https://en.m.wikipedia.org/wiki/Letter_frequency.