Fall 12-20-2016

Cryptanalysis of Homophonic Substitution Cipher Using Hidden Markov Models

Guannan Zhong
San Jose State University

Cryptanalysis of Homophonic Substitution Cipher Using Hidden Markov Models

A Project

Presented to

The Faculty of the Department of Computer Science

San Jose State University

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

by

Guannan Zhong

December 2016

The Designated Project Committee Approves the Project Titled

Cryptanalysis of Homophonic Substitution Cipher Using Hidden Markov Models

by

Guannan Zhong

APPROVED FOR THE DEPARTMENTS OF COMPUTER SCIENCE

SAN JOSE STATE UNIVERSITY

December 2016

Mark Stamp	Department of Computer Science
Thomas Austin	Department of Computer Science
Fabio Di Troia	Department of Computer Science
Richard M. Low	Department of Mathematics

ABSTRACT

Cryptanalysis of Homophonic Substitution Cipher Using Hidden Markov Models

by Guannan Zhong

We investigate the effectiveness of a Hidden Markov Model (HMM) with random restarts as a mean of breaking a homophonic substitution cipher. Based on extensive experiments, we find that such an HMM-based attack outperforms a previously developed nested hill climb approach, particularly when the ciphertext message is short. We then consider a combination cipher, consisting of a homophonic substitution and a column transposition. We develop and analyze an attack on such a cipher. This attack employs an HMM (with random restarts), together with a hill climb to recover the column permutation. We show that this attack can succeed on relatively short ciphertext messages. Finally, we test this combined attack on the unsolved Zodiac 340 cipher.

ACKNOWLEDGMENTS

I am very thankful to my advisor Prof. Stamp for his detailed and patient guidance. Also, I would like to thank Prof. Austin and Prof. Low for their valuable suggestions. Thanks to Fabio Di Troia for his support in double checking the correctness of the code and feedback on my work. Finally, I thank my husband for his support.

TABLE OF CONTENTS

CHAPTER

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

Introduction

Substitution ciphers are one of the earliest methods used to encrypt plaintext. While there are many different kinds of substitution ciphers, the most elementary type is the simple substitution, where each plaintext symbol is mapped in a one-to-one manner to a ciphertext symbol. There are mature algorithms to aid in the decryption of such ciphers, including Jakobsen's Algorithm [6] and methods based on Hidden Markov Models (HMM) [9]. When the ciphertext is short, an HMM with random restarts can be more effective than Jakobsen's Algorithm [11].

A variation on the simple substitution is the homophonic substitution cipher. In a homophonic substitution, a plaintext symbol can be encrypted to more than one ciphertext symbol, which has the effect of flattening the ciphertext statistics, thus making elementary cryptanalysis far more challenging as compared to a simple substitution [10]. In [4], the authors propose and analyze generalization of Jakobsen's Algorithm for homophonic substitution ciphers. This algorithm is based on a nested hill climb approach where an additional outer hill climb layer is added to find appropriate number of cipher symbol that each plain letter is mapped to. The work in [4] has recently been improved—both in terms of accuracy and efficiency—by Magnuson [7]. A method based on HMMs with random restarts has also been successfully applied to the homophonic substitution problem [1, 11].

In this project, we compare the efficiency and accuracy of Magnuson's nested hill climb [7] to an HMM with random restarts [11] for homophonic substitution ciphers. We show that HMM with random restarts has significantly better performance than

the nested hill climb, particularly when the message is short.

We then consider an attack on a cipher that combines a homophonic substitution with a column transposition. Our attack combines an outer hill climb (to recover the column transposition) with an inner HMM with random restarts (to break the homophonic substitution). We apply this attack to a message inspired by the unsolved Zodiac 340 cipher [2], and we also apply the attack to the actual Zodiac 340 cipher.

This paper is organized as follows. In Chapter 2, we review the concepts of simple and homophonic substitution ciphers as well as algorithms to decrypt them, including Jakobsen's Algorithm and its generalization based on a nested hill climb algorithm. In Chapter 3, we show that HMM with random restarts performs better than the nested hill climb in decrypting homophonic substitution ciphers. Then in Chapter 4, we test short homophonic substitution ciphertext messages to determine whether an HMM with random restarts can succeed against messages comparable to the Zodiac 340. Finally, in Chapter 5, we propose a model that combines Jakobsen's Algorithm and an HMM with random restarts and we apply this attack to the unsolved Zodiac 340 cipher.

CHAPTER 2

Background

In this Chapter, we briefly review substitution ciphers and some algorithms to decrypt such kind of ciphers. In Section 2.1, we introduce the concepts of simple substitution cipher and homophonic substitution cipher. In Section 2.2 and Section 2.3, we review the Jakobsen's Algorithm and its generalization based on a nested hill climb approach that are used to decrypt simple substitution ciphers and homophonic substitution ciphers, respectively. Besides these two algorithms, we review the concept and algorithm of hidden Markov model (HMM) in Section 2.4, as well as its application in decrypting substitution ciphers. Finally in Section 2.5, we summarize the work [11] on comparing the performances of HMM and the Jakobsen's Algorithm in decrypting the simple substitution ciphers.

2.1 Substitution Cipher

Substitution cipher is one of the earliest methods used to encrypt plaintext. While there are many different kinds of substitution ciphers, the most elementary type is the simple substitution, where each plaintext symbol is one-to-one mapped to a single unique ciphertext symbol. As a result, the frequency of each cipher symbol is the same as that of its corresponding symbol in the plaintext.

Suppose the ciphertext only contains 26 distinct alphabetic characters without spaces or punctuations. If we decrypt this kind of simple substitution cipher in a brute force way by trying all permutations in the key space, the time complexity is $26! \approx 2^{88}$. However, when the ciphertext is long enough, we can utilize the statistics information from the English corpus to reduce the time complexity significantly. For

example, the most frequent ciphertext symbol can be decrypted as the letter 'e', since 'e' is the most common letter in English. Similarly, we decrypt the second frequent ciphertext symbol to letter 't', and so on. Once we obtain part of the mapping, we can derive the whole mapping by common pattern analysis or exhaustively trying the permutations of the remaining cipher symbols.

A variation on the simple substitution cipher is the homophonic substitution cipher. In a homophonic substitution cipher, the mapping is not one-to-one. A plaintext symbol can be encrypted to more than one ciphertext symbols. A homophonic substitution cipher will tend to flatten the ciphertext statistics, making elementary cryptanalysis far more challenging as compared to a simple substitution [10].

2.2 Jakobsen's Algorithm

Jakobsen's Algorithm is an efficient and fast algorithm in decrypting simple substitution ciphers. We outline the main steps as follows [9]:

1. Make initial guess for the key using frequency counts.
2. Compute oldScore.
3. Modify key by swapping two elements.
4. Compute newScore.
5. If newScore $>$ oldScore: oldScore $=$ newScore; Else: undo the swap operation in the step 3.
6. Goto 3.

Here are some detailed explanations for the above procedure.

- Score: Let $D = d_{ij}$ be digraph distribution corresponding to putative key K. Let $E = e_{ij}$ be digraph distribution of English language. Both D and E are matrices with dimension to be 26×26. The score is then defined as follows:

$$\text{score}(K) = d(D, E) = \sum_{i,j} |d_{ij} - e_{ij}| \tag{1}$$

- Swap: Let $K = k_1, k_2, \ldots, k_{26}$ be the initial putative key. Let '|' represents the swap operation. We swap the elements in the following order:

round 1:	$k_1 \mid k_2$	$k_2 \mid k_3$	$k_3 \mid k_4$	$\ldots$	$k_{23} \mid k_{24}$	$k_{24} \mid k_{25}$	$k_{25} \mid k_{26}$
round 2:	$k_1 \mid k_3$	$k_2 \mid k_4$	$k_3 \mid k_5$	$\ldots$	$k_{23} \mid k_{25}$	$k_{24} \mid k_{26}$	
round 3:	$k_1 \mid k_4$	$k_2 \mid k_5$	$k_3 \mid k_6$	$\ldots$	$k_{23} \mid k_{26}$		
$\vdots$		$\vdots$		$\therefore$			
round 23:	$k_1 \mid k_{24}$	$k_2 \mid k_{25}$	$k_3 \mid k_{26}$				
round 24:	$k_1 \mid k_{25}$	$k_2 \mid k_{26}$					
round 25:	$k_1 \mid k_{26}$						

  Restart the swap sequence from the beginning whenever the score improves. The algorithm terminates when there is no improvement in score in round 25.

The core of the Jakobsen's Algorithm is that it provides a method to obtain the new D matrix from the old D through swapping rows and columns of the old D matrix instead of decrypting the ciphertext again. The rows and columns to be swapped are determined by elements in key to be swapped. Since Jakobsen's Algorithm decrypts the ciphertext only once, this algorithm is fast and its speed is independent of the length of the ciphertext.

2.3 Nested Hill Climb Algorithm

In paper [4], the authors propose an algorithm that is fast and effective in attacking the homophonic substitution cipher. Assume that the ciphertext is in English containing only 26 upper case letters with no spaces or punctuations. In a homophonic substitution cipher, a plain letter can be encrypted to one of a set of distinct cipher symbols. Let $n_a, n_b, \ldots, n_z$ be the number of ciphertext symbols that correspond to plain letters $A, B, \ldots, Z$. The algorithm consists of three layers:

- Inner hill climb: A generalization of the Jakobsen's Algorithm [6] that can quickly get the putative digram matrix through operations on the old putative digram matrix.
- Random initial key generator: To overcome the shortcoming of getting the local maximum in the inner hill climb layer.
- Outer hill climb: To obtain appropriate combination of values of $n_a, n_b, \ldots, n_z$.

This algorithm was proved to be effective by decrypting large number of test cases [4] and has recently been improved—both in terms of accuracy and efficiency—by Magnuson [7].

2.4 Hidden Markov Model

Hidden Markov Model (HMM) describes a Markov Process with hidden states. The observed states are probabilistically related to the hidden states and thus provide information on the hidden states. A high level structure of the HMM model is shown in Figure 1.

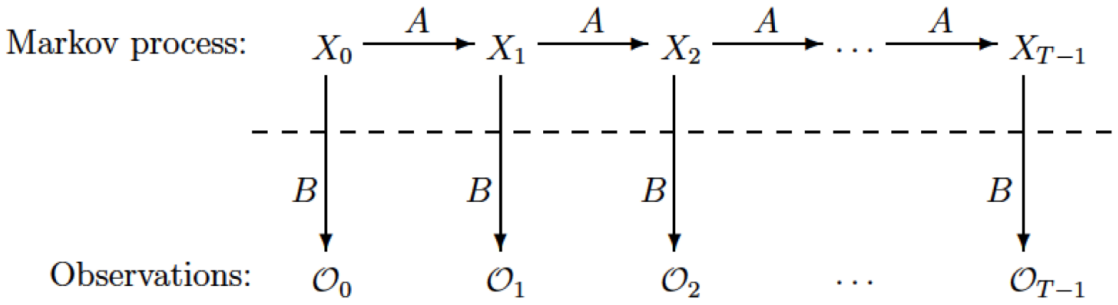

Figure 1: Structure of the HMM Model

2.4.1 Notations

We follow the same notations of symbols in HMM as that in [9]:

T = the length of the observation sequence

N = the number of hidden states in the model

M = the number of observation symbols

$Q = \{q_0, q_1, \cdots, q_{N-1}\}$ = the states of the Markov process

$V = \{0, 1, \ldots, M-1\}$ = set of possible observations

$X = (X_0, X_1, \cdots, X_{T-1})$ = hidden state sequence. Each X_t takes a value in Q.

$\mathcal{O} = (\mathcal{O}_0, \mathcal{O}_1, \cdots, \mathcal{O}_{T-1})$ = observation sequence. Each $\mathcal{O}_t$ takes a value in V.

A = the $N \times N$ state transition probability matrix

B = the $N \times M$ observation probability matrix

π = the N-dimensional row vector representing the initial state distribution

b_o = the N-dimensional row vector that is the transpose of the o-th column of the matrix B

$\cdot$: matrix-matrix multiplication or matrix-vector multiplication

$*$: element-wise multiplication between two vectors with same dimensions

$\|v\|_1$: the sum of the absolute value of each element in vector v

Especially, for the matrix $A = \{a_{ij}\}$ where $i, j = 0, \ldots, N-1$, the element a_{ij} is the probability of transiting from state q_i at time t to state q_j at time $t+1$:

$$a_{ij} = P(\text{state } q_j \text{ at } t+1 | \text{ state } q_i \text{ at } t) \tag{2}$$

For the matrix $B = \{b_j(k)\}$ where $j = 0, 1, \ldots, N-1, k = 0, 1, \ldots, M-1$, the element $b_j(k)$ is the probability of seeing the observation k given a hidden state q_j:

$$b_j(k) = P(\text{observation } k \text{ at } t | \text{ state } q_j \text{ at } t) \tag{3}$$

An HMM model can be denoted as $\lambda = (A, B, \pi)$.

2.4.2 Alpha Pass

Given a model $\lambda = (A, B, \pi)$ and an observation sequence $\mathcal{O}$, alpha pass can give the probability of the occurrence of the observation sequence $\mathcal{O}$, which is denoted as $P(\mathcal{O}|\lambda)$. The algorithm is described as follows:

For each $t = 0, 1, \ldots, T-1$, we define a N-dimensional row vector α_t, where its i-th component represents the conditional probability $P(\mathcal{O}_0, \mathcal{O}_1, \ldots, \mathcal{O}_t, x_t = q_i|\lambda)$. We can then compute iteratively:

$$\begin{aligned} \alpha_0 &= \pi * b_{\mathcal{O}_0}, \\ \alpha_t &= (\alpha_{t-1} \cdot A) * b_{\mathcal{O}_t}, \quad t = 1, 2, \cdots, T-1. \end{aligned} \tag{4}$$

Finally, we have

$$P(\mathcal{O}|\lambda) = \|\alpha_{T-1}\|_1 \tag{5}$$

Alpha pass is also called *forward algorithm*.

2.4.3 Beta Pass

Given a model $\lambda = (A, B, \pi)$ and an observation sequence $\mathcal{O}$ as well as the result of alpha pass, we can find the optimal state sequence. i.e., uncover the hidden sequence.

The algorithm is described as follows:

For each $t = 0, 1, \ldots, T-1$, we define a N-dimensional row vector β_t, where its i-th component represents the conditional probability $P(\mathcal{O}_{t+1}, \mathcal{O}_{t+2}, \ldots, \mathcal{O}_{T-1}, | x_t = q_i, \lambda)$. We can then compute iteratively:

$$\begin{aligned} \beta_{T-1} &= \{1, 1, \ldots, 1\}, \\ \beta_t &= (\beta_{t+1} * b_{O_{t+1}}) \cdot A^T, \quad t = T-2, \cdots, 1, 0. \end{aligned} \tag{6}$$

This iteration is also called *backwards algorithm* compared to the *forward algorithm*.

Finally, for each $t = 0, 1, \ldots, T-1$, we define a N-dimensional row vector γ_t, where its i-th component is the conditional probability $P(x_t = q_i | \mathcal{O}, \lambda)$. Since α_t represents the partial probability up to time t and β_t represents the partial probability after time t, we have

$$\gamma_t = \frac{\alpha_t * \beta_t}{P(\mathcal{O}|\lambda)} \tag{7}$$

The most likely hidden state at time t is the state q_m for which the m-th element in γ_t is the maximum.

2.4.4 HMM Training

Given N, M and the observation sequence $\mathcal{O}$, one common task is to train a model that best fits observation sequence $\mathcal{O}$. That is to find a model λ that maximizes the probability of the occurrence of $\mathcal{O}$. This is an Expectation—Maximization (EM) algorithm:

For each $t = 0, 1, \ldots, T-2$, we define a $N \times N$ matrix Γ_t, where its (i, j) component represents the conditional probability $P(x_t = q_i, x_{t+1} = q_j | \mathcal{O}, \lambda)$, the probability

of transiting from state q_i at time t to state q_j at time $t+1$. Γ_t can be computed as

$$\Gamma_t = \frac{(\alpha_t^T \cdot (b_{O_{t+1}} * \beta_{t+1})) * A}{P(\mathcal{O}|\lambda)}. \tag{8}$$

A model $\lambda = (A, B, \pi)$ can be re-estimated using γ and Γ as follows:

$$\pi = \gamma_0 \tag{9}$$

$$a_{ij} = \sum_{t=0}^{T-2} \Gamma_t(i,j) \Big/ \sum_{t=0}^{T-2} \gamma_t(i), \quad i,j = 0,\cdots,N-1 \tag{10}$$

$$b_j(k) = \sum_{\substack{t\in\{0,1,\ldots,T-1\} \\ \mathcal{O}_t=k}} \gamma_t(j) \Big/ \sum_{t=0}^{T-1} \gamma_t(j), \quad j = 0,\cdots,N-1, k = 0,\cdots,M-1. \tag{11}$$

The training steps can be summarized as follows:

1. Initialize the model, $\lambda = (A, B, \pi)$.
2. Compute α, β, γ and Γ.
3. Re-estimate the model λ.
4. If the score $\log(P(\mathcal{O}|\lambda))$ increases, goto step 2.

The stop condition can be customer defined maximum iterations or improvement of the score less than a predefined threshold.

2.4.5 HMM in Decrypting Substitution Cipher

To apply HMM in decrypting the substitution cipher, we set the hidden states to be the 26 English letters denoted as 0, 1, ..., 25. Then the A matrix is the digraph probability. For example, the element a_{01} is the probability of seeing 'b' for the next character given seeing 'a' in the current character. The A matrix can be initialized by counting the frequencies of the digram in the English corpus. It also can be initialized from the corpus in a specific field if we know the field of the plaintext.

Now B is a $(N \times M)$-dimensional matrix with M represents the number of distinct cipher symbols. The (j, k) entry of the B matrix, $b_j(k)$, represents the probability that the $(j+1)$-th letter in the plaintext is encrypted to the $(k+1)$-th cipher symbol. We usually randomly initialize the B matrix and then train the HMM as in Section 2.4.4. After finishing training the HMM, we can read the encrypt rule from the final B matrix. That is, the $(j+1)$-th plain letter is encrypted to the (k^*+1)-th cipher symbol where $k^* = \text{argmax}_{0 \le k \le M-1} b_j(k)$. For a homophonic substitution cipher, since a letter can be encrypted to more than one cipher symbols but each cipher symbol only corresponds to one plain letter, we say that the $(k+1)$-th cipher symbol represents the (j^*+1)-th plain letter where $j^* = \text{argmax}_{0 \le j \le N-1} b_j(k)$.

2.4.6 HMM with Random Restarts

When training an HMM model λ, we have to feed a random guess of the initial B matrix. As a hill climb approach, the algorithm is not guaranteed to reach the global maximum. So in practice, we usually train the model multiple times with different initial B matrices and then choose the best result among all these restarts. This greatly increases the probability to obtain or close to the global maximum.

2.5 Compare HMM to Jakobsen's Algorithm in Decrypting Simple Substitution Cipher

In paper [11], the authors compare the performances of the Jakobsen's Algorithm to that of HMM with random restarts in decrypting the simple substitution cipher. In cryptology, 80% accuracy or more means success in decrypting the ciphertext because the remaining 20% can be decrypted by its context. Figure 2 shows that when the ciphertext is short (a few more than 200 characters), HMM with random restarts can be more effective than Jakobsen's Algorithm. We thus expect HMM with random

restarts can perform better in the homophonic substitution cipher.

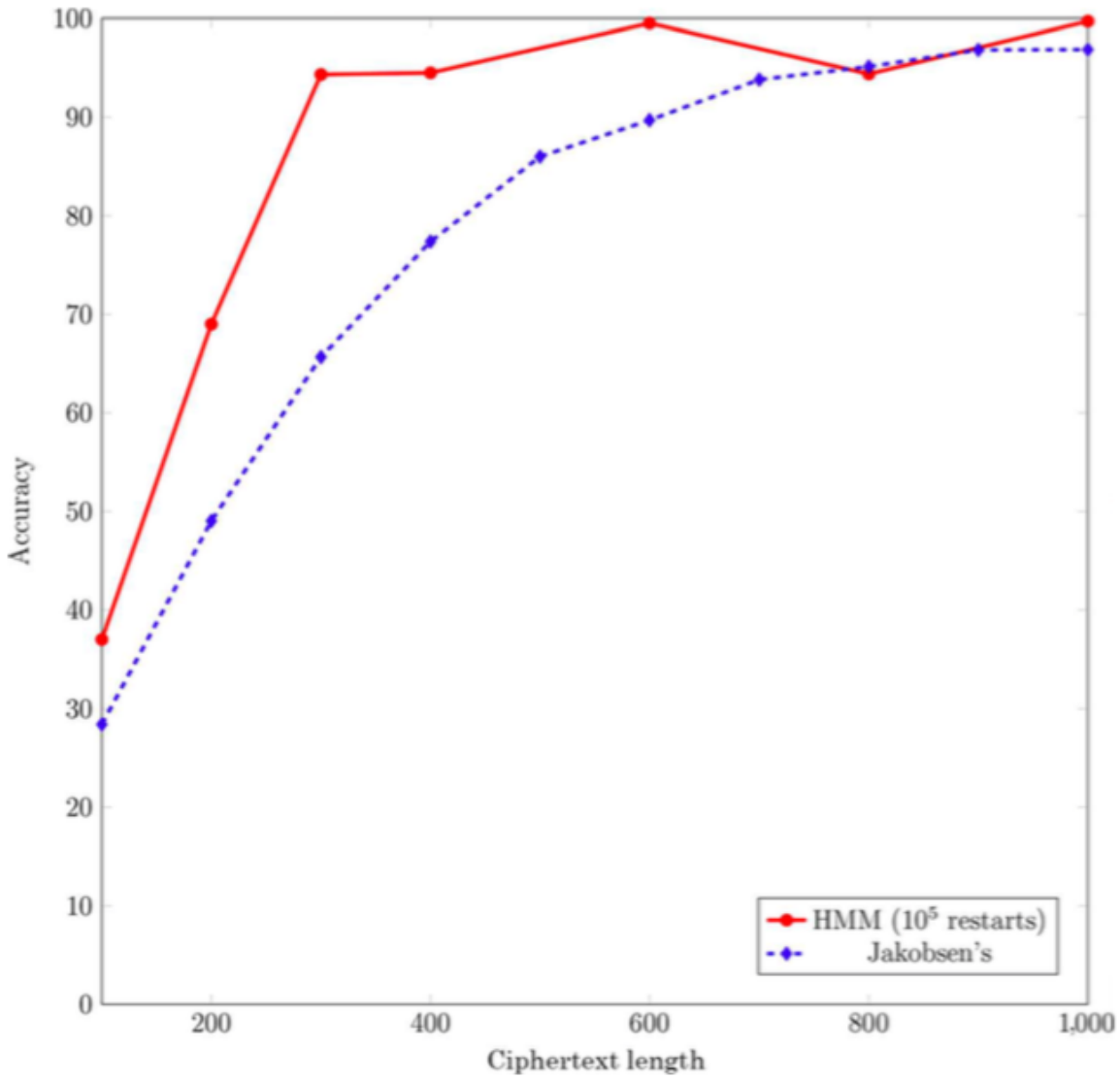

Figure 2: Jakobsen's vs. HMM [11].

CHAPTER 3

Compare HMM with Random Restarts to Jakobsen's Algorithm in Decrypting the Homophonic Substitution Cipher

Besides in decrypting simple substitution ciphers, HMM with random restarts is also successful in decrypting the homophonic substitution problem [1, 11]. In this Chapter, we do experiments to compare the efficiency and accuracy of Magnuson's nested hill climb algorithm [7] to that of HMM with multiple random restarts [11] in decrypting homophonic substitution ciphers. The results of this research leads to additional experiments on the unsolved Zodiac 340 cipher [2].

3.1 Experiment and Results on Magnuson's Nested Hill Climb Algorithm

The code we use for the nested hill climb algorithm is the improved version by Magnuson [7]. We create various test cases, run the code on those test cases and visualize the results in graphs.

3.1.1 Test Cases

We create various test cases with different parameters listed as follows:

- The plain texts are the different subsequences in BrownCorpus [5].
- Number of ciphertext symbols: 28, 35, 45, 55, 65, 75, 85, 90 (8 different values).
- Lengths of the plain texts: 300, 400, 500, 600, 700, 800, 900, 1000, 1200, 2000, 3000, 4000, 5000, 6000, 7000, 8000, 9000, 10000 (18 different values). For each length, 3 plain texts were generated.

- The frequency distribution for each plain text letter is shown in Table 1 as in [7]. n is the total cipher symbols and $n_e, n_t, \ldots, n_z$ are the number of symbols letter $e, t, \ldots, z$ can be mapped to and they are listed from the most common letter to the least common letter in English corpus. The sum of $n_e, n_t, \ldots, n_z$ is n. This frequency distribution is designed in this particular way to flatten the statistic information. Thus ciphertexts generated by the keys with this frequency distribution are hard to decrypt by simple frequency analysis.

Table 1: Frequency distribution [7]

n	n_e	n_t	n_a	n_o	n_i	n_n	n_s	n_r	n_h	n_d	n_l	n_c	n_u	n_m	n_f	n_w	n_g	n_y	n_p	n_b	n_v	n_k	n_x	n_j	n_q	n_z
28	2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
35	4	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
45	5	4	3	3	3	3	3	2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
55	7	5	4	4	4	3	3	3	3	2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
65	8	6	5	5	5	4	4	4	4	2	2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1
75	9	7	6	6	5	5	5	4	4	3	3	2	2	2	1	1	1	1	1	1	1	1	1	1	1	1
85	11	8	7	7	6	6	5	5	5	3	3	2	2	2	2	1	1	1	1	1	1	1	1	1	1	1
90	12	9	7	7	6	6	6	5	5	4	3	2	2	2	2	2	1	1	1	1	1	1	1	1	1	1

3.1.2 Results

For the Magnuson's nested hill climb algorithm, we summarize the 144 accuracies in a 3-d line graph in Figure 3. Figure 3 shows that the success rate increases with longer ciphertexts and decreases with those contain larger number of distinct cipher symbols, as expected. Figure 4 shows the results more clearly in a bar plot. Especially, in Figure 4, the nested hill climb algorithm performs perfect on ciphertext with length equal or larger than 3000, but not that good on ciphertext with length less than 1200 (number of distinct symbols less than or equal to 90). As a result, in order to compare the nested hill climb algorithm with the HMM with random restarts, we only focus on the ciphertext with length less than or equal to 1200 when doing experiments in HMM with random restarts.

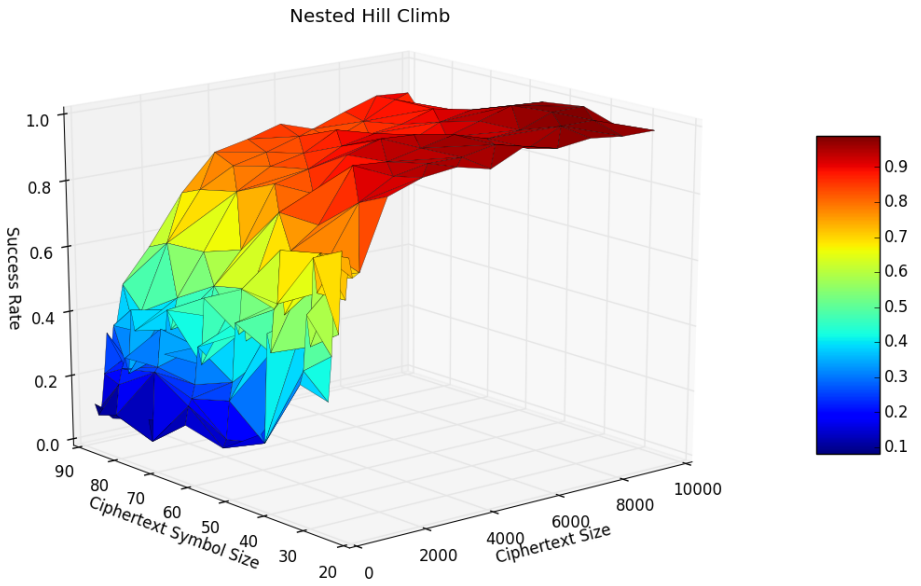

Figure 3: Success rate of the Nested Hill Climb Algorithm for different ciphertext length and different number of distinct cipher symbols

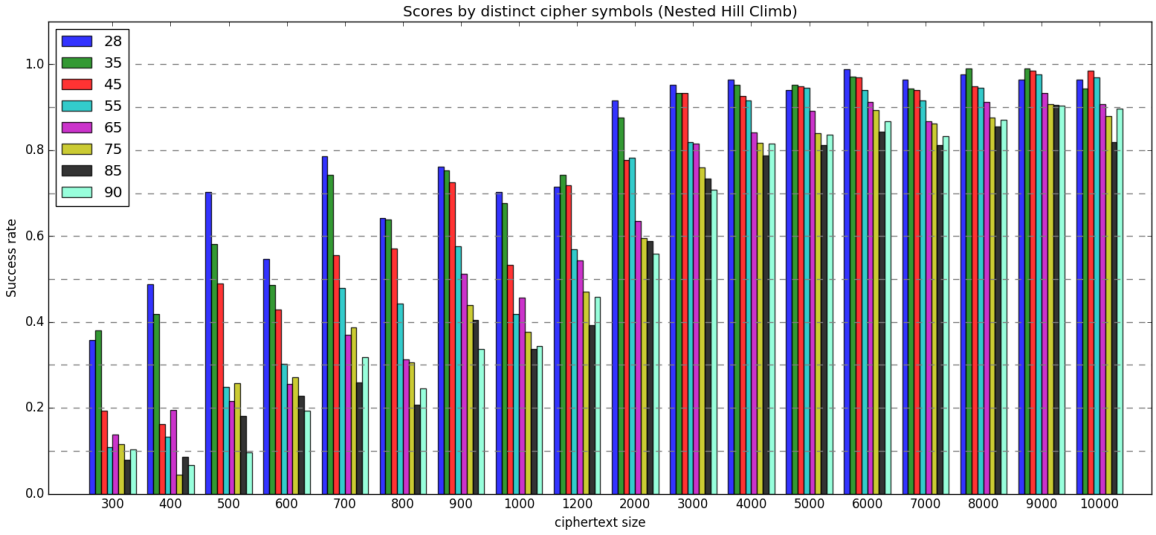

Figure 4: Success rate of the Nested Hill Climb Algorithm for different ciphertext length and different number of distinct cipher symbols

3.2 Experiment and Results on HMM with Random Restarts

As shown in Section 2.5, the HMM with random restarts performs better than the Jakobsen's Algorithm in decrypting the simple substitution cipher. Here we do

experiments to confirm our expectation that the HMM with random restarts would outperform the nested hill climb algorithm in decrypting the homophonic substitution cipher.

3.2.1 Test Cases

For HMM with random restarts, we do experiments on a subset of the test cases used in the nested hill climb algorithm, that is,

- number of distinct cipher symbols = [28, 35, 45, 55, 65, 75, 85, 90]
- lengths = [300, 400, 500, 600, 700, 800, 900, 1000, 1200]
- restarts = 200,000

The A matrix for the HMM algorithm is generated from English corpus. We keep the A matrix fixed when training the HMM.

3.2.2 Choose the Number of Iterations

When training HMM, more iterations in the EM process give better convergence but cost more time. Here we do experiments on two ciphertexts to determine the appropriate iteration steps that balances the convergence and running time. Both subplots in the first row of Figure 5 are the normalized histogram of accuracies consisting of 2000 restarts when the ciphertext has length 300 and consists of 28 distinct symbols. The difference is that the upper left one is the result when the number of iterations is 200 in each restart, whereas the upper right on is the result when the number of iterations is 500. There are no significant differences in the distribution of the accuracies between the case with 200 steps and that with 500 steps.

We do the same kind of experiment on another ciphertext which contains 1200 characters and 55 distinct cipher symbols. There are still no significant differences in the distribution of the accuracies. Therefore, it is safe to set the training steps to be 200 instead of 500. From now on, all the number of iterations in HMM experiments are set to be 200 if not explicitly specified.

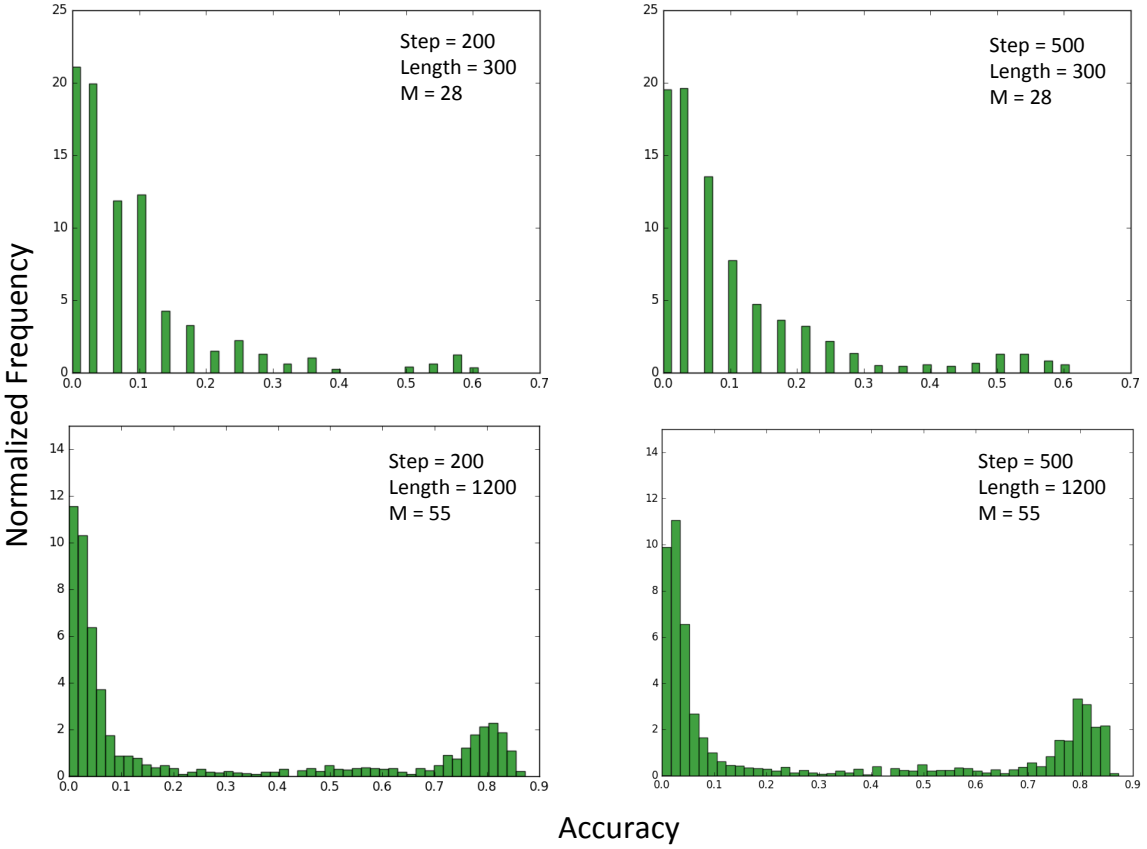

Figure 5: Normalized histograms of the accuracies with different steps and lengths.

3.2.3 Initialize the B Matrix

When training an HMM model, since we do not have any prior information on the B matrix or any preferences to a particular entry of the B matrix, it is reasonable to randomly initialize the B matrix and make each entry of the B matrix close to 1/26.

Here and afterwards, we always randomly initialize the B matrix by first sampling each entry with the following distribution and then normalizing B row by row:

$$b_i(j) \sim \text{base} + \text{uniform}[0, 1] \tag{12}$$

where 'base' is a constant.

To choose the proper value of base, we do the experiment on the ciphertext which has 300 characters and consists of 28 distinct cipher symbols. We compare the performances of HMM on this test case with different bases: 51, 5.5, 2.5 and 0.1. Figure 6 shows the normalized histograms of accuracies of these four base values by running the test case 1000 times. The lower right histogram gives the largest accuracy and larger accuracies appear more frequent. That is, base = 0.1 gives better performance than the other three choices of the value of base.

This result can be explained theoretically. Before normalization, for large base like 50.5, each element is around the value of 51. For small base such as 0.1, each element is almost uniformly random in $[0, 1]$. The differences between base = 50.5 and base = 0.1 is that the latter one provides relative larger fluctuations among the entries in the B matrix and thus HMM with random restarts can explore more search space. A rigorous analysis is as follows:

Before normalization, each entry in one row follows the distribution of

$$X_i' = b + Z_i, \tag{13}$$

where $Z_i \sim$ Uniform[0, 1] for $i = 1, \ldots, M$. After normalization, we have

$$X_i = \frac{X_i'}{\sum_{i=1}^{M} X_i'}, \tag{14}$$

such that $\sum_{i=1}^{M} X_i = 1$. We define the variance among all the normalized elements as

$$\text{Var} = \mathbb{E}\left[\sum_{i=1}^{M}\left(X_i - \frac{1}{M}\right)^2\right] = \mathbb{E}\left[\sum_{i=1}^{M} X_i^2\right] - \frac{1}{M} = \mathbb{E}\left[\frac{(b+Z_1)^2 + \cdots + (b+Z_M)^2}{(Mb + Z_1 + \cdots + Z_M)^2}\right] - \frac{1}{M}\,. \tag{15}$$

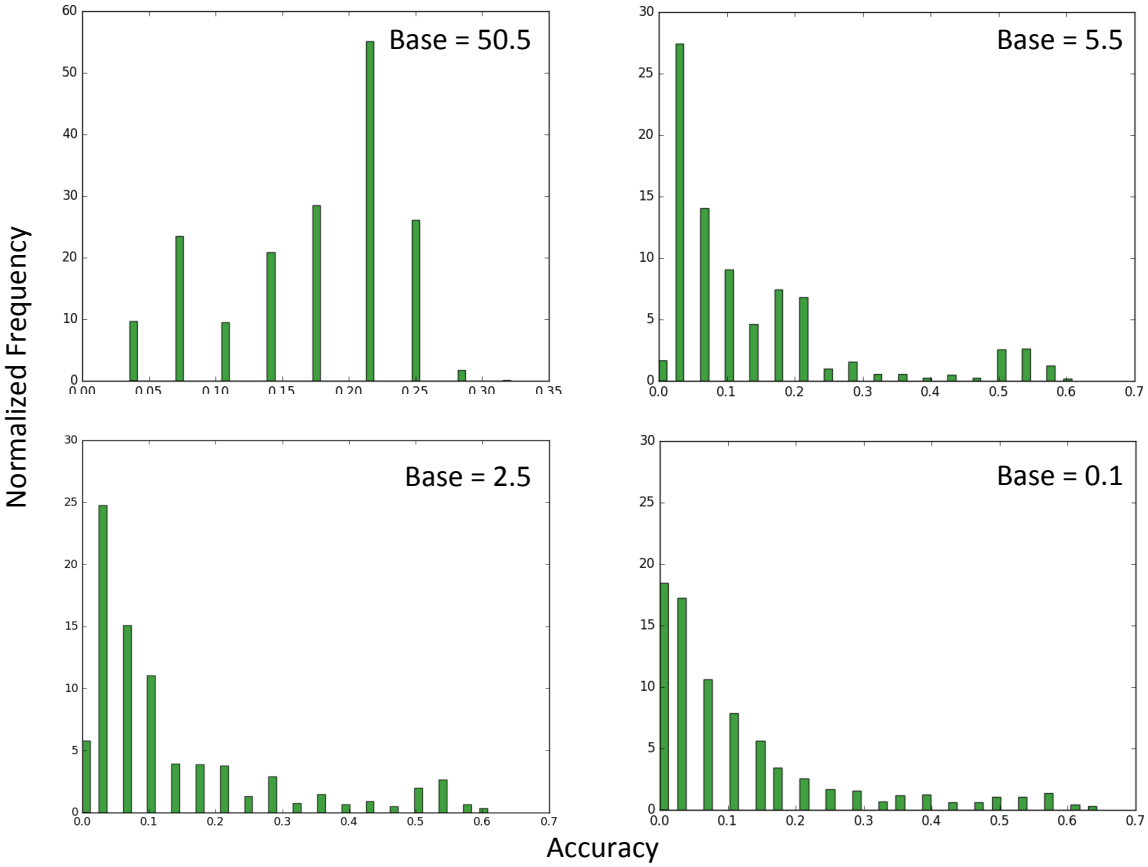

Figure 6: Normalized histograms of the accuracies on the same ciphertext but with different base values.

For each b, we can calculate the expectation in (15) numerically by first sampling P groups of random numbers $\left\{Z_1^{(i)}, \cdots, Z_M^{(i)}\right\}$ and then approximating the expectation by the sample mean as

$$\mathbb{E}\left[\frac{(b+Z_1)^2+\cdots+(b+Z_M)^2}{(Mb+Z_1+\cdots+Z_M)^2}\right] \approx \frac{1}{P}\sum_{i=1}^{P}\frac{\left(b+Z_1^{(i)}\right)^2+\cdots+\left(b+Z_M^{(i)}\right)^2}{\left(Mb+Z_1^{(i)}+\cdots+Z_M^{(i)}\right)^2}. \tag{16}$$

The scatter plot in Figure 7 shows the relation beween the base value and the variance among the entries in one row of the B matrix for $P = 10,000$. Indeed, smaller base value gives larger variance and thus the algorithm can explore more search space, leading to a better performance of HMM. From now on, all the base values in later HMM experiments are set to be 0.1 if not explicitly specified.

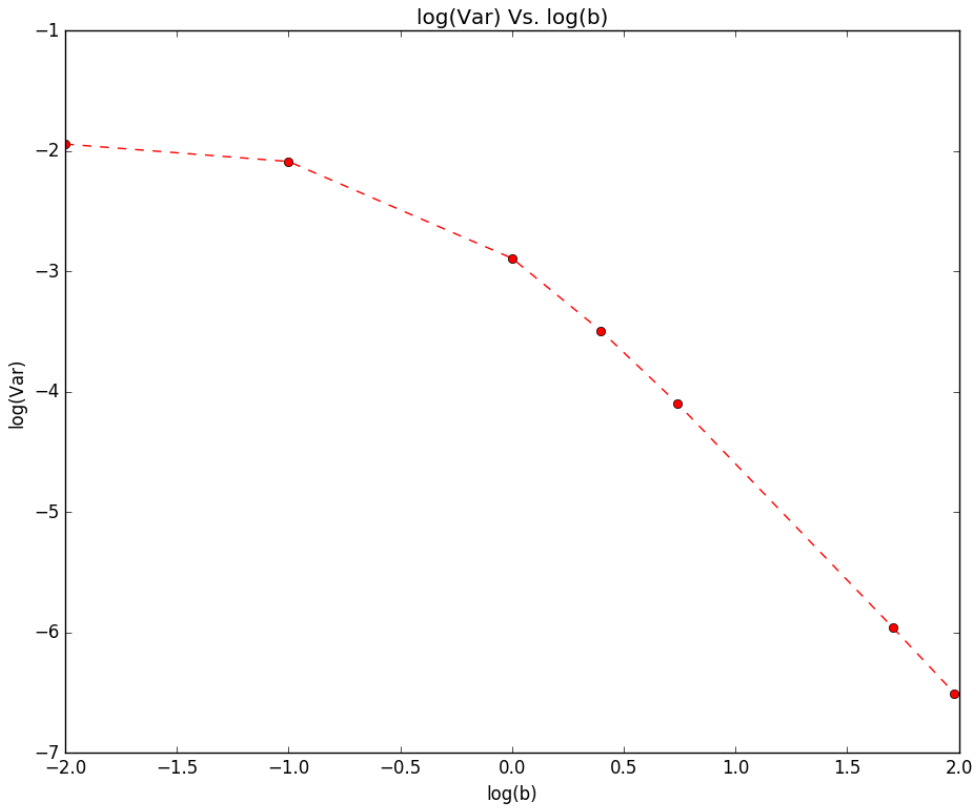

Figure 7: Base vs. Variance.

3.2.4 HMM Results

Due to time limit, we do the experiments with the number of random restarts up to 200,000. The relationships between the accuracies and ciphertext length, as well as the number of distinct symbols are shown as Figure 8 (3-d line graph) and in Figure 9 (bar plot). By comparing Figure 4 and Figure 9, we can see that HMM with random restart performs better than the nested hill climb algorithm in decrypting the homophonic substitution cipher when ciphertext length ≤ 1200. Appendix A shows the trend of accuracies when the number of restarts increases from 10 to 100,000. It is clear that more restarts gives higher accuracy, as expected.

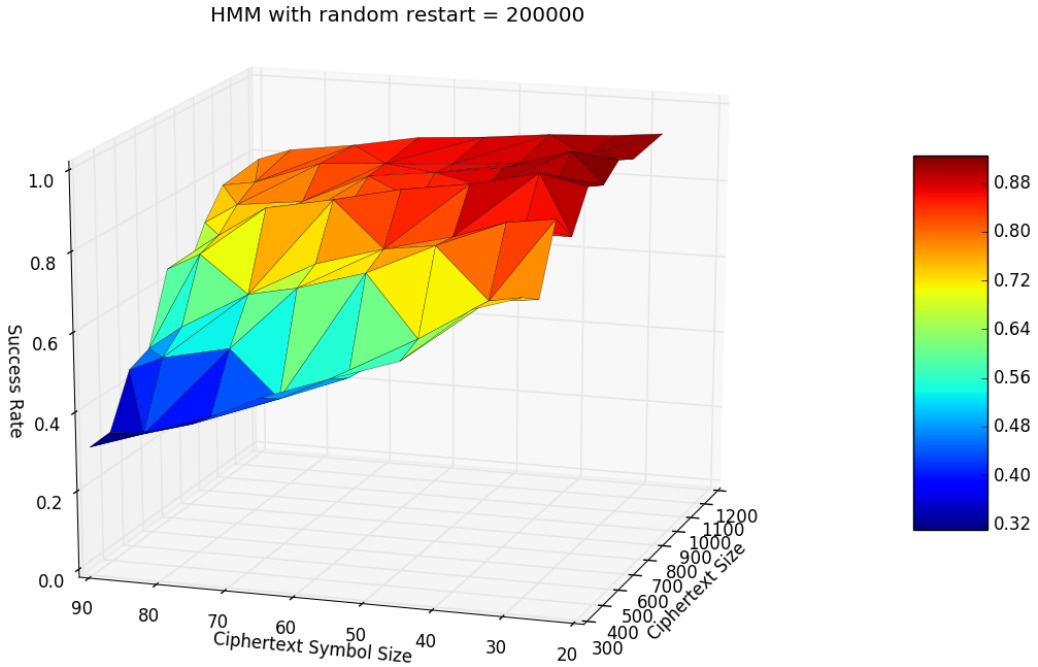

Figure 8: Restart = 200,000

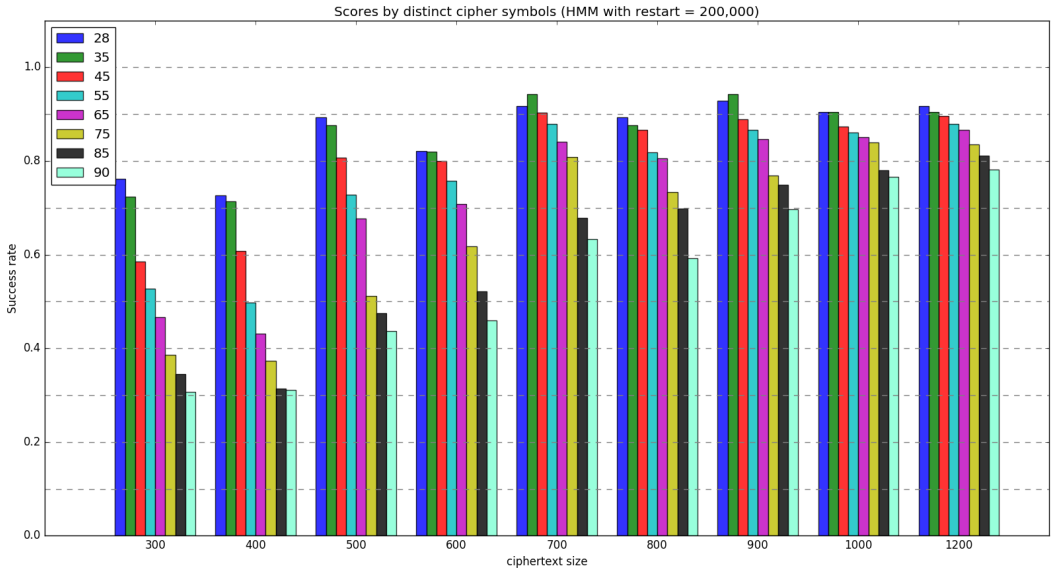

Figure 9: Restart = 200,000

CHAPTER 4

HMM with Random Restarts in Decrypting Fake Zodiac 340

Zodiac killer was a serial killer in late 1960s and seven persons in San Francisco were targeted. Five of them were dead and the other two were injured. Nobody knows who the killer is or whether the killer is still alive or not. During Zodiac's killing period, he sent letters to the police or newspaper. Some of them were threatening letter saying someones would be killed. Some of them were encrypted. Zodiac 408 is one ciphertext that is solved and Zodiac 340 has not been uncovered yet.

Before applying HMM with random restarts to decrypt Zodiac 340, in this Chapter, we test the performance of HMM with random restarts in decrypting the fake Zodiac 340. In Section 4.1, we briefly review the decrypted Zodiac 408. In Section 4.2, we introduce the fake Zodiac 340, a homophonic substitution cipher created from Zodiac 408. We apply the HMM with random restarts to the fake Zodiac 340 using fixed A matrix in Section 4.3 and reestimating the A matrix in Section 4.4. Finally, in Section 4.5, we compare the performances of keeping A fixed and reestimating A in decrypting a simple substitution cipher.

4.1 Zodiac 408

Zodiac 408 is a homophonic substitution cipher contains 408 characters and was solved within six days. The original Zodiac 408 ciphertext and the decrypted plain text are shown as Figure 10.

There are a few misspelling words. For example, "anamal" should be "animal" and "sli" should be "slow". The reason is probably that after encrypted using the

Figure 10: Zodiac 408 ciphertext [12]

homophonic substitution cipher, some cipher symbols were intensionally replaced by another similar symbol. Additionally, The last 18 characters are meaningless. Probably Zodiac thought creating those misspelled or meaningless words can bring the police some troubles in decrypting the cipher, however, understanding the content for those who have the key would not be affected.

4.2 Fake Zodiac 340

There is a cipher challenge on the MysteryTwister website [8] which is inspired by the solved Zodiac 408 and the unsolved Zodiac 340. The plaintext of the fake Zodiac 340 is a subsequence of the solved Zodiac 408 with length equals to 340 and was encrypted to be a homophonic substitution cipher using 65 distinct symbols. The purpose of generating the fake Zodiac 340 is to mimic the real Zodiac 340 and try to get some ideas and guideline to decrypt the Zodiac 340. The ciphertext are represented in a sequence of integers range from 1 to 65 and was formed as 20 rows × 17 columns [8]:

```
 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17
18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34
35 36 23 37 10 38  3 29 39 40 41 14 42 43 44 45 46
47 11 48 26  2  9 28 11 49 50  1  2 51 49 52 39 25
 1 11 45 53 54 36 15 55 55 25 24 38 19 54 21 22 43
30 25 32 52 37  1 57 27 53 14 58 40 30 27 51 22 11
12 20 41 15 43 57 47 37  3 58 22  2 15 59 52 17 17
38 31  4 26  8  9 57 60 32  5 27 46  3 44 12 12 10
61 20 24 32  5 45 53 20 39 31 24 45 27 46 56 26 17
 2  1 11 49 54 62 63 25 55 14 37 21 25  3 38 26 57
51 28 64  1 50  7  8 17 19  5 56 64 19 40 56 44 26
44 16 22 41 52 51  1 21 25 47 11 51 47  2  9 38 46
54 26 35 52 61  5  6 10 17  2 20 51 50  7 17  8 19
64 21 60 32  5 58 65 24  9 47 61 14 57 26 50  3 17
 9 44 40 38 12 26 61 54 65 60 23 39 65 44 52 39 20
19 25 34 52 43 30 54 65 31 43 50 10 17  2 45 55 65
27 15 24  2 60 50 51 40 50 11 31 41 24 27 60 13 58
65 34 15  2 17 14 21 38  1 44 49 60 36 57  8 22 61
54 30 59 31 55 32 65 52 42 27 25 56  2  1 59  5  5
19  5 15 41 28 20 27 54 58 54 27 53 35 63  7 38 28
```

In paper [4], the authors solved the fake Zodiac 340 using the nested hill climb algorithm. The plain text is the same as that of the solved Zodiac 408 except one sentence. We summarize the homophonic mapping in Table 2.

Table 2: Key of the fake Zodiac 340

plain letter	a	b	c	d	e	f	g	h	i	k	l	m	n	o	p	r	s	t	u	v	w	x	y
	22	19	21	51	5	36	12	35	1	4	2	32	11	15	13	41	24	27	23	61	50	62	65
	47		34		14	42	49	46	3	6	8	39	37	31	16	55	29	38	33				
	52				18	59		53	7	48	9	58	44	40	63	56	30	45	43				
cipher symbol					20				10		17			60			57						
					25				26														
					54				28														
					64																		

4.3 Decrypt the Fake Zodiac 340 Using HMM with the Fixed A Matrix

When applying HMM to decrypt submission ciphers, the A matrix is the digram frequency of the plaintext. In general, if we do not know other information of the plaintext except it is an English text, we usually generate A from the classic English corpus. On the other hand, if we know the plaintext is related to a specific field, we can generate a better A from the corpus in the particular field. In the next two subsections, we decrypt the fake Zodiac 340 using HMM with the fixed A matrix generated from the Zodiac 408 and English Corpus, respectively.

4.3.1 Decrypt the Fake Zodiac 340 Using HMM with the Fixed A From the Zodiac 408

The A matrix we use here is from the plain text of the solved Zodiac 408. The simulation result is summarized in Figure 11 as the histogram of the 100 accuracies, where each accuracy is the best result among all 10,000 restarts. Since the authors of paper [4] solve the fake Zodiac 340 cipher with accuracy equals 0.7231, here we consider an accuracy ≥ 0.72 as success. From Figure 11, we can see that most of the accuracies are above 0.72. As a result, 10,000 restarts is enough to decrypt the fake Zodiac 340 cipher.

The result seems too good to be true compared with Figure 9. In Figure 9, for

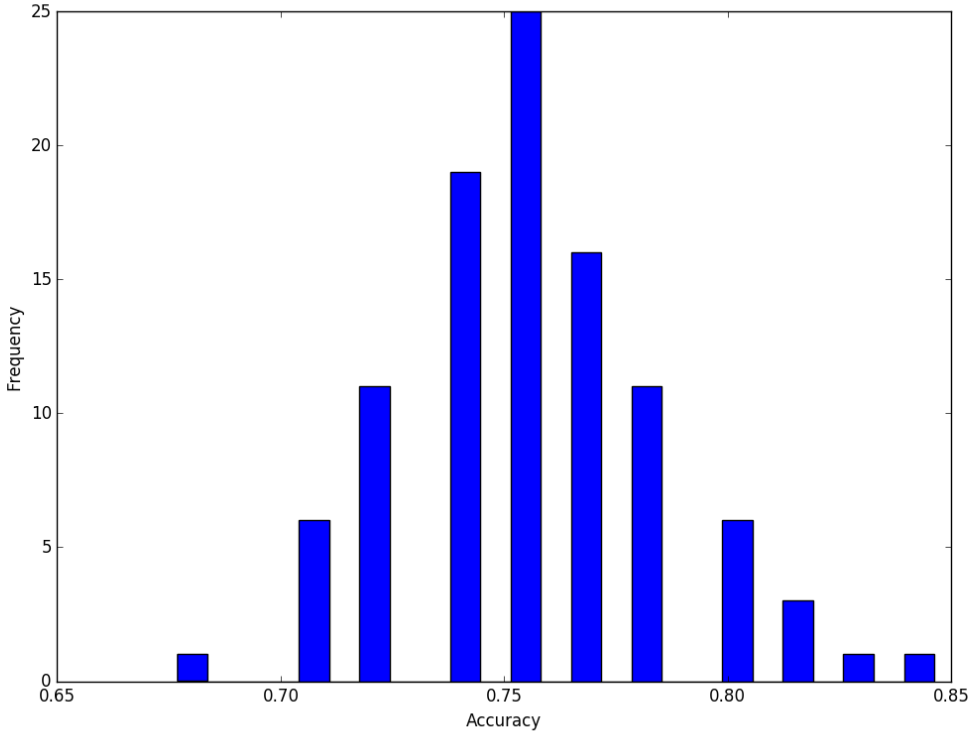

Figure 11: Histogram of the accuracies with fixed A from Zodiac 408

a ciphertext with length 400 and number of distinct symbols 65, the success rate is only about 0.43 when the number of restarts is 200,000. The fake Zodiac 340 here contains fewer characters, but we get much more better accuracies within fewer random restarts which contradict with our intuition and results in Figure 9. The reason comes from the different choices of the A matrix: we construct the A matrix from the Zodiac 408 cipher and the fake Zodiac 340 is almost the same as Zodiac 408, which makes the problem "overfit". Therefore, when decrypting Zodiac 340 without a close estimation of the A matrix for Zodiac 340, the success rate would go down and 10,000 number of restarts may not be enough. On other hand, this experiment shows choosing a close A will greatly increase the success rate.

4.3.2 Decrypt the Fake Zodiac 340 Using Fixed A from English Corpus

We repeat the experiment on the fake Zodiac 340 by setting A to be the one generated from English corpus. With 1,000,000 random restarts, the accuracy is only 0.476923, which is far below 0.8. From this result, we can infer that the real A is far away from that generated from the English corpus.

4.4 Decrypt the Fake Zodiac 340 Using HMM by Reestimating A

We generate an initial A from English Corpus and then do experiments with A reestimated during the training process. The motivation here is to check whether reestimating the general A matrix can give better results than the fixed one or even can successfully decrypt the fake Zodiac 340. Once decrypted, it is also very interesting to check whether the general A converges to the A from the Zodiac 408. However, the accuracy turns out to be only 0.369231 in 1,000,000 number of random restarts, even lower than that using the fixed A from English Corpus. Due to the time issue, we do not do experiment with more random restarts.

We could also set the initial A be the one from the Zodiac 408 and then reestimate A. But such experiment does not make sense because the initial A is already very close to the original one and it is unnecessary to reestimate it.

4.5 Compare the Performances of Reestimating A or not in Decrypting a Simple Substitution Cipher

In Section 4.3.2 and Section 4.4, we find that reestimating A may not be a good approach when training HMM. To verify this statement, in this Section, we do similar experiments but on a relative simpler case: Caesar cipher, a classic simple substitution cipher. The test case we use contains 10,000 characters and (N, M) are set to (26,

26) when training the HMM. The initial A matrix is from the English corpus and

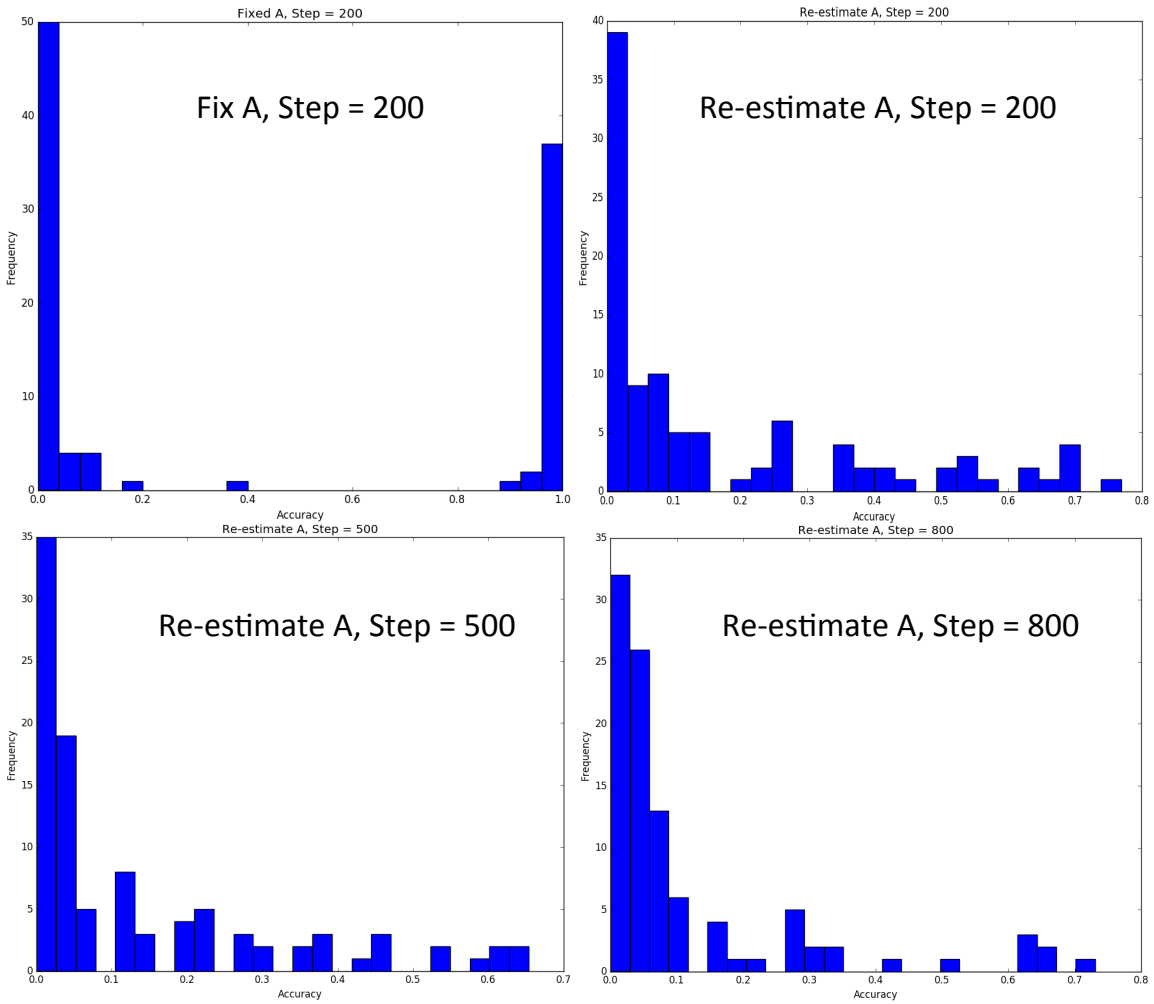

Figure 12: Histograms of the accuracies with A re-estimated or not and with different steps, T=10,000

the experiment results are summarized in four histograms of the 100 accuracies as in Figure 12. The four cases are those with A to be re-estimated and not and the number of steps equal to 200, 500 and 800. From Figure 12, we find that the Caesar cipher can be decrypted in about 3 random restarts with fixed A. However, it probably requires 100 restarts with re-estimated A and increasing the number of steps for each training process does not help. In this simple task, reestimating A increases the computational

complexity dramatically. We can suspect that for more complex cipher, reestimating A probably would fail the task.

Another possible reason for the failing in decrypting the Caesar cipher may come from the length. The test case we use has 10,000 characters so the digram frequency is probably very close to the A from English corpus. If this was the case, reestimating A would probably make A far away from the real digram frequency. Reestimating A may bring good effect when the initial digram frequency is far away from the human initialized A and reestimating A can make the initial A closer and closer to the real digram frequency. Therefore, we repeat the same experiments on shorter ciphertexts, because the digram frequency of shorter text may deviate from the one from English corpus. The experiment results are summarized in Figure 13, which shows that using a fixed A still performs better when the ciphertext length is only 300 or 500.

In summary, in Section 4.3 and Section 4.4, we find that the fake Zodiac 340 can be decrypted within 10,000 restarts with fixed A from Zodiac 408. However, using initial A from classic English corpus and reestimating A during the training process can not decrypt the fake Zodiac 340 within 1,000,000 restarts. We believe that reestimating A is not a good choice. As a result, when applying HMM with random restarts to decrypt real Zodiac 340, the important part is to try different possible A matrices and keep A fixed during the training process.

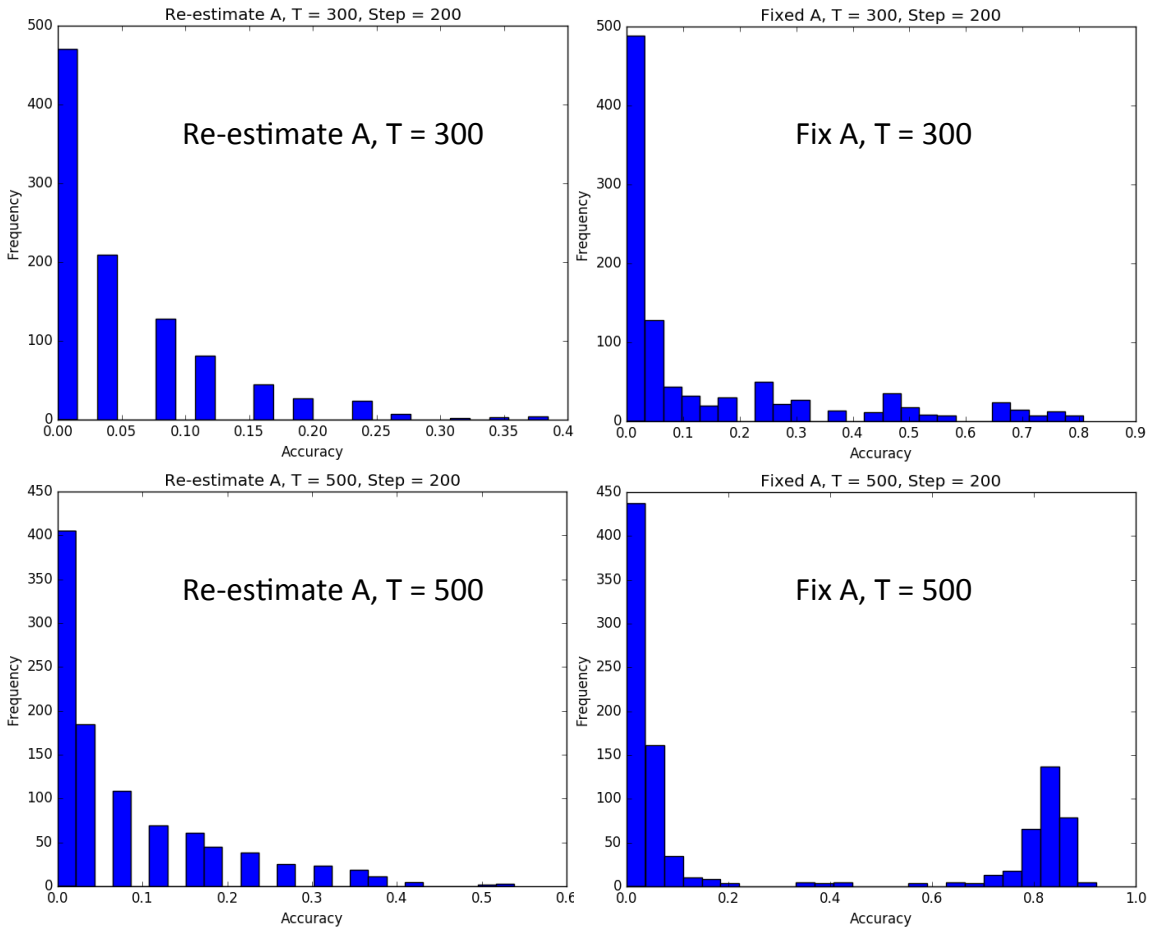

Figure 13: Histograms of the accuracies with A re-estimated or not and with different lengths, step=200

CHAPTER 5

Attempt to Decrypt Zodiac 340

There are three unsolved Zodiac ciphers and two of them are too short to uncover. Zodiac 340 is the remaining one that is relatively longer and has large possibility to be decypted. The cipher is shown as Figure 14. It contains 63 distinct symbols.

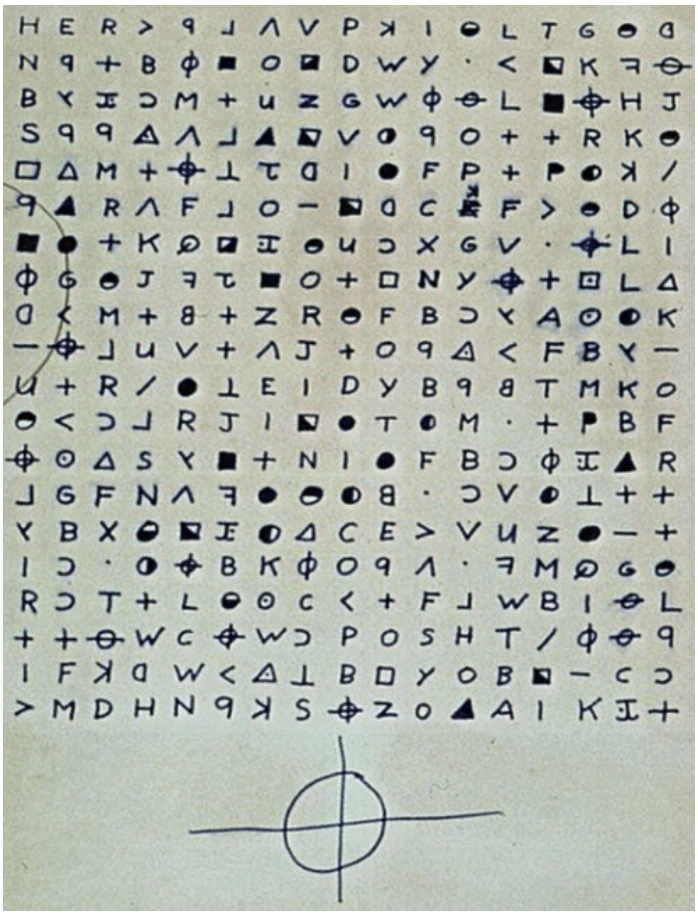

Figure 14: Zodiac 340 cipher [13]

Paper [1] shows that Zodiac 340 is not a pure homophonic substitution cipher, but a combination with another encryption method. It is also possible that the Zodiac 340 is not a cipher. It could be a random collection of the symbols Zodiac usually used.

In this Chapter, we attempt to decrypt Zodiac 340 by HMM with random restarts, assuming Zodiac 340 is the combination of a homophonic substitution cipher and a certain column transposition. In Section 5.1, we set up the score metric for evaluating training result of HMM. In section 5.2, we propose a model that combines Jakobsen's Algorithm [6] with HMM with random restarts targeting on this kind of combination cipher.

5.1 Choose the Score Metric

In previous Chapters, all the experiments we do are simulations where we know the plaintext. We compute the accuracy directly and use the accuracy as the metric to evaluate the training result of HMM. But here we do not know the plaintext or key, and thus we have to choose a different metric other than accuracy. There are at least two candidates: one is the HMM score $\log \mathbb{P}(\mathcal{O}|\lambda)$ as defined in Section 2.4.2 and the other one is the so called digraph score computed as follows:

1. Get putative key map K from the final B matrix.
2. Get the putative plain text by applying the putative key maps on the ciphertext.
3. Get the putative transition matrix D from the digram frequencies in the putative plaintext.
4. With the real transition matrix is E, i.e. A from Zodiac 408, we calculate the digraph score for the putative key K using equation (1).

In practice, we should choose the score metric that strongly correlates with accuracy. For this purpose, in this Section, we do experiments on the correlation between each score metric and accuracy by applying HMM with random restarts on the fake

Zodiac 340 with fixed A from the Zodiac 408. We then compute the sample correlation between the HMM score and accuracy and the sample correlation between the digraph score and the accuracy.

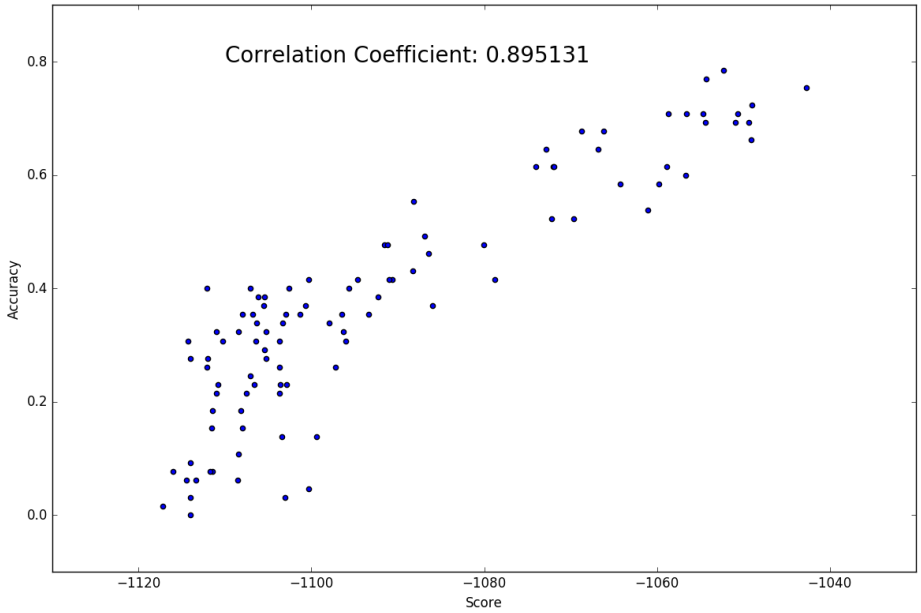

Figure 15: Scatter plot between accuracy and the HMM score

Figure 15 is the scatter plot of 100 pairs of (HMM score, accuracy). Each HMM score is the highest one among 100 random restarts and each accuracy is the one correspondes to the score. The sample correlation coefficient between the HMM score and the accuracy turns out be 0.895. Figure 16 is the scatter plot of 100 pairs of (digram score, accuracy), where the sample correlation coefficient is -0.682. The HMM score gives a better correlation to the accuracy compared to the digraph score. As a result, we will choose the HMM score as the score metric when applying HMM with random restarts to decrypt Zodiac 340.

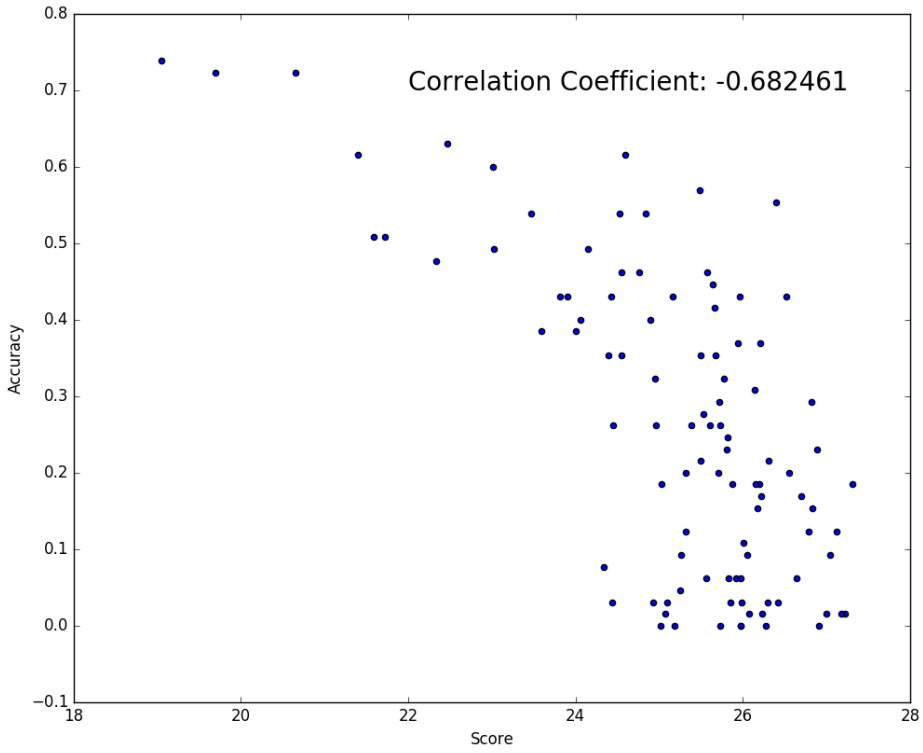

Figure 16: Scatter plot between accuracy and the Digram score

5.2 Use Combined Model to Decrypt Zodiac 340

There is a high probability that the Zodiac 340 is not a pure homophonic substitution cipher [1]. Here we propose the conjecture that the Zodiac 340 is a combination of a homophonic substitution cipher and a certain column transposition. Although we can exhaustively try all the permutations of columns and then apply an algorithm that decrypts homophonic substitution ciphers, the time complexity is huge: $17! \approx 2^{48}$. Enlightened by the Jakobsen's Algorithm, we propose a combined model to decrypt such a cipher.

5.2.1 Combined Model

There are 17 columns in the original ciphertext. We denote the ciphertext as 17 columns of symbols: $C = c_1, c_2, \ldots, c_{17}$. Let '|' represent the swap operation between

two columns. We swap the columns in the following order:

$$
\begin{array}{lllllll}
\text{round 1:} & c_1|c_2 & c_2|c_3 & c_3|c_4 & \dots & c_{14}|c_{15} & c_{15}|c_{16} \quad c_{16}|c_{17} \\
\text{round 2:} & c_1|c_3 & c_2|c_4 & c_3|c_5 & \dots & c_{14}|c_{16} & c_{15}|c_{17} \\
\text{round 3:} & c_1|c_4 & c_2|c_5 & c_3|c_6 & \dots & c_{14}|c_{17} & \\
\vdots & & \vdots & & \iddots & & \\
\text{round 14:} & c_1|c_{15} & c_2|c_{16} & c_3|c_{17} & & & \\
\text{round 15:} & c_1|c_{16} & c_2|c_{17} & & & & \\
\text{round 16:} & c_1|c_{17} & & & & &
\end{array} \tag{17}
$$

After each swap, we apply HMM with random restarts to the swapped ciphertext. We restart the swap from the beginning whenever the HMM score improves. The algorithm terminates when there is no improvement in the HMM score in round 16.

5.2.2 Result of Using Fixed A from Zodiac 408

We implement the model that combines HMM with random restarts with systematically swapping of columns of the Zodiac 340 cipher. As shown in Section 4.3.1, the fake Zodiac 340 can be decrypted within 10,000 random restarts using fixed A from Zodiac 408. If Zodiac 340 is similar to Zodiac 408, it should be also decrypted within 10,000 random restarts using the same A. So we choose the number of random restarts in the HMM to be 10,000.

The best HMM score is -1088.816037 and the corresponding putative plaintext is as follows. Unfortunately, there are no meaningful sequences. It only contains a few words like "the", "men".

```
jqkcdwilrbbvnvemo dtfangivexklsykda
ncofstyhexnangeca ddgwiwxylldittqkm
ustheryobyorthebe xqidowilyorbokmen
ytkgbvemyouellenb emandygithakethnu
```

```
sstorthqmofongpek ewylltiatidgsofnl
tqeyyrjbekfdrvski sowmqabyyveslthfo
puwengtabyofonexq eoawidymerlolertt
fuvnyceurjklyhylt ollbefknidildsbem
ovtqnvprstowxfban taxtrexoriwcvenad
obobxsgrfhkifylro seckadbwehixgbket
```

5.2.3 Result of Reestimating A

We initialize the A matrix from the Zodiac 408 and keep reestimating A during the training process of HMM. The number of random restarts is still set to be 10,000. The best HMM score is -899.765652 which is higher than the HMM score with fixed A as shown in the previous subsection. The putative plaintext is

```
jqyckuwikkygbvfrb tixloplsuuxreaolt
hdtgdldofustbppym ciwiaugwkwilllcor
legobmxpyosklenku wgicaulsabnksjrus
ooplrovfdhvfkrpby mfsrlxpllgxxplebe
lebocldorsthdveno dpaumlwkllicestda
uldcymqouxtiovool uerhamycovnorlets
wepexpldyosthsvgc xfusrlownorhknmll
gtdvegnanqjkddoal whyrstopliwrloffr
lhcvngebelsuutytb ullthpunklwyvusti
bsykmecutgxltaanh yojuwikxpdlgvyovl
```

This putative plain text is still not a meaningful message. But now we can find some words there such as "will", "us", "why" and "stop".

5.3 Discussion

Although our algorithm, which is designed to decrypt a homophonic substitution cipher combined with a certain column transposition, fails to decrypt the Zodiac 340 as shown in Section 5.2.2 and Section 5.2.3, it does not reject the possibility that the Zodiac 340 is a homophonic substitution cipher combined with a certain column transposition. Instead, our combined model may not have high success rate. In this section, we focus on checking the effectiveness of our algorithm.

5.3.1 Check the Effectiveness of the Combined Model

To check the effectiveness of our combined model, we create a ciphertext by randomly permuting the 17 columns of fake Zodiac 340. We then apply the combined model to decrypt this new cipher, where the fixed A from Zodiac 408 is used in the HMM process. The permutation of columns in the test case is [12, 10, 3, 7, 1, 13, 4, 2, 11, 5, 15, 9, 16, 8, 14, 0, 6] and the accuracy obtained from our combined model is only 0.03. In other words, the combined model also fails in this synthetic case of a homophonic substitution cipher combined with a certain column transposition. As a result, Zodiac 340 could still be a homophonic substitution cipher combined with a certain column transposition and we have to improve our combine model or propose a new algorithm.

5.3.2 Check the Effectiveness of Swapping Columns in Recovering the Permutation Based on the Digram Score

To check whether the columns swap part in our combined model works or not, we decrypt a version of the fake Zodiac 340 that is just a column transposition, without any simple or homophonic substitution.

The swapping procedure still follows Eq.(17). We generate test cases by randomly permuting the 17 columns of the plaintext of the fake Zodiac 340. The digram score is defined as

$$\text{Score} = \sum_{i,j} |P_{i,j} - O_{i,j}|, \quad i, j = 0, 1, \ldots, 25, \tag{18}$$

where $O_{i,j}$ is the digram frequency of the plaintext of the fake Zodiac 340 and $P_{i,j}$ is the digram frequency of the putative plaintext. Finally, to measure the performance of the swapping procedure, we compute the matching rate by comparing the characters in the putative (output) plaintext to those in the plaintext. In the comparison, we

match the putative (output) plaintext as well as all its circles with the plaintext. We take the highest matching rate as the accuracy.

The test result is shown in Figure 17. Each histogram contains 120 accuracies of decrypting 120 test cases generated from randomly permuting the first $k \geq 5$ columns of the plaintext. The result is not good. As shown in the lower right subplot (randomly generated from permuting the 17 columns), none of the 120 test cases can be recovered if we set accuracy 0.8 as the recover success threshold.

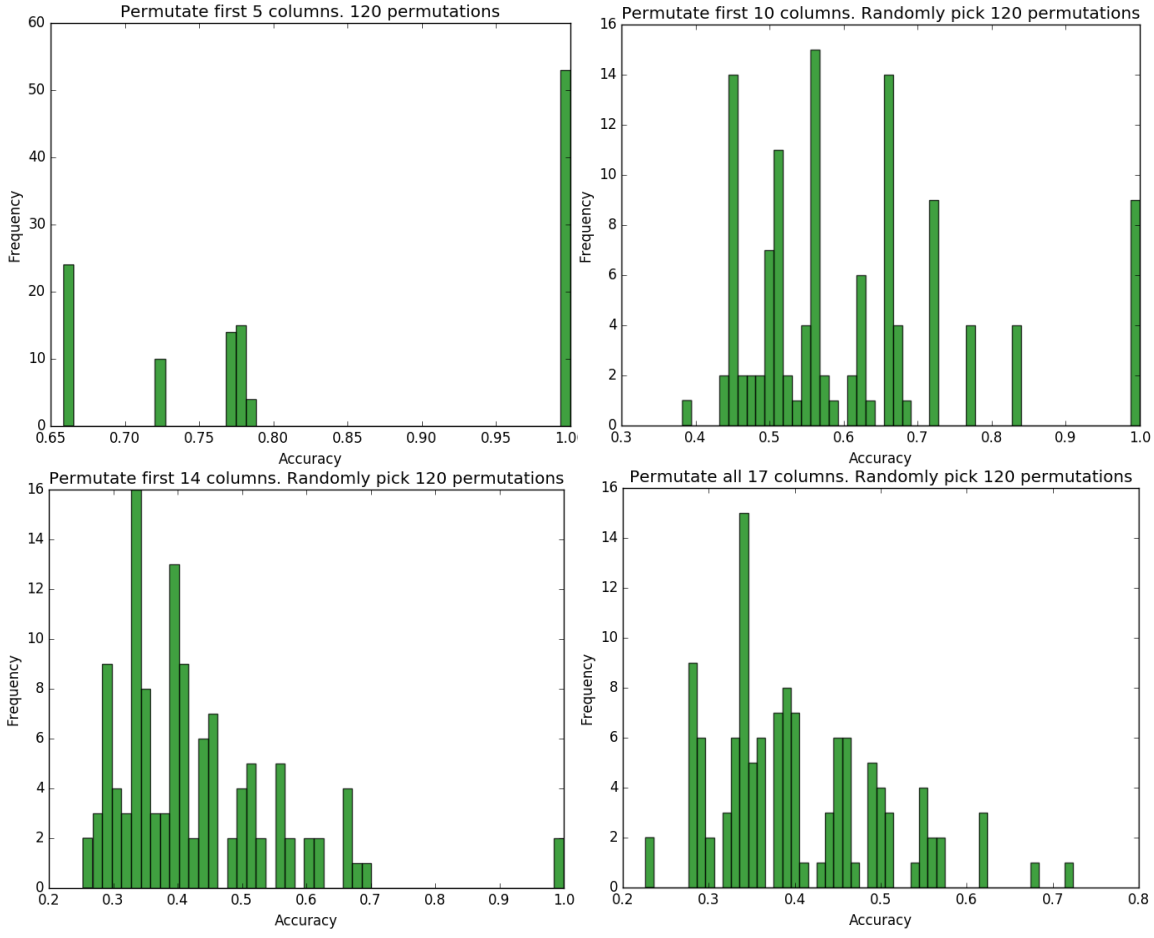

Figure 17: Test swapping columns.

5.3.3 Check the Effectiveness of Swapping Columns in Recovering the Permutation Based on the Score from Words

We modify the score metric for the columns swap so that it favors longer words. The score is computed base on the words that appears in the putative plaintext. If the sequence contains a word with length n, 2^n will be added to the score.

The test case is also the plaintext of the fake Zodiac 340 without without any simple or homophonic substitution. The dictionary is generated by all the words in the plaintext. We run the swapping procedure as in the last subsection but with the word score. The new result is shown in Figure 18. From the lower right subplot, we now find 32 out of 120 test cases can be recovered. Although this approach still can not recover all the permutations, using the word score here is much better than the digram score in the last subsection.

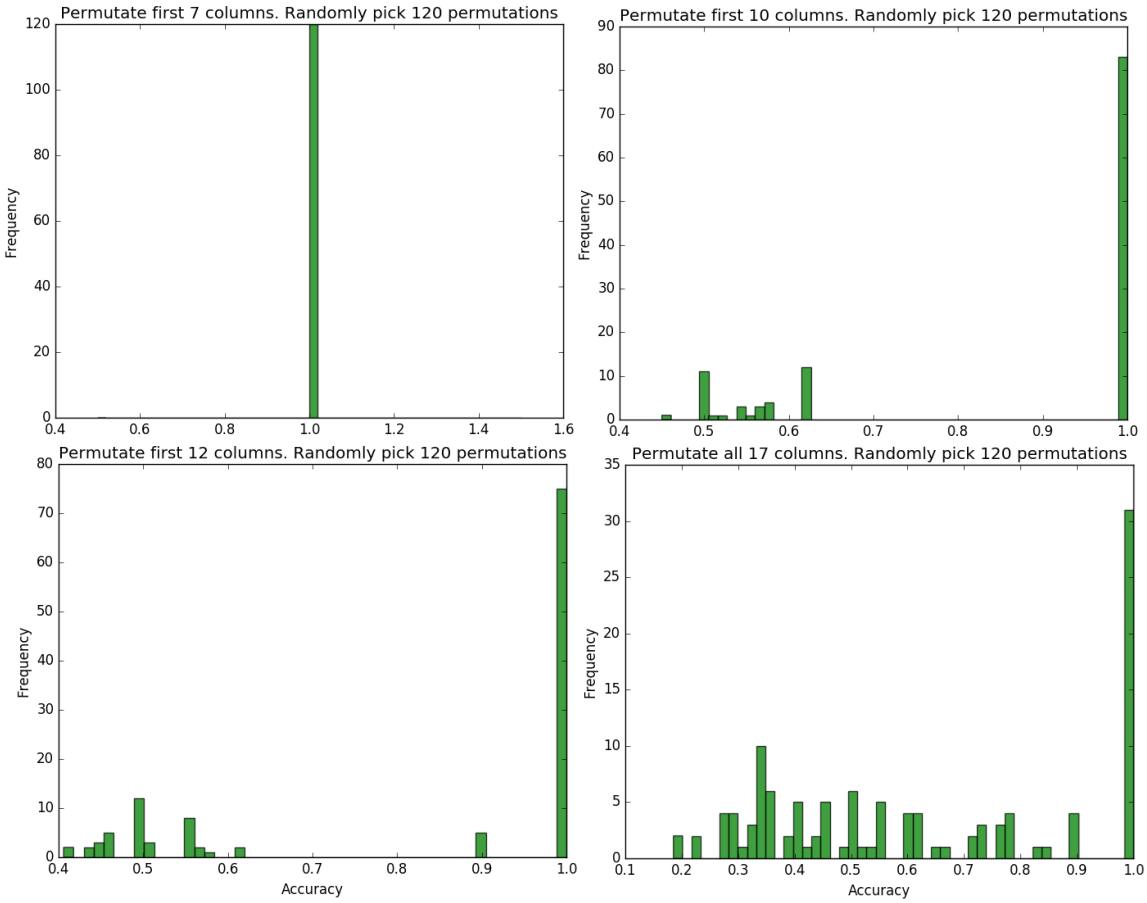

Figure 18: Test swapping columns based on the score from words.

CHAPTER 6

Conclusion and Future Work

6.1 Conclusion

In this project, we study the HMM with random restarts in decrypting homophonic substitution cipher. Our work can be summarized into three part.

The first one is to compare the performances of HMM with random restarts to the nested hill climb algorithm in decrypting the homophonic substitution cipher. We generate 432 cipher texts with lengths from 300 to 10,000 and number of distinct cipher symbols from 28 to 90, and then run the two algorithms on these test cases. The challenge here is to reduce the running time of HMM algorithm since the number of random restarts can be as large as 200,000. To speed up, we implement HMM in pure C code, vectorize subroutines using cblas/lapack. In the algorithm level, we reduce the number of training steps while making the results converge and initialize the B matrix with large variance to increase the success rate within the fixed number of restarts. We conclude that the HMM with random restarts performs better than the nested hill climb algorithm.

The second part of the project is to apply HMM with random restarts to decrypt the fake Zodiac 340 cipher. We demonstrate that HMM with random restarts can successfully decrypt the fake Zodiac 340 cipher. When applying HMM, we find that it is better to initialize the A matrix from Zodiac 408 and keep it fixed than to keep reestimating it during the training process.

In the final part of the project, we propose a combined model consisting of column swaps and HMM with random restarts to decrypt the famous Zodiac 340,

assuming that Zodiac 340 is the combination of a homophonic substitution cipher and a column transposition. We test our combine model on the fake Zodiac 340 cipher with random column permutation. We find that there exist permutations that our swapping procedure cannot recover. Finally, we design word score, a new score metric other than HMM score or digram score, which probably can improve the performance of our combined model.

6.2 Future Work

Although we show in Chapter 3 that HMM with random restarts performs better than the nested hill climb algorithm in decrypting the homophonic substitution cipher, the accuracy of HMM with 200,000 random restarts is less than 0.5 for the ciphertext with length 300-400 and 65 number of symbols. It is remains unclear whether HMM can decrypt such homophonic substitution cipher or not with even more random restarts. If not, HMM or our combine model cannot decrypt Zodiac 340 even it is the combination of a homophonic substitution cipher and a column transposition.

We also show that the performance HMM with random restarts is largely influenced by the A matrix. A further research is required on how to treat the A matrix effectively including whether to keep A matrix fixed or reestimate A during the training process or how to choose an appropriate A matrix if A is fixed.

Finally, to decrypt a cipher that is a combination of the homophonic substitution cipher and a column transposition, we hope to improve our algorithm to recover the most random permutations of the columns of a plaintext. We even hope to decrypt Zodiac 340 with the improved algorithm.

LIST OF REFERENCES

[1] T. Berg-Kirkpatrick and D. Klein, Decipherment with a million random restarts, *EMNLP*, pp. 874-878, Oct. 2013, http://www.cs.berkeley.edu/~tberg/papers/emnlp2013.pdf

[2] G. Claston, 340-Cipher—Overview and Examination, http://www.zodiackiller.com/mba/zc/69.html

[3] R. L. Cave and L. P. Neuwirth, Hidden Markov models for English, in J. D. Ferguson, editor, *Hidden Markov Models for Speech*, IDA-CRD, pp. 8-15, Princeton, NJ, Oct. 1980.

[4] A. Dhavare, R. M. Low, and M. Stamp, Efficient cryptanalysis of homophonic substitution ciphers, *Cryptologia*, 37(3), pp. 250-281, 2013, http://dx.doi.org/10.1080/01611194.2013.797041

[5] W. N. Francis and H. Kucera. Brown corpus manual, Brown University (1979).

[6] T. Jakobsen, A fast method for cryptanalysis of substitution ciphers, *Cryptologia* (19)3, pp. 265-274, 1995, http://www.tandfonline.com/doi/abs/10.1080/0161-119591883944

[7] M.A. Magnuson, An improved implementation of paper "Efficient Cryptanalysis of Homophonic Substitution Ciphers", May 2016, https://github.com/alimony/homophonic-cipher-attack

[8] "MysteryTwisterC3: The Crypto Challenge Contest", https://www.mysterytwisterc3.org/en/challenges/level-i/zodiac-cipher

[9] M. Stamp, A revealing introduction to hidden Markov models, Department of Computer Science, San Jose State University, 2004, http://www.cs.sjsu.edu/faculty/stamp/RUA/HMM.pdf

[10] M. Stamp, Information Security: Principles and Practice, 2nd edition, Wiley, 2011.

[11] R. Vobbilisetty, FD. Troia, RM. Low, CA. Visaggio, and M. Stamp, Classic cryptanalysis using hidden Markov models, *Cryptologia*, pp. 1-28, 2016, http://dx.doi.org/10.1080/01611194.2015.1126660

[12] Zodiac 408, http://www.zodiacciphers.com/complete-408-cipher.html

[13] Zodiac 340, http://www.zodiacciphers.com/340-cipher.html

APPENDIX

More Results of HMM with Random Restarts in Decrypting the Homophonic Substitution Cipher

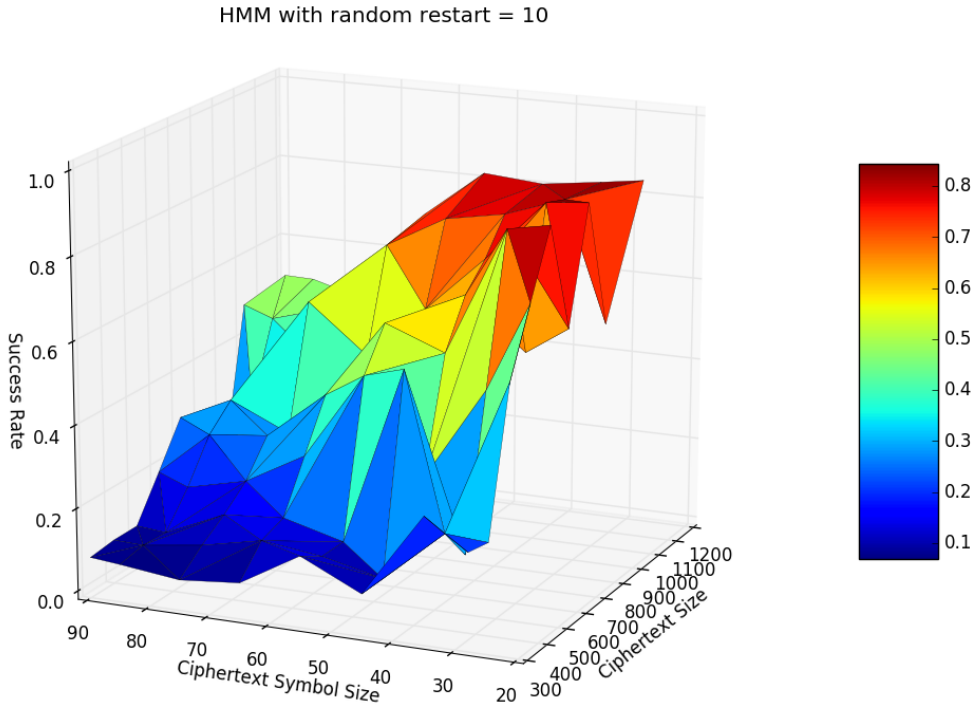

Figure A.19: Restart = 10

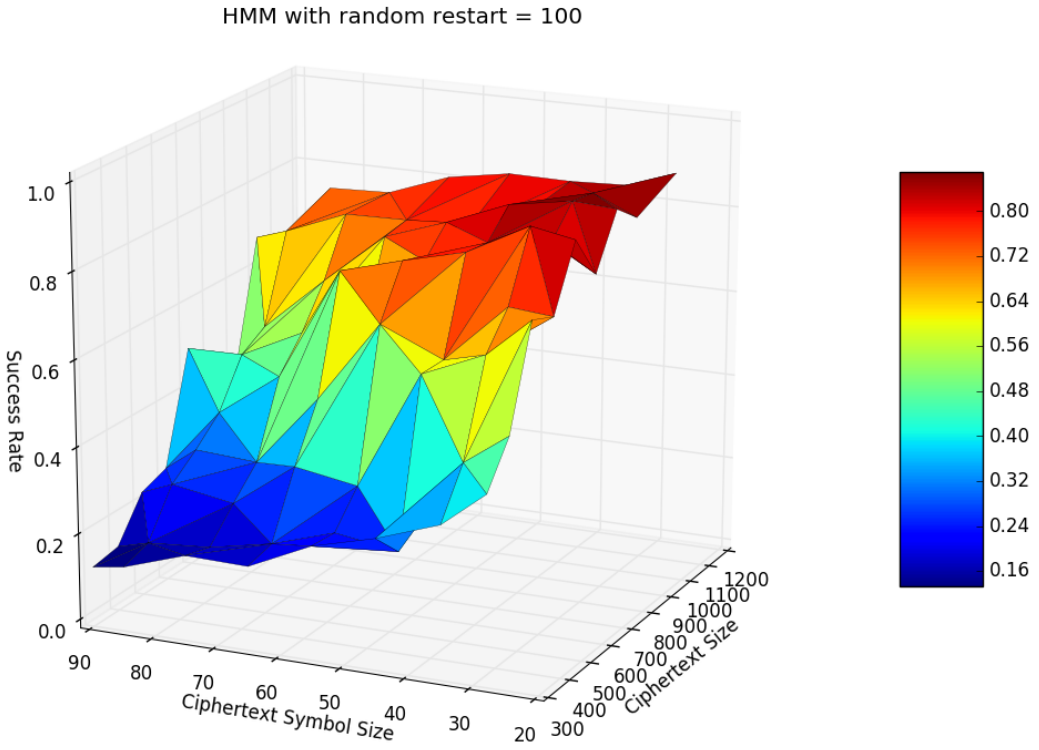

Figure A.20: Restart = 100

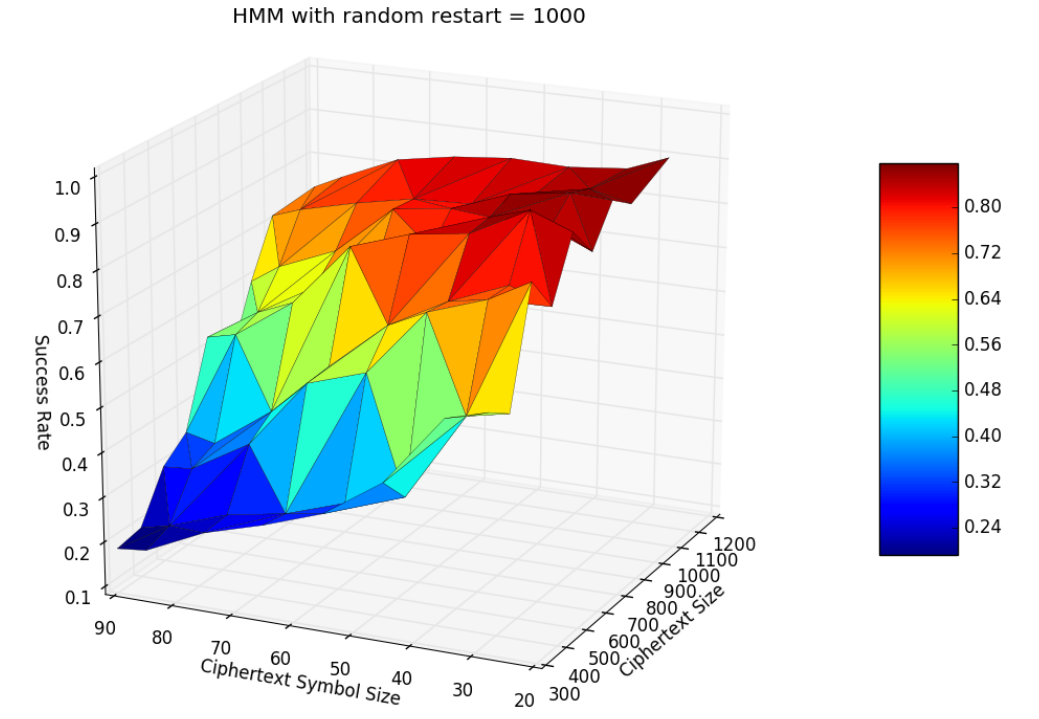

Figure A.21: Restart = 1000

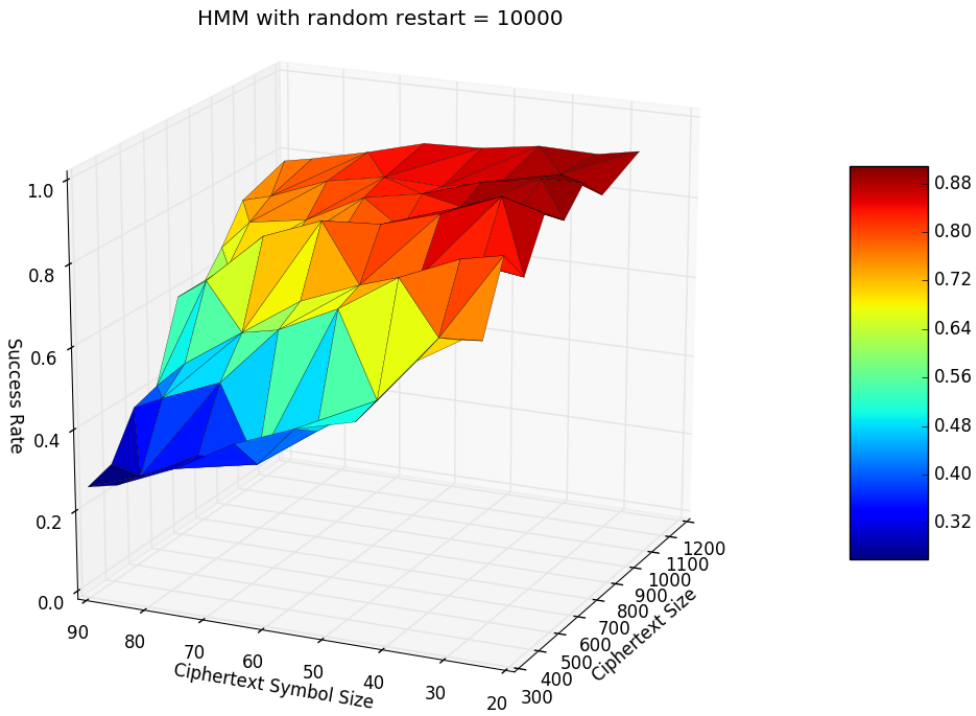

Figure A.22: Restart = 10,000

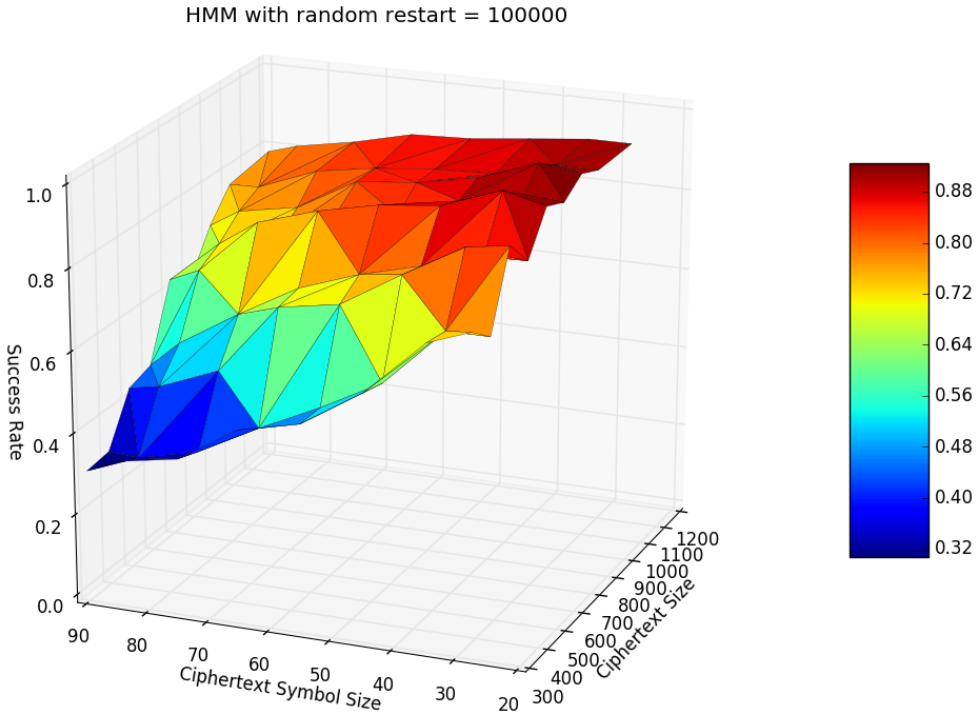

Figure A.23: Restart = 100,000

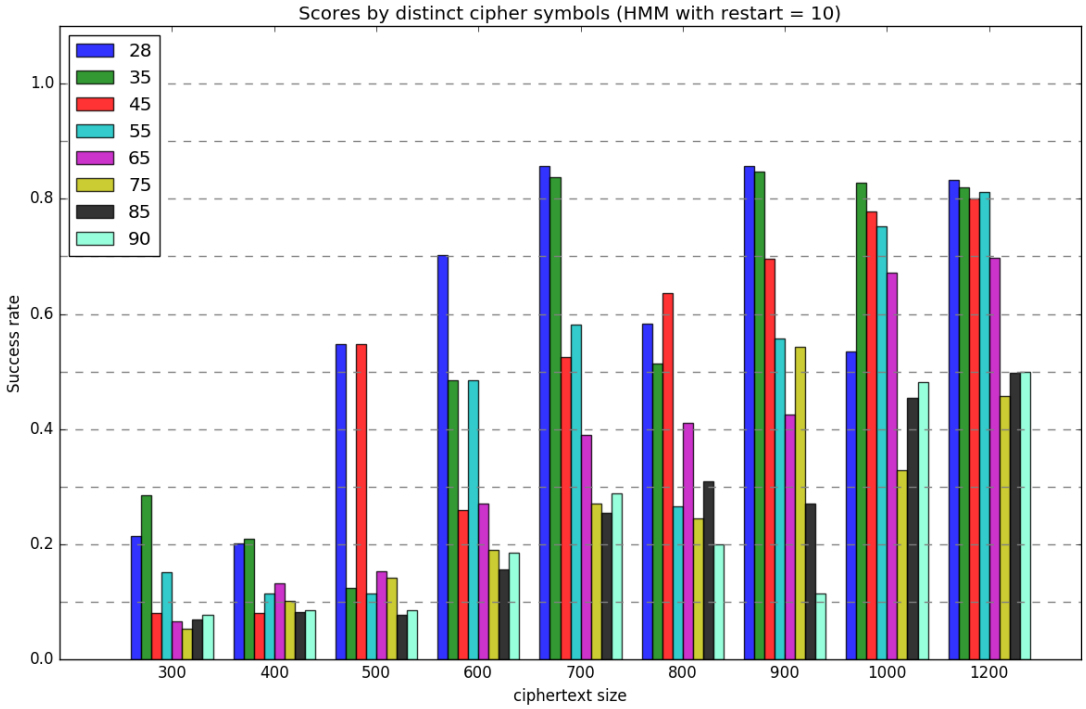

Figure A.24: Restart = 10

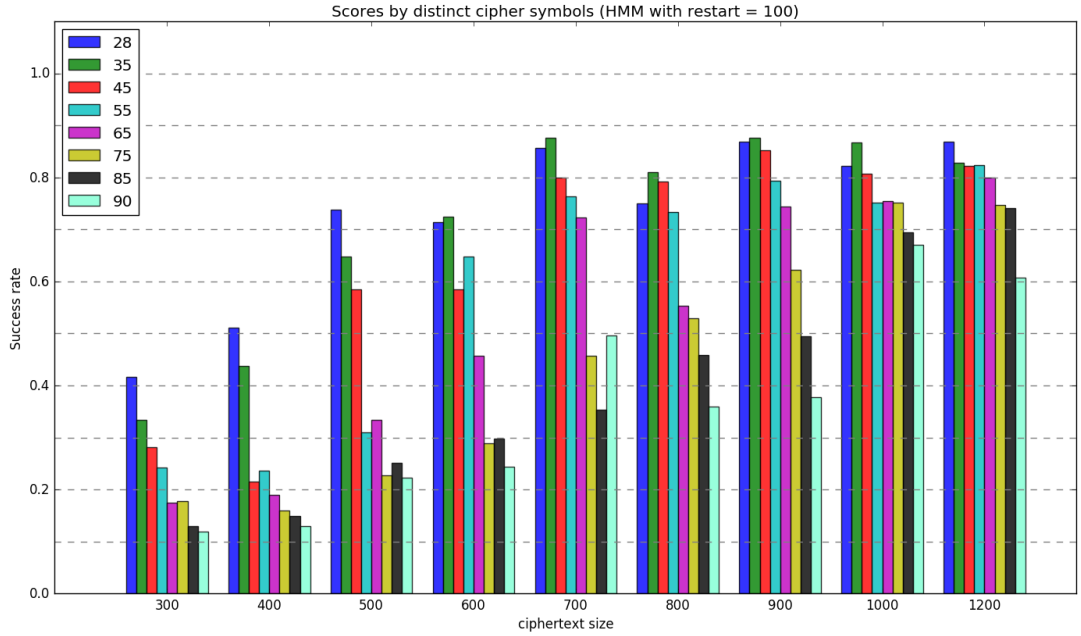

Figure A.25: Restart = 100

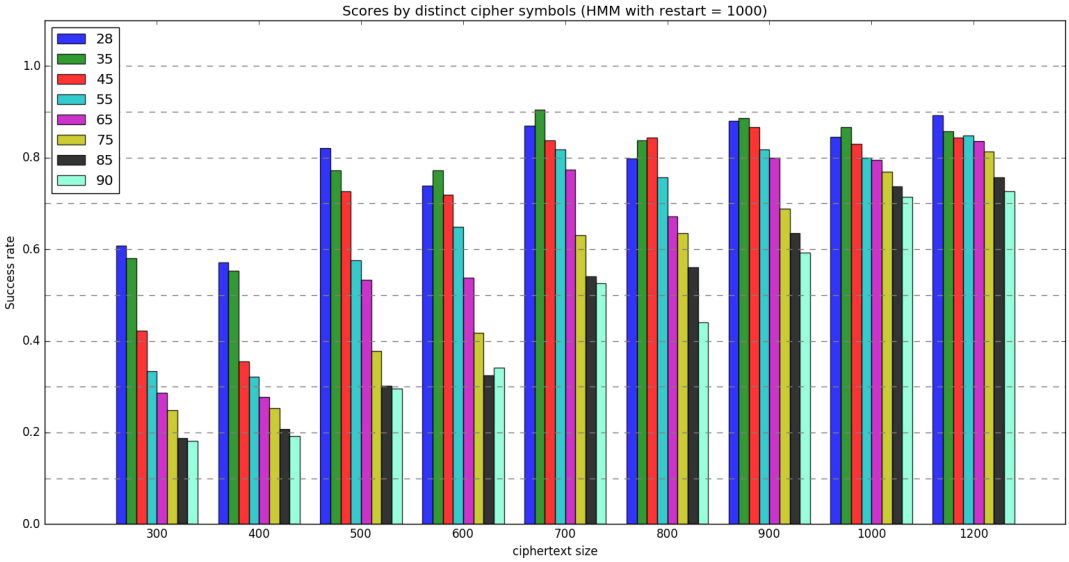

Figure A.26: Restart = 1000

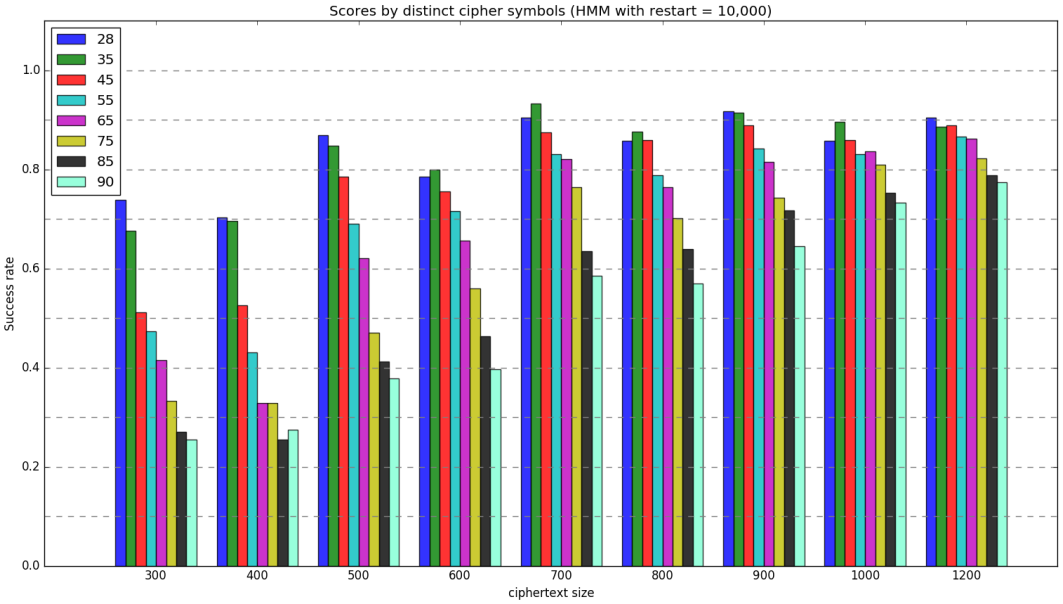

Figure A.27: Restart = 10,000

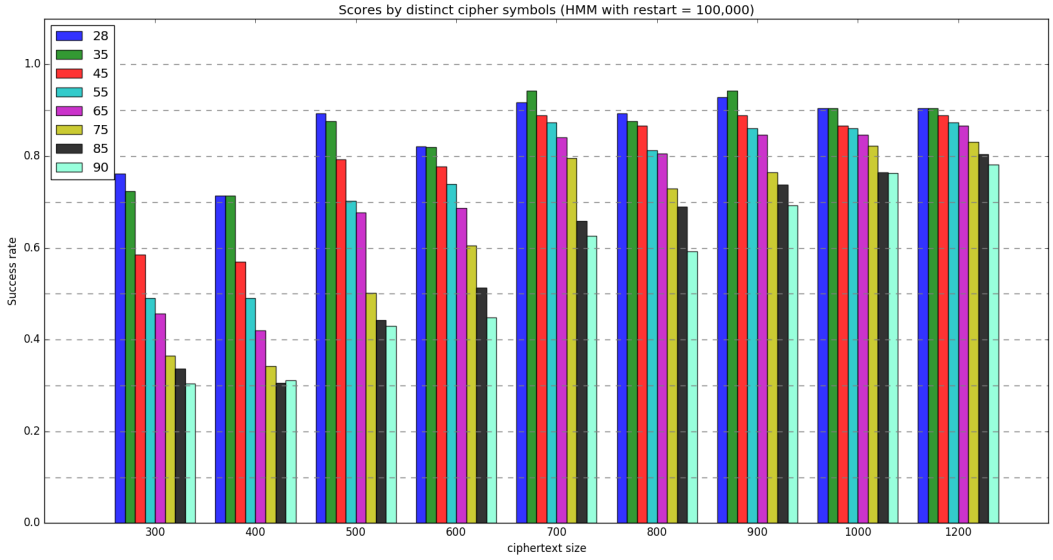

Figure A.28: Restart = 100,000