Answer to 6.437 Project Part I

Mucong Ding, mcding@mit.edu, Stu ID: 920494196

May 1, 2018

1 Problem 1

1.1 (a)

Since we assume English text can be modeled as a Markov chain, the probability of a plaintext $\mathbf{x}$ is:

$$\begin{aligned} p_{\mathbf{x}}(\mathbf{x}) &= \mathbb{P}(x_1)\mathbb{P}(x_2|x_1)\cdots\mathbb{P}(x_n|x_{n-1}) \\ &= P_{x_1}\Pi_{i=2}^{n-1}M_{x_{i+1},x_i} \end{aligned} \tag{1}$$

Thus the probability of corresponding ciphertext $\mathbf{y}$ encrypted by cipher function f is:

$$p_{\mathbf{y}|f}(\mathbf{y}|f) = P_{f^{-1}(y_1)}\Pi_{i=2}^{n-1}M_{f^{-1}(y_{i+1}),f^{-1}(y_i)} \tag{2}$$

And this is the likelihood of the observed ciphertext y under ciphering function f.

1.2 (b)

By Bayes' rule, the posterior distribution is simply:

$$p_{f|\mathbf{y}}(f|\mathbf{y}) = \frac{p_{\mathbf{y}|f}(\mathbf{y}|f)p_f(f)}{\Sigma_f p_{\mathbf{y}|f}(\mathbf{y}|f)p_f(f)} \tag{3}$$

Since we adopt a uniform prior:

$$p_f(f) = \frac{1}{|\mathcal{F}|} \tag{4}$$

where $\mathcal{F}$ is the set of all possible cyphering functions (permutations on $\mathcal{A}$, $\mathcal{F} = \text{Perm}(\mathcal{A})$). We can simplify the formula of $p_{f|\mathbf{y}}(f|\mathbf{y})$ to:

$$p_{f|\mathbf{y}}(f|\mathbf{y}) = \frac{p_{\mathbf{y}|f}(\mathbf{y}|f)}{\Sigma_f p_{\mathbf{y}|f}(\mathbf{y}|f)} \tag{5}$$

And the MAP estimator $\hat{f}_{\text{MAP}}(\mathbf{y})$ is:

$$\hat{f}_{\text{MAP}}(\mathbf{y}) = \arg\max_f p_{f|\mathbf{y}}(f|\mathbf{y}) \tag{6}$$

Again because we assume uniform prior, it could be simplify to:

$$\begin{aligned}\hat{f}_{\text{MAP}}(\mathbf{y}) &= \arg\max_{f} p_{\mathbf{y}|f}(\mathbf{y}|f) \\ &= \arg\max_{f} P_{f^{-1}(y_1)}\Pi_{i=2}^{n-1} M_{f^{-1}(y_{i+1}),f^{-1}(y_i)}\end{aligned} \tag{7}$$

It's the same as the ML estimator.

1.3 (c)

Direct evaluation of the MAP estimate $\hat{f}_{\text{MAP}}$ is computationally infeasible since the alphabet of ciphering functions, $\mathcal{F}$, is too large:

$$|\mathcal{F}| = |\mathcal{A}|! = 28! = 304888344611713860501504000000 \tag{8}$$

It's infeasible to iterate through this alphabet to find $\hat{f}_{\text{MAP}}$ which maximizes $p_{f|\mathbf{y}}(f|\mathbf{y})$.

2 Problem 2

2.1 (a)

Fist note that for each specific ciphering function f (permutation on $\mathcal{A}$), it has $\binom{|\mathcal{A}|}{2}$ "neighbors" which differ in exactly two symbol assignments. Thus, since there are $|\mathcal{F}| = |\mathcal{A}|!$ cipher functions, there are:

$$\frac{|\mathcal{A}|!}{2}\binom{|\mathcal{A}|}{2} \tag{9}$$

such "neighboring" pairs. And since there are $\binom{|\mathcal{A}|!}{2}$ cipher function pairs, and we assume they are uniformly distributed. The probability that a pair of ciphering functions f_1 and f_2 differ in exactly two symbol assignments is:

$$\begin{aligned}\mathbb{P}(\text{"pair differ exactly two assigments"}) &= \frac{\frac{|\mathcal{A}|!}{2}\binom{|\mathcal{A}|}{2}}{\binom{|\mathcal{A}|!}{2}} \\ = \frac{378}{28!-1} \approx 1.2397981 \times 10^{-27}\end{aligned} \tag{10}$$

2.2 (b)

Inspired by problem 2(a), we construct the proposal transition matrix as:

$$V(f'|f) = \begin{cases} \frac{1}{\binom{28}{2}} & \text{if } f' \in \text{Nei}(f) \\ 0 & \text{elsewise} \end{cases} \tag{11}$$

where $\text{Nei}(f)$ is the set of permutations which differs to f in exactly two symbol assignments.

Following the Metropolis-Hastings algorithm, the effective transition matrix is:

$$W(f'|f) = V(f'|f)a(f \to f') \quad \text{if } f' \in \text{Nei}(f) \tag{12}$$

where $a(f \to f') = \min\left\{1, \frac{p_{\mathbf{y}|f}(\mathbf{y}|f')}{p_{\mathbf{y}|f}(\mathbf{y}|f)}\right\}$ and the likelihood function $p_{\mathbf{y}|f}(\mathbf{y}|f)$ is understand as the "un-normalized" posterior.

By the normalization constraints, the full formula of the effective transition matrix W is:

$$W(f'|f) = \begin{cases} \frac{1}{378}\min\left\{1, \frac{p_{\mathbf{y}|f}(\mathbf{y}|f')}{p_{\mathbf{y}|f}(\mathbf{y}|f)}\right\} & \text{if } f' \in \text{Nei}(f) \\ 1 - \Sigma_{f'\in\text{Nei}(f)} W(f'|f) & \text{if } f' = f \\ 0 & \text{elsewise} \end{cases} \tag{13}$$

This defines a Markov chain whose stationary distribution is the posterior $p_{f|\mathbf{y}}(f|\mathbf{y})$.

2.3 (c)

Pseudo-code of Metropolis-Hastings algorithm for decoding the ciphertext:

```
Input: alphabet $\mathcal{A}$, ciphertext $\mathbf{y}$, maximum number of iterations $T_{\max}$, termination
       threshold $T_{\text{term}}$
Result: predicted cipher function $f_{\text{pred}}$
Initialization: randomly pick $f \in \text{Perm}(\mathcal{A})$;
$t = 0$, $t_{\text{fixed}} = 0$;
while $t_{fixed} < T_{term}$ and $t < T_{max}$ do
    Get $f'$ by randomly swap two mapped values in $f$;
    Calculate the likelihoods $p_{\mathbf{y}|f}(\mathbf{y}|f')$ and $p_{\mathbf{y}|f}(\mathbf{y}|f)$ by Eq. 2;
    if $p_{\mathbf{y}|f}(\mathbf{y}|f') > p_{\mathbf{y}|f}(\mathbf{y}|f)$ then
        f = f';
    end
    else
        Generate random number $x \in [0, 1]$;
        if $x < p_{\mathbf{y}|f}(\mathbf{y}|f')/p_{\mathbf{y}|f}(\mathbf{y}|f)$ then
            f = f';
        end
        else
            $t_{\text{fixed}} += 1$;
        end
    end
    $t += 1$;
end
$f_{\text{pred}} = f$;
```

Algorithm 1: MH decoding algorithm

The algorithm terminates when either the number of iterations exceed $T_{\max}$ or the current cipher function f is not updated for T_{term} iterations.

3 Problem 3

3.1 (a)

The log-likelihood of the accepted state versus iteration count is plotted as Fig. 1. This is the result of one single experiment on the full ciphertext. The x-axis range of the upper plot is $1 \leq \text{iter} \leq 3000$ and the range of lower plot is $1000 \leq \text{iter} \leq 3000$.

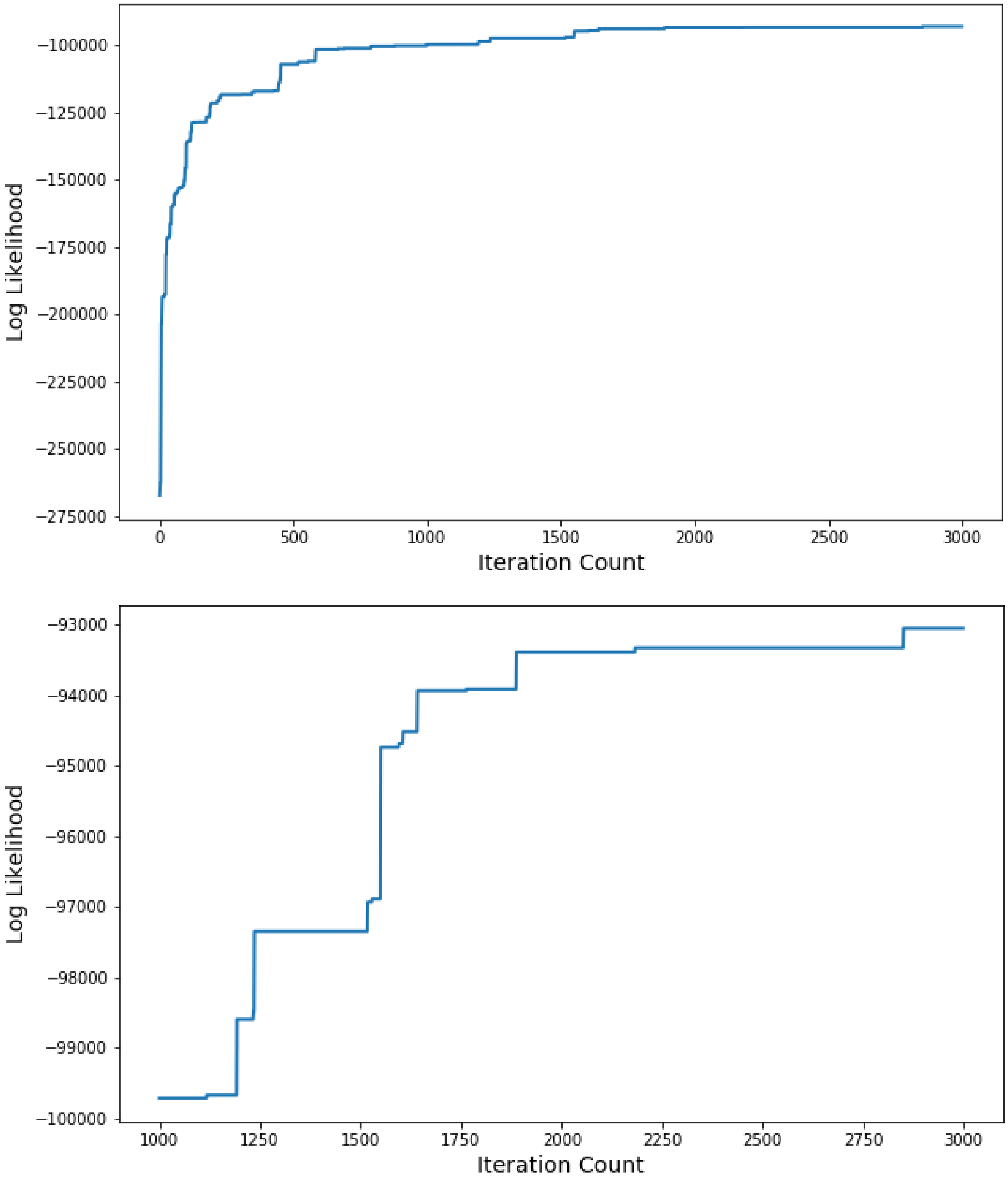

Figure 1: Log-likelihood of the accepted state versus iteration count.

We can see the log-likelihood of accepted state is monotonically increasing along with iteration count, and the increasing rate is much larger when the algorithm just starts.

3.2 (b)

The acceptance rate of state transitions versus iteration count is plotted as Fig. 2. Note that the acceptance rate is a sliding window average with window length $T = 100$. This is the result of one single experiment on the full ciphertext. The range of x-axis in Fig. 2 is $100 \leq \text{iter} \leq 3000$.

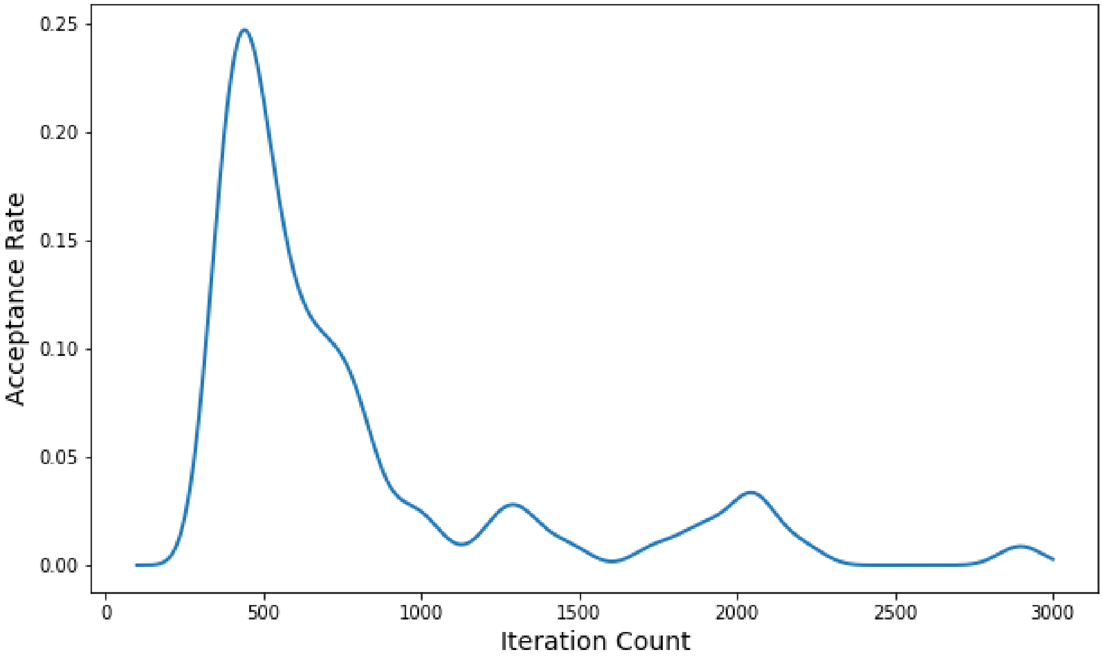

Figure 2: Acceptance rate of state transitions versus iteration count.

We can see the acceptance rate first increase rapidly and the drop rapidly. The curve of acceptance rate has a sharp peak at around iter ≈ 400.

3.3 (c)

The accuracy rate versus iteration count is plotted as Fig. 3. This is the result of one single experiment on the full ciphertext.

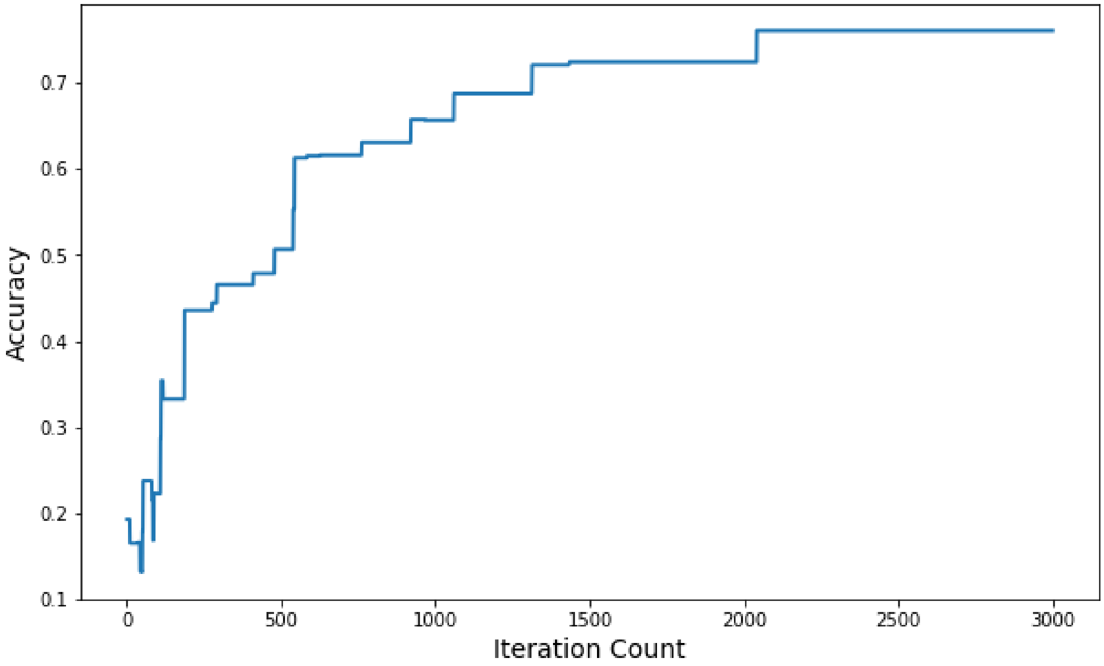

Figure 3: Accuracy rate versus iteration count.

We can see the accuracy increase monotonically along with the iteration count. After each increment, there is a plateau. Since it is likely that no updates of the predicted cipher function will take place (because of low acceptance rate) shortly after an update. The terminating accuracy is around 0.78, which is plausible.

3.4 (d)

I run the MCMC-based decoding algorithm on ciphertext with different lengths, (length $= 50, 100, 200, 400, 800, 1600, 3200$) independently. For each ciphertext, I run 20 independent trials and record the average accuracy of the 20 predictions. The average accuracies versus iteration count is plotted as Fig. 4.

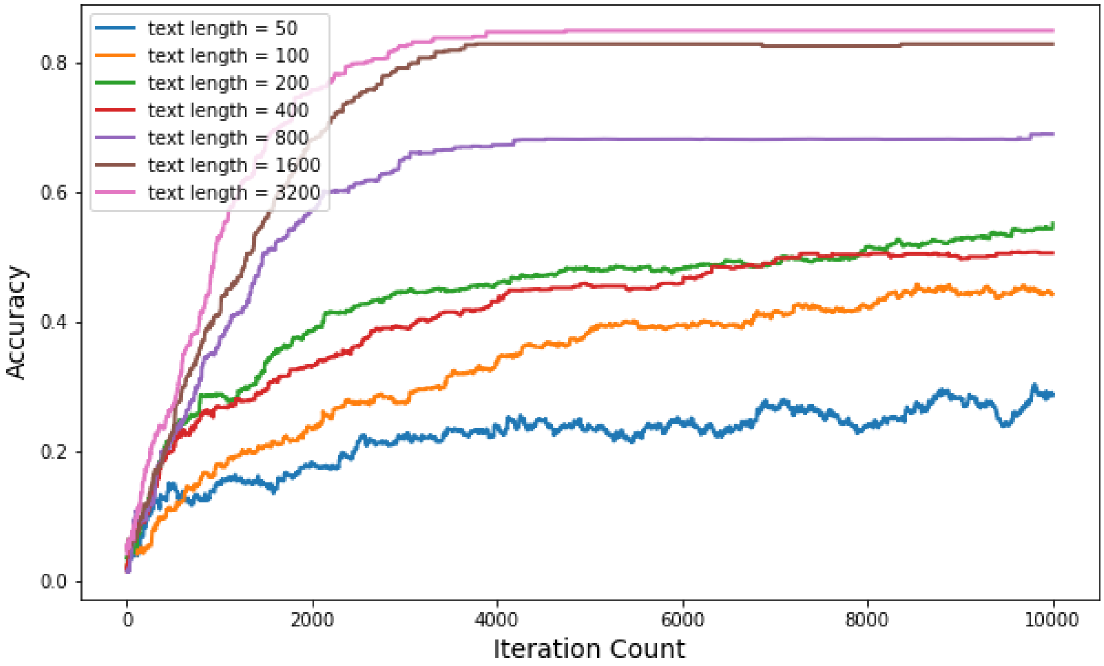

Figure 4: Average accuracies for ciphertexts of different lengths versus iteration count.

We can see that the average accuracies of longer ciphertext strictly bound the average accuracies when predicting on shorter ciphertext. In other word, with the same number of iterations, the average accuracy is monotonically increasing along with the length of input ciphertext. This is because when the input ciphertext is longer, the relative ratio of likelihoods of different cipher functions will be larger or smaller (far away from 1). In this sense, the MH algorithm will converge faster and achieve a higher accuracy within the same number of iterations.

3.5 (e)

I run the MCMC-based decoding algorithm on ciphertext with different lengths, (length $= 100, 200, 400, 800, 1600$) independently. For each ciphertext, I run 20 independent trials and record the average log-likelihood per symbol (in bits) of the 20 predictions. The average log-likelihood per symbol (in bits) versus iteration count is plotted as Fig. 4. The two plots are the same set of curves with different x-axis range. The x-axis range of the upper plot is $1 \leq \text{iter} \leq 5000$, and is $1000 \leq \text{iter} \leq 5000$ for the lower one.

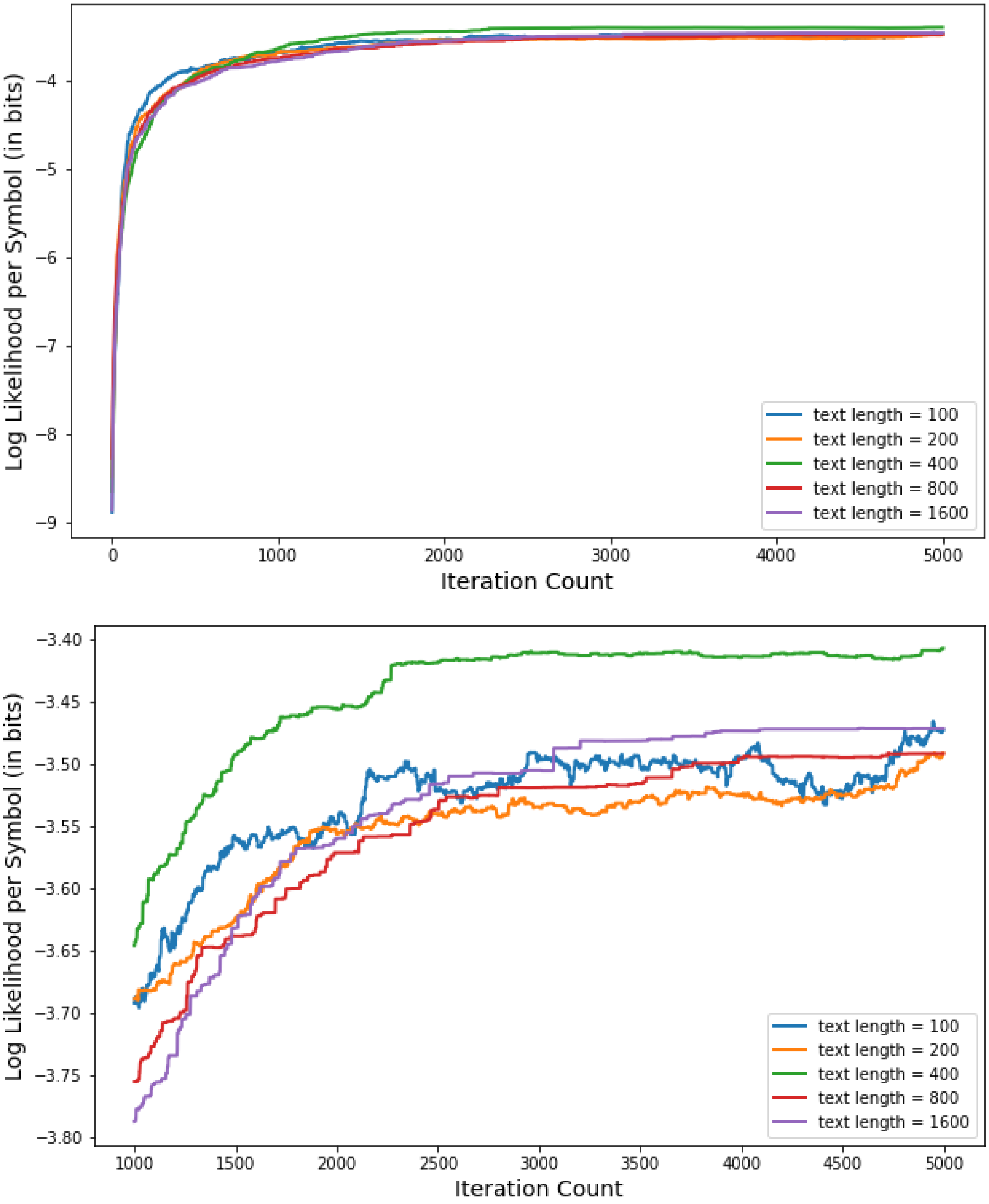

Figure 5: Average log-likelihood per symbol (in bits) versus iteration count.

We can see that the steady state values of these curves are generally independent of the length of ciphertext used. And this average is around −3.50. Since the steady state distribution of our MCMC-based algorithm is just the distribution of English texts (modeled as Markov chains), we can say that the steady value of these curves, −3.50, is the expectation value of the log-likelihood per symbol for English text. And clearly it is just the negative average Entropy per English symbol. From this, we can say the average entropy of English text is roughly 3.50 bits per symbol (letter). This empirical estimation is slightly larger than Shannon's estimation in "Prediction and Entropy of Printed English", which is 2.3 bits instead. But at least their magnitudes are similar, and this small difference is reasonable since the empirical entropy of English text depends on how the alphabet is modeled.