6.437 FINAL PROJECT — WRITE UP I

JUAN M ORTIZ

Problem I: A Bayesian Framework.

(a)

$$\begin{aligned} p_{\mathbf{y}|f}(\mathbf{y}|f) &= \mathbb{P}\left(\mathbf{x}_0 = f^{-1}(\mathbf{y}_0)\right) \prod_{j=1}^{n-1} \mathbb{P}\left(\mathbf{x}_j = f^{-1}(y_j)|\mathbf{x}_{j-1} = f^{-1}\right) \\ &= p_{f^{-1}(y)}(f^{-1}(\mathbf{y}_0)) \prod_{j=1}^{n-1} M_{f^{-1}(\mathbf{y})_j, f^{-1}(\mathbf{y})_{j-1}} \end{aligned}$$

(b)

$$\begin{aligned} p_{f|\mathbf{y}}(f|\mathbf{y}) &= \frac{p_{\mathbf{y}|f}(\mathbf{y}|f)p_f(f)}{p_{\mathbf{y}}(y)} \\ &= \frac{p_{f^{-1}(y)}(f^{-1}(\mathbf{y}_0)) \prod_{j=1}^{n} M_{f^{-1}(\mathbf{y})_j, f^{-1}(\mathbf{y})_{j-1}}}{\sum_{g\in\mathcal{F}}(p_{g^{-1}(y)}(g^{-1}(\mathbf{y}_0)) \prod_{j=1}^{n} M_{g^{-1}(\mathbf{y})_j, g^{-1}(\mathbf{y})_{j-1}})} \end{aligned}$$

Where $\mathcal{F}$ is the set of all permutations of $\mathcal{A}$. Thus, the MAP estimator is one that maximizes the above expression or, equivalently, its numerator. Thus

$$\hat{f}_{MAP} = \arg\max_{f\in\mathcal{F}} p_{f^{-1}(y)}(f^{-1}(\mathbf{y}_0)) \prod_{j=1}^{n} M_{f^{-1}(\mathbf{y})_j, f^{-1}(\mathbf{y})_{j-1}}$$

(c) Direct computation of the $\hat{f}_{MAP}$ is infeasable because it requires an optimization over a large, discrete, non-linear set. Computing expression on (b) would optimize over $\mathcal{F}$ which has a size of $|\mathcal{F}| = |\mathcal{A}|! = 28! \simeq 10^{29}$. Additionally, the constraints necessary to enforce that the $f \in \mathcal{F}$ is permutation will be hard to optimize over.

Problem 2: Markov Chain Monte Carlo method.

(a) After the first fixing one of the two cyphering functions $f_1 \in \mathcal{F}$, there are $\binom{|\mathcal{A}|}{2}$ possible $f_2 \in \mathcal{F}$ such that g differs in exactly two symbol assignments (i.e. those that swap two distinct element assignments in f_1). Thus, the probability that f_1 and f_2 differ in exactly two symbol assignments is given by:

$$\frac{\binom{\mathcal{A}}{2}}{|A|!} = \frac{1}{2(|A|-2)!}$$

(b) First, notice that the distribution $p_{f|\mathbf{y}}$ can be factorized in to hold the form $p_{f|\mathbf{y}}(f|\mathbf{y}) = \frac{1}{Z}\tilde{p}_{f|\mathbf{y}}$ where

$$\tilde{p}_{f|\mathbf{y}} = p_{f^{-1}(y)}(f^{-1}(\mathbf{y}_0)) \prod_{j=1}^{n} M_{f^{-1}(\mathbf{y})_j, f^{-1}(\mathbf{y})_{j-1}}$$

$$Z = \sum_{g\in\mathcal{F}}(p_{g^{-1}(y)}(g^{-1}(\mathbf{y}_0)) \prod_{j=1}^{n} M_{g^{-1}(\mathbf{y})_j, g^{-1}(\mathbf{y})_{j-1}})$$

If we let the proposal distribution $V(f'|f)$ be defined as

$$V(f'|f) = \begin{cases} \binom{|\mathcal{A}|}{2}^{-1} & \text{for f and f' differ in exactly two symbol assignments} \\ 0 & \text{otherwise} \end{cases}$$

When executing Metropolis-Hastings we compute the acceptance factor as follows:

$$\begin{aligned} a(f \to f') &\triangleq \min(1, \frac{\tilde{p}_{f|\mathbf{y}}(f'|\mathbf{y})}{\tilde{p}_{f|\mathbf{y}}(f|\mathbf{y})} \frac{V(f|f')}{V(f'|f)}) \\ &= \min(1, \frac{\tilde{p}_{f|\mathbf{y}}(f'|\mathbf{y})}{\tilde{p}_{f|\mathbf{y}}(f|\mathbf{y})}) \end{aligned}$$

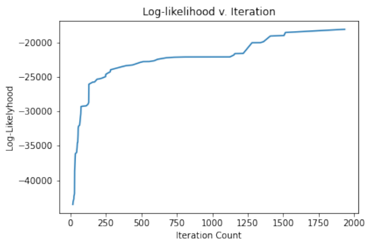
Note that in the event that $\tilde{p}_{f|\mathbf{y}}(f'|\mathbf{y}) = \tilde{p}_{f|\mathbf{y}}(f'|\mathbf{y}) = 0$, we will define $a(f \to f') = 1$.

Algorithm 0.1 Decoding using Metropolis-Hastings

(c)
```
$\hat{\mathcal{F}} \leftarrow \{\}$
$f \leftarrow$ random permutation of $\mathcal{A}$
while $\left|\hat{\mathcal{F}}\right| < 5000$ do
    $f' \leftarrow$ permutation obtained by swapping two assignments of $f$ at random.
    $a(f \to f') \leftarrow \min(1, \frac{\tilde{p}_{f|\mathbf{y}}(f'|\mathbf{y})}{\tilde{p}_{f|\mathbf{y}}(f|\mathbf{y})})$
    $r \sim \mathcal{U}(0,1)$
    if $r \leq a(f \to f')$ then
        $\hat{\mathcal{F}} = \hat{\mathcal{F}} \cup \{f'\}$
        $f \leftarrow f'$
    end if
end while
$f_{MAP} = \arg\max_{f \in \hat{\mathcal{F}}} \tilde{p}_{f|\mathbf{y}}(f|\mathbf{y})$
return $f_{MAP}^{-1}(\mathbf{y})$
```

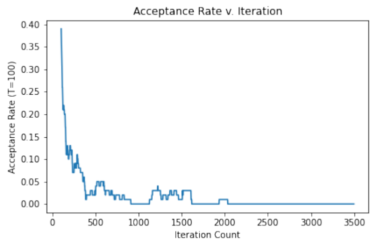
Problem 3: Implementation.

(a) Log-likelihood v. Iteration Count

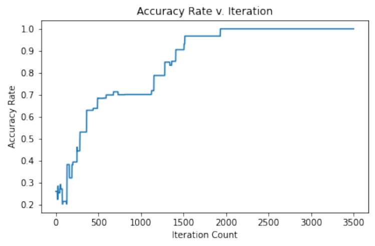


Note that in order to make convergence slightly faster, algorithm 0.1 was modified slightly by initializing the first permutation function to the permutation that matches the permutation that ensures that the ordering produced by the letter frequencies in P matches that of the emperical frequencies in the decoded ciphertext.

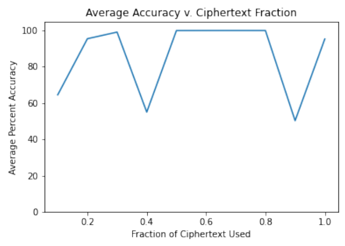
(b) Acceptance Rate (T = 100) v. Iteration Count



(c) Accuracy Rate v. Iteration Count

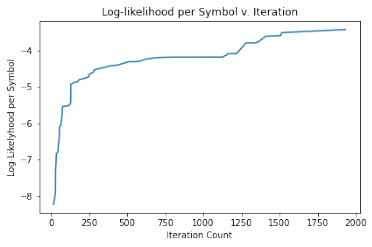


(d) The plot below depicts the average accuracy of decoded text as a function of the percent of the symbols of the ciphertext used. For each decoding run, the first $N * f$ symbols were used where f is the fraction of sybmols to be utilized during the run. For each fraction to tested, the average accuracy was computed over $\left\lceil \frac{1}{f} \right\rceil$ runs.



In general, the accuracy of the decoded text is highest when the entire ciphertext is used. This is because the longer the text, the fewer the permutations that generate an encoding that closely matches the true ciphertext. Thus, we have a higher probability of sampling an accurate ciphering function.
Note that the above graph does not always follow this conclusion however. This is partly due to randomness and the fact that, for a shorter text, one can achieve a high accuracy even if the ciphering function is not accurate for a small portion of the encoded text.

(e) As the number of incresases, the log-likelihood per symbol seems to approach a number close to -3.4. This suggests that emperical entropy per symbol is approximatelly 3.4 bits per symbol. Note this is higher than the reported entropy of english in the reference article of approximatelly one bit per symbol mostly due to our modeling choice of only taking into account the immediatelly preceding symbol.
The below plot depicts the evolution of the log-likelihood per symbol as we continue to iterate using (a slightly optimized version of) algorithm 0.1.


Log-likelihood per Symbol v. Iteration
Log-Likelyhood per Symbol
−4
−5
−6
−7
−8
0
250
500
750
1000
1250
1500
1750
2000
Iteration Count