HW1: Haplotype Phasing

CS 2840 Spring 2025

Released: Thursday, February 6, 2025

Due: Thursday, February 20, 2025, **8pm**

Overview

All homework assignments in this course will be submitted on gradescope, and must be completed in python. For this assignment, your submission should include a PDF file with your answers to 1 and 2, and the requested `clarks.py` for 3.

1 Warmup

a) How many unique explanations are there for the genotype 010220222121?

b) Pranav is pretty sure that we can come up with six explanations for 0221. In particular, he knows that the set of haplotypes involved with this genotype are 0001, 0011, 0101, 0111; he's pretty sure you can combine any pair of them and get a valid explanation (hence, six). Why is this not the case? Give an example of a pair that is not a valid explanation.

c) Will the Clark algorithm succeed with the set of genotypes $G = \{0201, 0222\}$? If so, explain why it starts and terminates, then solve for the explanations of each genotype according to the algorithm by hand. If not, explain why if fails to start or why it fails to terminate (note the specific problematic SNPs).

d) Answer *c*), but this time for $G = \{0112, 2201\}$.

e) Explain how we can reduce a sequence of SNP data for a given person (e.g., a list of k pairs of letters in $\{A, C, T, G\}$) into a sequence of the same length in $\{0, 1, 2\}$, if all we care about is testing SNPs and haplotypes for associations with disease.

f) Sequencing data, in this day and age, is generated by taking many short reads (less than 600bp...) over the genome and using algorithms to reconstruct a likely version of the whole genome. How would the haplotype phasing problem change if we could suddenly read the whole genome at once (or even read 1000bp segments)?

g) Describe the heuristic of the Clark algorithm; i.e., what assumptions does it make about the set of haplotypes in the population?

2 EM

Follow through the below story! There are a few signposted places where you need to provide question answers. Before reading, you might want to first read our note on MLE at the end of this document.

Maximum likelihood estimation is a very global notion (sometimes very hard to solve, as with the haplotype phasing likelihood function), and does not allow for the possible simplification gained by assuming latent variables govern the data X. As a result, in contexts where we have clear latent variables Z (haplotype phasing!), we instead use *Expectation Maximization*. Here, rather than directly maximizing the likelihood function, we maximize the *expected likelihood function, over all possible latent variables.* In particular, we compute:

$$\mathcal{E}(\theta) = \mathbb{E}_Z(\log \mathcal{L}(\theta)|X, \theta)$$

You can think of this as the average $\mathbb{P}(X|\theta)$ across all possible notions of Z, weighted by the probabilities of observing (or not observing!) Z as our latent variable. To maximize this, we solve a constrained optimization problem. **a: How would you write down the system of equations you need to solve, assuming θ is a distribution (hint: you can do it in one line!)?**

From here, we take the new θ we get from this maximization, and recompute $\mathbb{E}_Z(\log \mathcal{L}(\theta)|X, \theta)$.

We saw in class that EM is iterative. But, why do we even need to recompute $\mathbb{E}_Z(\log \mathcal{L}(\theta)|X, \theta)$ for iteration? In MLE, we only need to maximize once. The answer is in what has been a bit vague so far in this introduction to EM. In particular, we never observe Z; we only take a guess at it every iteration by computing $\mathbb{P}(Z|X, \theta)$ during the expectation. It only appears as part of the joint distribution over X and Z, parameterized by θ. So, EM iteration computes the distribution of Z from θ during the expectation step, then computes θ by maximizing the resulting expectation. We kind of iterate between "solving" for Z and solving for θ. That's the general form of EM!

One small caveat before we move to the world of haplotypes once and for all. We've misled you by implying that $\mathcal{L}(\theta)$ is the same in both the MLE and EM worlds. As is maybe clear from the setup, $\mathcal{L}(\theta)$ is related to $\mathbb{P}(X|\theta)$ in MLE world, and $\mathbb{P}(X, Z|\theta)$ in EM world. This leads to a different likelihood function in the EM case, which we call the *complete-data likelihood function.* The complete-data likelihood function assumes we also know the latent variables, which we did not assume in our haplotype phasing likelihood function above.

Before we come up with a complete-data likelihood function for haplotype phasing, we need to define our latent variables. Our observed variables are the number of people with each genotype. The natural latent variable Z, as a result, is the number of people with each *explanation*, n_{kl}, where $\sum_{h_k, h_l \in H_j} n_{kl} = n_j$. We can write a new multinomial $\mathcal{L}_c$ (as in the MLE note below) that encapsulates this information, assuming we also get to see the latent variables Z:

$$\begin{aligned}\mathcal{L}_c(\theta) &= \prod_{h_k, h_l \in \cup_{j=1}^m H_j} \mathbb{P}(h_k, h_l)^{n_{kl}} \\ &= \prod_{h_k, h_l \in \cup_{j=1}^m H_j} [(2 - \delta_{kl})\theta_k \theta_l]^{n_{kl}}\end{aligned}$$

b: Given this new complete data likelihood function, what is the expectation function we want to maximize? Simplify it so that the only remaining expectation is $\mathbb{E}_Z(n_{kl}|X, \theta)$ (hint: use log and expectation properties). Make sure to justify each step.

With this new expression, we need only solve for $\mathbb{E}_Z(n_{kl}|X, \theta)$ to finish the so-called "E-step". In other words, we want the expected number of people with each possible explanation, given the number of genotypes we saw and the parameters θ. We know that Z is a vector of numbers of each possible explanation n_{kl}. Since a given h_k, h_l pair can only be associated with a single observed genotype j, we can use something called the indicator trick.

Let $\mathbb{1}_{i,kl}$ be 1 if individual i has explanation h_k, h_l and 0 otherwise. Note that $\mathbb{E}(\mathbb{1}_{i,kl}|X, \theta, j) = (1)\mathbb{P}(h_k, h_l|j)$. We need to condition on the j associated with kl, as we only want to sum over the n_j people with genotype j when we are considering how many people have explanation (h_k, h_l); it allows us to bound the sum on n_j individuals in the manipulation you will now complete. **c: Using this notation, solve for $\mathbb{E}_Z(n_{kl}|X, \theta)$ in terms of n_j, $\mathbb{P}(j)$ and $\mathbb{P}(h_k, h_l)$. You should have no sums remaining; it should be a single expression (hint: how can you express n_{kl} as a sum?).**

The so-called "E-step" we saw in lecture (and in Excoffier-Slatkin, 1995) was:

$$P_j(h_k, h_l)^{(g)} = \frac{n_j}{n} \cdot \frac{P(h_k, h_l)^{(g)}}{P_j^{(g)}}$$

d: How does this relate to the expectation you just derived?

e: The maximization step, while possible to solve for directly, is highly impractical to derive. Write a few sentences about why this is the case. As it turns out, a gene-counting method is used to derive the maximization step, although slightly less rigorous than the direct approach. We won't cover exactly how it is done in this homework.

At the end of this question, you should have answers to parts $a - e$.

3 Clark Algorithm

In this problem, you will implement the Clark algorithm. In particular, in your (autograded) gradescope submission, you will submit one file (named `clarks.py`) that contains the following function:

```python
def clark_algorithm(genotypes):
    """
    Clark's algorithm for haplotype resolution.

    Args:
        genotypes (List[str]): List of genotypes to resolve, given as strings of 0, 1, and 2.

    Returns:
        Set[str]: Set of resolved haplotypes,
        List[Tuple[str, Tuple[str, str]]] List of tuples containing the original genotype and the
            resolved haplotype pair.
    """
```

Note that the returned list must be in the same order as the inputted list. It should fail when the Clark algorithm fails (any such case we discussed in class).

Supplement

No questions here!

MLE

In the standard version of EM (the successor of MLE), we have some observed data X that is governed by some set of unobserved variables Z. In general, such a model is probabilistic (i.e., X and Z follow a joint distribution), which we can parameterize with θ, a vector with entries $\theta_i|_{i=1}^{n}$. For our purposes, we'll always think of θ as a distribution, with $\sum_{i=1}^{n} \theta_i = 1$.

Additionally, we usually have some likelihood function $\mathcal{L}(\theta)$. You can think of the likelihood function as related to $\mathbb{P}(X, Z|\theta)$ – the probability we observe the data we did, given the parameters we have. The likelihood function is proportional to the (parameterized) joint distribution of X and Z discussed above.

In *Maximum Likelihood Estimation*, we maximize the this function *directly* with respect to θ (which is often easier to do by taking the log first). In particular, we maximize with respect to the constraint that $\sum_{i=1}^{n} \theta_i = 1$ (such that θ is a valid distribution). According to the method of Lagrange multipliers, this means we need to solve for the system of equations that arises from:

$$\nabla[\log \mathcal{L}(\theta) + \lambda(1 - \sum_{i=1}^{n} \theta_i)] = 0$$

Where we let λ and all θ_i. More concretely, the system of equations we have is:

$$\frac{\partial}{\partial \theta_i} \log \mathcal{L}(\theta) + \lambda(1 - \sum_{i=1}^{n} \theta_i) = 0$$

for all i and

$$\frac{\partial}{\partial \lambda} \log \mathcal{L}(\theta) + \lambda(1 - \sum_{i=1}^{n} \theta_i) = 0$$

Since we have $n+1$ equations and $n+1$ unknowns, we can solve for the θ that satisfies our constraint and maximizes the likelihood of observing the data, X. A beautiful concept! Notice that we did not rely on any latent variables here; in fact, we sort of assumed they weren't present, like $\mathcal{L}(\theta) \propto \mathbb{P}(X|\theta)$ instead. As we saw in lecture, when we apply this idea to haplotype phasing, we unfortunately get a difficult, non-convex $\mathcal{L}$ for which this system of equations is impractical to solve on a large scale. If you recall,

$$\begin{aligned}\mathcal{L}(\theta) &= \prod_{j=1}^{m} \Big(\sum_{h_k, h_l \in H_j} \mathbb{P}(h_k, h_l) \Big)^{n_j} \\ &= \prod_{j=1}^{m} \Big(\sum_{h_k, h_l \in H_j} (2 - \delta_{kl}) \theta_k \theta_l \Big)^{n_j}\end{aligned}$$

Where the observed data we get is $X = n_j|_{j=1}^{m}$ (a vector of counts). This $\mathcal{L}(\theta)$ is proportional to $\mathbb{P}(X = n_j|_{j=1}^{m} |\theta)$, which as above, has X following a multinomial distribution over the different possible genotypes in the population. Concretely, the likelihood function above takes the product over all possible genotypes j, raising the probability of their occurrence, $P_j = \sum_{h_k, h_l \in H_j} \mathbb{P}(h_k, h_l)$, to the number of times they occur, n_j. If you're familiar with the binomial distribution, this is the m dimensional version.

As an aside, we are denoting H_j as the set of haplotype pairs, (h_k, h_l), that can explain the jth genotype in the population. So, by summing over H_j, we get the probability that an individual has genotype j in the population. As another aside, $\mathbb{P}(h_k, h_l) = (2 - \delta_{kl})\theta_k \theta_l$ is a compact way of noting that the probability of an explanation (h_k, h_l) is $2\theta_k\theta_l$ if $k \neq l$ and $\theta_k \theta_l = \theta_k^2$ if $k = l$. If you're not familiar with the Kronecker delta, $\delta_{kl} = \mathbb{1}_{kl} = \begin{cases} 0 \text{ if } k \neq l \\ 1 \text{ otherwise} \end{cases}$. We distinguish between these cases, as we can order an observation of $(h_k, h_l), k \neq l$ in two ways, while we can only order it one way when $k = 1$. This is where the *random mating assumption* is essential: it allows us to claim that $\mathbb{P}(h_k, h_l) = (2 - \delta_{kl})\theta_k \theta_l$ because random mating implies seeing haplotype k and seeing haplotype l are independent events.