CS 124 Notes

Lecture Notes

1/7 - Week 1

- Goal is to understand what we can know based on our genetics.
    - Understanding through computational and statistical techniques.
- A,C,T,G are nucleotides or base pairs. We have 3 billion of these in the human genome.
- Humans have 23 pairs of chromosomes.
    - X and Y are the sex chromosomes.
- Each individual has a very unique genome, it's a unique identifier. However, the number of differences is small and thus most of our genomes are the same.
- Genetic variations or mutations are examples of base pairs that are different from person to person. A single nucleotide polymorphism (SNP) is the most common mutation and represents a situation where a single nucleotide is changed at the same position from person to person.
    - Some percentage of the population has a A while the rest has a T.
    - The majority of the SNPs have alleles, or two possible nucleotides for that position.
- For most diseases, we know about the mutations of genes that impact a person's likelihood of getting a particular disease.
- Penetrance is the probability of getting the disease if you have the mutation.
- Constructing disease models involves trying to determine what the genetic and environmental factors are for getting the disease.
    - Look specifically at the heritability of the disease. Estimate the contribution of the genetic vs environmental factors.
        - Heritability is a statistic used in the fields of breeding and genetics that estimates the degree of variation in a phenotypic trait in a population that is due to genetic variation between individuals in that population.
    - Once we have this disease model, we can try to predict treatment outcomes.
- Genome Wide Association Studies (GWAS) are where you study complex diseases by comparing control patients to cases. You want to find the differences between the patient and the non-patient DNA.
- DNA sequencing is the extraction of the string of A,G,C, and T from the tissue (blood, saliva, etc).

1/16 - Week 2

- I think the big picture view of MLE is that it's used in situations where we have some data that we observe or generate, and we want to figure out the parameters for the most likely hypothesis. Basically, we want to solve Pr(D | theta) for all possible thetas to see

which one is the most likely. In other words, want the hypothesis that maximizes the likelihood.
    - L(theta ; D) = Pr(D | theta) <- Find the theta that maximizes L
- Likelihood wants to find p given n and m (in a binomial variable context). We basically know that X is a binomial random variable. We take a random sample and figure out that there are m successes. From that info, we want to find the most likely value for p. You can see that depending on the data that gets generated, we can settle on a more informed value for p.

$$L(p; X = m) = \Pr(X = m \mid X \sim B(n, p))$$

    -
        - From a high level, this function assigns a number to each possible value of p.
    - In order to find that value for p, we write out the right hand side of the equation which is
        - $$L(p; X = m) = Pr(X = m \mid X \sim B(n, p)) = \binom{n}{m} p^m (1-p)^{n-m}.$$
    - And thereby create a f(p) and then take the derivative and set equal to 0
        - $$L'(p; X = m) = \binom{n}{m} p^{m-1} (1-p)^{n-m-1} (m(1-p) - (n-m)p)$$
    - And then we find p. I guess we are finding the p that results in the maximum value for likelihood.

$$Maximum\,likelihood = \arg\max_\theta L(\theta; D)$$

In the example above, the maximum is obtained

for $\hat{p} = \frac{m}{n}$

    -
    - The above p is for the case of X being a binomial RV.
- In all maximum likelihood, the likelihood is written in terms of theta and D where theta refers to a set of parameters (L(theta ; D)). Each hypothesis is specified by a different value for theta. We have an infinite number of hypotheses and thus we want to find the best one that fits the data D that is generated. This, in effect, is finding the theta/hypothesis the maximizes the likelihood function.
    - So in a sense, we have this data and we want to find the theta (in effect the hypothesis) that is most likely.
    - We use this approach because in most cases, we want to model P(theta | D), but we really are making use of P(D | theta).

- In the next example, let's remember the normal distribution which has the parameters mu and sigma squared. This refers to the mean and variance which affects the center of the curve as well as its shape. Now, let's sample from that distribution and get a set of n numbers. From that data we just generated, now let's see if we can use MLE to estimate the parameters mu and sigma.
    - First we set the likelihood equal to the probability of seeing the data given the info we have about the variable.
        - $$L(\mu, \sigma; x_1, \ldots, x_n) = Pr(x_1, \ldots, x_n | \mu, \sigma)$$
    - Basically it is just the product of all the PDFs for each of the x_i's.
        - $$\prod_{i=1}^{n} f(x_i) = \frac{1}{\sigma^n (2\pi)^{n/2}} e^{-\frac{1}{2\sigma^2} \sum_{i=1}^{n} (x_i - \mu)^2}$$
    - Then I think you calculate partial derivatives and differentiate that wrt mu and sigma separately and you get:
        - $$\hat{\mu} = \frac{\sum_{i=1}^{n} x_i}{n}$$
        - $$\hat{\sigma} = \sqrt{\frac{\sum_{i=1}^{n} (x_i - \hat{\mu})^2}{n}}$$
- Next example is the same thing except we sample numbers from a uniform distribution. We are given those n numbers and we want to find theta, which is the max upper limit for the uniform distribution. Intuitively, a good guess for that theta would be the largest x_i that we see. Or you could find the average/median of all the data points and multiply that by 2.
    - If the largest number sampled (X_n) is greater than theta, then the likelihood is 0 because there is no possibility that that data could have been generated if the upper limit of the distribution is smaller than one of the data points that we saw.
    - But if the largest number is less than or equal to theta, then the likelihood is a product of all the f(x_i) and thus the likelihood in the end is 1/theta^n. From that likelihood, we conclude that theta should be X_n (the largest number sampled).
        - $$L(\theta; x_1, \ldots, x_n) = \prod_{i=1}^{n} f(x_{(i)}) = \frac{1}{\theta^n}$$
- In MLE, you want the estimate with the least variance.

- When estimating haplotype frequency, you have a multinomial distribution since each haplotype will have a different frequency. The data that we have are the counts of each of the haplotypes.

1/23 - Week 3

- There is a value in trying to determine the particular areas within the genome that we want to sequence rather than sequencing everything. We want to find the areas that are most likely to be different between people and most likely to cause some downstream effect.
    - Want to identify the couple million SNPs that are involved in disease rather than looking at all the possible SNPs.
    - An associated SNP is a SNP where the allele frequency distribution is different between the cases and the controls.
- There are correlations between SNPs that are physically close together in the genome.
    - We can learn about one SNP based on other SNPS in its proximity without the need to measure it.
- Haplotype is a sequence of alleles across one of the chromosomes.
    - Each of the alleles represents one SNP
    - We always have two copies of a chromosome (one from mom and one from dad), this refers to a region on one of the chromosomes.
- On the block structure, every row is one haplotype and every column is one SNP. When you sort the haplotypes, you can see which haplotypes are different and what exact SNP they differ on.
- The red graph shows you the correlation between two different SNPs, a positive correlation indicate by a red dot at the intersection of the two SNPs.
- Recombination is the event where there is a random walk between the two copies of chromosomes that you receive.
- When you're looking at allele frequencies, it's just m / 2n since we draw from a Bin(2n, p) variable where p is the allele frequency. The MLE estimate is p = m / 2n.
- Haplotype frequency
    - But when you want to get haplotype frequencies, every haplotype is a binary string of k letters, and you have t of these haplotypes. Then, you get the t different counts for all the haplotypes. In a sense, this is like your observation.
    - You also have a set of t probabilities that each represent the probability that you see that particular haplotype. The sum of all the probabilities has to be 1. We don't know those probabilities rn, but we're going to use MLE to try to estimate it based on the observations that we see.
    - Now, instead of setting up a binomial distribution, we do a multinomial distribution where we describe the likelihood as:
        - $$L(p_1,\ldots,p_t;c_1,\ldots,c_t) = \binom{n}{c_1}\binom{n-c_1}{c_2}\cdots\binom{n-c_1-\ldots-c_{t-1}}{c_t} p_1^{c_1} p_2^{c_{21}} \cdots p_t^{c_t}$$

- Given that, we want to find

$$Max \quad \sum_i c_i \log(p_i)$$

$$s.t \sum p_i = 1, p_i > 0$$

  - ■
- After taking the derivative and solving for the highest probability, we see that

Let $p^*=(p_1,\ldots,p_t)$ be the optimum. Let $p' = (p_1,\ldots,p_i-\varepsilon,\ldots,p_j+\varepsilon,\ldots,p_t)$.

$$f(p') \approx f(p^*) + \varepsilon\left(\frac{c_j}{p_j} - \frac{c_i}{p_i}\right)$$

$$\Rightarrow \frac{c_j}{p_j} = \frac{c_i}{p_i}$$

$$\Rightarrow p_i = \frac{c_i}{n}$$

  - ■
- You can also get that same conclusion by using Langrange Multipliers where you maximize the following function instead

$$\max \sum_i c_i \log(p_i) + \lambda(1 - \sum_i p_i), \ s.t. \ p_i > 0$$

  - ■
- Then you compute gradients and shit

$$f(\vec{p}, \lambda) = \sum_i c_i \log(p_i) + \lambda(1 - \sum_i p_i)$$

Compute the gradients:

$$\frac{\partial f}{\partial p_i} = \frac{c_i}{p_i} - \lambda \qquad \frac{\partial f}{\partial \lambda} = 1 - \sum_i p_i$$

Equating to zero:

$$p_i = \frac{c_i}{\lambda}, \lambda = \sum_i c_i = n$$

  - ■

1/28 - Week 4

- The haplotype estimation that we've been discussing is a multinomial distribution.
- There are going to be situations where you have an individual with a haplotype that is missing some data. Our goal still remains to find the haplotype frequencies in the population.

    - Want to try to figure out the missing parts by looking at what is typically in that area and/or the info next to the missing part.
- A complete haplotype is a binary string of some length k
- A partial haplotype is string that has 0,1, and * for the unknown locations.
- Partial h compatible with complete h' if they match in any non * position.
    - In other words, you can assign values to all the asterisks to match h and h' equal to each other.
    - Compatible set C(h) = {h' | h' and h are compatible}
        - This is basically the set of all the complete haplotypes h' that are compatible with the partial h.
- Example: Let's say you have know you have m haplotypes where each haplotype is a complete haplotype - binary string of k characters.
    - We can think of these as being the data generated from drawing from some distribution with some unknown parameters at the moment. We basically have m probability parameters and they represent the probability of seeing those haplotypes. Last week, we saw that p = c / n
    - Now, with some probability q (that is the masking probability, aka the probability of flipping a known bit to an asterisk), let's convert each of the complete haplotypes to partial ones. This is the process for partial haplotype generation.
        - Repeat n times:
            - Pick a random haplotype h from the distribution p_1, p_2, …, p_m.
            - For each position j, $m \geq j \geq 1$, mask h(j) with probability q: h(j) = '*'.
    - We can assume the haplotypes themselves are known and the overall goal is to determine the probability to observe each of them.
- To overview the steps
    - 1) You know what the possible haplotypes are. Generate a set of complete ones
    - 2) Mask each of the values in each haplotype with some probability q.
- Let's say you draw a haplotype (and since you have a list of the possible haplotypes) and you know the haplotype is h5. Normally, the probability of getting it is p5. However, with the masking stuff, the probability becomes p5 * (1-q)^k.
- Let's say you draw a haplotype and they can be both partial or complete. If it's complete, then it's just the corresponding probability for that haplotype. If it's partial, then you add the probabilities of the different haplotypes it is compatible with.
    - An example would be:

INPUT

```
11111111
1111111*
0000000*
00000000
```

```
Possible distributions
11111111 w.p p1
11111110 w.p p2
00000000 w.p p3
00000001 w.p p4
Others w.p 0
```

- `Likelihood = p1×(p1+p2)×(p3+p4)×p3×f(q)`

- The total likelihood is just the product of all those terms. Therefore, the likelihood is

$$L(p_1, \ldots, p_m; \mathcal{H}) = f(q) \prod_{h \in \mathcal{H}} \sum_{i: h \in C(h_i)} p_i$$

- We want to maximize the likelihood. We take the gradient of the following function.

$$g(\vec{p}) = \sum_{h \in \mathcal{H}} \log \left(\sum_{i: h \in C(h_i)} p_i \right) \qquad \max g(\vec{p}) \quad s.t. \sum_{i=1}^{m} p_i = 1, \quad p_i \geq 0$$

Gradient of g:

$$\frac{\partial g}{\partial p_i} = \sum_{\{h: h \in C(h_i)\} \cap \mathcal{H}} \frac{1}{\sum_{j: h \in C(h_j)} p_j}$$

Projected on the space $\sum_i p_i = 1$:

$$u = \left(\frac{\partial g}{\partial p_1} - \lambda, \ldots, \frac{\partial g}{\partial p_m} - \lambda\right)$$

where $\lambda = \dfrac{\sum_{i=1}^{m} \frac{\partial g}{\partial p_i}}{m}$

- Then, we choose some random point p (m values in an m-dimensional space that represent the probability haplotypes). Once we get the gradient, then we perform gradient ascent to choose better points.

- Start from a point
  $$p^0 = (p_1^0, \ldots, p_m^0)$$
- Repeat:
  1. Current point is $p = (p_1, ..., p_m)$
  2. Calculate
     $$u = (\frac{\partial g}{\partial p_1} - \lambda, \ldots, \frac{\partial g}{\partial p_m} - \lambda)$$
  3. Find a new point for a small epsilon > 0:
     $$p' = p + \epsilon u$$
  4. Back to step (2) until there the gradient is zero or until you get to a corner (p_i=1 or p_i=0)

    - As a side note, in order to maintain the constraint of making sure the probabilities sum to 1, you need to subtract each value in the gradient by lambda.
        - The way you do that is you create an m dimensional plane where every point on that plane corresponds to a point where all the p values sum up to 1. At the beginning of the optimization, the first point is on that plane. Then, when you try to add the gradient, then you likely go to some place off of the plane. In order to enforce the constraint that the pi's sum to 1, then you need to project that new point onto the plane. This means the new point is on the plane again and will satisfy the constraint. The lambda is the vector that gets the point onto the plane.
- These methods of hill climbing don't guarantee a global optimum.
- A phase is a possible child genotype given the two haplotypes from the mother and father.
- In the phasing, slides, H represents heterozygous.
- The Pure Parsimony approach seeks to explain the genotypes using the fewest number of distinct haplotypes.
    - Basically, try to find a minimum set of haplotypes that are consistent with the phasing.

1/30 - Week 4

- TODO understand parsimony stuff
- Clark's method said that if you know what the set of haplotypes are, you can use those for phases?
- If you have a genotype, then you can determine its two current phases. For the phases that don't have any question marks (appear when you have multiple heterozygous), you have a list of known haplotypes.
    - Goal is to use those known haplotypes to try to resolve the unknown phases for the other genotypes. You're not introducing any new haplotypes, you're just making the known ones fit with the phases for the unknown genotype.
    - You can introduce more haplotypes if the ones you currently have don't properly explain the phases for all the genotypes.

- As a follow up to ^, the basic procedure is to take your genotypes, and for each of them, figure out the phases
- The big O of the above procedure is O(n^2m) where there are n genotypes and for each of them, there are 2n haplotypes (maximum number of haplotypes there could be) to compare. M is the number of SNPs
    - Basically it is 2n * n * m
- Let's say you have t haplotypes and for each you have a probability. All the haplotypes (h1 … ht) are getting drawn from a multinomial distribution with those probability values p1 … pt. Then you take that table of haplotypes, take the pairs, and create the genotypes (basically the sum of the two vectors). Basically the genotypes constitute of counts for how many times you see the allele (If you have a 1, then the allele at that position is heterozygous, and if 0/2 then it is homozygous).
    - Now we want to compute the likelihood L(p1 … pt | g1 … gt). It is equal to the product of all the probabilities of observing a certain genotype.
    - The probability of each genotype is the sum of all the probabilities of all the different haplotype combinations that result in the genotype. Those probabilities are made up by the product of those two individual probabilities for the haplotype.
        - For example if you wanted to find the probability of the genotype 000, then the only haplotype combination is 000 and 000. And so the probability for that combination is p_000 ^ 2.
        - Another example is if you wanted to find the probability of the genotype 111. Then the two different ways to get there are 000 / 111 and 010 / 101. The probability sum is 2 * p_000 * p_111 + 2 * p_010 * p_101
    - Overall the likelihood is:

$$\log L(p_1, \ldots, p_n; \mathcal{G}) = \sum_{g \in \mathcal{G}} \log \left(\sum_{h_1, h_2 \in \mathcal{H}, h_1 + h_2 = g} p_{h_1} p_{h_2} \right)$$

        -
    - S
- Hard decisions say that for each genotype we have hard guesses for what the phases are. Soft decisions are where you put a probability distribution over the different possibilities for which pair of phases the genotype is made of.
    - The pro of soft decision is that it is more representative and informative but the con is that it's harder to handle that uncertainty in later downstream tasks with the output.
- In EM, we start with the initial probabilities for each of the haplotypes as 1/ 2^m where m is the number of SNPs. You then do the expectation step which

2/4 - Week 5

- 2^n / 2 number of phases because you don't care which is on top and which is bottom TODO

- Pure parsimony is a method of kind of scoring each possibility of haplotype combinations to explain the genotype.
- MLE soft assignment is basically going over all the genotypes, and for each genotype, looking at all the pairs of haplotypes that can explain the genotype (the compatible ones) and look at the sum of the product of the probabilities of those haplotypes.
- In the slide with the iterative example, we see the probabilities in the middle (instead of 1/144) because we are calculating conditional probability of the phase combination given the genotype on the left.
    - It's basically just normalization tbh
- .75 / 6 for the first haplotype (10000) in that slide.
- Log of the weighted sum of a and x is greater than the sum of a times the log of x
    - In that proof on slide 33, we are showing how you can use Jensen's inequality to rewrite the likelihood formulation.
- In EM, the parameters we want to estimate are the probabilities of the haplotypes, the phase is the latent variable (the Z), but we only have the genotypes as the data D.

2/6 - Week 5

- In the M step of the EM algorithm, it is easier to optimize the Q function rather than the likelihood.
- When doing EM for SNP allele frequency, for each individual, we get a specific call (0,1,2) or an interval ({0,1}, {1,2}). For each individual, we get an interval and the allele frequency is denoted by p, and that is the quantify we want to estimate.
    - Given that we are able to find out that allele frequency p, we can obtain the probabilities for getting genotypes p_0 = (1-p)^2 and p_1 = 2p(1-p) and p_2 = p^2
- Example: Individuals have 0,1,{0,1},{0,1,2}, {1,2}
    - The probability of observing this group of individuals is (1-p)^2 * 2p(1-p) * [(1-p)^2 + 2p(1-p)] * [2p(1-p) + p^2]
    - The above would be the likelihood. If we want the log likelihood then we get 2log(1-p) + log(p) + log(2p) + log(1-p^2) + ...
- When we're doing EM, the Z would kinda represent the fixing of that individual interval to a specific assignment. We do this across all the possible values and we average over. That is the expectation part.
    - For example, if we look at Z = (0,1,0,0,1), then we calculate log(P(Z = (0,1,0,0,1))) = log((1-p)^2) + log(2p(1-p)) + ….
    - For the expectation part, we take the summation of P(Z_i | p_t) * log(P(Z_i | p)). This is equivalent to
      Sum over the n examples: p_t(Z_i = 0) * log((1-p)^2) + p_t(Z_i = 1) * log(2p(1-p)) + p_t(Z_i = 2) * log(p^2)
- In the slide 41, the reasoning for the 2nd to last genotype calculation is
  2 * P(g = 2 | interval {1,2}) + 1 * P(g = 1 | interval {1,2}) = (¼ / ¾) + (½ / ¾) = 2 * ⅓ + ⅔ = 4/3

- A diagonal matrix multiplied to a normal distribution will not change the mean, but will change the variance.
- A scalar added to a normal distribution will not change the variance, but will change the mean.
- For the multivariate normal distribution slides, the size of mu is d, and the size of sigma is d^2.
- When trying to apply EM, the theta is (mu, sigma, p's), the data D is (x_i's), and the Z is the assignment of the points to the clusters.
- The likelihood that you see the point in a particular cluster is the f function and you multiply it by the probability of seeing that cluster.

2/11 - Week 6

- Important to look at population substructure in order to make sure that your association signals are meaningful and not just the result of different characteristics of different populations.
- Another problem we can try to tackle is a prediction problem where we take a genotype and try to infer their ancestry, or in other words, where they come from.
- Some of the differences between people in different areas of the world could be differences in particular SNPs that give people in certain areas evolutionary advantages.
- You can also have mixed ancestries where your mom and dad come from different places.
- To formalize things a bit more, we can assume an individual is a mix of K populations.
    - Then you have a vector A with k different items and each item represents the proportion of ancestry coming from that kth population.
    - Then you also have a matrix f that contains the allele frequencies for each population at each SNP.
    - The genotype of an individual at a particular SNP is {0,1,2}
- Negative definite Hessian matrix is required for determining if a point is the global maximum.
- Another problem that we can solve is if we have reference populations P1 and P2, as well as some third population P3, can we detect whether that 3rd mixture is an admixture of P1 and P2?
    - In order to do this, we need to use linkage disequilibrium (correlation of linked SNPs). We need some measure of that disequilibrium which is basically looking at some difference between the product of individual probabilities and the joint probabilities for combinations of alleles.

2/13 - Week 6

- Going back to the prediction problem last week, measures of linkage disequilibirum could be
    - D = P_AB - P_A*P_B
    - D = P_A*P_b - P_Ab

    - D = P_ab - P_a*P_b
    - D = P_a*P_B - P_aB
- You can think of there being n individuals in the population, and X being a random variable that represents that number of individuals with the A allele. X is drawn from a binomial distribution with parameters n and P_A (time = 0).
    - E(X) = n * P_A (time = 0).
        - E(X / N) = P_A (time = 0)
    - Var(X) = n * P_A * (1 - P_A)
        - The variance goes to 0 as n goes to infinity and thus you can prove that P_A won't change from time step to time step.
- We prove the at P_A (time = t) = P_A (time = t+1)
- However, we cannot say the same about the relationship between P_AB at time t and P_AB at time t+1.
    - P_AB (time = t+1) = (1 - r) * P_AB (time = t) + r * P_A (time = t) * P_B (time = t)
- Now, because we've figured those two out, we can say
    - D (time = t+1) = P_AB (time = t+1) - P_A (time = t+1) * P_B (time = t+1)
    - After expansion with those time = t terms, we see that D (time = t+1) = (1 - r) * D (time = t). A more general form of that is D (time = t) = (1 - r)^t * D (time = 0)
- For a population with infinite population size, D decreases with time. When recombination rate is higher, D changes faster.

2/20 - Week 7

- Admixture mapping involves the goal of finding regions with disproportionate local ancestry.
- A hidden markov model in the genetics context has N different states for every SNP that we see. The state at time t depends only on the state at time t-1. This is the big Markov property.
    - We will have transition probabilities from each SNP to the SNP at the next time step. The outgoing edges for each state in the SNP need to sum up to 1. The incoming edges to the SNP at the next time step needs to sum to N which is the number of states previously.
    - At each state, you also have emission probabilities.
- The hidden aspect of the HMM is because we don't know about the "weight"/transition probabilities from SNP to SNP.
- Different HMMs can be created for each population.
- There is also a prediction problem where you are given a genotype with K SNPs, and then you try to predict the ancestry which is a vector of the same length but the value at the particular SNP refers to the population that allele is coming from.
    - Need a HMM for the different populations from a set of haplotypes from the populations. Need to estimate the graph, the transition probabilities, and the other HMM parameters.
        - Want to find the most likely parameters that represent the haplotypes.

- Baum-Welch is an algorithm that is related to the EM algorithm and is used to estimate those aforementioned parameters.
- To be more specific, the input is a set of haplotypes H, and we want to be able to estimate the emission probabilities (e) at all positions i at states j, as well as the transition probabilities (delta) from SNPs to SNPs at next time steps.

$$L((\delta, e); H) = \prod_{h \in H} Pr(h \mid \delta, e)$$

  - Each of the probabilities of the haplotypes
- F_h(m, i) is the probability that you emit everything from h_1 ro h_m - 1 and then you traverse through the i state, specifically the ith state in h_m.

$$Pr(h \mid \delta, e) = \sum_i F_h(m, i) \underbrace{(e_{mi} h_m + (1 - e_{mi})(1 - h_m))}_{emitprob(h_m, e_{mi})}$$

- In terms of the complexity, this algo is O(n * k^2 * m)
  - N haplotypes, m is the length of each haplotype, k states, and we need to sum over k things to get the F function.

2/25 - Week 8

- Markov model has a bunch of states per SNP. There are transition probabilities from each SNP s to each SNP s + 1. At each state, you have an emission probability as well. K is the number of states per SNP.
- One markov model for generating the haplotypes for each population. You also have transition probabilities to go from one population to another. You can get those probabilities regardless of the haplotype data. You can do it based on recombination rates, population stats, etc.
- In all, we have to estimate the transition probabilities and emission probabilities inside of each model. We have a set of haplotypes H that are binary strings of length m.
- Z is the random variable for haplotype h and SNP s, and the value is between 1 and k, and represents the state on the path. If you know the path through the states, then it is easy to write the likelihood of that data (which we saw above).

2/27 - Week 8

- Different types of regression models will compute the y approximation in different ways.
- Solving regression involves finding the parameters (a and b) that minimize the sum of least squares.
  - Finding a and b means that you've found the correct slope and y-intercept for the line that best fits the data (at least in the 2D case)
  - Ax_i + b is the prediction that the model gives for an input x_i.

- We can write the above and try to find a and b by observing some data and maximizing the likelihood.

$$Data = (x_1, y_1), \ldots, (x_n, y_n)$$

$$\log L(Data; a, b, \sigma) = -\frac{1}{2\sigma^2} \sum_i (y_i - ax_i - b)^2 - \frac{n}{2} \log(2\pi\sigma^2)$$

  -
  - The maximization of L means that we minimize that first y - pred term and thus we get

Likelihood maximized when $(a, b) = \arg\min_{a,b} \sum_i (y_i - ax_i - b)^2$

    -
- Fixed effect is where a is just a number and not associated with any distribution.
  - A better model is where you have a different a for every SNP. Problem is that know you have to estimate a million parameters for a lot less number of individuals.

3/11 - Week 10

- Short read sequencers work by generating short “reads” which are short random substrings of the 3 billion nucleotide sequence. The substrings are all of a certain length. Job is to put all of them together into the one really long 3 billion nucleotide sequence by looking at overlaps.
  - Making the overlap larger is helpful.
- Also, using the human genome is a good reference since my genome is 99.9 percent identical.
  - Find the areas of the most overlap between each location on that genome and the read that you have. In other words, slide your read across the genome and at each position count the number of mismatches, and if it is below a certain number then you can say it is a match.
- One approach to making the substring matching faster is to split your read into 3 pieces for example, and imagine that in the worst case, your 2 mutations will lie in between pieces, but there will be one piece that will be a perfect match.
  - Create a dictionary where the keys are sequences of length L/3 and the values will be a list of positions for where they occur.
  - Take a read, split into 3, and just check the 9000 positions (3000 each)
- One problem is that each sequence read can have some random errors and thus “mutations” can be added to the sequence when that mutation isn’t really there in your body.

Other Notes

Entry Exam Prep

- Exam is open notes and you can bring calculator
- P value - The basic idea is that it is a measure that tells you the significance of the results that you see for an experiment. The validity of the claim/hypothesis you make is dependent on the p-value.
    - Let's say you have an experiment and you want to test a claim (like I believe this medicine X cures the disease Y).
    - Null hypothesis is the hypothesis that the experimenter believes is incorrect - Generally it is saying that there is no statistical significance between the two variables. Alternative states there is significance.
        - Also, null is hypothesis assumes that whatever you are trying to prove did not happen
    - A small p-value (typically ≤ 0.05) indicates strong evidence against the null hypothesis, so you reject the null hypothesis.
    - The p value tells you the probability that you mistakenly reject the null. If the value is low, then it's a low probability and thus you can be confident that you haven't made a mistake.
    - Aka it is the probability of finding the observed, or more extreme, results when the null hypothesis (H0) of a study question is true.
- Type I error is the rejection of a true null hypothesis (also known as a "false positive" finding), while a type II error is failing to reject a false null hypothesis (also known as a "false negative" finding)
- The t-test is any statistical hypothesis test in which the test statistic follows a Student's t-distribution (any member of a family of continuous probability distributions that arises when estimating the mean of a normally distributed population in situations where the sample size is small and population standard deviation is unknown) under the null hypothesis.
- A chi-square statistic is one way to show a relationship between two categorical variables. The chi-squared statistic is a single number that tells you how much difference exists between your observed counts and the counts you would expect if there were no relationship at all in the population.
- Bayes Rule
    - $$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$
- Random variable is a real valued function of the experimental outcome.
    - Discrete if the number of values it can take on is finite.
- PMF tells you the probability that the random variable will take a certain value.
- Random variables and distributions

X	X Counts	p(x)	Values of X	E(x)	V(x)
Discrete uniform	Outcomes that are equally likely (finite)	$\frac{1}{b-a+1}$	$a \le x \le b$	$\frac{b+a}{2}$	$\frac{(b-a+2)(b-a)}{12}$
Binomial	Number of sucesses in n fixed trials	$\binom{n}{x}p^x(1-p)^{n-x}$	$x = 0,1,\ldots,n$	np	np(1−p)
Poisson	Number of arrivals in a fixed time period	$\frac{e^{-\lambda}\lambda^x}{x!}$	$x = 0,1,2,\ldots$	λ	λ
Geometric	Number of trials up through 1st success	$(1-p)^{x-1}p$	$x = 1,2,3,\ldots$	$\frac{1}{p}$	$\frac{1-p}{p^2}$

- Expectation is the weighted average of all the possible values of the random variable.

  We define the **expected value** (also called the **expectation** or the **mean**) of a random variable X, with PMF p_X, by

  $$\mathbf{E}[X] = \sum_x x p_X(x).$$

- Conditional PMF

  $$p_X(x) = \sum_y p_Y(y) p_{X|Y}(x \mid y).$$

- Independent random variables

  Consider two discrete random variables X and Y. We say that X and Y are independent if

  $$P\Big(X = x, Y = y\Big) = P(X = x)P(Y = y), \qquad \text{for all } x, y.$$

  In general, if two random variables are independent, then you can write

  $$P\Big(X \in A, Y \in B\Big) = P(X \in A)P(Y \in B), \qquad \text{for all sets } A \text{ and } B.$$

- Expectation Linearity

  For random variables X and Y (which may be dependent),

  $$E[X + Y] = E[X] + E[Y].$$

- Exp, var of ind variables

  $$\text{mean of } X = \mathbb{E}X = \mu_X = p_1x_1 + p_2x_2 + \ldots + p_kx_k = \sum_i X(s_i)\mathbb{P}\{s_i\}$$

  $$\text{variance of } X = \text{var}(X) = \sigma_X^2 = \sum_j p_j(x_j - \mu_X)^2 = \sum_i (X(s_i) - \mu_X)^2\mathbb{P}\{s_i\}$$

- PDF of normal distribution

$$f(x \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

where

- μ is the mean or expectation of the distribution (and also its median and mode),
- σ is the standard deviation, and
- σ^2 is the variance.

- Linear dependence can be thought of as one of the vectors being able to be represented as a sum of other ones. There are n scalars (at least one of them is non-zero) and the sum of the alphas and the vectors is 0, that's linear dependence.
  - If all the alphas are 0, then linearly independent.
  - Linearly independent if it has non-zero determinant.

$$|B| = det(B) = \begin{vmatrix} a & b & c \\ d & e & f \\ g & h & k \end{vmatrix} = a \begin{vmatrix} e & f \\ h & k \end{vmatrix} - b \begin{vmatrix} d & f \\ g & k \end{vmatrix} + c \begin{vmatrix} d & e \\ g & h \end{vmatrix}$$

    - A set of vectors $v_1, v_2, \ldots, v_n$ are linearly dependent if there are scalars $\alpha_1, \ldots, \alpha_n$ such that $\sum_{i=1}^{n} |\alpha_i| > 0$ and $\sum_{i=1}^{n} \alpha_i v_i = 0$.
  - Vector space is a collection of all the vectors that are linear combinations of all the other vectors in the space.
  - Span of a set of vectors is the vector space. Basically the vectors are some linear combination of the others.

$$span(v_1, v_2, \ldots, v_n) = \{v \mid v = \sum_i \alpha_i v_i\}$$

    - The basis of a vector space V is a minimal set of linearly independent vectors $v_1, v_2, \ldots, v_n$ such that $span(v_1, \ldots, v_n) = V$.
  - The set of *all* linear combinations of a collection of vectors $v_1, v_2, \ldots, v_r$ from R^n is called the span of $\{ v_1, v_2, \ldots, v_r \}$.
- Scalar product is the sum of the element wise products between each of the numbers in the two vectors. If scalar product between two vectors is 0, then the vectors are perpendicular. Technically, it is the length of the projection of a onto b. Finding the scalar product gives us some metric of the similarity between the vectors.
- Norm of a vector is the square root of the sum of all the elements in the vector, each squared.
- If the dimensions for a matrix are N x P, then you have N rows and P columns.
- The rank of a matrix A is the maximum number of columns of A that are linearly independent.

    - Also, rank of a matrix A is the maximum number of rows of A that are linearly independent. So it doesn't matter if you're considering rows or columns.
- Two matrices are symmetric if there are transposes of each other and they are equivalent.
    - These symmetric matrices will be seen if you have a matrix where the rows represent person 1 and the cols represent person 2 and the cell shows an interaction that is mutual.
- The inverse A of a matrix B means that AB = BA = Identity matrix
- Two vectors are orthogonal if the scalar product between them is 0.
    - Matrix is orthogonal if all its columns are orthogonal to each other and are unit vectors.
    - If A is orthogonal, then A transpose A is the identity matrix.
- Lambda is an eigenvalue of matrix A if there is an eigenvector x such that Ax = lambda*A.
    - For a symmetric matrix, the maximum and minimum eigenvalues can be found by solving:

$$\min_{\|x\|=1} x^t Ax$$

$$\max_{\|x\|=1} x^t Ax$$

- A symmetric matrix A is positive definite if x' Ax > 0 for every vector x > 0.
- Trace of matrix is sum of the elements on the diagonal. It is also equivalent to the sum of the eigenvalues.
- Gradient is a vector of partial derivatives of a function with respect to the different variables.
- Positive Semidefinite Matrices

The following are equivalent definitions of PD matrices (for symmetric matrix A):

1. $x^t Ax \geq 0$ **for every vector** $x \neq 0$.
2. **All the eigenvalues of A are nonnegative.**
3. **There exist a set of vectors $v_1,\ldots,v_n$, such that** $\langle v_i, v_j \rangle = a_{ij}$

- Other matrix properties

- A negative definite matrix A is such that –A is positive definite.
- If A,B are positive (semi)definite, then A+B is positive definite.
- If A is positive (semi)definite, then αA is also positive (semi)definite for α>0.
- If A is positive definite then A is invertible.

- Hessian is a matrix of all the partial second derivatives. It's a matrix now because you have two variables that you're taking the derivative with respect to.
- Taylor expansion
  - $$f(x+\Delta x) \approx f(x) + f'(x)\Delta x + \frac{f''(x)}{2}\Delta x^2$$
- PDF
  - $$\Pr(a < X < b) = \int_a^b f(x)dx$$
- CDF
  - $$F_X(x) = \Pr(X \leq x) = \int_{-\infty}^{x} f(x)dx$$
- Binomial Dist
  - $$f_X(k) = \Pr(X = k) = \binom{n}{k} p^k (1-p)^{n-k}$$
- Normal Dist. (Standard is mu of 0 and sigma of 1)
  - $$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$
    - If two distributions are normally distributed, then their linear combination is also normally distributed.
- The average of n random variables will get closer and closer to a normal distribution as n goes to infinity.
- Independence of random variables

  Two random variables X, Y are independent if for every x, y we have

  - $$\Pr(X \leq x, Y \leq y) = F_X(x)F_Y(y)$$
- Expectation - Expected value of a random variable. For independent random variables, the expectation follows the linear and multiplication rules
  - $$E[X] = \int_{-\infty}^{\infty} xf(x)dx.$$

  For every two independent variables

  $$E[X+Y] = E[X] + E[Y]$$
  $$E[\alpha X] = \alpha E[X]$$

  For independent random variables

  $$E[XY] = E[X]E[Y]$$

- Variance and covariance

  The variance of a random variable X is

  $$Var[X] = E[(X - E[X])^2] = E[X^2] - (E[X])^2$$

  For two random variables, X, Y,

  $$Cov(X, Y) = E[XY] - E[X]E[Y]$$

  - $Var(X) = Cov(X, X)$
  - If X, Y are independent then $Cov(X, Y) = 0$
  - $Var(X + Y) = E\big((X + Y)^2\big) - E(X + Y)^2 = E[X^2 + 2XY + Y^2] - (E[X] + E[Y])^2 = E[X^2] + E[Y^2] + 2E[XY] - E[X]^2 - E[Y]^2 - 2E[X]E[Y] = Var(X) + Var(Y) + 2Cov(X, Y)$
- Law of Total Prob:

  $$\Pr(B) = \sum_i \Pr(A_i \cap B) = \sum_i \Pr(B \mid A_i)\Pr(A_i)$$
- Taylor Expansion

  $$f(x) = \sum_{j=0}^{\infty} \frac{f^{(j)}(c)}{j!}(x - c)^j$$

  - In one dimension:

    $$f(x + \Delta x) \approx f(x) + f'(x)\Delta x + \frac{f''(x)}{2}\Delta x^2$$

  - In many dimensions:

    $$f(x + \Delta x) \approx f(x) + \nabla f(x)\Delta x + \frac{1}{2}\Delta x^t H(x)\Delta x$$

  - Useful observation: $\log(1 + \Delta x) \approx \Delta x - \frac{\Delta x^2}{2}$

$$f(x+\Delta x) \approx f(x) + \nabla f(x)\Delta x + \frac{1}{2}\Delta x^t H(x)\Delta x$$

If H(x) is positive definite, and $\nabla f(x) = 0$ then x is a local minimum.

If H(x) is negative definite, and $\nabla f(x) = 0$ then x is a local maximum.

If H(x) is negative definite for every x, then f is concave with a unique maximum point.

If H(x) is positive definite for every x, then f is convex with a unique minimum point.

- Newton's Method

$$f(x+\Delta x) \approx f(x) + f'(x)\Delta x + f''(x)\frac{\Delta x^2}{2}$$

f is maximized when

$$f'(x) + f''(x)\Delta x = 0$$

f is maximized when

$$\Delta x = -\frac{f'(x)}{f''(x)}$$

- Markov Inequality

Let X be a random variable with nonnegative range (i.e., $X \geq 0$). Then for every $a > 0$,

$$\Pr(X \geq a) \leq \frac{E[X]}{a}$$

- Cheby's Inequality

Let X be a random variable. Then for every $a > 0$,

$$\Pr(|X - E[X]| \geq a) \leq \frac{Var(X)}{a^2}$$