CM124/224, Fall 2023
Problem Set 3: Mixture models and dimensionality reduction
Due Dec 1, 2023 at 11:59pm PST

1 Course Challenge 2: Phenotype Prediction

The second challenge of the course will take place from Nov 20th to Dec 4th. The following link will take you to the second challenge (Link to the second challenge). Please register yourself to the competition by clicking `Participate > Register`. Similar to challenge 1, we have also provided detailed documentation under `Learn the Details` section and link to data under `Participate > Get Data` section. You may experiment with various models to achieve the most accurate predictions you can. **Unlike the first challenge, please keep track of the coding implementation used for each submission. After the challenge has closed, we will be asking you to submit your code that accomplished the highest score on BruinLearn.**

2 Mixture model [15 pts]

We will define a mixture model for genetic data that we will use to infer an individual's genetic ancestry (this is the unsupervised version of the challenge problem that you have been working on).

We are given the genotype of an individual for m SNPs: $(X_1, \ldots, X_m)$. We assume that $X_j \in \{0, 1\}$ (we simplify this model so that each X_j takes one of two values instead of the usual one of three values of $\{0, 1, 2\}$). We are told that the individual belongs to one of $K = 2$ populations and we would like to be able to identify the population. The population of the individual is denoted by the random variable Z and since there are only two populations, we have $Z \in \{0, 1\}$.

For population 0, we have information on how frequently the allele encoded as 1 is seen at each SNP j for individuals that belong to this population. We summarize this by a vector $\boldsymbol{f}_0 = (f_{0,1}, \ldots, f_{0,m})$ (we term this the allele frequency vector). Likewise, we have the allele frequency vector for population 1: $\boldsymbol{f}_1 = (f_{1,1}, \ldots, f_{1,m})$. Intuitively, if we have $f_{0,1} = 1$ and $f_{1,1} = 0$, then an individual from population 0 will always have the genotype value 1 at SNP 1 while an individual from population 1 will always have the genotype value 0 at the same SNP. In this case, we can label the population of the individual by simply examining their genotype at SNP 1. In general, one SNP alone might not be very informative of the population label.

So we model the genotype as follows. If this individual belongs to population 0, the genotype at SNP j (which takes values in 0 or 1) is drawn independently from a Bernoulli distribution with parameter $f_{0,j}$ (in words, the probability that the genotype is 1 is determined by the frequency of the 1 allele in the population). Likewise, if this individual belongs to population 1, the genotype at SNP j is drawn independently according to a Bernoulli distribution with parameter $f_{1,j}$.

(a) Write the likelihood of observing the genotype given that the individual belongs to population 0: $P(x_1, \ldots, x_m | Z = 0)$. Express your answer in terms of $x_1, \ldots, x_m$ and $\boldsymbol{f}_0$, $\boldsymbol{f}_1$ (Hint: the probability mass function for a Bernoulli distribution with parameter θ: $p(x) = \theta^x (1-\theta)^{1-x}$).

Solution: $p(\boldsymbol{x}_{1:m} | Z = 0) = \prod_{j=1}^{m} f_{0j}^{x_j} (1 - f_{0j})^{1-x_j}$
Although not asked, note the following:
$p(\boldsymbol{x}_{1:m} | Z = 1) = \prod_{j=1}^{m} f_{1j}^{x_j} (1 - f_{1j})^{1-x_j}$

(b) We observe the genotype at 4 SNPs: $(0, 1, 1, 0)$ and allele frequencies $\boldsymbol{f}_0 = (0.2, 0.5, 0.4, 0.4)$, $\boldsymbol{f}_1 = (0.1, 0.3, 0.3, 0.5)$. Substitute these numbers in the expressions above to compute the likelihoods $P(x_1, \ldots, x_4|Z = 0)$ and $P(x_1, \ldots, x_4|Z = 1)$. **Solution:**

$p(\boldsymbol{x}_{1:4}|Z = 0) = (1 - 0.2)(0.5)(0.4)(1 - 0.4) = 0.096$
$p(\boldsymbol{x}_{1:4}|Z = 1) = (1 - 0.1)(0.3)(0.3)(1 - 0.5) = 0.0405$

(c) One approach to assign an individual to a population is to pick the population that has the higher likelihood. Which of the populations has a higher likelihood (this is the maximum likelihood estimate of the population assignment)? **Solution:**

$p(\boldsymbol{x}_{1:4}|Z = 0) = 0.096 > 0.0405 = p(\boldsymbol{x}_{1:4}|Z = 1)$
Population 0 has a higher likelihood.

(d) We have some domain knowledge about which population the individual might belong to, even before we analyze their phenotype. To incorporate this prior knowledge, we will now consider a Bayesian model where the population membership Z is itself a random variable. Since Z takes one of two possible values, Z can be modeled by a Bernoulli random variable with parameter π. Specifically, $P(Z = 0) = 1 - \pi$ and $P(Z = 1) = \pi$. Using the likelihood and the prior, write the posterior probability: $P(Z = 1|x_1, \ldots, x_m)$. Express your answer in terms of $x_1, \ldots, x_m$, $\boldsymbol{f}_0$, $\boldsymbol{f}_1$, and π. **Solution:**

$p(Z = 1|\boldsymbol{x}_{1:m}) = \frac{p(\boldsymbol{x}_{1:m}|Z=1)p(Z=1)}{p(\boldsymbol{x}_{1:m}|Z=1)p(Z=1)+p(\boldsymbol{x}_{1:m}|Z=0)p(Z=0)}$
$= \frac{\pi(\prod_{j=1}^{m} f_{1j}^{x_j}(1-f_{1j})^{1-x_j})}{\pi(\prod_{j=1}^{m} f_{1j}^{x_j}(1-f_{1j})^{1-x_j})+(1-\pi)(\prod_{j=1}^{m} f_{0j}^{x_j}(1-f_{0j})^{1-x_j})}$

(e) An alternate approach to assign an individual to a population is to pick the population that has the higher posterior probability. If we choose the prior probability on the population to be $Z \sim Ber(\pi)$ where $\pi = 0.8$, compute the posterior probability $P(Z = 1|x_1, \ldots, x_4)$. **Solution:**

$p(Z = 1|\boldsymbol{x}_{1:4}) = \frac{0.0405*0.8}{0.0405*0.8+0.096*0.2} = 0.6279$
$p(Z = 0|\boldsymbol{x}_{1:4}) = 1 - p(Z = 1|\boldsymbol{x}_{1:4}) = 0.3721$

(f) Which of the populations has the higher posterior probability (this is the MAP estimate of the population assignment)? **Solution:**

$p(Z = 1|\boldsymbol{x}_{1:4}) = 0.6279 > 0.3721 = p(Z = 0|\boldsymbol{x}_{1:4})$
Population 1 has the higher posterior probability.

(g) Comment on any differences that you observe between the MLE and MAP estimates. **Solution:**

MLE estimates can be thought as MAP estimates with uniform posterior probability. Therefore, MAP estimate bias towards the population 1, indicating data is highly belonging to population 1 regardless of the data. It is only valid in our case since $P(Z = 1) > P(Z = 0)$. Our MLE estimate concludes that data belongs to population 0, whereas MAP estimate suggests that data belongs to population 1.

3 The EM algorithm[15 pts]

The model in the previous question assumed that the parameters $\boldsymbol{\theta} = (\pi, \boldsymbol{f}_0, \boldsymbol{f}_1)$ are known. We will now try to estimate these parameters from data.

For simplicity, we observe data at m SNPs across n individuals. For each individual i, let Z_i denote the population membership and $X_{i,j}$ denote the genotype at SNP j for this individual.

We can write our model as:

$$\begin{aligned} Z_i|\pi &\overset{iid}{\sim} \text{Ber}(\pi) \\ X_{i,j}|Z_i = 0, \boldsymbol{f}_0, \boldsymbol{f}_1 &\sim \text{Ber}(f_{0,j}) \\ X_{i,j}|Z_i = 1, \boldsymbol{f}_0, \boldsymbol{f}_1 &\sim \text{Ber}(f_{1,j}) \end{aligned}$$

(a) The complete data log likelihood for this model can be written as

$$\sum_{i=1}^{n} \log P(\boldsymbol{x}_{i,1:m}, z_i|\theta) = \sum_{i=1}^{n} \log \left(\left((1-\pi) \prod_{j=1}^{m} Ber(x_{i,j}|f_{0,j}) \right)^{1-z_i} \times \left(\pi \prod_{j=1}^{m} Ber(x_{i,j}|f_{1,j}) \right)^{z_i} \right)$$

Write the expected complete data log-likelihood: $Q(\boldsymbol{\theta}; \boldsymbol{\theta}^{(t)})$. Express your answer in terms of the soft assignments of individuals to populations based on the current parameter estimates $\boldsymbol{\theta}^{(t)}$: $r_i^{(t)} = P(Z_i = 1|\boldsymbol{x}_{i,1:m}, \boldsymbol{\theta}^{(t)})$. The notation $\boldsymbol{x}_{i,1:m}$ denotes the genotype vector for individual i at SNPs $1, \ldots, m$. The notation $Ber(x|p)$ denotes the Bernoulli probability mass function: $p^x(1-p)^{(1-x)}$.

Solution:

$$\log(p(D_{XZ}; \theta)) = \sum_{i=1}^{n} \log(p(\boldsymbol{x}_{i,1:m}, z_i|\theta)) = \sum_{i=1}^{n} \log(p(\boldsymbol{x}_{i,1:m}|z_i, \theta)) + \log(p(z_i|\theta))$$

where,

$$\log(p(\boldsymbol{x}_{i,1:m}|z_i, \theta)) = (1 - z_i) \log(\prod_{j=1}^{m} f_{0j}^{x_{ij}} (1 - f_{0j})^{1-x_{ij}}) + z_i \log(\prod_{j=1}^{m} f_{1j}^{x_{ij}} (1 - f_{1j})^{1-x_{ij}})$$

$$= (1 - z_i)(\sum_{j=1}^{m} x_{ij} \log(f_{0j}) + (1 - x_{ij}) \log(1 - f_{0j})) + z_i(\sum_{j=1}^{m} x_{ij} \log(f_{1j}) + (1 - x_{ij}) \log(1 - f_{1j}))$$

and

$$\log(p(z_i|\theta)) = z_i \log(\pi) + (1 - z_i) \log(1 - \pi)$$

Then, to get expected complete data log-likelihood take conditional expectation.

$$Q(\boldsymbol{\theta}; \boldsymbol{\theta}^{(t)}) = \mathbb{E}_{\theta^{(t)}}[\log(p(D_{XZ}; \theta] = \sum_{i=1}^{n} \mathbb{E}_{\theta^{(t)}}[\log(p(\boldsymbol{x}_{i,1:m})|z_i, \theta)] + \mathbb{E}_{\theta^{(t)}}[\log(p(z_i|\theta))]$$

where,

$$\mathbb{E}_{\theta^{(t)}}[\log(p(\boldsymbol{x}_{i,1:m})|z_i, \theta)] =$$

$$= (1-\mathbb{E}_{\theta^{(t)}}[z_i])(\sum_{j=1}^{m} x_{ij}\log(f_{0j})+(1-x_{ij})\log(1-f_{0j}))+\mathbb{E}_{\theta^{(t)}}[z_i](\sum_{j=1}^{m} x_{ij}\log(f_{1j})+(1-x_{ij})\log(1-f_{1j}))$$

$$= (1-r_i)(\sum_{j=1}^{m} x_{ij}\log(f_{0j})+(1-x_{ij})\log(1-f_{0j}))+r_i(\sum_{j=1}^{m} x_{ij}\log(f_{1j})+(1-x_{ij})\log(1-f_{1j}))$$

and,

$$\mathbb{E}_{\theta^{(t)}}[\log(p(z_i|\theta))] = \mathbb{E}_{\theta^{(t)}}[z_i]\log(\pi) + (1-\mathbb{E}_{\theta^{(t)}}[z_i]\log(1-\pi)$$
$$= r_i\log(\pi) + (1-r_i)\log(1-\pi)$$

After combining two terms we obtain expected complete data log-likeli,

$$Q(\boldsymbol{\theta};\boldsymbol{\theta}^{(t)}) = \sum_{i=1}^{n}((1-r_i)(\sum_{j=1}^{m} x_{ij}\log(f_{0j})+(1-x_{ij})\log(1-f_{0j}))+r_i(\sum_{j=1}^{m} x_{ij}\log(f_{1j})+(1-x_{ij})\log(1-f_{1j}))$$
$$+r_i\log(\pi) + (1-r_i)\log(1-\pi))$$

(b) Use the results from question 1 to write the expression for $r_i^{(t)}$ in terms of $\boldsymbol{x}_{i,1:m}$, $\pi^{(t)}$, $\boldsymbol{f}_0^{(t)}$, and $\boldsymbol{f}_1^{(t)}$. **Solution:**

$$r_i^{(t)} = \frac{\pi^{(t)}(\prod_{j=1}^{m} f_{1j}^{(t)^{x_{ij}}}(1-f_{1j}^{(t)})^{1-x_{ij}})}{\pi(\prod_{j=1}^{m} f_{1j}^{(t)^{x_{ij}}}(1-f_{1j}^{(t)})^{1-x_{ij}}) + (1-\pi)(\prod_{j=1}^{m} f_{0j}^{(t)^{x_{ij}}}(1-f_{0j}^{(t)})^{1-x_{ij}})}$$

(c) We would like to obtain the MLE of $\boldsymbol{\theta}$. Derive the M-step updates for ancestry proportion π and the allele frequency of feature j for both populations: $f_{0,j}$ and $f_{1,j}$. Express your answer in terms of $r_i^{(t)}$ and $\boldsymbol{x}_{i,1:m}$. (Hint: since we assume SNPs are independent of each other, optimization on parameters related to feature j only needs to take into account terms in $Q(\boldsymbol{\theta};\boldsymbol{\theta}^{(t)})$ that are associated with that particular feature). **Solution:**

M-step Update for π:

$$\frac{\partial Q\left(\theta;\theta^{(t)}\right)}{\partial\pi} = \sum_{i=1}^{n}(\frac{r_i}{\pi} - \frac{1-r_i}{1-\pi}) = 0$$

$$\sum_{i=1}^{n}(\frac{r_i}{\pi}) = \sum_{i=1}^{n}(\frac{1-r_i}{1-\pi})$$

$$\frac{1-\pi}{\pi} = \frac{\sum_{i=1}^{n} 1-r_i}{\sum_{i=1}^{n} r_i}$$

$$\frac{1}{\pi} = \frac{n+\sum_{i=1}^{n} -r_i}{\sum_{i=1}^{n} r_i} + 1$$

$$\pi = \frac{\sum_{i=1}^{n} r_i}{n}$$

I have omitted (t) term in derivation, therefore I need to add (t) at the result.

$$\pi = \frac{\sum_{i=1}^{n} r_i^{(t)}}{n}$$

M-step Update for f_{0j}:

$$\frac{\partial Q\left(\theta;\theta^{(t)}\right)}{\partial f_{0j}} = \sum_{i=1}^{n}(1-r_i)(\frac{x_{ij}}{f_{0j}} - \frac{1-x_{ij}}{1-f_{0j}}) = 0$$

$$\sum_{i=1}^{n}(\frac{(1-r_i)x_{ij}}{f_{0j}}) = \sum_{i=1}^{n}(\frac{(1-r_i)(1-x_{ij})}{1-f_{0j}})$$

$$\frac{1-f_{0j}}{f_{0j}} = \frac{\sum_{i=1}^{n}(1-r_i)(1-x_{ij})}{\sum_{i=1}^{n}(1-r_i)x_{ij}}$$

$$\frac{1}{f_{0j}} = \frac{\sum_{i=1}^{n}(1-r_i)(1-x_{ij})}{\sum_{i=1}^{n}(1-r_i)x_{ij}} + 1$$

$$f_{0j} = \frac{\sum_{i=1}^{n}(1-r_i)x_{ij}}{\sum_{i=1}^{n}(1-r_i)}$$

I have omitted (t) term in derivation, therefore I need to add (t) at the result.

$$f_{0j} = \frac{\sum_{i=1}^{n}(1-r_i^{(t)})x_{ij}}{\sum_{i=1}^{n}(1-r_i^{(t)})}$$

M-step Update for f_{1j}:

$$\frac{\partial Q\left(\theta;\theta^{(t)}\right)}{\partial f_{1j}} = \sum_{i=1}^{n}r_i(\frac{x_{ij}}{f_{1j}} - \frac{1-x_{ij}}{1-f_{1j}}) = 0$$

$$\sum_{i=1}^{n}(\frac{(r_i)(x_{ij})}{f_{1j}}) = \sum_{i=1}^{n}(\frac{(r_i)(1-x_{ij})}{1-f_{1j}})$$

$$\frac{1-f_{1j}}{f_{1j}} = \frac{\sum_{i=1}^{n}(r_i)(1-x_{ij})}{\sum_{i=1}^{n}(r_i)x_{ij}}$$

$$\frac{1}{f_{1j}} = \frac{\sum_{i=1}^{n}(r_i)(1-x_{ij})}{\sum_{i=1}^{n}(r_i)x_{ij}} + 1$$

$$f_{1j} = \frac{\sum_{i=1}^{n}(r_i)(x_{ij})}{\sum_{i=1}^{n}(r_i)}$$

I have omitted (t) term in derivation, therefore I need to add (t) at the result.

$$f_{1j} = \frac{\sum_{i=1}^{n}(r_i^{(t)})(x_{ij})}{\sum_{i=1}^{n}(r_i^{(t)})}$$

4 Principal Component Analysis [15 pts]

We will apply Principal Component Analysis on the genetic data of 1,092 (real) individuals from the 1000 Genomes Project.

You are given the genotype data containing $M = 13,237$ SNPs for $N = 1,092$ individuals of African, East Asian, (Admixed) American and European descent. Both the SNP data and the true population labels are given.

If using R, please set the seed for the random number generator to 0 in the beginning of your script with set.seed(0). If using Python sklearn, use np.random.seed(0).

(a) Use the prcomp function in R (Documentation: prcomp) or the sklearn.decomposition.PCA function in Python (Documentation: sklearn PCA) to run PCA on this dataset and plot PC1 against PC2 in a scatter plot. Color your points based on the population the individual comes from.

Solution:

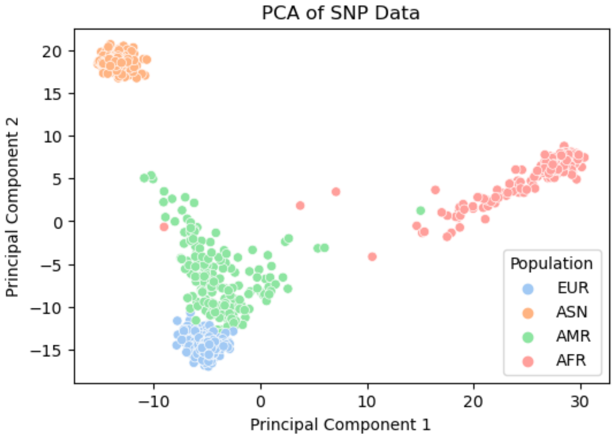

Figure 1: PCA of SNP Data

(b) Briefly comment on why you think the first two components of PCA exhibit clustering by population. Why can PCA successfully capture population membership?

Solution: First two components capture the most variance in the data compared to other components. The most varied component is their ancestry, therefore PCA capture population heritage.

(c) Use the kmeans function in R or Python (sklearn) to run the k-means algorithm on the SNP data (*not* the PCs) with K=4 and nstart=5 (in R) or n_init=5 (in Python) (*i.e.* use 5 different initializations since kmeans is not guaranteed to converge to a global optimum). Now rename all the clusters you have obtained by size: the largest cluster by number of individuals should be named Cluster1 and the smallest cluster should be named Cluster4. Generate the same plot as in part (a) (PC1 vs. PC2), but this time color your points based on cluster assignments (*i.e.* was an individual assigned to cluster 1, 2, 3 or 4) instead of the population labels. (The location of the points on the PC1 vs. PC2 plot should not change, but their colors

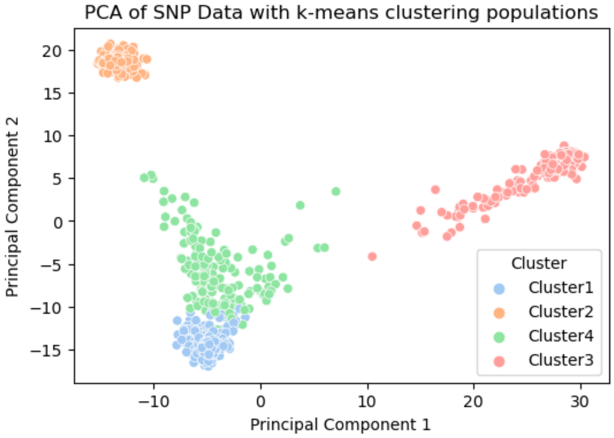

Figure 2: PCA of SNP Data with k-means clustering populations

should now indicate cluster membership instead of true population labels.)
Resources: R Kmeans Documentation, Python Kmeans Documentation

Solution:

(d) Match clusters to population labels by inspection (*e.g.* "Cluster 1 most closely resembles the ASN population.") What fraction of the cluster assignments agree with the true population labels?

Solution: I have set np.random.seed(0), and random_state of PCA and K-means to 0. In result, I have obtained 97.1611721%.