#homework

- A: In this question you will extend the HMM in https://stephens999.github.io/fiveMinuteStats/hmm.html to treat the means of the two states as unknown and to be estimated. (Note that the true values of the means in the simulation are 1 and 2).
  - Derive and implement the EM algorithm for estimating the means.
  - Check your implementation by running it on the example and seeing that the log-likelihood is increasing. [Hint: note that the forwards algorithm gives you the likelihood.]
  - Try running the EM algorithm multiple times from different starting points. Does it get stuck in local optima of the log-likelihood?
- B: Complete the exercise in https://stephens999.github.io/stat34800/hmm_exercise.html
- C: Spatial Gaussian Processes
  - Consider the data from http://journals.plos.org/plosbiology/article?id=10.1371/journal.pbio.0030339, available at https://github.com/stephens999/hgen48600/tree/master/data/CCR5, which you can read into R using code from https://stephens999.github.io/hgen48600/ccr5.html
  - These consist of latitude, longitude, and an allele frequency at each location. We will model these data as a Gaussian process. Since allele frequency lies in [0,1] start by $\hat{f}\in[0,1]$ using the transformation $x = \log(\hat{f}/(1-\hat{f}))$. (Here $\hat{f}$ is the estimated frequency in the code above.) We will let y denote locations in space (latitude, longitude) and $(-\infty,\infty)$ $x(\cdot)$ denote the allele frequency varying as a function of space, so $x(y) \in \cancel{[0,1]}$ is the allele frequency at location y. We will model $x(\cdot)$ as a Gaussian process, with constant mean $\mu = m$ and squared exponential covariance function of the form $a_1 \exp(-(d/a_2)^2)$.
    - Hence, $a = (a_1, a_2)$ and the mean m are the parameters to be estimated.
  - Write a function to compute the covariance matrix for $x^{\text{obs}} := (x(y_1), ..., x(y_r))$ given a value of a. Here $y_1, \ldots, y_r$ are the locations at which you have observations in the dataset. Try a few values of a and check that the resulting covariance matrix is valid - that is, it is positive semi-definite. (The best way to check that a covariance is positive semi-definite is to attempt to perform a cholesky decomposition: if the decomposition succeeds then the matrix must be PSD).
  - Write a function to compute the log-likelihood for the data x^{obs}, given a, m. [Here we assume the mean is constant across the whole region, so m is the same at every location].

  - The model here is that \(x^{\text{obs}} | m, a \sim N_r(\mu, \Sigma)\) where \(\Sigma=\Sigma(a)\) is the function of \(a\) that you coded above and \(\mu=rep(m,r)\). So your likelihood just involves computing a multivariate normal density. You can use the R function mvtnorm::dmvnorm (with log=TRUE)
- Try using the R function optim (or another approach if you prefer) to optimize the likelihood numerically over a, m. (I found it seemed to work OK, in that it gave similar answers from different starting points).
- Now we are going to try deleting each of the observed data points in turn and "impute" its value using our model. This process is sometimes known as Kriging.
  - Let $X = (X_1, \dots, X_r)$ be r-variate normal with mean μ and variance covariance Σ. Write a function to compute the conditional expectation of X_1 given $X_2, \dots, X_r$. [This is an application of standard results for the conditional mean of a Gaussian from, e.g. https://en.wikipedia.org/wiki/Multivariate_normal_distribution#Conditional_distributions]
  - Apply this function to compute $\mathrm{E}(x(y_1)|x(y_2), ..., x(y_r))$. Notice that this expectation ends up being a weighted linear combination of the other datapoints. Intuitively, if allele frequencies vary smoothly in space then this weighted linear combination should weight the nearby data points more. Does it?
  - Repeat this for each of the r datapoints.
  - How does the accuracy of this imputation scheme compare with just using the mean of the other datapoints to impute each datapoint?