Applied Probabilistic Machine Learning
Project: Clustering of pathogen genomes

Hugues Richard - RichardH@rki.de - hugues.richard@sorbonne-universite.fr

Winter 2023

Accompanying information sheet to the Jupyter notebook, you just have to answer to the questions on the notebook

This project is an application of clustering to the problem of molecular epidemiology. Nowadays, genetic sequence data is routinely used to identify the pathogen strain a patient was infected with and inform on his treatment (for instance to predict antimicrobial resistance).

However the genomic information of the pathogen can also be used to identify clusters of transmission and analyse outbreak. Molecular epidemiology directly makes use of the fact that a pathogen will accumulate mutations between each transmission event. Thus, contrary to traditional epidemiology, where transmission is infered by tracing chains of partners, molecular data allows to infer epidemiological connection between infected individuals by directly considering the genetic similarity between pathogen sequences.

In this project, we will design and use a mixture model that can cluster genomic sequences. The part detailed in this exercise sheet present the specification of the model and its testing it on simulated sequences.

1 Description of the model

We observe genomic sequences $\mathbf{y}_1, \dots, \mathbf{y}_n$ of the same length ℓ. Each letter $y_{i,j}$ is a nucleotide in $\Sigma = \{A, C, G, T\}$. Our goal will be to group the sequences that have lower genetic distance together, and report a representative $\mathbf{c}_k$ for each group. We want to take into account the fact that the sequences can mutate between each transmission event.
A simple way to define a cluster can be done in the following way:

- Each cluster is defined by its cluster representative $\mathbf{c}_k$ which is a genomic sequence of length ℓ: $\mathbf{c}_k = (c_{k,j})_{j=1:\ell}$
- The sequences belonging to a cluster k have a probability μ_k of mutating at each given position. The mutations are supposed independent and each nucleotide has the same probability to mutate. For instance, if the sequence y belongs to cluster k.

$$\mathbb{P}(y_j = A \mid c_j = A) = (1 - \mu_k) \quad \text{and} \quad \mathbb{P}(y_j \neq A \mid c_j = A) = \mu_k$$

(note that the probability for having a given nucleotide is $\mu_k/3$)

- Finally we use the variable Z_i to denote the class membership of sequence i and note as π_k the class priors

$$\pi_k = \mathbb{P}(Z_i = k)$$

The set of all parameters Θ we wish to estimate is:

$$\Theta = \Big\{(\pi_k, \mu_k, (c_{k,j})_{j=1:\ell}), k = 1 : K\Big\}$$

Once the model is specified (as we did now), we will need to compute some quantities to be able to apply the EM algorithm.

1. Compute the probability of a sample given its cluster assignment.
2. Compute the posterior probability of a sample to belong to each cluster (the q value, needed for the E-step)
3. Compute the likelihood of the data given the parameters (optional).
4. Compute the new estimates of the parameters (M step).

Note that a similar model for binary values (mixture of Bernoulli distributions) is presented in detail in Bishop[1], section 9.3.3.

The project is divided into three part: (1) evaluate the likelihood of sequences, (2) apply the EM algorithm on exemplary sequences, (3) simulate sequences.

For the exercise, we will use relatively short sequences of length $\ell = 100$ (as a comparison the genome of SARS-CoV2 is around 30,000 bp long, HIV is 9,200 bp long, and Ebola is around 18,900 bp long).

2 Simulations and samples likelihood

1. The first step with a model is to understand how to generate samples from it. Here, to simulate a sequence, we go along the representative sequence c, and at each position, we mutate it with a probability μ. This way of simulating is easy but not the fastest, an alternative could be to directly sample the vector of positions that should be modified.

2. Verify that the probability of a sequence $\mathbf{y}$ given it belongs to a cluster $\mathbf{c}$ can be written as:

$$\begin{aligned}\mathbb{P}(\mathbf{y} \mid \mathbf{c}, \mu) &= \prod_{j=1}^{\ell} (1-\mu)^{\mathbf{I}_{\{y_j = c_j\}}} \cdot \frac{\mu}{3}^{\mathbf{I}_{\{y_j \neq c_j\}}} \\ &= (1-\mu)^{\ell - n_k} \cdot \left(\frac{\mu}{3}\right)^{n_k}\end{aligned}$$

   where n_k is the number of position in the sequence where $\mathbf{y}$ and $\mathbf{c}$ are different (in other words, the Hamming distance between the two sequences).

3. Write a function `LogProbabilities` that takes as input a (n, ℓ) matrix of sequences, a (K, ℓ) matrix of cluster representative and the parameters μ_k and returns the (n, K) matrix of log probabilities $\big(\log \mathbb{P}(y_i \mid c_k)\big)_{i=1:n, k=1:k}$

[1] Pattern Recognition and Machine Learning, Christopher M. Bishop, Springer, available here

4. Deduce a fonction `ClassPosterior` that takes the same arguments as the previous function and returns the posterior probability the sample to belong to each of the classes.

$$\begin{aligned} Q_i^{(t+1)}(k) &= \mathbb{P}(Z_i = k \mid \mathbf{y}_i, \Theta^{(t)}) \\ &\propto \mathbb{P}(\mathbf{y}_i \mid Z_i = k, \Theta^{(t)}) \cdot \pi_k \end{aligned}$$

5. Write a function `Loglikelihood` that, given sequences $\mathbf{y}_i$, cluster representatives $\mathbf{c}_k$, and mutation rates μ_k, returns a vector with the log likelihood of each of the sequences $(\log \mathbb{P}(\mathbf{y}_i \mid \Theta))_{i=1:n}$. (Note that this is the marginal log probability, summed over all possible class allocations of the sequence $\mathbf{y}_i$)

3 EM algorithm

1. Implement an `Mstep` function that computes the update of the parameters $\Theta^{(t)}$ in the following way:

$$\begin{aligned} N_k^{(t)} &= \sum_{i=1}^{n} Q_i^{(t)}(k) \quad \text{(effective size of cluster } k\text{)} \\ p_{k,j}^{(t)}(\sigma) &= \frac{\sum_{i=1}^{n} Q_i^{(t)}(k) \cdot \mathbf{I}_{\{y_{i,j}=\sigma\}}}{N_k^{(t)}} \quad \text{(profile of cluster representative } k\text{)} \\ c_{k,j}^{(t)} &= \arg\max_{\sigma \in \Sigma} p_{k,j}^{(t)}(\sigma) \\ \mu_k^{(t)} &= \frac{\sum_{i=1}^{n} Q_i^{(t)}(k) \cdot \sum_{j=1}^{\ell} \mathbf{I}_{\{y_{i,j} \neq c_{k,j}^{(t)}\}}}{\ell \cdot N_k^{(t)}} \end{aligned}$$

2. Implement the EM algorithm with an `EM` function. As a stopping criterion, you will monitor that the cluster representatives $\mathbf{c}_k$ do not change anymore (optionally you can monitor the relative change in the log likelihood).