TU München Summersemester 2020
Prof. Dr. Dr. Fabian Theis, Dr. Tingying Peng

MA4802: Statistical learning

Sheet 1: Introduction: An warm-up of Python and Machine learning

1 K-nearest-neighbor classifier

Get familiar with the "Python + Google Colab" framework used in the lecture. You can just follow the example "IntroToPython.ipynb", which deals with a $k-$nearest-neighbor classifier applied to data for Iris flowers (which is a well-known example in Machine Learning). The solutions to other exercises should be structured and documented similarly to the "IntroToPython" notebook.

(a)

(b)

(c)

2 Random forest

Random forests is a supervised learning algorithm. It can be used both for classification and regression. It is also the most flexible and easy to use algorithm. A forest is comprised of trees. It is said that the more trees it has, the more robust a forest is. Random forests creates decision trees on randomly selected data samples, gets prediction from each tree and selects the best solution by means of voting. It also provides a pretty good indicator of the feature importance. Please refer to wikipedia page, `https://en.wikipedia.org/wiki/Random_forest` for further reading.

In this exercise, we use a Random forest classifier applied to data for Iris flowers. In additional to achieve good classification results, we can retrieve feature importance through random forest, which adds interpretability to the classifier why it makes such a prediction.

3 Optional

In this example, we are going to consider a simple model of molecular evolution, where each nucleic acid in the DNA-sequence evolves independently according to a dynamic

model that depends upon the time between observations.

Sequences evolve according to a complicated interaction of random mutations and selection, where the random mutations can be single nucleotide substitutions, deletions or insertions, or higher order events like inversions or crossovers. We will only consider the substitution process. Thus we consider two DNA sequences that are evolutionary related via a number of nucleotide substitutions. We will regard each nucleotide position as unrelated to each other, meaning that the substitution processes at each position are independent.

We consider a data set obtained for the H strain of the Hepatitis C virus (HCV) (Ogata et al., 1991) and study its evolution. A patient was infected by HCV in 1977 and remained infected at least until 1990 - for a period of 13 years. In 1990 a research group sequenced three segments of the HCV genome obtained from plasma collected in 1977 as well as in 1990. The three segments, denoted segment A, B and C, were all directly alignable without the need to introduce insertions or deletions. The lengths of the three segments are 2610 (A), 1284 (B) and 1029 (C) respectively.

Read the following data set into Python

```
import pandas as pd
pd.read_csv("HepCevol.txt",delimiter=' ').
```

The file contains the position for the first 24 mutations for segment A out of the total of 78 mutations on this segment. Two nucleic acids (nucleotide.77 denoted by X and nucleotide.90 denoted by Y) take values from the sample space $\{A, C, G, T\} \times \{A, C, G, T\}$ and their evolution is modelled as a random process (i.e., X, Y are random variables).

We model the substitution (evolution) process in the DNA sequence in a continuous-time fashion using Jukes-Cantor model, where the transition probabilities of x mutating into y within time t is given as

$$P_\alpha^t(x,y) = \begin{cases} (0.25 + 0.75\exp(-4\alpha t)) & \text{if } x = y \\ (0.25 - 0.25\exp(-4\alpha t)) & \text{if } x \neq y, \end{cases}$$

where α is the unknown parameter. Note that the probability $P_\alpha^t(x,x)$ is the probability that a nucleotide does not change over the considered time period given the parameter α, it is hence a conditional probability (conditioned on α). For the Hepatitis C virus data set, out of the 2610 nucleotides in segment A there are 78 that have mutated over the period of 13 years leaving 2532 unchanged.

Assuming that the pairs $(X_i, Y_i), i = 1, \ldots, n$ are i.i.d., we can write

$$P_{t,p,\alpha}((X_i, Y_i) = (x,y)) = p(x)P_\alpha^t(x,y),$$

where p is a vector of point probabilities on $\{A, C, G, T\}$ and assumed to be constant, i.e. $p(x) = 0.25, x \in \{A, C, G, T\}$. $P_\alpha^t(x,y)$ is the conditional probability that x mutates into y in time t.

- Write a function in *Python* that, given an α parameter and time value, returns the matrix of transition probabilities, i.e., calculate P_t^{α}. The default is `time = 1`.

- Derive and implement the computation of the log-likelihood function for $P_{t,p,\alpha}((X_1, Y_1) = (x_1, y_1), \ldots, (X_n, Y_n) = (x_n, y_n))$, where (x_i, y_i) is the observed nucleotides x_i that mutates into y_i at ith position.

- Find the optimal α that maximize the likelihood estimation (MLE) for the Jukes-Cantor model given the observed counts in `HepCevol.txt`. It should be noted that `HepCevol.txt` only gives values for off-diagonal terms, the diagonal terms for three segments are $[470, 761, 746, 555]$ (SegmentA), $[252, 389, 347, 271]$ (SegmentB), $[230, 299, 282, 198]$ (SegmentC).