Quiz 2: Clustering and Classification
შეფასება: 8 ქულა

მოცემულია 3 დავალება, თითოეული ფასდება 4 ქულით. ქვიზის ჯამური შეფასებაა 8 ქულა + 4 ბონუს ქულა. სურვილისამებრ შეგიძლიათ, ამოარჩიოთ და შეასრულოთ თქვენთვის სასურველი 2 სავარჯიშო.
ზოგიერთ სავარჯიშოში საჭიროა შეკითხვებზე სიტყვიერი პასუხის გაცემა, რაც სასურველია ატვირთოთ word/pdf ფაილის სახით.
საბოლოოდ ატვირთეთ ყველა სამუშაო ფაილი დაზიპული ფაილის სახით.

1 Naive Bayes Classification (4 ქულა)

In this problem, we will use a Naive Bayes classifier to label fragments of the genome based on sequence properties.

(a) Suppose we want to classify sequence fragments into categories (represented by random variable Y): genes, regulatory motifs, or repetitive elements. We want to use the following features: length X_1, GC content (proportion of bases which are G or C) X_2, and *complexity* X_3 (intuitively, what fraction of possible k-mers are observed).

Does the naive Bayes assumption hold in this setting? Explain why or why not.

(b) Regardless of whether the naive Bayes assumption holds, we can still build a classifier. (Surprisingly, naive Bayes classifiers perform well in many applications where this assumption does not hold.) To simplify, we will discretize each of the features.

Given the training set below, write down the maximum likelihood estimates (recall these are relative frequencies) of each of the conditional probability distributions $P(X_i \mid Y)$ and the prior probability distribution $P(Y)$.

GC Content	Length	Complexity	Class
Low	Long	High	Gene
Low	Long	Low	Gene
High	Long	High	Repeat
Medium	Short	High	Motif
Medium	Short	Low	Motif
High	Long	Low	Repeat
High	Short	High	Motif
Medium	Long	High	Gene
High	Long	Low	Repeat
High	Short	High	Motif

(c) Given the model, compute the maximum a posteriori estimate of the class of the new observation below. (Hint: Is it necessary to compute the denominator in Bayes Theorem?)

GC Content	Length	Complexity
Medium	Long	Low

2 Classification of conserved regions (4 ქულა)

In this problem, we will use simulation to study the problem of classifying conserved sequence fragments given multiple alignments of four species. Submit all code you write.

(a) To simplify our classification problem, we will consider *alignment scores* at each position. We define the alignment score of a column of a multiple alignment to be the number of unique pairs that share the same symbol. An example multiple alignment and the score for each column is given below:

```
GACTA
TACTA
AGTTA
CTTAA
-----
01236
```

Consider two models C for conserved regions and N for unconserved regions. Assuming the alignment score at every position is independent, the conditional probability of observing a particular score in a column given each model is tabulated below:

Score	N	C
0	0.1	0.05
1	0.35	0.15
2	0.25	0.2
3	0.2	0.3
6	0.1	0.3

Compute the conditional probabilities of observing each of the following alignments given each of the models:

```
ACGACGACTA
CAGACGCTGA
TTCCTCTGAT
AGATGTGACT
```

```
ACAACGAGTA
AAAACGAATA
TCATCGAGTT
ACATCTAACT
```

(b) Simulate 10,000 sequences S of alignment scores of length 10 from N. How often is $P(S \mid C) > P(S \mid N)$?

(c) Simulate 10,000 sequences S of alignment scores of length 10 from C. How often is $P(S \mid N) > P(S \mid C)$?

(d) One way to reduce the rate of classification errors on short fragments is to favor using scores that are better at discriminating between the two models. Please provide:

i. a pair of score values that is good at discriminating between the two models and

ii. a pair of score values that is not good at discriminating between the two models.

Would the rate of classification errors decrease if we dismissed any alignment (column) with a score of 0?

(e) How could we reduce the rate of classification errors for much longer sequences?

3 K-means clustering (4 ქულა)

In this problem, you will implement k-means clustering on the expression profiles of two genes across a set of breast cancer patients. We have collected expression data from a pair of tissue types from the same set of 700 patients. We now wish to find clusters in this data that correspond to different breast cancer subtypes.

To run the code in this problem, you will have to install R on your computer. The kmeans zipped folder available through the Materials tab on the Stellar course website contains the following files:

- **kmeans.py,** which contains skeleton functions you will have to implement
- **kmeans_plot.R**, which plots the k-means clusters at each iteration of the algorithm (don't mess with this code unless you know your way around R and want to make your plots look prettier)
- a set of **tissue∗_data.txt** files, which contain gene expression data from 700 patients on a series of tissues

(a) Your first task is to add code to kmeans.py to implement the k-means algorithm. To do this, you will have to complete the **assignPoints** and **recalculateCtrs** functions, and then add calls to these functions to the main() function in kmeans.py where indicated. Submit your version of kmeans.py.

(b) Run your code on tissue1 using the command python kmeans.py tissue1. If your implementation is correct, the algorithm will converge in four steps. Submit the plots generated by the code.

(c) Now run your code on tissue2 (your algorithm should converge in six steps this time). What went wrong? What strategy would you employ to find the settings of the algorithm so that it identifies the most obvious clusters, assuming you couldn't see the clusters ahead of time?

Experiment with the code in main() to try different approaches. Use insights from your strategy to make corresponding changes to the main() function in kmeans.py (you should only have to tinker with one line). Run the algorithm again, and describe how your solution addressed the problem using some of the output plots for reference.

Hand in your write-up, with the figures in the same document if possible.