MTH 511a - Mini Project

Instructor: Dootika Vats

Due: 2nd November at 8:00pm

Please read the instructions on submission **very** carefully. The checking will be **automatic**, so if you do not follow proper directions, you will not be able to get any marks. This mini-project counts for 10% of the overall grade.

Data

Each and every one of you is provided with a unique dataset of 4 columns and 500 rows. You can load the data in R using the following command:

```
dat <- read.table("https://dvats.github.io/assets/data/rollnumber.txt")
```

where replace `rollnumber` with your roll number.

Model

Your goal is to fix a mixture model to the data provided to you. The data provided to you may have any number of clusters with a maximum cluster size of $C = 7$.

You may use any technique you like to fit the best mixture model to this four-dimensional data. Your final chosen model must be saved in a list called `model`. For example, in the Gaussian mixture model case, `model` would have list components

- `Clusters` = number of fitted clusters (C^*). C^* will be a number between 2 and 7 (inclusive)

- `pi` = a numeric vector of mixture probabilities of length C^*
- `mu` = a $C^* \times 4$ matrix of mean vectors of the C^* different mixture components
- `Sigma` = a list of C^* 4×4 covariance matrices; one for each component.
- `log.like` = negative log-likelihood value on the given data.

Prediction

Write a function `pred.loss` that has arguments `X.new` (for test data input) and `model` (a fitted mixture model discussed above). This function calculates the loss on a new dataset `X.new`. I will input an `X.new` matrix of size $n_1 \times 4$ for some $n_1 > 0$ and the function should output `loss`, a scalar number.

```
pred.loss <- function(X.new, model)
{
  ...
  loss <- ...
  return(loss)
}
```

There should be no other input arguments in the function and the function should not use any other global variables aside from `X.new` and `model`. The `loss` is calculated based on the model you've fit.
Any external packages needed to run this function, must be loaded *inside* the function.

Submission

At the end of your script, run the following code:

```
save(pred.loss, model, file = "rollnumer.Rdata")
```

This will save your function `pred.loss` and `model` in a file with your roll number as the name, in your current working directory. (To see your working directory, type `getwd()` in the `R` console).

When done, please upload your `rollnumber.Rdata` file in the following Dropbbox link:

`https://www.dropbox.com/request/3qxQJd59SnUZUhPeSVzB`

When submitting, please follow the instructions:

- **Please sign out** of Dropbox if you are signed in to Dropbox.
- Under "Your Name" write down your **roll number**
- Under "Your Email" use your **iitk email id**.

Some words of caution

- Make sure your optimization routines don't consume all `max.iter`. If this happens, then this means you haven't converged as yet to the estimator.
- If your estimate of the covariance matrix Σs is not positive-definite in any iteration, I recommend making your code run in such a way that it repeats that run with a different starting value.
- Follow the naming conventions I use here, otherwise you may not get any points. Particularly, make sure your final estimated model is stored in `model` and your function name is `pred.loss`. Do not use any other names for these two objects!
- Before submitting the `rollnumber.Rdata` file, make sure everything works. Change your working directory to the folder that contains `rollnumber.Rdata`. Run the following lines

  ```
  rm(list = ls())
  load("rollnumber.Rdata")
  ```

  This will first clear all memory of the R session and then load your `.Rdata` file. After this, call `pred.loss(X, model)` using a dummy X matrix you create of size $n_1 \times 4$, for any $n_1 > 1$. If you get back a scalar number, then that means your function should give me no errors. The only possibility of an error is if your function uses a package that is not loaded *inside* the function.

Evaluation

A unique `X.new` has been created for each and every one of you. This dataset will not be revealed to you. When you submit `rollnumber.Rdata`, I will load this in `R` and calculate

```
test.error <- pred.loss(X.new, model)
```

using your function. I will also calculate the loss under the *true model* for your dataset. (This true model is what was used to generate the data – this is unknown to you).

```
true.error <- true.loss(X.new, true.model)
```

Finally, I will calculate

$$\text{Score} = \frac{\texttt{test.error}}{\texttt{true.error}}$$

The closer your score is to 1, the higher marks you get. Note, I will also check your `pred.loss` function. If there is an error in that function, then I will deduct points, and use my own version of the `pred.loss` function.

Good luck and SUBMIT ON TIME! Late submissions will be flagged by Dropbox and marks will be deducted.