Statistical Learning and Data Analysis 2020 - 52525
Lab 2 - Elections and RNA-sequencing

Yuval Benjamini

Due May 26th before 4:30pm

Hand In Procedure: Labs can be handed in alone or in pairs (no more than 2 per lab!). Please prepare a file with a writeup and code (the writeup can be in Hebrew or English). Please make sure the reports are less than 6 pages long (not including code).

1 Simulation Study

Here you are going to explore the behavior of running K-means with or without extracting the top 3 PCs, using a simulated dataset. We would like to see whether clustering can accurately recover the original distribution of each of the points. The goal is to compare accuracy and the speed of two algorithms, for different dimension p and different levels of variation.

1.1 The data generation process

In each simulation we create $n = 100$ vectors of p dimensions from 3 multivariate Gaussians. Each distribution has a different mean vector; they all have the same covariance:

$$\mathbf{x}_i \sim Normal_p(\mu_{\pi(i)}, \sigma_\epsilon^2 I) \quad \pi(i) = \begin{cases} 1 & \text{if } i = 1, ..., 20 \\ 2 & \text{if } i = 21, ..., 50 \\ 3 & \text{if } i = 51, ..., 100 \end{cases}$$

The mean vectors μ_1, μ_2, μ_3 are different only in their first 10 coordinates. We will generate these coordinates once for all the simulations [[why?]]:

$$\mu_{jk} \sim N(\mathbf{0}, \sigma_\mu^2) \text{ if } k = 1, ..., 10, \quad \text{and 0 otherwise.}$$

We will fix $\sigma_\mu^2 = 1$. Note that 2 vectors should be sufficient to tell apart the means of the three distributions.

1. Generate the first 10 coordinates of each μ_j vector $j = 1, .., 3$.
2. Write a function that outputs a simulated dataset of dimension $100 \times p$ given (a) the first 10 coordinates of each mu_j, (b) p, and (c) σ_ϵ^2.
3. Choose 4 levels of σ_ϵ^2 and use $p = 10, 20, 50$.
4. For each combination of σ_ϵ^2 and p, generate multiple datasets (say $B = 50$).
5. For each data-set:
   - Run K-means once on the p dimensional data.
   - Use PCA to get a 3 dimensional representation of the data (100×3). Run K-means again on this data-set.

- For each run, save the accuracy and the run-time.

6. Compute the average accuracy and the std-err for each $(p, \sigma_\epsilon^2, algorithm)$. Display these in a figure and a table.
7. Show a plot describing run-time. Here, you don't have to show all the data.
8. Summarize briefly your findings.

Comments:

- In computing accuracy, you should compare the clustering result to the data-generating partition of the data (the first 20 examples belong to the first cluster, etc).
- You should choose σ_ϵ^2 so that there will be meaningful differences between the levels.

2 Comparing demographic and election data

In this part we will explore how socio-economical similarity between cities relates to similarity in voting patterns. We will use the demographic dataset `csb_demographics.txt` [1] that is used to create the socio-economic ranking by the Israeli Statistical Bureau (ISB); each row is a town or "moatza mekomit", and the variables represent some demographic property. We will compare the demographics to 23'th Knesset election results found in file `knesset23_res.csv` [2]. Both are on Moodle. We will add an example for how to read and preprocess the demographics file (see Tirgul 4 for the elections data). I suggest you take a look at the original websites and files (in the footnotes) before starting the analysis.

For this set, it will be useful to use the `dendextend` library which is downloadable from cran. Write `install.packages('dendextend')` in R, and import the library. I will provide a link to the package tutorial in Moodle.

1. Randomly choose a set of 20 cities described in the ISB data sets. Identify these cities in the election data set, and construct a vector summarizing all votes for each of these cities. You should now have two data sets with the same cities.
2. Construct a hierarchical tree for the elections data. Decide on how to define distances between two cites so that the results are meaningful.
3. Construct a hierarchical tree for the demographic data. Decide on how to define distances between two cities so that the results are meaningful.
4. Compare the two hierarchies. Comment on similarities and differences.
5. Choose a similarity score for the two trees. You can base your score on one of the scores implemented in the `dendextend` package, including Baker's Gamma, the cophenetic correlation or the Fowlkes-Mallows (Bk) index. The score should be high if the trees (and matching labels) are identical, and low otherwise Explain what the score is measuring. (I'll upload the tutorial of dendextend which gives leads to all these functionalities.) Calculate the score for your trees.
6. Find a background distribution for this score, assuming the labels of the trees are completely unrelated. To do this, randomly permute the labels (city names) of one tree, keeping the labels of the other tree fixed. For each of the randomizations, compute the similarity score. State the null hypothesis for this randomization, use a plot to compare the randomization scores to the score of the observed trees, and compute a p-value. What have you learned?

[1] http://old.cbs.gov.il/hodaot2016n/24_16_330t3.pdf
[2] https://votes23.bechirot.gov.il

3 Exploratory analysis of RNA seq data

The 'gtex' data set contains Gene Expression estimates collected by the Genotype Tissue Expression (gtex-portal.org). Each row signifies a gene, and every column a tissue type (e.g. Heart, Exposed Skin, Unexposed Skin). The value measures the median expression level of the gene across multiple samples of the same tissue. Values are positive, with zero meaning there is no indication of the gene in the tissue. Note that some genes might not be expressed in any tissue.

Because clustering is usually exploratory, it is important to allow easy interaction between the researcher and the data. Your goal is:

1. Write a function that will run the K-means algorithm for this data. [Do not use the R-implementation].
2. Create an interactive tool based on the shiny app.

The app should allow a researcher to find an interesting clustering of the tissues, and to visually inspect her/his results. This will be based on the a ShinyApp. To create the app you will need to use RStudio. Go to NewFile-NewShinyApp and create a new single-file app.

Shiny apps

A shiny app requires inputs, and a plotting function. Most of your changes are to the plotting function `renderPlot`. Change it so that it would

1. Run a K-means algorithm (using your function).
2. Show a two-dimensional rendering of your tissues, with colors corresponding to the resulting clusters.

The input function is in the slider bar. Change it so it would read the number of clusters.

Comments

- For better results, it is preferable to normalize the data before running the algorithm. Normalization steps might include removing genes with very small mean expression, scaling the different genes, or taking a log transformation. Feel free to explore on your own, trying to get samples from related regions close to each other. (The Brain samples should cluster nicely together).
- If you find the app confusing, focus first on making a function that would run the clustering algorithms and show the results nicely. It is very easy to turn this into an app, and we will help you.
- Feel free to add functionality, change clustering algorithm, or any other modification that will improve your app.
- Submit the code of the app, as well as a screen shot of a nice clustering result.

Reading the data

To read the file, use:

```
med_dat = read.delim(''gtex.txt'',skip = 2, row.names=c(1),header = TRUE)
```

Notice that the first column is the gene name, so you might want to save it on the side:

```
gen_names =med_dat[,1];   med_dat = med_dat[,-1]
```