VE472 — Methods and tools for big data

Lab 7

Manuel — UM-JI (Summer 2021)

Goals of the lab

- Work with a real dataset
- Practice PCA and Gradient Descent in Spark
- Get familiar with Dataframe and MLlib

1 Introduction

While Reapor was planning to expand his cinema concept in parallel of his new music platform, he was badly hit by the COVID-19 pandemic: due to the lock-down his cinema had to shutdown. Krystor and Frank had already started to work on a Video On Demand platform to adjust to the new economical context when Reapor Rich had what he called "a much lucrative" idea. Most sectors of activity were in a very bad economical state but pharmacology was shining. It was time for Reapor to diversify himself, or even totally switch field!

Counting on Krystor, Frank, and you to lead the effort, Reapor's new challenge is to become a prominent a actor in the field of bioinformatics. His feeling is that all the analyses they your ran on the Million song dataset, or on his movie project could somehow to adapted to studying biology. As a first small scale test he would like you to cooperate with Krystor and Frank in oder to categorize cells based on the amount of mRNAs they produce.

It feels Reapor is very excited about his new idea: if he can find more effective cell to produce mRNA than his competitors, then he well be able to produce more vaccine at a lower cost, meaning he will get more money and become even richer!

2 Tasks

Krystor found an interesting dataset, `PBMC.tar.gz`,[1] which feature a pre-processed table from the Peripheral blood mononuclear cell (PBMC) [1]. More specifically it contains the amount of mRNA produced by each type of cell. Each cell works using the following steps:

1. Produce some mRNA, which are copies of substring of its DNA;
2. The mRNA can then be used to produce certain protein;

The idea is now to categorize cells based on the amount of mRNAs they produce.

Pre-processing

First you need to prepare the data:

- Import the csv file `PBMC_16k_RNA.csv` into a spark dataframe. *Hint:* use `inferSchema='true'`.
- Find the range of data in the first two features KLHL17 and HES4 inside the dataframe.
- Is it necessary to perform data standardization? Explain.

[1] Available on the data server.

PCA Analysis

As you always want to ensure you do not make any mistake you proceed with PCA analysis but stop at each step to consider how valid and useful your work is.

- Why could PCA be useful to analyse this dataset?
- Use `org.apache.spark.mllib.feature.PCA` to perform PCA analysis on the dataset. For now only retrieve the first two principal components.
- How much of teh dataset do the two first principal components explain?
- In this context how useful is PCA?
- Import the csv file `PBMC_16k_RNA_label.csv`, which contains the type of cell for each index.
- Plot a scatter plot with the obtained two principal components, with different types of cell in different colours.
- What can you observe?
- Plot another plot using two random columns of the original dataframe.
- Describe the difference in the two plots.

Categorization using Gradient Descent

We now try to perform some predictions: given the amount of mRNA a cell produces, predict what type of cell it is. For those initial tests Krystor suggest to apply logistic regression with the aim of splitting the data into two categories: B and non-B cells.

- Prepare two subsets of the datasets: 70% as training set and 30% as test set. Frank reminds you to only use the training set to the perform logistic regression.
- Determine the type of cell with the most data points and call it A cells, labelled (1) while non-A cells are labelled (0).
- To categorize A and non-A cells, we use `spark.mllib.classification.LogisticRegressionWithLBFGS`[2] with the principal components p_1 and p_2.
- Run Logistic Regression with gradient descent on the training dataset, and apply your model to predict the label in the test set.
- What can your report to Reapor, and more importantly should you be afraid of his reaction?
- Propose ways to improve the model.

References

[1] Artificial-cell-type aware cell-type classification in CITE-seq Lian, Qiuyu and Xin, Hongyi and Ma, Jianzhu and Konnikova, Liza and Chen, Wei and Gu, Jin and Chen, Kong, Bioinformatics, Volume 36, pages i542-i550, 2020/07, https://doi.org/10.1093/bioinformatics/btaa467

[2] BFGS is an advanced version of gradient descent that utilizes quasi-newton method and Hessian matrix.