Seminar in Data Science
Summer 2020

Homework 1

Due date: 7/24/20, at 23:59. Please LaTeXor handwrite your homework solution and submit an electronic version.

1. **Factor model** Suppose we have obtained a covariance matrix $\Sigma \in \mathbb{R}^{p\times p}$ and Σ is positive semidifinite. We seek to approximate it with a factor model of the form $\hat{\Sigma} = \lambda I_p + FF^T$, where F is a $p \times k$ matrix, or "factor loadings". $\lambda > 0$ is the "idiosyncratic noise" variance, and I_p a $p \times p$ identity matrix. The stochastic model that corresponds to this is

$$y = Ff + \sigma e,$$

where $y \in \mathbb{R}^p$ is the (random) vector of centered observations, $f \in \mathbb{R}^k$ is a random variable with zero mean and $e \in \mathbb{R}^p$ is a noise vector with identity covariance matrix. $\sigma = \sqrt{\lambda}$ is the standard deviation of the idiosyncratic noise component σe. To approximate F, λ, we seek to solve

$$\hat{\Sigma} := \arg\min_{F,\lambda>0} \|\Sigma - \lambda I_p - FF^T\|_F,$$

where $\|\cdot\|_F$ stands for the Frobenius norm of its matrix argument. While a seemingly simple problem, without additional assumptions, this problem is difficult to solve numerically to global optimality. In this exercise, we describe one approach to solving the problem which we will implement later in the homework.

   (a) Suppose $\lambda = \bar{\lambda} > 0$ is given and fixed. Express the eigenvalues of $\Sigma - \lambda I_p$ in terms of the eigenvalues of Σ. Use your result to express and optimal F as a function of λ which we'll denote $F(\lambda)$. *Hint: If $\Sigma = U\Lambda U^\top$ then $F(\lambda) = UD^{1/2}I_{p\times k}$ for some D that you will calculate where $I_{p\times k}$ is the first k columns of a $p \times p$ Identity matrix.*

   (b) Now we optimize over $\lambda > 0$. Plugging in the $F(\bar{\lambda})$ (note $\bar{\lambda}$ is fixed, it is not variable) matrix computed in the previous part, find the optimal λ.

2. **Clustering time-series** In this exercise, the focus is on clustering time-series, and on reproducing the results of a recent paper[1] which argues that, for time-series, the popular k-means algorithm may be meaningless when the subsequences are extracted using moving windows.

   (a) Explain in a short paragraph the main point of the paper.

   (b) Load the daily close log-return data from the Dow Jones Industrial Average data, in the file `dji_5yr_data.csv`. Generate subsequences based on a 5-year data span, and month-length subsequences (that is, each subsequence has 20 data points).

   (c) Implement and run a time-series clustering method based on a hierarchical clustering approach. Make sure you show the resulting tree, and comment; in particular, how many clusters appear to be reasonable? Discuss.

[1] Keogh, Eamonn, and Jessica Lin. "Clustering of time-series subsequences is meaningless: implications for previous and future research." In *Knowledge and information systems* 8.2 (2005): 154-177. Available on bCourses.

(d) Try to validate the claim that the cluster centroids found by using the moving window approach resemble sine waves, as shown in Figure 9 in the paper. Remember to center and normalize your data!

(e) Load the Cylinder-Bell-Funnel (CBF) data, as posted on bCourses, in file `30_128_X.csv` and `30_y.csv`, where `30_128_X.csv` contains 30 sequences of length 128 and `30_y.csv` contains 30 corresponding labels. where label 0, 1, and 2 represents cylider, bell, and funnel respectively, which are 3 different classes of sequences. Visualize the 3 different classes of sequence. What have you observed?

(f) Repeat part (c) and (d) on the CBF data. You don't need to center and normalize the dataset.

(g) Concatenate the 30 sequences from `30_128_X.csv` into a single long sequence and perform the moving window approach with window size 128 to extract subsequences. Then repeat part (c) and (d) on the extracted subsequences. Contrary to part (f), you should observe sinusoidal waves like in part (d). Remember to center and normalize your data!

3. Generalized low-rank models Consider the data given in the file `INTLFXD_csv.zip`, which contains the exchange rates against the USD for several currencies, from 1999 until present.

(a) Use PCA to try to answer the following qualitative questions. Note that there are no unique and definitive answers to these—simply do your best!

i. Prepare data sets that correspond to daily, weekly, monthly, or annual fluctuations using `1999_2018_complete.csv`. Normalize the data so that the data points are all on the same scale.

ii. Run PCA using the entire data set. Based on coefficients found in the eigenvectors, what are the main drivers (that is, countries) of exchange rate fluctuations?

iii. Re-do the previous part for every day, week and month per year. How do coefficients in the eigenvectors shift over the years for each country, or a particular group of countries? Please summarize the findings in a few sentences, and illustrate those with a few appropriate tables or line graphs.

iv. Based on the answers from the above two parts, what are the main clusters of currencies? How do these compare to external knowledge, such as geography?

(b) Another way to discover the main drivers of exchange rate fluctuations is to use a factor model. Using the results of Problem 1, we can approximate a covariance matrix Σ using a factor model $\hat{\Sigma} = \lambda I_p + FF^\top$ using Algorithm 1.

Algorithm 1 Factor Model Approximation to Σ

Input: Σ, $\lambda = 0$, k
for $k \leftarrow 1$ to N **do**
    $F \leftarrow$ Solution to 1(a)
    $\lambda \leftarrow$ Solution to 1(b)
end for
return $\lambda I_p + FF^\top$

We can then look at the columns of F to infer the important factors. For example, if $k = 1$, then F is just a vector and the largest entry of F (in absolute value) would be a strong indication that the corresponding currency has a dominant effect on the structure of the covariance matrix Σ.

Take the normalized daily fluctuations data from part (a) i. and construct Σ (it should be a 23×23 matrix). For simplicity, we will deal with the correlation matrix – that is scale the Σ calculate by the number of data points such that $\Sigma_{ii} = 1$ for $i = 1, \ldots, 23$. Then, with $k = \{1, 2\}$ and $N = 5$, get two factor models of Σ. What is $\|\Sigma - \lambda I_p - FF^\top\|_F^2$ for both factor models? For $k = 1$, what seem to be the main drivers of the currency fluctuations? How do these differ with the results you got from PCA? Finally, briefly discuss the influence of N on the final result (i.e. how large does N need to be for our objective $||\Sigma - \lambda I - FF^\top||_F^2$ to converge)?

(c) There are many variants of PCA, some of which are described in a paper by Udell *et al.*[2] Focus on the *matrix completion* functionality that allows to run PCA on an data matrix with some elements unknown. For the rest of this problem, use the currency data only for the year 2018. For this problem, we will use the `h2o` package in python.

Instructions on Downloading `h2o` package in python We will be using the package for Python 3. First it requires installing Java from this website `https://www.oracle.com/java/technologies/javase-downloads.html` (make sure you install Java SE 11 and not Java SE 14). After doing so, follow the python installation instructions here: `https://docs.h2o.ai/h2o/latest-stable/h2o-docs/downloading.html`. In order to verify everything was installed properly, you must be able to run `h2o.init()` without any errors and be able to run the demo `h2o.demo("glm")` without any errors. For instructions on how to use the package, see the example here: `https://docs.h2o.ai/h2o/latest-stable/h2o-docs/data-science/glrm.html`.

i. Remove between 5% to 40% of the currency data matrix randomly in 5% increments and test whether the algorithm successfully recovers the removed data. Discuss the appropriate metrics. Make a plot where the y-axis is your appropriate metric and the x-axis is the percentage of the data which was randomly removed (include error bars).
Parameters: Set $k = 5$ (the rank of the approximation) and don't use any regularization. Se the number of max_iterations to 100.
Note: In order to use the matrix completion functionality in `h2o`, your training frame must include `NaN` entries. You may then find the indices of the randomly places `NaN` entries, train your model, and then view the prediction on these entries.

ii. In many practical instances, a whole chunk of data is missing; for example, some currency rates might not be available before a certain date. How do you expect matrix completion algorithms to behave in that case? Test this on the currency data, where the data for the first few months of a specific currency is removed. Discuss the performance in this scenario compared with removing data randomly.

[2] At `https://web.stanford.edu/~boyd/papers/pdf/glrm.pdf`.