Statistical Machine Learning GU4241/GR5241
Spring 2019
https://courseworks.columbia.edu/

Homework 2

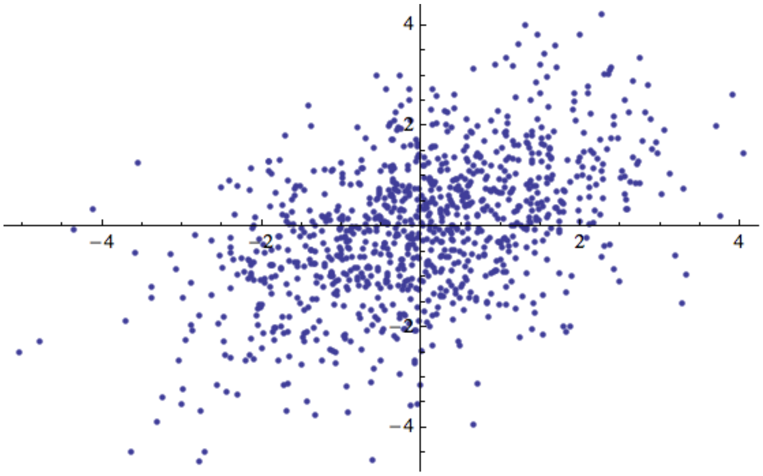
Due: Sunday, Feb. 17th, 2019

Homework submission: Please submit your homework electronically on Gradescope and the code on Canvas by 11:59pm on the due date. Please note that you need to submit both the pdf file and your code (either in R or Python).

Problem 1 (PCA, 4+6 points)

1. Please graph approximately into the following picture:

   - The first principle component (the 1-dimensional subspace onto which PCA with one component would project). Mark this ξ_1.
   - Pick an arbitrary data point (reasonably far away from the principal component) and graph the direction in which it will be projected.



2. The next figure shows (on the left) 100 examples from a data set of face images. Each image is a 92×112 grayscale image and can be interpreted as a vector $x \in \mathbb{R}^{10304}$. The figure on the right shows the first 48 principal components $\xi_1, \ldots, \xi_{48}$, visualized as images.

(a) How many principal components are there in total?

(b) Can you describe a method that *approximately* reconstructs a specific face image $x \in \mathbb{R}^{10304}$ using only the first 48 principal components? To describe your method, denote by $\hat{x}$ the approximate representation of x. Represent $\hat{x}$ as an equation $\hat{x} = \ldots.$ Please define precisely what each variable occurring on the right-hand side of the equation means.

Problem 2 (PCA of Stock Prices, 10 points)

1. For each of the 30 stocks in the Dow Jones Industrial Average, download the closing prices for every trading day from January 1, 2018 to January 1, 2019. To download the prices, for example for symbol `AAPL`, we use the R package `quantmod`. The code is as the following:

```
library(quantmod)
data<-getSymbols("AAPL", auto.assign = F, from = "2018-01-01", to = "2019-01-01")
```

   Please find a way to download data for the 30 stocks efficiently.

2. Perform a PCA on the closing prices and create the biplot (call function `princomp()` and use `cor=FALSE`). Do you see any structure in the biplot, perhaps in terms of the types of stocks? How about the screeplot – how many important components seem to be in the data?

3. Repeat part 2 with `cor=TRUE`. This is equivalent to scale each column of the data matrix.

4. Use the closing prices to calculate the return for each stock, and repeat part 3 on the return data. In looking at the screeplot, what does this tell you about the 30 stocks in the DJIA? If each stock were fluctuating up and down randomly and independent of all the other stocks, what would you expect the screeplot to look like?

Problem 3 (Classical Multidimensional Scaling, 8 extra points)

Multidimensional scaling is an approach for dimension reduction. Assume we have a data set $x_1, x_2, \ldots, x_n \in \mathbb{R}^p$. We first need to define a similarity matrix $\mathbf{S}$. In *classical scaling*, $\mathbf{S}$ is the centered inner product matrix with elements $s_{i,i'} = \langle x_i - \overline{x}, x_{i'} - \overline{x} \rangle$. We want to find a set of points $z_1, z_2, \ldots, z_n$ from a low dimensional space $\mathbb{R}^k$, such that the following quantity is minimized

$$S_C(z_1, z_2, \ldots, z_n) = \sum_{i,i'} (s_{i,i'} - \langle z_i - \overline{z}, z_{i'} - \overline{z} \rangle)^2. \quad (1)$$

Let $d_1^2 \geq d_2^2 \geq \cdots \geq d_k^2$ be the k largest eignvalues of $\mathbf{S}$, with associated eigenvectors $\mathbf{E}_k = (\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_k)$. Let $\mathbf{D}_k$ be a diagonal matrix with diagonal entries $d_1, d_2, \ldots, d_k$. Show that the solutions z_i to the classical scaling problem (1) are the *rows* of $\mathbf{E}_k\mathbf{D}_k$. In practice, usually we choose $k = 2$.

We can show this through the following steps.

- Let $\mathbf{T}$ be the centered inner product matrix with elements $\langle z_i - \overline{z}, z_j - \overline{z}\rangle$. Then $\sum_{i,i'}(s_{i,i'} - \langle z_i - \overline{z}, z_{i'} - \overline{z}\rangle)^2 = \text{tr}[(\mathbf{S} - \mathbf{T})^2]$.
- Assume that the eigen decomposition of $\mathbf{S}$ and $\mathbf{T}$ are $\mathbf{U}\mathbf{D}^2\mathbf{U}^T$ and $\widetilde{\mathbf{U}}\widetilde{\mathbf{D}}^2\widetilde{\mathbf{U}}^T$, where $\widetilde{\mathbf{D}}$ is a diagonal matrix, with diagonal elements $\tilde{d}_1 \geq \tilde{d}_2 \geq \ldots \geq \tilde{d}_k$. Then the problem boils down to minimize

$$S_C(z_1, z_2, \ldots, z_N) = \text{tr}[(\mathbf{U}\mathbf{D}^2\mathbf{U}^T - \widetilde{\mathbf{U}}\widetilde{\mathbf{D}}^2\widetilde{\mathbf{U}}^T)^2] = \text{tr}[\mathbf{D}^4 + \widetilde{\mathbf{D}}^4 - 2\mathbf{D}^2\mathbf{U}^T\widetilde{\mathbf{U}}\widetilde{\mathbf{D}}^2\widetilde{\mathbf{U}}^T\mathbf{U}]. \qquad (2)$$

- Let $\mathbf{A} = \mathbf{U}^T\widetilde{\mathbf{U}}$. Show that

$$\frac{\partial S_c}{\partial \tilde{d}_j^2} = 2\tilde{d}_j^2 - 2\sum_{i=1}^{n} d_i^2 a_{ij}^2, \quad \text{for } j = 1, 2, \ldots, k,$$

  where a_{ij} is the (i,j)th element of $\mathbf{A}$.

- By setting the partial derivative to be zero, we have $\tilde{d}_j^2 = \sum_{i=1}^n d_i^2 a_{ij}^2$. Then minimizing (2) is equivalent to maximizing $\sum_{j=1}^k(\sum_{i=1}^n d_i^2 a_{ij}^2)^2 = \sum_{j=1}^k(\tilde{\mathbf{u}}_j^T\mathbf{U}\mathbf{D}^2\mathbf{U}^T\tilde{\mathbf{u}}_j)^2$, where $\tilde{\mathbf{u}}_j$ is the jth column of $\widetilde{\mathbf{U}}$. This is equivalent to finding the first k principle components of $\mathbf{X}^T$. The solution should be the first k eigen vectors of $\mathbf{S}$.