CmpE 482: Numerical Linear Algebra and Its Applications
Programming Projects

Instructor: A. Taylan Cemgil
Department of Computer Engineering, Boğaziçi University
34342 Bebek, Istanbul, Turkey
taylan.cemgil@boun.edu.tr

June 4, 2017

1 Project 1: Interpolation with Autoregressive (AR) Model

In this project, you are given a time series for monthly totals of international airline passengers which have missing observations, and you are expected to interpolate them by using AR model.

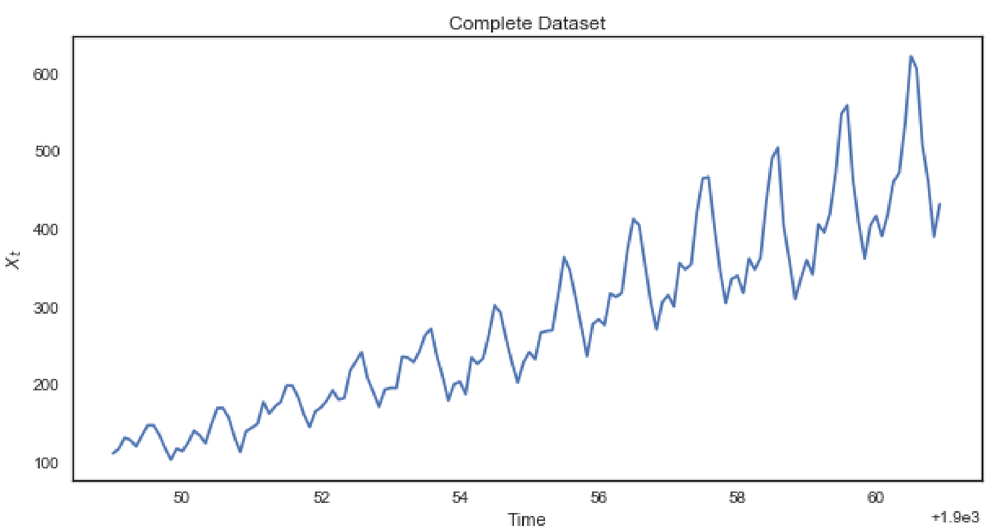


Figure 1: Original dataset

1.1 Autoregressive Model

Autoregressive model is a model where every element X_t of the time series can be represented by the weighted sum of the terms $X_{t-1}, X_{t-2}, \cdots, X_{t-M}$ plus a gaussian error term ϵ_t.

$$X_t = A_1 X_{t-1} + A_3 X_{t-2} + \cdots + A_M X_{t-M} + \epsilon_t$$

1.1.1 Alternating Least Squares For the AR Model

Let $X = \begin{bmatrix} X_0 & X_1 & X_2 & \cdots & X_N \end{bmatrix}^T$ be the vector of the time series, $A = \begin{bmatrix} A_1 & A_2 & \cdots & A_M \end{bmatrix}^T$ be our coefficients, and $\epsilon = \begin{bmatrix} \epsilon_M & \epsilon_{M+1} & \cdots & \epsilon_N \end{bmatrix}$ be the vector of gaussian errors. Assume X is fully observed, then we can write following system of linear equations:

$$\begin{bmatrix} X_M \\ X_{M+1} \\ \vdots \\ X_N \end{bmatrix} = \begin{bmatrix} X_{M-1} & X_{M-2} & \cdots & X_0 \\ X_M & X_{M-1} & \cdots & X_1 \\ \vdots & \vdots & \ddots & \vdots \\ X_{N-1} & X_{N-2} & \cdots & X_{N-M} \end{bmatrix} \begin{bmatrix} A_1 \\ A_2 \\ \vdots \\ A_M \end{bmatrix} + \begin{bmatrix} \epsilon_M \\ \epsilon_{M+1} \\ \vdots \\ \epsilon_N \end{bmatrix} \tag{1}$$

which we will be denoted by $X_{M:N} = G_X A + \epsilon$. Similarly, if we assume A is known:

$$\begin{bmatrix} \epsilon_M \\ \epsilon_{M+1} \\ \vdots \\ \epsilon_{N-1} \\ \epsilon_N \end{bmatrix} = \begin{bmatrix} -\text{flip}(A)^T & 1 & 0 & 0 & \cdots & 0 \\ 0 & -\text{flip}(A)^T & 1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & -\text{flip}(A)^T & 1 & 0 \\ 0 & \cdots & 0 & 0 & -\text{flip}(A)^T & 1 \end{bmatrix} \begin{bmatrix} X_0 \\ X_1 \\ \vdots \\ X_{N-1} \\ X_N \end{bmatrix} \tag{2}$$

which will be denoted by $\epsilon = G_A X$ where $\text{flip}(A) = \begin{bmatrix} A_M & A_{M-1} & \cdots & A_1 \end{bmatrix}^T$. In this system, if X is partially observed we can also define the following vectors X^{ob} and X^{mi}:

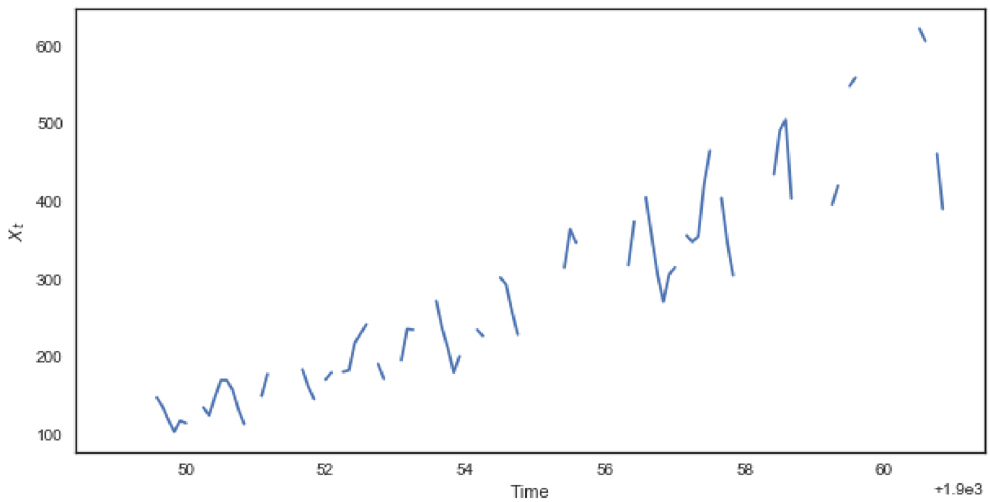
$$X_i^{ob} = \begin{cases} X_i, & \text{if } X_i \text{ is observed} \\ 0, & \text{if } X_i \text{ is missing} \end{cases} \qquad X_i^{mi} = \begin{cases} 0, & \text{if } X_i \text{ is observed} \\ X_i, & \text{if } X_i \text{ is missing} \end{cases} \tag{3}$$

where $X = X^{ob} + X^{mi}$. Then, we can reformulate our last system as $\epsilon - G_A X^{ob} = G_A X^{mi}$.

Both problems can be written in the form of $Cx = b + \epsilon$ and the least squares solution of that is $x = (C^T C)^{-1}(C^T b)$. That means if we would know A, then we can find X, vice versa. However, we don't know neither A nor X, completely. Nevertheless, we can overcome that problem by **Alterating Least Squares** method which is iteratively fixing X and solving $X_{M:N} = G_X A + \epsilon$ for A; then fixing A and solving $\epsilon - G_A X^{ob} = G_A X^{mi}$ for X^{mi}, in a coordinate ascent manner.

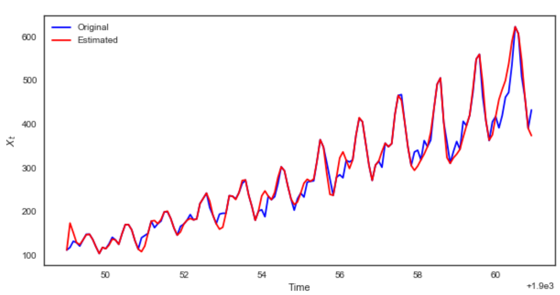
1.2 Things to do

- Read the *flights.csv* file into two column vectors T and X, where X is partially observed monthly totals of international airline passengers and T is the corresponding months. Plot T vs. X.

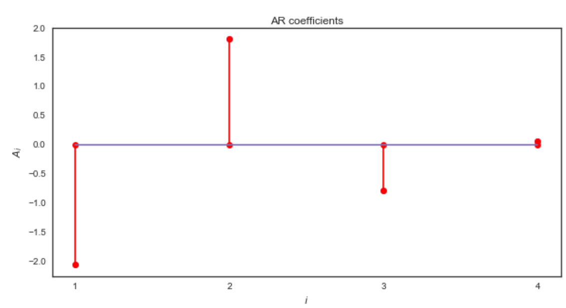


- Iteratively construct G_X and solve for A, then construct G_A and solve for X^{mi} by least squares (Alternating Least Squares). You can choose $M = 4$.

- Plot X with the estimated values and parameters A.
Note: You will plot only the red graphs, since you don't have the original data.



(a) Interpolated X_t values with the original data



(b) A_i values

1.2.1 Bonus

- Try to improve your model by adding 2 more terms that forms an affine function of time (trend):

$$\epsilon_t = A_0 X_t + A_1 X_{t-1} + A_3 X_{t-2} + \cdots + A_M X_{t-M} + Ct + D$$

Follow the same steps above for the new model and report the results. You should see a small improvement.

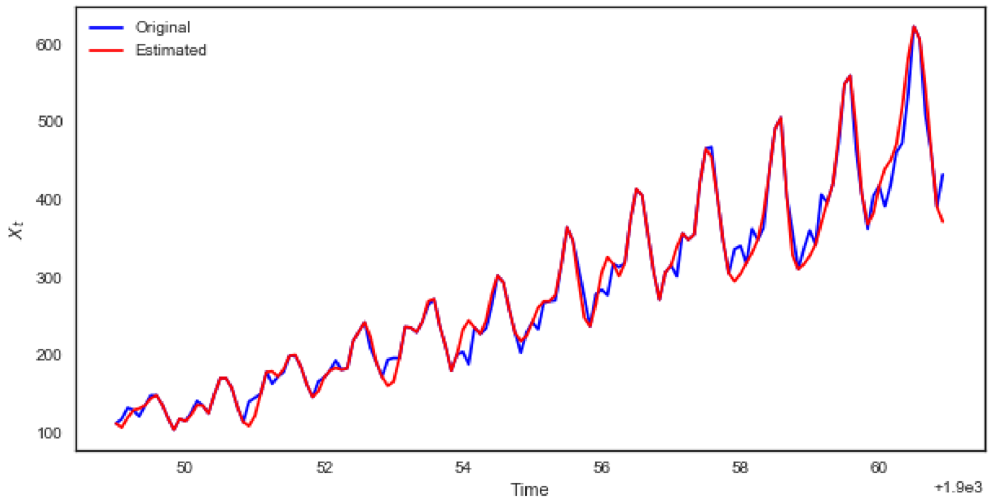


Figure 2: Output of the bonus part

Deadline: 5 June, 2017 10:00

2 Project 2: Image Segmentation with Spectral Clustering

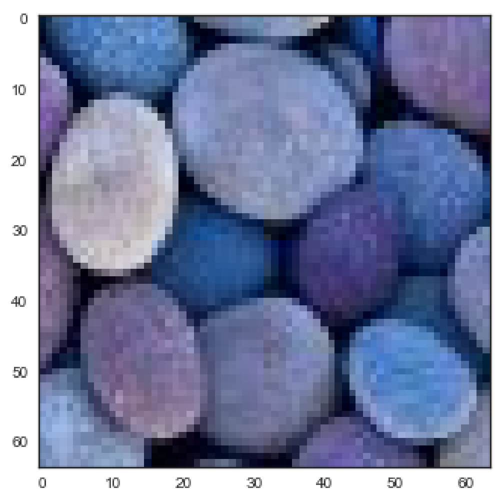
Consider an undirected graph $G = (V, E)$ with vertex set $V = \{1, \cdots, N\}$ and edge set $E \subset V \times V$. Each edge $e = (i, j) \in E$ has a nonnegative weight $W_{ij} = W_{ji}$. We will denote the $N \times N$ symmetric weight matrix as W. The degree of vertex i is $d_i = \sum_j W_{ij}$. The degree matrix D is defined as the diagonal matrix with $D_{ii} = d_i$.

Note that, $T = WD^{-1}$ is a matrix where each column sums to one. Hence, T can be interpreted as the transition probabilities of a random walk. In other words, T_{ij} gives us the probability of the next vertex to be visited is j, when our random walker is at vertex i.

In this project, you are expected to do segmentation on an image by following the steps below. The algorithm below is sometimes called Normalized Cut and is a kind of spectral clustering algorithm [1].

2.1 Things to do

- Read the *stones_rgb.jpg* into a tensor $X \in \mathbb{R}^{64\times 64\times 3}$ where the color intensity of channel c (red, green, or blue) at pixel (x, y) is denoted by X_{xyc} (see: imread). Then, plot the image (see: imshow).



- Let $h = 64$ be the side length of the image and $n = h \times h$ be the number of pixels. Note that, we can order pixels by a single index $i \in \{1, 2, \cdots, n\}$. Namely, p_i is the pixel at position (x, y) if and only if $i = x + h \times (y - 1)$ and define $c_i = \begin{bmatrix} X_{xy1} & X_{xy2} & X_{xy3} \end{bmatrix}^T$ as the color of pixel p_i.

- Construct an $n \times n$ similarity matrix W, so that

$$W_{ij} = \begin{cases} sim(i,j), & \text{if } i \neq j,\ \mid x_i - x_j \mid \leq r \text{ and } \mid y_i - y_j \mid \leq r. \\ 0, & \text{otherwise.} \end{cases}$$

$$sim(i,j) = \exp\left(-\frac{\|c_i - c_j\|_\Lambda^2}{2\sigma^2}\right) \tag{4}$$

- Calculating the similarity of colors directly by Euclidean distance is not a good idea in general. Therefore, we used a weighted Euclidean distance $\|c_i - c_j\|_\Lambda$, where

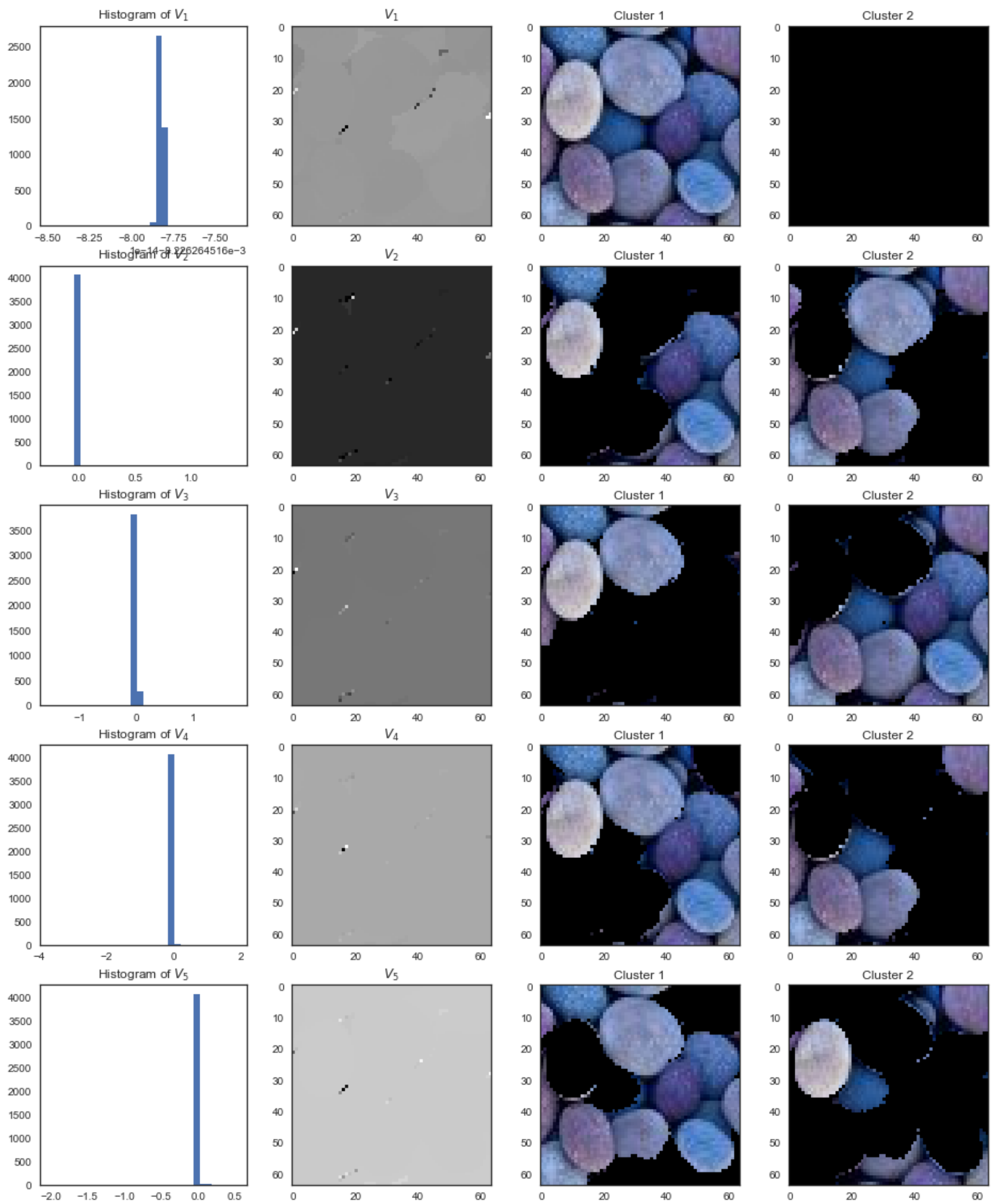
$$\Lambda = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.14713 & -0.28886 & 0.436 \\ 0.615 & -0.51499 & -0.10001 \end{bmatrix} \tag{5}$$

- You can choose $r = 2$, $\sigma = 10.0$ for calculating similarity. Here W can be thought of as the adjacency matrix of a graph that consists of pixels where $W_{ij} = W_{ji}$ is the weight of the edge between nodes (pixels) i and j. Our aim is seperating underlying graph structure into clusters (segments), and we will use Normalized Cut as our criteria.

- Define the Laplacian matrix $L = D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$, where D is the diagonal degree matrix, i.e. $D_{ii} = \sum_j W_{ij}$.

- Let $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$ be the eigenvalues of L and $q_1, q_2, \cdots, q_n$ be the corresponding eigenvectors. Calculate the first k eigenvectors of L (calculating all eigenvectors is a terrible idea) and construct $Q = \begin{bmatrix} q_1 & q_2 & \cdots & q_k \end{bmatrix}$ by QR iterations. You can choose $k = 5$.

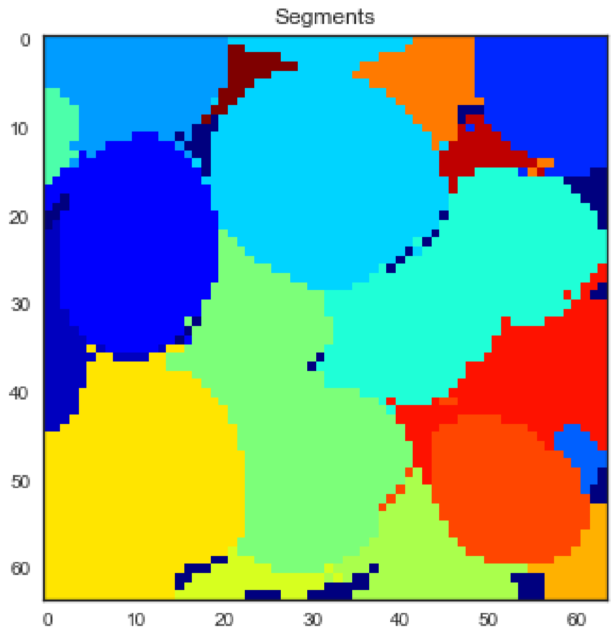
- Let $\tau = 0$, and $V = D^{-\frac{1}{2}} Q$. Now, each column of V, suggests a bi-clustering scheme. To see that, for each column V_k:
  - Plot the histogram of the values in V_k
  - Plot V_k after reshaping it back to $h \times h$
  - If an element of V_k is less than 0, it is assigned to cluster 1. Otherwise, it is assigned to cluster 2. Plot these 2 clusters in picture.



2.1.1 Bonus

- By using V, that is calculated from W, decide segments of the pictures.
  Hint: Normalize the rows of V. By using K-means algorithm, cluster the rows of the V. For further information:
  - `www.youtube.com/watch?v=0MQEt10e4NM`
  - `www.youtube.com/watch?v=4shfFAArxSc`

- Plot the segments.



Deadline: 5 June, 2017 10:00

References

[1] Shi, Jianbo, and Jitendra Malik. "Normalized cuts and image segmentation." *IEEE Transactions on pattern analysis and machine intelligence* 22.8 (2000): 888-905.