Intro to Machine Learning

Computer Assignment 1

Group Prof. Amini

Fall 2022

Abstract

In this exercise you will get hands-on experience with exploratory data analysis (EDA) and linear algebra in Python.
For further questions contact TA via telegram id: @Kiawesome

I. Bitcoin Market Analysis

In this problem we look at bitcoin market summary at late 2020 and try to denoise the time-series to obtain a more realistic graph.

Figure 1: Bitcoin price; data points are measured with time difference of 2 hours

Background from linear algebra

We know the least-squared answer for $Ax = y$ is obtained from

$$\hat{x} = \underset{x}{\operatorname{argmin}} ||Ax - y||^2$$

Now, assume that vector y is the price of bitcoin (over time), and vector x is the original (unnoisy) price tags that we are looking for. Vector v is the unwanted noise added to price tags.

$$y = x + v$$

or

$$y = Ix + v$$

Therefore, to find leas-squared answer, we must solve the following problem:

$$\min_x ||Ix - y||^2$$

Minimizing the mentioned function makes vector x converge to vector y. However, this condition is insufficient for denoising the data. As a result, we add a penalty term to also minimize the difference between two consecutive indices of x.
New problem to solve is equation below, in which λ is the penalty coefficient.

$$\min_x ||Ix - y||^2 + \lambda \sum_{i=1}^{n-1} (x_{i+1} - x_i)^2$$

We can reform the second term as $||Dx||^2$, such that:

$$D = \begin{bmatrix} 1 & -1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 1 & -1 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & -1 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 1 & -1 \end{bmatrix}$$

Finally our target is to minimize the following problem

$$\min_x ||Ix - y||^2 + ||\sqrt{\lambda} Dx||^2$$

We can also reform it as block matrices of the form below

$$\min_x || \begin{bmatrix} I \\ \sqrt{\lambda} D \end{bmatrix} x - \begin{bmatrix} y \\ 0 \end{bmatrix} ||^2$$

Problem

Solve the aformentioned problem and denoise the given array of price tags. Plot the time-series for at least 4 different values of λ. How does the parameter λ change the plots? For which value of λ the data looks more realistic (denoised)?

II. Hutchinson's Trace Estimator

Background

Hutchinson's estimator is a simple way to obtain a stochastic estimate of the trace of a matrix. In some cases, it may be expensive to evaluate the matrix A, but we may be able to cheaply evaluate matrix-vector products Av (click to find out more).
Suppose v is a random vector such that $\mathbb{E}[vv^T] = I$. In this case, we can create a Monte Carlo approximation to $tr(A)$ using the following identity:

$$tr(A) = tr(A\mathbb{E}[vv^T]) = \mathbb{E}[tr(Avv^T)] = \mathbb{E}[tr(v^TAv)]$$

For calculating $\mathbb{E}[tr(v^TAv)]$, we can rewrite

$$\mathbb{E}[tr(v^TAv)] = \frac{1}{m}\sum_{i=1}^{m} v_i^T A v_i = \frac{1}{m} tr(V^TAV)$$

Therefore Hutchinson's trace estimator is obtained from equation below

$$H_m(A) = \frac{1}{m} tr(V^TAV)$$

where

$$V = [v_1|v_2|\ldots|v_m]$$

Problem

(a) Complete the two functions for Hutchinson's estimator. One for v_i as a random sign vector, and one for v_i as a standard normal random vector.

(b) For $m = 10000$, run the test & evaluate cell.

(c) Plot the histograms of relative error counts that fall in different intervals. Plot one histogram for **each** function. An instance of one histogram is shown below:

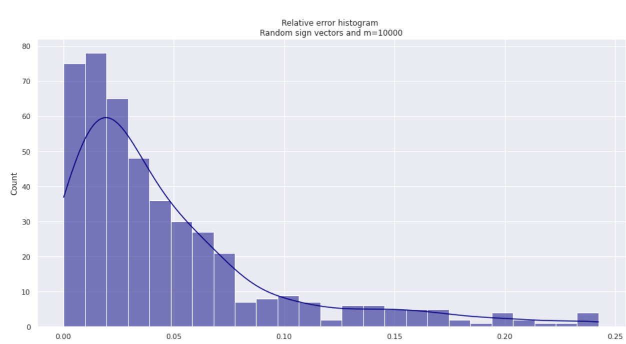

Figure 2: Histogram for random sign vectors and $m = 10000$

NOTE

- Do not import new packages, you are expected to solve the problems with the provided libraries.
- Feel free to surf the internet and learn more about the required items. However, any duplication (copy-and-paste) will set your score equal to zero.