Homework Set 2, CPSC 8420, Fall 2024

Matthew Collins

Due 10/28/2024, Monday, 11:59PM EST

Problem 1

For Principle Component Analysis (PCA), from the perspective of maximizing variance (assume the data is already self-centered)

- show that the first column of $\mathbf{U}$, where $[\mathbf{U}, \mathbf{S}] = svd(\mathbf{X}^T\mathbf{X})$ will maximize $\|\mathbf{X}\boldsymbol{\phi}\|_2^2$, $s.t.$ $\|\boldsymbol{\phi}\|_2 = 1$. (Note: you need prove why it is optimal than any other reasonable combinations of $\mathbf{U}_i$, say $\hat{\boldsymbol{\phi}} = 0.8 * \mathbf{U}(:,1) + 0.6 * \mathbf{U}(:,2)$ which also satisfies $\|\hat{\boldsymbol{\phi}}\|_2 = 1$.)
- show that the solution is not unique, say if $\boldsymbol{\phi}$ is the optimal solution, so is $-\boldsymbol{\phi}$.
- show that first r columns of $\mathbf{U}$, where $[\mathbf{U}, \mathbf{S}] = svd(\mathbf{X}^T\mathbf{X})$ maximize $\|\mathbf{X}\mathbf{W}\|_F^2$, $s.t.$ $\mathbf{W}^T\mathbf{W} = \mathbf{I}_r$.
- Assume the singular values are all different in $\mathbf{S}$, then how many possible different $\mathbf{W}$'s will maximize the objective above?

Answer

Part (a)

Show that the first column of $\mathbf{U}$, where $[\mathbf{U}, \mathbf{S}] = \text{svd}(\mathbf{X}^T\mathbf{X})$, maximizes $\|\mathbf{X}\boldsymbol{\phi}\|_2^2$ subject to $\|\boldsymbol{\phi}\|_2 = 1$.

Proof:

The variance of the data when projected onto a unit vector $\boldsymbol{\phi}$ is

$$\|\mathbf{X}\boldsymbol{\phi}\|_2^2 = \boldsymbol{\phi}^T\mathbf{X}^T\mathbf{X}\boldsymbol{\phi} \quad \text{subject to} \quad \|\boldsymbol{\phi}\|_2 = 1$$

$\mathbf{X}^T\mathbf{X}$ is a symmetric matrix, decompose it as

$$\mathbf{X}^T\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{U}^T,$$

where $\mathbf{U}$ is an orthogonal matrix and $\mathbf{S}$ is a diagonal matrix containing the singular values

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \geq 0.$$

Substituting into the objective function

$$\boldsymbol{\phi}^T\mathbf{X}^T\mathbf{X}\boldsymbol{\phi} = \boldsymbol{\phi}^T\mathbf{U}\mathbf{S}\mathbf{U}^T\boldsymbol{\phi}.$$

Let $\boldsymbol{\psi} = \mathbf{U}^T\boldsymbol{\phi}$. Since $\mathbf{U}$ is orthogonal, $\|\boldsymbol{\psi}\|_2 = \|\boldsymbol{\phi}\|_2 = 1$. Thus, the expression becomes

$$\boldsymbol{\phi}^T\mathbf{U}\mathbf{S}\mathbf{U}^T\boldsymbol{\phi} = \boldsymbol{\psi}^T\mathbf{S}\boldsymbol{\psi}.$$

The quantity $\boldsymbol{\psi}^T\mathbf{S}\boldsymbol{\psi}$ is maximized when $\boldsymbol{\psi}$ aligns with the eigenvector corresponding to the largest singular value σ_1. Therefore, the maximum is achieved when $\boldsymbol{\phi} = \mathbf{U}(:, 1)$.

$\hat{\boldsymbol{\phi}} = 0.8\mathbf{U}(:, 1) + 0.6\mathbf{U}(:, 2)$ satisfies $\|\hat{\boldsymbol{\phi}}\|_2 = 1$, the contribution from σ_2 will reduce the value of $\boldsymbol{\phi}^T\mathbf{X}^T\mathbf{X}\boldsymbol{\phi}$ since $\sigma_1 \geq \sigma_2$. Hence, $\mathbf{U}(:, 1)$ is the optimal choice.

—

Part (b)

Show that the solution is not unique if $\boldsymbol{\phi}$ is an optimal solution, then $-\boldsymbol{\phi}$ is also optimal.

Proof:

Consider the objective function

$$\|\mathbf{X}\boldsymbol{\phi}\|_2^2 = \boldsymbol{\phi}^T\mathbf{X}^T\mathbf{X}\boldsymbol{\phi}$$

Take $-\boldsymbol{\phi}$

$$\|\mathbf{X}(-\boldsymbol{\phi})\|_2^2 = (-\boldsymbol{\phi})^T\mathbf{X}^T\mathbf{X}(-\boldsymbol{\phi}) = \boldsymbol{\phi}^T\mathbf{X}^T\mathbf{X}\boldsymbol{\phi}.$$

The value of the objective function remains unchanged, which implies that both $\boldsymbol{\phi}$ and $-\boldsymbol{\phi}$ are optimal solutions.

—

Part (c)

Show that the first r columns of $\mathbf{U}$, where $[\mathbf{U}, \mathbf{S}] = \text{svd}(\mathbf{X}^T\mathbf{X})$, maximize $\|\mathbf{X}\mathbf{W}\|_F^2, s.t.\ \mathbf{W}^T\mathbf{W} = \mathbf{I}_r$.

Proof:

The objective function can be written as

$$\|\mathbf{X}\mathbf{W}\|_F^2 = \text{Tr}(\mathbf{W}^T\mathbf{X}^T\mathbf{X}\mathbf{W}).$$

By SVD $\mathbf{X}^T\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{U}^T$,

$$\text{Tr}(\mathbf{W}^T\mathbf{U}\mathbf{S}\mathbf{U}^T\mathbf{W}).$$

Let $\mathbf{V} = \mathbf{U}^T\mathbf{W}$. Since $\mathbf{U}$ is orthogonal, $\mathbf{V}^T\mathbf{V} = \mathbf{I}_r$. The objective becomes

$$\text{Tr}(\mathbf{V}^T\mathbf{S}\mathbf{V}).$$

The trace $\text{Tr}(\mathbf{V}^T\mathbf{S}\mathbf{V})$ is maximized when $\mathbf{V}$ aligns with the first r columns of $\mathbf{U}$, corresponding to the largest r singular values $\sigma_1, \ldots, \sigma_r$. Therefore, $\mathbf{W} = \mathbf{U}(:, 1:r)$ is the optimal solution.

—

Part (d)

Assuming the singular values in $\mathbf{S}$ are different, how many possible different $\mathbf{W}$'s will maximize the objective?

Answer:

Given that the singular values in $\mathbf{S}$ are all distinct, the optimization problem of maximizing $\|\mathbf{XW}\|_F^2$ under the constraint $\mathbf{W}^T\mathbf{W} = \mathbf{I}_r$ has solutions that can be expressed as:

$$\mathbf{W} = \mathbf{U}_r\mathbf{Q},$$

where $\mathbf{U}_r$ consists of the first r columns of $\mathbf{U}$, and $\mathbf{Q}$ is any orthogonal $r \times r$ matrix such that $\mathbf{Q}^T\mathbf{Q} = \mathbf{I}_r$. The set of all such orthogonal matrices $\mathbf{Q}$ forms the orthogonal group $O(r)$.

Problem 2

Given matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$ (assume each column is centered already), where n denotes sample size while p feature size. To conduct PCA, we need find eigenvectors to the largest eigenvalues of $\mathbf{X}^T\mathbf{X}$, where usually the complexity is $\mathcal{O}(p^3)$. Apparently when $n \ll p$, this is not economic when p is large. Please consider conducting PCA based on $\mathbf{X}\mathbf{X}^T$ and obtain the eigenvectors for $\mathbf{X}^T\mathbf{X}$ accordingly and use experiment to demonstrate the acceleration.

Answer

To efficiently perform PCA when $n \ll p$, the eigenvectors of $\mathbf{X}^T\mathbf{X}$ can be obtained from the eigenvectors of $\mathbf{X}\mathbf{X}^T$. This is computationally cheaper when p is large and n is relatively small.

By decomposition of $\mathbf{X}\mathbf{X}^T$,

$$\mathbf{X}\mathbf{X}^T = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T,$$

The eigenvectors of $\mathbf{X}^T\mathbf{X}$ can be obtained from $\mathbf{U}$.

$$\mathbf{V} = \mathbf{X}^T\mathbf{U}\mathbf{\Lambda}^{-\frac{1}{2}}$$

Python code was created below to expiriment and demonstrate the differences between eigenvalues of $\mathbf{X}^T\mathbf{X}$ and $\mathbf{X}\mathbf{X}^T$. The python code iterates through different sample sizes (n) and different feature sizes (p) and keeps track of the time it takes to comput the eigenvalues using both $\mathbf{X}^T\mathbf{X}$ and $\mathbf{X}\mathbf{X}^T$. From the experiment, we expect that the indirect method using $\mathbf{X}\mathbf{X}^T$ will be significantly faster than the direct method when $n \ll p$. The computational complexity of finding eigenvectors of an $n \times n$ matrix $\mathbf{X}\mathbf{X}^T$ is $\mathcal{O}(n^3)$, which is much more efficient when $n \ll p$. As can be seen in Figure 1 and Figure 2, for cases where $n \ll p$, the dramatic speedup of using $\mathbf{X}\mathbf{X}^T$ can be seen that appears to be very close to $\mathcal{O}(n^3)$.

Python Code

```
import numpy as np
import time
import pandas as pd
import matplotlib.pyplot as plt

n_values = [50, 100, 200]
p_values = [500, 1000, 2000]

results = []

for n in n_values:
 for p in p_values:
  np.random.seed(0)
  X = np.random.randn(n, p)

  # Direct Eigenvalue Decomposition of X^T X
  start_time = time.time()
```

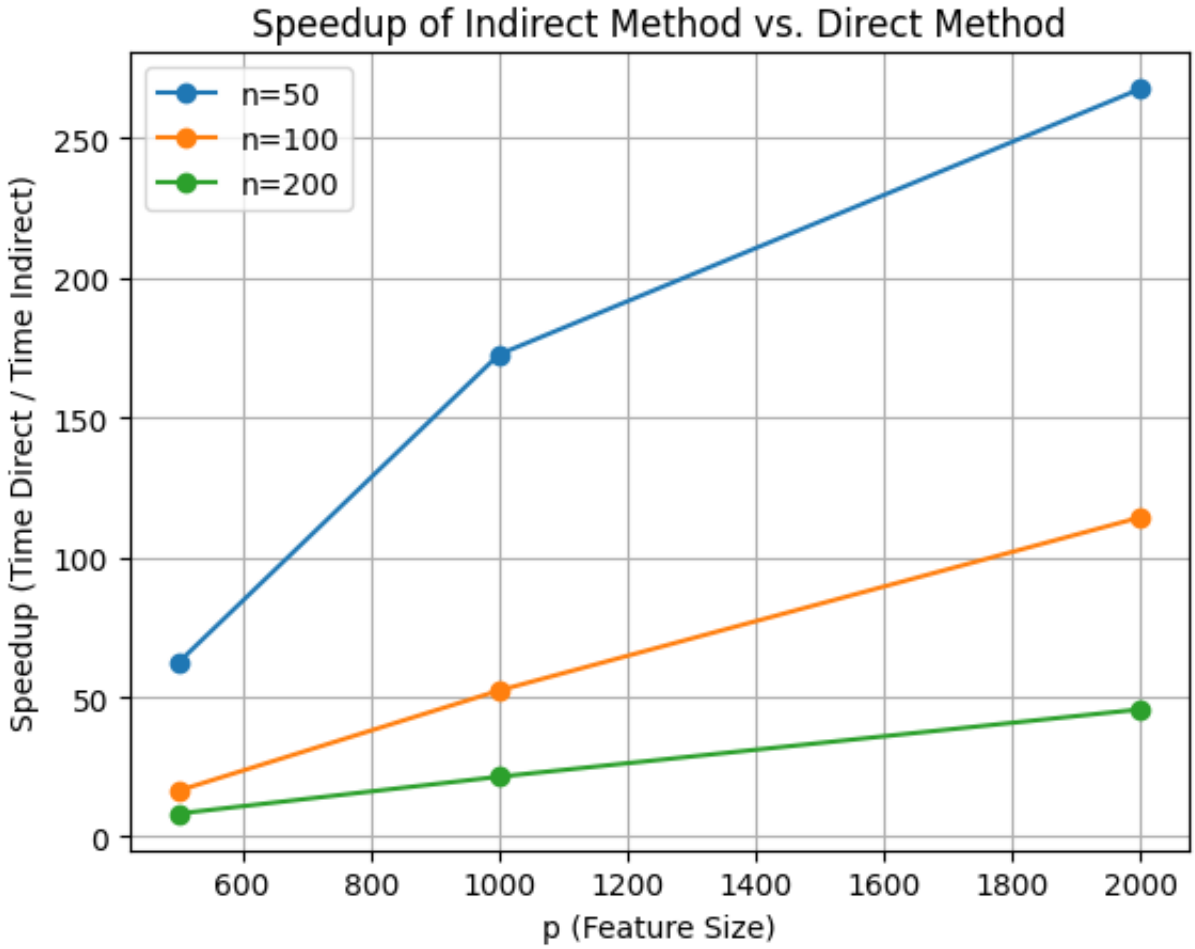

Figure 1: Speedup Compared to Feature Size

```
(hw2) (base) PS E:\CPSC 8420 Advances Machine Learning\HW2_24F> python .\problem_2.py
     n     p  time_direct  time_indirect     speedup
0   50   500     0.063036       0.001013   62.224988
1   50  1000     0.189120       0.001096  172.552954
2   50  2000     0.554991       0.002073  267.686510
3  100   500     0.066495       0.004047   16.430162
4  100  1000     0.211799       0.004049   52.311152
5  100  2000     0.582603       0.005095  114.342471
6  200   500     0.067038       0.008113    8.262592
7  200  1000     0.185035       0.008619   21.468630
8  200  2000     0.558259       0.012244   45.594586
```

Figure 2: Speedup Table

```
  XtX = np.dot(X.T, X)
  _, V1 = np.linalg.eigh(XtX)
  time_direct = time.time() - start_time

  # Indirect (Efficient Computation) Eigenvalue Decomposition using X X^T
  start_time = time.time()
  XXt = np.dot(X, X.T)
  D, U = np.linalg.eigh(XXt)
  V2 = np.dot(X.T, U) / np.sqrt(D)  # Normalize eigenvectors
  time_indirect = time.time() - start_time

  epsilon = 1e-10
  speedup = time_direct / (time_indirect + epsilon)

  results.append({
   'n': n,
   'p': p,
   'time_direct': time_direct,
   'time_indirect': time_indirect,
   'speedup': speedup
  })

results_df = pd.DataFrame(results)
print(results_df)

for n in n_values:
 subset = results_df[results_df['n'] == n]
 plt.plot(subset['p'], subset['speedup'], marker='o', label=f'n={n}')

plt.xlabel('p (Feature Size)')
plt.ylabel('Speedup (Time Direct / Time Indirect)')
plt.title('Speedup of Indirect Method vs. Direct Method')
plt.legend()
plt.grid(True)
plt.show()
```

Problem 3

Let $\theta^* \in \mathbb{R}^d$ be the ground truth linear model parameter and $\mathbf{X} \in \mathbb{R}^{N \times d}$ be the observing matrix and each column of $\mathbf{X}$ is independent. Assume the linear model is $\mathbf{y} = \mathbf{X}\theta^* + \epsilon$ where ϵ follows $Gaussian(0, \sigma^2\mathbf{I})$. Assume $\hat{\theta} = \arg\min_{\theta} \|\mathbf{X}\theta - \mathbf{y}\|^2$.

- Please show that $\mathbf{X}^T\mathbf{X}$ is invertible.
- Show that $MSE(\theta^*, \hat{\theta}) := E_\epsilon\{\|\theta^* - \hat{\theta}\|^2\} = \sigma^2 trace((\mathbf{X}^T\mathbf{X})^{-1})$
- Show that as N increases, MSE decreases. (hint: make use of 'Woodbury matrix identity')

Answer

Part (a)

Show that $\mathbf{X}^T\mathbf{X}$ is invertible.

Proof:

Each column of $\mathbf{X}$ is independent. The columns of $\mathbf{X}$ form a linearly independent set.

$$\text{rank}(\mathbf{X}) = d, \quad \text{where } d \text{ is the number of columns in } \mathbf{X}.$$

Therefore, if $\mathbf{X}$ has full column rank, then $\mathbf{X}^T\mathbf{X}$ is a $d \times d$ matrix. Since $\mathbf{X}$ has full column rank, $\mathbf{X}^T\mathbf{X}$ is a $d \times d$ matrix, $\text{rank}(\mathbf{X}^T\mathbf{X}) = d$, and $\mathbf{X}^T\mathbf{X}$ is invertible.

Part (b)

Proof: The estimator $\hat{\theta}$ is given by the least squares solution:

$$\hat{\theta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}.$$

Substituting $\mathbf{y} = \mathbf{X}\theta^* + \epsilon$,

$$\hat{\theta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T(\mathbf{X}\theta^* + \epsilon).$$

Simplifying,

$$\hat{\theta} = \theta^* + (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\epsilon.$$

The mean squared error (MSE) is given by:

$$\text{MSE}(\theta^*, \hat{\theta}) = \mathbb{E}_\epsilon\left\{\|\theta^* - \hat{\theta}\|^2\right\} = \mathbb{E}_\epsilon\left\{\|(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\epsilon\|^2\right\}.$$

Since $\epsilon \sim \mathcal{N}(0, \sigma^2\mathbf{I})$,

$$\mathbb{E}_\epsilon\left\{\epsilon\epsilon^T\right\} = \sigma^2\mathbf{I}.$$

$$\mathbb{E}_\epsilon\left\{(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\epsilon\epsilon^T\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\right\} = \sigma^2(\mathbf{X}^T\mathbf{X})^{-1}.$$

$$\text{MSE}(\theta^*, \hat{\theta}) = \sigma^2\text{trace}((\mathbf{X}^T\mathbf{X})^{-1}).$$

—

Part (c)

Proof: The Woodbury matrix identity states that for matrices $\mathbf{A}$, $\mathbf{U}$, $\mathbf{C}$, and $\mathbf{V}$ of appropriate dimensions:

$$(\mathbf{A}+\mathbf{UCV})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{U}(\mathbf{C}^{-1}+\mathbf{VA}^{-1}\mathbf{U})^{-1}\mathbf{VA}^{-1}.$$

where $\mathbf{A} \in \mathbb{R}^{d\times d}$, $\mathbf{U} \in \mathbb{R}^{d\times k}$, $\mathbf{C} \in \mathbb{R}^{k\times k}$, and $\mathbf{V} \in \mathbb{R}^{k\times d}$. Given $\mathbf{X} \in \mathbb{R}^{N\times d}$, where N is the number of observations and d is the number of features. Suppose we have an initial sample matrix $\mathbf{X}_0 \in \mathbb{R}^{N_0\times d}$ and add a new sample $\mathbf{x} \in \mathbb{R}^{1\times d}$. The updated observation matrix becomes:

$$\mathbf{X} = \begin{pmatrix}\mathbf{X}_0 \\ \mathbf{x}\end{pmatrix} \in \mathbb{R}^{(N_0+1)\times d}.$$

$\mathbf{X}^T\mathbf{X}$ is updated as:

$$\mathbf{X}^T\mathbf{X} = \mathbf{X}_0^T\mathbf{X}_0 + \mathbf{x}^T\mathbf{x}.$$

Let $\mathbf{A} = \mathbf{X}_0^T\mathbf{X}_0$, $\mathbf{U} = \mathbf{x}^T$, $\mathbf{C} = 1$, and $\mathbf{V} = \mathbf{x}$. Applying the Woodbury matrix identity, we get:

$$(\mathbf{X}_0^T\mathbf{X}_0 + \mathbf{x}^T\mathbf{x})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{U}(1+\mathbf{VA}^{-1}\mathbf{U})^{-1}\mathbf{VA}^{-1}.$$

Simplifying:

$$(\mathbf{X}_0^T\mathbf{X}_0 + \mathbf{x}^T\mathbf{x})^{-1} = (\mathbf{X}_0^T\mathbf{X}_0)^{-1} - \frac{(\mathbf{X}_0^T\mathbf{X}_0)^{-1}\mathbf{x}^T\mathbf{x}(\mathbf{X}_0^T\mathbf{X}_0)^{-1}}{1+\mathbf{x}(\mathbf{X}_0^T\mathbf{X}_0)^{-1}\mathbf{x}^T}.$$

The term:

$$\frac{(\mathbf{X}_0^T\mathbf{X}_0)^{-1}\mathbf{x}^T\mathbf{x}(\mathbf{X}_0^T\mathbf{X}_0)^{-1}}{1+\mathbf{x}(\mathbf{X}_0^T\mathbf{X}_0)^{-1}\mathbf{x}^T}$$

is a positive semi-definite matrix. Subtracting this term from $(\mathbf{X}_0^T\mathbf{X}_0)^{-1}$ decreases the overall trace. Therefore, as we add more samples (i.e., as N increases), trace$((\mathbf{X}^T\mathbf{X})^{-1})$ decreases, leading to a decrease in MSE.

References

[1] Towards Data Science, "Principal Component Analysis (Part 1) — The Different Formulations," *Towards Data Science*, Aug. 30, 2019. [Online]. Available: `https://towardsdatascience.com/principal-component-analysis-part-1-the-different-formulations-6508f63a5553`.

[2] A. Alekhyo, "Computational Complexity of PCA," *Medium*, Jan. 10, 2021. [Online]. Available: `https://alekhyo.medium.com/computational-complexity-of-pca-4cb61143b7e5`.

[3] Stats Stack Exchange, "Is MSE decreasing with increasing number of explanatory variables?" *Stats Stack Exchange*, Dec. 18, 2017. [Online]. Available: `https://stats.stackexchange.com/questions/306267/is-mse-decreasing-with-increasing-number-of-explanatory-variables`.

[4] Wikipedia, "Mean squared error," *Wikipedia, The Free Encyclopedia*, Mar. 1, 2024. [Online]. Available: `https://en.wikipedia.org/wiki/Mean_squared_error`.

[5] Wikipedia, "Woodbury matrix identity," *Wikipedia, The Free Encyclopedia*, Mar. 1, 2024. [Online]. Available: `https://en.wikipedia.org/wiki/Woodbury_matrix_identity`.