

## CS/DSC/AI 391L: Machine Learning

### Homework 3 - Theory

Lecture: Prof. Adam Klivans

Keywords: SVD, PCA

Name: Patrick Brown

EID: pbb468

**Instructions:** Please either typeset your answers (L<sup>A</sup>T<sub>E</sub>X recommended) or write them very clearly and legibly and scan them, and upload the PDF on edX. Legibility and clarity are critical for fair grading.

1. Let  $A$  be an  $m \times d$  matrix, and let  $X = AA^T$ . Assume that  $X$  has  $d$  distinct, non-zero eigenvalues. Assume that  $m \gg d$ . In order to find the eigendecomposition of  $X$ , we will need to find the eigendecomposition of an  $m \times m$  matrix. Since  $m$  is much larger than  $d$ , this is slow. Give an algorithm for finding the eigenvectors and eigenvalues of  $X$  that only requires computing the eigendecomposition of a  $d \times d$  matrix. You can use simple matrix operations and assume that you have an eigendecomposition “black box” subroutine, but avoid using the SVD as a black box.

### Solution to Problem 1

To efficiently find the eigenvectors and eigenvalues of the matrix  $X = AA^T$  without directly computing the eigendecomposition of the large  $m \times m$  matrix  $X$ , you can leverage the relationship between  $X$  and the smaller  $d \times d$  matrix  $Y = A^T A$ . Here’s the step-by-step algorithm:

#### 1. Compute $Y = A^T A$ :

- $Y$  is a  $d \times d$  matrix, which is much smaller than  $X$  since  $m \gg d$ .
- This computation is efficient because it involves multiplying a  $d \times m$  matrix  $A^T$  with an  $m \times d$  matrix  $A$ .

#### 2. Perform Eigendecomposition of $Y$ :

- Use your eigendecomposition black-box subroutine to find the eigenvalues  $\lambda_i$  and corresponding eigenvectors  $v_i$  of  $Y$ :  $Yv_i = \lambda_i v_i$ ,  $i = 1, 2, \dots, d$
- Since  $Y$  is symmetric and positive definite (because  $X$  has  $d$  distinct, non-zero eigenvalues), all eigenvalues  $\lambda_i$  are positive and eigenvectors  $v_i$  can be chosen to be orthonormal.

#### 3. Compute the Eigenvectors of $X$ :

- For each eigenvalue  $\lambda_i$  and eigenvector  $v_i$  of  $Y$ , compute the corresponding eigenvector  $u_i$  of  $X$  using:  $u_i = \frac{1}{\sqrt{\lambda_i}} Av_i$
- This shows that  $u_i$  satisfies the eigenvalue equation  $Xu_i = \lambda_i u_i$ .

#### Summary of the Algorithm:

**Input:**  $A \in \mathbb{R}^{m \times d}$

**Output:** Eigenvalues  $\lambda_i$  and eigenvectors  $u_i$  of  $X = AA^T$

**Steps:**

1. Compute  $Y = A^T A$ .
2. Compute the eigenvalues  $\lambda_i$  and eigenvectors  $v_i$  of  $Y$ .
3. For each  $i$ , compute  $u_i = \frac{1}{\sqrt{\lambda_i}} Av_i$ .

4. The pairs  $(\lambda_i, u_i)$  are the eigenvalues and eigenvectors of  $X$ .

**Advantages of This Approach:**

- **Efficiency:** Only requires eigendecomposition of a  $d \times d$  matrix instead of an  $m \times m$  matrix.
- **Avoids SVD:** Does not rely on singular value decomposition as a black box.
- **Simplicity:** Utilizes basic linear algebra operations and a standard eigendecomposition subroutine.

By following this algorithm, you can efficiently compute the eigendecomposition of  $X = AA^T$  without the computational burden associated with large matrices.

## Problem 2

In this problem we explore some relationships between SVD, PCA and linear regression.

1. True or false: linear regression is primarily a technique of supervised learning, i.e. where we are trying to fit a function to labeled data.

**Solution:** True.

Linear regression is indeed primarily a technique of supervised learning. In linear regression, we have a set of input features (independent variables) and corresponding output values (dependent variable). The goal is to find a linear function that best fits this labeled data, allowing us to predict the output for new, unseen inputs. This process of learning from labeled examples (where we have both inputs and their corresponding outputs) is the essence of supervised learning.

Key points:

- We have labeled data: pairs of inputs (x) and outputs (y).
  - We're trying to fit a function ( $y = wx + b$ ) to this data.
  - The learned function can then be used to make predictions on new, unseen data.
  - The learning process involves minimizing the error between predicted and actual output values, which requires knowing the true output values (labels).
2. True or false: PCA is primarily a technique of unsupervised learning, i.e. where we are trying to find structure in unlabeled data.

**Solution:** True.

Principal Component Analysis (PCA) is indeed primarily a technique of unsupervised learning. PCA is used to find patterns and structure in data without the need for labeled outputs. It aims to reduce the dimensionality of the data while preserving as much variance as possible.

Key points:

- PCA works with unlabeled data: it doesn't require output labels for the input features.
  - The goal is to find the directions (principal components) of maximum variance in the data.
  - PCA discovers structure and patterns in the data without guidance from labeled outcomes.
  - It can be used for dimensionality reduction, feature extraction, and data visualization without supervision.
3. True or false: SVD is primarily an operation on a dataset whereas PCA is primarily an operation on a matrix.

**Solution:** False.

This statement is incorrect. Both Singular Value Decomposition (SVD) and Principal Component Analysis (PCA) are primarily operations on matrices, not datasets directly.

Key points:

- SVD is a matrix factorization method that decomposes a matrix into three matrices:  $A = U\Sigma V^T$ .
  - PCA is often implemented using SVD and operates on the covariance matrix of the data.
  - Both SVD and PCA work with matrices that represent datasets, but the operations themselves are matrix operations.
  - While PCA is often used for data analysis and dimensionality reduction, the core algorithm operates on a matrix (usually the covariance matrix or directly on the centered data matrix).
  - SVD can be applied to any matrix, not necessarily representing a dataset, and is used in various applications beyond data analysis.
4. A common problem in linear regression is multicollinearity, where the input variables are themselves linearly dependent. For example, imagine a healthcare data set where height is measured both in inches and centimetres. This is a problem because there may now be multiple  $w$  satisfying  $y = w \cdot x$ . Explain how you could use a preprocessing step to solve this problem.

**Solution:** To address the problem of multicollinearity, we can use Principal Component Analysis (PCA) as a preprocessing step:

- **Apply PCA:** Transform the input space into orthogonal principal components, effectively removing linear dependencies.
- **Dimensionality Reduction:** Keep only the top  $k$  principal components that explain most of the variance in the data.
- **Benefits:** This approach not only solves multicollinearity but also reduces the dimensionality of the input space, potentially improving model performance and interpretability.

By using PCA, we create new uncorrelated features from the original linearly dependent variables, addressing multicollinearity while preserving the important information in the data.