Mathematical Foundation of Data Science **2025**

Part 1.2 Linear Algebra Applications

Scriber: Yingyu Liang

1 Linear Regression

Suppose we are given a set of data points $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ where $\boldsymbol{x}_i \in \mathbb{R}^d$ and $y \in \mathbb{R}$. We would like to learn a linear regression model to fit the data:

$$y = \boldsymbol{\theta}_x^\top \boldsymbol{x} + \theta_0$$

where $\boldsymbol{\theta}_x \in \mathbb{R}^d, \theta \in \mathbb{R}$.

First, we conveniently rewrite the problem. Let $\bar{\boldsymbol{x}}_i \in \mathbb{R}^{d+1}$ be the concatenation of $\boldsymbol{x}_i$ and 1, and $\boldsymbol{\theta} \in \mathbb{R}^{d+1}$ be the concatenation of $\boldsymbol{\theta}_x$ and θ_0. Then the model becomes

$$y = \boldsymbol{\theta}^\top \bar{\boldsymbol{x}}.$$

Next, we consider learning a model by optimizing the mean square error (MSE):

$$\widehat{\boldsymbol{\theta}} = \operatorname*{argmin}_{\boldsymbol{\theta}} \frac{1}{n} \sum_{i=1}^n (y_i - \boldsymbol{\theta}^\top \bar{\boldsymbol{x}}_i)^2.$$

Let $\boldsymbol{X} \in \mathbb{R}^{n \times (d+1)}$ be a matrix with $\bar{\boldsymbol{x}}_i$ as the i-th row, and let $\boldsymbol{y} \in \mathbb{R}^n$ be a vector with y_i as the i-th entry. Then the MSE objective can be rewritten in the matrix form:

$$\widehat{\boldsymbol{\theta}} = \operatorname*{argmin}_{\boldsymbol{\theta}} \frac{1}{n} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\theta}\|^2 = \operatorname*{argmin}_{\boldsymbol{\theta}} (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\theta})^\top (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\theta})$$

where $\|\boldsymbol{v}\|^2 = \boldsymbol{v}^\top \boldsymbol{v}$ denotes the Euclidean norm.

Now we do the optimization. Rewrite the objective as:

$$(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\theta})^\top (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\theta}) = \boldsymbol{y}^\top \boldsymbol{y} - 2\boldsymbol{\theta}^\top \boldsymbol{X}^\top \boldsymbol{y} + \boldsymbol{\theta}^\top \boldsymbol{X}^\top \boldsymbol{X}\boldsymbol{\theta}.$$

Take the gradient with respect to $\boldsymbol{\theta}$ and set it to the zero vector:

$$\begin{aligned} -2\boldsymbol{X}^\top \boldsymbol{y} + 2\boldsymbol{X}^\top \boldsymbol{X}\boldsymbol{\theta} &= 0. \\ \boldsymbol{X}^\top \boldsymbol{X}\boldsymbol{\theta} &= \boldsymbol{X}^\top \boldsymbol{y}. \end{aligned}$$

Case 1. *Assuming $\boldsymbol{X}^\top \boldsymbol{X}$ is invertible*, we obtain the solution

$$\widehat{\boldsymbol{\theta}} = (\boldsymbol{X}^\top \boldsymbol{X})^{-1} \boldsymbol{X}^\top \boldsymbol{y}.$$

Case 2. *If $\boldsymbol{X}^\top \boldsymbol{X}$ is not invertible*, i.e., $\text{rank}(\boldsymbol{X}^\top \boldsymbol{X}) < d$, we can use its pseudo-inverse. Suppose $\text{rank}(\boldsymbol{X}) = r < d$. Consider the (truncated) SVD $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$, where $\boldsymbol{\Sigma} \in \mathbb{R}^{r\times r}$ is the matrix removing the 0 singular values, and $\boldsymbol{U} \in \mathbb{R}^{n\times r}, \boldsymbol{V} \in \mathbb{R}^{d\times r}$ are those matrices removing the columns associated with 0 singular values. Then the pseudo-inverse $\boldsymbol{X}^\dagger = \boldsymbol{U}\boldsymbol{\Sigma}^{-1}\boldsymbol{V}^\top$. And $\boldsymbol{X}^\top\boldsymbol{X} = \boldsymbol{V}\boldsymbol{\Sigma}^2\boldsymbol{V}^\top$ and the pseudo-inverse $(\boldsymbol{X}^\top\boldsymbol{X})^\dagger = \boldsymbol{V}(\boldsymbol{\Sigma}^2)^{-1}\boldsymbol{V}^\top$. Then we can obtain

$$\widehat{\boldsymbol{\theta}} = (\boldsymbol{X}^\top\boldsymbol{X})^\dagger\boldsymbol{X}^\top\boldsymbol{y}.$$

It turns out $(\boldsymbol{X}^\top\boldsymbol{X})^\dagger\boldsymbol{X}^\top = \boldsymbol{X}^\dagger$, since

$$\begin{aligned}(\boldsymbol{X}^\top\boldsymbol{X})^\dagger\boldsymbol{X}^\top &= \boldsymbol{V}(\boldsymbol{\Sigma}^2)^{-1}\boldsymbol{V}^\top\boldsymbol{X}^\top \\ &= \boldsymbol{V}(\boldsymbol{\Sigma}^2)^{-1}\boldsymbol{V}^\top\boldsymbol{V}\boldsymbol{\Sigma}\boldsymbol{U}^\top \\ &= \boldsymbol{V}(\boldsymbol{\Sigma}^2)^{-1}\boldsymbol{V}^\top\boldsymbol{V}\boldsymbol{\Sigma}\boldsymbol{U}^\top \\ &= \boldsymbol{V}\boldsymbol{\Sigma}^{-1}\boldsymbol{U}^\top \\ &= \boldsymbol{X}^\dagger.\end{aligned}$$

So when $\boldsymbol{X}^\top\boldsymbol{X}$ is not invertible, we can use

$$\widehat{\boldsymbol{\theta}} = \boldsymbol{X}^\dagger\boldsymbol{y}.$$

Ridge Regression. However, in practice, we typically consider the regularized version of linear regression (for statistical reasons we will learn later):

$$\widehat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\text{argmin}}\ \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\theta}\|^2 + \lambda\|\boldsymbol{\theta}\|^2$$

where $\lambda > 0$ is the regularization coefficient. It turns out that this also helps mitigate the invertibility issue.

Taking the derivative again and setting it to 0:

$$\begin{aligned}-2\boldsymbol{X}^\top\boldsymbol{y} + 2\boldsymbol{X}^\top\boldsymbol{X}\boldsymbol{\theta} + 2\lambda\boldsymbol{\theta} &= 0 \\ (\boldsymbol{X}^\top\boldsymbol{X} + \lambda\boldsymbol{I})\boldsymbol{\theta} &= \boldsymbol{X}^\top\boldsymbol{y} \\ \widehat{\boldsymbol{\theta}} &= (\boldsymbol{X}^\top\boldsymbol{X} + \lambda\boldsymbol{I})^{-1}\boldsymbol{X}^\top\boldsymbol{y}.\end{aligned}$$

Note that $\boldsymbol{X}^\top\boldsymbol{X} + \lambda\boldsymbol{I}$ is always invertible. To see this, consider the (full) SVD $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$, where $\boldsymbol{U} \in \mathbb{R}^{n\times n}, \boldsymbol{\Sigma} \in \mathbb{R}^{n\times d}, \boldsymbol{V} \in \mathbb{R}^{d\times d}$. Then

$$\begin{aligned}\boldsymbol{X}^\top\boldsymbol{X} + \lambda\boldsymbol{I} &= (\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top)^\top(\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top) + \lambda\boldsymbol{I} \\ &= \boldsymbol{V}\boldsymbol{\Sigma}^\top\boldsymbol{\Sigma}\boldsymbol{V}^\top + \lambda\boldsymbol{I} \\ &= \boldsymbol{V}(\boldsymbol{\Sigma}^\top\boldsymbol{\Sigma} + \lambda\boldsymbol{I})\boldsymbol{V}^\top.\end{aligned}$$

Here $\boldsymbol{\Sigma}^\top\boldsymbol{\Sigma} + \lambda\boldsymbol{I}$ is a $d\times d$ diagonal matrix with diagonal entries $\sigma_i^2 + \lambda > 0$, so it is invertible.

We can also see the form of the solution:

$$\begin{aligned}\widehat{\boldsymbol{\theta}} &= (\boldsymbol{X}^\top \boldsymbol{X} + \lambda \boldsymbol{I})^{-1} \boldsymbol{X}^\top \boldsymbol{y} \\ &= \boldsymbol{V}(\boldsymbol{\Sigma}^\top \boldsymbol{\Sigma} + \lambda \boldsymbol{I})^{-1} \boldsymbol{V}^\top (\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top)^\top \boldsymbol{y} \\ &= \boldsymbol{V}(\boldsymbol{\Sigma}^\top \boldsymbol{\Sigma} + \lambda \boldsymbol{I})^{-1} \boldsymbol{\Sigma}^\top \boldsymbol{U}^\top \boldsymbol{y}.\end{aligned}$$

Here $(\boldsymbol{\Sigma}^\top \boldsymbol{\Sigma} + \lambda \boldsymbol{I})^{-1} \boldsymbol{\Sigma}^\top \in \mathbb{R}^{d \times n}$ has diagonal entries $\frac{\sigma_i}{\sigma_i^2 + \lambda}$.

We can also use gradient descent or its variants to solve the optimization, which will be discussed in the later lectures.

2 Principal Component Analysis

Suppose we are given a set of data points $\{\boldsymbol{x}_i\}_{i=1}^n$ where $\boldsymbol{x}_i \in \mathbb{R}^d$. We would like to learn a lower-dimensional subspace (of dimension r) and represent the data points as the projections in the subspace (i.e., a point in $\mathbb{R}^r$). For convenience, we shall *assume the data points are centered* for the rest of this section. That is,

- **(Assumption)** The data points have zero mean: $\mu := \frac{1}{n} \sum_{i=1}^n \boldsymbol{x}_i = 0$.

If the mean $\mu \neq 0$, then we can centralize the data by subtracting the mean: $\boldsymbol{x}_i \leftarrow \boldsymbol{x}_i - \mu$.

Algorithm 1 Principal Component Analysis (PCA)

Require: $\{\boldsymbol{x}_i\}_{i=1}^n \subseteq \mathbb{R}^d$ such that $\sum_{i=1}^n \boldsymbol{x}_i = 0$, number of principal components $r \leq d$

Form the sample covariance matrix $\boldsymbol{S} := \frac{1}{n} \sum_{i=1}^n \boldsymbol{x}_i \boldsymbol{x}_i^\top$

Do eigendecomposition: $\boldsymbol{S} = \boldsymbol{U}\Lambda\boldsymbol{U}^\top$, where Λ is a diagonal matrix with eigenvalues on the diagonal, and the columns of $\boldsymbol{U} = [\boldsymbol{u}_1, \boldsymbol{u}_2, \cdots, \boldsymbol{u}_d]$ are the corresponding eigenvectors.

Suppose the eigenvalues are sorted in decreasing order: $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d$.

Form $\boldsymbol{U}_r = [\boldsymbol{u}_1, \boldsymbol{u}_2, \cdots, \boldsymbol{u}_r]$, and let $\widetilde{\boldsymbol{x}}_i = \boldsymbol{U}_r^\top \boldsymbol{x}_i = [\boldsymbol{u}_1^\top \boldsymbol{x}_i, \boldsymbol{u}_2^\top \boldsymbol{x}_i, \cdots, \boldsymbol{u}_r^\top \boldsymbol{x}_i]^\top$

Ensure: the principal components $\boldsymbol{U}_r$, and the new representations $\{\widetilde{\boldsymbol{x}}_i\}_{i=1}^n$

2.1 The Variance Preservation View

PCA can be justified in several ways. Here we consider the goal of finding a subspace to preserve the variance as much as possible.

Consider $r = 1$. Our goal is to find a 1-dimensional subspace (a line going through the origin) such that the projections of the data points on this line have as large a variance as possible, in the hope that this maximally preserves the distinction among the points.

A line going through the origin can be represented by a vector $\boldsymbol{w} \in \mathbb{R}^d$. The projection of a point $\boldsymbol{x}$ onto this line is then:

$$\boldsymbol{p} = \frac{\boldsymbol{w}^\top \boldsymbol{x}}{\|\boldsymbol{w}\|}$$

where $\|\boldsymbol{w}\| = \sqrt{\boldsymbol{w}^\top \boldsymbol{w}}$ is the Euclidean norm.

For simplicity, let us consider $\boldsymbol{w}$ with unit length. Then the projections of the data points are $\boldsymbol{w}^\top \boldsymbol{x}_i$. Since their mean is $\frac{1}{n}\sum_{i=1}^n \boldsymbol{w}^\top \boldsymbol{x}_i = \frac{1}{n}\boldsymbol{w}^\top(\sum_{i=1}^n \boldsymbol{x}_i) = 0$, their variance is

$$\frac{1}{n}\sum_{i=1}^n (\boldsymbol{w}^\top \boldsymbol{x}_i)^2 = \frac{1}{n}\sum_{i=1}^n \boldsymbol{w}^\top \boldsymbol{x}_i \boldsymbol{x}_i^\top \boldsymbol{w} = \boldsymbol{w}^\top \boldsymbol{S}\boldsymbol{w}$$

where $\boldsymbol{S} = \frac{1}{n}\sum_{i=1}^n \boldsymbol{x}_i \boldsymbol{x}_i^\top$ is the sample covariance.

Recall that our goal is to maximize the variance of the projections:

$$\begin{aligned} &\max_{\boldsymbol{w}} \ \boldsymbol{w}^\top \boldsymbol{S}\boldsymbol{w} \\ &\text{s.t. } \|\boldsymbol{w}\| = 1. \end{aligned}$$

This is exactly finding the first eigenvector of the sample covariance $\boldsymbol{S}$ (recall the proof of the eigendecomposition). We can also show it by solving the optimization via the Lagrangian:

$$\boldsymbol{w}^\top \boldsymbol{S}\boldsymbol{w} + \lambda(1 - \boldsymbol{w}^\top \boldsymbol{w}).$$

The gradient w.r.t. $\boldsymbol{w}$ is

$$2\boldsymbol{S}\boldsymbol{w} - 2\lambda \boldsymbol{w}.$$

The optimal solution satisfies that the gradient equals 0, so

$$\boldsymbol{S}\boldsymbol{w} = \lambda \boldsymbol{w}$$

showing that it is the first eigenvector of the sample covariance $\boldsymbol{S}$. Furthermore, the projected variance for this optimal solution is

$$\boldsymbol{w}^\top \boldsymbol{S}\boldsymbol{w} = \boldsymbol{w}^\top \lambda \boldsymbol{w} = \lambda$$

showing that the projected variance is just the first eigenvalue of the sample covariance $\boldsymbol{S}$.

Consider $r > 1$. After finding the first principal component $\boldsymbol{u}_1$, we can recursively find the second principal component $\boldsymbol{w}$: find the direction that is orthogonal to the first principal component and maximizes the projected variance. Again, it can be shown that the second principal component is just the second eigenvector of the sample covariance, and the projected variance is just the second eigenvalue. Similarly for the other principal components.

In summary, PCA to dimension r is just finding the top r eigenvectors of the sample covariance, and projecting the original points to this subspace.

2.2 The Minimum Reconstruction Error View

Using any orthonormal basis $\boldsymbol{w}_1, \cdots, \boldsymbol{w}_d$, a point $\boldsymbol{x}_i$ can be written as

$$\boldsymbol{x}_i = \sum_{j=1}^d \alpha_{ij}\boldsymbol{w}_j = \sum_{j=1}^d (\boldsymbol{w}_j^\top \boldsymbol{x}_i)\boldsymbol{w}_j = \sum_{j=1}^d \boldsymbol{w}_j \boldsymbol{w}_j^\top \boldsymbol{x}_i$$

where

$$\alpha_{ij} = \boldsymbol{w}_j^\top \boldsymbol{x}_i.$$

Consider the r-term approximation to $\boldsymbol{x}_i$:

$$\widehat{\boldsymbol{x}}_i = \sum_{j=1}^{r} \alpha_{ij} \boldsymbol{w}_j.$$

We want the approximation error to be small for all training points:

$$\begin{aligned}
\frac{1}{n}\sum_{i=1}^{n} \|\widehat{\boldsymbol{x}}_i - \boldsymbol{x}_i\|^2 &= \frac{1}{n}\sum_{i=1}^{n} \| \sum_{j=r+1}^{d} \alpha_{ij} \boldsymbol{w}_j\|^2 \\
&= \frac{1}{n}\sum_{i=1}^{n} \sum_{j=r+1}^{d} \alpha_{ij}^2 \\
&= \frac{1}{n}\sum_{i=1}^{n} \sum_{j=r+1}^{d} \boldsymbol{w}_j^\top \boldsymbol{x}_i \boldsymbol{x}_i^\top \boldsymbol{w}_j \\
&= \sum_{j=r+1}^{d} \boldsymbol{w}_j^\top \boldsymbol{S} \boldsymbol{w}_j.
\end{aligned}$$

If $r = d - 1$, i.e., we need to remove a single dimension, it is easy to see that $\boldsymbol{w}_d$ should be the least eigenvector $\boldsymbol{u}_d$, because $\boldsymbol{u}_d^\top \boldsymbol{S} \boldsymbol{u}_d$ is the smallest among all unit vectors. Similarly, the other dimensions to remove are subsequently the eigenvectors corresponding to the least eigenvalues. In other words, we should use the top r eigenvectors for the r-term approximation.

2.3 Equivalence between the Two Views

The optimization objectives of the two views are equivalent, since the r-term approximation $\widehat{\boldsymbol{x}}_i$ is just the coordinates of the projection in the original coordinate system.

Consider a subspace spanned by r orthogonal basis vectors $\boldsymbol{w}_1, \cdots, \boldsymbol{w}_r$. The projection of $\boldsymbol{x}_i$ into this subspace is

$$\boldsymbol{p}_i = [\alpha_{i1}, \cdots, \alpha_{id}]^\top = [\boldsymbol{w}_1^\top \boldsymbol{x}_i, \cdots, \boldsymbol{w}_r^\top \boldsymbol{x}_i]^\top.$$

Its contribution to the projected covariance is $\boldsymbol{p}_i^\top \boldsymbol{p}_i$.

Geometrically, $\boldsymbol{p}_i^\top \boldsymbol{p}_i = \|\boldsymbol{p}_i\|^2$ is just the square of the length of the projection and equals $\|\widehat{\boldsymbol{x}}_i\|^2$. On the other hand, $\|\boldsymbol{x}_i - \widehat{\boldsymbol{x}}_i\|^2$ is the distance between $\boldsymbol{x}_i$ and its projection, which is its distance to the subspace spanned by $\boldsymbol{w}_1, \cdots, \boldsymbol{w}_r$. Clearly, $\|\widehat{\boldsymbol{x}}_i\|^2 + \|\boldsymbol{x}_i - \widehat{\boldsymbol{x}}_i\|^2 = \|\boldsymbol{x}_i\|^2$ (recall the Pythagorean Theorem!). This is a constant. Therefore, maximizing the projected variance $\|\boldsymbol{p}_i\|^2 = \|\widehat{\boldsymbol{x}}_i\|^2$ is equivalent to minimizing the reconstruction error $\|\boldsymbol{x}_i - \widehat{\boldsymbol{x}}_i\|^2$.

Why $\|\boldsymbol{p}_i\|^2 = \|\widehat{\boldsymbol{x}}_i\|^2$? This is because $\boldsymbol{p}_i$ is the coordinate vector in the basis $\boldsymbol{W} = [\boldsymbol{w}_1, \cdots, \boldsymbol{w}_r]$, and back to the original coordinate system of $\boldsymbol{x}_i$, the coordinate vector is just:

$$\boldsymbol{W}\boldsymbol{p}_i = \sum_{j=1}^{r} \alpha_{ij}\boldsymbol{w}_j = \widehat{\boldsymbol{x}}_i.$$

Therefore, their square lengths are the same:

$$\|\widehat{\boldsymbol{x}}_i\|^2 = \|\boldsymbol{W}\boldsymbol{p}_i\|^2 = (\boldsymbol{W}\boldsymbol{p}_i)^\top \boldsymbol{W}\boldsymbol{p}_i = \boldsymbol{p}_i^\top \boldsymbol{W}^\top \boldsymbol{W}\boldsymbol{p}_i = \boldsymbol{p}_i^\top \boldsymbol{p}_i = \|\boldsymbol{p}_i\|^2.$$

2.4 Solved by Singular Value Decomposition

PCA is just finding the top eigenvectors of the sample covariance $\boldsymbol{S}$. Note that

$$\boldsymbol{S} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{x}_i \boldsymbol{x}_i^\top = \frac{1}{n}\boldsymbol{X}^\top \boldsymbol{X}$$

where $\boldsymbol{X} \in \mathbb{R}^{n\times d}$ is the data matrix with $\boldsymbol{x}_i$ as the i-th row. By the connection between the eigendecomposition of $\boldsymbol{X}^\top \boldsymbol{X}$ and the SVD of $\boldsymbol{X}$, we know that PCA is just finding the top right singular vectors of the data matrix $\boldsymbol{X}$.