CSC 246 Machine Learning | Lecture 9

Principal Component Analysis

Lecturer: Jiaming Liang | *October 20, 2026*

In previous lectures, we studied *supervised learning*: given labeled examples $(\mathbf{x}_i, y_i)$, we learned a model for predicting y_i from $\mathbf{x}_i$. In this lecture, we introduce a basic task in *unsupervised learning*. We are given only the input vectors $\mathbf{x}_1, \ldots, \mathbf{x}_n$ and want to discover a useful low-dimensional representation of the data.

Our main method is *principal component analysis* (PCA). We will derive PCA from two viewpoints: maximizing variance and minimizing reconstruction error, and see how the singular value decomposition (SVD) computes its solution. We will then connect PCA to the kernel methods from Lecture 8.

Modern datasets often have many features. An image can have thousands of pixels, and a document can be represented by thousands of word frequencies. Nevertheless, some features may contain redundant information: for example, recording a person's height in both centimeters and inches does not provide two independent measurements.

Dimensionality reduction seeks a representation

$$\mathbf{x} \in \mathbb{R}^d \longmapsto \mathbf{z} \in \mathbb{R}^k, \qquad k < d.$$

Typical purposes include data visualization (especially $k = 2$ or 3), compression, noise reduction, and the construction of new features for subsequent learning tasks.

There are two distinct approaches. *Feature selection* keeps a subset of the original coordinates, such as (x_1, x_3). *Feature extraction* constructs new coordinates, such as

$$z_1 = \frac{x_1 + x_2}{\sqrt{2}}, \qquad z_2 = \frac{x_1 - x_2}{\sqrt{2}}.$$

PCA is a linear feature extraction method: each new coordinate is a linear combination of the original features.

1 Projections and the Geometry of PCA

Let $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^d$ be the data. Define their sample mean and centered vectors by

$$\overline{\mathbf{x}} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{x}_i, \qquad \widetilde{\mathbf{x}}_i = \mathbf{x}_i - \overline{\mathbf{x}}.$$

Thus $\sum_i \widetilde{\mathbf{x}}_i = 0$. PCA begins by *centering* the data, because the goal is to understand variation around the sample mean.

1.1 Projection onto a Line

Choose a unit vector $\mathbf{w} \in \mathbb{R}^d$, so that $\|\mathbf{w}\| = 1$. The scalar $z_i = \mathbf{w}^\top \widetilde{\mathbf{x}}_i$ is the coordinate of the centered data point along the direction $\mathbf{w}$. If we retain only this coordinate, our approximation to the original data point is

$$\widehat{\mathbf{x}}_i = \overline{\mathbf{x}} + z_i \mathbf{w} = \overline{\mathbf{x}} + \mathbf{w}\mathbf{w}^\top \widetilde{\mathbf{x}}_i.$$

Geometrically, $\mathbf{w}\mathbf{w}^\top \widetilde{\mathbf{x}}_i$ is the orthogonal projection onto the line spanned by $\mathbf{w}$. The difference $\mathbf{x}_i - \widehat{\mathbf{x}}_i$ is the reconstruction error.

The main question is: *which direction should we choose?* PCA selects a line that follows the dominant variation in the point cloud.

1.2 Projection onto a k-Dimensional Subspace

More generally, let

$$W = [\mathbf{w}_1, \ldots, \mathbf{w}_k] \in \mathbb{R}^{d \times k}, \qquad W^\top W = I_k.$$

The columns of W are orthonormal directions. We *encode* and *reconstruct* a data point by

$$\mathbf{z}_i = W^\top (\mathbf{x}_i - \overline{\mathbf{x}}) \in \mathbb{R}^k, \qquad \widehat{\mathbf{x}}_i = \overline{\mathbf{x}} + W \mathbf{z}_i.$$

Thus $WW^\top$ projects a centered vector onto the subspace spanned by $\mathbf{w}_1, \ldots, \mathbf{w}_k$. PCA finds an appropriate matrix W.

2 PCA as Maximum Variance

For centered data, define the *empirical covariance matrix*

$$\Sigma = \frac{1}{n} \sum_{i=1}^{n} \widetilde{\mathbf{x}}_i \widetilde{\mathbf{x}}_i^\top \in \mathbb{R}^{d \times d}.$$

The matrix Σ is symmetric and positive semidefinite, since

$$\mathbf{w}^\top \Sigma \mathbf{w} = \frac{1}{n} \sum_{i=1}^{n} (\mathbf{w}^\top \widetilde{\mathbf{x}}_i)^2 \geq 0$$

for all $\mathbf{w}$. We use the factor $1/n$ throughout; the common statistical convention $1/(n-1)$ gives the same principal directions.

2.1 The First Principal Component

Consider the projected coordinates $z_i = \mathbf{w}^\top \widetilde{\mathbf{x}}_i$. Their sample mean is zero, so their empirical variance is

$$\frac{1}{n}\sum_{i=1}^{n} z_i^2 = \frac{1}{n}\sum_{i=1}^{n} (\mathbf{w}^\top \widetilde{\mathbf{x}}_i)^2 = \mathbf{w}^\top \Sigma \mathbf{w}.$$

We want the direction that gives the greatest projected variance:

$$\max_{\mathbf{w}\in\mathbb{R}^d} \mathbf{w}^\top \Sigma \mathbf{w} \quad \text{subject to} \quad \|\mathbf{w}\| = 1.$$

To solve this optimization problem, consider the Lagrangian $\mathcal{L}(\mathbf{w}, \lambda) = \mathbf{w}^\top \Sigma \mathbf{w} - \lambda(\mathbf{w}^\top \mathbf{w} - 1)$. The first-order stationarity condition is

$$\nabla_{\mathbf{w}} \mathcal{L} = 2\Sigma\mathbf{w} - 2\lambda\mathbf{w} = 0, \qquad \text{or} \qquad \Sigma\mathbf{w} = \lambda\mathbf{w}.$$

Thus, at a solution, $\mathbf{w}$ must be an eigenvector of the covariance matrix.

Because Σ is symmetric, it has orthonormal eigenvectors $\mathbf{v}_1, \ldots, \mathbf{v}_d$ with corresponding eigenvalues

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d \geq 0, \qquad \Sigma\mathbf{v}_j = \lambda_j \mathbf{v}_j.$$

Any unit vector can be written as $\mathbf{w} = \sum_{j=1}^{d} a_j \mathbf{v}_j$ with $\sum_{j=1}^{d} a_j^2 = 1$. Hence

$$\mathbf{w}^\top \Sigma \mathbf{w} = \sum_{j=1}^{d} \lambda_j a_j^2 \leq \lambda_1 \sum_{j=1}^{d} a_j^2 = \lambda_1.$$

Equality is achieved by $\mathbf{w} = \mathbf{v}_1$.

Theorem 1. *A unit eigenvector of Σ corresponding to its largest eigenvalue λ_1 maximizes the variance of the projected data. The maximum variance is λ_1.*

2.2 The First k Principal Components

To retain more than one direction, maximize the total projected variance over all matrices with orthonormal columns:

$$\max_{W\in\mathbb{R}^{d\times k}} \operatorname{tr}(W^\top \Sigma W) \qquad \text{subject to} \qquad W^\top W = I_k.$$

Here $\operatorname{tr}(A)$ denotes the sum of diagonal entries of A. Because $\operatorname{tr}(W^\top \Sigma W) = \sum_{j=1}^{k} \mathbf{w}_j^\top \Sigma \mathbf{w}_j$, this objective sums the variances in the k selected directions.

The optimal choice is

$$W_* = [\mathbf{v}_1, \ldots, \mathbf{v}_k], \qquad \max_{W^\top W = I_k} \operatorname{tr}(W^\top \Sigma W) = \sum_{j=1}^{k} \lambda_j.$$

In words, PCA keeps the eigenvectors with the k largest eigenvalues. An important consequence is that the coordinates of PCA are uncorrelated:

$$\frac{1}{n}\sum_{i=1}^{n}\mathbf{z}_i\mathbf{z}_i^\top = W_*^\top \Sigma W_* = \operatorname{diag}(\lambda_1,\ldots,\lambda_k).$$

3 PCA as Minimum Reconstruction Error

Instead of maximizing variance, suppose we want the reconstructed points $\widehat{\mathbf{x}}_i$ to be as close as possible to the original observations. Consider

$$\min_{W^\top W = I_k} \frac{1}{n}\sum_{i=1}^{n}\|\mathbf{x}_i - \widehat{\mathbf{x}}_i\|^2 = \min_{W^\top W = I_k} \frac{1}{n}\sum_{i=1}^{n}\left\|(I - WW^\top)\widetilde{\mathbf{x}}_i\right\|^2.$$

Since $W^\top W = I_k$, $WW^\top$ is an orthogonal projection. The projected part and the residual are perpendicular. By the Pythagorean theorem,

$$\|\widetilde{\mathbf{x}}_i\|^2 = \|WW^\top\widetilde{\mathbf{x}}_i\|^2 + \|(I - WW^\top)\widetilde{\mathbf{x}}_i\|^2 = \|W^\top\widetilde{\mathbf{x}}_i\|^2 + \|(I - WW^\top)\widetilde{\mathbf{x}}_i\|^2.$$

Averaging over the data points gives

$$\underbrace{\frac{1}{n}\sum_{i=1}^{n}\|(I - WW^\top)\widetilde{\mathbf{x}}_i\|^2}_{\text{reconstruction error}} = \underbrace{\operatorname{tr}(\Sigma)}_{\text{total variance}} - \underbrace{\operatorname{tr}(W^\top\Sigma W)}_{\text{retained variance}}.$$

The total variance is fixed, so minimizing reconstruction error is exactly equivalent to maximizing retained variance. Consequently,

$$\min_{W^\top W = I_k} \frac{1}{n}\sum_{i=1}^{n}\|\mathbf{x}_i - \widehat{\mathbf{x}}_i\|^2 = \sum_{j=k+1}^{d}\lambda_j.$$

Thus, PCA keeps the directions with the greatest variation and discards the ones that contribute the least to the squared reconstruction error.

4 Computing PCA with the Singular Value Decomposition

Computing the eigenvectors of Σ is one way to find PCA, but it is often more convenient to work directly with the data matrix. Let $X \in \mathbb{R}^{n\times d}$ have the centered observations as its rows:

$$X = \begin{pmatrix}\widetilde{\mathbf{x}}_1^\top \\ \vdots \\ \widetilde{\mathbf{x}}_n^\top\end{pmatrix}, \qquad \Sigma = \frac{1}{n}X^\top X.$$

The SVD of X is

$$X = USV^\top,$$

where U and V have orthonormal columns and S is diagonal with nonnegative singular values $s_1 \geq s_2 \geq \cdots \geq s_r > 0$; $r = \operatorname{rank}(X)$. (Here we use the compact SVD.)

Substituting the SVD into $X^\top X$ gives

$$\Sigma = \frac{1}{n} VSU^\top USV^\top = V \operatorname{diag}\left(\frac{s_1^2}{n}, \ldots, \frac{s_r^2}{n}\right) V^\top.$$

Therefore, the right singular vectors of the centered data matrix are the principal directions, and $\lambda_j = \frac{s_j^2}{n}$. If $k \leq r$, the matrix of PCA scores and the reconstructed centered data are

$$Z = XV_k = U_k S_k \in \mathbb{R}^{n \times k}, \qquad \widehat{X} = ZV_k^\top = U_k S_k V_k^\top.$$

Thus, $\widehat{X}$ is a rank-k approximation to the data matrix. The SVD provides a computationally useful way to obtain PCA without explicitly constructing the $d \times d$ covariance matrix. In fact, the truncated SVD also provides the best rank-k approximation to X in Frobenius norm (a result we state without proof).

4.1 Explained Variance and Choosing k

The fraction of the total variance captured by the first k principal components is

$$\text{ExplainedVariance}(k) = \frac{\lambda_1 + \cdots + \lambda_k}{\lambda_1 + \cdots + \lambda_d}.$$

Here we assume the total variance is positive. One common approach is to choose the smallest k for which this fraction exceeds a target, such as 90% or 95%. A plot of the eigenvalues, or of cumulative explained variance, can also help identify an appropriate dimension.

For example, suppose the eigenvalues are $6, 3, 1$. Keeping one component retains 60% of the variance; keeping two retains 90%; retaining all three gives exact reconstruction of the centered data.

4.2 A Practical PCA Procedure

1. Compute the mean $\overline{\mathbf{x}}$ of the training data, and center the data matrix.
2. Compute the top k right singular vectors of the centered matrix (or the top eigenvectors of its covariance matrix).
3. Transform a new observation using $\mathbf{z} = V_k^\top(\mathbf{x} - \overline{\mathbf{x}})$.
4. If reconstruction is desired, compute $\widehat{\mathbf{x}} = \overline{\mathbf{x}} + V_k \mathbf{z}$.

Several details matter in applications. First, PCA is sensitive to the units of different features. If one feature is measured on a much larger scale, it may dominate the variance; standardizing the features can be useful, depending on the problem. Second, if PCA is used before training a predictive model, the centering, standardization, and PCA directions must be learned from the training data only, not from the test set. Finally, retaining high variance does not guarantee retaining information useful for predicting labels.

For image data, PCA can be used to represent each image with a relatively small number of components. Increasing k typically makes reconstructed images more accurate, at the cost of storing more coordinates.

5 Kernel PCA: Nonlinear Dimensionality Reduction

Ordinary PCA finds linear directions in the original input space. However, if the data have nonlinear structure, a linear representation may not capture the important patterns.

The idea of kernel PCA is to first map the data into a feature space through a nonlinear feature map $\psi : \mathcal{X} \to \mathcal{F}$, and then apply ordinary PCA to the feature vectors $\psi(\mathbf{x}_1), \ldots, \psi(\mathbf{x}_n)$. The principal components are linear in the feature space, but they may correspond to nonlinear functions of the original input.

As in kernel SVM, we do not need to construct the feature vectors explicitly. Instead, we use a kernel function

$$K(\mathbf{x}, \mathbf{x}') = \langle \psi(\mathbf{x}), \psi(\mathbf{x}') \rangle_{\mathcal{F}}$$

to compute inner products in the feature space. This allows us to perform PCA using kernel evaluations, after centering the feature vectors, without explicitly computing $\psi(\mathbf{x})$.

For example, the polynomial kernel $K(\mathbf{x}, \mathbf{x}') = (1 + \langle \mathbf{x}, \mathbf{x}' \rangle)^2$ implicitly introduces quadratic features. Kernel PCA can therefore capture nonlinear patterns that ordinary PCA in the original input space may miss.

In summary, kernel PCA is ordinary PCA applied in a feature space, implemented using the kernel trick.