CS 412 — Introduction to Machine Learning (UIC) April 01, 2025

Lecture 19

Instructor: Aadirupa Saha *Scribe(s): Jiachen Tao/ Aksun Agnihotri*

[This draft is not fully proofread. Please email any typos/errors to the instructor or directly edit the latex file.]

Overview

In the last lecture, we covered the following main topics:

1. Boosting
2. Ada Boost
3. Mistake Bounds

This lecture focuses on:

1. Linear Algebra Preliminaries
2. Orthonormal Basis
3. Principal Component Analysis (PCA)

1 Linear Algebra Preliminaries (Required)

1.1 Definition: Vector Space

Vector Space: A vector space consists of:

- a set $\mathcal{V}$
- a scalar field $\mathbb{Q}$ (usually $\mathbb{R}$ or $\mathbb{C}$)
- and two operations: vector addition $+$ and scalar multiplication $\cdot$

These must satisfy the following properties:

1. For any pair of elements $x, y \in \mathcal{V}$, the sum $x + y \in \mathcal{V}$ (closure under addition).

2. For any $x \in \mathcal{V}$ and scalar $\alpha \in \mathbb{Q}$, we have $\alpha \cdot x \in \mathcal{V}$ (closure under scalar multiplication).

3. There exists a zero vector $0 \in \mathcal{V}$ such that $x + 0 = x$ for any $x \in \mathcal{V}$.

4. For every $x \in \mathcal{V}$, there exists an additive inverse $-x \in \mathcal{V}$ such that $x + (-x) = 0$.

5. The addition operation $+$ is:

- **Commutative:** $x + y = y + x$
- **Associative:** $(x + y) + z = x + (y + z)$

6. Scalar multiplication is associative: for any scalars $\alpha, \beta \in \mathbb{Q}$ and $x \in \mathcal{V}$,

$$\alpha(\beta \cdot x) = (\alpha\beta) \cdot x$$

7. Scalar and vector sums are distributive:

- $(\alpha + \beta) \cdot x = \alpha \cdot x + \beta \cdot x$
- $\alpha \cdot (x + y) = \alpha \cdot x + \alpha \cdot y$

Example: Vector Spaces

- $\mathbb{R}^n$: The set of all n-dimensional real-valued vectors. Closed under addition and scalar multiplication.
- $\mathbb{R}^{m \times n}$: The set of all $m \times n$ real matrices. Matrix addition and scalar multiplication satisfy all vector space axioms.
- P_n: The set of all polynomials of degree at most n. Closed under polynomial addition and scalar multiplication.
- $\mathcal{F} = \{f : \mathbb{R} \to \mathbb{R}\}$: The set of all real-valued functions defined on $\mathbb{R}$. Addition and scalar multiplication of functions preserve closure.

1.2 Definition: Subspace of a Vector Space

A **subspace** of a vector space $\mathcal{V}$ is any subset $\mathcal{W} \subseteq \mathcal{V}$ that is itself a vector space under the same operations as $\mathcal{V}$.

Examples of Vector Spaces

- $\mathbb{R}^n$: Set of all n-dimensional real vectors
- $\mathbb{R}^{m \times n}$: Set of all real $m \times n$ matrices
- P_n: Set of all polynomials of degree at most n
- $\mathcal{F}$: Set of all real (or complex) valued functions, i.e., $\{f : \mathbb{R} \to \mathbb{R}\}$

Example: Subspaces

- The set of all vectors in $\mathbb{R}^3$ of the form $(x, y, 0)$ is a subspace of $\mathbb{R}^3$. It is closed under addition and scalar multiplication.
- The set of all 3×3 symmetric matrices forms a subspace of $\mathbb{R}^{3\times3}$.
- The set $W = \{x \in \mathbb{R}^3 : x_1 + x_2 + x_3 = 0\}$ is a subspace of $\mathbb{R}^3$. It contains the zero vector and is closed under linear combinations.

1.3 Definition: Linear Independence

A set of m vectors $v_1, v_2, \ldots, v_m$ is said to be **linearly dependent** if there exist scalars $\alpha_1, \alpha_2, \ldots, \alpha_m$, not all zero, such that:

$$\sum_{i=1}^{m} \alpha_i v_i = 0$$

Otherwise, the set is called **linearly independent**.

Example: Columns of the Identity Matrix

The columns of the $n \times n$ identity matrix are linearly independent:

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

These columns can be written as:

$$e_1 = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad e_2 = \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix}, \quad \ldots \quad e_n = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}$$

These are called the standard unit vectors in $\mathbb{R}^n$, and they are orthogonal and linearly independent.

To test any set of vectors $v_1, \ldots, v_n$ for linear independence:

- Form the matrix $A = [v_1 \, v_2 \, \ldots \, v_n]$
- Solve the homogeneous system $Ac = 0$
- If the only solution is $c = 0$, then the set is linearly independent
- If $n > m$ (more vectors than dimensions), the set must be linearly dependent

Theorem: A set of n vectors in $\mathbb{R}^m$ must be linearly dependent if $n > m$.

1.4 Definition: Span of a Set of Vectors

Let $v_1, v_2, \ldots, v_m$ be a set of vectors in a vector space $\mathcal{V}$. Then the **span** of this set is defined as the set of all possible linear combinations:

$$\text{Span}(v_1, \ldots, v_m) = \left\{ y \in \mathcal{V} \;\middle|\; y = \sum_{i=1}^{m} \alpha_i v_i \text{ for some } \alpha_i \in \mathbb{Q} \right\}$$

Let $v_1, v_2, \ldots, v_m \in \mathcal{V}$. Then:

$$\text{Span}(v_1, \ldots, v_m) = \left\{ y \;\middle|\; y = \sum_{i=1}^{m} \alpha_i v_i, \; \alpha_i \in \mathbb{Q} \right\}$$

This set consists of all linear combinations of $v_1, \ldots, v_m$.

Example: Span

Let $v_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ and $v_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ in $\mathbb{R}^2$. Then:

$$\text{Span}(v_1, v_2) = \{\alpha_1 v_1 + \alpha_2 v_2 \mid \alpha_1, \alpha_2 \in \mathbb{R}\}$$

$$= \left\{ \begin{pmatrix} \alpha_1 \\ \alpha_2 \end{pmatrix} \middle| \alpha_1, \alpha_2 \in \mathbb{R} \right\} = \mathbb{R}^2$$

This means the vectors v_1 and v_2 span the entire $\mathbb{R}^2$ space.

Alternate example: Let $v_1 = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$. Then:

$$\text{Span}(v_1) = \left\{ \alpha \begin{pmatrix} 1 \\ 2 \end{pmatrix} \middle| \alpha \in \mathbb{R} \right\}$$

This is a line through the origin in the direction of v_1, a 1D subspace of $\mathbb{R}^2$.

1.5 Definition: Basis of a Vector Space

A **basis** of a vector space $\mathcal{V}$ is a set of linearly independent vectors $a_1, \ldots, a_n \in \mathcal{V}$ such that:

$$\mathcal{V} = \text{Span}(a_1, \ldots, a_n)$$

Examples:

1. The standard basis for $\mathbb{R}^n$ is $\{e_1, \ldots, e_n\}$, where e_i is the unit vector with a 1 in the i^{th} coordinate:

$$e_i = (0, \ldots, 1, \ldots, 0)^T, \quad i \in [n]$$

2. A basis for $\mathbb{R}^{3\times 2}$ consists of 6 matrices:

$$\begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}$$

3. A basis of $\mathbb{R}^3$ is:

$$\left\{ \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \right\}$$

Note: Basis and Coordinates

If x is a vector and $\mathcal{S} \subseteq \mathcal{V}$ is a subspace of dimension n, then for any basis $b_1, \ldots, b_n \in \mathcal{S}$, if:

$$\langle x, b_i \rangle > 0, \quad \forall i \in [n],$$

then x is aligned (non-orthogonal) to all vectors in $\mathcal{S}$, and the representation is unique.

1.6 Definition: Orthonormal Basis

A basis $\{v_1, \ldots, v_n\}$ of a vector space $\mathcal{V}$ is called an **orthonormal basis** if:

- $\langle v_i, v_j \rangle = 0 \quad \forall i \neq j$ (vectors are orthogonal)
- $\langle v_i, v_i \rangle = 1 \quad \forall i$ (unit norm)

Example: Two orthonormal bases for $\mathbb{R}^3$ could be:

$$\left\{ \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \right\} \quad \text{and} \quad \left\{ \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \\ 0 \end{pmatrix}, \begin{pmatrix} \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \right\}$$

1.7 Algorithm: Gram–Schmidt Orthonormalization

A method to convert a set of linearly independent vectors $a_1, \ldots, a_m \in \mathbb{R}^n$ into an orthonormal basis.

Input:

Linearly independent vectors $a_1, \ldots, a_m \in \mathbb{R}^n$

Initialize:

$$v_1 = \frac{a_1}{\|a_1\|_2}$$

For $i = 2$ **to** m**:**

$$v_i' = a_i - \sum_{j=1}^{i-1} \langle a_i, v_j \rangle v_j \quad \Rightarrow \quad v_i = \frac{v_i'}{\|v_i'\|_2}$$

Output:

Orthonormal vectors $v_1, \ldots, v_m$

Exercise:

1. Can you show that $v_1, \ldots, v_m$ are orthogonal to each other?
2. Do $v_1, \ldots, v_m$ form an orthonormal basis for the span of $\{a_1, \ldots, a_m\}$?

Theorem 1.1: Summary of Key Vector Space Properties

A vector space $\mathcal{V}$ over a field $\mathbb{Q}$ must satisfy the following:

1. Closure under vector addition and scalar multiplication
2. Existence of zero vector and additive inverses
3. Associativity and commutativity of addition
4. Distributive properties of scalar multiplication over vectors and scalars
5. Associativity of scalar multiplication

In addition:

- A basis spans $\mathcal{V}$ and is linearly independent.
- An orthonormal basis satisfies:

$$\langle v_i, v_j \rangle = 0 \text{ for } i \neq j, \quad \text{and} \quad \langle v_i, v_i \rangle = 1$$

- Gram–Schmidt converts any linearly independent set into an orthonormal basis for its span.

Exercise 1.1: Conceptual Check

1. Prove that a set of vectors that is both spanning and linearly independent forms a basis.
2. Give an example of a set of vectors in $\mathbb{R}^3$ that is linearly dependent but spans a 2D subspace.
3. Use Gram–Schmidt to convert the set $\left\{ \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix} \right\}$ into an orthonormal basis.
4. Explain why orthonormal bases simplify projection and coordinate computations.

2 Principal Component Analysis (PCA)

2.1 Motivation and Problem Setup

Principal Component Analysis (PCA) is a technique for **dimensionality reduction** of data instances. It is an **unsupervised learning algorithm**.

Consider a given dataset:

$$\mathcal{D} = \{x_1, x_2, \ldots, x_n\} \quad \text{where} \quad x_i \in \mathbb{R}^D$$

Here:

- $\mathcal{D}$ is the dataset
- D is the dimensionality of the data
- D is typically **very large**

Representing or transmitting such high-dimensional data:

- Requires significant memory, time, and bandwidth
- Is computationally expensive

Therefore, we need a technique to **reduce the dimensionality** D, ideally:

- Retaining the most important features of the data
- Reducing redundancy and noise in the representation

2.2 Applications of PCA

Principal Component Analysis (PCA) is widely used in unsupervised learning and data preprocessing. Its applications include:

- **Dimensionality Reduction:** Reducing the number of features while preserving most of the data variance. Often used as a preprocessing step before supervised learning models.
- **Noise Reduction:** PCA eliminates components that capture very low variance — often attributable to noise — thereby improving signal quality.
- **Data Visualization:** High-dimensional datasets (e.g., $D > 100$) can be projected into 2D or 3D for visualization using the top principal components.
- **Image Compression:** In computer vision, PCA is used to compress images by storing only the most significant basis vectors and their projections.
- **Feature Decorrelation:** PCA produces orthogonal (uncorrelated) components, which can improve learning performance in models sensitive to correlated features.
- **Genomics and Signal Processing:** PCA is used to analyze expression patterns in gene data and in filtering signals for noise separation.

These practical applications motivate the need for a mathematically principled way to project data into lower dimensions with minimal information loss.

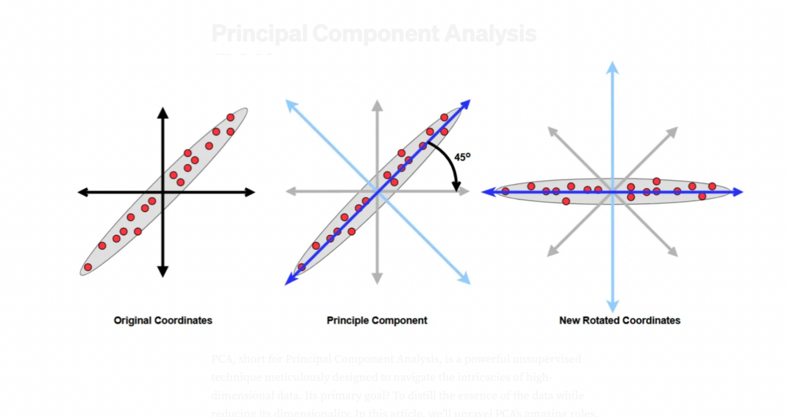

Figure 1: PCA projects high-dimensional data (e.g., 2D) onto a lower-dimensional subspace (e.g., 1D) by finding the direction of maximum variance.

2.3 Step-by-Step Derivation of PCA

Let us assume that $\mathcal{D} = \{x_1, \ldots, x_n\} \subset \mathbb{R}^D$, and that we have an orthonormal basis $B = \{u_1, \ldots, u_D\}$ such that:

$$u_i \in \mathbb{R}^D, \quad \|u_i\| = 1, \quad \langle u_i, u_j \rangle = 0 \text{ for } i \neq j$$

Since B is a complete basis for $\mathbb{R}^D$, any datapoint $x_n \in \mathbb{R}^D$ can be expressed as:

$$x_n = \sum_{i=1}^{D} \alpha_{ni} u_i, \quad \text{where } \alpha_{ni} = \langle x_n, u_i \rangle$$

Each coefficient α_{ni} is the projection of x_n along the basis direction u_i. This uses D numbers to represent each point.

Theorem 2.1: PCA Representation via Orthonormal Basis

Any datapoint $x_n \in \mathbb{R}^D$ can be exactly represented using an orthonormal basis $\{u_1, \ldots, u_D\}$ as:

$$x_n = \sum_{i=1}^{D} \alpha_{ni} u_i \quad \text{where } \alpha_{ni} = \langle x_n, u_i \rangle$$

To reduce dimensionality, PCA approximates x_n using only the top $M < D$ components:

$$x_n \approx \sum_{j=1}^{M} \beta_{nj} u_j$$

This provides a compact, noise-reduced representation in an M-dimensional subspace.

Exercise 2.1: Understanding PCA Projections

1. Given an orthonormal basis $\{u_1, u_2, u_3\}$, compute the projection coefficients $\alpha_{ni} = \langle x_n, u_i \rangle$ for a given point x_n.
2. If $x_n \in \mathbb{R}^5$ is projected using only the first 2 basis vectors, how many components are ignored? What does this imply geometrically?
3. Why is the orthonormality of the basis crucial in PCA? What would happen if the basis vectors were not orthogonal?
4. Can PCA increase accuracy in a supervised learning task? Why or why not?

2.4 M-Component PCA

Our goal is to approximate every $x_n \in \mathbb{R}^D$ using a **representation involving only a subset** $M < D$ of the basis vectors, i.e., a projection of x_n onto a lower-dimensional subspace.
Let us assume that each datapoint x_n can be approximated using only the first M directions:

$$x_n \approx \sum_{j=1}^{M} \beta_{nj} u_j + \sum_{j=M+1}^{D} c_{nj} u_j$$

Where: - β_{nj}: coefficients capturing meaningful projection (learned) - c_{nj}: treated as residuals (ignored in low-dimensional representation)

Goal: Find the $\{u_j\}_{j=1}^{M}$ and $\{\beta_{nj}\}_{j=1}^{M}$ for all $n \in [N]$ such that each x_n is well-approximated.

2.5 Minimum-Error Formulation of PCA

Let us analyze the average ℓ_2-approximation error defined as:

$$\mathcal{J} = \frac{1}{N} \sum_{n=1}^{N} \|x_n - \tilde{x}_n\|^2$$

Our goal is to minimize $\mathcal{J}$ over:

$$\{z_{ni}\}, \quad \{b_j\}_{j=1}^{D}, \quad \{u_j\}_{j=1}^{D}, \quad i \in [M],\ n \in [N]$$

—

Step 1: Minimizing over z_{ni}

We take the derivative of $\mathcal{J}$ with respect to z_{ni} and set it to zero:

$$\nabla_{z_{ni}} \mathcal{J} = 0 \quad \forall\, i, n$$

$$\Rightarrow \frac{1}{N} \sum_{n=1}^{N} (x_n - \tilde{x}_n)^T \frac{d\tilde{x}_n}{dz_{ni}} = 0$$

Note that:

$$x_n = \sum_{i=1}^{D} \alpha_{ni} u_i, \quad \tilde{x}_n = \sum_{i=1}^{M} z_{ni} u_i + \sum_{j=M+1}^{D} b_j u_j$$

So the gradient condition becomes:

$$\left\langle \sum_{i=1}^{D} \alpha_{ni} u_i - \left(\sum_{i=1}^{M} z_{ni} u_i + \sum_{j=M+1}^{D} b_j u_j \right), u_i \right\rangle = 0 \quad \text{(for all } i, n)$$

Using orthonormality of u_i, we get:

$$\alpha_{ni} - z_{ni} = 0 \quad \Rightarrow \quad z_{ni} = \alpha_{ni} = \langle x_n, u_i \rangle$$

—

Step 2: Minimizing over b_j

We similarly take the derivative of $\mathcal{J}$ with respect to b_j, and get:

$$\nabla_{b_j}\mathcal{J} = 0 \quad \Rightarrow \quad b_j = \bar{x}^T u_j$$

Where $\bar{x}$ is the mean of the dataset:

$$\bar{x} = \frac{1}{N}\sum_{n=1}^{N} x_n$$

Substitute Back to Get Residual Error

Substituting values $z_{ni} = \alpha_{ni}$, $b_j = \bar{x}^T u_j$, we have:

$$x_n - \tilde{x}_n = \sum_{j=M+1}^{D} \langle x_n - \bar{x}, u_j \rangle u_j$$

Hence the reconstruction error becomes:

$$\mathcal{J} = \frac{1}{N}\sum_{n=1}^{N} \|x_n - \tilde{x}_n\|^2 = \frac{1}{N}\sum_{n=1}^{N}(x_n - \bar{x})^T(x_n - \bar{x})$$
$$= \frac{1}{N}\sum_{n=1}^{N}\sum_{j=M+1}^{D} \left((x_n - \bar{x})^T u_j\right)^2$$
$$= \sum_{j=M+1}^{D} u_j^T S u_j \quad \text{where} \quad S = \frac{1}{N}\sum_{n=1}^{N}(x_n - \bar{x})(x_n - \bar{x})^T$$

Here, S is the average data covariance matrix.

Conclusion

We have shown that the reconstruction error for approximating x_n using only the top M directions of an orthonormal basis is given by:

$$\mathcal{J} = \sum_{j=M+1}^{D} \left(\frac{1}{N}\sum_{n=1}^{N} \left(u_j^T(x_n - \bar{x})\right)^2 \right)$$

Using the fact that:

$$\frac{1}{N}\sum_{n=1}^{N} \left(u_j^T(x_n - \bar{x})\right)^2 = u_j^T S u_j \quad \text{where} \quad S = \frac{1}{N}\sum_{n=1}^{N}(x_n - \bar{x})(x_n - \bar{x})^T$$

we finally arrive at:

$$\boxed{\mathcal{J} = \sum_{j=M+1}^{D} u_j^T S u_j}$$

where S is the empirical data covariance matrix.

Theorem 2.2: Minimum-Error PCA Objective

Let S be the data covariance matrix:

$$S = \frac{1}{N}\sum_{n=1}^{N}(x_n - \bar{x})(x_n - \bar{x})^T$$

Then the average reconstruction error from projecting each x_n onto the top M components is:

$$\mathcal{J} = \sum_{j=M+1}^{D} u_j^T S u_j$$

To minimize this error, PCA chooses $\{u_1, \ldots, u_M\}$ as the top M eigenvectors of S, corresponding to the largest eigenvalues.

References:

1. Linear Algebra Notes – Chapter 2: Vector Spaces [Link]
2. Gilbert Strang, *Introduction to Linear Algebra*, 5th Edition – Vector Space Examples and Basis
3. Sebastian Raschka, Lecture Notes on Regularization and PCA [Link]
4. Wikipedia: Principal Component Analysis [Link]
5. CMU Deep Learning Reading on PCA [Link]
6. YouTube: PCA Intuition and Step-by-Step Derivation [Link]
7. Gram-Schmidt Orthonormalization [Link]
8. Setosa.io: Interactive Visual Explanation of PCA [Link]