Principal Component Analysis and Dimensionality Reduction: Mathematical Foundations and Algorithmic Implementation

Course Notes

July 7, 2025

Contents

1 Introduction to Dimensionality Reduction

The fundamental challenge in modern data analysis lies in handling high-dimensional datasets where the number of features often exceeds the number of observations. This phenomenon, commonly referred to as the *curse of dimensionality*, necessitates sophisticated mathematical techniques for extracting meaningful patterns while preserving essential information.

Principal Component Analysis (PCA) emerges as one of the most powerful and widely-used techniques for dimensionality reduction, providing both theoretical elegance and practical utility. At its core, PCA seeks to identify a lower-dimensional subspace that captures the maximum variance in the original data, thereby enabling efficient representation and analysis.

1.1 Motivation and Problem Statement

Consider a dataset consisting of n observations in d-dimensional space, represented as column vectors $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ where each $\mathbf{x}_i \in \mathbb{R}^d$. The central objective of dimensionality reduction is to find a mapping:

$$\phi : \mathbb{R}^d \to \mathbb{R}^k \tag{1}$$

where $k \ll d$, such that the essential structure of the data is preserved in the lower-dimensional representation.

Key Challenges:

- **Information Preservation**: Maintaining the most important characteristics of the original data
- **Computational Efficiency**: Ensuring tractable algorithms for large-scale applications
- **Interpretability**: Providing meaningful insights into the underlying data structure

2 Mathematical Foundation: Regularized Linear Regression

2.1 Vector Calculus and Matrix Derivatives

Before deriving the PCA algorithm, we must establish the mathematical framework for computing derivatives with respect to vectors and matrices. This foundation is essential for understanding the optimization procedures underlying dimensionality reduction techniques.

Definition 2.1 (Gradient with Respect to a Vector)**.** *Given a scalar function* $f : \mathbb{R}^n \to \mathbb{R}$*, the gradient with respect to vector* $\mathbf{x} = [x_1, x_2, \ldots, x_n]^T$ *is defined as:*

$$\nabla_{\mathbf{x}} f(\mathbf{x}) = \left[\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \ldots, \frac{\partial f}{\partial x_n}\right]^T \tag{2}$$

2.1.1 Fundamental Derivative Rules

The following lemmas establish the basic rules for matrix calculus that are essential for PCA derivations:

Lemma 2.2 (Linear Function Derivative)**.** *For a linear function* $f(\mathbf{x}) = \mathbf{a}^T\mathbf{x}$ *where* $\mathbf{a}$ *is a constant vector:*

$$\nabla_{\mathbf{x}}(\mathbf{a}^T\mathbf{x}) = \mathbf{a} \tag{3}$$

Lemma 2.3 (Quadratic Form Derivative)**.** *For a quadratic function* $f(\mathbf{x}) = \mathbf{x}^T\mathbf{A}\mathbf{x}$ *where* $\mathbf{A}$ *is a symmetric matrix:*

$$\nabla_{\mathbf{x}}(\mathbf{x}^T\mathbf{A}\mathbf{x}) = 2\mathbf{A}\mathbf{x} \tag{4}$$

Proof of Lemma 2.3. Consider the quadratic form $f(\mathbf{x}) = \mathbf{x}^T\mathbf{A}\mathbf{x} = \sum_{i=1}^n \sum_{j=1}^n A_{ij}x_ix_j$.

Taking the partial derivative with respect to x_k:

$$\begin{aligned}
\frac{\partial f}{\partial x_k} &= \frac{\partial}{\partial x_k}\left(\sum_{i=1}^n \sum_{j=1}^n A_{ij}x_ix_j\right) && (5)\\
&= \sum_{j=1}^n A_{kj}x_j + \sum_{i=1}^n A_{ik}x_i && (6)\\
&= 2\sum_{j=1}^n A_{kj}x_j \quad \text{(since } \mathbf{A} \text{ is symmetric)} && (7)\\
&= 2(\mathbf{A}\mathbf{x})_k && (8)
\end{aligned}$$

Therefore, $\nabla_{\mathbf{x}} f(\mathbf{x}) = 2\mathbf{A}\mathbf{x}$. □

2.2 Regularized Least Squares Problem

Consider the regularized least squares problem, which forms the basis for many dimensionality reduction techniques:

$$\text{minimize}_{\mathbf{w}} \quad \|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2 + \lambda\|\mathbf{w}\|_2^2 \tag{9}$$

where:

- $\mathbf{y} \in \mathbb{R}^n$ is the target vector
- $\mathbf{X} \in \mathbb{R}^{n\times d}$ is the design matrix
- $\mathbf{w} \in \mathbb{R}^d$ is the parameter vector
- $\lambda > 0$ is the regularization parameter

2.2.1 Analytical Solution Derivation

The objective function can be written as:

$$J(\mathbf{w}) = (\mathbf{y} - \mathbf{X}\mathbf{w})^T(\mathbf{y} - \mathbf{X}\mathbf{w}) + \lambda\mathbf{w}^T\mathbf{w} \tag{10}$$

Expanding the first term:

$$(\mathbf{y} - \mathbf{X}\mathbf{w})^T(\mathbf{y} - \mathbf{X}\mathbf{w}) = \mathbf{y}^T\mathbf{y} - 2\mathbf{y}^T\mathbf{X}\mathbf{w} + \mathbf{w}^T\mathbf{X}^T\mathbf{X}\mathbf{w} \tag{11}$$

Therefore, the complete objective function becomes:

$$J(\mathbf{w}) = \mathbf{y}^T\mathbf{y} - 2\mathbf{y}^T\mathbf{X}\mathbf{w} + \mathbf{w}^T\mathbf{X}^T\mathbf{X}\mathbf{w} + \lambda\mathbf{w}^T\mathbf{w} \tag{12}$$

Taking the gradient with respect to $\mathbf{w}$:

$$\begin{aligned}
\nabla_{\mathbf{w}} J(\mathbf{w}) &= \nabla_{\mathbf{w}}(\mathbf{y}^T\mathbf{y}) - 2\nabla_{\mathbf{w}}(\mathbf{y}^T\mathbf{X}\mathbf{w}) + \nabla_{\mathbf{w}}(\mathbf{w}^T\mathbf{X}^T\mathbf{X}\mathbf{w}) + \lambda\nabla_{\mathbf{w}}(\mathbf{w}^T\mathbf{w}) && (13)\\
&= 0 - 2\mathbf{X}^T\mathbf{y} + 2\mathbf{X}^T\mathbf{X}\mathbf{w} + 2\lambda\mathbf{w} && (14)\\
&= -2\mathbf{X}^T\mathbf{y} + 2(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})\mathbf{w} && (15)
\end{aligned}$$

Setting the gradient to zero for the optimal solution:

$$2(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})\mathbf{w}^* = 2\mathbf{X}^T\mathbf{y} \tag{16}$$

Theorem 2.4 (Regularized Least Squares Solution)**.** *The optimal solution to the regularized least squares problem is given by:*

$$\boxed{\mathbf{w}^* = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y}} \tag{17}$$

provided that $\lambda > 0$, ensuring the invertibility of the matrix $(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})$.

Remark 2.5. *The regularization term $\lambda\mathbf{I}$ ensures that the solution exists and is unique even when $\mathbf{X}^T\mathbf{X}$ is singular, which commonly occurs in high-dimensional settings where $d > n$.*

3 Principal Component Analysis: Theory and Derivation

3.1 Problem Formulation and Assumptions

Principal Component Analysis addresses the fundamental question: How can we find a lower-dimensional representation of high-dimensional data that preserves the maximum amount of variance?

Definition 3.1 (PCA Problem Statement)**.** *Given a dataset* $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n] \in \mathbb{R}^{d\times n}$ *where each column represents an observation, find an orthogonal transformation* $\mathbf{W} \in \mathbb{R}^{d\times k}$ *(where* $k < d$*) such that the projected data* $\mathbf{Y} = \mathbf{W}^T\mathbf{X}$ *maximizes the variance in the lower-dimensional space.*

3.1.1 Key Assumptions

The PCA framework relies on several fundamental assumptions:

1. ***Linear Subspace Assumption****: The data lies approximately in a* k*-dimensional linear subspace of* $\mathbb{R}^d$
2. ***Gaussian Distribution****: The data follows a multivariate Gaussian distribution (optimal for maximum variance preservation)*
3. ***Centered Data****: The data has been centered, i.e.,* $\sum_{i=1}^{n} \mathbf{x}_i = \mathbf{0}$
4. ***Variance Maximization****: The most important information is captured by directions of maximum variance*

3.2 Matrix Factorization Perspective

From a matrix factorization viewpoint, PCA seeks to approximate the data matrix $\mathbf{X}$ *as:*

$$\mathbf{X} \approx \mathbf{W}\mathbf{H} \tag{18}$$

where:

- $\mathbf{W} \in \mathbb{R}^{d\times k}$ *contains the principal components (basis vectors)*
- $\mathbf{H} \in \mathbb{R}^{k\times n}$ *contains the coefficients (projected coordinates)*
- $k \ll d$ *ensures dimensionality reduction*

3.2.1 Optimization Formulation

The optimal factorization is obtained by solving:

$$\text{minimize}_{\mathbf{W},\mathbf{H}} \quad \|\mathbf{X} - \mathbf{W}\mathbf{H}\|_F^2 \quad \text{subject to} \quad \mathbf{W}^T\mathbf{W} = \mathbf{I}_k \tag{19}$$

where $\|\cdot\|_F$ *denotes the Frobenius norm and the constraint ensures orthonormality of the basis vectors.*

Theorem 3.2 (PCA Optimization Problem)**.** *The PCA problem can be equivalently formulated as:*

$$\text{minimize}_{\mathbf{W}} \quad \|\mathbf{X} - \mathbf{W}\mathbf{W}^T\mathbf{X}\|_F^2 \quad \text{subject to} \quad \mathbf{W}^T\mathbf{W} = \mathbf{I}_k \tag{20}$$

Proof. For a given $\mathbf{W}$, the optimal $\mathbf{H}$ is obtained by minimizing:

$$\|\mathbf{X} - \mathbf{W}\mathbf{H}\|_F^2 = \operatorname{tr}[(\mathbf{X} - \mathbf{W}\mathbf{H})^T(\mathbf{X} - \mathbf{W}\mathbf{H})] \tag{21}$$

Taking the derivative with respect to $\mathbf{H}$ and setting it to zero:

$$\frac{\partial}{\partial \mathbf{H}}\|\mathbf{X} - \mathbf{W}\mathbf{H}\|_F^2 = -2\mathbf{W}^T\mathbf{X} + 2\mathbf{W}^T\mathbf{W}\mathbf{H} = 0 \tag{22}$$

$$\Rightarrow \mathbf{H}^* = (\mathbf{W}^T\mathbf{W})^{-1}\mathbf{W}^T\mathbf{X} = \mathbf{W}^T\mathbf{X} \tag{23}$$

where the last equality uses the orthonormality constraint $\mathbf{W}^T\mathbf{W} = \mathbf{I}_k$. □

3.3 Eigenvalue Decomposition Solution

The solution to the PCA optimization problem is intimately connected to the eigenvalue decomposition of the data covariance matrix.

Definition 3.3 (Sample Covariance Matrix)**.** *For centered data* $\mathbf{X}$*, the sample covariance matrix is:*

$$\mathbf{C} = \frac{1}{n-1}\mathbf{X}\mathbf{X}^T \in \mathbb{R}^{d\times d} \tag{24}$$

Theorem 3.4 (PCA Solution via Eigendecomposition)**.** *The principal components are given by the eigenvectors of the covariance matrix* $\mathbf{C}$ *corresponding to the* k *largest eigenvalues.*

Specifically, if $\mathbf{C} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T$ *where* $\mathbf{\Lambda} = diag(\lambda_1, \lambda_2, \ldots, \lambda_d)$ *with* $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d \geq 0$*, then:*

$$\boxed{\mathbf{W} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k]} \tag{25}$$

where $\mathbf{v}_i$ *is the eigenvector corresponding to* λ_i*.*

Proof. We need to show that the eigenvectors of $\mathbf{C}$ solve the optimization problem in Theorem 3.2.

The objective function can be rewritten as:

$$\|\mathbf{X} - \mathbf{W}\mathbf{W}^T\mathbf{X}\|_F^2 = \operatorname{tr}[(\mathbf{X} - \mathbf{W}\mathbf{W}^T\mathbf{X})^T(\mathbf{X} - \mathbf{W}\mathbf{W}^T\mathbf{X})] \tag{26}$$

$$= \operatorname{tr}[\mathbf{X}^T\mathbf{X}] - 2\operatorname{tr}[\mathbf{X}^T\mathbf{W}\mathbf{W}^T\mathbf{X}] + \operatorname{tr}[\mathbf{X}^T\mathbf{W}\mathbf{W}^T\mathbf{W}\mathbf{W}^T\mathbf{X}] \tag{27}$$

$$= \operatorname{tr}[\mathbf{X}^T\mathbf{X}] - 2\operatorname{tr}[\mathbf{X}^T\mathbf{W}\mathbf{W}^T\mathbf{X}] + \operatorname{tr}[\mathbf{X}^T\mathbf{W}\mathbf{W}^T\mathbf{X}] \tag{28}$$

$$= \operatorname{tr}[\mathbf{X}^T\mathbf{X}] - \operatorname{tr}[\mathbf{X}^T\mathbf{W}\mathbf{W}^T\mathbf{X}] \tag{29}$$

Since $\operatorname{tr}[\mathbf{X}^T\mathbf{X}]$ is constant, minimizing the objective is equivalent to maximizing:

$$\operatorname{tr}[\mathbf{X}^T\mathbf{W}\mathbf{W}^T\mathbf{X}] = \operatorname{tr}[\mathbf{W}^T\mathbf{X}\mathbf{X}^T\mathbf{W}] = (n-1)\operatorname{tr}[\mathbf{W}^T\mathbf{C}\mathbf{W}] \tag{30}$$

Using Lagrange multipliers for the constraint $\mathbf{W}^T\mathbf{W} = \mathbf{I}_k$, the optimal $\mathbf{W}$ satisfies:

$$\mathbf{C}\mathbf{W} = \mathbf{W}\mathbf{\Lambda}_k \tag{31}$$

where $\mathbf{\Lambda}_k$ is a diagonal matrix of eigenvalues. This confirms that the columns of $\mathbf{W}$ are eigenvectors of $\mathbf{C}$. □

3.4 Variance Explained and Dimensionality Selection

Definition 3.5 (Explained Variance Ratio)**.** *The proportion of total variance explained by the first k principal components is:*

$$EVR(k) = \frac{\sum_{i=1}^{k} \lambda_i}{\sum_{i=1}^{d} \lambda_i} \tag{32}$$

Corollary 3.6 (Cumulative Variance Property)**.** *The eigenvalues $\{\lambda_i\}$ represent the variance captured by each principal component, and:*

$$\sum_{i=1}^{d} \lambda_i = \mathrm{tr}[\mathbf{C}] = \frac{1}{n-1} \sum_{i=1}^{n} \|\mathbf{x}_i\|_2^2 \tag{33}$$

Practical Dimensionality Selection Criteria:

1. ***Variance Threshold****: Choose k such that $EVR(k) \geq 0.95$*
2. ***Elbow Method****: Select k at the "elbow" of the eigenvalue plot*
3. ***Cross-Validation****: Use validation performance to guide selection*

4 Computational Algorithms and Implementation

4.1 Singular Value Decomposition Approach

While the eigendecomposition approach is theoretically elegant, the Singular Value Decomposition (SVD) provides a more numerically stable and computationally efficient alternative.

Theorem 4.1 (SVD-based PCA)**.** *Given the centered data matrix* $\mathbf{X} \in \mathbb{R}^{d\times n}$ *with SVD* $\mathbf{X} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$*, the principal components are given by the first* k *columns of* $\mathbf{U}$*:*

$$\mathbf{W} = \mathbf{U}_{:,1:k} \tag{34}$$

where the eigenvalues of the covariance matrix are $\lambda_i = \frac{\sigma_i^2}{n-1}$.

Proof. The covariance matrix can be expressed as:

$$\mathbf{C} = \frac{1}{n-1}\mathbf{X}\mathbf{X}^T \tag{35}$$

$$= \frac{1}{n-1}\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T\mathbf{V}\boldsymbol{\Sigma}^T\mathbf{U}^T \tag{36}$$

$$= \frac{1}{n-1}\mathbf{U}\boldsymbol{\Sigma}^2\mathbf{U}^T \tag{37}$$

This shows that $\mathbf{U}$ contains the eigenvectors of $\mathbf{C}$ and $\frac{\sigma_i^2}{n-1}$ are the corresponding eigenvalues. □

4.2 Algorithmic Implementation

Algorithm 1 Principal Component Analysis Algorithm

```
Require: Data matrix $\mathbf{X} \in \mathbb{R}^{d\times n}$, number of components $k$
Ensure: Principal components $\mathbf{W} \in \mathbb{R}^{d\times k}$, projected data $\mathbf{Y} \in \mathbb{R}^{k\times n}$
Center the data: $\mathbf{X} \leftarrow \mathbf{X} - \frac{1}{n}\mathbf{X}\mathbf{1}\mathbf{1}^T$
Compute SVD: $\mathbf{X} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$
Extract components: $\mathbf{W} \leftarrow \mathbf{U}_{:,1:k}$
Project data: $\mathbf{Y} \leftarrow \mathbf{W}^T\mathbf{X}$
Compute explained variance: $\text{EVR} \leftarrow \frac{\sum_{i=1}^{k}\sigma_i^2}{\sum_{i=1}^{\min(d,n)}\sigma_i^2}$
return W, Y, EVR
```

4.3 Computational Complexity Analysis

Theorem 4.2 (Computational Complexity)**.** *The computational complexity of PCA using SVD is:*

$$\mathcal{O}(\min(d^2n, dn^2)) \tag{38}$$

where d *is the dimensionality and* n *is the number of samples.*

Memory Requirements:

- ***Storage***: *$\mathcal{O}(dn + dk)$ for data matrix and principal components*
- ***Working Memory***: *$\mathcal{O}(d^2)$ for covariance matrix computation*

4.4 Numerical Stability Considerations

Remark 4.3 (Numerical Stability)**.** *The SVD-based approach is preferred over direct eigendecomposition for several reasons:*

1. *Avoids explicit computation of $\mathbf{X}\mathbf{X}^T$, which can be ill-conditioned*
2. *Provides better numerical accuracy for nearly singular matrices*
3. *Handles the case where $d > n$ more efficiently*

5 Advanced Topics and Extensions

5.1 Kernel PCA and Nonlinear Extensions

While classical PCA assumes linear relationships, many real-world datasets exhibit nonlinear structure. Kernel PCA extends the framework to capture nonlinear patterns.

Definition 5.1 (Kernel PCA)**.** *Kernel PCA performs PCA in a feature space $\mathcal{F}$ induced by a kernel function $k(\mathbf{x}, \mathbf{y}) = \langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle_{\mathcal{F}}$, where $\phi : \mathbb{R}^d \to \mathcal{F}$ is a potentially infinite-dimensional mapping.*

The kernel matrix $\mathbf{K} \in \mathbb{R}^{n \times n}$ with entries $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$ replaces the covariance matrix in the eigendecomposition:

$$\mathbf{K}\alpha = \lambda n \alpha \tag{39}$$

where α contains the coefficients for expressing principal components in terms of the training data.

5.2 Incremental and Online PCA

For large-scale applications, batch PCA may be computationally prohibitive. Incremental PCA algorithms update the principal components as new data arrives.

Theorem 5.2 (Incremental PCA Update Rule)**.** *Given existing principal components $\mathbf{W}^{(t)}$ and a new observation $\mathbf{x}_{t+1}$, the updated components can be computed using:*

$$\mathbf{W}^{(t+1)} = SVD_k\left([\mathbf{W}^{(t)}, \mathbf{x}_{t+1}]\right) \tag{40}$$

where SVD_k denotes truncated SVD keeping only the first k components.

5.3 Robust PCA and Outlier Detection

Classical PCA is sensitive to outliers due to its reliance on second-order statistics. Robust PCA variants address this limitation.

Definition 5.3 (Robust PCA Problem)**.** *Robust PCA decomposes the data matrix as:*

$$\mathbf{X} = \mathbf{L} + \mathbf{S} + \mathbf{N} \tag{41}$$

where:

- $\mathbf{L}$ *is a low-rank matrix (true signal)*
- $\mathbf{S}$ *is a sparse matrix (outliers)*
- $\mathbf{N}$ *is dense noise*

The optimization problem becomes:

$$\text{minimize}_{\mathbf{L},\mathbf{S}} \quad \|\mathbf{L}\|_* + \lambda\|\mathbf{S}\|_1 \quad \text{subject to} \quad \mathbf{L} + \mathbf{S} = \mathbf{X} \tag{42}$$

where $\|\cdot\|_*$ is the nuclear norm and $\|\cdot\|_1$ is the ℓ_1 norm.

6 Applications and Case Studies

6.1 Image Processing and Computer Vision

In computer vision applications, PCA serves multiple purposes:

Dimensionality Reduction: *Digital images with millions of pixels can be effectively represented using much fewer principal components. For an image dataset* $\{\mathbf{I}_1, \mathbf{I}_2, \ldots, \mathbf{I}_n\}$ *where each* $\mathbf{I}_i \in \mathbb{R}^{h \times w}$ *is vectorized, PCA finds the most significant visual patterns.*

Face Recognition (Eigenfaces): *The eigenfaces method represents facial images as linear combinations of principal components:*

$$\mathbf{I}_{face} \approx \mu + \sum_{i=1}^{k} w_i \mathbf{v}_i \tag{43}$$

where μ *is the mean face and* $\mathbf{v}_i$ *are the eigenfaces.*

6.2 Financial Data Analysis

In quantitative finance, PCA is used for:

Risk Factor Modeling: *Stock returns can be decomposed into systematic risk factors:*

$$\mathbf{r}_t = \mathbf{B}\mathbf{f}_t + \epsilon_t \tag{44}$$

where $\mathbf{f}_t$ *are the principal component factors and* $\mathbf{B}$ *contains factor loadings.*

Portfolio Optimization: *The reduced-dimension representation enables efficient portfolio optimization:*

$$\text{minimize}_{\mathbf{w}} \quad \mathbf{w}^T \boldsymbol{\Sigma} \mathbf{w} \quad \text{subject to} \quad \mathbf{w}^T \mu \geq r_{\min}, \quad \mathbf{w}^T \mathbf{1} = 1 \tag{45}$$

where $\boldsymbol{\Sigma}$ *is approximated using the first* k *principal components.*

6.3 Bioinformatics and Genomics

Gene Expression Analysis: *PCA reveals patterns in high-dimensional gene expression data:*

- ***Dimensionality****: Typically* $d \sim 10^4$ *genes,* $n \sim 10^2$ *samples*
- ***Interpretation****: Principal components often correspond to biological pathways*
- ***Visualization****: First 2-3 components enable 2D/3D visualization of sample relationships*

Population Genetics: *PCA of genetic variation data reveals population structure and migration patterns, with principal components correlating with geographical distances.*