Support Vector Machines and Principal Component Analysis: Mathematical Foundations and Applications

Naman Goel

October 21, 2025

Abstract

This report explores two fundamental machine learning techniques: Support Vector Machines (SVMs) for classification and Principal Component Analysis (PCA) for dimensionality reduction. We examine their mathematical foundations, practical applications, advantages, and limitations with illustrative examples.

1 Support Vector Machines (SVMs)

1.1 Mathematical Foundation

SVMs aim to find an optimal hyperplane that maximally separates different classes by maximizing the margin between support vectors [1, 2]. For a binary classification problem, the hyperplane is defined as:

$$\mathbf{w}^T\mathbf{x} + b = 0$$

where $\mathbf{w}$ is the normal vector and b is the bias term.

The optimization problem for soft-margin SVM is:

$$\min_{\mathbf{w},b,\zeta} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\zeta_i$$

subject to $y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \zeta_i$ and $\zeta_i \geq 0$, where C is the regularization parameter and ζ_i are slack variables for misclassifications [3].

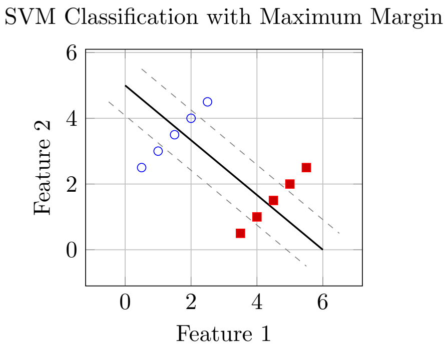

Figure 1: SVM finds the hyperplane with maximum margin between classes.

1.2 Key Concepts and Applications

SVMs use the **kernel trick** to handle non-linearly separable data by mapping inputs to higher-dimensional spaces. Common kernels include: - Linear: $K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T\mathbf{x}_j$ - RBF/Gaussian: $K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma\|\mathbf{x}_i - \mathbf{x}_j\|^2)$ - Polynomial: $K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i^T\mathbf{x}_j + 1)^q$

1.3 Advantages and Pitfalls

Advantages:

- Effective in high-dimensional spaces
- Memory efficient (only uses support vectors)
- Versatile with different kernel functions
- Works well with small to medium datasets

Pitfalls:

- Computationally expensive for large datasets
- Sensitive to feature scaling
- No probabilistic output
- Difficult to interpret and tune hyperparameters

2 Principal Component Analysis (PCA)

2.1 Mathematical Foundation

PCA transforms data to a lower-dimensional space while preserving maximum variance [4, 5]. Given a data matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$, PCA finds principal components as eigenvectors of the covariance matrix:

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

The eigenvalue decomposition gives:

$$\mathbf{C}\mathbf{v}_i = \lambda_i \mathbf{v}_i$$

where $\mathbf{v}_i$ are eigenvectors (principal components) and λ_i are eigenvalues representing variance along each component [6].

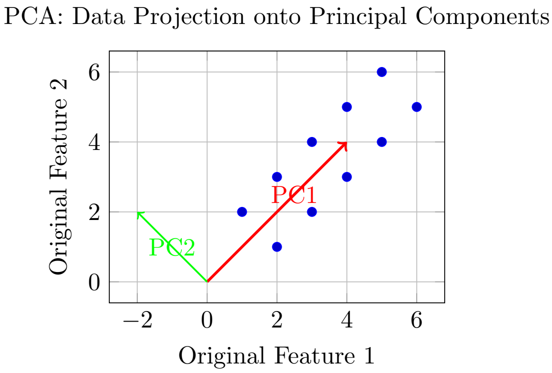

Figure 2: PCA projects data onto principal components maximizing variance.

2.2 Algorithm Steps

1. Standardize the data: $\mathbf{X}_{std} = \frac{\mathbf{X}-\boldsymbol{\mu}}{\boldsymbol{\sigma}}$
2. Compute covariance matrix: $\mathbf{C} = \frac{1}{n-1}\mathbf{X}_{std}^T\mathbf{X}_{std}$
3. Find eigenvalues and eigenvectors: $\mathbf{C}\mathbf{v}_i = \lambda_i \mathbf{v}_i$
4. Sort by eigenvalues and select top k components

5. Transform data: $\mathbf{Y} = \mathbf{X}_{std}\mathbf{W}$

The proportion of variance explained by component i is:

$$\text{Explained Variance Ratio}_i = \frac{\lambda_i}{\sum_{j=1}^{d} \lambda_j}$$

2.3 Applications and Use Cases

PCA is widely used for:

- **Dimensionality reduction** for visualization and computational efficiency
- **Feature extraction** in image processing and computer vision
- **Noise reduction** by removing components with small eigenvalues
- **Data compression** while preserving essential information

2.4 Advantages and Pitfalls

Advantages:

- Reduces computational complexity
- Eliminates multicollinearity
- Provides interpretable variance explanation
- Works well for linearly correlated features

Pitfalls:

- Linear transformation only (cannot capture non-linear relationships)
- Components may not be interpretable in original feature context
- Sensitive to feature scaling
- May lose important information in discarded components

3 Conclusion

SVMs and PCA serve complementary roles in machine learning. SVMs excel at classification tasks with complex decision boundaries through kernel methods, while PCA provides efficient dimensionality reduction for preprocessing and visualization [1, 4]. Understanding their mathematical foundations, strengths, and limitations is crucial for appropriate application in real-world scenarios.

References

[1] Support vector machine. *Wikipedia*. `https://en.wikipedia.org/wiki/Support_vector_machine`, accessed October 2025.

[2] GeeksforGeeks, "Support Vector Machine (SVM) Algorithm." `https://www.geeksforgeeks.org/machine-learning/support-vector-machine-algorithm/`, accessed October 2025.

[3] Scikit-learn, "Support Vector Machines." `https://scikit-learn.org/stable/modules/svm.html`, accessed October 2025.

[4] Principal component analysis. *Wikipedia*. https://en.wikipedia.org/wiki/Principal_component_analysis, accessed October 2025.

[5] Built In, "Principal Component Analysis (PCA): Explained Step-by-Step." https://builtin.com/data-science/step-step-explanation-principal-component-analysis, accessed October 2025.

[6] GeeksforGeeks, "Dimensionality Reduction with PCA: Selecting the Largest Eigenvalues and Eigenvectors." https://www.geeksforgeeks.org/machine-learning/dimensionality-reduction-with-pca-selecting-the-largest-eigenvalues-and-eigenvectors/, accessed October 2025.