Embedded method and Subspace method

Jery FOTO (+918734036128)

Content

- **Introduction**
- Definition and Overview of Both Methods
- Difference Between Embedded and Subspace Methods
- **Case Studies / Examples**
- **Conclusion**

Introduction

Imagine you're working with a dataset that has **thousands of features**—some useful, many irrelevant, and others just adding noise. In such scenarios, how do you make your model efficient, faster, and more accurate?

This is where **Embedded Methods** and **Subspace Methods** come into play. These two powerful techniques help us reduce dimensionality, identify the most important features, and simplify complex data while preserving its essence.

Today, I'll walk you through:

1. **What these methods are**,
2. **How they work**,
3. **Key differences**, and
4. **Real-world applications** that make them essential in machine learning and data science.

By the end, you'll see how these techniques strike the perfect balance between simplicity and performance in modern data-driven systems.

Definition and Overview of Both Methods

1. Embedded Methods

What are Embedded Methods?

- Embedded methods combine **feature selection** and **model training** into a single process.
- Feature importance is determined **during the learning phase** as the model optimizes its objective function.

How it works

Regularization Techniques

Penalize less important features by shrinking their coefficients.

Example: LASSO (L1 Regularization) and Ridge (L2 Regularization).

Tree-based Methods

Decision Trees prioritize features based on their ability to reduce error or impurity (e.g., Gini Index, Entropy).

Integration with Models

Feature selection happens **as part of model training**

Examples of Embedded Methods

1. **LASSO (Least Absolute Shrinkage and Selection Operator)**
   - Adds an **L1 penalty** to the model:

$$\text{Loss Function} = \text{Prediction Error} + \lambda \sum |w_j|$$

   - Features with small weights w_j are reduced to zero.
2. **Decision Trees**
   - Splits data at each node using the most **informative features**.
   - Automatically ranks features based on importance.

Advantages & Disadvantages

Advantages

- Integrated into the model, reducing computational overhead.
- Avoids overfitting by eliminating irrelevant features.
- Efficient for high-dimensional data.

Disadvantages

- Model-specific; cannot generalize to all algorithms.
- Regularization may miss non-linear relationships.

2. Subspace Methods

What are Subspace Methods?

- Subspace methods reduce the **dimensionality** of a dataset by projecting it onto a **lower-dimensional subspace**.
- Focus on retaining maximum **variance** or **class separability** in the data.

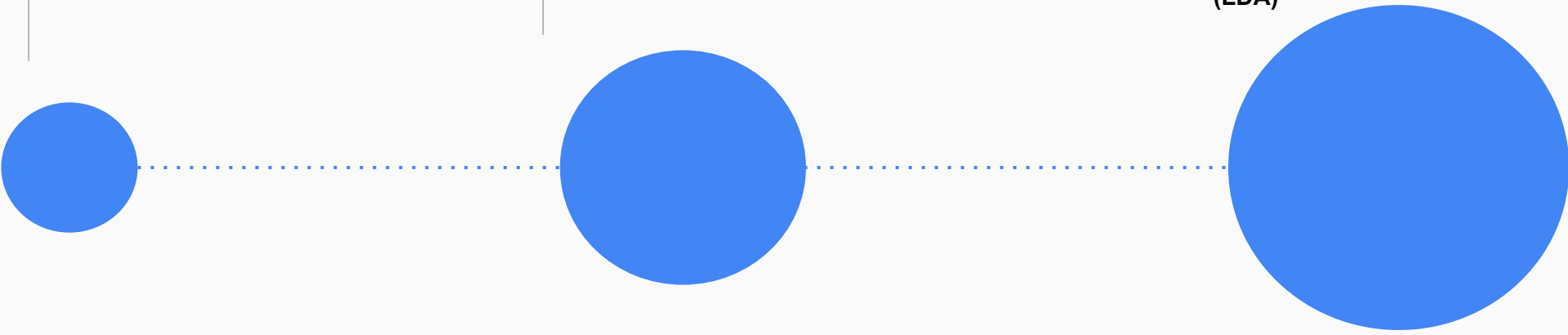
How it works

Dentify the directions (subspace) that capture the most important information in the data.

Transform data from a high-dimensional space to a lower-dimensional space.

Key techniques include

- **Principal Component Analysis (PCA)**
- **Linear Discriminant Analysis (LDA)**



Examples of Embedded Methods

Principal Component Analysis (PCA)

- Finds **principal components** (orthogonal directions) where data variance is maximized.
- Steps:
  - Standardize the data.
  - Compute the covariance matrix.
  - Extract eigenvalues and eigenvectors.
  - Project the data onto top kkk components.

PCA Formula:
Z=XWZ = X WZ=XW
Where WWW is the transformation matrix and ZZZ is the lower-dimensional representation.
Linear Discriminant Analysis (LDA)

- Maximizes **class separability** by projecting data onto a lower-dimensional space.
- Commonly used in classification tasks.

Advantages & Disadvantages

Advantages

- Reduces computational complexity for models.
- Simplifies data visualization and analysis.
- Retains most of the original information.

Disadvantages

- Transformation can make features less interpretable.
- Computationally expensive for very large datasets.

Key Difference

Aspect	Embedded Methods	Subspace Methods
Objective	Select features during training	Transform features into new space
Approach	Integrated into model training	Post-processing (feature extraction)
Techniques	LASSO, Ridge, Decision Trees	PCA, LDA, SVD
Output	Subset of original features	New set of transformed features
Use Case	Feature selection for models	Dimensionality reduction

Case Studies

Bringing Theory to Practice

Let’s explore a real-world example to see how **Embedded Methods** and **Subspace Methods** solve practical problems and improve performance.

Conclusion

- **Embedded Methods** and **Subspace Methods** are powerful techniques for handling high-dimensional data.
- **Embedded Methods** focus on feature selection during model training, ensuring efficiency and relevance.
- **Subspace Methods** transform data into lower dimensions, retaining key information and simplifying analysis.