

Complete Linear Discriminant Analysis (LDA) Notes

(With Parameter Estimation Details)

1. ★ What is LDA?

Linear Discriminant Analysis (LDA) is a supervised machine learning technique used for:

- **Dimensionality reduction**
- **Classification**
- **Finding axes that best separate different classes**

Core Idea: Find directions in the data where:

- Class means are **far apart** (between-class variance is large)
 - Points within each class are **close together** (within-class variance is small)
-

2. ☺ Two Equivalent Views of LDA

2.1 Bayes View (Probabilistic)

LDA is a **Bayesian classifier** using:

$$\$P(C_k \mid x) = \frac{P(x \mid C_k) \cdot P(C_k)}{P(x)}\$$$

Assumptions:

1. Each class follows a **Gaussian distribution**: $x \mid C_k \sim \mathcal{N}(\mu_k, \Sigma)$
2. All classes share the **same covariance matrix**: $\Sigma_1 = \Sigma_2 = \dots = \Sigma$

Under these assumptions, the optimal decision boundary is **linear**:

$$\$w_k^T x + b_k\$$$

2.2 Geometric View (Fisher LDA)

Fisher LDA finds a projection vector **w** that maximizes:

$$\$J(w) = \frac{w^T S_b w}{w^T S_w w}\$$$

Where:

- S_w = **within-class scatter** (spread inside each class)
- S_b = **between-class scatter** (separation between class means)

Solution (2-class case):

$$\$w = S_w^{-1} (\mu_1 - \mu_2)$$

This is the **best linear discriminant direction**.

2.3 Why Are They Equivalent?

Bayes classifier gives: $\$w \propto \Sigma^{-1} (\mu_1 - \mu_2)$

Fisher LDA gives: $\$w = S_w^{-1} (\mu_1 - \mu_2)$

Since $S_w \approx \Sigma$ (estimated covariance), both yield the **same direction**.

Bayes provides the probabilistic justification; Fisher provides the computational method.

3. Parameter Estimation (What We Learn From Data)

Parameters to Estimate:

Parameter	Symbol	What It Represents	How to Estimate
Prior probabilities	$\hat{\pi}_k = P(C_k)$	Proportion of each class	$\hat{\pi}_k = \frac{N_k}{N}$
Class means	$\hat{\mu}_k$	Center of each class	$\hat{\mu}_k = \frac{1}{N_k} \sum_{i: y_i = k} x_i$
Shared covariance	$\hat{\Sigma}$ or S_w	Common spread of all classes	$\hat{\Sigma} = \frac{1}{N-K} \sum_{k=1}^K \sum_{i: y_i = k} (x_i - \hat{\mu}_k)(x_i - \hat{\mu}_k)^T$
Overall mean	$\hat{\mu}$	Grand mean (multiclass)	$\hat{\mu} = \frac{1}{N} \sum_{i=1}^N x_i$
Projection vectors	w	Discriminant directions	From eigenvalue problem: $S_b w = \lambda S_w w$

Detailed Estimation Formulas:

1. Prior Probabilities

$$\hat{\pi}_k = \frac{N_k}{N}$$

Where:

- N_k = number of samples in class k
- N = total number of samples

2. Class Means

$$\hat{\mu}_k = \frac{1}{N_k} \sum_{x_i \in C_k} x_i$$

Average of all points belonging to class k .

3. Within-Class Scatter Matrix

$$S_w = \sum_{k=1}^K \sum_{x_i \in C_k} (x_i - \hat{\mu}_k)(x_i - \hat{\mu}_k)^T$$

$$\text{Alternative (pooled covariance): } \hat{\Sigma} = \frac{1}{N - K} \sum_{k=1}^K (N_k - 1) \hat{\Sigma}_k$$

Where $\hat{\Sigma}_k$ is the covariance of class k .

4. Between-Class Scatter Matrix

$$S_b = \sum_{k=1}^K N_k (\hat{\mu}_k - \hat{\mu})(\hat{\mu}_k - \hat{\mu})^T$$

Measures how far each class mean is from the global mean.

5. Projection Vectors

$$\text{Solve the generalized eigenvalue problem: } S_b w = \lambda S_w w$$

$$\text{Or equivalently: } S_w^{-1} S_b w = \lambda w$$

Take the top $K-1$ eigenvectors (those with largest eigenvalues).

4. Projection onto New Axes

Once we have the projection vector(s) w (or matrix W), transform data:

Single discriminant: $y = w^T x$

Multiple discriminants: $Y = W^T X$

Where:

- $W = [w_1, w_2, \dots, w_{K-1}]$ is a matrix of discriminant vectors
- Each column is one discriminant direction

Result: Data is projected into a lower-dimensional space where classes are maximally separated.

5. Multiclass LDA (K Classes)

Steps:

1. **Compute class means:** $\hat{\mu}_k$ for each class $k = 1, \dots, K$
2. **Compute overall mean:** $\hat{\mu} = \frac{1}{N} \sum x_i$
3. **Compute within-class scatter:** $S_w = \sum_k \sum_{x_i \in C_k} (x_i - \hat{\mu}_k)(x_i - \hat{\mu}_k)^T$

4. **Compute between-class scatter:** $S_b = \sum_k N_k (\mu_k - \mu)(\mu_k - \mu)^T$
5. **Solve eigenvalue problem:** $S_w^{-1} S_b w = \lambda w$
6. **Select top $K-1$ eigenvectors:** Form projection matrix W
7. **Project data:** $Y = W^T X$

Output dimensions: Maximum of $K - 1$ (you can't separate K classes in more than $K-1$ dimensions with linear boundaries).

6. LDA vs PCA

Aspect	LDA	PCA
Supervised?	Yes (uses class labels)	No (unsupervised)
Goal	Maximize class separation	Maximize variance
Criterion	$\frac{w^T S_b w}{w^T S_w w}$	$w^T \Sigma w$
Max dimensions	$K - 1$	d (original features)
Decision boundaries	Linear	Not a classifier
Use case	Classification + reduction	Dimensionality reduction only

Key difference:

PCA ignores class labels and finds directions of maximum variance.
LDA uses class labels to find directions of maximum class separability.

7. Intuitive Summary

What LDA wants:

- Classes **far apart** (large between-class scatter)
- Each class **tightly clustered** (small within-class scatter)

How it works:

1. Estimates class means and covariances from training data
2. Finds the best direction(s) to "look" at the data from
3. Projects data onto these direction(s)
4. Performs classification in the transformed space

Why it works:

- Mathematical guarantee from Bayes theorem (optimal under Gaussian assumptions)
 - Geometric intuition from Fisher's criterion (maximizing separation ratio)
-

8. Key Formulas Reference

Two-class discriminant: $w = S_w^{-1} (\mu_1 - \mu_2)$

Fisher's criterion: $J(w) = \frac{w^T S_b w}{w^T S_w w}$

Projection: $y = w^T x$

Classification (Bayes): $\hat{y} = \arg\max_k [\mu_k^T x + b_k]$

Where: $w_k = \Sigma^{-1} \mu_k$ $b_k = -\frac{1}{2} \mu_k^T \Sigma^{-1} \mu_k + \log(\pi_k)$

9. Complete LDA Algorithm

Training Phase:

Input: Training data $(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)$ where $y_i \in \{1, 2, \dots, K\}$

Steps:

1. **Estimate priors:** $\hat{\pi}_k = \frac{N_k}{N}$
2. **Estimate class means:** $\hat{\mu}_k = \frac{1}{N_k} \sum_{i:y_i=k} x_i$
3. **Estimate shared covariance:** $\hat{\Sigma} = \frac{1}{N-K} \sum_{k=1}^K \sum_{i:y_i=k} (x_i - \hat{\mu}_k)(x_i - \hat{\mu}_k)^T$
4. **Compute discriminant function parameters:** $w_k = \hat{\Sigma}^{-1} \hat{\mu}_k$ $b_k = -\frac{1}{2} \hat{\mu}_k^T \hat{\Sigma}^{-1} \hat{\mu}_k + \log(\hat{\pi}_k)$

Output: Parameters (w_k, b_k) for each class k

Prediction Phase:

Input: New data point x

Steps:

1. **Compute discriminant scores:** $\delta_k(x) = w_k^T x + b_k$ for all k
2. **Classify:** $\hat{y} = \arg\max_k \delta_k(x)$

Output: Predicted class label \hat{y}

Dimensionality Reduction (Fisher) Phase:

If you want to reduce dimensions for visualization or further processing:

1. **Compute overall mean:** $\mu = \frac{1}{N} \sum_{i=1}^N x_i$
2. **Compute within-class scatter:** $S_w = \sum_{k=1}^K \sum_{i:y_i=k} (x_i - \mu_k)(x_i - \mu_k)^T$
3. **Compute between-class scatter:** $S_b = \sum_{k=1}^K N_k (\mu_k - \mu)(\mu_k - \mu)^T$
4. **Solve eigenvalue problem:** $S_w^{-1} S_b w = \lambda w$

5. **Select top m eigenvectors** (where $m \leq K-1$): $W = [w_1, w_2, \dots, w_m]$

6. **Project data:** $y = W^T x$

10. 🎉 Final One-Sentence Summary

LDA estimates class means and a shared covariance from data, then uses these to find linear projection directions that maximize the ratio of between-class to within-class scatter, enabling optimal linear classification under Gaussian assumptions.

11. ⚠️ Important Notes

Assumptions:

- Classes are normally distributed
- Equal covariance matrices across all classes
- Linearly separable (or approximately so)

When LDA Fails:

- Highly non-Gaussian data
- Very different class covariances (use QDA instead)
- Non-linear decision boundaries needed (use kernel methods or non-linear classifiers)
- Small sample size with many features ($N < d \rightarrow S_w$ is singular)

Advantages:

- Simple and interpretable
- Computationally efficient
- Optimal under Gaussian assumptions
- Provides dimensionality reduction + classification
- Works well with small training sets (compared to neural networks)

Disadvantages:

- Strong assumptions (Gaussian + equal covariance)
- Linear boundaries only
- Sensitive to outliers
- Cannot handle more than $K-1$ discriminants

Computational Complexity:

- **Training:**
 - Estimating parameters: $O(Nd^2)$ where N = samples, d = features
 - Eigenvalue decomposition: $O(d^3)$
 - Total: $O(Nd^2 + d^3)$
- **Prediction:** $O(Kd)$ per sample (evaluate K discriminant functions)

12. Relationship to Other Methods

LDA vs QDA (Quadratic Discriminant Analysis):

- QDA relaxes the equal covariance assumption
- Each class has its own Σ_k
- Decision boundaries become **quadratic** instead of linear
- More flexible but requires more parameters to estimate

LDA vs Logistic Regression:

- Both produce linear decision boundaries
- LDA is **generative** (models $P(x|C_k)$)
- Logistic regression is **discriminative** (models $P(C_k|x)$ directly)
- LDA more efficient with small samples; logistic regression more robust

LDA as a Special Case:

- **LDA = Naive Bayes** with Gaussian distributions and shared covariance
- **LDA = Least Squares Classification** with specific encoding

13. Summary Checklist

What LDA Estimates:

- Prior probabilities π_k for each class
- Class means μ_k for each class
- Shared covariance matrix Σ (or S_w)
- Projection vectors w (for dimensionality reduction)
- Discriminant function weights w_k and biases b_k (for classification)

What LDA Produces:

- Linear decision boundaries
- Up to $K-1$ discriminant directions
- Reduced-dimensional representation
- Class predictions with posterior probabilities

What You Need to Know:

- The two views (Bayes + Fisher) are equivalent
- How to estimate all parameters from training data
- How to project data onto discriminant axes
- How to classify new points
- When LDA is appropriate vs when it fails
- How LDA differs from PCA and QDA

14. Exam/Study Tips

Key Derivations to Understand:

1. How equal covariance leads to linear boundaries
2. Why Fisher's criterion maximizes class separation
3. The eigenvalue problem for finding discriminant directions

Common Questions:

- "Derive the LDA discriminant function"
- "Why does LDA produce at most K-1 components?"
- "Compare LDA and PCA"
- "What are the assumptions of LDA and when do they break?"

Calculation Practice:

- Compute class means and covariance from data
 - Calculate discriminant scores
 - Find projection vectors for 2-class problems
 - Classify new points using discriminant functions
-

15. Further Reading

Topics to Explore:

- Quadratic Discriminant Analysis (QDA)
- Regularized Discriminant Analysis (RDA)
- Flexible Discriminant Analysis (FDA)
- Kernel LDA (for non-linear boundaries)
- Multi-class classification strategies

Mathematical Background:

- Multivariate Gaussian distributions
 - Maximum likelihood estimation
 - Eigenvalue decomposition
 - Bayesian decision theory
-

End of Notes

These notes cover everything from basic intuition to mathematical details of Linear Discriminant Analysis. Use the parameter estimation section (§3) as a quick reference for what LDA learns from data.