Contents

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**: $\Sigma_1 = \Sigma_2 = \cdots = \Sigma$

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

$$\delta_k(x) = w_k^T x + b_k$$

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	$\pi_k = P(C_k)$	Proportion of each class	$\hat{\pi}_k = \frac{N_k}{N}$
Class means	μ_k	Center of each class	$\hat{\mu}_k = \frac{1}{N_k} \sum_{i:y_i=k} x_i$

Parameter	Symbol	What It Represents	How to Estimate
Shared covariance	Σ 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	μ	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

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

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

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

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

---

5. Projection Vectors Solve the generalized eigenvalue problem:

$$S_b w = \lambda S_w w$$

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:

- $W = [w_1, w_2, \ldots, w_{K-1}]$ 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:** μ_k for each class $k = 1, \ldots, K$
2. **Compute overall mean:** $\mu = \frac{1}{N} \sum x_i$
3. **Compute within-class scatter:** $S_w = \sum_k \sum_{x \in C_k} (x - \mu_k)(x - \mu_k)^T$
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):

$$\hat{y} = \arg\max_k \left[w_k^T x + b_k \right]$$

Where:

$$w_k = \Sigma^{-1} \mu_k$$

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

---

9. 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.

---

10. Important Notes

Assumptions:

- Classes are normally distributed
- Equal covariance matrices
- Linearly separable (or close to it)

When LDA fails:

- Highly non-Gaussian data
- Very different class covariances
- Non-linear decision boundaries needed

Computational complexity:

- Estimating parameters: $O(Nd^2)$ where N = samples, d = features
- Eigenvalue decomposition: $O(d^3)$
- Prediction: $O(d)$ per sample

---

11. Summary of Complete LDA Algorithm

Training Phase:

1. **Estimate parameters from training data:**
   - Prior probabilities: $\hat{\pi}_k = \frac{N_k}{N}$
   - Class means: $\hat{\mu}_k = \frac{1}{N_k} \sum_{x_i \in C_k} x_i$
   - Pooled covariance: $\hat{\Sigma} = \frac{1}{N-K} \sum_k \sum_{x_i \in C_k} (x_i - \hat{\mu}_k)(x_i - \hat{\mu}_k)^T$
2. **Compute scatter matrices:**
   - Within-class: S_w
   - Between-class: S_b
3. **Find discriminant directions:**
   - Solve: $S_w^{-1} S_b w = \lambda w$
   - Take top $K - 1$ eigenvectors

Prediction Phase:

1. **Project new data:** $y = W^T x$
2. **Classify using:**

$$\hat{y} = \arg\max_k \left[\log(\hat{\pi}_k) - \frac{1}{2}(x - \hat{\mu}_k)^T \hat{\Sigma}^{-1} (x - \hat{\mu}_k) \right]$$

---

12. What Makes LDA "Linear"?

The decision boundaries are **hyperplanes** in the original feature space because:

1. The discriminant functions are **linear combinations** of the input features
2. Under Gaussian assumptions with equal covariances, the log-posterior becomes quadratic, but equal covariances cause quadratic terms to cancel
3. The result is a decision rule of the form: $w^T x + b = 0$

This is fundamentally different from:

- **QDA (Quadratic Discriminant Analysis):** Allows different covariances $\rightarrow$ quadratic boundaries
- **Logistic Regression:** Uses a different criterion (maximum likelihood) but also produces linear boundaries
- **Perceptron/SVM:** Different optimization objectives but similar linear geometry

---

13. Practical Considerations

When to use LDA:

- Classes are roughly Gaussian
- Sample size is sufficient ($N \gg d$)
- Need interpretable features
- Want probabilistic predictions
- Classes have similar spread

Alternatives to consider:

- **QDA:** When class covariances differ significantly
- **Regularized LDA:** When d is large relative to N
- **Kernel LDA:** For non-linear decision boundaries
- **PCA + Classifier:** When dimensionality reduction without supervision is preferred

Common issues:

- **Singularity of S_w:** Occurs when $N < d$ or features are linearly dependent
- **Solution:** Use regularization or dimensionality reduction first
- **Imbalanced classes:** Prior probabilities heavily favor majority class
- **Solution:** Adjust priors or use balanced sampling

Created for ST310 Machine Learning Course
Last Updated: November 2025