# Quadratic Discriminant Analysis (QDA)

## Complete Guide with Intuition, Examples, and Mathematics

---

## 1. What is QDA? (The Big Picture)

**Quadratic Discriminant Analysis (QDA)** is the **relaxed cousin of LDA**.

**The ONE key difference:** - **LDA:** All classes share the SAME covariance matrix → Linear boundaries - **QDA:** Each class has its OWN covariance matrix → Quadratic (curved) boundaries

That's it! Everything else follows from this one change.

---

## 2. Intuitive Understanding

### The Basketball vs Football Analogy

Imagine you're trying to classify sports balls by their measurements (width and weight).

**LDA says:** > "All balls have the same SHAPE of spread (same covariance), just different centers." > > Like basketballs and footballs that are both circular clouds, just in different locations. > > Decision: Draw a straight line between the centers.

**QDA says:** > "Different types of balls can have DIFFERENT SHAPES of spread." > > Basketballs form a circular cloud (equal variance in all directions). > Footballs form an elliptical cloud (longer in one direction). > > Decision: Use a CURVED boundary that accounts for these different shapes.

### Visual Intuition

```
LDA (same covariance):
    Class 1:           Class 2:

                  |
                  |
                  straight
                  boundary

QDA (different covariances):
    Class 1:           Class 2:

                       curved
                      boundary
         (wide circle)            (tall ellipse)
```

---

## 3. The Mathematics (From Bayes to Quadratic)

### 3.1 Starting Point: Bayes' Rule (Same as LDA)

We want to classify a new point $x$ into class $k$:

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

To classify, we pick the class with highest posterior probability:

$$\hat{y} = \arg\max_k P(C_k \mid x)$$

Since $P(x)$ is the same for all classes, we can ignore it:

$$\hat{y} = \arg\max_k \left[P(x \mid C_k) \cdot P(C_k)\right]$$

---

### 3.2 The Gaussian Assumption

Each class follows a **multivariate Gaussian distribution**:

$$P(x \mid C_k) = \frac{1}{(2\pi)^{d/2}|\Sigma_k|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu_k)^T\Sigma_k^{-1}(x-\mu_k)\right)$$

**KEY DIFFERENCE FROM LDA:** - Notice $\Sigma_k$ has subscript $k \rightarrow$ **each class has its own covariance** - In LDA, it was just $\Sigma$ (same for all classes)

---

### 3.3 Taking the Log (To Simplify)

Working with logs is easier (and doesn't change the arg max):

$$\log P(C_k \mid x) = \log P(x \mid C_k) + \log P(C_k) + \text{constant}$$

Substituting the Gaussian formula:

$$\log P(C_k \mid x) = \log P(C_k) - \frac{d}{2}\log(2\pi) - \frac{1}{2}\log|\Sigma_k| - \frac{1}{2}(x-\mu_k)^T\Sigma_k^{-1}(x-\mu_k)$$

Drop constants (same for all classes):

$$\log P(C_k \mid x) = \log P(C_k) - \frac{1}{2}\log|\Sigma_k| - \frac{1}{2}(x-\mu_k)^T\Sigma_k^{-1}(x-\mu_k)$$

---

### 3.4 The Discriminant Function (QDA)

Define the **quadratic discriminant function**:

$$\boxed{\delta_k(x) = -\frac{1}{2}\log|\Sigma_k| - \frac{1}{2}(x-\mu_k)^T\Sigma_k^{-1}(x-\mu_k) + \log\pi_k}$$

Where: - $\mu_k$ = mean of class $k$ - $\Sigma_k$ = covariance of class $k$ (different for each class!) - $\pi_k = P(C_k)$ = prior probability of class $k$

**Classification rule:**

$$\hat{y} = \arg\max_k \delta_k(x)$$

---

### 3.5 Why "Quadratic"? (Expanding the Math)

Let's expand the quadratic term:

$$(x-\mu_k)^T\Sigma_k^{-1}(x-\mu_k) = x^T\Sigma_k^{-1}x - 2\mu_k^T\Sigma_k^{-1}x + \mu_k^T\Sigma_k^{-1}\mu_k$$

So:

$$\delta_k(x) = -\frac{1}{2}x^T\Sigma_k^{-1}x + \mu_k^T\Sigma_k^{-1}x + \text{constant}_k$$

This is **quadratic in** $x$ because of the $x^T\Sigma_k^{-1}x$ term!

**Crucially:** Since $\Sigma_k$ is different for each class, the quadratic term $x^T\Sigma_k^{-1}x$ doesn't cancel out when comparing classes.

---

### 3.6 Comparison: LDA vs QDA Decision Functions

**LDA:**

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

- Linear in $x$ (form: $w^Tx + b$) - $\Sigma$ is the same for all classes $\rightarrow$ $x^T\Sigma^{-1}x$ cancels when comparing

**QDA:**

$$\delta_k^{QDA}(x) = -\frac{1}{2}x^T\Sigma_k^{-1}x + \mu_k^T\Sigma_k^{-1}x - \frac{1}{2}\mu_k^T\Sigma_k^{-1}\mu_k - \frac{1}{2}\log|\Sigma_k| + \log\pi_k$$

- Quadratic in $x$ (form: $x^TAx + w^Tx + c$) - $\Sigma_k$ is different $\rightarrow$ $x^T\Sigma_k^{-1}x$ DOESN'T cancel

---

## 4. Parameter Estimation (What We Learn From Data)

**Parameters to Estimate:**

| Parameter | Symbol | What It Represents | How to Estimate |
|---|---|---|---|
| **Prior probabilities** | $\pi_k$ | Proportion of each class | $\hat{\pi}_k = \frac{N_k}{N}$ |
| **Class means** | $\mu_k$ | Center of each class | $\hat{\mu}_k = \frac{1}{N_k} \sum_{i:y_i=k} x_i$ |
| **Class covariances** | $\Sigma_k$ | **Separate** spread for each class | $\hat{\Sigma}_k = \frac{1}{N_k-1} \sum_{i:y_i=k}(x_i - \hat{\mu}_k)(x_i - \hat{\mu}_k)^T$ |

**Key Difference from LDA:**

**LDA:** Estimate ONE shared covariance $\Sigma$

$$\hat{\Sigma} = \frac{1}{N-K} \sum_{k=1}^{K} \sum_{i:y_i=k} (x_i - \hat{\mu}_k)(x_i - \hat{\mu}_k)^T$$

**QDA:** Estimate K separate covariances $\Sigma_1, \Sigma_2, \ldots, \Sigma_K$

$$\hat{\Sigma}_k = \frac{1}{N_k - 1} \sum_{i:y_i=k} (x_i - \hat{\mu}_k)(x_i - \hat{\mu}_k)^T$$

---

## 5. Complete QDA Algorithm

### Training Phase:

```
For each class k = 1, 2, ..., K:
    1. Compute prior: ^_k = N_k / N
    2. Compute mean: ^_k = (1/N_k) Σ x_i  (for all x_i in class k)
    3. Compute covariance: Σ_k = (1/(N_k-1)) Σ (x_i - ^_k)(x_i - ^_k)^T
```

### Prediction Phase:

```
For new point x:
    1. For each class k, compute:
       _k(x) = -½ log|Σ_k| - ½(x - ^_k)^T Σ_k^(-1) (x - ^_k) + log(^_k)

    2. Classify: ŷ = argmax_k  _k(x)
```

---

## 6. Geometric Intuition: Why Quadratic Boundaries?

### Example: Two Classes in 2D

**Class 1:** Wide horizontal spread

$$\Sigma_1 = \begin{bmatrix} 4 & 0 \\ 0 & 1 \end{bmatrix}$$

**Class 2:** Wide vertical spread

$$\Sigma_2 = \begin{bmatrix} 1 & 0 \\ 0 & 4 \end{bmatrix}$$

The decision boundary is where $\delta_1(x) = \delta_2(x)$:

$$-\frac{1}{2}x^T\Sigma_1^{-1}x + \text{linear terms}_1 = -\frac{1}{2}x^T\Sigma_2^{-1}x + \text{linear terms}_2$$

Since $\Sigma_1^{-1} \neq \Sigma_2^{-1}$, the $x^T\Sigma_k^{-1}x$ terms DON'T cancel!

This leaves us with:

$$x^T(\Sigma_2^{-1} - \Sigma_1^{-1})x + \text{linear terms} = 0$$

This is a **quadratic equation** → gives ellipses, parabolas, or hyperbolas as decision boundaries!

---

## 7. Worked Example (2D, 2 Classes)

**Setup:**

**Class 1 (Red):** - Mean: $\mu_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ - Covariance: $\Sigma_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ (circular) - Prior: $\pi_1 = 0.5$

**Class 2 (Blue):** - Mean: $\mu_2 = \begin{bmatrix} 3 \\ 0 \end{bmatrix}$ - Covariance: $\Sigma_2 = \begin{bmatrix} 4 & 0 \\ 0 & 0.5 \end{bmatrix}$ (elliptical, wide horizontally) - Prior: $\pi_2 = 0.5$

**Question: Classify point** $x = \begin{bmatrix} 1.5 \\ 0 \end{bmatrix}$

**Step 1: Compute** $\delta_1(x)$

$$\Sigma_1^{-1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$|\Sigma_1| = 1$$

$$(x - \mu_1)^T\Sigma_1^{-1}(x - \mu_1) = \begin{bmatrix} 1.5 \\ 0 \end{bmatrix}^T \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1.5 \\ 0 \end{bmatrix} = 2.25$$

$$\delta_1(x) = -\frac{1}{2}\log(1) - \frac{1}{2}(2.25) + \log(0.5)$$
$$= 0 - 1.125 - 0.693 = -1.818$$

**Step 2: Compute** $\delta_2(x)$

$$\Sigma_2^{-1} = \begin{bmatrix} 0.25 & 0 \\ 0 & 2 \end{bmatrix}$$

$$|\Sigma_2| = 4 \times 0.5 = 2$$

$$(x-\mu_2)^T\Sigma_2^{-1}(x-\mu_2) = \begin{bmatrix} -1.5 \\ 0 \end{bmatrix}^T \begin{bmatrix} 0.25 & 0 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} -1.5 \\ 0 \end{bmatrix} = 0.5625$$

$$\begin{aligned} \delta_2(x) &= -\frac{1}{2}\log(2) - \frac{1}{2}(0.5625) + \log(0.5) \\ &= -0.347 - 0.281 - 0.693 = -1.321 \end{aligned}$$

**Step 3: Classification**

$$\delta_2(x) = -1.321 > \delta_1(x) = -1.818$$

**Classify as Class 2 (Blue)!**

**Intuition:** Even though $x = [1.5, 0]$ is exactly halfway between the means, Class 2 wins because: 1. Class 2 has larger variance in the horizontal direction 2. The point falls within Class 2's wide horizontal spread 3. It's relatively far from Class 1's tight circular distribution

---

## 8. LDA vs QDA: Complete Comparison

| Aspect | LDA | QDA |
|---|---|---|
| **Covariance assumption** | Same for all classes ($\Sigma$) | Different per class ($\Sigma_k$) |
| **Decision boundary** | Linear (hyperplane) | Quadratic (ellipse, parabola, hyperbola) |
| **Parameters to estimate** | $K$ means + 1 covariance | $K$ means + $K$ covariances |
| **Number of parameters** | $Kd + \frac{d(d+1)}{2}$ | $Kd + K\frac{d(d+1)}{2}$ |
| **Flexibility** | Less flexible (more bias) | More flexible (less bias) |
| **Sample size needed** | Smaller | Larger (needs more data) |
| **Overfitting risk** | Lower | Higher |
| **When to use** | Similar class spreads | Very different class spreads |
| **Discriminant function** | $w_k^T x + b_k$ | $x^T A_k x + w_k^T x + c_k$ |

---

## 9. When to Use QDA vs LDA

**Use LDA when:**

Classes have similar covariance structures You have limited training data ($N$ is small relative to $d$) You want simpler, more interpretable model You want to avoid overfitting Decision boundaries look roughly linear

**Use QDA when:**

Classes have clearly different spreads/shapes You have plenty of training data ($N \gg Kd^2$) Decision boundaries are clearly non-linear You can afford more parameters Classes have very different structures (e.g., one tight, one spread out)

**Example Scenarios:**

**LDA is better:** - Classifying similar-sized fruits by weight and diameter - Text classification with normalized features - Medical diagnosis where symptoms have similar variability

**QDA is better:** - Classifying animals with very different sizes (mouse vs elephant) - Financial data where volatility varies by market regime - Image classification where object scales vary dramatically

---

## 10. The Bias-Variance Tradeoff

**LDA:** - **Higher bias** (assumes same covariance) - **Lower variance** (fewer parameters to estimate) - More stable with small datasets

**QDA:** - **Lower bias** (allows different covariances) - **Higher variance** (more parameters to estimate) - Needs more data to estimate reliably

**Rule of thumb:** - If you have < 30-50 samples per class per feature → use LDA - If you have > 50-100 samples per class per feature → consider QDA

---

## 11. Mathematical Properties

### 11.1 Decision Boundary Equation

The boundary between classes $i$ and $j$ is where $\delta_i(x) = \delta_j(x)$:

$$
\begin{aligned}
&-\frac{1}{2}x^T\Sigma_i^{-1}x + \mu_i^T\Sigma_i^{-1}x - \frac{1}{2}\log|\Sigma_i| + \text{const}_i \\
&= -\frac{1}{2}x^T\Sigma_j^{-1}x + \mu_j^T\Sigma_j^{-1}x - \frac{1}{2}\log|\Sigma_j| + \text{const}_j
\end{aligned}
$$

Rearranging:

$$
x^T\left(\frac{\Sigma_j^{-1} - \Sigma_i^{-1}}{2}\right)x + (\mu_i^T\Sigma_i^{-1} - \mu_j^T\Sigma_j^{-1})x + C = 0
$$

This is a **conic section** (ellipse, parabola, or hyperbola)!

---

### 11.2 Mahalanobis Distance Interpretation

QDA classifies based on **class-specific Mahalanobis distances**:

$$d_k(x) = (x - \mu_k)^T \Sigma_k^{-1} (x - \mu_k)$$

This measures how many "standard deviations" $x$ is from $\mu_k$ in the direction defined by $\Sigma_k$.

**The twist:** Each class uses its own "ruler" ($\Sigma_k$) to measure distance!

**LDA:** All classes use the same ruler ($\Sigma$) **QDA:** Each class uses its own custom ruler ($\Sigma_k$)

---

## 12. Computational Complexity

**Training:**

- Compute $K$ means: $O(Nd)$
- Compute $K$ covariances: $O(Nd^2K)$
- **Total:** $O(Nd^2K)$

**Prediction (per sample):**

- Compute $K$ discriminant functions: $O(d^2K)$ (matrix-vector products)
- **Total:** $O(d^2K)$ per classification

**Comparison to LDA:** - LDA training: $O(Nd^2)$ (faster, only one covariance) - LDA prediction: $O(dK)$ (faster, precomputed linear terms)

---

## 13. Practical Considerations

**Issues with QDA:**

1. **Singular covariance matrices:**
    - Happens when $N_k < d$ (not enough samples in class $k$)
    - Solution: Regularization or use LDA
2. **Numerical stability:**
    - Inverting $K$ covariance matrices can be unstable
    - Solution: Add small value to diagonal (regularization)
3. **Overfitting:**
    - Many parameters can lead to overfitting with small data
    - Solution: Cross-validation, use LDA, or regularized QDA

**Regularized QDA:**

Compromise between LDA and QDA:

$$\Sigma_k(\alpha) = \alpha \Sigma_k + (1 - \alpha)\Sigma$$

Where: - $\alpha = 0 \rightarrow$ LDA (pooled covariance) - $\alpha = 1 \rightarrow$ QDA (separate covariances) - $0 < \alpha < 1 \rightarrow$ Shrinks class covariances toward pooled covariance

---

## 14. Summary: The One-Sentence Explanation

> **QDA is LDA without the equal covariance assumption, allowing each class to have its own spread, which leads to quadratic (curved) decision boundaries instead of linear ones, at the cost of requiring more data and more parameters to estimate.**

---

## 15. Key Takeaways

1. **QDA = LDA + different covariances per class**
2. **Quadratic boundaries** come from the $x^T\Sigma_k^{-1}x$ term not canceling
3. **More flexible but needs more data** (bias-variance tradeoff)
4. **Use QDA when** classes have obviously different shapes/spreads
5. **Each class has its own Mahalanobis distance metric**

---

**Created for ST310 Machine Learning Course**
**Last Updated:** November 2025