Table of Contents

1. An Overview of Classification
2. Why Not Linear Regression
3. Motivation and Background
4. The Logistic Function and Transformations
5. Cost Function and Optimization
6. Assumptions, Limitations, and Conclusion
7. LDA and QDA

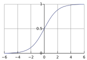
The Logistic (Sigmoid) Function

Definition

$$p(\mathbf{x}) = \sigma(t = \theta^T \mathbf{x}) = \frac{e^{\theta_0 + \theta_1 x_1}}{1 + e^{\theta_0 + \theta_1 x_1}}$$

- S-shaped (sigmoid) curve bounded between 0 and 1.
- Changes slowly at the tails and rapidly near the midpoint.
- **Numerical Example:**
  - Let $\theta_0 = -4$ and $\theta_1 = 0.02$.
  - For $x_1 = 50$: $t = -4 + 0.02 \times 50 = -3$, so

$$p(50) = \frac{e^{-3}}{1 + e^{-3}} \approx \frac{0.0498}{1.0498} \approx 0.0474.$$



  - For $x_1 = 200$: $t = -4 + 0.02 \times 200 = 0$, hence $p(200) = 0.5$.
  - For $x_1 = 300$: $t = -4 + 0.02 \times 300 = 2$, so

$$p(300) = \frac{e^2}{1 + e^2} \approx \frac{7.389}{8.389} \approx 0.88.$$

The Logit Transformation

Key Relationships

- The **logit function** is the log-odds:

$$t = \text{logit}(p) = \log\left(\frac{p}{1-p}\right)$$

- The logistic function is its inverse:

$$p = \sigma(t) = \frac{1}{1+e^{-t}}$$

- And the linear predictor is given by $t = \theta^\intercal \mathbf{x}$

Example

Numerical Example If $\widehat{p} = 0.7$, then

$$\text{log-odds} = \log\left(\frac{0.7}{0.3}\right) \approx \log(2.33) \approx 0.85.$$

Interpreting Odds and Log-Odds

Key Concepts

- **Odds:** The ratio $\frac{p(\mathbf{x})}{1-p(\mathbf{x})}$.
- **Log-Odds:**

$$\log\left(\frac{p(\mathbf{x})}{1-p(\mathbf{x})}\right) = t = \theta_0 + \theta_1 x_1$$

Example

Coefficient Interpretation If $\theta_1 = 0.5$, then for every unit increase in x_1:

- The log-odds increases by 0.5.
- The odds multiply by $e^{0.5} \approx 1.65$, meaning the odds of a positive outcome are 65% higher.

Prediction Rule: Thresholding

Threshold-based Prediction

$$\hat{y} = \begin{cases} 0 & \text{if } \widehat{p} < 0.5 \\ 1 & \text{if } \widehat{p} \geq 0.5 \end{cases}$$

- The decision boundary occurs when the linear predictor $t = \theta^\intercal \mathbf{x} = 0$.
- One predicts class 1 if the corresponding logit is nonnegative.
- The threshold (here, 0.5) can be adjusted to suit different risk tolerances.
- **Example:** If $\hat{p} = 0.3$, predict 0; if $\hat{p} = 0.8$, predict 1.

Table of Contents

1. An Overview of Classification
2. Why Not Linear Regression
3. Motivation and Background
4. The Logistic Function and Transformations
5. Cost Function and Optimization
6. Assumptions, Limitations, and Conclusion
7. LDA and QDA

Per-Instance Cost Function

Definition

$$c(\theta) = \begin{cases} -\log(\widehat{p}) & \text{if } y = 1 \\ -\log(1-\widehat{p}) & \text{if } y = 0 \end{cases}$$

- This cost penalizes confident but wrong predictions—cost increases steeply.
- **Example:** If $y = 1$ but $\widehat{p} = 0.1$, then cost $= -\log(0.1) \approx 2.3$.

Log Loss: Overall Cost Function

Logistic Regression Cost Function

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}\log\left(\widehat{p}^{(i)}\right) + (1 - y^{(i)})\log\left(1 - \widehat{p}^{(i)}\right)\right]$$

- This function, also known as log loss, is convex—ensuring a global minimum.
- It is derived from the maximum likelihood estimation approach.

Likelihood Function

$$\ell(\theta) = \prod_{i:y_i=1} p(x_i) \prod_{i:y_i=0} [1 - p(x_i)]$$

- Sensitive to extreme predictions; even one outlier can have a high cost.

Optimization and Gradient Descent

Gradient of the Cost Function

$$\frac{\partial J(\theta)}{\partial \theta_j} = \frac{1}{m} \sum_{i=1}^{m} \left[\sigma(\theta^\mathsf{T} \mathbf{x}^{(i)}) - y^{(i)} \right] x_j^{(i)}$$

- **Batch Gradient Descent:** Uses the entire dataset per update.
- **Stochastic Gradient Descent:** Updates parameters using a single instance.
- **Mini-batch Gradient Descent:** Uses a subset of data for each update.
- Note: Unlike linear regression, there is no closed-form solution.

Practical Training Tips

- **Feature Scaling:** Normalizing features is critical for the efficiency of gradient descent.
- **Learning Rate:** Choose carefully; consider adaptive learning rates or a line search.
- **Regularization:** To prevent overfitting, add a penalty such as:

$$\frac{\lambda}{2m}\|\theta\|^2$$

- **Convergence Monitoring:** Regularly check the cost and stop when changes become negligible.

Example: Iris dataset

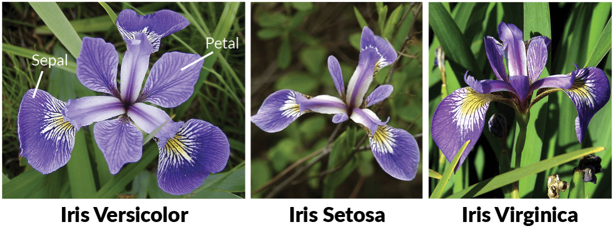


Figure: Iris Flower.

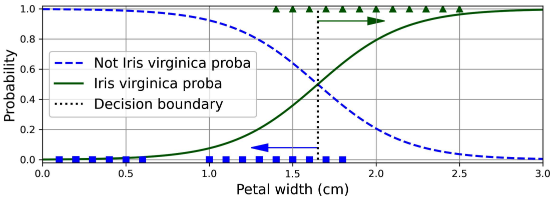


Figure: Estimated probabilities and decision boundary.

Multivariate Logistic Regression

$$\log\left(\frac{p(\mathbf{x})}{1-p(\mathbf{x})}\right) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_p X_p$$

- Incorporates multiple predictors simultaneously.
- **Example:**
  - Univariate Model: In a study, being "Young" (e.g., under 25 years old) might show a higher car accident rate.
  - Multivariate Model: When adjusting for driving experience (e.g., years of driving), "Young" drivers might not have an inherently higher risk—their inexperience may explain the correlation.

Example: Iris dataset, Multivariate Reg.

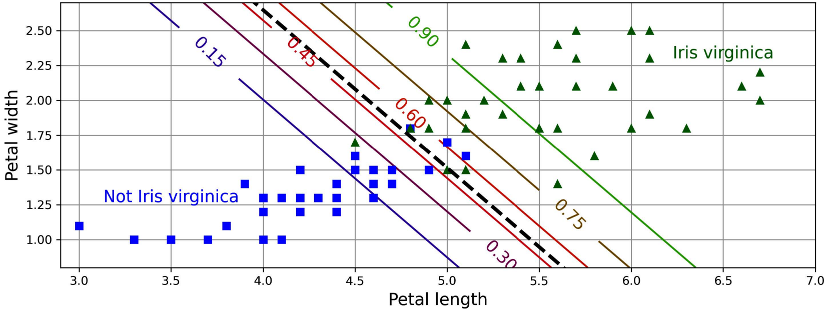


Figure: Using two features in Regression.

Multi-Class Logistic Regression

In multinomial (multi-class) logistic regression, one category is chosen as the baseline (often the K^{th} class). For each class $k \in \{1, \ldots, K-1\}$, the probability is given by:

$$\Pr(Y = k \mid X) = \frac{e^{\theta_{k0}+\theta_{k1}X_1+\cdots+\theta_{kp}X_p}}{1 + \sum_{l=1}^{K-1} e^{\theta_{l0}+\theta_{l1}X_1+\cdots+\theta_{lp}X_p}}$$

For the baseline category (class K):

$$\Pr(Y = K \mid X) = \frac{1}{1 + \sum_{l=1}^{K-1} e^{\theta_{l0}+\theta_{l1}X_1+\cdots+\theta_{lp}X_p}}$$

Multi-Class Logistic Regression

- This formulation ensures that all predicted probabilities are non-negative and sum to 1.
- While this approach models the probabilities of all classes simultaneously, an alternative is the One-vs.-Rest strategy, where separate binary classifiers are built for each class.
- Common examples include medical diagnosis (e.g., predicting whether a patient is Healthy, has a Cold, or has the Flu) and image recognition (e.g., classifying objects among several categories).
- An alternative approach is using Softmax coding as follows:

$$\Pr(Y = k \mid X) = \frac{e^{\theta_{k0} + \theta_{k1} X_1 + \cdots + \theta_{kp} X_p}}{\sum_{l=1}^{K} e^{\theta_{l0} + \theta_{l1} X_1 + \cdots + \theta_{lp} X_p}}$$

Multi-Class Logistic Regression, Example

The cost function is obtained using Cross entropy cost function:

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}\sum_{k=1}^{K}\left[y_k^{(i)}\log\left(\widehat{p}_k^{(i)}\right)\right]$$

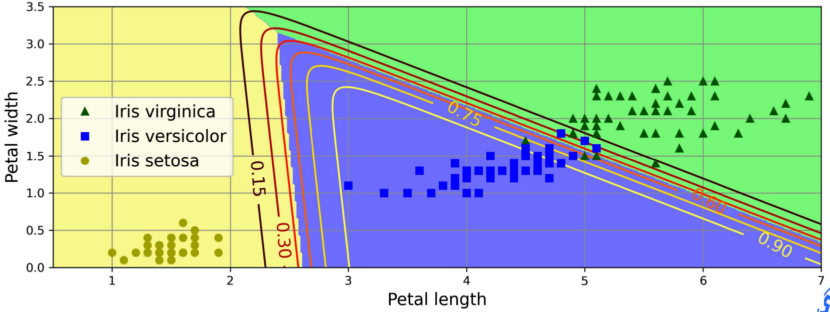


Table of Contents

1. An Overview of Classification
2. Why Not Linear Regression
3. Motivation and Background
4. The Logistic Function and Transformations
5. Cost Function and Optimization
6. Assumptions, Limitations, and Conclusion
7. LDA and QDA

Assumptions and Limitations

- **Linear Decision Boundary:**
  - Assumes that the log-odds of the outcome are a *linear combination* of the predictors.
  - This means any non-linear relationship must be handled through transformations or additional terms.
- **Feature Independence:**
  - Predictors should have minimal multicollinearity.
  - High correlation between features can lead to unstable and unreliable estimates.
- **Distribution of Errors:**
  - Implicitly assumes that the error distribution is roughly logistic.
  - Deviations from this assumption might affect model performance.
- **Sample Size:**
  - A general guideline is to have about 10 events (i.e., occurrences of the less frequent outcome) per predictor.
  - This helps ensure stable and reliable parameter estimates.

Conclusion and Key Takeaways

When to Use Logistic Regression

- Baseline Model: Ideal for binary and categorical classification tasks due to its simplicity.
- Interpretability: Provides clear, interpretable estimates of feature importance through odds ratios.
- Probability Estimates: Outputs well-calibrated probabilities, useful for decision-making processes.

- Logistic Regression is robust and interpretable for classification tasks.
- Its foundation in probability theory (MLE) makes it statistically sound.
- Understanding the transformation from linear predictors to probabilities (via the logistic function) is key.
- Practical considerations—feature scaling, regularization, and optimization—can significantly enhance model performance.

Table of Contents

1 An Overview of Classification

2 Why Not Linear Regression

3 Motivation and Background

4 The Logistic Function and Transformations

5 Cost Function and Optimization

6 Assumptions, Limitations, and Conclusion

7 LDA and QDA

Motivation — An Alternative Approach

- Traditional approaches (like Logistic Regression) directly model the conditional probability $\Pr(Y \mid X)$.
- LDA adopts a generative approach that models $\Pr(X \mid Y)$ and then uses Bayes' theorem:
- This formulation can be more efficient when class-conditional densities $(Pr(X|Y = k))$ are Gaussian and the sample size is small.

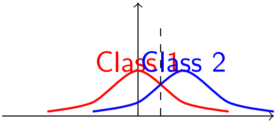


Figure: Visualization of Gaussian class densities and decision boundary

Comparison: Logistic Regression vs. LDA

Logistic Regression

- Direct probability modeling: $\Pr(Y \mid X)$
- No explicit distributional assumptions
- Tends to be more stable when there are many predictors

LDA

- Assumes Gaussian class-conditional densities
- Naturally extends to multi-class problems
- Often more efficient when sample size is small

Example: For 3 classes with 10 samples per class, the parametric form of LDA can be advantageous compared to the flexibility (and potential overfitting) of logistic regression.

Bayes' Theorem & Classifier

Bayes' Rule for Classification:

$$\Pr(Y = k | X = x) = \frac{\pi_k f_k(x)}{\sum_{l=1}^{K} \pi_l f_l(x)}$$

- π_k: Prior probability of class k $\rightarrow \Pr(Y=k)$
- $f_k(x)$: Class-conditional density of X for $Y = k$ $\rightarrow \Pr(X=x | Y=k)$
- Decision Rule: Assign x to class k maximizing $\Pr(Y = k | X = x)$

Bayes Error Rate: Minimum possible error rate if true densities $f_k(x)$ are known.

Challenge: Estimating $f_k(x)$ from data.

Assume class densities are multivariate Gaussian with **shared covariance**:

$$f_k(x) = \frac{1}{(2\pi)^{p/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu_k)^T\Sigma^{-1}(x-\mu_k)\right)$$

Key Implications:

- Linear decision boundaries
- Homoscedasticity: Same covariance structure across classes

$$\pi_k f_k(x) = \pi_k \cdot \frac{1}{\sqrt{2\pi}\sigma} \exp\left[\frac{-(x-\mu_k)^2}{2\sigma^2}\right] \rightarrow \frac{-(x^2+\mu_k^2-2x\mu_k)}{2\sigma^2}$$

$$\log \rightarrow \quad \log \pi_k - \frac{x^2}{2\sigma^2} - \frac{\mu_k^2}{2\sigma^2} + x\mu_k/\sigma^2$$

Derivation of LDA Discriminant Function

Starting from Bayes' rule, take log and simplify:

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

- Quadratic terms cancel due to shared Σ
- Decision boundary between classes k and l: $\delta_k(x) = \delta_l(x)$

For $p = 1$ **(Simplified)**:

$$\delta_k(x) = \frac{\mu_k}{\sigma^2} x - \frac{\mu_k^2}{2\sigma^2} + \log \pi_k$$

Boundary: $x = \frac{\mu_1 + \mu_2}{2}$ when $\pi_1 = \pi_2$

Graphical Illustration — 1D Case

- **Left:** Two Gaussian densities $f_1(x)$ and $f_2(x)$
- **Dashed line:** Bayes decision boundary

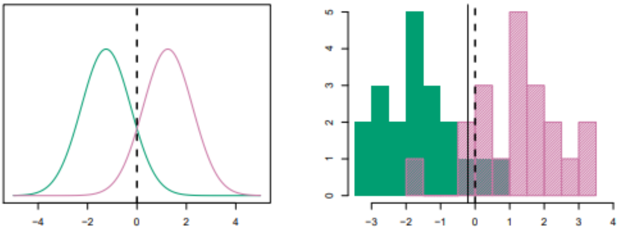


Figure: 1D Gaussian densities and decision boundary.

Parameter Estimation in LDA

Maximum Likelihood Estimators:

- Class mean:

$$\hat{\mu}_k = \frac{1}{n_k} \sum_{i:y_i=k} x_i$$

- Pooled covariance:

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

- Prior probabilities:

$$\hat{\pi}_k = \frac{n_k}{n}$$

Intuition: $\hat{\Sigma}$ is a weighted average of class-specific covariances.

Example: LDA for $p = 1$ with Calculations

Simulated Data:

- Class 1: $\mu_1 = -1.25$, Class 2: $\mu_2 = 1.25$, $\sigma^2 = 1$
- Training data: $n_1 = n_2 = 20$

Estimates:

$$\hat{\mu}_1 = -1.2,\ \hat{\mu}_2 = 1.3,\ \hat{\sigma}^2 = 0.95$$

Decision Boundary:

$$x = \frac{\hat{\mu}_1 + \hat{\mu}_2}{2} = \frac{-1.2 + 1.3}{2} = 0.05$$

- Bayes boundary: $x = 0$ (vs. LDA: $x = 0.05$)
- Error rates: Bayes (10.6%), LDA (11.1%)

Case Study: Default Data Analysis

Confusion Matrix (Threshold=0.5):

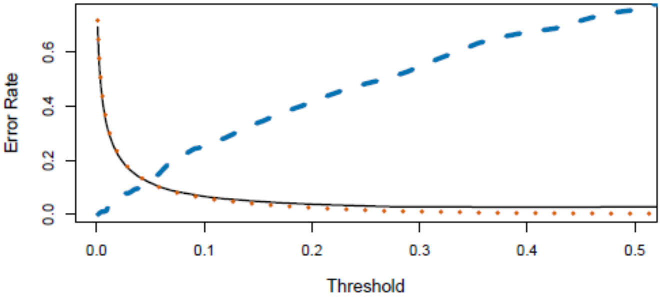
		True No	True Yes	Total
Predicted	**No**	9644 (TN)	252 (FN)	9896
	Yes	23 (FP)	81 (TP)	104
	Total	9667	333	10000

Confusion Matrix (Threshold=0.2):

		True No	True Yes	Total
Predicted	**No**	9432 (TN)	138 (FN)	9570
	Yes	235 (FP)	195 (TP)	430
	Total	9667	333	10000

Threshold Tuning

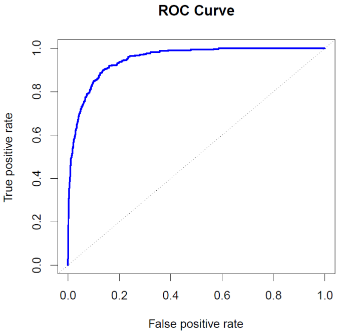
- The black solid line displays the overall error rate.
- The blue dashed line represents the fraction of defaulting customers that are incorrectly classified,
- The orange dotted line indicates the fraction of errors among the non-defaulting customers.



Threshold Tuning & ROC Curve

Adjusting Posterior Threshold:

- Lower threshold (e.g., 0.2) increases sensitivity (TP/P) but decreases specificity ($FP/N = 1-$specificity)
- Trade-off captured by ROC curve



ROC curve: AUC $= 0.95$ (Excellent discrimination)

Graphical Illustration — Multivariate LDA

- **Left:** Three-class example with 95% probability ellipses
- **Dashed lines:** Bayes decision boundaries
- **Solid lines:** LDA estimated boundaries

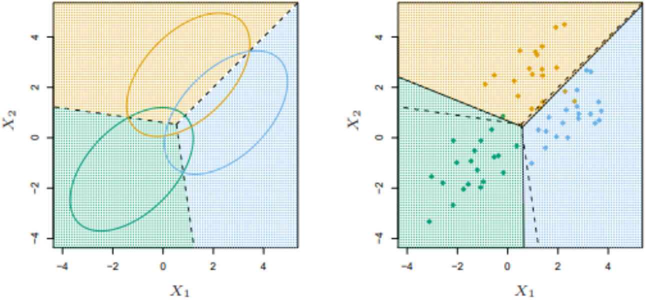


Figure: Multivariate LDA: Bayes vs. LDA boundaries.

QDA: Model Assumptions

Relax LDA's assumption: Class-specific covariances:

$$f_k(x) = \frac{1}{(2\pi)^{p/2}|\Sigma_k|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu_k)^T\Sigma_k^{-1}(x-\mu_k)\right)$$

Discriminant Function:

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

- Quadratic terms in x remain $\Rightarrow$ Quadratic boundaries

LDA vs. QDA: Mathematical Comparison

	LDA	**QDA**
Covariance	Shared (Σ)	Class-specific (Σ_k)
Discriminant	Linear in x	Quadratic in x
Parameters	$Kp + \frac{p(p+1)}{2}$	$K\left(p + \frac{p(p+1)}{2}\right)$
Bias-Variance	Low variance, high bias*	High variance, low bias*

*When assumptions are violated.

Example: For $p = 2$, $K = 2$: LDA (7 parameters), QDA (11 parameters)

When to Use LDA vs. QDA?

- **LDA preferred**:
  - Small sample size ($n < 5p$)
  - Shared covariance structure (e.g., similar class spreads)
  - High-dimensional data (regularization needed)
- **QDA preferred**:
  - Large sample size ($n > 10p$)
  - Heteroscedastic classes (unequal covariances)
  - Complex decision boundaries

Summary

- **LDA**: Efficient, linear boundaries, requires $n > p$
- **QDA**: Flexible, quadratic boundaries, needs larger n
- Model choice depends on bias-variance trade-off and data structure
- Threshold tuning critical for imbalanced class problems
- Always validate assumptions and consider alternatives

LDA and QDA

- Left: The Bayes (purple dashed), LDA (black dotted), and QDA (green solid) decision boundaries for a two-class problem. $\Sigma_1 = \Sigma_2$. The shading indicates the QDA decision rule. Since the Bayes decision boundary is linear, it is more accurately approximated by LDA than by QDA.
- Right: Details are as given in the left-hand panel, except that $\Sigma_1 \neq \Sigma_2$