Gaussian Discriminant Analysis

🧠 1️⃣ Gaussian (Normal) Random Variable

Definition:

A **Gaussian random variable** (or **Normal random variable**) is one whose probability distribution follows a **bell-shaped curve** — called the **Normal Distribution**.

Mathematically,

$$X \sim \mathcal{N}(\mu, \sigma^2)$$

means

"X is normally distributed with mean μ and variance σ^2."

📈 Probability Density Function (PDF)

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

Where:

- μ = **mean** → controls *center / location*
- σ^2 = **variance** → controls *spread / width*

🧩 Intuition

Parameter	Meaning	Effect
μ	Center	Shifts the curve left/right

Parameter	Meaning	Effect
σ^2	Spread	Wider curve = more uncertainty

🪶 Properties

1. **Symmetrical** about the mean μ
2. **Mean = Median = Mode =** μ
3. Fully defined by just **two parameters:** μ**,** σ^2
4. The **68–95–99.7 rule (Empirical rule)**:
    - 68% of values within 1σ
    - 95% within 2σ
    - 99.7% within 3σ

🧮 2️⃣ Multivariate Gaussian Random Variable

Now we extend this concept to **multiple dimensions** — i.e., several variables that may be correlated.

Let's say we have a random vector:

$$X = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_d \end{bmatrix}$$

Then X follows a **multivariate normal distribution** if every *linear combination* of its components is normally distributed.

$$X \sim \mathcal{N}(\mu, \Sigma)$$

📊 PDF Formula

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

Where:

- $x \in \mathbb{R}^d$: a d-dimensional data vector
- $\mu \in \mathbb{R}^d$: mean vector
- $\Sigma \in \mathbb{R}^{d \times d}$: **covariance matrix**

🧩 Intuition for Parameters

Parameter	Meaning
Mean vector μ	Center (expected value) — where the distribution is centered
Covariance matrix Σ	Shape, orientation, and spread of the distribution

---

Covariance Matrix Example

$$\Sigma = \begin{bmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{bmatrix}$$

- σ_i^2 : variance of each variable
- ρ : correlation coefficient between X_1 and X_2

What is ρ? It's defined as:

$\rho = \frac{\text{Cov}(X_1, X_2)}{\sigma_1\sigma_2}$

📘 Where:

- $\text{Cov}(X_1, X_2) = E[(X_1 - \mu_1)(X_2 - \mu_2)]$
- $\sigma_1 = \sqrt{\text{Var}(X_1)}$
- $\sigma_2 = \sqrt{\text{Var}(X_2)}$

- Measures **linear relationship** strength and direction.
- Determines the **shape and orientation** of the multivariate Gaussian's contours.

📉 Visual Intuition (2D Example)

Imagine $X = X_1, X_2^T$:

- If X_1 and X_2 are **uncorrelated** → covariance = 0 → **circular** contour.
- If they are **correlated** → covariance ≠ 0 → **elliptical** contour (tilted ellipse).

The ellipse shows regions of **equal probability density**.

The orientation and length of its axes depend on Σ.

🧮 Mahalanobis Distance

Inside the exponential term:

$(x-\mu)^T\Sigma^{-1}(x-\mu)$

This is known as the **Mahalanobis distance**, a generalized measure of "distance" that takes into account correlations among variables.

It replaces the simple Euclidean distance used in the univariate case.

📚 3️⃣ Summary Table

Concept	Univariate Gaussian	Multivariate Gaussian
Variable	Scalar x	Vector $x \in \mathbb{R}^d$
Parameters	μ, σ^2	μ, Σ
Shape	Bell curve	Elliptical contours
PDF	$\frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{(x-\mu)^2}{2\sigma^2}}$	$\frac{1}{(2\pi)^{d/2}\lvert\Sigma\rvert^{1/2}}\exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right)$
Covariance	Scalar variance	Covariance matrix
Independence	Not applicable	If Σ is diagonal → independent dimensions

💡 Key Intuition

> The multivariate Gaussian is a natural generalization of the bell curve to multiple dimensions — where the covariance matrix controls the shape and orientation of the probability "ellipsoid."

Discriminative learning algorithms

Definition:

Discriminative learning algorithms are models that **learn the boundary** (or function) that **directly maps inputs x** to **outputs y — i.e., they learn P(y | x)**.

So instead of modeling how the data was *generated* (as generative models do), discriminative models focus purely on **distinguishing** between classes or predicting outcomes.

The Formula in the Image

In the image, the model is defined as:

$$P(y|\bar{x}, \bar{\theta})$$

This is exactly the **conditional probability** that discriminative models aim to learn.

Examples in the Image

- **Linear Regression:**

  $$y = \bar{\theta}^T \bar{x} + \xi$$

  Predicts a continuous output y.

- **Logistic Regression:**

  $$y = \frac{1}{1+e^{-\bar{\theta}^T \bar{x}}}$$

  Predicts a probability (between 0 and 1), often used for classification.

The Learning Objective

Discriminative models find the **best parameters** $\bar{\theta}^*$ that maximize the **likelihood** of the observed labels given the inputs:

$$bar\theta^* = \arg\max_{\bar{\theta}} \mathcal{L}(\bar{\theta})$$

Here, $\mathcal{L}(\bar{\theta})$ is the likelihood function:

$$\mathcal{L}(\bar{\theta}) = P(y|x, \bar{\theta})$$

💬 Intuitive Summary
A **discriminative model** just wants to **draw the line —** "where should I draw the boundary between cats and dogs?"

Generative learning algorithms

Definition

Generative learning algorithms try to **model how the data is generated —** that is, they learn the **joint probability distribution**:

$$P(x, y)$$

From this joint distribution, they can use **Bayes' Theorem** to predict the label y given a new input x:

$$P(y|x) = \frac{P(x|y)\,P(y)}{P(x)}$$

Think of a **generative model** as a model that **tries to simulate reality**.

It asks:

> "If I know the class y, what kind of x values do I expect to see?"

In other words:

- It first learns how **each class generates its data** (through P(x ∣ y))
- Then, combine this with how likely each class is overall (P(y))

Algorithm	Key Idea
Naïve Bayes	Assumes features are conditionally independent given the class. Fast and simple.

Algorithm	Key Idea
Gaussian Discriminant Analysis (GDA)	Assumes each class's data follows a Gaussian distribution with parameters (μ_y, Σ_y).
Hidden Markov Models (HMM)	Models sequences — how observations evolve over time.
Variational Autoencoders (VAE)	Deep generative models that learn data distributions and can generate new samples.
Generative Adversarial Networks (GANs)	Use two neural networks (Generator + Discriminator) to produce realistic synthetic data.

Gaussian Discriminant Analysis

🔷 Definition

Gaussian Discriminant Analysis (GDA) is a **generative learning algorithm** that assumes:

> Each class generates data points according to a multivariate Gaussian (Normal) distribution.

It models how the data in each class is distributed in feature space.

The Big Picture

We want to predict $y \in \{0, 1\}$ given features $x \in \mathbb{R}^n$.

Instead of directly modeling P(y ∣ x) (like Logistic Regression),

GDA models the **joint distribution** P(x,y)=P(x ∣ y)P(y).

Then, it uses **Bayes' theorem** to find P(y ∣ x):

$$P(y|x) = \frac{P(x|y)P(y)}{P(x)}$$

⚙️ The Model Assumptions

1. The prior over classes is Bernoulli:

$y \sim \text{Bernoulli}(\phi)$

where $\phi = P(y = 1)$.

2. The class-conditional distribution of features is Gaussian:

   $x|y = 0 \sim \mathcal{N}(\mu_0, \Sigma)$

   $x|y = 1 \sim \mathcal{N}(\mu_1, \Sigma)$

   Here:

   - μ_0, μ_1 are the **mean vectors** for each class.
   - Σ is the **shared covariance matrix** across both classes.

     The shape and spread of the class's "blob" in feature space tells us **how stretched, tilted, or round** that class's data cloud is.

     In **Gaussian Discriminant Analysis (GDA)**, we assume:

     $\Sigma_0 = \Sigma_1 = \Sigma$

     i.e. both classes share the **same covariance matrix**.

     That means both classes have:

     - The same "shape" and "spread"
     - But *different centers* (means)

     Visually: 🍎🍊

     Two equally shaped blobs are placed at different locations.

     ❓ Why We Share Covariance Matrices?

     1. Simpler and More Stable Model:

        If we allow each class to have its own covariance Σ_i:

        - We'd have to estimate two full matrices from data.
        - For high-dimensional data, that's **a lot of parameters**.

        By assuming both classes share the same covariance:

        ✅ Fewer parameters →

        ✅ More robust estimates →

✅ Less risk of overfitting.

2. Linear Decision Boundary:

   When the covariance matrices are **equal**, the log-likelihood ratio between the two Gaussians becomes **linear in x**.

   This gives you a **linear decision boundary**, i.e. a straight line (or hyperplane):

   $\log \frac{P(y=1|x)}{P(y=0|x)} = \theta^T x + \theta_0$

   ➡️ So **GDA with shared covariance** behaves like **Logistic Regression**,

   but from a *generative* viewpoint. If we let each class have its own covariance:

   $\Sigma_0 \neq \Sigma_1$

   Then the decision boundary becomes **quadratic** (curved) — that's known as **Quadratic Discriminant Analysis (QDA)**.

3. Real-world Intuition:

   Sometimes it's reasonable to assume:

   > "All classes have roughly the same variability, just centered at different means."

In one line:

> GDA shares the covariance matrix to simplify learning and produce a linear decision boundary — just like Logistic Regression, but derived from a probabilistic model of how data is generated.

🧮 Step 1: Estimate the Parameters

We learn parameters $\phi, \mu_0, \mu_1, \Sigma$ from the training data:

$$\phi = \frac{1}{m} \sum_{i=1}^{m} 1\{y^{(i)} = 1\}$$

$$\mu_0 = \frac{\sum_{i:y^{(i)}=0} x^{(i)}}{\sum_{i:y^{(i)}=0} 1}$$

$$\mu_1 = \frac{\sum_{i:y^{(i)}=1} x^{(i)}}{\sum_{i:y^{(i)}=1} 1}$$

$$\Sigma = \frac{1}{m} \sum_{i=1}^{m} (x^{(i)} - \mu_{y^{(i)}})(x^{(i)} - \mu_{y^{(i)}})^T$$

🧮 Step 2: Use Bayes' Rule to Classify

We compute:

$$P(y=1|x) = \frac{P(x|y=1)P(y=1)}{P(x|y=1)P(y=1) + P(x|y=0)P(y=0)}$$

and predict:

$$\hat{y} = \begin{cases} 1 & \text{if } P(y=1|x) \geq 0.5 \\ 0 & \text{otherwise} \end{cases}$$

🧭 Step 3: Connection to Logistic Regression

When you take the **log odds ratio** of P(y=1 | x), it turns out to be a **linear function of x**:

$$\log \frac{P(y=1|x)}{P(y=0|x)} = \theta^T x + \theta_0$$

That's the same form as **Logistic Regression**!

So, logistic regression can be viewed as the **discriminative counterpart** of GDA.

The Intuition Behind Gaussian Discriminant Analysis (GDA)

Imagine you're a detective trying to identify which **group (class)** a data point belongs to — but instead of directly drawing a line between the groups (like Logistic Regression does), you try to **understand how each group *produces* its data**.

That's the essence of **GDA**:

it tries to **model how each class generates its data points**, assuming they look like "bell-shaped blobs" (Gaussians) in feature space.
