# Introduction to Generative Models

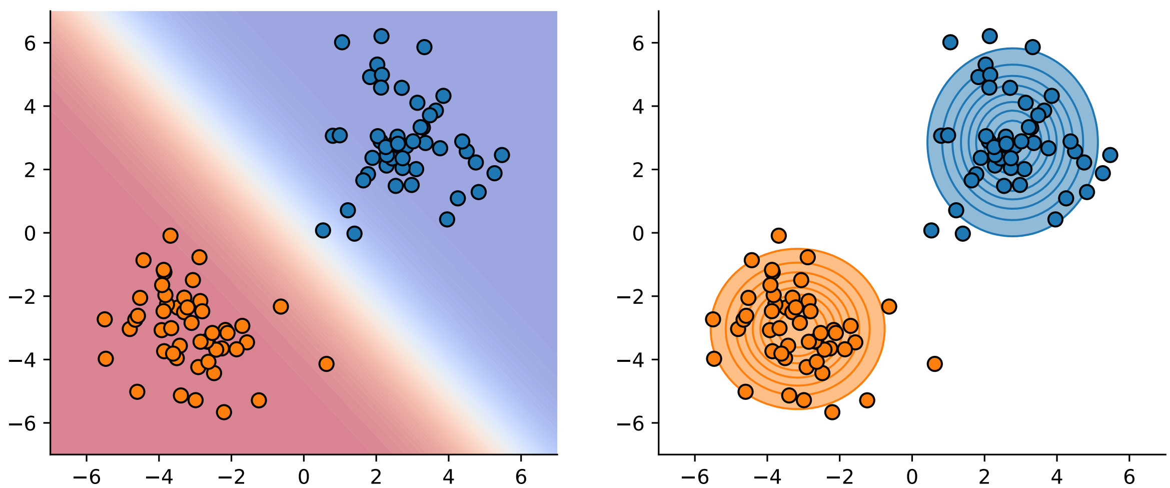
**Advanced Statistical Inference**

Simone Rossi

## Discriminative vs generative models

Discriminative models learn the **conditional distribution** of the labels given the data of the form $p(y \mid x)$.

Generative models learn the **joint distribution** of the data and the labels of the form $p(y, x)$.




## Why generative models?

So far, we have focused on **discriminative models**:

1. Given the data, we want to predict the label, which can be continuous (regression) or discrete (classification).
2. We are modeling the likelihood as conditional on the data $p(y \mid x)$.

But they cannot:

1. Tell you how the probability of seeing a certain data point.
2. Generate new data points.

Generative models can do both!

### Generative doesn't mean unsupervised

While generative models are often used in unsupervised learning (e.g. images without labels, text without labels, etc.), they can also be used in supervised learning.

Before we look at more classical generative models, let's look at a simple example of a generative model that is also supervised: **Naive Bayes classification**.

## Classification as a generative model

### Naive Bayes

- Naive Bayes is an example of **generative classifier**.
- It is based on the **Bayes theorem** and assumes that the features are **conditionally independent** given the class.

$$p(y_\star = k|\boldsymbol{x}_\star, \boldsymbol{X}, \boldsymbol{y}) = \frac{p(\boldsymbol{x}_\star|y_\star = k, \boldsymbol{X}, \boldsymbol{y})p(y_\star = k)}{\sum_j p(\boldsymbol{x}_\star|y_\star = j, \boldsymbol{X}, \boldsymbol{y})p(y_\star = j)}$$

**Questions**:

1. What is the likelihood $p(\boldsymbol{x}_\star|y_\star = k, \boldsymbol{X}, \boldsymbol{y})$?
2. What is the prior $p(y_\star = k)$?

### Naive Bayes: Likelihood

$$p(\boldsymbol{x}_\star|y_\star = k, \boldsymbol{X}, \boldsymbol{y}) = p(\boldsymbol{x}_\star|y_\star = k, \boldsymbol{f}(\boldsymbol{X}, \boldsymbol{y}))$$

- How likely is the new data $\boldsymbol{x}_\star$ given the class $k$ and the training data $\boldsymbol{X}, \boldsymbol{y}$?
- The function $\boldsymbol{f}(\boldsymbol{X}, \boldsymbol{y})$ encodes some likelihood parameters, given the training data.
- Free to choose the form of the likelihood, but it needs to be *class-conditional*.
- Common choice: **Gaussian** distribution (for each class).
- Training data for class $k$ used to estimate the parameters of the Gaussian (mean and variance).

### Naive Bayes: Likelihood

Naive Bayes makes an additional assumption on the likelihood:

- The features are **conditionally independent** given the class.

$$p(\boldsymbol{x}_\star|y_\star = k, \boldsymbol{f}(\boldsymbol{X}, \boldsymbol{y})) = \prod_{d=1}^{D} p(x_{\star d}|y_\star = k, \boldsymbol{f}_d(\boldsymbol{X}, \boldsymbol{y}))$$

- The likelihood parameters $\boldsymbol{f}_d(\boldsymbol{X}, \boldsymbol{y})$ are specific to each feature $d$.
- In the Gaussian case:
  - $\boldsymbol{f}_d(\boldsymbol{X}, \boldsymbol{y}) = \{\mu_{kd}, \sigma^2_{kd}\}$.
  - $p(x_{\star d}|y_\star = k, \boldsymbol{f}_d(\boldsymbol{X}, \boldsymbol{y})) = \mathcal{N}(x_{\star d}|\mu_{kd}, \sigma^2_{kd})$.

### Naive Bayes: Prior

$$p(y_\star = k)$$

- How likely is the class $k$?

Examples:

- Uniform prior: $p(y_\star = k) = \frac{1}{K}$.
- Class imbalance: $p(y_\star = k) \ll p(y_\star = j)$ for $j \neq k$ if class $k$ is rare.

### Training Naive Bayes classifier

**Step 1**: fit the class-conditional distributions using the training data. $\boldsymbol{f}(\boldsymbol{X}, \boldsymbol{y})$ needs to estimate mean and variance for each class.

1. **Empirical mean**:

$$\mu_{kd} = \frac{1}{N_k} \sum_{i=1}^{N} x_{id} \mathbb{I}(y_i = k)$$

2. **Empirical variance**:

$$\sigma^2_{kd} = \frac{1}{N_k} \sum_{i=1}^{N} (x_{id} - \mu_{kd})^2 \mathbb{I}(y_i = k)$$

**Step 2**: predict the class of new data using Bayes theorem.

$$p(y_\star = k|\boldsymbol{x}_\star, \boldsymbol{X}, \boldsymbol{y}) = \frac{p(\boldsymbol{x}_\star|y_\star = k, \boldsymbol{X}, \boldsymbol{y})p(y_\star = k)}{\sum_j p(\boldsymbol{x}_\star|y_\star = j, \boldsymbol{X}, \boldsymbol{y})p(y_\star = j)}$$

### Summary

Naive Bayes is a generative classifier based on the Bayes theorem.

**Advantages**:

- Simple and fast.
- Works well with high-dimensional data.

**Disadvantages**:

- Strong assumption of conditional independence.
- Hides the classification problem as a density estimation problem.

### Generative models

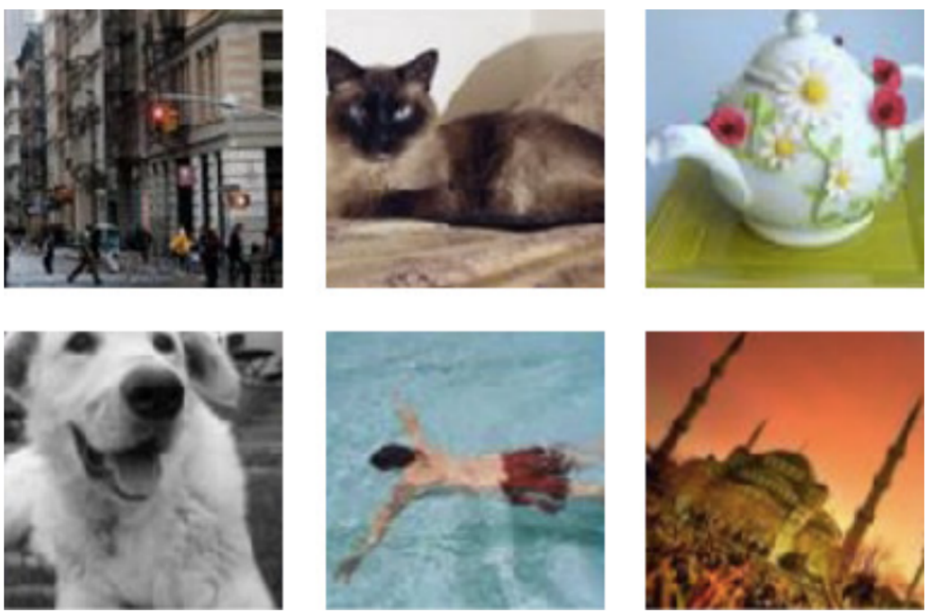
**Generative models**:

1. Model the distribution of the data $p_{\text{data}}(\boldsymbol{x})$.
2. Can generate new data points by sampling from the learnt distribution.
3. Evaluate the likelihood of the data $p_{\text{model}}(\boldsymbol{x})$.
4. Find conditional relationships between variables, e.g., $p(\boldsymbol{x}_1 \mid \boldsymbol{x}_2)$.
5. Can be used for computing complexity measures, e.g., entropy, mutual information.

### Generative models

Given training data from an unknown distribution $p_{\text{data}}(\boldsymbol{x})$, we want to learn a model $p_{\text{model}}(\boldsymbol{x})$ that approximates the true distribution.

Train from $x \sim p_{\text{data}}(x)$.



Generate from $x \sim p_{\text{model}}(x)$.

## Classes of generative models

Classic generative models:

- **Linear Latent Variable models**

**Deep** generative models:

- **Variational Autoencoders**
- **Generative Adversarial Networks**
- Autoregressive models
- Diffusion Models
- Normalizing Flows
- Flow Matching Models