Machine Learning and Artificial Intelligence

Lab 02 – Bayes decision theory

and discriminant functions

15/03/2021

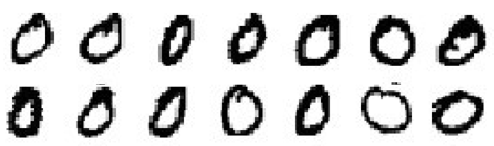
Recap

In **supervised** machine learning, we address problems by **learning from experience**.

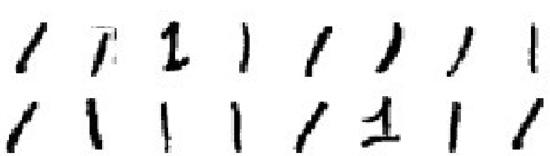
Each object in the training set is:

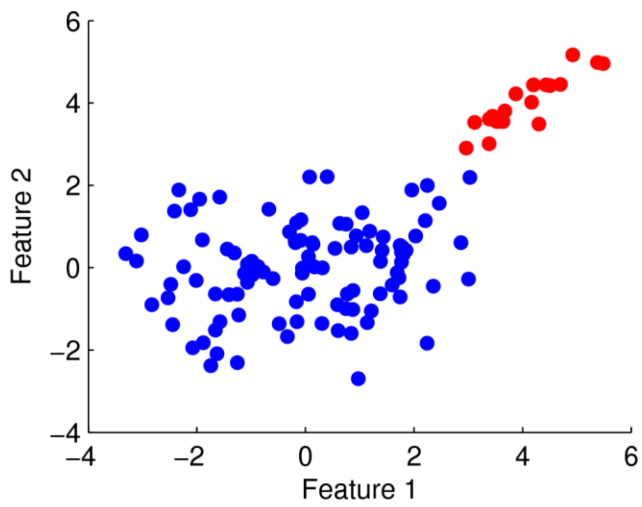
- Represented by a set of features
- Labeled

«0»



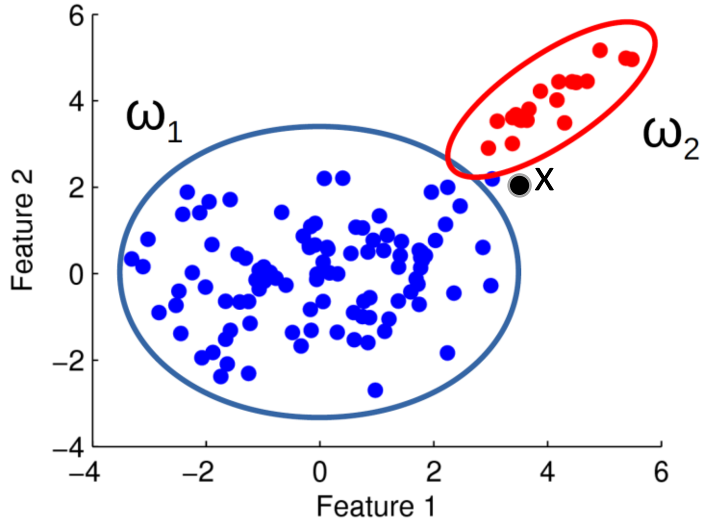
«1»





Bayes’ decision rule

- Given an input x, do I assign it to class ω_1 or ω_2 ?



Assign to $\boldsymbol{\omega_1}$ if $\boldsymbol{P(\omega_1 \mid x) > P(\omega_2 \mid x)}$, $\boldsymbol{\omega_2}$ otherwise

Bayes’ decision rule

- **Bayes’ formula**

Likelihood **Prior**

$$P(\omega_j \mid x) = \frac{P(x \mid \omega_j)\, P(\omega_j)}{P(x)} = P(x \mid \omega_j)\, P(\omega_j)$$

Posterior **Evidence**

- Steps:
  - **Inference** → calculate $P(\omega_j \mid x)$
  - **Decision** → using the decision rule

Discriminant Functions

- Alternative: Inference and Decision are solved simultaneously by training a function that maps x to the decision space.

Discriminant functions

$$\boldsymbol{g_i(x), i = 1, \ldots, c}$$

- The classifier assigns the feature vector x to the class $\boldsymbol{\omega_i}$ if:

$$g_i(x) > g_j(x) \qquad \forall\, j \neq i$$

Discriminant Functions

- Goal: get a form of $\boldsymbol{g_i(x)}$ that is easy to understand and *calculate*.

$$\boldsymbol{g_i(x) = P(x \mid \omega_i)\, P(\omega_i)}$$

$$\Downarrow$$

$$\underline{\boldsymbol{g_i(x) = \ln P(x \mid \omega_i) + \ln P(\omega_i)}}$$

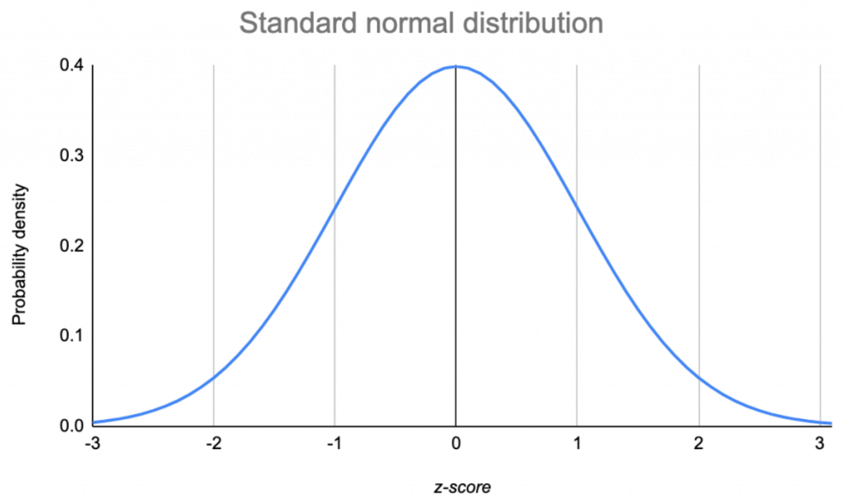
- How can we model probability?

The Normal Distribution

Univariate Gaussian

Parameters:

- Mean μ
- Standard deviation σ



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

```
>> from scipy.stats import norm
>> norm.pdf(x, mu, sigma)
```

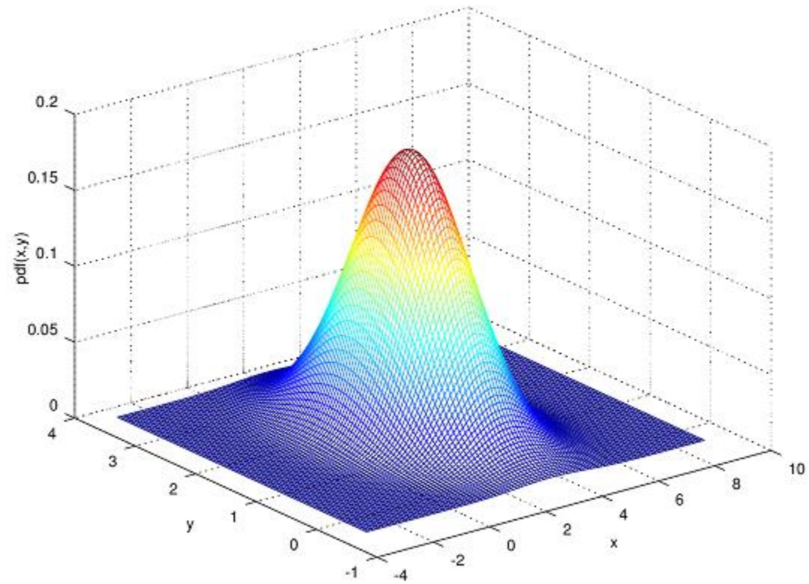
https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.norm.html

The Normal Distribution

Multivariate Gaussian

Parameters:

- d number of features
- Vector of averages μ
- Covariance matrix Σ



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

```
>> from scipy.stats import multivariate_normal
>> multivariate_normal.pdf(x, mu, sigma)
```

https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.multivariate_normal.html

Discriminant Functions - Normal Distribution

- Translation of the discriminant function in the case of normal probability density:

$$\boldsymbol{g_i(x) = \ln P(x \mid \omega_i) + \ln P(\omega_i)}$$

$$\boldsymbol{g_i(x) = -\frac{1}{2}(x - \mu_i)^t \, \Sigma_i^{-1}(x - \mu_i) - \frac{d}{2}\ln 2\pi - \frac{1}{2}\ln |\Sigma_i| + \ln P(\omega_i)}$$

- Depending on $\boldsymbol{\Sigma}$, the formula can be simplified.

Variations according to $\boldsymbol{\Sigma}$

- $\boldsymbol{\Sigma}_{\mathrm{i}} = \sigma^2 I$ → statistically independent features

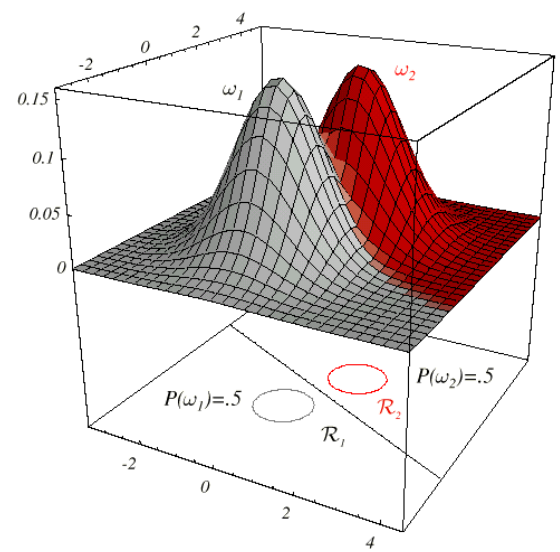
$$g_{i(x)} = w_i^t x + w_{i0} \quad , \qquad w_i = \frac{1}{\sigma^2}\,\mu_i \quad , \qquad w_{i0} = -\frac{1}{2\sigma^2}\mu_i^t \mu_i + \ln \mathrm{P}(\omega_i)$$

The discriminant function defines a hyperplane that passes through x_0 and orthogonal to w:

$$w^t(x - x_0) = 0$$

$$w = \mu_i - \mu_j$$

$$x_0 = \frac{1}{2}(\mu_i + \mu_j) - \frac{\sigma^2}{||\mu_i - \mu_j||^2} \ln \frac{P(\omega_i)}{P(\omega_j)} (\mu_i - \mu_j)$$



Variations according to $\boldsymbol{\Sigma}$

- $\boldsymbol{\Sigma_\mathrm{i}} = \sigma^2\Sigma$ with Σ equal for all classes

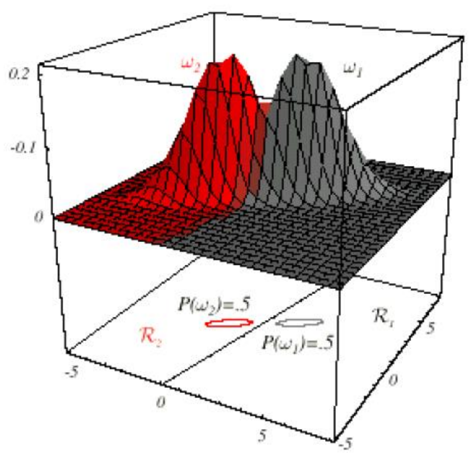
$$g_{i(x)} = w_i^t x + w_{i0} \quad , \quad w_i = \Sigma^{-1}\mu_i \quad , \quad w_{i0} = -\frac{1}{2}\mu_i^t\Sigma^{-1}\mu_i + \ln \mathrm{P}(\omega_i)$$

The discriminant function defines a hyperplane that passes through x_0 and orthogonal to w:

$$w^t(x - x_0) = 0$$

$$w = \Sigma^{-1}(\mu_i - \mu_j)$$

$$x_0 = \frac{1}{2}(\mu_i + \mu_j) - \frac{\ln\left(\frac{P(\omega_i)}{P(\omega_j)}\right)}{(\mu_i - \mu_j)^t\Sigma^{-1}(\mu_i - \mu_j)}(\mu_i - \mu_j)$$



GAUSSIAN NAIVE BAYES CLASSIFIER

"Gaussian" because this is a normal distribution

This is our prior belief

$$p(\text{class} \mid \text{data}) = \frac{p(\text{data} \mid \text{class}) \times p(\text{class})}{p(\text{data})}$$

We don't calculate this in naive bayes classifiers

Exercises