# Machine learning

## Probabilistic learning

Corso di Laurea Magistrale in Informatica

Università di Roma Tor Vergata

Giorgio Gambosi

a.a. 2024-2025

# Supervised learning framework: deriving a probabilistic predictor

As done before, we assume that the observed dataset (features and target) has been derived by randomly sampling:

- $\mathcal{X}$ according to the probability distribution $p_M(x)$ (usually the uniform distribution)
- $\mathcal{Y}$ according to the conditional distribution $p_C(t|x)$

Deriving a probabilistic predictor results into deriving, from the training set $\mathcal{T}$, an algorithm computing a conditional distribution $\hat{p}(t|\mathbf{x})$ which approximates the correct, unknown distribution $p_C$.

An independent decision strategy must be applied to $p^*(t|\mathbf{x})$ to return a specific prediction $h(\mathbf{x})$

# Decision theory

- Selecting the best actions to perform, given a cost associated to each action, in order to minimize the expected cost.
- Two phases
  - Inference: determine $p(\mathbf{x}, y)$ (in the case of classification $p(\mathbf{x}, C_0)$, $p(\mathbf{x}, C_1)$ or $p(\mathbf{x}|C_0)$, $p(\mathbf{x}|C_1)$), to provide a probabilistic description of data.
  - Decision: given a new value $\mathbf{x}$, predict $\mathbf{t}$ given $\mathbf{x}$, and determine the action to perform
- Input data $\mathbf{x}$, target variables $\mathbf{t}$. We wish to predict $\mathbf{t}$ for any new value of $\mathbf{x}$
  - Regression: $\mathbf{t}$ variables are in $\mathbb{R}$
  - Classification: $\mathbf{t}$ variables are class labels
- Important quantity: $p(C_k|\mathbf{x})$

$$p(C_k|\mathbf{x}) = \frac{p(\mathbf{x}, C_k)}{p(\mathbf{x})} = \frac{p(\mathbf{x}, C_k)}{\sum_i p(\mathbf{x}, C_i)}$$

# Decision regions

- Partition of the input space into decision regions $\mathcal{R}_k$, separated by decision surfaces
- All points in a same $\mathcal{R}_k$ are assigned to a $\mathcal{C}_k$
- Binary case: a classification error occurs with probability

$$p_e = p(\mathbf{x} \in \mathcal{R}_1, \mathcal{C}_2) + p(\mathbf{x} \in \mathcal{R}_2, \mathcal{C}_1) = \int_{\mathcal{R}_1} p(\mathbf{x}, \mathcal{C}_2) d\mathbf{x} + \int_{\mathcal{R}_2} p(\mathbf{x}, \mathcal{C}_1) d\mathbf{x}$$

# Minimization of classification error

- $p_e$ is minimized if $\mathbf{x}$ is assigned to the class $\mathcal{C}_k$ with maximum probability $p(\mathbf{x}, \mathcal{C}_k)$
- Since $p(\mathbf{x}, \mathcal{C}_k) = p(\mathcal{C}_k|\mathbf{x})p(\mathbf{x})$, $p_e$ is minimum if $\mathbf{x}$ is assigned to the class $\mathcal{C}_k$ with maximum probability $p(\mathcal{C}_k|\mathbf{x})$

Extendable to multiclass classification. Probability of correct classification:

$$p_c = \sum_{i=1}^{K} p(\mathbf{x} \in \mathcal{R}_i, C_i) = \sum_{i=1}^{K} \int_{\mathcal{R}_i} p(\mathbf{x}, C_i) d\mathbf{x}$$

# Misclassification cost

## Classification cost function

- $L_{kj}$ is a cost associated to classify $\mathbf{x} \in \mathcal{C}_k$ in $\mathcal{C}_j$
- Expected misclassification cost

$$E[L] = \sum_k \sum_j \int_{\mathcal{R}_j} L_{kj} p(\mathbf{x}, \mathcal{C}_k) d\mathbf{x} = \sum_k \sum_j \int_{\mathcal{R}_j} L_{kj} p(\mathcal{C}_k | \mathbf{x}) p(\mathbf{x}) d\mathbf{x}$$

## Minimization of expected misclassification cost

- Choose $\mathcal{R}_j$ to minimize $E[L]$
- Assign each $\mathbf{x}$ to a class $\mathcal{C}_i$ such that

$$i = \operatorname*{argmin}_j \sum_k L_{kj} p(\mathbf{x}, \mathcal{C}_k)$$

- Or, to a class $\mathcal{C}_i$ such that

$$i = \operatorname*{argmin}_j \sum_k L_{kj} p(\mathcal{C}_k | \mathbf{x})$$

# Misclassification cost in the binary case

- By applying the considerations above, $\mathcal{R}_1$ corresponds to cost value

  $$L_{11}p(C_1|\mathbf{x}) + L_{12}p(C_2|\mathbf{x})$$

  while $\mathcal{R}_2$ to

  $$L_{21}p(C_1|\mathbf{x}) + L_{22}p(C_2|\mathbf{x})$$

- $\mathbf{x}$ is assigned to $\mathcal{R}_1$ if

  $$L_{11}p(C_1|\mathbf{x}) + L_{12}p(C_2|\mathbf{x}) < L_{21}p(C_1|\mathbf{x}) + L_{22}p(C_2|\mathbf{x})$$

  that is, if

  $$(L_{21} - L_{11})p(C_1|\mathbf{x}) > (L_{12} - L_{22})p(C_2|\mathbf{x})$$

  or if

  $$\frac{p(C_1|\mathbf{x})}{p(C_2|\mathbf{x})} > \frac{L_{12} - L_{22}}{L_{21} - L_{11}} \frac{p(C_2)}{p(C_1)}$$

# Example

Typical example: medical diagnosis

- $C_k = \{1, 2\}$ (sick, healthy)
- $L = \left[\begin{array}{cc} 0 & 100 \\ 1 & 0 \end{array}\right]$: strong cost of not realizing of a sick patient

Expected loss

$$\begin{aligned} E[L] &= \int_{\mathcal{R}_2} L_{1,2}p(x, C_1)dx + \int_{\mathcal{R}_1} L_{2,1}p(x, C_2)dx \\ &= \int_{\mathcal{R}_2} 100 \cdot p(x, C_1)dx + \int_{\mathcal{R}_1} p(x, C_2)dx \end{aligned}$$

Assignment rule to $\mathcal{R}_1$ is

$$p(C_1|\mathbf{x}) > \frac{L_{10}}{L_{01}}p(C_0|\mathbf{x}) = 100p(C_0|\mathbf{x}) = 100(1 - p(\mathcal{C}_1|\mathbf{x}))$$

# Classification rejection

- The use of a threshold makes it possible to define situations when elements are not classified.
- For example, in the binary case, if ratio of the expected cost is around 1 (for example $[1-\varepsilon, 1+\varepsilon]$) the prediction is not returned.
- Then, $\mathbf{x}$ is in $\mathcal{R}_0$ if

$$L_{00}p(C_0|\mathbf{x}) + L_{01}p(C_1|\mathbf{x}) < (1-\varepsilon)(L_{10}p(C_0|\mathbf{x}) + L_{11}p(C_1|\mathbf{x}))$$

that is, if $p(\mathcal{C}_0|\mathbf{x}) > \theta'$, where

$$\theta' = \frac{L_{01} - L_{11} + \varepsilon L_{11}}{L_{10} + L_{01} - L_{00} - L_{11} + \varepsilon(L_{11} - L_{10})} \qquad (1)$$

# Classification rejection

Similarly, $\mathbf{x}$ is in $\mathcal{R}_1$ if

$$L_{10}p(C_0|\mathbf{x}) + L_{11}p(C_1|\mathbf{x}) < (1-\varepsilon)(L_{00}p(C_0|\mathbf{x}) + L_{01}p(C_1|\mathbf{x})) \tag{2}$$

which leads to $p(\mathcal{C}_1|\mathbf{x}) > \theta''$, where

$$\theta'' = \frac{L_{10} - (1-\varepsilon)L_{00}}{(1-\varepsilon)L_{01} + L_{10} - L_{11} - (1-\varepsilon)L_{00}} \tag{3}$$

which corresponds to

$$p(\mathcal{C}_0|\mathbf{x}) < 1 - \theta'' = \frac{L_{01} - L_{11} - \varepsilon L_{01}}{L_{10} + L_{01} - L_{00} - L_{11} - \varepsilon(L_{01} - L_{00})} \tag{4}$$

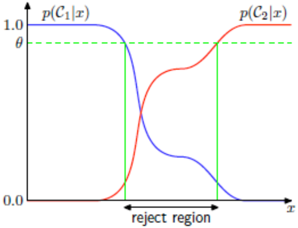
# Classification rejection

In summary, we have that:

$$h(\mathbf{x}) = \begin{cases} 1 & \text{if } p(C_0|\mathbf{x}) < 1 - \theta'' \\ 0 & \text{if } p(C_0|\mathbf{x}) > \theta' \\ \text{undefined} & \text{otherwise} \end{cases} \tag{5}$$

As a simpler approach,

- Predefined threshold $\theta$ $(0 < \theta < 1)$
- If $\max\limits_k p(\mathcal{C}_k|\mathbf{x}) < \theta$, a classification is not performed



# Decision and regression

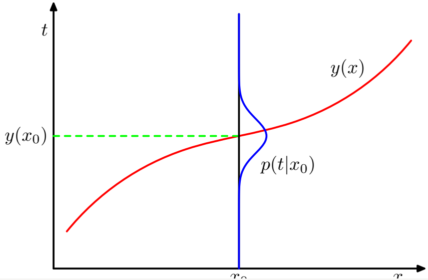
- In the case of regression, target is a numeric value, $t \in \mathbb{R}$
- Typical loss function for regression: squared difference $L(t, y(x)) = (y(x) - t)^2$

## Decision

Minimization of the expected loss w.r.t. $y(x)$ (functional minimization)

$$E[L] = \int \int (y(x) - t)^2 dxdt$$

minimized by the regression function $y(x) = \int tp(t|x)dt = E[t|x]$



# Decision and regression

To show that the regression function minimizes the loss, observe that

$$\begin{aligned}(y(x)-t)^2 &= (y(x)-E[t|x]+E[t|x]-t)^2\\ &= (y(x)-E[t|x])^2+(E[x|t]-t)^2+2\left((y(x)-E[t|x])(E[t|x]-t)\right)\end{aligned}$$

Then,

$$E[L]=\int (y(x)-E[t|x])^2p(x)dx+\int (E[t|x]-t)^2p(x)dx$$

which is minimized w.r.t. $y(x)$ by setting the first term to 0, that is when $y(x)=E[t|x]$

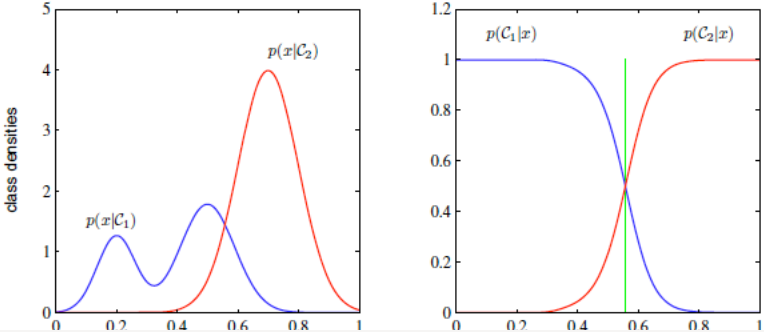
# Approximating $p_C(\mathbf{x}, t)$

we need to infer for $\mathcal{T}$ a distribution $p(t|\mathbf{x})$ which is a good approximation of $p_C$.
Two different approaches can be applied here:

1. Generative probabilistic models. Inference of conditional probabilities $p(\mathbf{x}|\mathcal{C}_k)$ for all classes. Inference of prior probabilities $p(\mathcal{C}_k)$. Use of Bayes' rule

$$p(\mathcal{C}_k|\mathbf{x}) = \frac{p(\mathbf{x}|\mathcal{C}_k)p(\mathcal{C}_k)}{p(\mathbf{x})} \approx p(\mathbf{x}|\mathcal{C}_k)p(\mathcal{C}_k)$$

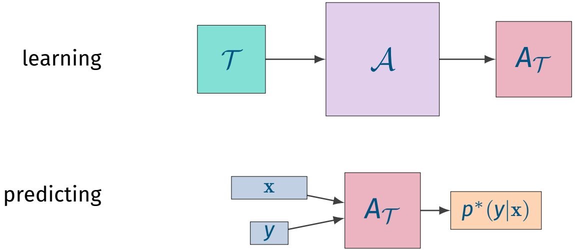
   to derive (at least up to a multiplicative constant) the posterior probabilities $p(\mathcal{C}_k|\mathbf{x})$
2. Discriminative probabilistic models. Inference of class probabilities $p(\mathcal{C}_k|\mathbf{x})$ directly from $\mathcal{T}$



# Supervised learning framework: deriving a probabilistic predictor

First approach

1. we may then consider a class of possible conditional distributions $\mathcal{P}$ and
2. select (infer) the "best" conditional distribution $p^* \in \mathcal{P}$ from the available knowledge (that is, the dataset), according to some measure $q$
3. given any new item $x$, apply $p^*(t|\mathbf{x})$ to assign probabilities for each possible value of the corresponding target



# Inferring a best distribution

How to define the class of possible conditional distributions $p(t|\mathbf{x})$?

- usually, parametric approach: distributions defined by a common (arbitrary) structure and a set of parameters

Example: logistic regression for binary classification

The probability $p(t|\mathbf{x})$, where $t \in \{0, 1\}$, is assumed to be a Bernoulli distribution

$$p(t|\mathbf{x}) = \pi(\mathbf{x})^t (1 - \pi(\mathbf{x}))^{1-t}$$

with

$$\pi(\mathbf{x}) = p(t = 1|\mathbf{x}) = \frac{1}{1 + e^{-\sum_{i=1}^{d} w_i x_i + w_0}}$$

# Inferring a best distribution

What is a measure $q(p, \mathcal{T})$ of the quality of the distribution (given the dataset $\mathcal{T} = (\mathbf{X}, \mathbf{t})$)?

- this is related to how a dataset generated by randomly sampling from $\mathcal{D}_1$ (usually uniform) and $p(t|\mathbf{x})$ (instead of the unknown distribution $\mathcal{D}_2$) could be similar to the available dataset $\mathcal{T}$
- in particular, what is the probability that the dataset $\mathcal{T} = (\mathbf{X}, \mathbf{t})$ is obtained under the following hypotheses?
  - $n = |\mathbf{t}|$ pairs $\mathbf{x}_i, t_i$ are each other independently sampled
  - $\mathbf{x}_i$ is sampled from $\mathcal{D}_1$ (which we assume uniform)
  - $t_i$ is sampled from $p(t|\mathbf{x}_i)$
- we may use such probability as the quality measure $q(p, \mathcal{T})$ and search the distribution $p^*(t|\mathbf{x})$ that makes $p(\mathbf{X}, \mathbf{t})$ maximum assuming $\mathcal{D}_1$ is the uniform distribution and $\mathcal{D}_2$ is $p^*(t|\mathbf{x})$

# Inferring a best distribution

That is, we consider the probability

$$p(\mathbf{X}, \mathbf{t}) = \prod_{i=1}^{n} p(\mathbf{x}_i, t_i) = \prod_{i=1}^{n} p(t_i|\mathbf{x}_i)p(\mathbf{x}_i) \propto \prod_{i=1}^{n} p(t_i|\mathbf{x}_i) = q(p, \mathcal{T})$$

and look (within some class of distributions) for the conditional probability $p^*(t|\mathbf{x})$ which makes $p(\mathbf{X}, \mathbf{t})$ maximum

Observe that learning the distribution $p^*(t|\mathbf{x})$ which maximizes $q(p, \mathcal{T})$ corresponds, in the probabilistic predictor case, to learning the function $h^*$ which minimizes the empirical risk $\overline{\mathcal{R}}_{\mathcal{T}}(h)$ in the functional predictor case. In both cases, learning is performed through optimization.

# Inferring a best distribution

The same considerations done wrt the inductive bias in the case of a functional predictor, and related to overfitting and underfitting, can be rephrased here wrt the class of possible conditional distributions.

# A different approach

Instead of finding a best distribution $p^* \in \mathcal{P}$ and use it to predict target probabilities as $p^*(y|\mathbf{x})$ for any element $\mathbf{x}$, we could

- consider for each possible conditional distribution $p \in \mathcal{P}$ its quality $q(p, \mathcal{T})$
- compose all conditional distributions $p(y|\mathbf{x})$ each weighted by its quality $q(p, \mathcal{T})$ (for example by means of a weighted averaging)
- apply the resulting distribution

# Different strategies

Assume $q$ takes the form of a probability distribution (of probability distribution)

- first approach: take the modal value (the distribution of maximum quality) and apply it to perform predictions
- second approach: compute the expectation of the distributions, wrt the probability distribution $q$

# Inference of predictive distribution

### Dataset

We assume elements in $\mathcal{T}$ correspond to a set of $n$ samples, independently drawn from the same probability distribution (that is, they are independent and identically distributed, i.i.d): they can be seen as $n$ realizations of a single random variable.

We are interested in learning, starting from $\mathcal{T}$, a predictive distribution $p(\mathbf{x}|\mathbf{X})$ (or $p(\mathbf{x}, t|\mathbf{X}, \mathbf{t})$) for any new element (or element-target pair). We may interpret this as the probability that, in a random sampling, the element actually returned is indeed $\mathbf{x}$ (or $\mathbf{x}, t$).

- in the case that $\mathcal{T} = \mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$, we are interested in deriving the probability distribution $p(\mathbf{x}|\mathbf{X})$ of a new element, given the knowledge of the set $\mathbf{X}$
- in the case that $\mathcal{T} = (\mathbf{X}, \mathbf{t}) = \{(\mathbf{x}_1, t_1), \ldots, (\mathbf{x}_n, t_n)\}$, we are interested in deriving the joint probability distribution $p(\mathbf{x}, t|\mathbf{X}, \mathbf{t})$ or, assuming $p(\mathbf{x}|\mathbf{X}, \mathbf{t})$ uniform and thus also independent from $\mathbf{X}, \mathbf{t}$, the conditional distribution $p(t|\mathbf{x}, \mathbf{X}, \mathbf{t})$, given the knowledge of the set of pairs $\mathbf{X}, \mathbf{t}$

# Probabilistic models

A probabilistic model is a collection of probability distributions with the same structure, defined over the data domain. Probability distribution are instances of the probabilistic model and are characterized by the values assumed by a set of parameters.

Example

In a bivariate gaussian probabilistic model, distributions are characterized by the values assumed by:

1. the mean $\boldsymbol{\mu} = (\mu_1, \mu_2)$
2. the covariance matrix $\boldsymbol{\Sigma} = \begin{pmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{21} & \sigma_{22} \end{pmatrix}$

where $\sigma_{12} = \sigma_{21}$

A probabilistic model could be

**Parametric** if the set of parameters is given, finite, and independent from the data

**Non parametric** if the set of parameters is not given in advance, but derives from the data

# Likelihood

Given a dataset $\mathcal{T}$ and a probability distribution $p$ of parameters $\boldsymbol{\theta}$ defined on the same data domain,

- the likelihood of $\boldsymbol{\theta}$ wrt $\mathcal{T}$ is defined as
$$L(\boldsymbol{\theta}|\mathcal{T}) = p(\mathcal{T}|\boldsymbol{\theta})$$
the probability of the dataset (that the dataset is generated) under distribution $p$ with parameters $\boldsymbol{\theta}$
- while the probability $p(\mathcal{T}|\boldsymbol{\theta})$ is considered as a function of $p(\mathcal{T}|\boldsymbol{\theta})$ with $\boldsymbol{\theta}$ fixed, the likelihood $L(\boldsymbol{\theta}|\mathcal{T})$ is a function of $\boldsymbol{\theta}$ with $\mathcal{T}$ fixed
- parameters $\boldsymbol{\theta}$ are considered as (independent) variables (frequentist interpretation of probability)

# Likelihood

- By assuming that elements in $\mathcal{T}$ are i.i.d.,

$$L(\boldsymbol{\theta}|\mathcal{T}) = p(\mathbf{X}|\boldsymbol{\theta}) = \prod_{i=1}^{n} p(\mathbf{x}_i|\boldsymbol{\theta})$$

in the first case

$$L(\boldsymbol{\theta}|\mathcal{T}) = p(\mathbf{X},\mathbf{t}|\boldsymbol{\theta}) = \prod_{i=1}^{n} p(\mathbf{x}_i, t_i|\boldsymbol{\theta}) = \prod_{i=1}^{n} p(t_i|\mathbf{x}_i,\boldsymbol{\theta})p(\mathbf{x}_i|\boldsymbol{\theta}) = p(\mathbf{x}|\boldsymbol{\theta})\prod_{i=1}^{n} p(t_i|\mathbf{x}_i,\boldsymbol{\theta})$$

$$= p(\mathbf{x})\prod_{i=1}^{n} p(t_i|\mathbf{x}_i,\boldsymbol{\theta}) \propto \prod_{i=1}^{n} p(t_i|\mathbf{x}_i,\boldsymbol{\theta})$$

in the second case, assuming $p(\mathbf{x}|\boldsymbol{\theta})$ uniform

# Maximum likelihood estimate

## Approach

Frequentist point of view: parameters are deterministic variables, whose value is unknown and must be estimated.
Determine the parameter value that maximize the likelihood

$$\boldsymbol{\theta}^* = \operatorname*{argmax}_{\boldsymbol{\theta}} L(\boldsymbol{\theta}|\mathcal{T}) = \operatorname*{argmax}_{\boldsymbol{\theta}} p(\mathbf{X}|\boldsymbol{\theta}) = \operatorname*{argmax}_{\boldsymbol{\theta}} \prod_{i=1}^{n} p(\mathbf{x}_i|\boldsymbol{\theta})$$

or

$$\boldsymbol{\theta}^* = \operatorname*{argmax}_{\boldsymbol{\theta}} L(\boldsymbol{\theta}|\mathcal{T}) = \operatorname*{argmax}_{\boldsymbol{\theta}} p(\mathbf{X}, \mathbf{t}|\boldsymbol{\theta}) = \operatorname*{argmax}_{\boldsymbol{\theta}} p(\mathbf{x}) \prod_{i=1}^{n} p(t_i|\mathbf{x}_i, \boldsymbol{\theta}) = \operatorname*{argmax}_{\boldsymbol{\theta}} \prod_{i=1}^{n} p(t_i|\mathbf{x}_i, \boldsymbol{\theta})$$

# Maximum likelihood estimate

## Log-likelihood

$$l(\boldsymbol{\theta}|\mathcal{T}) = \ln L(\boldsymbol{\theta}|\mathcal{T})$$

is usually preferrable, since products are turned into sums, while $\boldsymbol{\theta}^*$ remains the same (since log is a monotonic function), that is

$$\underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, l(\boldsymbol{\theta}|\mathcal{T}) = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, L(\boldsymbol{\theta}|\mathcal{T})$$

## Estimate

$$\boldsymbol{\theta}^*_{ML} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, p(\mathbf{X}|\boldsymbol{\theta}) = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \sum_{i=1}^{n} \ln p(\mathbf{x}_i|\boldsymbol{\theta})$$

or

$$\boldsymbol{\theta}^*_{ML} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, p(\mathbf{X}, \mathbf{t}|\boldsymbol{\theta}) = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \sum_{i=1}^{n} \ln p(t_i|\mathbf{x}_i, \boldsymbol{\theta})$$

# Maximum likelihood estimate

## Solution

Solve the system

$$\frac{\partial l(\boldsymbol{\theta}|\mathcal{T})}{\partial \theta_i} = 0 \qquad\qquad i = 1, \ldots, d$$

more concisely,

$$\nabla l(\boldsymbol{\theta}|\mathcal{T}) = \mathbf{0}$$

## Prediction

Probability of a new observation $\mathbf{x}$:

$$p(\mathbf{x}|\mathbf{X}) = \int_{\boldsymbol{\theta}} p(\mathbf{x}|\boldsymbol{\theta})p(\boldsymbol{\theta}|\mathbf{X})d\boldsymbol{\theta} \approx \int_{\boldsymbol{\theta}} p(\mathbf{x}|\boldsymbol{\theta}^*_{ML})p(\boldsymbol{\theta}|\mathbf{X})d\boldsymbol{\theta} = p(\mathbf{x}|\boldsymbol{\theta}^*_{ML})\int_{\boldsymbol{\theta}} p(\boldsymbol{\theta}|\mathbf{X})d\boldsymbol{\theta} = p(\mathbf{x}|\boldsymbol{\theta}^*_{ML})$$

Predictive distribution $t|\mathbf{x}$:

$$p(t|\mathbf{x}, \mathbf{X}, \mathbf{t}) = \int_{\boldsymbol{\theta}} p(t|\mathbf{x}, \boldsymbol{\theta})p(\boldsymbol{\theta}|\mathbf{X}, \mathbf{t})d\boldsymbol{\theta} \approx \int_{\boldsymbol{\theta}} p(t|\mathbf{x}, \boldsymbol{\theta}^*_{ML})p(\boldsymbol{\theta}|\mathbf{X})d\boldsymbol{\theta} = p(\mathbf{x}|\boldsymbol{\theta}^*_{ML})\int_{\boldsymbol{\theta}} p(\boldsymbol{\theta}|\mathbf{X}, \mathbf{t})d\boldsymbol{\theta} = p(t|\mathbf{x}, \boldsymbol{\theta}^*_{ML})$$

# Maximum likelihood estimate

Example

Collection $\mathbf{X}$ of $n$ binary events, modeled through a Bernoulli distribution with unknown parameter $\phi$

$$p(x|\phi) = \phi^x(1-\phi)^{1-x}$$

Likelihood: $L(\phi|\mathbf{X}) = \prod_{i=1}^n \phi^{x_i}(1-\phi)^{1-x_i}$

Log-likelihood: $l(\phi|\mathbf{X}) = \sum_{i=1}^n (x_i \ln\phi + (1-x_i)\ln(1-\phi)) = n_1 \ln\phi + n_0 \ln(1-\phi)$

where $n_0$ ($n_1$) is the number of events $x \in \mathbf{X}$ equal to 0 (1)

$$\frac{\partial l(\phi|\mathbf{X})}{\partial\phi} = \frac{n_1}{\phi} - \frac{n_0}{1-\phi} = 0 \qquad \Longrightarrow \qquad {\phi^*}_{ML} = \frac{n_1}{n_0+n_1} = \frac{n_1}{n}$$

# Maximum likelihood estimate

## Example

Linear regression: collection $\mathbf{X}, \mathbf{t}$ of value-target pairs, modeled as $p(\mathbf{x}, t) = p(\mathbf{x})p(t|\mathbf{x}, \mathbf{w}, \sigma^2)$, with $\mathbf{w} \in \mathbb{R}^d$, $w_0 \in \mathbb{R}$:

- $p(\mathbf{x})$ uniform
- $p(t|\mathbf{x}, \mathbf{w}, \sigma^2) = \mathcal{N}(\mathbf{w}^T\mathbf{x} + w_0, 1/\beta)$ ($\beta$, the inverse of the variance, is the precision)

Likelihood:

$$L(\mathbf{t}|\mathbf{X}, \mathbf{w}, w_0, \beta) = \prod_{i=1}^{n} p(t_i|\mathbf{x}_i, \mathbf{w}, w_0, \beta) = \prod_{i=1}^{n} \mathcal{N}(\mathbf{w}^T\mathbf{x}_i + w_0, \beta)$$

Log-likelihood:

$$l(\mathbf{t}|\mathbf{X}, \mathbf{w}, w_0, \beta) = -\frac{\beta}{2}\sum_{i=1}^{n}(\mathbf{w}^T\mathbf{x}_i + w_0 - t_i)^2 + \frac{n}{2}\ln\beta - \frac{n}{2}\ln(2\pi)$$

# Maximum likelihood estimate

Example

$$\frac{\partial}{\partial w_k} l(\mathbf{t}|\mathbf{X},\mathbf{w},w_0,\beta) = -\frac{\beta}{2}\sum_{i=1}^n(\mathbf{w}^T\mathbf{x}_i + w_0 - t_i)x_{ik} \qquad k=1,\ldots,d$$

$$\frac{\partial}{\partial w_0} l(\mathbf{t}|\mathbf{X},\mathbf{w},w_0,\beta) = -\frac{\beta}{2}\sum_{i=1}^n(\mathbf{w}^T\mathbf{x}_i + w_0 - t_i)$$

$$\frac{\partial}{\partial \beta} l(\mathbf{t}|\mathbf{X},\mathbf{w},w_0,\beta) = -\frac{1}{2}\sum_{i=1}^n(\mathbf{w}^T\mathbf{x}_i + w_0 - t_i)^2 + \frac{n}{2\beta}$$

The ML estimation for $\mathbf{w}, w_0$ (linear regression coefficients) is obtained as the solution of the $(d+1, d+1)$ linear system

$$\sum_{i=1}^n(\mathbf{w}^T\mathbf{x}_i + w_0 - t_i)x_{ik} = 0 \qquad k=1,\ldots,d$$

$$\sum_{i=1}^n(\mathbf{w}^T\mathbf{x}_i + w_0 - t_i) = 0$$

The ML estimation for $\beta$ is obtained by

# ML AND OVERFITTING

## Overfitting

Maximizing the likelihood of the observed dataset tends to result into an estimate too sensitive to the dataset values, hence into overfitting. The obtained estimates are suitable to model observed data, but may be too specialized to be used to model different datasets.

## Penalty functions

An additional function $P(\boldsymbol{\theta})$ can be introduced with the aim to limit overfitting and the overall complexity of the model. This results in the following function to maximize

$$C(\boldsymbol{\theta}|\mathbf{X}) = l(\boldsymbol{\theta}|\mathbf{X}) - P(\boldsymbol{\theta})$$

as a common case, $P(\boldsymbol{\theta}) = \frac{\gamma}{2}||\boldsymbol{\theta}||^2$, with $\gamma$ a tuning parameter.

# Maximum a posteriori estimate

## Idea

Inference through maximum a posteriori (MAP) is similar to ML, but $\boldsymbol{\theta}$ is now considered as a random variable (bayesian approach), whose distribution has to be derived from observations, also taking into account previous knowledge (prior distribution). The parameter value maximizing

$$p(\boldsymbol{\theta}|\mathcal{T}) = \frac{p(\mathcal{T}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\mathcal{T})}$$

is computed.

# Maximum a posteriori estimate

## Estimate

$$\begin{aligned}\boldsymbol{\theta}^*_{MAP} &= \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, p(\boldsymbol{\theta}|\mathcal{T}) = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, p(\mathcal{T}|\boldsymbol{\theta})p(\boldsymbol{\theta}) \\ &= \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, L(\boldsymbol{\theta}|\mathcal{T})p(\boldsymbol{\theta}) = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, \left(l(\boldsymbol{\theta}|\mathcal{T}) + \ln p(\boldsymbol{\theta})\right)\end{aligned}$$

which results into

$$\boldsymbol{\theta}^*_{MAP} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \left(\sum_{i=1}^{n} \ln p(\mathbf{x}_i|\boldsymbol{\theta}) + \ln p(\boldsymbol{\theta})\right)$$

or

$$\boldsymbol{\theta}^*_{MAP} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \left(\sum_{i=1}^{n} \ln p(t_i|\mathbf{x}_i, \boldsymbol{\theta}) + \ln p(\boldsymbol{\theta})\right)$$

# MAP AND GAUSSIAN PRIOR

## Hypothesis

Assume $\boldsymbol{\theta}$ is distributed around the origin as a multivariate gaussian with uniform variance and null covariance.That is,

$$p(\boldsymbol{\theta}) \sim \mathcal{N}(\boldsymbol{\theta}|\mathbf{0}, \sigma^2) = \frac{1}{(2\pi)^{d/2}\sigma^d} e^{-\frac{\|\boldsymbol{\theta}\|^2}{2\sigma^2}} \propto e^{-\frac{\|\boldsymbol{\theta}\|^2}{2\sigma^2}}$$

## Inference

From the hypothesis,

$$\begin{aligned}\boldsymbol{\theta}^*_{MAP} &= \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\; p(\boldsymbol{\theta}|\mathcal{T}) = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\; (l(\boldsymbol{\theta}|\mathcal{T}) + \ln p(\boldsymbol{\theta})) \\ &= \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \left( l(\boldsymbol{\theta}|\mathcal{T}) + \ln e^{-\frac{\|\boldsymbol{\theta}\|^2}{2\sigma^2}} \right) = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \left( l(\boldsymbol{\theta}|\mathcal{T}) - \frac{\|\boldsymbol{\theta}\|^2}{2\sigma^2} \right)\end{aligned}$$

which is equal to the penalty function introduced before, if $\gamma = \frac{1}{\sigma^2}$

# MAP ESTIMATE

### Example

Collection $\mathbf{X}$ of $n$ binary events, modeled as a Bernoulli distribution with unknown parameter $\phi$. Initial knowledge of $\phi$ is modeled as a Beta distribution:

$$p(\phi|\alpha,\beta) = \text{Beta}(\phi|\alpha,\beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\phi^{\alpha-1}(1-\phi)^{\beta-1}$$

Log-likelihood

$$l(\phi|\mathbf{X}) = \sum_{i=1}^{n}(x_i \ln\phi + (1-x_i)\ln(1-\phi)) = n_1\ln\phi + n_0\ln(1-\phi)$$

$$\frac{\partial}{\partial\phi}(l(\phi|\mathbf{X}) + \ln\text{Beta}(\phi|\alpha,\beta)) = \frac{n_1}{\phi} - \frac{n_0}{1-\phi} + \frac{\alpha-1}{\phi} - \frac{\beta-1}{1-\phi} = 0 \qquad \Longrightarrow$$

$$\phi^*_{MAP} = \frac{n_1+\alpha-1}{n+\alpha+\beta-2}$$

# Note

## Gamma function

The function

$$\Gamma(x) = \int_0^{\infty} t^{x-1} e^{-t} dt$$

is an extension of the factorial to the real numbers field: in fact, for any integer $x$,

$$\Gamma(x) = (x-1)!$$

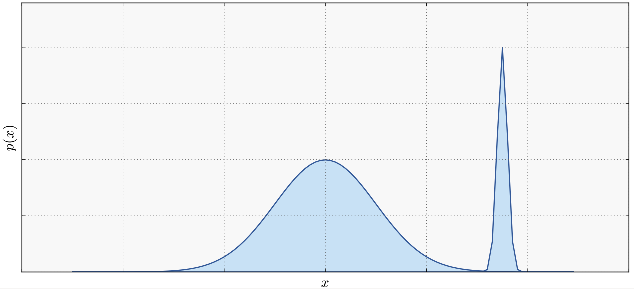
# Applying Bayesian inference

## Mode and mean

Once the posterior distribution

$$p(\boldsymbol{\theta}|\mathbf{X}) = \frac{p(\mathbf{X}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\mathbf{X})} = \frac{p(\mathbf{X}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{\int_{\theta} p(\mathbf{X}|\boldsymbol{\theta})d\boldsymbol{\theta}}$$

is available, MAP estimate computes the most probable value (mode) $\boldsymbol{\theta}_{MAP}$ of the distribution. This may lead to inaccurate estimates, as in the figure below:



# Applying bayesian inference

### Mode and mean

A better estimation can be obtained by applying a fully bayesian approach and referring to the whole posterior distribution, for example by deriving the expectation of $\boldsymbol{\theta}$ w.r.t. $p(\boldsymbol{\theta}|\mathbf{X})$,

$$\boldsymbol{\theta}^* = E_{p(\boldsymbol{\theta}|\mathbf{X})}[\boldsymbol{\theta}] = \int_{\boldsymbol{\theta}} \boldsymbol{\theta} p(\boldsymbol{\theta}|\mathbf{X}) d\boldsymbol{\theta}$$

# Bayesian estimate

Example

Collection $\mathbf{X}$ of $n$ binary events, modeled as a Bernoulli distribution with unknown parameter $\phi$. Initial knowledge of $\phi$ is modeled as a Beta distribution:

$$p(\phi|\alpha,\beta) = \text{Beta}(\phi|\alpha,\beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\phi^{\alpha-1}(1-\phi)^{\beta-1}$$

Posterior distribution

$$p(\phi|\mathbf{X},\alpha,\beta) = \frac{\prod_{i=1}^{N}\phi^{x_i}(1-\phi)^{1-x_i}p(\phi|\alpha,\beta)}{p(\mathbf{X})} = \frac{\phi^{N_1+\alpha-1}(1-\phi)^{N_0+\beta-1}}{Z}$$

Hence,

$$p(\phi|\mathbf{X},\alpha,\beta) = \text{Beta}(\phi|\alpha+N_1,\beta+N_0)$$

# Model selection

In the process described, a model (structure, hyper-parameter values) must be identified, in some way. How can we deal with this problem?

This is performed through model selection: identify, in a set of possible models, the one which we expect is best to represent the available data.

Indeed, the one whose best (or a good) instantiation is best to represent the available data

We need a way to compare models (not their instantiations), given the dataset

# Model selection in practice

## Validation

**Test set** Dataset is split into Training set (used for learning parameters) and Test set (used for measuring effectiveness). Good for large datasets: otherwise, small resulting training and test set (few data for fitting and validation)

**Cross validation** Dataset partitioned into $K$ equal-sized sets. Iteratively, in $K$ phases, use one set as test set and the union of the other $K-1$ ones as training set ($K$-fold cross validation). Average validation measures.

As a particular case, iteratively leave one element out and use all other points as training set (Leave-one-out cross validation).

Time consuming for large datasets and for models which are costly to fit.

# Model selection in practice

## Information measures

Faster methods to compare model effectiveness, based on computing measures which take into account data fitting and model complexity.

**Akaike Information Criterion (AIC)** Let $\boldsymbol{\theta}$ be the set of parameters of the model and let $\boldsymbol{\theta}_{ML}$ be their maximum likelihood estimate on the dataset $\mathbf{X}$. Then,

$$AIC = 2|\boldsymbol{\theta}| - 2\log p(\mathbf{X}|\boldsymbol{\theta}_{ML}) = 2|\boldsymbol{\theta}| - 2\max_{\boldsymbol{\theta}} l(\boldsymbol{\theta}|\mathbf{X})$$

lower values correspond to models to be preferred.

**Bayesian Information Criterion (BIC)** A variant of the above, defined as

$$\begin{aligned} BIC &= |\boldsymbol{\theta}| - \log|\mathbf{X}| 2\log p(\mathbf{X}|\boldsymbol{\theta}_{ML}) \\ &= |\boldsymbol{\theta}|\log|\mathbf{X}| - 2\max_{\boldsymbol{\theta}} l(\boldsymbol{\theta}|\mathbf{X}) \end{aligned}$$

# Language modeling

A language model is a (categorical) probability distribution on a vocabulary of terms (possibly, all words which occur in a large collection of documents).

## Use

A language model can be applied to predict the next term occurring in a text. The probability of occurrence of a term is related to its information content and is at the basis of a number of information retrieval techniques.

## Hypothesis

It is assumed that the probability of occurrence of a term is independent from the preceding terms in a text (bag of words model).

## Generative model

Given a language model, it is possible to sample from the distribution to generate random documents statistically equivalent to the documents in the collection used to derive the model.

# Language model

- Let $\mathcal{T} = \{t_1, \ldots, t_n\}$ be the set of terms occurring in a given collection $\mathcal{C}$ of documents, after stop word (common, non informative terms) removal and stemming (reduction of words to their basic form).
- For each $i = 1, \ldots, n$ let $m_i$ be the multiplicity (number of occurrences) of term $t_i$ in $\mathcal{C}$
- A language model can be derived as a categorical distribution associated to a vector $\hat{\boldsymbol{\phi}} = (\hat{\phi}_1, \ldots, \hat{\phi}_n)^T$ of probabilities: that is,

$$0 \leq \hat{\phi}_i \leq 1 \quad i = 1, \ldots, n \qquad\qquad \sum_{i=1}^{n} \hat{\phi}_i = 1$$

where $\hat{\phi}_j = p(t_j|\mathcal{C})$

# Learning a language model by ML

Applying maximum likelihood to derive term probabilities in the language model results into setting

$$\hat{\phi}_j = p(t_j|\mathcal{C}) = \frac{m_j}{\sum_{k=1}^{n} m_k} = \frac{m_j}{N}$$

where $N = \sum_{i=1}^{n} m_i$ is the overall number of occurrences in $\mathcal{C}$ after stopword removal.

## Smoothing

According to this estimate, a term $t$ which never occurred in $\mathcal{C}$ has zero probability to be observed (black swan paradox). Due to overfitting the model to the observed data, typical of ML estimation.

Solution: assign small, non zero, probability to events (terms) not observed up to now. This is called smoothing.

# Bayesian learning of a language model

We may apply the dirichlet-multinomial model:

- this implies defining a Dirichlet prior $\text{Dir}(\boldsymbol{\phi}|\boldsymbol{\alpha})$, with $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_n)$ that is,

$$p(\phi_1, \ldots, \phi_n|\boldsymbol{\alpha}) = \frac{1}{\Delta(\alpha_1, \ldots, \alpha_n)} \prod_{i=1}^{n} \phi_i^{\alpha_i - 1}$$

- the posterior distribution of $\boldsymbol{\phi}$ after $\mathcal{C}$ has been observed is then $\text{Dir}(\boldsymbol{\phi}|\boldsymbol{\alpha}')$, where

$$\boldsymbol{\alpha}' = (\alpha_1 + m_1, \alpha_2 + m_2, \ldots, \alpha_n + m_n)$$

  that is,

$$p(\phi_1, \ldots, \phi_n|\boldsymbol{\alpha}') = \frac{1}{\Delta(\alpha_1 + m_1, \ldots, \alpha_n + m_n)} \prod_{i=1}^{n} \phi_i^{\alpha_i + m_i - 1}$$

# Bayesian learning of a language model

The language model $\hat{\phi}$ corresponds to the predictive posterior distribution

$$\hat{\phi}_j = p(t_j|\mathcal{C}, \boldsymbol{\alpha}) = \int p(t_j|\phi)p(\phi|\mathcal{C}, \alpha)d\phi = \int \phi_j \text{Dir}(\boldsymbol{\phi}|\boldsymbol{\alpha}')d\boldsymbol{\phi} = E[\phi_j]$$

where $E[\phi_j]$ is taken w.r.t. the distribution $\text{Dir}(\boldsymbol{\phi}|\boldsymbol{\alpha}')$. Then,

$$\hat{\phi}_j = \frac{\alpha_j'}{\sum_{k=1}^n \alpha_k'} = \frac{\alpha_j + m_j}{\sum_{k=1}^n (\alpha_k + m_k)} = \frac{\alpha_j + m_j}{\alpha_0 + N}$$

The $\alpha_j$ term makes it impossible to obtain zero probabilities (Dirichlet smoothing).

Non informative prior: $\alpha_i = \alpha$ for all $i$, which results into

$$p(t_j|\mathcal{C}, \boldsymbol{\alpha}) = \frac{m_j + \alpha}{\alpha V + N}$$

where $V$ is the vocabulary size.

# Naive bayes classifiers

A language model can be applied to derive document classifiers into two or more classes.

- given two classes $C_1, C_2$, assume that, for any document $d$, the probabilities $p(C_1|d)$ and $p(C_2|d)$ are known: then, $d$ can be assigned to the class with higher probability
- how to derive $p(C_k|d)$ for any document, given a collection $\mathcal{C}_1$ of documents known to belong to $C_1$ and a similar collection $\mathcal{C}_2$ for $C_2$? Apply Bayes' rule:

  $$p(C_k|d) \propto p(d|C_k)p(C_k)$$

  the evidence $p(d)$ is the same for both classes, and can be ignored.
- we have still the problem of computing $p(C_k)$ and $p(d|C_k)$ from $\mathcal{C}_1$ and $\mathcal{C}_2$

# NAIVE BAYES CLASSIFIERS

## Computing $p(C_k)$

The prior probabilities $p(C_k)$ ($k = 1, 2$) can be easily estimated from $\mathcal{C}_1, \mathcal{C}_2$: for example, by applying ML, we obtain

$$p(C_k) = \frac{|\mathcal{C}_1|}{|\mathcal{C}_1| + |\mathcal{C}_2|}$$

## Computing $p(d|C_k)$

For what concerns the likelihoods $p(d|C_k)$ ($k = 1, 2$), we observe that $d$ can be seen, according to the bag of words assumption, as a multiset of $n_d$ terms

$$d = \{\bar{t}_1, \bar{t}_2, \ldots, \bar{t}_{n_d}\}$$

By applying the product rule, it results

$$p(d|C_k) = p(\bar{t}_1, \ldots, \bar{t}_{n_d}|C_k) = p(\bar{t}_1|C_k)p(\bar{t}_2|\bar{t}_1, C_k) \cdots p(\bar{t}_{n_d}|\bar{t}_1, \ldots, \bar{t}_{n_d-1}, C_k)$$

# Naive bayes classifiers

## The naive Bayes assumption

Computing $p(d|C_k)$ is much easier if we assume that terms are pairwise conditionally independent, given the class $C_k$, that is, for $i, j = 1. \dots, n_d$ and $k = 1, 2$,

$$p(\bar{t}_i, \bar{t}_j|C_k) = p(\bar{t}_i|C_k)p(\bar{t}_2|C_k)$$

as, a consequence,

$$p(d|C_k) = \prod_{j=1}^{n_d} p(\bar{t}_j|C_k)$$

## Language models and NB classifiers

The probabilities $p(\bar{t}_j|C_k)$ are available for all terms if language models have been derived for $C_1$ and $C_2$, respectively from documents in $\mathcal{C}_1$ and $\mathcal{C}_2$.

# Feature selection by mutual information

## Feature selection

The set of probabilities in a language model can be exploited to identify the most relevant terms for classification, that is terms whose presence or absence in a document best characterizes the class of the document.

## Mutual information

To measure relevance, we can apply the set of mutual informations $\{I_1, \ldots, I_n\}$

$$
\begin{aligned}
I_j &= \sum_{k=1,2} p(t_j, C_k) \log \frac{p(t_j, C_k)}{p(t_j)p(C_k)} \\
&= \sum_{k=1,2} p(C_k|t_j)p(t_j) \log \frac{p(C_k|t_j)}{p(C_k)} = p(t_j) KL(p(C_k|t_j)||p(C_k))
\end{aligned}
$$

here, $KL$ is a measure of the amount of information on class distributions provided by the presence of $t_j$. This amount is weighted by the probability of occurrence of $t_j$.

# Feature selection by mutual information

## Mutual information

Since $p(t_j, C_k) = p(C_k|t_j)p(t_j) = p(t_j|C_k)p(C_k)$, $I_j$ can be estimated as

$$\begin{aligned} I_j &= p(t_j|C_1)p(C_1)\log\frac{p(t_j|C_1)}{p(t_j)} + p(t_j|C_2)p(C_2)\log\frac{p(t_j|C_2)}{p(t_j)} \\ &= \phi_{j1}\pi_1\log\frac{\phi_{j1}}{\phi_{j1}\pi_1+\phi_{j2}\pi_2} + \phi_{j2}\pi_2\log\frac{\phi_{j2}}{\phi_{j1}\pi_1+\phi_{j2}\pi_2} \end{aligned}$$

where $\phi_{jk}$ is the estimated probability of $t_j$ in documents of class $C_k$ and $\pi_k$ is the estimated probability of a document of class $C_k$ in the collection.

A selection of the most significant terms can be performed by selecting the set of terms with highest mutual information $I_j$.