Probabilistic learning

Course of Machine Learning
Master Degree in Computer Science
University of Rome “Tor Vergata”
a.a. 2024-2025

Giorgio Gambosi

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

Deriving a probabilistic predictor results into deriving, from the training set $\mathcal{T}$, an algorithm computing a conditional distribution $p^*(t|\mathbf{x})$ which approximates the correct, unknown distribution p_C. An independent **decision strategy** will then be applied to $p^*(t|\mathbf{x})$ to return a specific prediction $h(\mathbf{x})$: this usually results into returning the target value t^* with maximum probability according to $p^*(t|\mathbf{x})$, that is $h(\mathbf{x}) = \underset{t\in\mathcal{Y}}{\text{argmax}}\, p^*(t|\mathbf{x})$. However, different decision strategies could be applied, for example in the case of different cost for different error types.

Decision theory

In general, we wish to select the best actions to perform, given a cost associated to each action, in order to minimize the expected cost. Let us consider the case of classification: in this case, assuming the joint distribution $p(\mathbf{x}, \mathcal{C}) = p(\mathcal{C}|\mathbf{x})p(\mathbf{x})$ is known, we want to derive a partition of the input space $\mathcal{X}$ into **decision regions** $\mathcal{R}_k$, with region $\mathcal{R}_k$ containing all elements which are predicted as belonging to class $\mathcal{C}_k$.

Let us first consider, for simplicity, the binary case $\{0, 1\}$: here, an item $\mathbf{x}$ is classified correctly with probability

$$\hat{p}(\mathbf{x}) = p(\mathbf{x} \in \mathcal{R}_0, \mathcal{C}_0) + p(\mathbf{x} \in \mathcal{R}_1, \mathcal{C}_1) = \int_{\mathcal{R}_0} p(\mathbf{x}, \mathcal{C}_0)d\mathbf{x} + \int_{\mathcal{R}_1} p(\mathbf{x}, \mathcal{C}_1)d\mathbf{x}$$

this probability is maximized if each $\mathbf{x}$ is assigned to the class $\mathcal{C}_k$ with maximum probability $p(\mathbf{x}, \mathcal{C}_k)$, hence the decision region $\mathcal{R}_i$ corresponds to the set of points $\mathbf{x}$ such that $i = \text{argmax}\, p(\mathbf{x}, \mathcal{C}_i)$. This results in assigning $\mathbf{x}$ to class $\mathcal{C}_0$ if

$$p(\mathbf{x}, \mathcal{C}_0) > p(\mathbf{x}, \mathcal{C}_1)$$

that is if

$$\frac{p_C(\mathcal{C}_0|\mathbf{x})p_M(\mathbf{x})}{p_C(\mathcal{C}_1|\mathbf{x})p_M(\mathbf{x})} = \frac{p_C(\mathcal{C}_0|\mathbf{x})}{p_C(\mathcal{C}_1|\mathbf{x})} > 1$$

These considerations are immediately extendable to K-class classification ($K > 2$): the probability of correct classification is now

$$\hat{p}(\mathbf{x}) = \sum_{i=0}^{K-1} p(\mathbf{x} \in \mathcal{R}_i, C_i) = \sum_{i=1}^{K-1} \int_{\mathcal{R}_i} p(\mathbf{x}, C_i)d\mathbf{x}$$

which is maximized, again, if $\mathcal{R}_i$ corresponds to the set of points $\mathbf{x}$ such that $i = \underset{0\leq i\leq K-1}{\operatorname{argmax}}\; p(\mathbf{x}, \mathcal{C}_i)$, resulting in the generalization of the decision rule above

$$h(\mathbf{x}) = \underset{0\leq i\leq K-1}{\operatorname{argmax}}\; p(\mathbf{x}, \mathcal{C}_i)$$

If we assume now that error cost is not uniform, we have to take into account the values of such costs: by $L_{ij}, 0 \leq i, j \leq K-1$ we denote the cost associated to the event that $\mathbf{x} \in \mathcal{C}_i$ is classified as belonging to $\mathcal{C}_j$, that is $\mathbf{x} \in \mathcal{R}_j$.

The expected classification cost of element $\mathbf{x}$ is then

$$\underset{(\mathbf{x},\mathcal{C})\sim p}{\mathbb{E}}[L] = \sum_i \sum_j \int_{\mathcal{R}_j} L_{ij} p(\mathbf{x}, \mathcal{C}_i) d\mathbf{x} = \sum_i \sum_j \int_{\mathcal{R}_j} L_{ij} p(\mathcal{C}_i|\mathbf{x}) p(\mathbf{x}) d\mathbf{x}$$

we assume that, in general, a classification cost is defined even if $\mathbf{x}$ is classified correctly: in many cases however it is assumed $L_{ii} = 0$ for all i.

In order to minimize such value, we choose $\mathcal{R}_j$ assigning $\mathbf{x}$ to class $\mathcal{C}_j$ (hence including it in $\mathcal{R}_j$) if

$$j = \underset{k}{\operatorname{argmin}} \sum_i L_{ik} p(\mathbf{x}, \mathcal{C}_i)$$

or

$$j = \underset{k}{\operatorname{argmin}} \sum_i L_{ik} p(\mathcal{C}_i|\mathbf{x})$$

In the binary case, applying the considerations above the expected cost of classifying element $\mathbf{x}$ as belonging to $\mathcal{C}_0$ is

$$L_{00} p(C_0|\mathbf{x}) + L_{01} p(C_1|\mathbf{x})$$

while

$$L_{10} p(C_0|\mathbf{x}) + L_{11} p(C_1|\mathbf{x})$$

is the cost of classifying it as belonging to $\mathcal{C}_1$.

Element $\mathbf{x}$ is then assigned to $\mathcal{R}_0$ if

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

that is, if

$$(L_{10} - L_{00}) p(C_0|\mathbf{x}) > (L_{01} - L_{11}) p(C_1|\mathbf{x})$$

hence if

$$\frac{p(C_0|\mathbf{x})}{p(C_1|\mathbf{x})} > \frac{L_{01} - L_{11}}{L_{10} - L_{00}}$$

that is, we compare the increase of cost occurring when an element is assigned to the wrong class, for the cases when it is assigned to class 0 or to class 1. Observe that this just the ratio between the misclassification cost in the two cases when we assume no cost in classifying correctly.

This corresponds to having a threshold

$$\theta = \frac{L_{01} - L_{11}}{L_{10} + L_{01} - L_{00} - L_{11}}$$

such that $\mathbf{x}$ is assigned to class $\mathcal{C}_0$ (that is $\mathcal{R}_0$ includes $\mathbf{x}$) iff $p(\mathcal{C}_0|\mathbf{x}) > \theta$. Note that if we assume $L_{00} = L_{11} = 0$ (no cost for correct predictions) and $L_{01} = L_{10}$ (errors have the same cost), then $\theta = \frac{1}{2}$

Typical example: medical diagnosis

- $C_k = \{0, 1\}$ (sick, healthy)
- $L = \begin{bmatrix} 0 & 100 \\ 1 & 0 \end{bmatrix}$: strong cost of not realizing of a sick patient

The expected loss

$$\underset{(\mathbf{x},\mathcal{C})\ p}{\mathbb{E}}[L] = \int_{\mathcal{R}_1} L_{01} p(x, C_0) dx + \int_{\mathcal{R}_0} L_{10} p(x, C_1) dx$$
$$= \int_{\mathcal{R}_1} 100 \cdot p(x, C_0) dx + \int_{\mathcal{R}_0} p(x, C_1) dx$$

and the assignment rule to $\mathcal{R}_1$ is

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

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 in a neighborhood of 1 (for example in $[1-\varepsilon, 1+\varepsilon]$) the prediction is not returned. Then, in this framework $\mathbf{x}$ is in $\mathcal{R}_0$ if

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

By the same considerations above, this results into $p(\mathcal{C}_0|\mathbf{x}) > \theta'$, where

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

it is not hard to prove that $\theta' > \theta$ if $L_{01}L_{10} > L_{00}L_{11}$ that is if the cost of errors is greater than the cost of correct predictions (as we expect).

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

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

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

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

this corresponds to

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

again, $1 - \theta''$ can be proved to be smaller than θ.

In summary, we have that:

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

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

we wish to minimize the expected loss w.r.t. $h(\mathbf{x})$ (functional minimization)

$$\underset{(\mathbf{x},t)\sim p}{\mathbb{E}}[L] = \int\int (h(\mathbf{x}) - t)^2 p(\mathbf{x}, t) d\mathbf{x} dt$$

this expectation is minimized by the regression function

$$h^*(\mathbf{x}) = \int p(t|\mathbf{x}) t dt = \underset{t\sim p_c(\cdot|\mathbf{x})}{\mathbb{E}}[t]$$

that is usually denoted as $\mathbb{E}[t|\mathbf{x}]$

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

$$\begin{aligned}(h(\mathbf{x}) - t)^2 &= (h(\mathbf{x}) - \mathbb{E}[t|\mathbf{x}] + \mathbb{E}[t|\mathbf{x}] - t)^2 \\ &= (h(\mathbf{x}) - \mathbb{E}[t|x])^2 + (\mathbb{E}[\mathbf{x}|t] - t)^2 + 2\Big((h(\mathbf{x}) - \mathbb{E}[t|x])(\mathbb{E}[t|x] - t)\Big)\end{aligned}$$

Then,

$$\underset{(\mathbf{x},t)\sim p}{\mathbb{E}}[L] = \int\int (h(\mathbf{x}) - \mathbb{E}[t|\mathbf{x}])^2 p(\mathbf{x}) d\mathbf{x} dt + \int\int (\mathbb{E}[t|\mathbf{x}] - t)^2 p(\mathbf{x}) d\mathbf{x} dt$$

which is minimized w.r.t. $h(\mathbf{x})$ by setting the first term to 0, that is when $h(\mathbf{x}) = \mathbb{E}[t|\mathbf{x}]$.

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

The considerations above refer to the case that the real conditional distribution $p_C(t|\mathbf{x})$ is available. Since this is not the case, we need to infer for $\mathcal{T}$ a distribution $p(t|\mathbf{x})$ which is a good approximation of p_C.

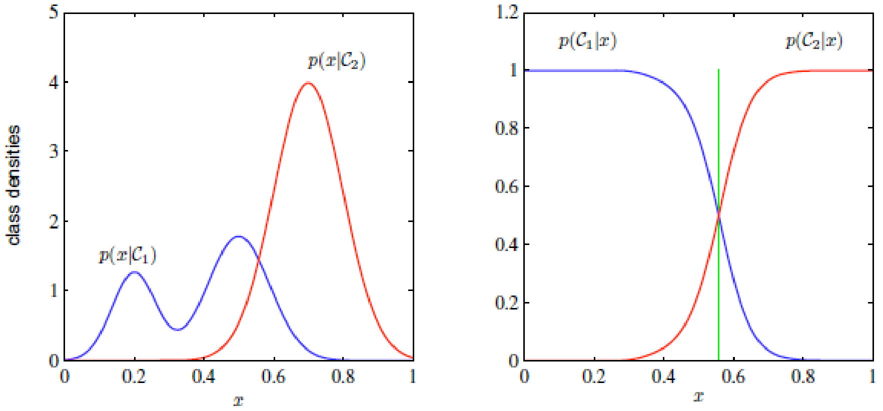
Two different approaches can be applied here:

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

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

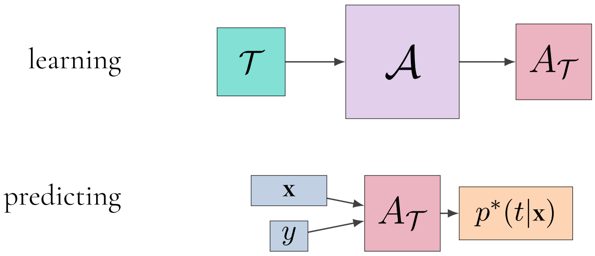
   to derive (at least up to a multiplicative constant) the posterior probabilities $p(\mathcal{C}_k|\mathbf{x})$

2. Discriminative probabilistic models. Inference of class probabilities $p(\mathcal{C}_k|\mathbf{x})$ directly from $\mathcal{T}$



With this aim,

1. we may 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 $\mathbf{x}$, apply $p^*(t|\mathbf{x})$ to assign probabilities for each possible value of the corresponding target



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 $p^*(t|\mathbf{x})$

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.

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

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

We assume elements in the dataset $\mathcal{T}$ 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**.

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 $\Sigma = \begin{pmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{21} & \sigma_{22} \end{pmatrix}$

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

Given a dataset $\mathcal{T}$ and a probability distribution p with 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 under distribution p with parameters $\boldsymbol{\theta}$, that is that the dataset is generated by independently sampling points from $p(\mathbf{x}, t; \boldsymbol{\theta})$.

- 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)

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

$$L(\boldsymbol{\theta}|\mathcal{T}) = p(\mathrm{X}|\boldsymbol{\theta}) = \prod_{i=1}^{n} p(\mathrm{x}_i|\boldsymbol{\theta}) \qquad \text{in the first case}$$
$$L(\boldsymbol{\theta}|\mathcal{T}) = p(\mathrm{X}, \mathbf{t}|\boldsymbol{\theta}) = \prod_{i=1}^{n} p(\mathrm{x}_i, t_i|\boldsymbol{\theta}) = \prod_{i=1}^{n} p(t_i|\mathrm{x}_i, \boldsymbol{\theta})p(\mathrm{x}_i|\boldsymbol{\theta}) = p(\mathrm{x}|\boldsymbol{\theta})\prod_{i=1}^{n} p(t_i|\mathrm{x}_i, \boldsymbol{\theta})$$
$$= p(\mathrm{x})\prod_{i=1}^{n} p(t_i|\mathrm{x}_i, \boldsymbol{\theta}) \propto \prod_{i=1}^{n} p(t_i|\mathrm{x}_i, \boldsymbol{\theta}) \qquad \text{in the second case, assuming } p(\mathrm{x}|\boldsymbol{\theta}) \text{ uniform}$$

Maximum likelihood estimate

Frequentist point of view: parameters are deterministic variables, whose value is unknown and must be estimated.

Determine the parameter value that maximize the likelihood

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

or

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

The 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

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

The resulting optimization problem is then

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

or

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

A solution is computed solving the set of equations

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

more concisely, setting the gradient to 0

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

Notice that the null gradient condition is only a necessary condition for the maximization of the ML function considered, since in this case we can only say that the corresponding point is a stationary point (that is a maximum, a minimum, or a saddle point). Even in the case that the point is a maximum (which could be verified by estimating the second derivative or in general the Hessian), we may conclude that it is a **local** maximum, while we are interested to the global maximum.

These issues are tipically dealt with either by considering cases where, for example, there is only a stationary point and such a point is a maximum (hence the global one), or applying more complex maximum search strategies.

Once the optimum $\boldsymbol{\theta}^*_{ML}$ is computed, predictions can be performed by estimating, for any new observation $\mathbf{x}$, its probability:

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

and the conditional distribution $t|\mathbf{x}$ of the associated target value:

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

Collection X of n binary events, modeled through a Bernoulli distribution with unknown parameter ϕ

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

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

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

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

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

Linear regression: collection $\mathrm{X},\mathbf{t}$ 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
- $p(t|\mathbf{x},\mathbf{w},\sigma^2) = \mathcal{N}(\mathbf{w}^T\mathbf{x} + w_0, 1/\beta)$ (β, the inverse of the variance, is the **precision**)

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

Log-likelihood:

$$\begin{aligned} l(\mathbf{t}|\mathrm{X},\mathbf{w},w_0,\beta) &= \sum_{i=1}^n \ln p(t_i|\mathbf{x}_i,\mathbf{w},w_0,\beta) = \sum_{i=1}^n \ln\left(\sqrt{\frac{\beta}{2\pi}}e^{-\frac{\beta(\mathbf{w}^T\mathbf{x}_i+w_0-t_i)^2}{2}}\right) \\ &= \sum_{i=1}^n \left(-\frac{\beta(\mathbf{w}^T\mathbf{x}_i+w_0-t_i)^2}{2} + \frac{1}{2}\ln\beta - \frac{1}{2}\ln(2\pi)\right) \\ &= -\frac{\beta}{2}\sum_{i=1}^n (\mathbf{w}^T\mathbf{x}_i+w_0-t_i)^2 + \frac{n}{2}\ln\beta - \frac{n}{2}\ln(2\pi) \end{aligned}$$

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

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

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

The ML estimation for β is obtained by

$$-\frac{1}{2}\sum_{i=1}^n (\mathbf{w}^T\mathbf{x}_i + w_0 - t_i)^2 + \frac{n}{2\beta} = 0 \qquad \Longrightarrow \qquad \beta_{ML} = \left(\frac{1}{n}\sum_{i=1}^n (\mathbf{w}^T\mathbf{x}_i + w_0 - t_i)^2\right)^{-1}$$

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.

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

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

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

Maximum a posteriori estimate

Inference through maximum a posteriori (MAP) is similar to ML, but $\boldsymbol{\theta}$ is now considered as a random variable (following a 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 then computed.

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

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

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

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

From the hypothesis,

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

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

Collection X of n binary events, modeled as a Bernoulli distribution with unknown parameter ϕ. Initial knowledge of ϕ is modeled as a Beta distribution:

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

Log-likelihood

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

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

$$\phi^*_{MAP} = \frac{N_1+\alpha-1}{n_0+n_1+\alpha+\beta-2} = \frac{n_1+\alpha-1}{n+\alpha+\beta-2}$$

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,

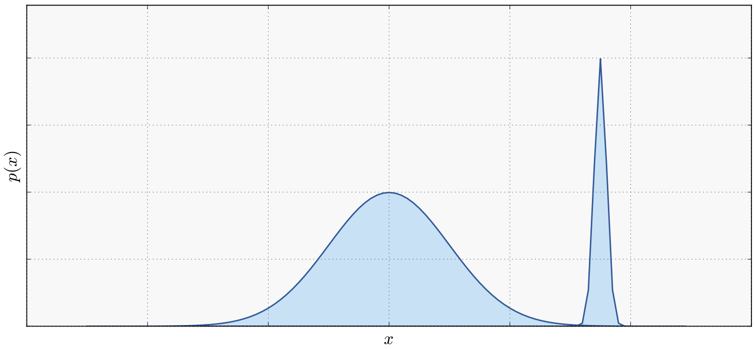
$$\Gamma(x) = (x-1)!$$

Applying bayesian inference

Once the posterior distribution

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

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:



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. $p(\boldsymbol{\theta}|\mathrm{X})$,

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

Collection X of n binary events, modeled as a Bernoulli distribution with unknown parameter ϕ. Initial knowledge of ϕ 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

$$\begin{aligned} p(\phi|\mathrm{X},\alpha,\beta) &= \frac{\prod_{i=1}^N \phi^{x_i}(1-\phi)^{1-x_i} p(\phi|\alpha,\beta)}{p(\mathrm{X})} \\ &= \frac{\phi^{N_1}(1-\phi)^{N_0}\phi^{\alpha-1}(1-\phi)^{\beta-1}}{\frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)}p(\mathrm{X})} = \frac{\phi^{N_1+\alpha-1}(1-\phi)^{N_0+\beta-1}}{Z} \end{aligned}$$

Hence,

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

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.

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,

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

lower values correspond to models to be preferred.

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

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

Language models

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

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.

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

- Let $\mathcal{D} = \{t_1, \ldots, t_n\}$ be the dictionary, that is 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.

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 $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \dots, \alpha_n)$ that is,

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

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

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

  that is,

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

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

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

that is, the expectation of ϕ_j w.r.t. the distribution $\text{Dir}(\boldsymbol{\phi}|\overline{\boldsymbol{\alpha}})$. Then,

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

The α_j term makes it impossible to obtain zero probabilities (**Dirichlet smoothing**).

The non informative prior here is $\alpha_i = \alpha$ for all i, which results into

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

where $|\mathcal{D}|$ is the vocabulary size.

Naive bayes classifiers

A language model, which is generative, can be applied to derive document classifiers into two or more classes as described above:

- 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 (for example) 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? We 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 from the collections

- we have still the problem of computing $p(C_k)$ and $p(d|C_k)$ from the collections $\mathcal{C}_1$ and $\mathcal{C}_2$

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

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

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

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

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 $i, j = 1. \ldots, n_d$ 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,

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

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 $\mathcal{C}_2$.

By applying these considerations, we obtain a Naive Bayes classifier which behaves as follows, in order to classify a document d:

1. let $\bar{t}_1, \ldots, \bar{t}_m$ be the bag of words representaion of d, where $m = |\mathcal{D}|$
2. for each $i = 1, \ldots, m$ and $k = 0, 1$ compute $p(\bar{t}_i|C_k)$
3. for $k = 0, 1$ compute, by applying the naive Bayes assumption, $p(d|C_k)$
4. assign d to class $\mathcal{C}_r$ where $r = \underset{k \in \{0,1\}}{\operatorname{argmax}}\, p(d|C_k)p(C_k)$

Observe that the same approach can be applied to the classification of items $\mathbf{x} = (x_0, \ldots, x_d)$. In this case, the Naive Bayes assumption is that features are conditionally independent given the class, hence $p(\mathbf{x}|C_k) = \prod_{i=0}^{d} p(x_i|C_k)$.

Feature selection by mutual information

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.

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.

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 ϕ_{jk} is the estimated probability of t_j in documents of class C_k and π_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.

*As done before, these considerations can be extended to any set of features. Just consider the number of occurrences of a term as a particular type of feature.