Introduction

Course of Machine Learning
Master Degree in Computer Science
University of Rome "Tor Vergata"
a.a. 2019-2020

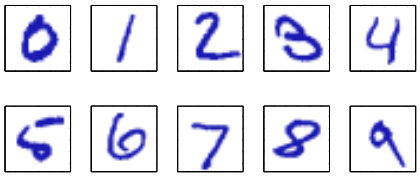
Giorgio Gambosi

Pattern recognition
Automatic identification of shared commonalities in a collection of data by means of algorithms; use of such commonalities as guides to actions to perform.

Particular cases:

- item classification into a set of categories
- identification of associations among items and among values
- derivation of preference functions
- identification of sets of "similar" items
- identification of the most informative features for item characterization
- . . .

An example
Automatic recognition of handwritten text



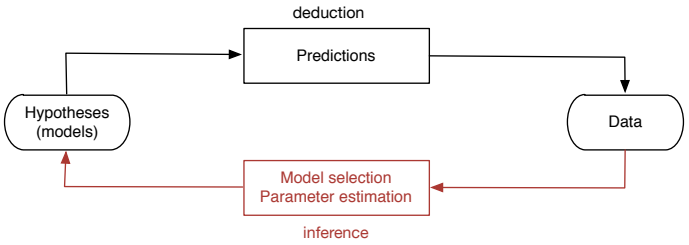
A possibile solution: deductive approach

- Identification, through problem analysis, of the most relevant features for character discrimination;
- definition of a set of rules to be applied for character recognition;
- implementation of those rules in a program.

Hard to apply: the problem is not easy to formalize; hard to identify all necessary rules and express them algorithmically; need of domain experts

Machine learning: inductive approach
Learning of commonalities through analysis of a set of examples (*training set*), which is assumed to be available.

- A training set of n items is represented as a set of input vectors $\mathbf{x}_1, \ldots, \mathbf{x}_n$, used to derive a *model*.
- If the purpose is item classification with respect to a collection of predefined classes, the training set also includes a *target* vector $\mathbf{t} = \{t_1, \ldots, t_n\}$, where the class of each training set item is specified.



Supervised learning

- We want to predict, given the values of a set (*features*) of an item $\mathbf{x}$, the unknown value of an additional feature *target* of the item
  - Target in **R**: *regression*. Target in $\{1, \ldots, K\}$: *classification*.
- General approach: defined (by means of learning from a set of examples) a *model* of the relation between feature and target values.
- The training set $\mathbf{X}, \mathbf{t}$ includes a feature vector $\mathbf{x}_i = \{x_{i1}, \ldots, x_{im}\}$ and the corresponding target t_i for each item.
- The model could be:
  1. a function $y()$ which, for any item $\mathbf{x}$, returns a value $y(\mathbf{x})$ as an estimate of t
  2. a probability distribution which associates to each possible value $\overline{y}$ in the target domain, the corresponding probability $p(y = \overline{y}|\mathbf{x})$

Unsupervised learning

- We wish to extract, froma a given collection of items *dataset*) $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$, with no target associated, some synthetic information, such as:
  - subsets of similar items *clustering*)
  - the distribution of items in their domain (*density estimation*)
  - the projection, as informative as possible, of items on lower dimensional subspaces, that is, their characterization by means of a smaller set of features *feature selection*, *feature extraction*)
- A suitable *model*, of just the data features, is usually defined and applied also in the case of unsupervised learning.

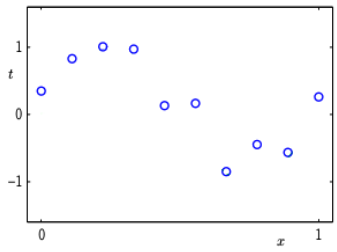
Reinforcement learning

- We want to identify, in a given framework, a sequence of actions to be performed in order to maximize a certain profit
- As in supervised learning, no examples are given, but an environment is available which returns a profit in correspondance to the execution of any action

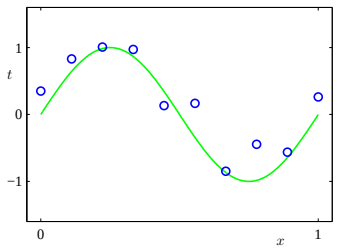
Problem

- A set of n observations of two variables $x, t \in \mathbb{R}$: $(x_1, t_1), \ldots, (x_n, t_n))$ is available. We wish to exploit these observations to predict, for any value $\tilde{x}$ of x, the corresponding unknown value of the target variable t
- The training set is a pair of vectors $\mathbf{x} = (x_1, \ldots, x_n)^T$ and $\mathbf{t} = (t_1, \ldots, t_n)^T$, related through an unknown rule (function)

Example of a training set.



Training set
In this case, we assume that the (unknown) relation between x and t in the training set is provided by the function $t = \sin(2\pi x)$, with an additional gaussian noise with mean 0 and given variance σ^2.



Hence, $t_i = \sin(2\pi x_i) + \varepsilon_i$, with $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$.

Purpose
Guessing, or approximating as well as possible, the deterministic relation $t = \sin(2\pi x)$, on the basis of the analysis of data in the training set.

Approach
Let us approximate the unknown function through a suitable polynomial of given degree $M > 0$

$$y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \ldots + w_M x_M = \sum_{j=0}^{M} w_j x^j$$

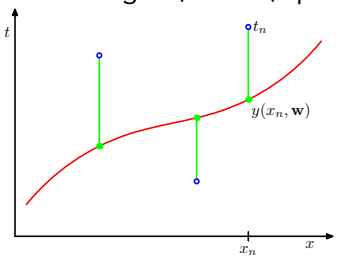
whose coefficients $\mathbf{w} = (w_0, w_1, \ldots, w_M)^T$ are to be computed.

Linear models
$y(x, \mathbf{w})$ is a nonlinear function of x, but is a linear function (model) of $\mathbf{w}$.

Parameter estimation
The values assigned to coefficients should minimize some *error function* (a.k.a. *cost function*), when applied to data in the training set (then, to $\mathbf{x}$, $\mathbf{t}$ and $\mathbf{w}$).

Least squares

A most widely adopted error function is *least quares*, i.e. the sum, for all items in the training set, of the (squared) difference between the value returned by the model and the target value.



$$E(\mathbf{w}) = \frac{1}{2}\sum_{i=1}^{n}\left(y(x_i,\mathbf{w}) - t_i\right)^2 = \frac{1}{2}\sum_{i=1}^{n}\left(w_0 + w_1 x_i + w_2 x_i^2 + \ldots + w_M x_i^M - t_i\right)^2$$

Error minimization

- To minimize $E(\mathbf{w})$, set its derivative w.r.t. $\mathbf{w}$ to $\mathbf{0}$
- $E(\mathbf{w})$ quadratic implies that its derivative is linear, hence that it is zero in one point w^*
- The resulting function is $y(x, \mathbf{w}^*)$

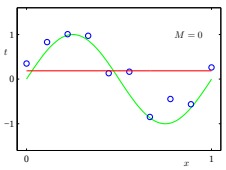
Derivative with respect to $\mathbf{w}$
The derivative w.r.t. $\mathbf{w}$ is indeed a collection of derivatives. A linear system is then obtained:

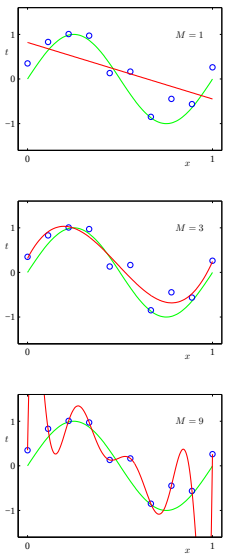
$$\begin{aligned}
\frac{\partial E(\mathbf{w})}{\partial w_0} &= \sum_{i=1}^{n}\left(y(x_i,\mathbf{w}) - t_i\right) = 0\\
\frac{\partial E(\mathbf{w})}{\partial w_1} &= \sum_{i=1}^{n} x_i\left(y(x_i,\mathbf{w}) - t_i\right) = 0\\
&\ldots\\
\frac{\partial E(\mathbf{w})}{\partial w_M} &= \sum_{i=1}^{n} x_i^M\left(y(x_i,\mathbf{w}) - t_i\right) = 0
\end{aligned}$$

Each of the $M+1$ equations is linear w.r.t. each coefficient in $\mathbf{w}$. A linear system results, with $M+1$ equations and $M+1$ unknowns, which, in general and with the exceptions of degenerate cases, has precisely one solution.

Polynomial degree

- Example of *model selection*: assigning a value to M determines the model to be used, the choice of M implies the number of coefficients to be estimated
- increasing M allows to better approximate the training set items, decreasing the error
- if $M+1 = n$ the model allows to obtain a null error (*overfitting*)





Overfitting

- The function $y(x, \mathbf{w})$ is derived from items in the training set, but should provide good predictions for other items.
- It should provide a suitable generalization to all items in the whole domain.
- If $y(x, \mathbf{w})$ is derived as a too much accurate depiction of the training set, it results into an unsuitable generalization to items not in the training set

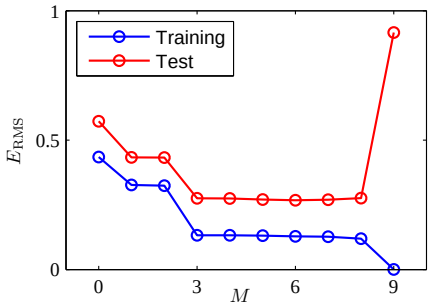
Evaluation of the generalization

- Test set $\mathbf{X}_{test}$ of 100 new items, generated by uniformly sampling x in $[0, 1,]$ and ε from $\mathcal{N}(0, \sigma^2)$, and computing $t = \sin 2\pi x + \varepsilon$
- For each M:
  - derives $\mathbf{w}^*$ from the training set $\mathbf{X}_{train}$
  - compute the error $E(\mathbf{w}^*, \mathbf{X}_{test})$ on the test set, or the square root of its mean

$$E_{RMS}(\mathbf{w}^*, \mathbf{X}_{test}) = \sqrt{\frac{E(\mathbf{w}^*, \mathbf{X}_{test})}{|\mathbf{X}_{test}|}} = \sqrt{\frac{1}{2|\mathbf{X}_{test}|} \sum_{x \in \mathbf{X}_{test}} (y(x, \mathbf{w}) - t)^2}$$

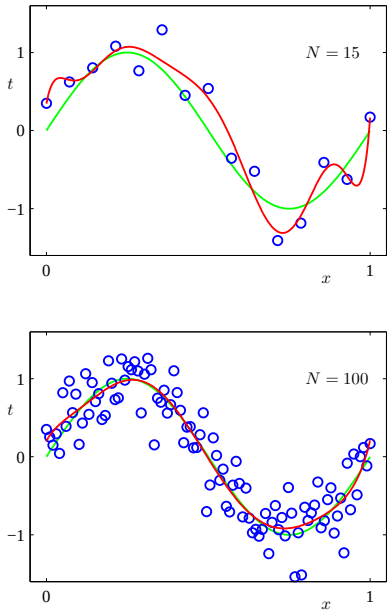
- a lower value of $E_{RMS}(\mathbf{w}^*, \mathbf{X}_{test})$ denotes a good generalization

Plot of E_{RMS} w.r.t. M, on the training set and on the test set.



- As M increases, the error on the training set tends to 0.
- On the test set, the error initially decreases, since the higher complexity of the model allows to better represent the characteristics of the data set. Next, the error increases, since the model becomes too dependent from the training set: the noise component in t is too represented.

For a given model complexity (such as the degree in our example), overfitting decreases as the dimension of the dataset increases.



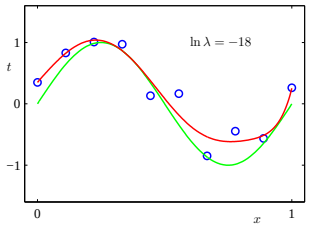
The larger the dataset, the higher the acceptable complexity of the model.

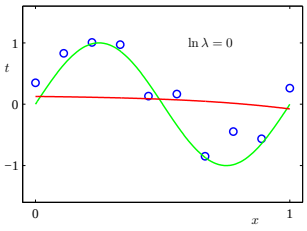
Use of *regularization* to limit complexity and overfitting.

- inclusion of a penalty term in the error function
- purpose: limiting the possible values of coefficients
- usually: limiting the modulus of the coefficient values

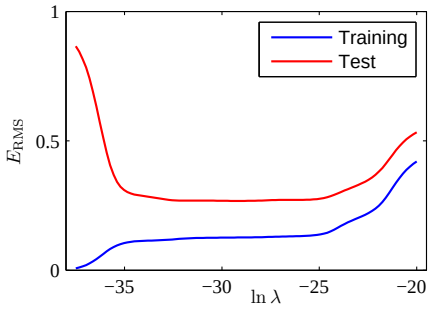
$$\tilde{E}(\mathbf{w}) = \frac{1}{2}\sum_{i=1}^{n}(y(x_i,\mathbf{w}) - t_i)^2 + \frac{\lambda}{2}\sum_{k=0}^{M} w_k^2$$
$$= \frac{1}{2}\sum_{i=1}^{n}(y(x_i,\mathbf{w}) - t_i)^2 + \frac{\lambda}{2}||\mathbf{w}||^2$$

Dependance from the value of the hyperparameter λ.





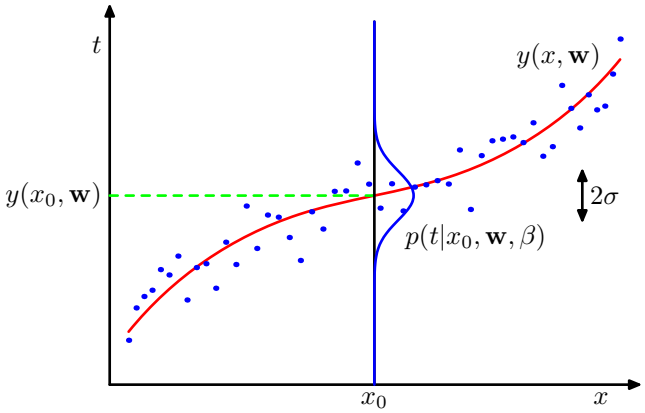
Plot of the error w.r.t λ.



- Small λ: overfitting. Small error on the training set, large error on the test set.
- Large λ: the effect of data values decreases. Large error on both test and training sets.
- Intermediate λ. Intermediate error on training set, small error on test set.

Assume that, given an item $\mathbf{x}$, the corresponding unknown target t is normally distributed around the value returned by the model $\mathbf{w}^T\overline{\mathbf{x}}$, with a given variance $\sigma^2 = \beta^{-1}$:

$$p(t|\mathbf{x}, \mathbf{w}, \beta) = \mathcal{N}(t|y(x, \mathbf{w}), \beta^{-1})$$



An estimate of both β_{ML} and the coefficients $\mathbf{w}_{ML}$ can be performed on the basis of the likelihood w.r.t. the assumed normal distribution:

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

Parameters $\mathbf{w}$ and β can be estimated as the values which maximize the data likelihood, or its logarithm

$$l(\mathbf{t}|\mathbf{X}, \mathbf{w}, \beta) = \log p(\mathbf{t}|\mathbf{X}, \mathbf{w}, \beta) = \sum_{i=1}^{n} \log \mathcal{N}(t_i|y(\mathbf{x}_i, \mathbf{w}), \beta^{-1})$$

which results into

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

The maximization w.r.t. $\mathbf{w}$ is performed by determining a maximum w.r.t. $\mathbf{w}$ of the function

$$-\frac{1}{2}\sum_{i=1}^{n}\left(t_i - y(\mathbf{x}_i,\mathbf{w})\right)^2$$

this is equivalent to minimizing the least squares sum.

The maximization w.r.t. the *precision* β is done by setting to 0 the corresponding derivative

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

which results into

$$\beta_{ML}^{-1} = \frac{1}{n}\sum_{i=1}^{n}\left(t_i - y(\mathbf{x}_i,\mathbf{w})\right)^2$$

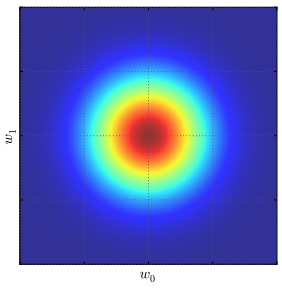
As a side result, the parameter estimate provides a *predictive distribution* of t given $\mathbf{x}$, that is the (gaussian) distribution of the target value for a given item $\mathbf{x}$.

$$p(t|\mathbf{x};\mathbf{w},\beta) = \mathcal{N}(t|y(\mathbf{x},\mathbf{w}),\beta^{-1}) = \sqrt{\frac{\beta_{ML}}{2\pi}} e^{-\frac{\beta_{ML}}{2}(t - y(\mathbf{x},\mathbf{w}_{ML}))^2}$$

- In the maximum likelihood framework parameters are considered as (unknown) values to determine with the best possible precision (*frequentist* approach).
- An alternative framework (*bayesian*) looks at parameters as random variables, whose probability distribution has to be derived.

Prior distribution of parameters: gaussian with mean $\mathbf{0}$ and diagonal covariance matrix with variance equal to the inverse of *hyperparameter* α

$$p(\mathbf{w}|\alpha) = \mathcal{N}(\mathbf{w}|\mathbf{0},\alpha^{-1}\mathbf{I}) = \left(\frac{\alpha}{2\pi}\right)^{\frac{M+1}{2}} e^{-\frac{\alpha}{2}\mathbf{w}^T\mathbf{w}}$$



From Bayes rule, the distribution of $\mathbf{w}$, given the observed training set $(\mathbf{X}, \mathbf{t})$, is proportional to the product of the prior distribution and the likelihood of the training set

$$\begin{aligned} p(\mathbf{w}|\mathbf{X}, \mathbf{t}; \alpha, \beta) &=\propto p(\mathbf{t}|\mathbf{X}, \mathbf{w}; \beta) p(\mathbf{w}|\alpha) \\ &= \prod_{i=1}^{n} \left(\frac{\sqrt{\beta}}{\sqrt{2\pi}} e^{-\frac{\beta}{2}(t_i - y(\mathbf{x}_i, \mathbf{w}))^2} \right) \left(\frac{\alpha}{2\pi} \right)^{\frac{M+1}{2}} e^{-\frac{\alpha}{2}\mathbf{w}^T\mathbf{w}} \end{aligned}$$

Computing the maximum of the posterior distribution (*MAP*): equivalent to the maximization of the corresponding logarithm

$$-\frac{\beta}{2}\sum_{i=1}^{n}(t_i - y(\mathbf{x}_i, \mathbf{w}))^2 + \frac{n}{2}\log\beta - \frac{\alpha}{2}\mathbf{w}^T\mathbf{w} + \frac{M+1}{2}\log\frac{\alpha}{2\pi} + \text{cost}$$

The value $\mathbf{w}_{MAP}$ which maximize the probability (*mode* of the distribution) also minimizes

$$\frac{\beta}{2}\sum_{i=1}^{n}(t_i - y(\mathbf{x}_i, \mathbf{w}))^2 + \frac{\alpha}{2}\mathbf{w}^T\mathbf{w} = \beta\left(\frac{1}{2}\sum_{i=1}^{n}(t_i - y(\mathbf{x}_i, \mathbf{w}))^2 + \frac{\alpha}{2\beta}||\mathbf{w}||^2\right)$$

The ratio $\frac{\alpha}{\beta}$ corresponds to a regularization hyperparameter.

The same considerations of ML appy here for what concerns deriving the *predictive distribution* of t given $\mathbf{x}$, which results now

$$p(t|\mathbf{x}; \mathbf{w}, \beta_{MAP}) = \mathcal{N}(t|y(\mathbf{x}, \mathbf{w}), \beta_{MAP}^{-1}) = \sqrt{\frac{\beta_{MAP}}{2\pi}} e^{-\frac{\beta_{MAP}}{2}(t - y(\mathbf{x}, \mathbf{w}_{MAP}))^2}$$

where, as it is easy to see, $\beta_{MAP} = \beta_{ML}$

A *pure bayesian* approach to this problem aims to derive the predictive distribution of t given $\mathbf{x}$ from the training set $(\mathbf{X}, \mathbf{t})$ directly, without assuming any type of distribution and then performing some kind of evaluation of the corresponding parameters.

From the basic properties of probabilities,

$$\begin{aligned} p(t|\mathbf{x}; \mathbf{X}, \mathbf{t}) &= \int p(t, \mathbf{w}|\mathbf{x}; \mathbf{X}, \mathbf{t}) d\mathbf{w} \\ &= \int p(t|\mathbf{w}, \mathbf{x}; \mathbf{X}, \mathbf{t}) p(\mathbf{w}|\mathbf{x}; \mathbf{X}, \mathbf{t}) d\mathbf{w} \\ &= \int p(t|\mathbf{w}, \mathbf{x}) p(\mathbf{w}|\mathbf{X}, \mathbf{t}) d\mathbf{w} \end{aligned}$$

since $\mathbf{w}$ is independent from $\mathbf{x}$, and, once $\mathbf{w}$ is given, t is independent from $\mathbf{X}$ and $\mathbf{t}$. That is, $\mathbf{x} \perp\!\!\!\perp \mathbf{w}$, $t \perp\!\!\!\perp \mathbf{X}|\mathbf{w}$, $t \perp\!\!\!\perp \mathbf{t}|\mathbf{w}$.

We shall see later that, if we assume that:

$p(t|\mathbf{w}, \mathbf{x}; \beta)$

$p(\mathbf{w}; \alpha)$

$p(\mathbf{t}|\mathbf{w}, \mathbf{X})$

are all gaussians, then the evidence is gaussian too.

$$p(t|\mathbf{x}, \mathbf{X}, \mathbf{t}) = \mathcal{N}(t|m(\mathbf{x}), s^2(\mathbf{x}))$$

with mean and variance deriving from the training set as follows

$$\begin{aligned} m(\mathbf{x}) &= \beta\mathbf{x}^T\mathbf{S}\sum_{i=1}^{n}\mathbf{x}_i\mathbf{t}_i \\ s^2(\mathbf{x}) &= \beta^{-1} + \mathbf{x}^T\mathbf{S}\mathbf{x} \\ \mathbf{S} &= \alpha\mathbf{I} + \beta\sum_{i=1}^{n}\mathbf{x}\mathbf{x}^T \end{aligned}$$

- The mean is clearly a function $m(x)$ of x.
- The variance is also dependent from x. The term β^{-1} is independent from x and represents the uncertainty in the knowledge of $\mathbf{t}$.