

MACHINE LEARNING

Linear, Weighted and Probabilistic Regression

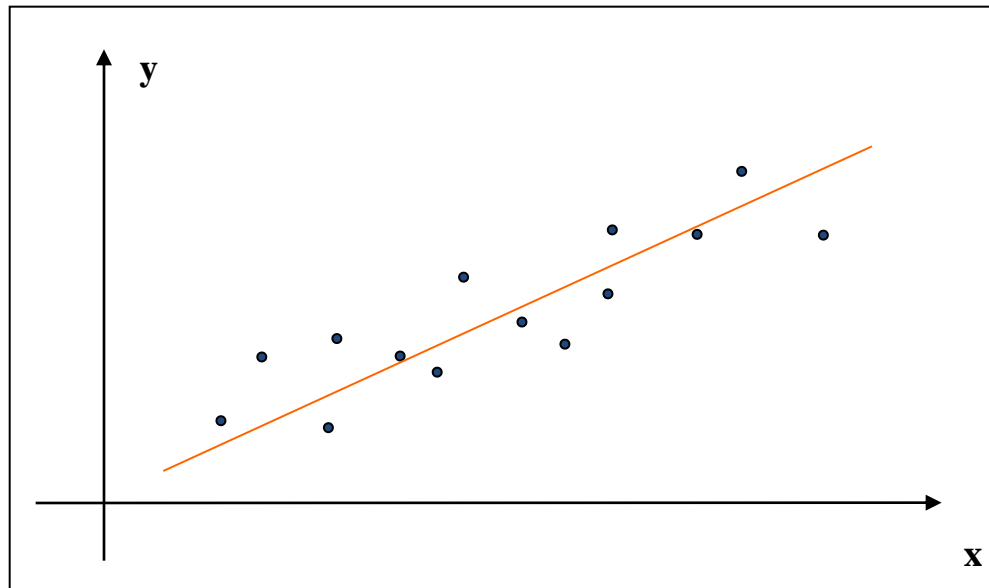


ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Linear Regression

Linear regression searches a linear mapping between input x and output y , parametrized by the slope vector w and *intercept* b .

$$y = f(x; w, b) = w^T x + b$$



Linear Regression

Linear regression searches a linear mapping between input x and output y , parametrized by the slope vector w and *intercept* b .

$$y = f(x; w, b) = w^T x + b$$

One can always omit the intercept by centering the data:

$$y' = y - \bar{y} \text{ and } x' = x - \bar{x}$$

$$y' = w^T x' + b'$$

$$\text{with } b' = b + w^T \bar{x} - \bar{y}$$

$$\text{Least-square estimate of } (b')^* = \bar{y} - w^T \bar{x} = 0$$

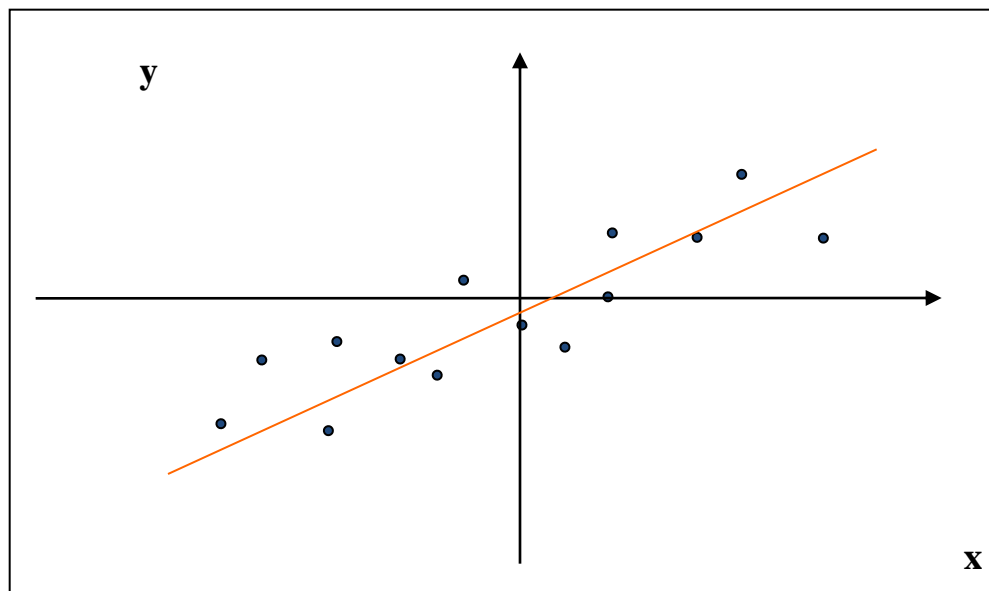
$$\Rightarrow y' = w^T x'$$

In the rest of these notes, we consider the problem when the data has been centered.

Linear Regression

Linear regression searches a linear mapping between input x and output y , parametrized by the slope vector w .

$$y = f(x; w) = w^T x$$



Linear Regression

Pair of M training points $X = [x^1 \ x^2 \ \dots \ x^M]$ and $y = [y^1 \ y^2 \ \dots \ y^M]$

$x^i \in \mathbb{R}^N$, $y^i \in \mathbb{R}$.

Find the optimal parameter w through *least-square regression*:

$$w^* = \min_w \left(\sum_{i=1}^M \frac{1}{2} (w^T x^i - y^i)^2 \right)$$

Finds an analytical solution through partial differentiation:

$$w^* = (X^T X)^{-1} X^T y$$

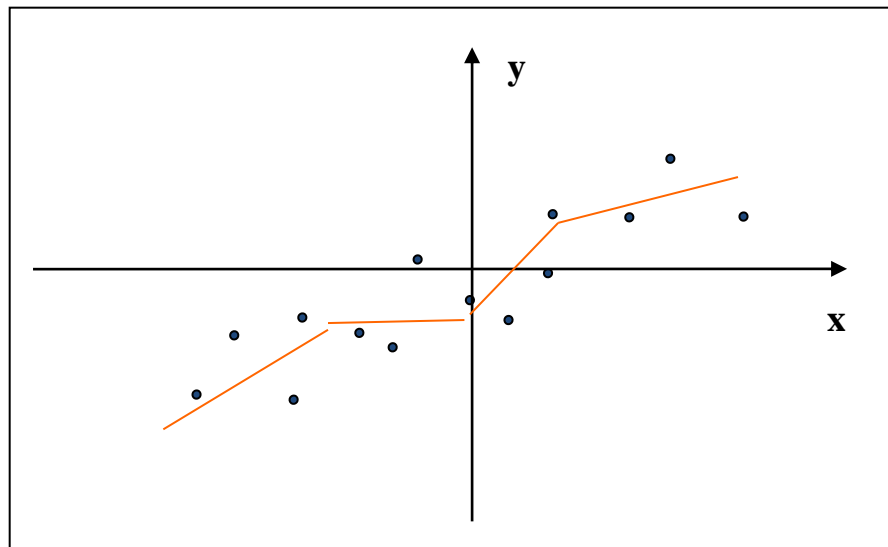
Limitations of classical linear regression

Regression through Least Square

$$w^* = \min_w \left(\sum_{i=1}^M \frac{1}{2} (w^T x^i - y^i)^2 \right)$$

Assumes that **a single linear dependency** applies everywhere.

Not true for data sets with **local dependencies**.



Limitations of classical linear regression

Regression through Least Square

$$w^* = \min_w \left(\sum_{i=1}^M \frac{1}{2} \left(w^T x^i - y^i \right)^2 \right)$$

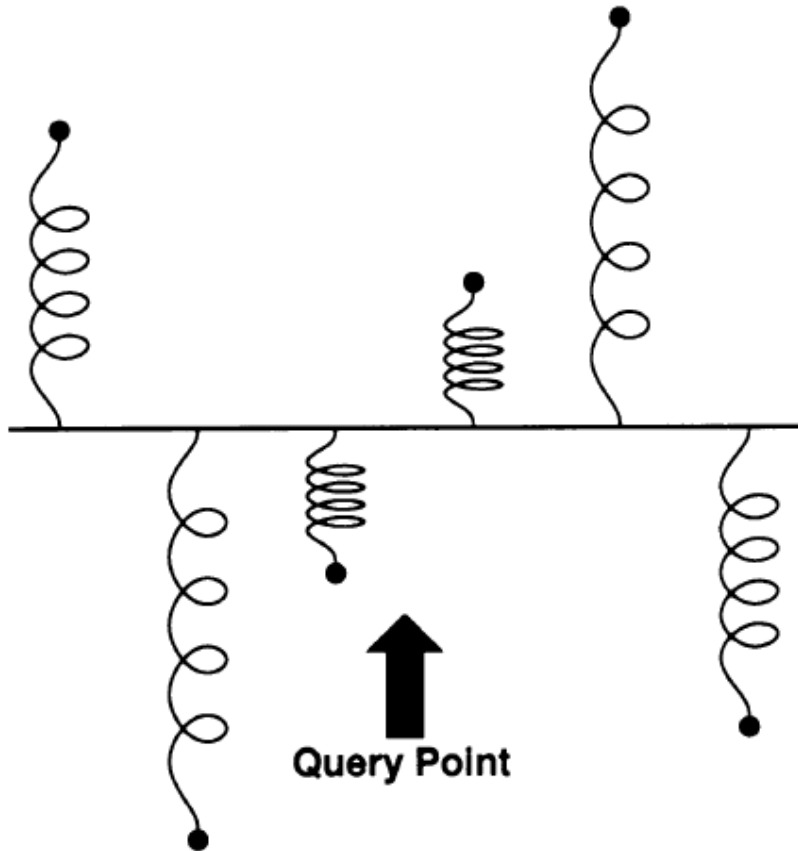
Assumes that **a single linear dependency** applies everywhere.

Not true for data sets with **local dependencies**.

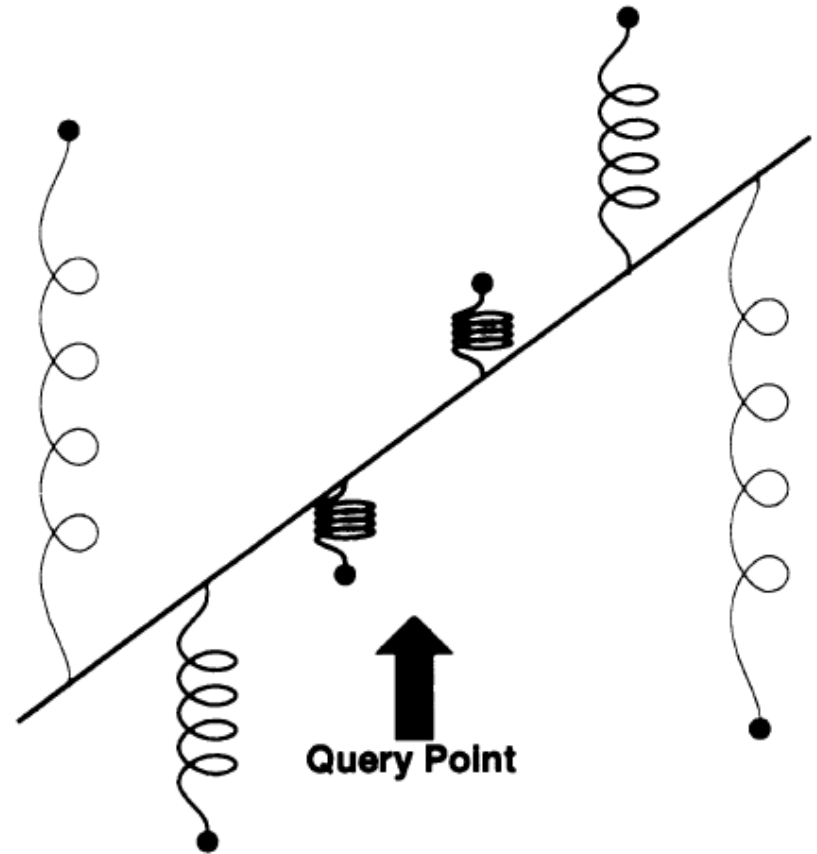
→ It would be useful to design a regression method that estimates best the linear dependencies locally.

→ **Weighted** least-square

Unweighted vs Weighted Regression



Unweighted Regression

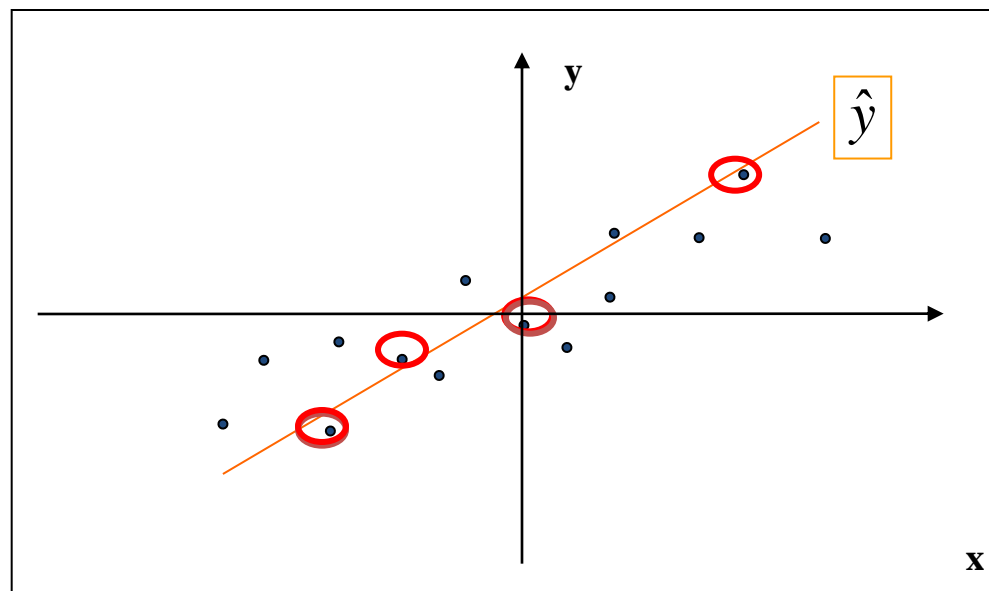


Weighted Regression

Weighted Regression

Regression through **weighted** Least Square

$$w^* = \min_w \left(\sum_{i=1}^M \frac{1}{2} \beta_i (w^T x^i - y^i)^2 \right), \quad \beta_i \in \mathbb{R} : \text{constant weights}$$



Points in red have large *error* weight

Weighted Regression

Regression through **weighted** Least Square

$$w^* = \min_w \left(\sum_{i=1}^M \frac{1}{2} \beta_i \left(w^T x^i - y^i \right)^2 \right), \quad \beta_i \in \mathbb{R} : \text{constant weights}$$

Weight datapoints according to how close they are to the query point.

$$\min_w J(w) = \min_w \sum_{i=1}^M \left(y^i - w^T x^i \right)^2 K(d(x^i, x))$$

where $K(\cdot)$ is the weighting or kernel function and $d(x^i, x)$ is the distance between the data point x^i and the query point x .

Weighted Regression

The cost function at each query point x becomes a local model with a different set of parameters.

Weighted regression:

Data set is tailored to the query point x by emphasizing nearby points in the regression. One can do this by weighting the training criterion

$$\min_w J(w) = \min_w \sum_{i=1}^M \left(y^i - w^T x^i \right)^2 K(d(x^i, x))$$

where $K(\cdot)$ is the weighting or kernel function and $d(x^i, x)$ is the distance between the data point x^i and the query point x .

Weighted Regression

Assuming a set of weights β_i for all datapoints, we set B a diagonal matrix

with entries β_i , $B = \begin{bmatrix} \beta_1 & & & \\ & \beta_2 & & \\ & & \dots & \\ & & & \dots \dots \dots \beta_N \end{bmatrix}$

Change of variable: $Z = BX$ and $v = By$.

Minimizing for MSE, one gets an estimator for y at the query point:

$$\hat{y}(x) = x^T (Z^T Z)^{-1} Z^T v$$

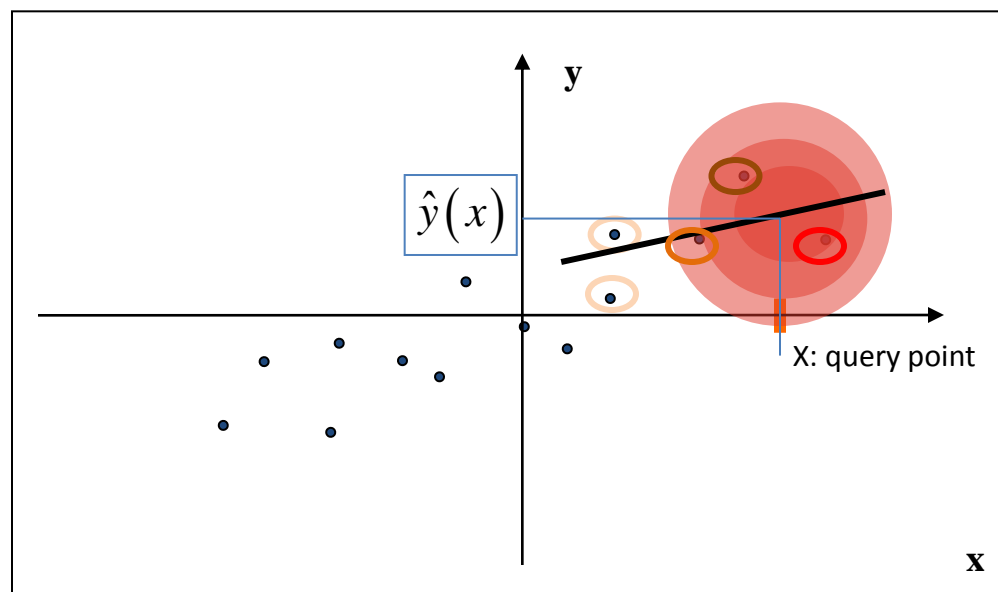
Contrast to the solution for unweighted linear regression

$$w^* = (X^T X)^{-1} X^T y$$

Locally weighted learning

Determining the set of weights determines the local influence of each group of datapoints

$$\beta_i(x) = \sqrt{K(d(x^i, x))}, \quad \text{with } K(d(x^i, x)) = e^{-d(x^i, x)}, \quad d(x^i, x) = \|x^i - x\|.$$



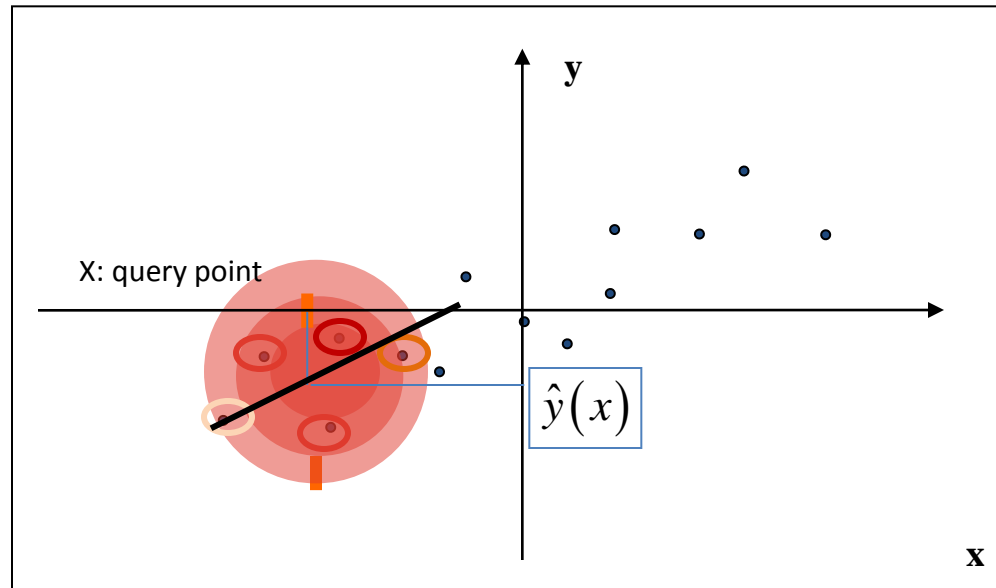
Weighting Kernel:



Locally weighted learning

Determining the set of weights determines the local influence of each group of datapoints

$$\beta_i(x) = \sqrt{K(d(x^i, x))}, \quad \text{with } K(d(x^i, x)) = e^{-d(x^i, x)}, \quad d(x^i, x) = \|x^i - x\|.$$

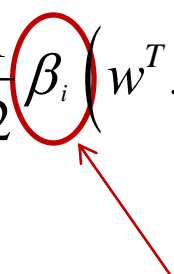


Weighting Kernel:



Other approaches not covered in this lecture

Regression through **weighted** Least Square

$$w^* = \min_w \left(\sum_{i=1}^M \frac{1}{2} \beta_i \left(w^T x^i - y^i \right)^2 \right), \quad \beta_i \in \mathbb{R} : \text{constant weights}$$


How to determine these weights automatically?

→ Locally weighted learning (see review by Atkeson et al, 1997)

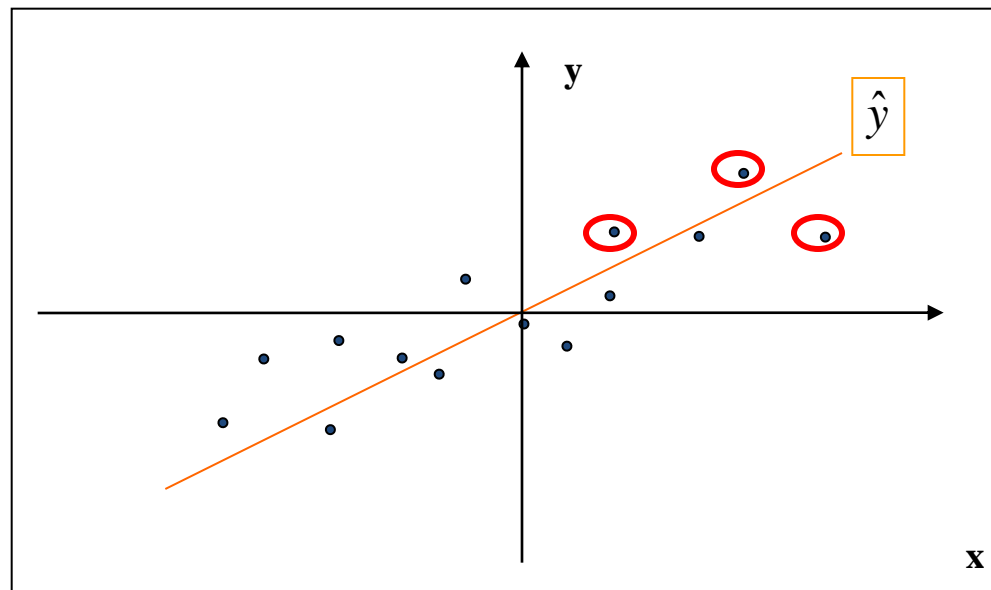
→ Locally weight projected regression (LWPR, Vijayakumar et al 2005)
available in mldemos

Estimating from sampling the datapoints

Regression through Least Square

$$w^* = \min_w \left(\sum_{i=1}^M \frac{1}{2} (w^T x^i - y^i)^2 \right)$$

Sampling

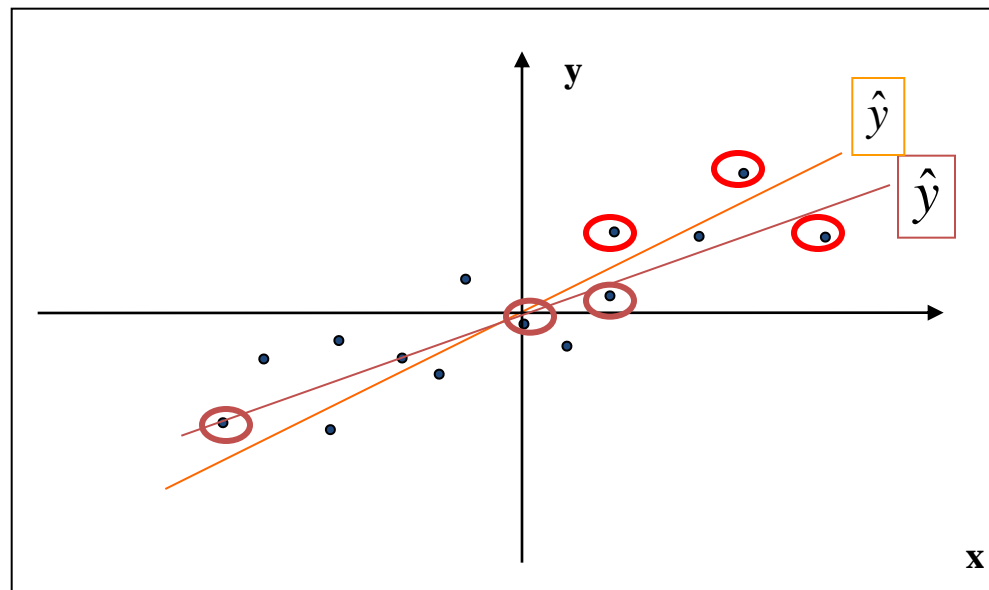


Estimating from sampling the datapoints

Regression through Least Square

$$w^* = \min_w \left(\sum_{i=1}^M \frac{1}{2} (w^T x^i - y^i)^2 \right)$$

Sampling

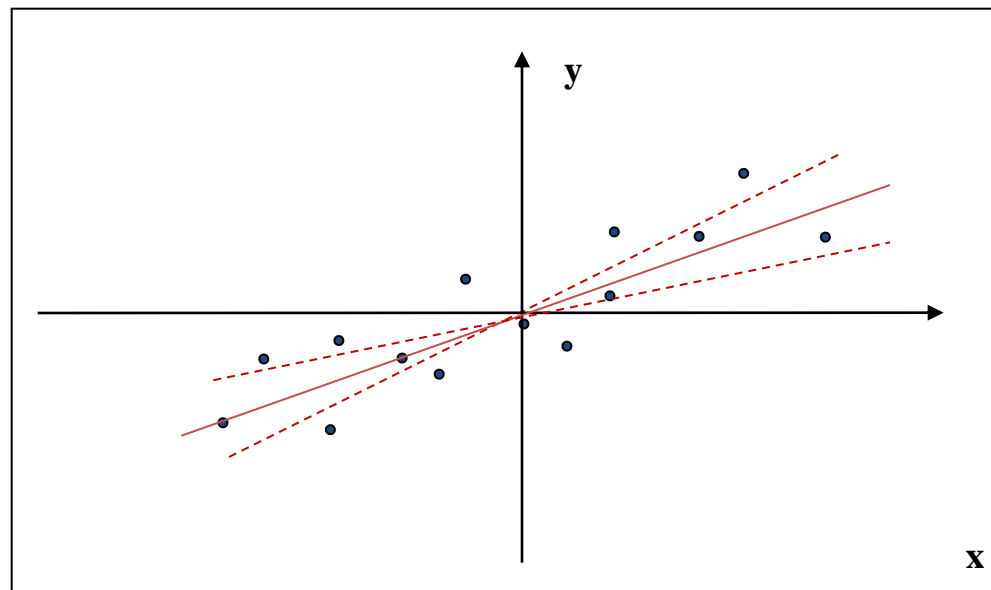


The choice of training data (training set) is crucial → **Crossvalidation**

Estimating from sampling the datapoints

Assume a distribution of possible w and searches for the optimal w^* through maximum-likelihood \rightarrow probabilistic regression

Sampling



Probabilistic Regression

Statistical approach to classical linear regression: estimate the relationship between zero-mean variables y and x by building a linear model of the form:

$$y = f(x, w) = w^T x, \quad w, x \in \mathbb{R}^N$$

If one assumes that the observed values of y differ from $f(x)$ by an additive noise ε that follows a zero-mean Gaussian distribution (such an assumption consists of putting a *prior distribution* over the noise), then:

$$y = f(x, w) + \varepsilon = w^T x + \varepsilon, \quad \text{with } \varepsilon = N(0, \sigma^2)$$

Probabilistic Regression

Training set of M pairs of data points $\{X, y\} = \{x^i, y^i\}_{i=1}^M$

Each pair is independently and identically distributed (i.i.d) according to a Gaussian distribution.

The likelihood of the regressive model $y = w^T X + N(0, \sigma^2)$ is given by computing the probability density of each training pair given the parameters of the model (w, σ) :

$$p(y | X, w, \sigma) = \prod_{i=1}^M p(y^i | x^i, w, \sigma) = \prod_{i=1}^M \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y^i - w^T x^i)^2}{2\sigma^2}\right)$$

Probabilistic Regression

For a query point x , one then computes the expected response of the probabilistic model

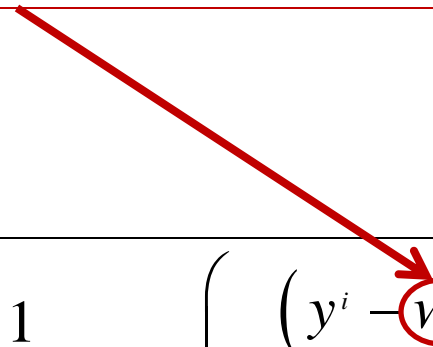
$$\hat{y} = E \{ p(y | X, w, \sigma, x) \}$$



$$p(y | X, w, \sigma) = \prod_{i=1}^M p(y^i | x^i, w, \sigma) = \prod_{i=1}^M \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{(y^i - w^T x^i)^2}{2\sigma^2}\right)$$

Probabilistic Regression

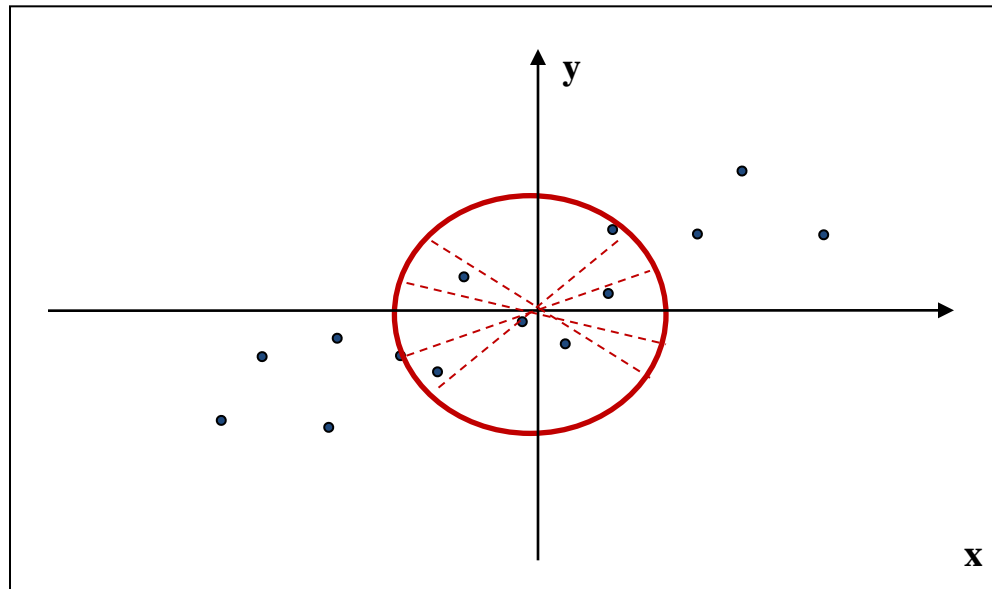
Need to determine the open parameter w so as to maximize the likelihood of the model under the choice of parameters:

$$p(y | X, w, \sigma) = \prod_{i=1}^M p(y^i | x^i, w, \sigma) = \prod_{i=1}^M \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{(y^i - w^T x^i)^2}{2\sigma^2}\right)$$


Probabilistic Regression

In *Bayesian formalism*, one starts by specifying a *prior* over the parameter w . Typical prior is to assume a *zero mean Gaussian* prior with fixed covariance matrix:

$$p(w) = N(0, \Sigma_w) = \exp\left(-\frac{1}{2} w^T \Sigma_w^{-1} w\right)$$



Isotropic distribution
of w .

Probabilistic Regression

In *Bayesian formalism*, one starts by specifying a *prior* over the parameter w . Typical prior is to assume a *zero mean Gaussian* prior with fixed covariance matrix:

$$p(w) = N(0, \Sigma_w) = \exp\left(-\frac{1}{2} w^T \Sigma_w^{-1} w\right)$$

One can then compute the *posterior distribution* over the parameter w using Bayes' Theorem:

$$\text{posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{marginal likelihood}}, \quad p(w | y, X) = \frac{p(y | X, w) p(w)}{p(y | X)}$$

Probabilistic Regression

The posterior distribution of the weight is a **Gaussian distribution**

$$p(w | X, y) \propto \exp\left(-\frac{1}{2}(w - v)^T \Sigma_v^{-1} (w - v)\right)$$

$$v = \sigma^{-2} \left(\sigma^{-2} XX^T + \Sigma_w^{-1}\right)^{-1} Xy$$

$$\Sigma_v = \left(\sigma^{-2} XX^T + \Sigma_w^{-1}\right)^{-1}$$

Probabilistic Regression

Computing the expectation over the posterior distribution gives us the best estimate over the weight:

$$w^* = E \{ p(w | y, X) \} = \sigma^{-2} \left(\sigma^{-2} X X^T + \Sigma_w^{-1} \right)^{-1} X y.$$

This is called the *maximum a posteriori (MAP)* estimate of w .

Computation grows quadratically
with number of datapoints

Probabilistic Regression


Probabilistic Regressive Model is finally given by:

$$\begin{aligned}
 p(y | x, X, y) &= \int p(y | x, w) p(w | X, y) dw \\
 &= N\left(\frac{1}{\sigma^2} x^T A^{-1} X y, x^T A^{-1} x\right)
 \end{aligned}$$

Query point



$$\text{with } A^{-1} = \frac{1}{\sigma^2} X X^T + \Sigma_w^{-1}$$

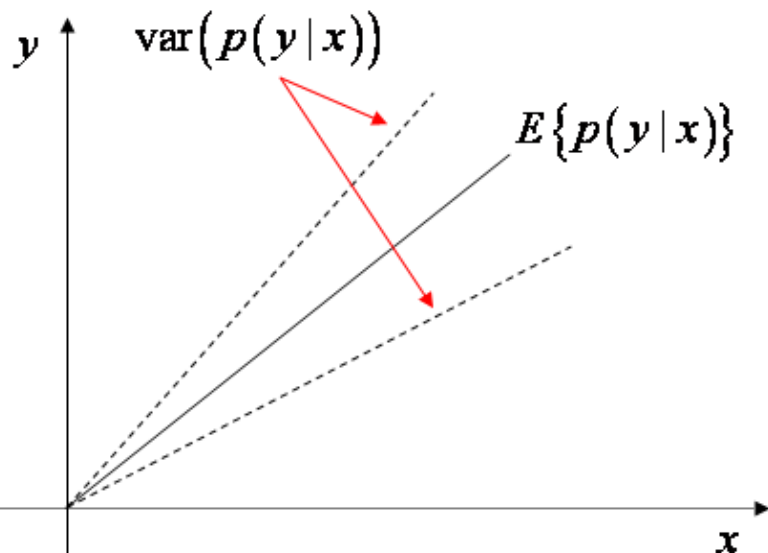


Variance grows quadratically with value of query point.

Probabilistic Regression

Probabilistic Regressive Model is finally given by:

$$\begin{aligned}
 p(y | x, X, y) &= \int p(y | x, w) p(w | X, y) dw \\
 &= N\left(\frac{1}{\sigma^2} x^T A^{-1} X y, x^T A^{-1} x\right)
 \end{aligned}$$



Variance grows quadratically with value of query point.

Least Squares as Maximum Likelihood Estimator

Probabilistic regression: $\hat{y} = E \{ p(y | w, x^i) \}$

Least-square estimate:

$$\min \sum_{i=1}^M \left(y^i - E \{ p(y | w, x^i) \} \right)^2$$

Assume:

i) all pairs $\{x^i, y^i\}$ are i.i.d.

ii) error between each pair of estimated and training data points follows normal distribution:

$$\Delta y^i = y^i - \hat{y}^i \sim N(y^i, \sigma_i)$$

Prob of data given choice of w : $P(X | w) \sim \prod_{i=1}^M e^{-\frac{1}{2} \left(\frac{y^i - \hat{y}^i}{\sigma_i} \right)^2}$ Δy

Using Bayes: $P(w | X) = P(X | w) P(w)$

Assume non-informative prior $p(w)=cst$
 Taking log-likelihood yields msq estimate

Summary

Linear regression can be solved through Least-Mean-Square estimation and yields an optimal analytical solution.

Weighted regression offers the possibility to perform a local regression and yields also an optimal analytical solution. The estimate is no longer global and is computed around each group of data point!

Equivalence Linear and Probabilistic Regression:

The solution through least mean square is equivalent to the solution found through maximum-likelihood assuming that the noise on our estimate follows a zero mean Gaussian distribution.