Computational learning and discovery

CSI 873 / MATH 689

Instructor: I. Griva

Wednesday 7:20 - 10 pm

Regression

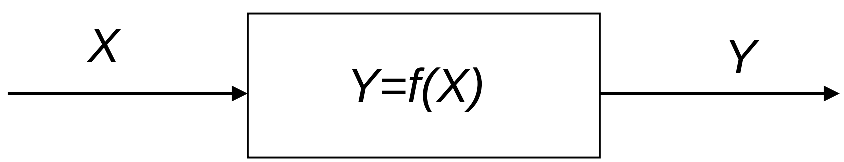
Given a set of training data

$$(x_1, y_1), \ldots, (x_l, y_l),\; x_i \in \Re^n,\; y_i \in \Re^1$$

find a function that can estimate

$$y_j^* \in \Re^1 \text{ given new } x_j^* \in \Re^n$$

and minimize the future error.



Regression

Two principle design questions:

1. How to build a black box.
2. How to measure the empirical risk.

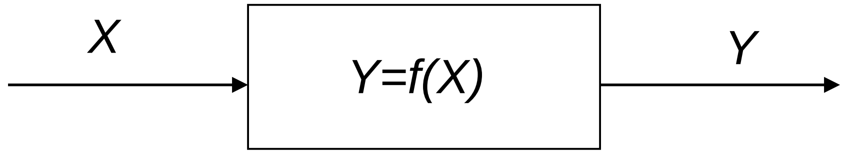
Lost function - measuring empirical risk

1. The least squares.
2. Least modulus.
3. Huber.
4. ε-insensitive loss.

Models

1. Linear.
2. Nonlinear model, but can be handled by a linear regression.
3. Nonlinear model, handled by nonlinear regression.
4. k - nearest neighbor, local regression (linear, nonlinear)
5. Artificial neural networks.
6. Support vector regression.

Linear model of a black box



$$y = \alpha_0 + \alpha_1 x_1 + \alpha_2 x_2 + \cdots + \alpha_n x_n$$

Note that y is a linear function of *x* and *α*!

Nonlinear model of a black box

Examples:

Polynomial:

e.g. quadratic:

$$y = \beta_{11}x_1^2 + \beta_{12}x_1x_2 + \cdots + \beta_{1n}x_1x_n + \beta_{22}x_2^2 + \beta_{23}x_2x_3 + \cdots + \beta_{nn}x_n^2 + \alpha_0 + \alpha_1x_1 + \alpha_2x_2 + \cdots + \alpha_nx_n$$

cubic, etc.

Note that y is a nonlinear function of *x*, but linear of *α*!

Kernel functions:

$$y = \alpha_0 + \sum_i \alpha_i K(x_i, x)$$

Note that y is a linear function of *α*!

Linear Least Squares

$$\min_{\alpha \in \Re^n} f(\alpha) = \min_{\alpha \in \Re^n} \left\| X\alpha - b \right\|_2^2 = \min_{\alpha \in \Re^n} \left\langle X\alpha - b, X\alpha - b \right\rangle = \min_{\alpha \in \Re^n} (X\alpha - b)^T (X\alpha - b)$$

Various optimization methods can be used, but if *X* has a full rank, then *α* can be found by solving the normal linear system.

$$\nabla f(\alpha) = 2X^T (X\alpha - b) = 0 \Rightarrow$$

$$(X^T X)\alpha = X^T b$$ **Normal system of equations**

If X has a full rank, then $X^T X$ is nonsingular.

Linear Least Max Modulus

$$\min_{\alpha\in\Re^n} f(\alpha) = \min_{\alpha\in\Re^n}\|X\alpha - b\|_\infty = \min_{\alpha\in\Re^n}\max_{1\le i\le l} | x_i^T\alpha - b_i |$$

Linear Programming can be used

$$\min_{\alpha\in\Re^n, y\in\Re^1} y$$

s.t. $-y \le x_i^T\alpha - b_i \le y,\ \ i = 1,\dots,l$

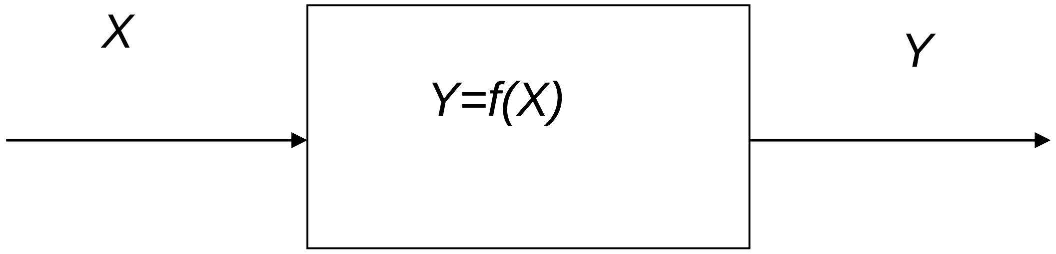
Linear Least Sum of Modula

$$\min_{\alpha\in\Re^n} f(\alpha) = \min_{\alpha\in\Re^n}\|X\alpha - b\|_1 = \min_{\alpha\in\Re^n}\sum_{i=1}^{l} | x_i^T\alpha - b_i |$$

$$\min_{\alpha\in\Re^n, y\in\Re^l} \sum_{i=1}^{l} y_i$$

s.t. $-y_i \le x_i^T\alpha - b_i \le y_i,\ \ i = 1,\dots,l$

Nonlinear Least Squares



$$(x_1, y_1), \ldots, (x_l, y_l),\; x_i \in \Re^n,\; y_i \in \Re^1$$

$$y = f(x, \alpha),\; f \text{ is a nonlinear function of } \alpha$$

Levenberg-Marquardt Method is used to minimize:

$$\|r(\alpha)\|_2^2 = \sum_{i=1}^{l} (f(x_i, \alpha) - y_i)^2 \to \min_{\alpha}$$

k - nearest neighbor

Nearest neighbor:

- Given query instance x_q, first locate nearest training example x_n, then estimate $\hat{f}(x_q) \leftarrow f(x_n)$

k-Nearest neighbor:

- Given x_q, take vote among its k nearest nbrs (if discrete-valued target function)
- take mean of f values of k nearest nbrs (if real-valued)

$$\hat{f}(x_q) \leftarrow \frac{\sum_{i=1}^{k} f(x_i)}{k}$$

Local Linear or Nonlinear Regression: Least Squares, Least Modulus, etc.

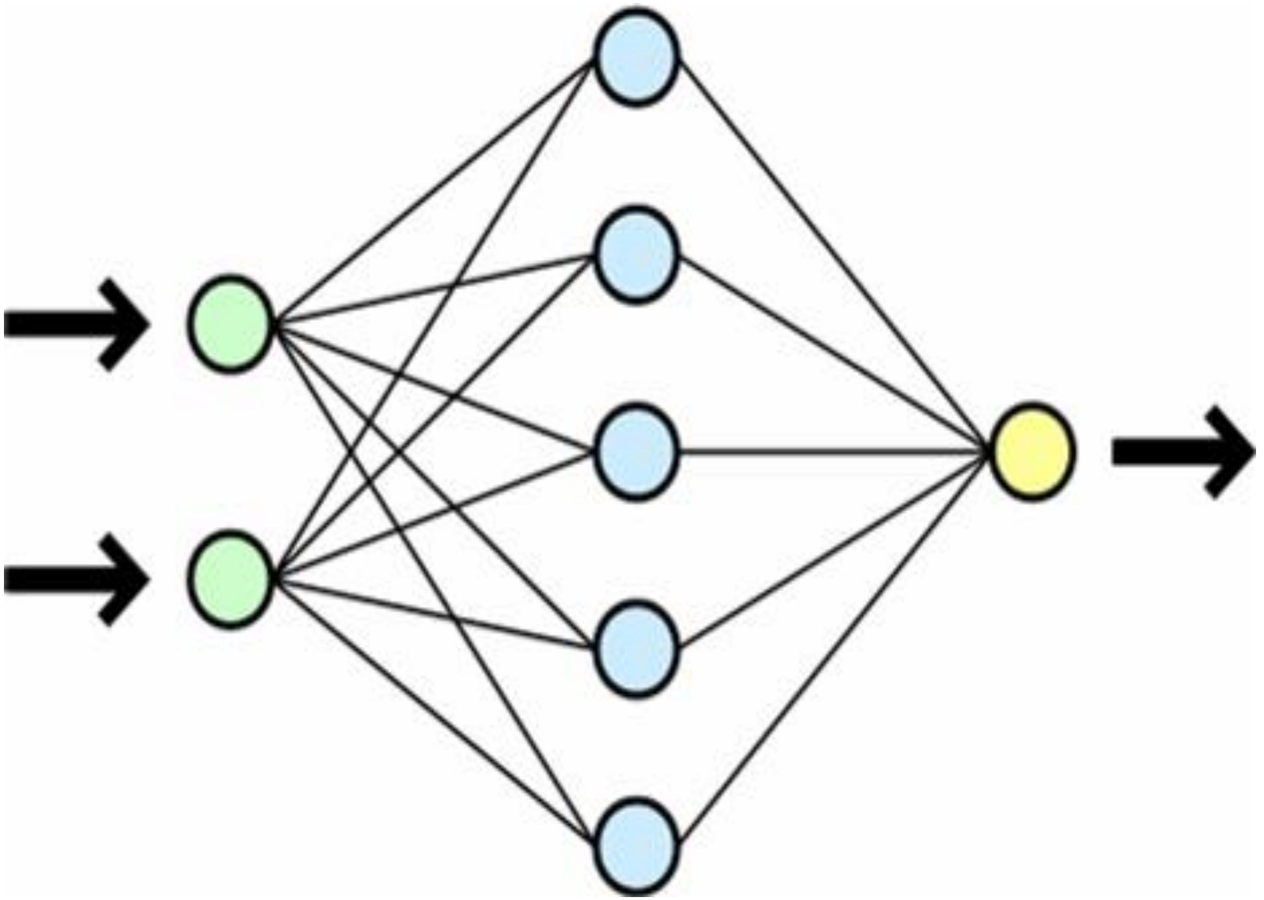
Consider *k* nearest points and run the linear regression on them.

Note that no work is done on the training stage.

All the work is done on the test, but since *k* is small comparing to the size of the all training data, a local linear regression performs quickly.

The time is usually spent to select *k* nearest neighbors, however.

Artificial Neural Networks (ANN)



Support Vector Regression (SVR)

The least squares vs. the least absolute modulus?

If the noise is subject to a normal distribution then the least squares to be used.

$$L = | f(x, \alpha) - y |^2$$

If there is no information on the noise except it is being symmetric, then the best strategy would be the least max modulus (Huber 1964):

$$L = | f(x, \alpha) - y |$$

For a mixture of the normal noise and unknown symmetric noise, Huber suggested

$$L = \begin{cases} 0.5 | f(x, \alpha) - y |^2, \text{if } | f(x, \alpha) - y | \leq c, \\ c | f(x, \alpha) - y | - \dfrac{c^2}{2}, \text{otherwise.} \end{cases}$$

c is defined by the proportion of the mixture.

Support Vector Regression (SVR)

Vapnik suggested ε-insensitive loss functions.

Linear:

$$L = | f(x,\alpha) - y |_{\varepsilon} = \begin{cases} 0, & \text{if } | f(x,\alpha) - y | \le \varepsilon, \\ | f(x,\alpha) - y | - \varepsilon, & \text{otherwise,} \end{cases}$$

Quadratic:

$$L = \begin{cases} 0, & \text{if } | f(x,\alpha) - y | \le \varepsilon, \\ | f(x,\alpha) - y |_{\varepsilon}^{2}, & \text{otherwise,} \end{cases}$$

Note that if $\varepsilon = 0$, then the linear ε-insensitive LF becomes the absolute modulus LF, while the quadratic ε-insensitive LF becomes the quadratic LF.

Support Vector Regression (SVR)

Using the linear ε-insensitive loss function leads to SVR.

Suppose we want $f(x,\alpha) = \langle w, x \rangle - b$

Then the minimization of the empirical risk (error measure)

$$R_{emp}(w,b) = \frac{1}{l}\sum_{i=1}^{l} |\langle w, x_i \rangle - b - y_i|_{\varepsilon} \rightarrow \min_{w,b}$$

is equivalent to solving

$$\sum_{i=1}^{l}(\xi_i + \xi_i^*) \rightarrow \min_{w,b,\xi,\xi^*}$$

$$\text{s.t.} \quad \begin{aligned} \langle w, x_i \rangle - b - y_i &\le \varepsilon + \xi_i, & i &= 1,\dots,l, \\ \langle w, x_i \rangle - b - y_i &\ge -\varepsilon - \xi_i^*, & i &= 1,\dots,l, \\ \xi_i &\ge 0, & i &= 1,\dots,l, \\ \xi_i^* &\ge 0, & i &= 1,\dots,l. \end{aligned}$$

Support Vector Regression (SVR)

$$0.5\langle w, w\rangle + C\sum_{i=1}^{l}(\xi_i + \xi_i^*) \to \min_{w,b,\xi,\xi^*}$$

s.t.

$$\begin{aligned}
\langle w, x_i\rangle - b - y_i &\le \varepsilon + \xi_i, & i &= 1,\dots,l,\\
\langle w, x_i\rangle - b - y_i &\ge -\varepsilon - \xi_i^*, & i &= 1,\dots,l,\\
\xi_i &\ge 0, & i &= 1,\dots,l,\\
\xi_i^* &\ge 0, & i &= 1,\dots,l.
\end{aligned}$$

Using duality, the above problems is equivalent to

$$-\varepsilon\sum_{i=1}^{l}(\alpha_i^* + \alpha_i) + \sum_{i=1}^{l} y_i(\alpha_i^* - \alpha_i) - 0.5\sum_{i,j=1}^{l}(\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j)\langle x_i, x_j\rangle \to \max_{\alpha,\alpha^*}$$

s.t.

$$\sum_{i=1}^{l}(\alpha_i^* - \alpha_i) = 0,$$

$$0 \le \alpha_i \le C,\ \ i = 1,\dots,l,$$

$$0 \le \alpha_i^* \le C,\ \ i = 1,\dots,l.$$

Support Vector Regression (SVR)

If Kernels are used:

$$f(x,\alpha)=(\sum_{i=1}^{l}\beta_i K(x_i,x))-b$$

$$-\varepsilon\sum_{i=1}^{l}(\alpha_i^*+\alpha_i)+\sum_{i=1}^{l}y_i(\alpha_i^*-\alpha_i)-0.5\sum_{i,j=1}^{l}(\alpha_i^*-\alpha_i)(\alpha_j^*-\alpha_j)K(x_i,x_j)\to\max_{\alpha,\alpha^*}$$

s.t.

$$\sum_{i=1}^{l}(\alpha_i^*-\alpha_i)=0,$$

$$0\le\alpha_i\le C,\ \ i=1,\dots,l,$$

$$0\le\alpha_i^*\le C,\ \ i=1,\dots,l.$$

Then $\beta_i=\alpha_i^*-\alpha_i,\ i=1,\dots,l.$ $\quad f(x,\alpha)=(\sum_{i=1}^{l}\beta_i K(x_i,x))-b$

$$b=(\sum_{i=1}^{l}y_i\alpha_i K(x_i,x_{i_0}))-y_{i_0}-\varepsilon \quad \text{for some } \alpha_{i_0}:\ 0<\alpha_{i_0}<C,\ (\alpha_{i_0}\ne 0,\ \alpha_{i_0}\ne C)$$

or

$$b=(\sum_{i=1}^{l}y_i\alpha_i^* K(x_i,x_{i_0}))-y_{i_0}+\varepsilon \quad \text{for some } \alpha_{i_0}^*:\ 0<\alpha_{i_0}^*<C,\ (\alpha_{i_0}^*\ne 0,\ \alpha_{i_0}^*\ne C)$$