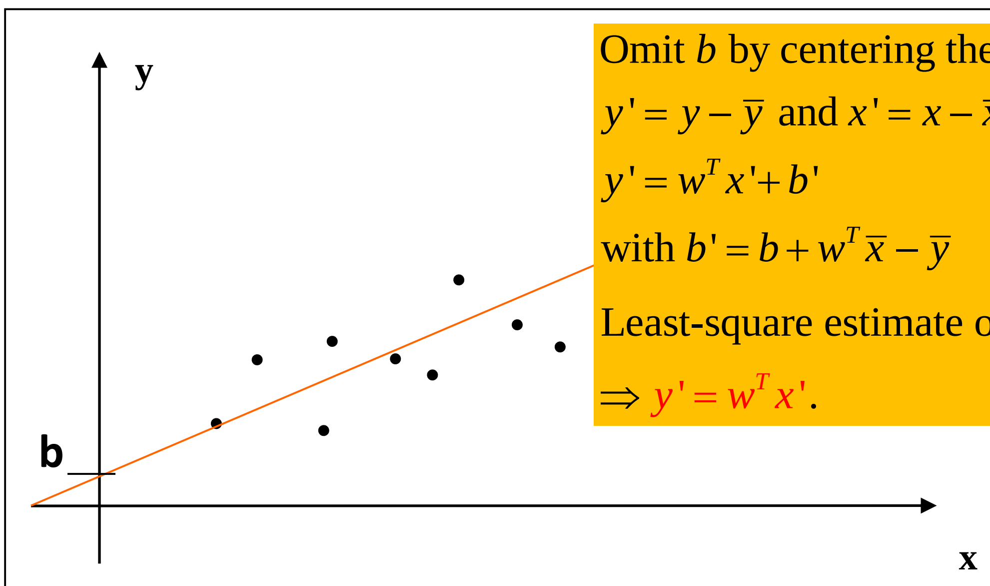
Kernel trick

Ridge regression

Recap: Linear regression

Linear regression : linear map between input $x \in \mathbb{R}^N$ and output $y \in \mathbb{R}$, parametrized by the slope vector $w \in \mathbb{R}^N$ and the intercept $b \in \mathbb{R}$.

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



Omit b by centering the data:

$y' = y - \bar{y}$ and $x' = x - \bar{x}$, $\quad \bar{x}, \bar{y}$: mean on x and y

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

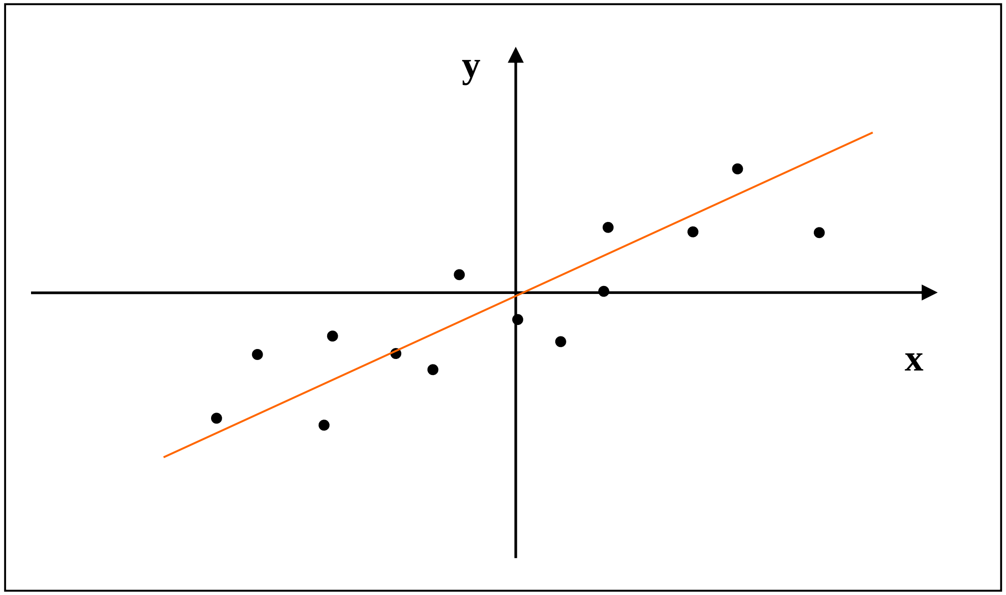
with $b' = b + w^T \bar{x} - \bar{y}$

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

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

Recap: Linear regression

$$y = f\left(x; w\right) = w^T x$$



Loss function in linear regression

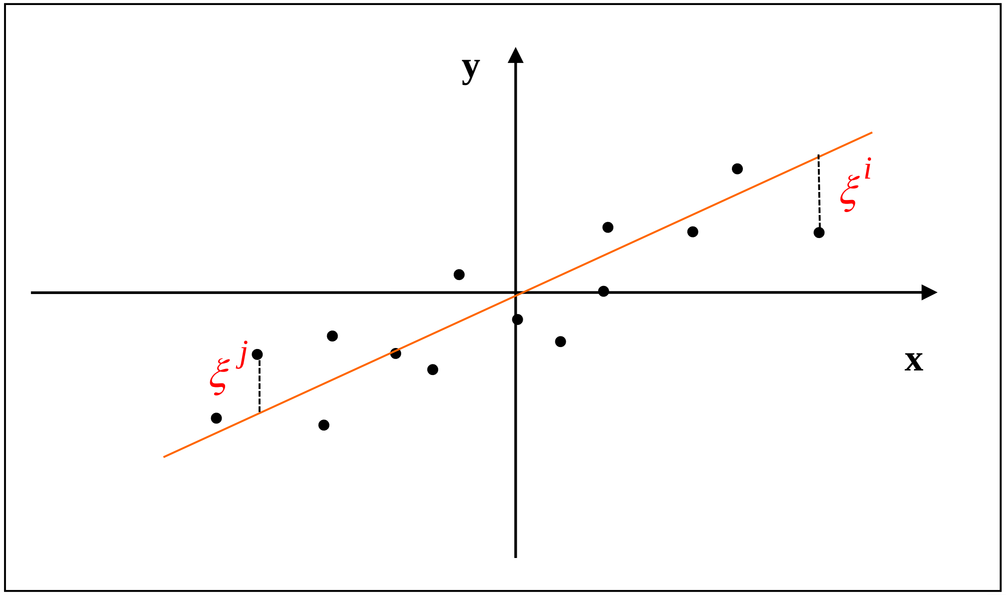
Minimizing quadratic loss function $L(x, y; w) = \left\| y - \langle w, x \rangle \right\|$

$\Rightarrow$ Find w such that $L(x, y; w) \approx 0$

Pair of M training points $X = [x^1 \; x^2 \; \dots \; x^M]$ and $\mathbf{y} = [y^1 \; y^2 \; \dots \; y^M]^T$

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

$$L(x, y; w) = \sum_{i=1}^{M} \underbrace{\left\| y^i - \langle w, x^i \rangle \right\|}_{\left\| \xi^i \right\|}$$



Closed-form solution in linear regression

$$L(X, \mathbf{y}; w) = \sum_{i=1}^{M} \frac{1}{2}\left(y^i - w^T x^i\right)^2 = \frac{1}{2}\left(\mathbf{y} - X^T w\right)^T \left(\mathbf{y} - X^T w\right)$$

$$X = [x^1 \ x^2 \ \dots x^M]$$

$$\mathbf{y} = [y^1 \ y^2 \ \dots y^M]^T$$

Optimal w given by:

$$w^* = \min_w \left(\frac{1}{2}\left(\mathbf{y} - X^T w\right)^T \left(\mathbf{y} - X^T w\right) \right)$$

The problem has an analytical solution:

$$X\left(\mathbf{y} - X^T w\right)^T = 0 \Rightarrow X\mathbf{y} - XX^T w = 0$$

$$\Rightarrow \quad w^* = \left(XX^T\right)^{-1} X\mathbf{y}$$

Singularity

It has an exact solution if:

a) XX^T is not singular (it is singular with not enough datapoints)

b) Data is not noisy (otherwise no single match to $y^i = \langle w, x^i \rangle$)

$w \in \mathbb{R}^N$

$\Rightarrow$ requires N datapoints at minimum to solve

Generally not too computationally intensive as $N << M$.

Requires $O(N^3)$ operations!

$$w^* = \left(XX^T\right)^{-1} X\mathbf{y}$$

Singularity

It has an exact solution if:

a) XX^T is not singular (it is singular with not enough datapoints)

b) Data is not noisy (otherwise no single match to $y^i = \langle w, x^i \rangle$)

If XX^T is singular, two solutions:

a) Approximate $\left(XX^T\right)^{-1}$ with pseudo-inverse (minimize norm)

b) Tradeoff size of norm against loss (Ridge Regression)

Regularizing

$$\min_w L(\mathrm{X},\mathbf{y};w) = \min_w \left(\frac{1}{2}\left(\left(\mathbf{y}-X^T w\right)^T\left(\mathbf{y}-X^T w\right)+\lambda w^T w\right)\right), \quad \lambda \geq 0$$

Regularization Term
Introduces penalty for large weights
$\rightarrow$ Reduces number of solutions

Take derivative for w:

$$X\mathbf{y} - XX^T w - \lambda w = 0$$

$$\Rightarrow w^* = \left(XX^T + \lambda I\right)^{-1} X\mathbf{y}$$

Complexity still $O\left(N^3\right)$

Always invertible for $\lambda > 0$

Closed-form solution

Take derivative for w:

$$X\mathbf{y} - XX^{T}w - \lambda w = 0$$

Rewrite $\Rightarrow w = \lambda^{-1} X\left(\mathbf{y} - X^{T}w\right)$

Define $\alpha := \lambda^{-1}\left(\mathbf{y} - X^{T}w\right)$ $\Rightarrow w = X\alpha$

Replacing, we get: $\lambda\alpha = \left(\mathbf{y} - X^{T}X\alpha\right)$

The optimum is:

$\Rightarrow \alpha = \left(X^{T}X + \lambda I\right)^{-1}\mathbf{y}$: This solution is called the Dual.

The kernel trick to enable nonlinear regression

Problem: Estimate a non-linear function $y = f(x; w)$

The exists a non-linear transformation ϕ, such that the problem becomes linear.

$\exists \phi$, s.t. $y = w^T \phi(x)$.

$$\Rightarrow w = \Phi(X)\alpha \quad \text{Columns of } \Phi(X) \text{ are } \phi(x^i)$$

$$\alpha = \left(\Phi(X)^T \Phi(X) + \lambda I\right)^{-1} \mathbf{y}$$

The solution is then: $w^* = \Phi(X)\left(\Phi(X)^T \Phi(X) + \lambda I\right)^{-1} \mathbf{y}$.

For a query point x, we compute $y = f(x) = w^T x$

$$\Rightarrow y = \sum_{i=1}^{M} \left\langle \phi(x^i), \phi(x) \right\rangle \left(\Phi(X)^T \Phi(X) + \lambda I\right)^{-1} \mathbf{y}$$

The kernel trick to enable nonlinear regression

Replace all inner products between training points

by kernel function $k : X \times X \to \mathbb{R}$ $\quad k\left(x^i, x^j\right) \to \left\langle \phi\left(x^i\right), \phi\left(x^j\right) \right\rangle$.

The kernel function is easier to compute and does not require to know ϕ.

Predicted output for a query point x becomes:

$$y = k(X,x)\Big(\underbrace{K(X,X)}_{\text{Gram Matrix in feature space}} + \lambda I\Big)^{-1}\mathbf{y}, \quad k(X,x) = \begin{bmatrix} k\left(x^1, x\right) \\ . \\ . \\ k\left(x^M, x\right) \end{bmatrix}^{\mathrm{T}}$$

$K(X,X)$ Gram matrix $M \times M$,

M : number of datapoints

Complexity $O\left(M^3\right)$

$$\Rightarrow y = \sum_{i=1}^{M} \left\langle \phi\left(x^i\right), \phi(x) \right\rangle \left(\Phi(X)^T \Phi(X) + \lambda I\right)^{-1} \mathbf{y}$$