Why $w = X^{\dagger}y$

Explaining Linear Regression famous *one-step learning*

Fábio Colaço (fabioiuri@live.com)

8 de setembro de 2018

1 Defining Notation

- $\boldsymbol{X} = \begin{bmatrix} -(x_1)^T- \\ -(x_2)^T- \\ ... \\ -(x_N)^T- \end{bmatrix}_{N \times d+1}$ Set of N inputs (transposed input sets).
- $\boldsymbol{x_i} = \begin{bmatrix} x_{i,0} \\ x_{i,1} \\ ... \\ x_{i,d} \end{bmatrix}_{d+1 \times 1}$ Single input set, with $d+1$ features.
- $\boldsymbol{y} = \begin{bmatrix} y_1 \\ y_2 \\ ... \\ y_N \end{bmatrix}_{N \times 1}$ Output set with the correct results for each input set.
- $\boldsymbol{w} = \begin{bmatrix} w_0 \\ w_1 \\ ... \\ w_d \end{bmatrix}_{d+1 \times 1}$ Weight set with real-valued weights, for each feature respectively.
- $\boldsymbol{f(x)}$ represents the output value of the (unknown) target function $\boldsymbol{f}$, when given the input set x (matrix of dimensions $d+1 \times 1$).
- $\boldsymbol{h(x)}$ represents the output value of the hypothesis function $\boldsymbol{h}$, when given the input set x (matrix of dimensions $d+1 \times 1$).

$$h(x) = \sum_{i=0}^{d} w_i x_i \tag{1}$$

$$= w^T x \tag{2}$$

- $\boldsymbol{E_{in}(h)}$ represents the in sample error of $\boldsymbol{h}$.
  More formally:

$$E_{in}(h) = \frac{1}{N}\sum_{n=1}^{N}(h(x_n) - f(x_n))^2 \tag{1}$$

$$= \frac{1}{N}\sum_{n=1}^{N}(h(x_n) - y_n)^2 \tag{2}$$

  The step from (1) to (2) comes due to the fact that, in sample, $f(x_n)$ is known; and so we can rewrite it as y_n.

2 Expanding E_{in}

Lets try to expand E_{in} so we get a full matrix formula out of it:

$$E_{in}(h) = \frac{1}{N}\sum_{n=1}^{N}(h(x_n) - y_n)^2 \tag{1}$$

$$E_{in}(w) = \frac{1}{N}\sum_{n=1}^{N}((\sum_{i=0}^{d} w_i x_{n,i}) - y_n)^2 \tag{1}$$

$$= \frac{1}{N}\sum_{n=1}^{N}(w^T x_n - y_n)^2 \tag{2}$$

$$= \frac{1}{N}\left\|Xw - y\right\|_e^2 \tag{3}$$

$$= \frac{1}{N}(w^T X^T X w - 2y^T X w + y^T y) \tag{4}$$

Clarifying the **step 3**: here we use the euclidean norm of the column vector Xw - y. Assuming $P = Xw - y$, then $\left\|P\right\|_e = \sqrt{\sum_{n=1}^{N}(P_n)^2}$. To get rid of the square root, we square the euclidean norm of P and we have: $\left\|P\right\|_e^2 = \sum_{n=1}^{N}(P_n)^2$, giving us an equivalent, but prettier formula.
On **step 4**, we just apply the distributive rule on the column vector $P = Xw - y : P^2 = P.P$.

3 Minimizing $E_{in}(w)$

Since $\frac{1}{N}(w^T X^T X w - 2y^T X w + y^T y)$ is differentiable, $E_{in}(w)$ is differentiable as well. So, we can apply the standard matrix calculus and find w that minimizes $E_{in}(w)$, by requiring that the gradient of E_{in} with respect to w is the zero vector: $\nabla E_{in}(w) = \vec{0}$.

Helper note: Important identities to apply on vector by vector differentiation ($\frac{\partial y}{\partial x}$):

$$\frac{\partial x^T A}{\partial x} = A \textbf{ and } \frac{\partial x^T A x}{\partial x} = (A + A^T)w$$

$$\begin{aligned}
\nabla E_{in}(w) &= \nabla_w(\frac{1}{N}(w^T X^T X w - 2y^T X w + y^T y)) && (1)\\
&= \frac{1}{N}\nabla_w(w^T X^T X w - 2y^T X w + y^T y)) && (2)\\
&= \frac{1}{N}[\nabla_w(w^T X^T X w) - \nabla_w(2y^T X w) + \nabla_w(y^T y))] && (3)\\
&= \frac{1}{N}[(w(X^T X + (X^T X)^T)) - \nabla_w(2w^T X^T y) + 0] && (4)\\
&= \frac{1}{N}[(X^T X w + w(X^T X)^T) - 2X^T y] && (5)\\
&= \frac{1}{N}[(X^T X w + X^T X w) - 2X^T y] && (6)\\
&= \frac{1}{N}(2X^T X w - 2X^T y) && (7)\\
&= \frac{2}{N}(X^T X w - X^T y) && (8)\\
& && (9)
\end{aligned}$$

With $\nabla E_{in}(w) = \vec{0}$, then:

$$\begin{aligned}
\frac{2}{N}(X^T X w - X^T y) &= \vec{0} && (1)\\
X^T X w - X^T y &= \vec{0} && (2)\\
X^T X w &= X^T y && (3)\\
(X^T X)^{-1} X^T X w &= (X^T X)^{-1} X^T y && (4)\\
I_{d+1} w &= (X^T X)^{-1} X^T y && (5)\\
w &= X^\dagger y && (6)
\end{aligned}$$

By definition, $X^\dagger = (X^T X)^{-1} X^T$ and it is called by the pseudoinverse of X (or the Moore–Penrose inverse of X).

$\therefore$ By minimizing $E_{in}(w)$, using the standard matrix calculus, we found that if we compute w to be $X^\dagger y$, we actually get the best linear formula to minimize the in sample error in one step only - this is called the *one-step learning*.