# Least-Squares Regression

In least-squares regression, the training data contains $n$ different training pairs $(\overline{X_1}, y_1) \dots (\overline{X_n}, y_n)$, where each $\overline{X_i}$ is a $d$-dimesnional representation of the data points, and each $y_i$ is a real-valued target. The fact that the target is *real-valued* is important, because the underlying problem is then referred to as *regression* rather than *classification*. In fact, as we will see later, one can also use least-squares regression on binary targets by "pretending" that these targets are real-valued. The resulting approach is equivalent to the Widrow-Hoff learning algorithm, which is famous in the neural network literature as the second learning algorithm proposed after the perceptron.

In least-squares regression, the target variable is related to the feature variables using the relationship $\hat{y}_i = \overline{W} \cdot \overline{X_i}$.

The portion of the loss that is specific to the $i$th training instance is given by the following $L_i = e_i^2 = (y_i - \hat{y}_i)^2$.

The stochastic gradient-descent steps are determined by computing the gradient of $e_i^2$ with respect to $\overline{W}$, when the training pair $(\overline{X_i}, y_i)$ is presented to the neural network.

• This gradient can be computed as follows $\frac{\partial L_i}{\partial \overline{W}} = -e_i \overline{X_i}$.
$\Rightarrow L_i = y_i^2 + \hat{y}_i^2 - 2y_i\hat{y}_i = y_i^2 + (\overline{W} \cdot \overline{X_i})^2 - 2y_i(\overline{W} \cdot \overline{X_i}) \Rightarrow$
$\Rightarrow \frac{\partial L_i}{\partial \overline{W}} = 2(\overline{W} \cdot \overline{X_i}) \cdot \overline{X_i} - 2y_i\overline{X_i} = 2(\hat{y}_i - y_i)\overline{X_i} = -2e_i\overline{X_i}$

• Therefore, the gradient-descent updates for $\overline{W}$ are computed using the above gradient and step-size $\alpha$ (hyperparameter), $\overline{W} \Leftarrow \overline{W} + \alpha e_i \overline{X_i} = \overline{W} + \alpha(y_i - \hat{y}_i)\overline{X_i}$.
$\Rightarrow \overline{W} \Leftarrow \overline{W} + 2\alpha e_i \overline{X_i}$ so we could to not take into account the 2 if we think that $\alpha \Leftarrow 2\alpha$.
With regularization, the update is as follows $\overline{W} \Leftarrow \overline{W}(1 - \alpha\lambda) + \alpha(y_i - \hat{y}_i)\overline{X_i}$, where $\lambda > 0$ is the regularization parameter.

What if we applied least-squares regression directly to minimize the squared distance of the real-valued prediction $\hat{y}_i$ from the observed binary targets $y_i \in \{-1, +1\}$? The direct application of least-squares regression to binary targets is referred to as least-squares classification. The gradient-descent is the same as the one shown above.
This direct application of least-squares regression to binary targets is referred to as Widrow-Hoff learning.

## Widrow-Hoff Learning

The loss function of the Widrow-Hoff method can be rewritten slightly from least-squares regression because of its binary responses, when working with binary responses in $\{-1, +1\}$

$\Rightarrow L_i = (y_i - \hat{y}_i)^2 = y_i^2(y_i - \hat{y}_i)^2 = (y_i^2 - y_i\hat{y}_i)^2 = (1 - y_i\hat{y}_i)^2$

One of the flows of this method is that it penalizes over-performance, and other methods can be shown to be closely related the Widrow-Hoff loss function by using different ways of repairing the loss so that over-performance is not penalized.

The gradient-descent updates of least-squares regresion can be rewritten slightly for Widrow-Hoff learning becuase of binary response variables:

$$\overline{W} \Leftarrow \overline{W}(1 - \alpha \cdot \lambda) + \alpha(y_i - \hat{y}_i)\overline{X_i} = \overline{W}(1 - \alpha \cdot \lambda) + \alpha y_i(1 - y_i\hat{y}_i)\overline{X_i}$$

## Closed Form Solutions

The special case of least-squares regression and classification is solvable in closed form (without gradient-descent) by using the pseudo-inverse of the $n \times d$ training data matrix $D$, whose rows are $\overline{X_1}, \ldots, \overline{X_n}$. Let the $n$-dimensional column vector of dependent variables be denoted by $\overline{y} = [y_1, \ldots, y_n]^T$ .

- The pseudo-inverse of matrix $D$ is defined as $D^+ = (D^T D)^{-1} D^T$.
- Then, the row-vector $\overline{W}$ is defined by $\overline{W}^T = D^+ \overline{y}$.
- If regularization is incorporated, $\overline{W}^T = (D^T D + \lambda I)^{-1} D^T \overline{y}$.

One rarely inverts large matrices like $D^T D$. In fact, the Widrow-Hoff updates provide a very efficient way of solving the problem without using the closed form solution.