# 1 Regression

**Introduction** source code

Regression based methods are trained on input data samples having output responses that are continuous numeric values unlike classification, where we have discrete categories or classes.

Regression models make use of input data attributes or features (explanatory or independent variables) and their corresponding continuous numeric output values (dependent or outcome variable) to learn specific relationships and associations between the inputs and their corresponding outputs.

## 1.1 Linear Regression

- Feature mapping $\Phi : X \longrightarrow \mathbb{R}^N$
- Hypothesis set: linear functions.

$$x \longmapsto \beta \cdot \Phi(x) + b : \beta \in \mathbb{R}^N, b \in \mathbb{R} \tag{1}$$

- Optimization problem: empirical risk minimization.

$$\min_{\mathbf{w},b} F(\mathbf{w}, b) = \frac{1}{m} \sum_{i=1}^{m} (\beta \cdot \Phi(x_i) + b - y_i)^2 \tag{2}$$

### 1.1.1 Linear Regression Solution

- Rewrite eq(2) as

$$F(\boldsymbol{\beta}) = \frac{1}{m} ||X^T \beta - Y||^2 \tag{3}$$

where

$$X = \begin{bmatrix} 1 & \dots & 1 \\ \Phi(x_1) & \dots & \Phi(x_m) \end{bmatrix} \tag{4}$$

$$\mathbf{X}^T = \begin{bmatrix} 1 & \boldsymbol{\Phi(x_1)} \\ & \vdots \\ 1 & \boldsymbol{\Phi(x_1)} \end{bmatrix}, \boldsymbol{\beta} = \begin{bmatrix} b \\ \beta_1 \\ \vdots \\ \beta_N \end{bmatrix} \tag{5}$$

- Convex and differentiable function.

$$\Delta F(\boldsymbol{\beta}) = \frac{2}{m}\boldsymbol{X}(\boldsymbol{X^T}\boldsymbol{\beta} - \boldsymbol{Y}) \tag{6}$$

$$\Delta F(\boldsymbol{\beta}) = 0 \Leftrightarrow \boldsymbol{X}(\boldsymbol{X^T}\boldsymbol{\beta} - \boldsymbol{Y}) = 0 \Leftrightarrow \boldsymbol{X}\boldsymbol{X^T}\boldsymbol{\beta} = \boldsymbol{X}\boldsymbol{Y} \tag{7}$$

$$\boldsymbol{\beta} = \begin{cases} \boldsymbol{(XX^T)^{-1}XY} & \textit{if } \boldsymbol{XX^T} \textit{is invertible} \\ \boldsymbol{(XX^T)^{\dagger}XY} & \textit{in general} \end{cases} \tag{8}$$

  - Computational complexity: $O(mN + N^3)$ if matrix inversion is $O(N^3)$
  - For output labels in $\mathbb{R}^p$, $p > 1$, solve $p$ distinct linear regression problem.
  - No regularization.
  - Note: That $\beta$ (model parameters) can also be estimated using method of Least squares (based on linear algebra.)

### 1.1.2 Closed form Solution

- Given a minimization problem such that $F(\boldsymbol{\beta}) = ||X^T\boldsymbol{\beta} - Y||^2$ where $h_{\boldsymbol{\beta}}(X) = X^T\boldsymbol{\beta}$ is the hypothesis. $F(\boldsymbol{\beta})$ is the sum of square error between the hypothesis (predicted value) and the real objective values $(Y)$.

- $F(\boldsymbol{\beta})$ can be expanded as

$$F(\boldsymbol{\beta}) = (X^T\boldsymbol{\beta} - Y)^T(X^T\boldsymbol{\beta} - Y) \tag{9}$$

$$F(\boldsymbol{\beta}) = XX^T\boldsymbol{\beta}^T\boldsymbol{\beta} - X^T\boldsymbol{\beta}y - y^TX^T\boldsymbol{\beta} + y^Ty \tag{10}$$

Introducing *trace* technique from linear algebra to manipulate the two middle matrices. Given a *matrix* A. the $trace(A) = trace(A^T)$ and *vice-vera*. Note also, $XX^T\boldsymbol{\beta}^T\boldsymbol{\beta}$ can be rewritten as $XX^T||\beta||^2$.

$$F(\boldsymbol{\beta}) = tr(XX^T\boldsymbol{\beta}^T\boldsymbol{\beta}) - tr(X^T\boldsymbol{\beta}y) - tr(y^TX^T\boldsymbol{\beta}) + tr(y^Ty) \tag{11}$$

$$F(\boldsymbol{\beta}) = tr(X^TX\boldsymbol{\beta}^T\boldsymbol{\beta}) - tr(2X^T\boldsymbol{\beta}y) + tr(y^Ty) \tag{12}$$

Trace is not needed anymore, hence

$$F(\boldsymbol{\beta}) = X^TX\boldsymbol{\beta}^T\boldsymbol{\beta} - 2X^T\boldsymbol{\beta}y + y^Ty \tag{13}$$

- Convex and differentiable

$$\frac{\partial F(\boldsymbol{\beta})}{\partial \beta} = 0 \Leftrightarrow 2X^TX\boldsymbol{\beta} - 2X^Ty = 0 \tag{14}$$

$$\boldsymbol{\beta} = (X^TX)^{-1}X^Ty \tag{15}$$

## 1.2 Lasso Regression

- Optimization problem: least absolute shrinkage and selection operator.

$$\min_{w} F(\boldsymbol{\beta}, b) = \lambda||\boldsymbol{\beta}||_1 + \sum_{i=1}^{m}(\boldsymbol{\beta}\cdot\mathbf{x}_i + b - y_i)^2 \tag{16}$$

where $\lambda \geq 0$ is the regularization parameter.

- Solution: equivalent convex quadratic program (QP).
  - General: standard QP solvers
  - specific algorithm: LARS (least angular regression procedure), entire path solutions.

## 1.3 Kernel Ridge Regression

### 1.3.1 Mean Square Bound- kernel-based Hypothesis

- **Theorem**: Let $K : X \times X \rightarrow \mathbb{R}$ be a PDS kernel (Positive Definite Symmmetric kernel function- implicitly specify an inner product in a high-dimension Hilbert space where large-margin solutions are sought.

Algorithm convergence is always guaranteed.) Let Φ: $X \rightarrow H$ be a feature mapping associated to $K$. Let $H = \{x \mapsto \beta \cdot \Phi(x) : ||\beta||_H \leq \Lambda\}$. Assume $K(x,x) \leq R^2$ and $|f(x)| \leq \Lambda R$ for all $x \in X$. Then, for any $\delta \gg 0$, with probability at least $1-\delta$, for any $h \in H$,

$$R(h) \leq \widehat{R}(h) + \frac{8R^2\Lambda^2}{\sqrt{m}}\left(1 + \frac{1}{2}\sqrt{\frac{log\frac{1}{\delta}}{2}}\right) \tag{17}$$

$$R(h) \leq \widehat{R}(h) + \frac{8R^2\Lambda^2}{\sqrt{m}}\left(\sqrt{\frac{Tr[\mathbf{K}]}{mR^2}} + \frac{3}{4}\sqrt{\frac{log\frac{2}{\delta}}{2}}\right) \tag{18}$$

- **Proof**:direct application of the Rademacher Complexity Regression Bound and bound on the Rademacher complexity of kernel-based hypotheses

$$\Re_S(H) \leq \frac{\Lambda\sqrt{Tr[\mathbf{K}]}}{m} \leq \sqrt{\frac{R^2\Lambda^2}{m}} \tag{19}$$

### 1.3.2 Ridge Regression

- Optimzation problem:

$$\min_{\boldsymbol{\beta}} F(\boldsymbol{\beta}, b) = \lambda||\boldsymbol{\beta}||^2 + \sum_{i=1}^{m}(\boldsymbol{\beta} \cdot \Phi(x_i) + b - y_i)^2 \tag{20}$$

  where $\lambda \geq 0$ is the regularization parameter.

  - directly based on generalization bound.
  - generalization of linear regression.
  - has a closed form solution.

- Assume $b = 0$: often constant features used (but not equivalent to the use of original offset).

- Rewrite objective function as:

$$\boldsymbol{\beta} = \lambda||\boldsymbol{\beta}||^2 + ||\mathbf{X}^T\boldsymbol{\beta} - \mathbf{Y}||^2 \tag{21}$$

- Convex and differentiable function

$$\Delta F(\boldsymbol{\beta}) = 2\lambda\boldsymbol{\beta} + 2\mathbf{X}(\mathbf{X}^T\boldsymbol{\beta} - \mathbf{Y}) \tag{22}$$

$$\Delta F(\boldsymbol{\beta}) = 0 \Leftrightarrow (\mathbf{X}\mathbf{X}^T + \lambda\mathbf{I})\boldsymbol{\beta} = \mathbf{X}\mathbf{Y} \tag{23}$$

- Solution:

$$\boldsymbol{\beta} = (\mathbf{X}\mathbf{X}^T + \lambda\mathbf{I})^{-1}\mathbf{X}\mathbf{Y} \qquad \textit{always invertible} \tag{24}$$

  Note: when $\lambda$ is too small, the solution of the $\beta$ parameters is same with ordinary linear regression.

- Computational complexity:
  - solution → $O(mN^2 + N^3)$
  - prediction → $O(N)$

### 1.3.3 Kernel Ridge Regression

We now replace all data-cases with their feature vector: $x_i \to \Phi_i = \Phi(x_i)$. The number of dimentsions can be much higher, or even inifitely higher that the number of data-samples. Algebraic trick to perform the inverse of equation (17). using dimension of feature space is given by:

$$(P^{-1} + B^T R^{-1} B)^{-1} = PB^T(BPB^T + R)^{-1}) \tag{25}$$

Now note that if B is not square, the inverse is performed in spaces of different dimensionality. To apply this to our case we define $\Phi = \Phi_a i$ and $y = y_i$.

- Optimization problem:

$$\max_{\Phi\in\mathbb{R}^m} -\lambda\Phi^T\Phi + 2\Phi^T y - \Phi^T K\Phi \tag{26}$$

- Solution:

$$\boldsymbol{\beta} = (\lambda\boldsymbol{I}_d + \boldsymbol{\Phi^T\Phi})^{-1}\boldsymbol{\Phi^T y} = \boldsymbol{\Phi}(\boldsymbol{\Phi^T\Phi} + \lambda\boldsymbol{I}_n)^{-1}y \tag{27}$$

  This equation can be rewritten as $w = \sum_i \alpha\Phi(\boldsymbol{x}_i)$ where $\alpha = (\boldsymbol{\Phi^T\Phi} + \lambda\boldsymbol{I}_n)^{-1}y$. The solution $\beta$ must lie in the span of the data-samples, even if the dimensionality of the feature space is much larger than the number of data-samples.

$$y = \boldsymbol{\beta^T \Phi(x)} = \boldsymbol{y}(\boldsymbol{\Phi^T \Phi} + \boldsymbol{\lambda I_n})^{-1}\boldsymbol{\Phi^T \Phi(x)} = \boldsymbol{y}(\boldsymbol{K} + \boldsymbol{\lambda I_n})^{-1}\boldsymbol{\kappa(x)} \tag{28}$$

where $K(bx_i, bx_j) = \Phi(x_i)^T \Phi(x_j)$ and $\kappa(x) = K(\boldsymbol{x_i, x})$

## 1.4 Gradient Descent

**Cost**

$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(h_{(\theta)}^{(i)} - y^{(i)})^2 \tag{29}$$

**Gradient**

$$\frac{\partial J(\theta)}{\partial \theta_j} = \frac{1}{m}\sum_{i=1}^{m}(h_{(\theta)}^{(i)} - y^{(i)}).X_j^{(i)} \tag{30}$$

**Gradients**

$$\theta_0 := \theta_0 - \alpha.(\frac{1}{m}.\sum_{i=1}^{m}(h_{(\theta)}^{(i)} - y^{(i)}).X_0^{(i)}) \tag{31}$$

$$\theta_1 := \theta_1 - \alpha.(\frac{1}{m}.\sum_{i=1}^{m}(h_{(\theta)}^{(i)} - y^{(i)}).X_1^{(i)}) \tag{32}$$

$$\theta_2 := \theta_2 - \alpha.(\frac{1}{m}.\sum_{i=1}^{m}(h_{(\theta)}^{(i)} - y^{(i)}).X_2^{(i)}) \tag{33}$$

$$\theta_j := \theta_j - \alpha.(\frac{1}{m}.\sum_{i=1}^{m}(h_{(\theta)}^{(i)} - y^{(i)}).X_0^{(i)}) \tag{34}$$

**Gradient Descent**
*Repeat*

$$\theta_j := \theta_j - \alpha^* \frac{\partial J(\theta)}{\partial \theta_j} \tag{35}$$

**Stochastic Gradient Descent**
*Repeat*

$$\theta_j := \theta_j - \alpha^* \frac{\partial J(\theta)}{\partial \theta_j}(X^{(i)}, y^{(i)}) \tag{36}$$

### Minibatch Gradient Descent

*Repeat*

$$\theta_j := \theta_j - \alpha^* \frac{\partial J(\theta)}{\partial \theta_j}(X_b^{(i)}, y_b^{(i)}) \tag{37}$$