Linear Regression

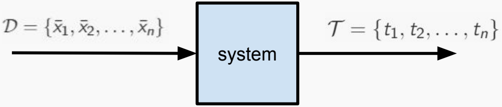
Pattern Recognition and Machine Learning, Jul-Nov 2019

Indian Institute of Technology Madras

August 19, 2019

Linear Regression

- Let $\mathcal{D} = \{\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_N\}$ be the data set of n feature vectors.
- Let $\mathcal{T} = \{t_1, t_2, \ldots, t_N\}$ be the target value of all n data points.
- Then linear regression is problem of predicting the system shown below.



- $t_n = Y(x_n)$, where, $Y()$ is the function which needs to be estimated.

Polynomial Regression

- Let x_n be a scalar and we need to find the system $Y()$ such that $t_n = Y(x_n)$
- $Y(x) = w_0 + w_1 x + w_2 x^2 + \ldots + w_{m-1} x^{m-1}$ be the required polynomial system, we need to find the weights $W = [w_0, w_1, \ldots, w_{m-1}]$ that minimizes the error $E = \sum_{n=1}^{N} t_n - Y(x_n)$
- To find the best weights we need to solve the derivative $\frac{\partial E}{\partial w_i} = 0$ for all w_i.

$$\frac{\partial E}{\partial w_i} = 0$$
$$\implies \sum_{n=1}^{N} \sum_{j=0}^{m-1} w_i \, x_n^i \, x_n^j = \sum_{n=1}^{N} t_n x_n^i$$

Polynomial Regression (contd..)

$$\sum_{n=1}^{N}\sum_{j=0}^{m-1} w_i \, x_n^i \, x_n^j = \sum_{n=1}^{N} t_n x_n^i \; \forall i$$

- This can be rewritten as

$$A\bar{w} = \bar{b}$$

- The solution is given by

$$\bar{w} = A^{-1}\bar{b}$$

Linear Regression (contd..)

- For linear regression the basis function need not be always a polynomial hence the $Y(\bar{x}_n)$ can be rewritten as

$$Y(\bar{x}) = w_0 + \sum_{i=1}^{m-1} w_i \phi_i(\bar{x})$$

- where, ϕ_i is the i^{th} basis function.
- Some examples are given below

$$\phi(\bar{x}) = \bar{x}^i \qquad \text{polynomial basis}$$

$$\phi(\bar{x}) = e^{\frac{-(\bar{x}-\bar{\mu})^2}{\sigma}} \qquad \text{Gaussian basis}$$

$$\phi(\bar{x}) = \sigma\left(\frac{-(\bar{x}-\bar{\mu})^2}{s}\right) \qquad \text{sigmoid basis}$$

Linear Regression (contd..)

- Let X and T be matrices defined as follows.

$$X = \begin{pmatrix} 1 & \phi_1(\bar{x}_1) & \dots & \phi_n(\bar{x}_1) \\ 1 & \phi_1(\bar{x}_2) & \dots & \phi_n(\bar{x}_2) \\ \vdots & \ddots & \ddots & \vdots \\ 1 & \phi_1(\bar{x}_n) & \dots & \phi_n(\bar{x}_n) \end{pmatrix} T = \begin{pmatrix} t_1 \\ t_2 \\ \vdots \\ t_n \end{pmatrix}$$

- Now, the error can be rewritten as

$$\begin{aligned} E &= (T - X\bar{w})^t(T - X\bar{w}) \\ &= T^tT - T^tX\bar{w} - \bar{w}^tX^tT + \bar{w}^tX^tX\bar{w} \end{aligned}$$

Linear Regression (contd..)

- To find the least squared error solution we need to solve the derivate of E w.r.t $\bar{w}$ for zero

$$\frac{\partial E}{\partial \bar{w}} = 0$$

$$\implies 0 - 2X^t T + 2X^t X \bar{w} = 0 \qquad \therefore \frac{\partial \bar{w}^t X^t X \bar{w}}{\bar{w}} = 2X^t X \bar{w}$$

$$\therefore \frac{\partial \bar{w}^t X^t T}{\bar{w}} = X^t T$$

- Solving the above equation for $\bar{w}$, we get

$$\bar{w} = (X^t X)^{-1} X^t T$$

Ridge Regression

- In ridge regression the weights are restricted to prevent over fitting.
- The error function for ridge regression which limits the weight is given by

$$E = (T - X\bar{w})^t(T - X\bar{w}) + \lambda \underbrace{||\bar{w}||^2}_{\text{L2 -norm}}$$

- Solving the derivate of E w.r.t $\bar{w}$ for zero we get the solution as

$$\bar{w} = (X^tX + \lambda I)^{-1}X^tT$$

- This technique of constraining weights is also called as L2-regression

Other types of linear regression

Lasso Regression

- Instead of constraining the weights by L2-norm in Lasso regression the weights are constrained by the L1-norm.
- The error function for Lasso regression which limits the weight is given by L-1 norm is given by

$$E = (T - X\bar{w})^t (T - X\bar{w}) + \lambda \underbrace{||\bar{w}||^1}_{\text{L1 -norm}}$$

- In Lasso regression some of the weights even becomes zero.
- There is no closed form solution for Lasso regression, The solution needs to be obtained by quadratic programing.
- In **Hybrid Regression** the weights are constrained by both Lasso and L2 regression.

Algorithm for Lasso Regression under a constraint

Let $\sum_{i=0}^{M-1} |w_j|^p \leq \eta$

- Start with an initial estimate using ordinary least squares.
- Move the offending weights as part of the cost function.
- Reestimate Least Squares Solution.
- Repeat until the weights converge.

$p = 2$ corresponds to "Ridge Regression," $p = 1$ corresponds to "Lasso." Aside: Partial derivate of $|w_j|$ is given by:

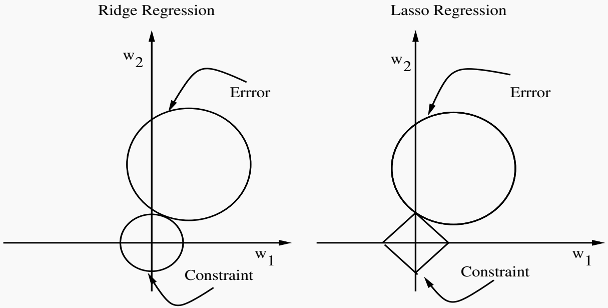
$$|w_j| = \sqrt{w_j^2} \quad (1)$$

$$/dw_j of |w_j| = \frac{1}{2}((w_j)^2)^{-\frac{1}{2}} 2w_j \quad (2)$$

$$= \frac{w_j}{|w_j|} \quad (3)$$

This is fine as long as w_j is not zero. $w_j = 0$ reduces the complexity.

Illustration of Regression with Constraints in two dimensions



Bias Variance Tradeoff I

Bias Variance Tradeoff Decomposition
Consider the following:
Let $\mathcal{D} = \bar{x}_1, \bar{x}_2, ..., \bar{x}_N$ be a limited dataset.
Let t be the actual output, and let $y(\bar{x}; \mathcal{D})$ be the output estimated given the model that is estimated using $\mathcal{D}$.
Let $\hat{t} = y(\bar{x}; \mathcal{D})$

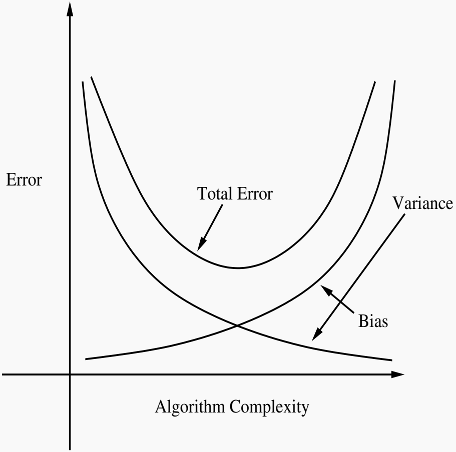
$$\begin{aligned}
TotalError &= E_{\mathcal{D}}[(\hat{t} - t)^2] \\
&= E_{\mathcal{D}}[(\hat{t} - E_{\mathcal{D}}[\hat{]} + E_{\mathcal{D}}[\hat{]} - t)^2] \\
&= E_{\mathcal{D}}[((E_{\mathcal{D}}[\hat{]} - t) + (\hat{t} - E_{\mathcal{D}}[\hat{]}))^2] \\
&= E_{\mathcal{D}}[(E_{\mathcal{D}}[\hat{]} - t)^2 + (\hat{t} - E_{\mathcal{D}}[\hat{]})^2 + 2(E_{\mathcal{D}}[\hat{]} - t)(\hat{t} - E_{\mathcal{D}}[\hat{]})]
\end{aligned}$$

Taking $E_{\mathcal{D}}$ inside the bracket, the last term disappears, leaving

Bias Variance Tradeoff II

$TotalError = bias^2 + variance$.
If $E[]$ was used rather than $E_{\mathcal{D}}$,
$TotalError = bias^2 + variance + noise$.
bias corresponds to the bias of the estimator, *variance* corresponds to variance of the estimator.

Bias Variance Tradeoff – Illustration



Bias Variance Tradeoff – Example