ML Supervised Learning 4 by ambedkar@IISc

- Linear Regression
- Probabilistic view of linear regression
- Logistic regression
- Hyperplane based classifiers and perceptron

Linear Regression

Linear Regression: One dimensional Case

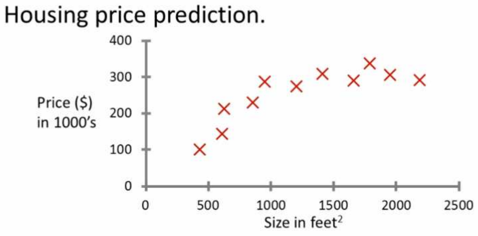
- Given N data samples of features x_n and response y_n pairs

$$(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)$$

- For now, assume that dimension of feature vector x_n is one

Problem: *Find a straight line that* ***best*** *fits these set of points*

Linear Regression: One dimensional case (contd...)



Linear Regression: One dimensional Case (contd...)

Assumption: Input and response relationship is ***linear*** (We hope so)

Problem Statement: Given data $\{(x_1, y_1, \ldots, (x_N, y_N)\}$

- find a straight line that **best** fits these set of points.
- (Rephrase) Given .... choose a straight line that best fits these set of points
  - i.e $\mathcal{F}$ is set of all linear functions.
  - In this case $\mathcal{F}$ denotes set of all straight lines on a plane.

Linear Regression: One dimensional Case (contd...)

From where do we choose or learn our solution from?

- Assume that $\mathcal{F}$ is set of all straight lines
- Further assume that $\mathcal{F}$ is set of all straight lines that are passing through origin.
  - Is this reasonable?
  - Yes! With some preprocessing we can transform the data
- That is define $\mathcal{F}$ as

$$\mathcal{F} = \{f_w(x) = wx : w \in \mathbb{R}\}$$

- We say that the class of functions $\mathcal{F}$ is paramerized by w

Note: Since f can be identified by w, our aim is to just learn w from the given data

Linear Regression: One dimensional Case (contd...)

'Best' with respect to what?

- We need some mechanism to evaluate our solution.
- For this we need to define a **loss function**
- A loss function takes two inputs: (i) response given by our solution, and (ii) groundtruth
- Loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ is defined as

$$\ell(f) = \sum_{n=1}^{N} (y_n - f_w(x_n))^2$$

which is a least squared error.

Linear Regression: One dimensional Case (contd...)

Recall what we are trying to do

$$\ell(f_w) = \sum_{n=1}^{N} (y_n - f_w(x_n))^2$$

- Note that $y_n - f_w(x_n)$ is per sample loss
- $\ell(f_w)$ is the total loss
- Now aim is to find $w \in \mathbb{R}$ that minimizes empirical risk $\ell(f_w)$.

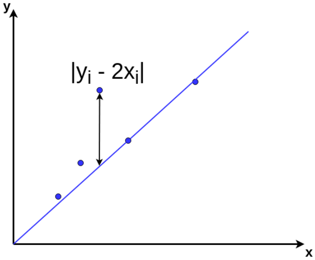
Note: Remember that we supposed to minimize true risk, since we do not know the underlying distribution we minimize empirical risk.

- **Optimization Problem:** Find f in $\mathcal{F}$ that minimizes $\ell(f)$

  $|||$

  Find $w \in \mathbb{R}$ that minimizes $\ell(w)$

  Since f is completely determined by w.



Linear Regression in one dimension.

Solution: A solution to this problem is given by

$$\frac{\mathrm{d}\ell}{\mathrm{d}w} = 0$$

This can be calculated as follows. First we will calculate the derivative of ℓ w.r.t w.

$$\ell(w) = \sum_{n=1}^{N} (y_n - wx_n)^2$$

$$\frac{\mathrm{d}\ell}{\mathrm{d}w} = \sum_{n=1}^{N} 2(y_n - wx_n)(-x_n)$$

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

$$\implies \sum_{n=1}^{N} (wx_n^2 - x_n y_n) = 0$$

Linear Regression: One dimensional Case (cont...)

Solution: A solution to this problem is given by

$$\frac{\mathrm{d}\ell}{\mathrm{d}w} = 0$$

Now by equating the derivative to 0 we get

$$\begin{aligned}
&\implies \sum_{n=1}^{N} (w x_n^2 - x_n y_n) = 0 \\
&\implies w \sum_{n=1}^{N} x_n^2 = \sum_{n=1}^{N} x_n y_n \\
&\implies w = \frac{\sum_{n=1}^{N} x_n y_n}{\sum_{n=1}^{N} x_n^2}
\end{aligned}$$

Linear Regression (cont ...)

Given a training data $\{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\}$, where

- $x_n \in \mathbb{R}^D$ is a feature vector
- $y_n \in \mathbb{R}$ is the corresponding response

Model: $y = b + w^\intercal x$

We can also write this interms of data matrices

$$X = \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix}_{N \times D} \qquad Y = \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix}_{N \times 1}$$

We get

$$Y = XW + b$$

Linear Regression(cont ...)

We have

$$Y = XW + b$$

$$\begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix} = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1D} \\ & & \vdots & \\ x_{N1} & x_{N2} & \dots & x_{ND} \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_D \end{bmatrix}_{D\times 1} + \begin{bmatrix} b \\ \vdots \\ b \end{bmatrix}$$

$$\implies \underbrace{\begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix}}_{\substack{N\times 1 \\ \text{Matrix}}} = \underbrace{\begin{bmatrix} 1 & x_{11} & x_{12} & \dots & x_{1D} \\ 1 & x_{21} & x_{22} & \dots & x_{2D} \\ & & \vdots & & \\ 1 & x_{N1} & x_{N2} & \dots & x_{ND} \end{bmatrix}}_{\substack{N\times(D+1) \\ \text{Matrix}}} \underbrace{\begin{bmatrix} b \\ w_1 \\ \vdots \\ w_D \end{bmatrix}}_{\substack{(D+1)\times 1 \\ \text{Matrix}}}$$

$$\implies Y = XW$$

Linear Regression (cont...)

Now we need to solove the following system of linear equations.

Given $Y = \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix}$ and $X = \begin{bmatrix} 1 \; x_{11} \; x_{12} \; \dots \; x_{1D} \\ \vdots \\ 1 \; x_{N1} \; x_{N2} \; \dots \; x_{ND} \end{bmatrix}$

find $W = \begin{bmatrix} b \\ w_1 \\ \vdots \\ w_D \end{bmatrix}$ that satisfies

$$Y = XW$$

On solving linear system: The above system may not have a solution i.e parameter that satisfies

$$y_n = w^\intercal x_n, \quad n = 1, 2, \dots, N$$

may not exists.

Least Square Approximation

Let us try to find an approximate solution by employing Least Square Error

$$\ell(y_n, w^\intercal x_n) = (y_n - w^\intercal x_n)^2$$

Note that one can also use

$$\ell(y_n, w^\intercal x_n) = |y_n - w^\intercal x_n|$$

which is more robust to outliers.

The total empirical error

$$L_{\text{emp}}(w) = \sum_{x=1}^{N} \ell(y_n, w^\intercal x_n) = \sum_{n=1}^{N} (y_n - w^\intercal x_n)^2$$

$$w^* = \arg\min_{w} \sum_{n=1}^{N} (y_n - w^\intercal x_n)^2$$

Least Square Objective

Objective: Given a training data

$$\{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\}$$

find w such that

$$L_{\mathsf{emp}}(w) = \sum_{n=1}^{N} (y_n - w^\mathsf{T} x_n)^2$$

is minimum.

Least Square Solution

We have

$$L_{\mathsf{emp}}(w) = \sum_{n=1}^{N} (y_n - w^\intercal x_n)^2$$

Solution

$$\frac{\partial L_{emp}}{\partial w} = \sum_{n=1}^{N} 2(y_n - w^\intercal x_n) \frac{\partial}{\partial w}(y_n - w^\intercal x_n) = 0$$

$$\implies \sum_{n=1}^{N} x_n(y_n - x_n^\intercal w) = 0 \qquad \text{(Note: } x_n^\intercal w = w^\intercal x_n\text{)}$$

$$\implies \sum_{n=1}^{N} x_n y_n - \sum_{n=1}^{N} x_n x_n^\intercal w = 0$$

$$\implies \sum_{n=1}^{N} x_n x_n^\intercal w = \sum_{n=1}^{N} x_n y_n$$

Least Square Solution (Cont...)

Objective: Given data $\{(x_n, y_n)\}_{n=1}^{N}$, find w such that minimize

$$L_{\text{emp}}(w) = \sum_{n=1}^{N} (y_n - w^\intercal x_n)^2$$

Final Solution:

$$\begin{aligned} w &= (\sum_{n=1}^{N} x_n x_n^\intercal)^{-1} \sum_{n=1}^{N} y_n x_n \\ &= (X^\intercal X)^{-1} X^\intercal Y \end{aligned}$$

When output is vector valued:

- The same solution holds if response y is vector valued i.e Y is $N \times K$ matrix (i.e K responses per input)
- In this case W will be $D \times K$ matrix

Some Remarks

- $X^\intercal X$ is a $D \times D$ matrix (D is the dimension of the data) and it can be very expensive to invert $X^\intercal X$
- $W = [b, w_1, \ldots, w_D]$, w_is can become very large trying to fit the training data
- IMPLICATION: The model becomes very complicated
- RESULT: The model overfits
- SOLUTION: Penalize large values of the parameter
- Regularization

Ridge Regression (Linear Regression with Regularization)

Modified Objective: Given data $\{(x_n, y_n)\}_{n=1}^N$, find w such that

$$L_{emp}(w) = \sum_{n=1}^{N} (y_n - w^\intercal x_n)^2 + \lambda ||w||^2$$

- Here $||w||^2 = w^\intercal w$
- λ is the hyperparameter, that controls amount of regularization.

Solution:

$$\frac{\partial L(W)}{\partial w} = \sum_{n=1}^{N} 2(y_n - w^\intercal x_n)(-x_n) + 2\lambda w = 0$$

Ridge Regression(cont...

$$
\begin{aligned}
&\implies \lambda(w) = \sum_{n=1}^{N} x_n(y_n - x_n^\intercal w) \\
&\implies \lambda(w) = \sum_{n=1}^{N} x_n y_n - \sum_{n=1}^{N} x_n x_n^\intercal w \\
&\implies \lambda W = X^\intercal Y - X^\intercal X W \\
&\implies \lambda W + X^\intercal X W = X^\intercal Y \\
&\implies (\lambda \mathtt{I}_d + X^\intercal X) W = X^\intercal Y \\
&\implies W = (X^\intercal X + \lambda \mathtt{I}_d)^{-1} X^\intercal Y
\end{aligned}
$$

Note that $X^\intercal X$ is a $D \times D$ matrix

On Regularization

Claim: Small weights, $w = (w_1, \ldots, w_d)$ ensure that the function $y = f(x) = w^\intercal x$ is *smooth*.

Justification:

- Let x_n, x_m be two D-dimensional feature vectors such that

$$x_{n_j} = x_{m_j}, \quad j = 1, 2, \ldots, D-1 \quad \text{but } |x_{n_D} - x_{m_D}| = \epsilon$$

  That is all the features are same except that last feature differs in x_n and x_n only by small amout of ϵ
- Now $|y_n - y_m| = \epsilon w_D$
- If w_D is very large then $|y_n - y_m|$ is large.
- This implies in this case $f(x) = w^\intercal x$ does not behave smoothly.

On Regularization (cont...)

- Hence regularization helps: which makes the individual components of w small.

- That is, **Do not** learn a model that gives a simple feature too much importance

- Regularization is very important when N is small and D is very large.

Ridge Regression Solution

- Directly with matrices

$$
\begin{aligned}
L(w) &= \frac{1}{2}(Y - XW)^\intercal(Y - XW) + \frac{\lambda}{2}W^\intercal W \\
\nabla L(w) &= -X^\intercal(Y - XW) + \lambda W = 0 \\
&\implies X^\intercal XW + \lambda W = X^\intercal Y \\
&\implies (X^\intercal X + \lambda \mathtt{I})W = X^\intercal Y \\
\text{Hence } W^* &= (X^\intercal X + \lambda \mathtt{I})^{-1} X^\intercal Y
\end{aligned}
$$

- One more advantage of Regression:
- If $X^\intercal X$ is not invertible, one can make $(X^\intercal X + \lambda \mathtt{I}_d)$ invertible.

Gradient Descent Solution for Least Squares

- We have the following least square solution

$$W^* = (X^\intercal X)^{-1} X^\intercal Y$$
$$W^*_{reg} = (X^\intercal X + \lambda \mathtt{I}_d)^{-1} X^\intercal Y$$

- Which involves inverting a $d \times d$ matrix.
- In the case of high dimensional data it is prohibitively difficult.
- Hence we turn to gradient Descent Solution.
  - Optimization methods that is based on gradients.
  - May stuck in a local optima.

Gradient Descent Procedure

Procedure:

1. Start with an initial value $w = w^{(0)}$
2. Update w by moving along the gradient of the loss function $L(L_{emp}$ or $L_{reg})$

$$w^{(t)} = w^{(t-1)} - \eta \frac{\partial L}{\partial w}\bigg|_{w=w^{(t-1)}}$$

3. Repeat until convergence.

Gradient Descent Procedure (contd...)

We have

$$\frac{\partial L}{\partial w} = \sum_{n=1}^{N} x_n(y_n - x_n^{\intercal} w)$$

Procedure:

1. Start with an initial value $w = w^{(0)}$
2. Update w by moving along the gradient of the loss function L(L_{emp} or L_{reg})

$$w^{(t)} = w^{(t-1)} - \eta \sum_{n=1}^{N} x_n(y_n - x_n^{\intercal} w^{(t-1)})$$

3. Repeat until convergence.

On Convexity

- The squared loss function in linear regression is convex.
  - With ℓ_2 regularizer it is strictly convex.

Convex Functions:

For scalar functions :	Convex if the second derivative is nonnegative everywhere
For vector valued :	Convex if Hessian is positive semi definite

On ℓ_1 Regularizer

ℓ_1 regularizer $\quad R(w) = ||w||_1 = \sum_{d=1}^{D} |w_d|$

- Promotes w to have very few non zero components.
- Optimization in this case is not straight forward.