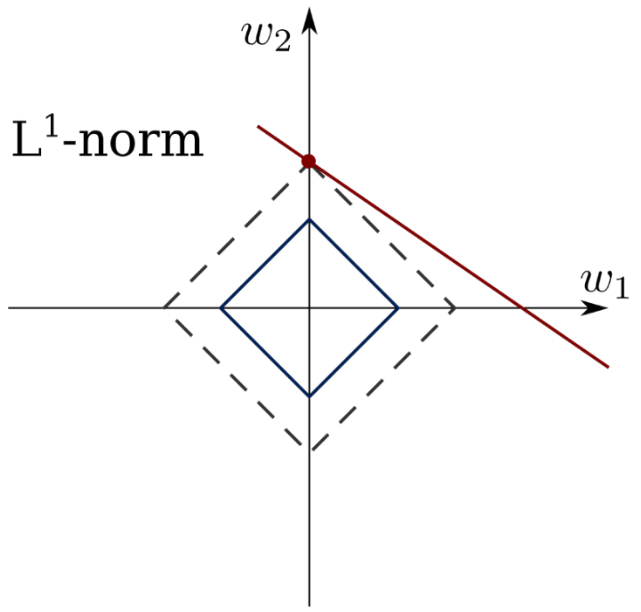
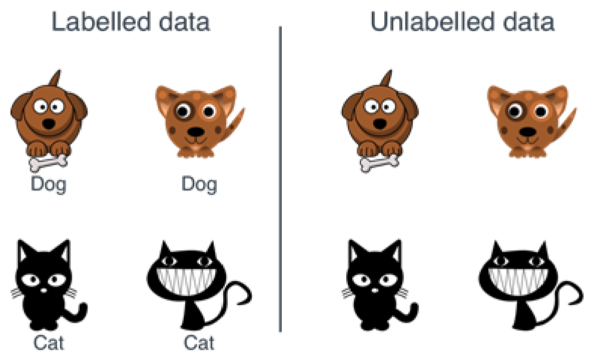
Siddhardhan

Math behind Lasso Regression

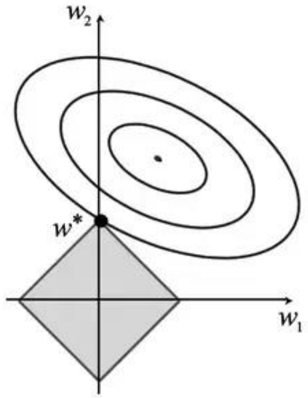


Lasso Regression

About Lasso Regression:



1. Supervised Learning Model
2. Regression model
3. **L**east **A**bsolute **S**hrinkage and **S**election **O**perator
4. Implements Regularization (L1) to avoid Overfitting



L1

Regularization

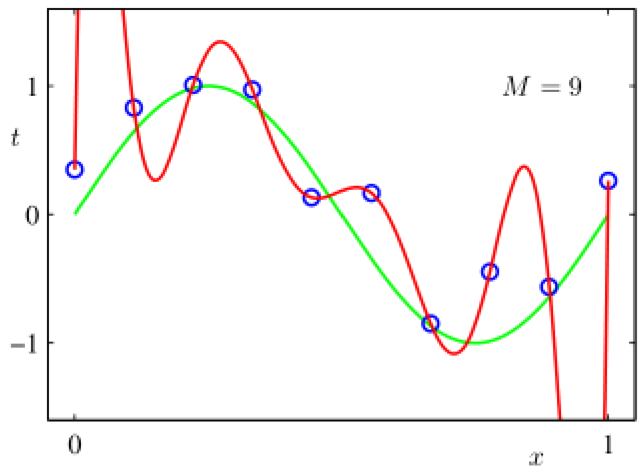
Regularization is used to reduce the overfitting of the model by adding a penalty term (λ) to the model. Lasso Regression uses L1 regularization technique.

The "penalty" term reduces the value of the coefficients or eliminate few coefficients, so that the model has fewer coefficients. As a result, overfitting can be avoided.

3rd order Polynomial equation : $y = ax^3 + bx^2 + cx + d$

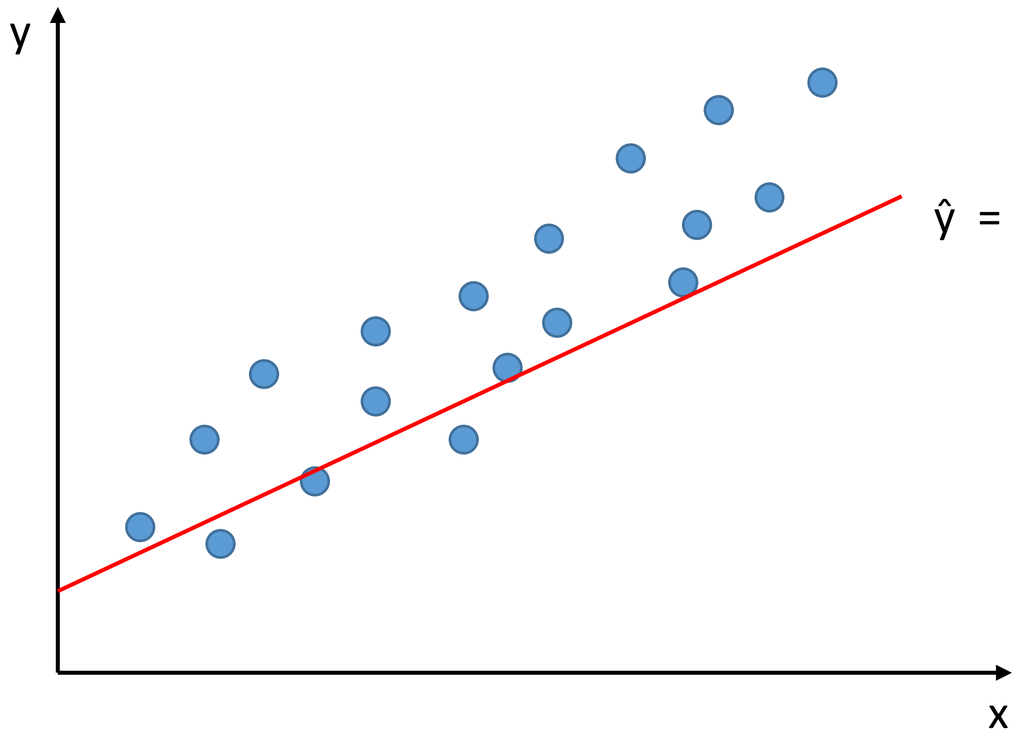
This Process is called as Shrinkage.

LASSO --> Least **A**bsolute **S**hrinkage and **S**election **O**perator



Linear Regression

Randomly assigned Parameters: w = 3; b = 2



$\hat{y} = 3X + 2$

x	y	$\hat{y}$
2	10	8
3	14	11
4	18	14
5	22	17
6	26	20

Cost Function

x	y	ŷ
2	10	8
3	14	11
4	18	14
5	22	17
6	26	20

$$\textit{Cost (J)} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

Cost $= [\ (10 - 8)^2 + (14 - 11)^2 + (18 - 14)^2 + (22 - 17)^2 + (26 - 20)^2\ ]\ /\ 5$

Cost $= [\ 4 + 9 + 16 + 25 + 36]\ /\ 5$

Cost $=$ 18

Low Cost value → High Accuracy

Lasso Regression

Cost Function for Lasso Regression :

$$J = \frac{1}{m}\left[\sum_{i=1}^{m} \left(y^{(i)} - \hat{y}^{(i)} \right)^2 + \lambda \sum_{j=1}^{n} w_j\right]$$

m --> Total number of Data Points

n --> Total number of input features

$y^{(i)}$ --> True Value

$\hat{y}^{(i)}$ --> Predicted Value

λ --> Penalty Term

w --> Parameter of the model

Boston House Price Dataset

The dataset used in this project comes from the UCI Machine Learning Repository. This data was collected in 1978 and each of the 506 entries represents aggregate information about 14 features of homes from various suburbs located in Boston.

crim	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	b	lstat	price
0.00632	18	2.31	0	0.538	6.575	65.2	4.09	1	296	15.3	396.9	4.98	24
0.02731	0	7.07	0	0.469	6.421	78.9	4.9671	2	242	17.8	396.9	9.14	21.6
0.02729	0	7.07	0	0.469	7.185	61.1	4.9671	2	242	17.8	392.83	4.03	34.7
0.03237	0	2.18	0	0.458	6.998	45.8	6.0622	3	222	18.7	394.63	2.94	33.4

$$J = \frac{1}{m}\left[\sum_{i=1}^{m}\left(y^{(i)} - \hat{y}^{(i)} \right)^2 + \lambda \sum_{j=1}^{n} w_j\right]$$