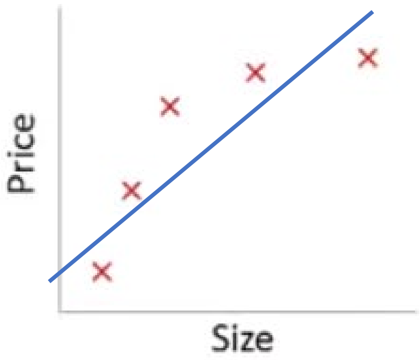
Regularization

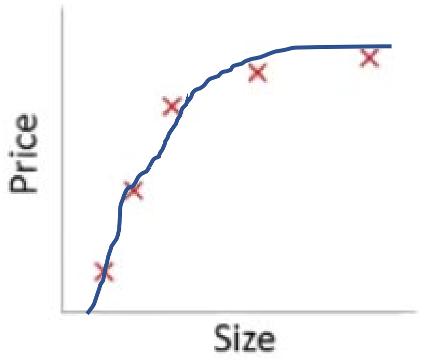
Recall the prediction problem using regression models (price of house)



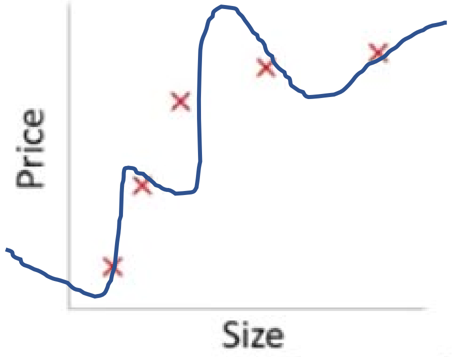
$\theta_0 + \theta_1 x$

“Underfit”

“High bias”



$\theta_0 + \theta_1 x + \theta_2 x^2$



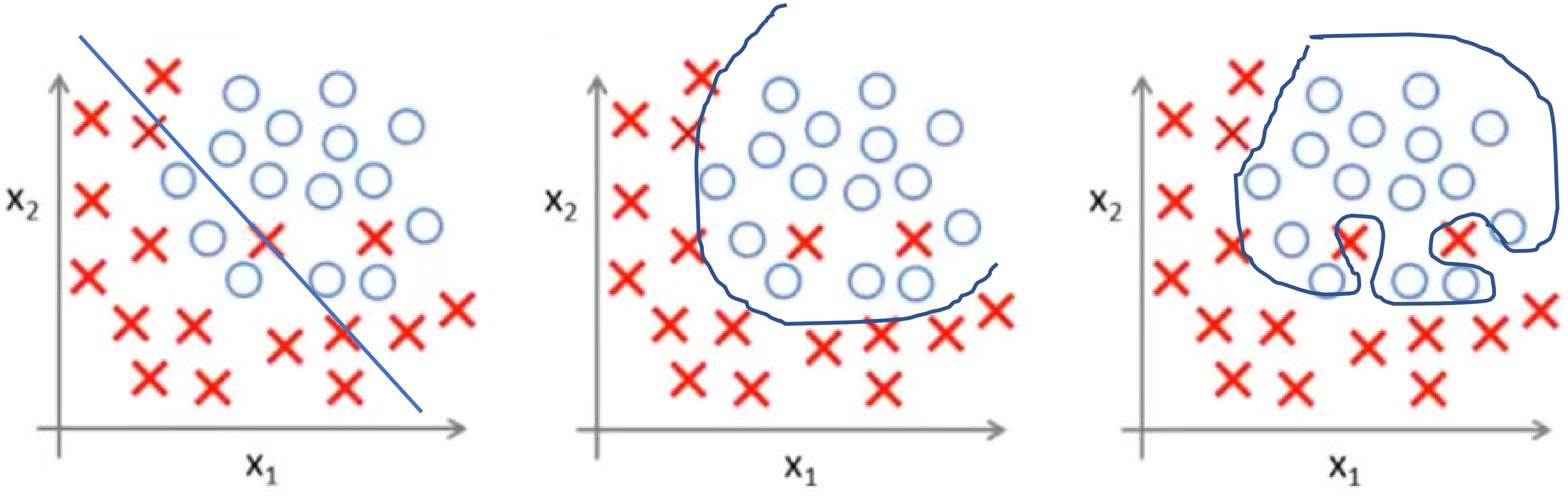
$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$

“Overfit”

“High variance”

Bias: errors caused by simplifying assumptions made by a model when approximating the target function (due to overly simple model)
Variance: measures how sensitive a model is in response to small fluctuations in the dataset (due to overly complex model)

Example: predict class labels using logistic regression



$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$

underfit

$g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2 + \theta_5 x_1 x_2)$

$g\big(\theta_0 + \theta_1 x_1 + \theta_2 x_1^2 + \theta_3 x_1^2 x_2 + \theta_4 x_1^2 x_2^2 + \theta_5 x_1^2 x_2^3 + \theta_6 x_1^3 x_2 + \cdots\big)$

overfit

The bias-variance trade-off

Classical view

- At the center of machine learning field
- A model should balance underfitting and overfitting, i.e., model should be rich enough to express underlying structure in data, and simple enough to avoid fitting “noise”

Modern development

- Rich models such as deep neural networks can fit or interpolate the date really well
- Current evidence show that they also are accurate on test data
- The “double-descent” performance curve instead of the U-shaped bias–variance trade-off curve, for beyond the point of interpolation

Ideas to overcome overfitting

1. Reduce the number of features
   - Manually select useful features to keep
   - Use model selection algorithms to determine model complexity
2. Regularization (error loss + regularization term)
   - Use all features but reduce magnitudes/values of some parameters θ. This works well for a large number of features so that each feature contributes a bit to predicting y
3. Drop out, early stopping, augmenting data…

Regularization:

Small values for parameters $\theta_0\ \theta_1 \ldots \theta_n$

- "simpler" hypothesis
- less prone to overfitting

Housing example

- Features: $x_0\ x_1 \ldots x_{100}$
- Parameters: $\theta_0\ \theta_1 \ldots \theta_{100}$

Introduce a new cost term into the cost function: the larger the $\theta_j's$, the higher the cost

$$J(\theta) = \frac{1}{2m}\left[\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2 + \lambda \parallel \theta \parallel_p\right]$$

$$J(\theta) = \frac{1}{2m}\left[\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2 + \lambda \parallel \theta \parallel_p \right]$$

Where for real number $p \geq 1$, the p−norm for vector θ

$$\parallel \theta \parallel_p = \left(\sum_{i=1}^{n} |\theta_i|^p\right)^{1/p}$$

$p = 1, \quad \rightarrow$ Lasso regression

$(L_1$ regularization$)$

$$J(\theta) = \frac{1}{2m}\left[\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2 + \lambda \sum_{i=1}^{n} |\theta_i|\right]$$

$p = 2, \quad \rightarrow$ Ridge regression

$(L_2$ regularization$)$

$$J(\theta) = \frac{1}{2m}\left[\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2 + \lambda \sum_{i=1}^{n} \theta_j^2\right]$$

For 0<$p < 1$, quasi-norm for vector θ, causes more elements of θ to be zeroed out

$L2$ regularization:

Introduce a new cost term into the cost function: the larger the $\theta_j's$, the higher the cost

Parameters θ are determined such that $J(\theta)$ is minimized

$$J(\theta) = \frac{1}{2m}\left[\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2 + \lambda\sum_{i=1}^{n}\theta_j^2\right]$$

Recall the linear regression problem with hypothesis below,

$$h_\theta(x) = \theta^T x = \theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$$

The parameters θ_0 θ_1 … θ_n are determined from minimizing the following cost function,

$$J(\theta) = \frac{1}{2m}\left[\sum_{i=1}^{m} \left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2\right]$$

Solution:

Update parameters θ_0 θ_1 … θ_n using gradient descent until convergence

Simultaneously update θ_j , $j = 0, 1, \ldots, n$, according to

$$\theta_j := \theta_j - \eta \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1, \ldots, \theta_n)$$

η is the learning rate

To be explicit, simultaneously update θ_0 and $\theta_j, j = 1, 2, \ldots, n$, according to

$$\theta_0 := \theta_0 - \eta \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right) x_0^{(i)}$$

$$\theta_j := \theta_j - \eta \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right) x_j^{(i)}$$

Now consider regularized linear regression – how to determine the parameters in a given hypothesis?

With the regularization term,
simultaneously update θ_0 and $\theta_j, j = 1, 2, \dots, n$, according to

$$\theta_0 := \theta_0 - \eta \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right) x_0^{(i)}$$

$$\theta_j := \theta_j - \eta \left[\frac{1}{m} \sum_{i=1}^{m} \left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right) x_j^{(i)} + \frac{\lambda}{m} \theta_j \right]$$

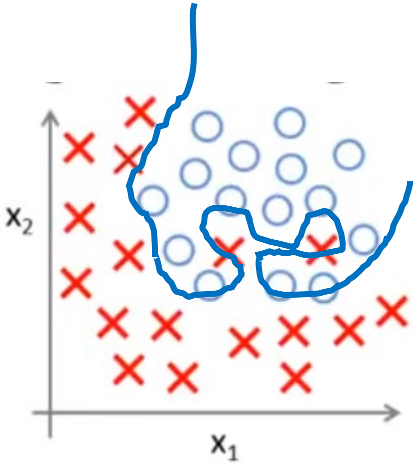
Notice that:

$$\theta_j := \theta_j \left(1 - \eta \frac{\lambda}{m}\right) - \eta \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right) x_j^{(i)}$$

$\left(1 - \eta \frac{\lambda}{m}\right)$ is usually less than 1 (weight decay) → smaller θ_j

The idea of regularization can be used to regulate other regression models

Regularized logistic regression



Consider a logistic regression model,

$$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_1^2 \& + \theta_3 x_1^2 x_2 + \theta_4 x_1^2 x_2^2 + \theta_5 x_1^2 x_2^3 + \cdots)$$

To regulate parameters $\theta_1 \ldots \theta_n$, minimize the following cost function

$$J(\theta) = -\left[\frac{1}{m}\sum_{i=1}^{m} y^{(i)} \log h_\theta\left(x^{(i)} + \left(1 - y^{(i)}\right) \& \log\left(1 - h_\theta\left(x^{(i)}\right)\right)\right] + \frac{\lambda}{2m}\sum_{j=1}^{n} \theta_j{}^2$$

$$\text{previously, } J(\theta) = \frac{1}{2m}\left[\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2 + \lambda \sum_{i=1}^{n} \theta_j^2\right]$$

With the regularization term,
simultaneously update θ_0 and $\theta_j, j = 1, 2, \ldots, n$, according to

$$\theta_0 := \theta_0 - \eta \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right) x_0^{(i)}$$

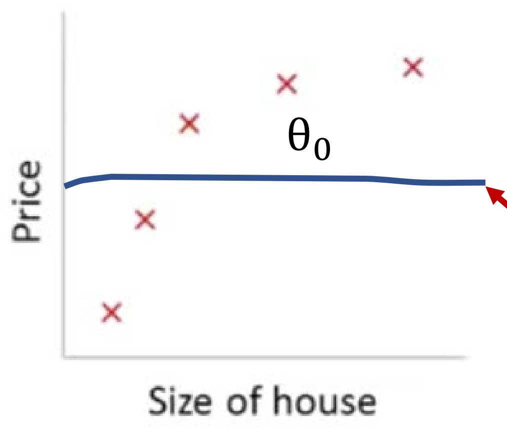
$$\theta_j := \theta_j - \eta \left[\frac{1}{m} \sum_{i=1}^{m} \left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right) x_j^{(i)} + \frac{\lambda}{m} \theta_j \right]$$

Note that this appears the same as in the case of linear regression, but they are different. Why?

Consider a regularized (generalized) linear regression problem,

$$J(\theta) = \frac{1}{2m}\left[\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2 + \lambda \sum_{i=1}^{n} \theta_j^2\right]$$

What if λ is very large, as large as say $\lambda = 10^{10}$?



$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_1^2 + \theta_3 x_1^2 x_2 + \theta_4 x_1^2 x_2^2$$

$\theta_1\ \theta_2 \ldots \theta_4 \rightarrow 0$

after regularization

$$h_\theta(x) = \theta_0$$

Severe under-fitting