COMP 5630/6630:Machine Learning

Lecture 4: Logistic Regression and Regularization

Slide Credit: : Drs. Aakur Andrew Ng, Byron Boots

Predicting House Type: Learn a Mapping from $x \rightarrow y$

Dataset of the living areas, bedrooms, prices, and types of 47 houses

Living area (ft^2)	# bedrooms	Price (1000$s)	Type
1643	4	256	Apartment
1356	3	202	Apartment
1678	3	287	House
…	…	…	
3000	4	400	House

X = Input features

y = Class label

y = Class label
1 = House
0 = Apartment/ not a house

House Type Prediction

- Supervised Learning
  - Given a *training dataset*, learn a *mapping (hypothesis, h)* from x→ y, where y is labelled
  - Goal: Given a new datapoint, x (*test data*), predict the most accurate output, y, using the *learned hypothesis, h*
    - *learned mapping = trained model*

x → Classification → y

Discreet

y = 0: Apartment/
not a house

y = 1: House

Logistic Regression: Hypothesis

- Recap: Linear regression hypothesis

$$h(x) = \sum_{i=0}^{n} \theta_i x_i = \theta^T x,$$

- *How does the logistic regression hypothesis differ from that of linear regression?*

Logistic Regression: Hypothesis

- Recap: Linear regression hypothesis

$$h(x) = \sum_{i=0}^{n} \theta_i x_i = \theta^T x,$$

- *How does the logistic regression hypothesis differ from that of linear regression?*

- Does not make sense to have predictions greater than 1 or less then 0.
  - Since the labels of y are 1 and 0 and y ∈ {0, 1}.

Logistic Regression: $0 \leq h_\theta(x) \leq 1$

Logistic Regression Model

- Goal
  - Define a function to satisfy $0 \leq h_\theta(x) \leq 1$
  - Take a probabilistic approach to represent *h = p(y =1|* ***x; θ****)*

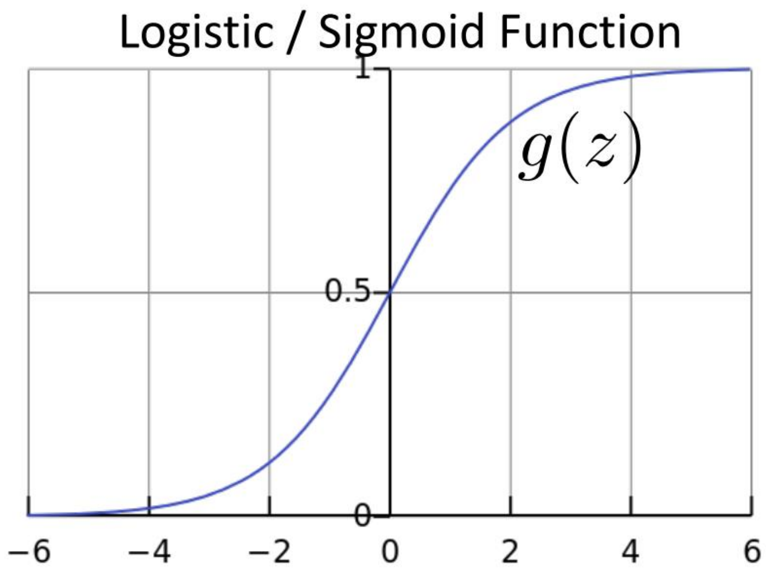
- Logistic regression model

$$h_{\boldsymbol{\theta}}(\boldsymbol{x}) = g\left(\boldsymbol{\theta}^\mathsf{T}\boldsymbol{x}\right)$$

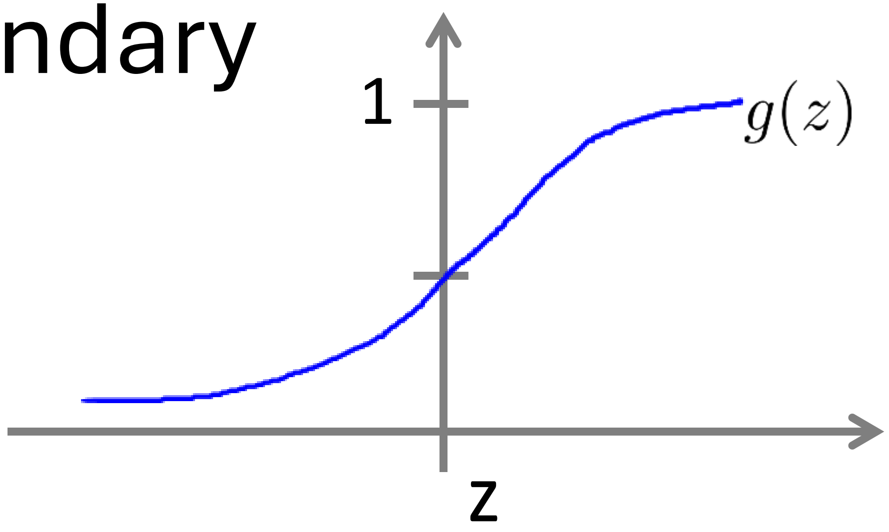
$$g(z) = \frac{1}{1+e^{-z}}$$

$$\boxed{h_{\boldsymbol{\theta}}(\boldsymbol{x}) = \frac{1}{1+e^{-\boldsymbol{\theta}^\mathsf{T}\boldsymbol{x}}}}$$

$$\theta^T x = \theta_0 + \sum_{j=1}^{n} \theta_j x_j$$



Logistic Regression Decision Boundary



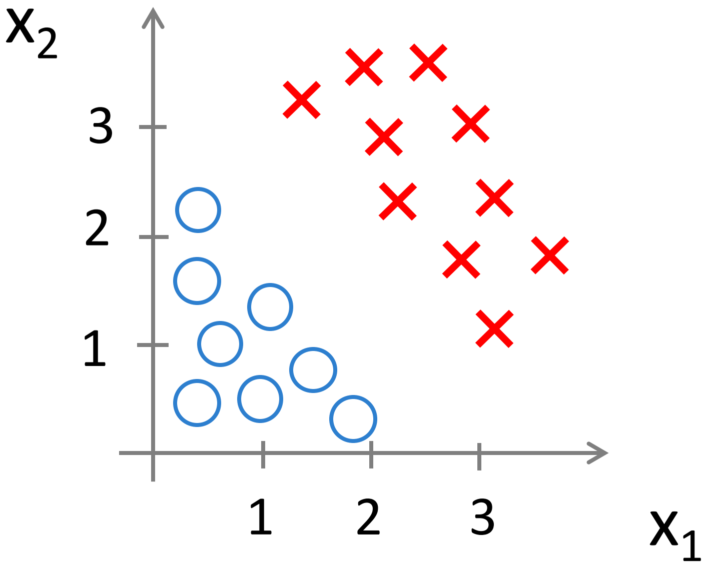
$$h_\theta(x) = g(\theta^T x)$$

$$g(z) = \frac{1}{1+e^{-z}}$$

Suppose predict "$y = 1$" if $h_\theta(x) \geq 0.5$

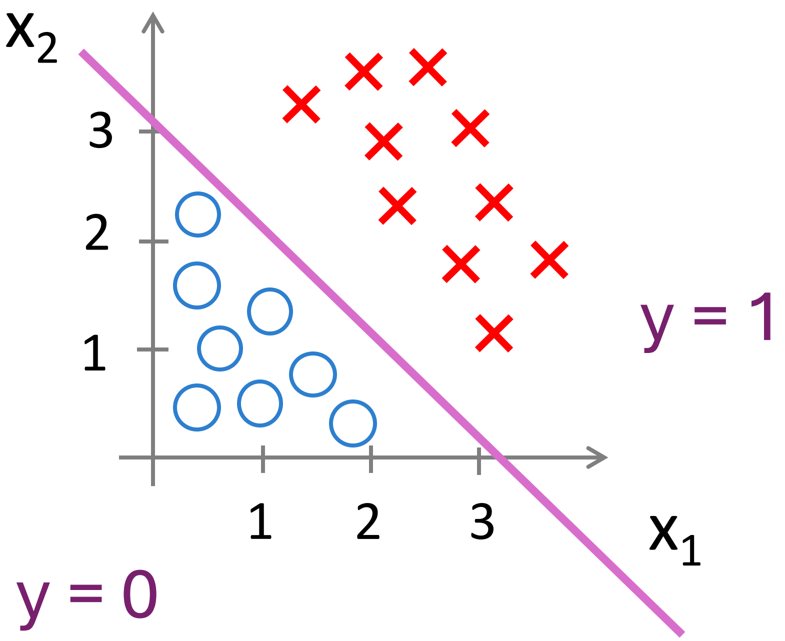
predict "$y = 0$" if $h_\theta(x) < 0.5$

Logistic Regression Decision Boundary



Any Idea?

Logistic Regression Decision Boundary



Predict y = 1 if $-3 + x_1 + x_2 \geq 0$

Logistic Regression: Cost/ Objective Function to get Θ

$$h_{\boldsymbol{\theta}}(\boldsymbol{x}) = \frac{1}{1+e^{-\boldsymbol{\theta}^{\mathsf{T}}\boldsymbol{x}}}$$

$$\theta^T x = \theta_0 + \sum_{j=1}^{n} \theta_j x_j$$

- Gradient Descent!
- What do we need for gradient descent?
  - An objective function J(θ)
  - A learning rate α
  - An initial "guess" for θ called θ_j
- Then, update θ_j until convergence as follows

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

Defining the Objective Function J(θ)

- Let say,

$$P(y=1 \mid x;\theta) = h_\theta(x)$$

Chance of predicting a house given X

$$P(y=0 \mid x;\theta) = 1 - h_\theta(x)$$

Chance of predicting an apartment given X

Sum of two probabilities must equal to one

- Combining:

$$p(y \mid x;\theta) = (h_\theta(x))^y \, (1 - h_\theta(x))^{1-y}$$

Plugging y = 1 into $(h_\theta(x))^y \, (1 - h_\theta(x))^{1-y}$ yields $h_\theta(x)$

Plugging y = 0 into $(h_\theta(x))^y \, (1 - h_\theta(x))^{1-y}$ yields $1 - h_\theta(x)$

Defining the Objective Function J(θ)

- Given: $$p(y \mid x;\theta) = (h_\theta(x))^y \, (1 - h_\theta(x))^{1-y}$$

- We want to estimate θ that will capture the dependency between y and x.
- We will instead call it the ***likelihood function*** that **maximizes** $p(y|\boldsymbol{x};\ \theta)$ and is given by

$$L(\theta) = \prod_{i=1}^{m} p(y^{(i)} \mid x^{(i)};\theta) = \prod_{i=1}^{m} \left(h_\theta(x^{(i)})\right)^{y^{(i)}} \left(1 - h_\theta(x^{(i)})\right)^{1-y^{(i)}}$$

Defining the Objective Function J(θ)

- How can we get θ?
- The principal of ***maximum likelihood***
  - Choose θ that makes the data as high probability as possible
  - Choose θ to maximize L(θ).
- Instead of maximizing L(θ), we can also maximize any strictly increasing function of L(θ).
  - We will maximize the log likelihood ℓ(θ):

$$\ell(\theta) = \log L(\theta) = \sum_{i=1}^{m} y^{(i)} \log h(x^{(i)}) + (1 - y^{(i)}) \log(1 - h(x^{(i)}))$$

Recap: House Type Prediction Problem

- Given a *training dataset*, learn a mapping *(hypothesis, h)* from x→ y, where y is labelled
  - Chance of predicting a house given X
  - Chance of predicting an apartment given X

y = 0: Apartment/
not a house
y = 1: House

- h→ *Choose θ* that will capture the dependency between y and x
  - Define likelihood
  - Take log likelihood

- Goal of learning *h* :
  - Maximize log likelihood
  - Given a new datapoint, x (*test data*), predict the most accurate output, y, using the *learned hypothesis, h*

How do we get θ?

- Gradient Descent!
- Then, update θ_j until convergence as follows

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

- So what is $\frac{\partial}{\partial \theta_j} J(\theta)$?

Gradient Descent J(θ)

- Goal
  - *Maximize* the log likelihood $= \sum_{i=1}^{m} y^{(i)} \log h(x^{(i)}) + (1 - y^{(i)}) \log(1 - h(x^{(i)}))$

  - Equivalently, *minimize* J(θ)

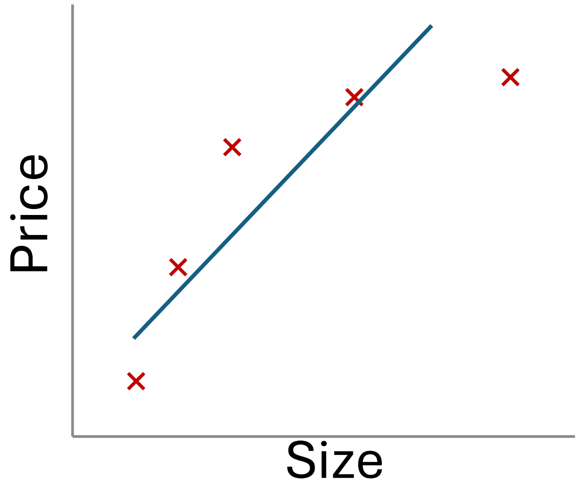
$$J(\theta) = -\frac{1}{m}\left[\sum_{i=1}^{m} y^{(i)} \log h_\theta(x^{(i)}) + (1 - y^{(i)}) \log\left(1 - h_\theta(x^{(i)})\right)\right]$$

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

Derivation is in the handout

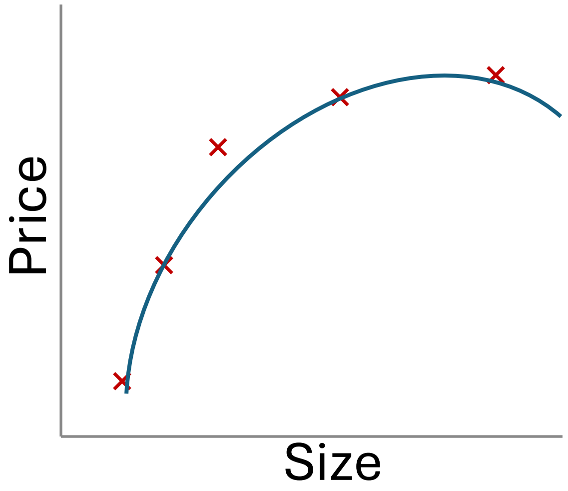
Regularization

Example: Linear regression (housing prices)



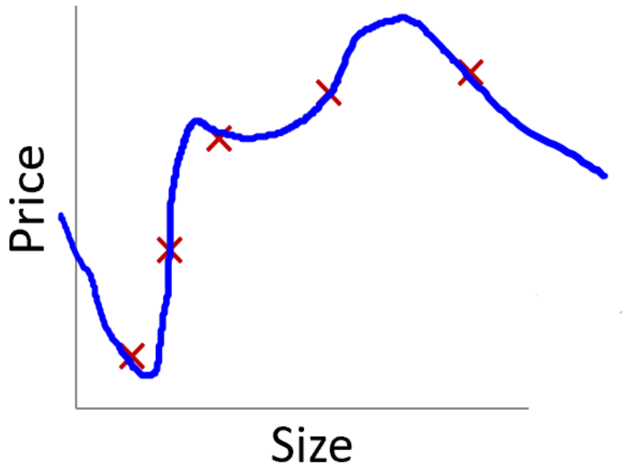
$$\theta_0 + \theta_1 x$$

Underfit



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

Just right

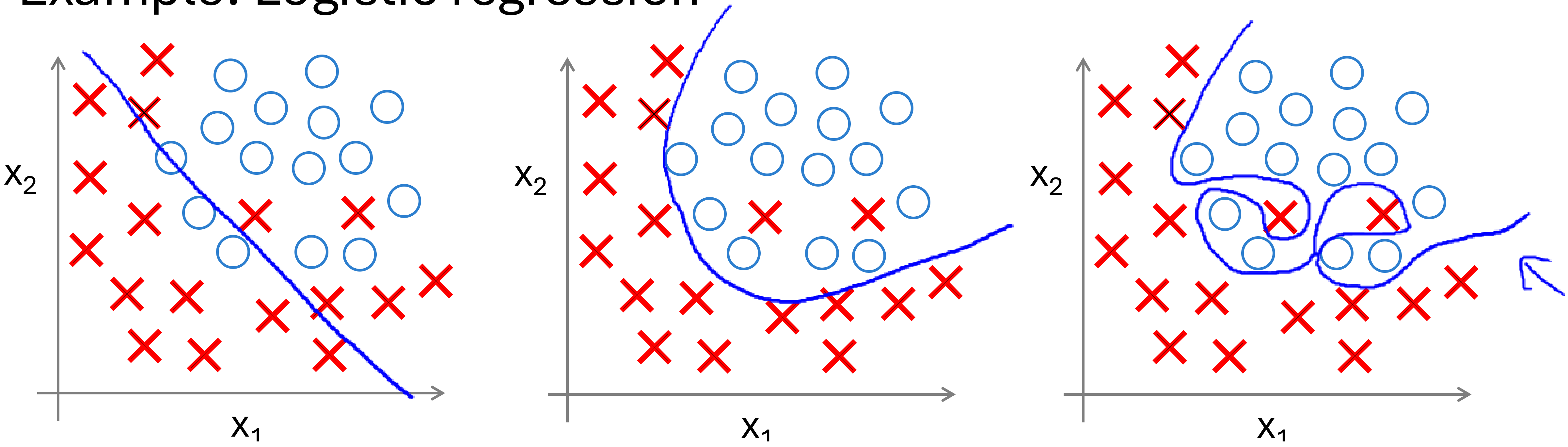


$$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$$

Overfit

Overfitting: If we have too many features, the learned hypothesis may fit the training set very well, but fail to generalize to new examples (predict prices on new examples).

Example: Logistic regression



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

(g= sigmoid function)

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

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

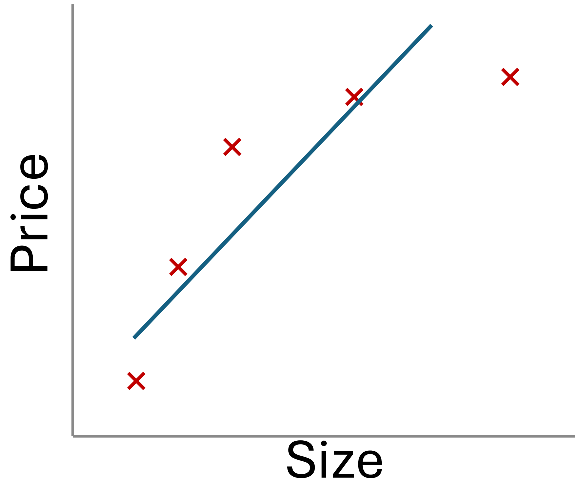
Addressing Overfitting

Options:

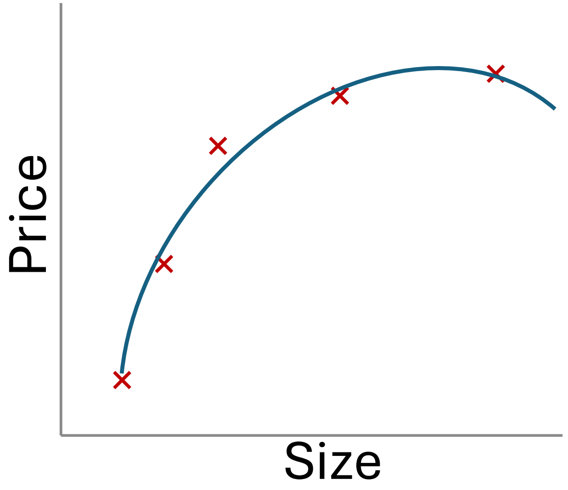
1. Reduce number of features.
   - Manually select which features to keep.
   - Model selection algorithm (later in course).

2. Regularization
   - Keep all the features, but reduce magnitude/values of parameters θ_j
   - Works well when we have a lot of features, each of which contributes a bit to predicting y

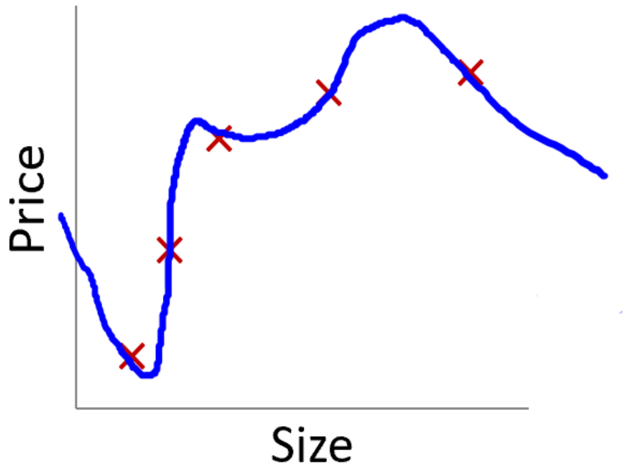
Intuition of Regularization



$\theta_0 + \theta_1 x$



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



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

Suppose we penalize and make Θ_3 and Θ_4 really small.

Minimize $J(\Theta)$

$$J(\Theta) = \frac{1}{2} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2 + \boxed{1000\,\Theta_3 + 2000\,\Theta_4} \quad \sim 0$$

Regularization in Linear Regression

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

$$\min_{\theta} J(\theta)$$

Normal Equation with Regularization

- **X=** $\begin{bmatrix} 1 & 1643 & 4 \\ 1 & 1356 & 3 \\ 1 & 1678 & 3 \end{bmatrix}$ $\quad \boldsymbol{y} = \begin{bmatrix} 256 \\ 202 \\ 287 \end{bmatrix}$

Dimension of X, mxn = m examples, n features
Dimension of y = mx1

x_0	x_1 = Living area (ft^2)	x_2=#bedrooms	y= Price(1000$s)
1	1643	4	256
1	1356	3	202
1	1678	3	287

- Regularization: add an extra Identity matrix (n+1, n+1) to (X^TX) where n =# features, and extra 1 denotes the extra column of 1s for bias term.

$$\Theta = \left(X^TX + \lambda \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ . & . & . & 1 & . & . & . \\ . & . & . & . & 1 & . & . \\ . & . & . & . & . & 1 & . \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}_{(n+1,n+1)} \right)^{-1} X^TY$$

Non regularized: $\theta = (X^TX)^{-1}X^Ty$

Logistic Regression with Regularization

- log L(θ) + $\lambda \sum_{j=1}^{p} |\beta_j|$

  L1 penalty, known as "Lasso regression"

- log L(θ) + $\lambda \sum_{j=1}^{p} (\alpha \beta_j^2 + (1-\alpha)|\beta_j|)$

  L1 and L2 penalty, known as "Elastic net"