Maior Dúvida da Aula

Logistic Regression

MC886/MO444, September 8, 2022

Regressão Logística *vs.* Regressão Linear

1. Não entendi quando devo aplicar regressão linear ou quando devo aplicar regressão logística para solucionar um problema dado. Imagino que no mundo real eu não irei ganhar um enunciado falando para resolver com tal regressão, então como escolher a melhor para o caso? Devo tentar usar as duas e ver qual sai melhor?

Regressão

Linear

resposta é **contínua**

Logística

resposta é **categórica**

Binomial
categórica binária

Multinomial
categórica 3+

Mínimos Locais

2. Não entendi o motivo de ser dito que regressão logística não possui um mínimo global, apenas mínimos locais. O menor valor, entre todos os mínimos locais, não seria considerado o mínimo global?

3. Como é um caso não convexo, se o tempo de processamento do modelo não for tão alto, faria sentido eu fazer vários testes com thetas iniciais diferentes para ver se encontro um mínimo melhor pra mesma configuração de parâmetros? porque o ponto de partida pode me levar a um lugar diferente, correto?

   https://math.stackexchange.com/questions/1582452/logistic-regression-prove-that-the-cost-function-is-convex

Classificação Multiclasse

4. Não entendi muito bem a técnica de One-vs-All para classificação de diversas classes. Por essa técnica teríamos diversas funções de custo mínimo para cada uma das classes? Se sim, como fazer para consolidar essas equações em apenas um modelo?
5. Notei que foi mencionado apenas o one-vs-all, mas existe classificação all-vs-all? Se sim, onde se aplica?
6. Para regressão logística multiclasse ainda utilizamos a função sigmóide? Poderíamos usar uma softmax, por exemplo?
   Softmax Regression https://web.stanford.edu/~jurafsky/slp3/5.pdf (Seção 5.3.1)

Perguntas Gerais

7. Em uma aplicação na qual um resultado falso positivo seja menos prejudicial do que um falso negativo, é comum (ou certo) alterar o valor do threshold classifier output para um valor maior que 0.5? Ou vice-versa ?

8. Existe algum tipo de técnica, passo a passo ou conjunto de princípios para se fazer uma boa análise exploratória? Em geral os cursos de machine learning disponíveis na internet falam muito sobre modelos, porém não ensinam bem como fazer a análise exploratória.

Regularization

Machine Learning

(Largely based on slides from Andrew Ng)

Prof. Sandra Avila

Institute of Computing (IC/Unicamp)

MC886/MO444, September 8, 2022

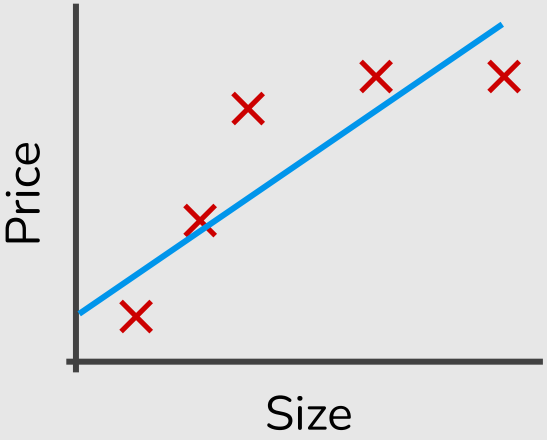
Today's Agenda

- Regularization
    - The Problem of Overfitting
    - Diagnosing Bias vs. Variance
    - Cost Function
    - Regularized Linear Regression
    - Regularized Logistic Regression

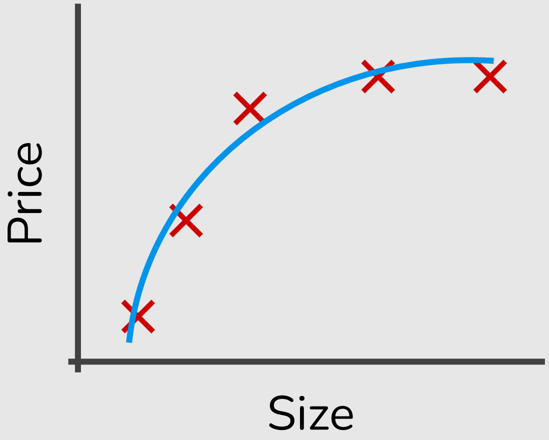
The Problem of Overfitting


THE BEST WAY TO
EXPLAIN OVERFITTING
imgflip.com

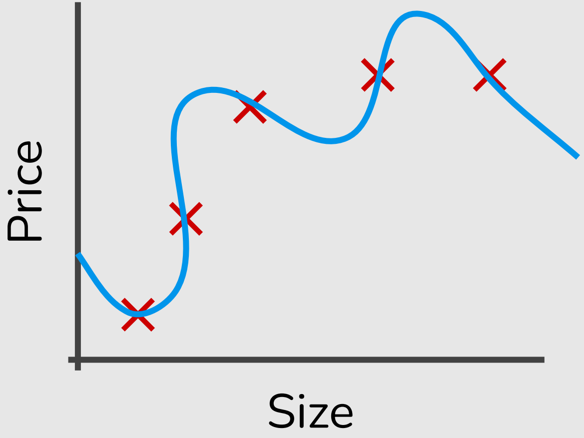
Example: Linear Regression



$\theta_0 + \theta_1 x$



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



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

Underfitting

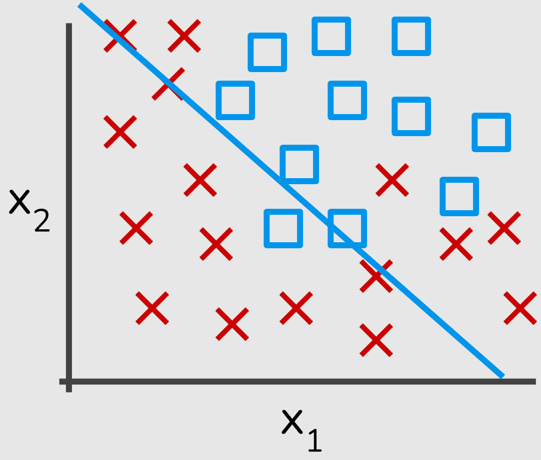
High bias

Overfitting

High variance

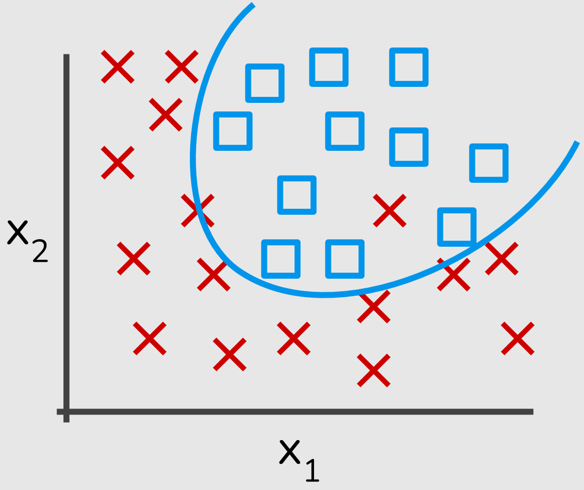
Example: Logistic Regression

Overfitting
High variance

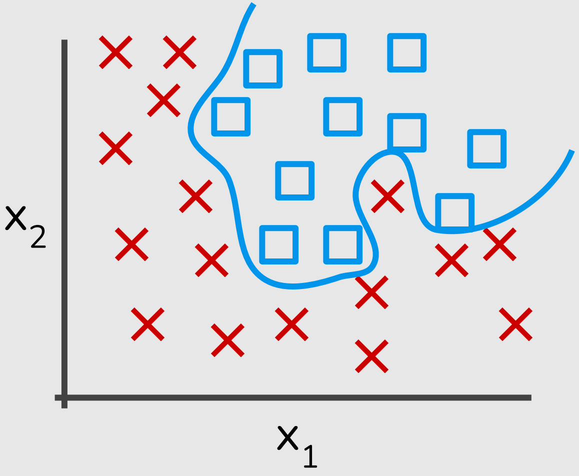


$$g(\theta_0+\theta_1 x_1+\theta_2 x_2)$$

Underfitting
High bias



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



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

The Bias/Variance Tradeoff

The Bias/Variance Tradeoff

A model's generalization error can be expressed as the sum of **three** very different errors:

- Bias
- Variance
- Irreducible error

The Bias/Variance Tradeoff

A model's generalization error can be expressed as the sum of **three** very different errors:

- **Bias**
  - Due to wrong assumptions, such as assuming that the data is linear when it is actually quadratic.
  - A **high-bias** model is most likely to **underfit** the training data.
- Variance
- Irreducible error

The Bias/Variance Tradeoff

A model's generalization error can be expressed as the sum of **three** very different errors:

- Bias
- **Variance**
  - Due to the model's excessive sensitivity to small variations in the training data.
  - A model with many degrees of freedom is likely to have **high variance**, and thus to **overfit** the training data.
- Irreducible error

The Bias/Variance Tradeoff

A model's generalization error can be expressed as the sum of **three** very different errors:

- Bias
- Variance
- **Irreducible error**
  - Due to the noisiness of the data itself.
  - The only way to reduce this part of the error is to clean up the data.

The Bias/Variance Tradeoff

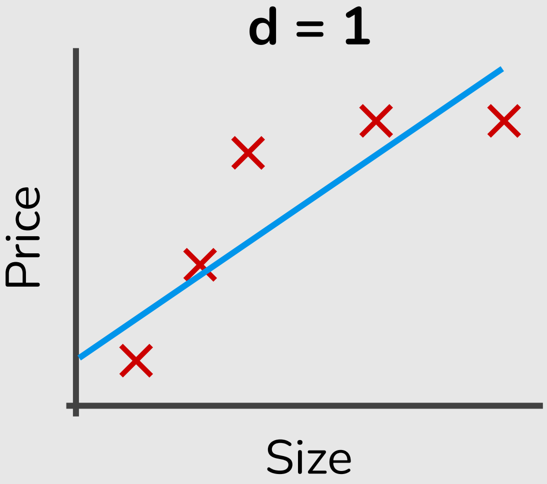
Increasing a model's complexity will typically increase its variance and reduce its bias.

Reducing a model's complexity increases its bias and reduces its variance.

This is why it is called a **tradeoff**.

Diagnosing Bias vs. Variance

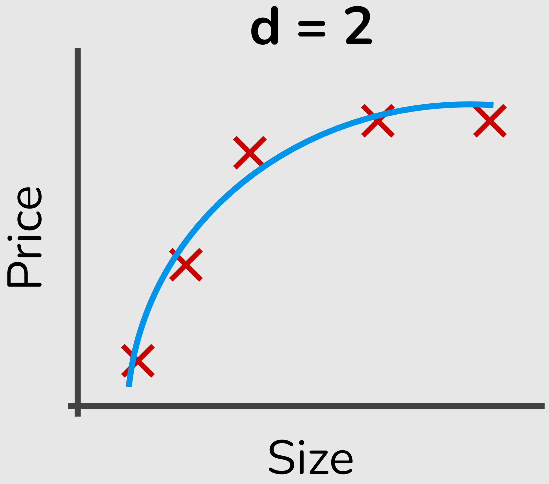
Bias/Variance



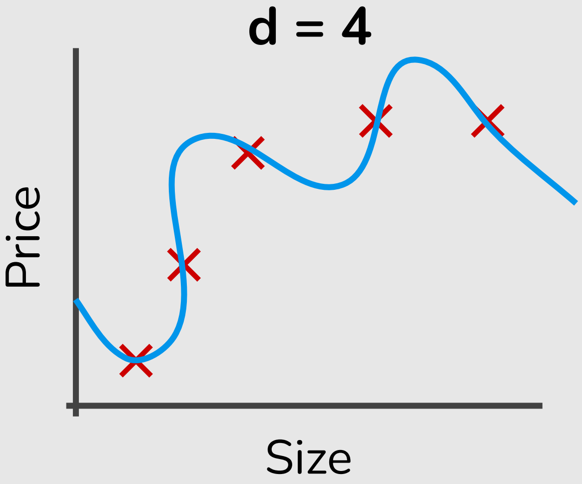
$$\theta_0 + \theta_1 x$$

Underfitting

High bias



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



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

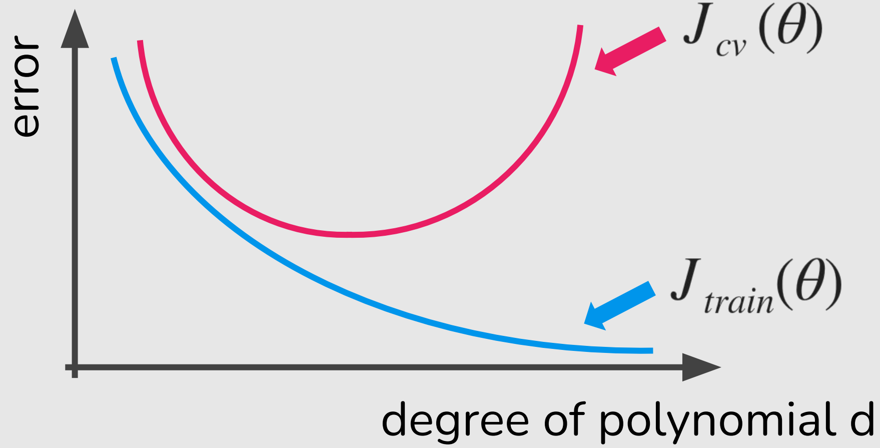
Overfitting

High variance

Bias/Variance

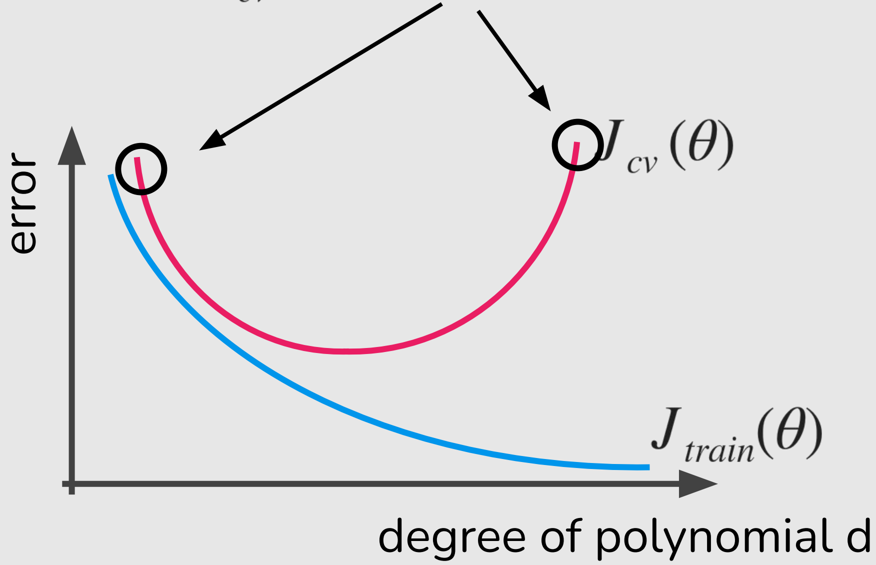
Training error: $J_{train}(\theta) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$

Cross-validation error: $J_{cv}(\theta) = \frac{1}{2m_{cv}} \sum_{i=1}^{m_{cv}} (h_\theta(x_{cv}^{(i)}) - y_{cv}^{(i)})^2$



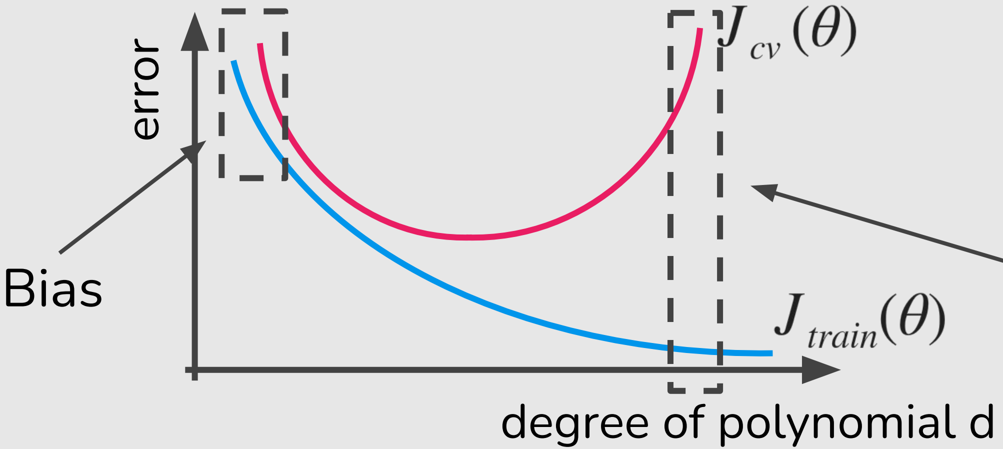
Diagnosing Bias vs. Variance

Suppose your learning algorithm is performing less well than you were hoping: $J_{cv}(\theta)$ is high. Is it a bias problem or a variance problem?



Diagnosing Bias vs. Variance

Suppose your learning algorithm is performing less well than you were hoping: $J_{cv}(\theta)$ is high. Is it a bias problem or a variance problem?



Bias (underfit):

$J_{train}(\theta)$ will be high

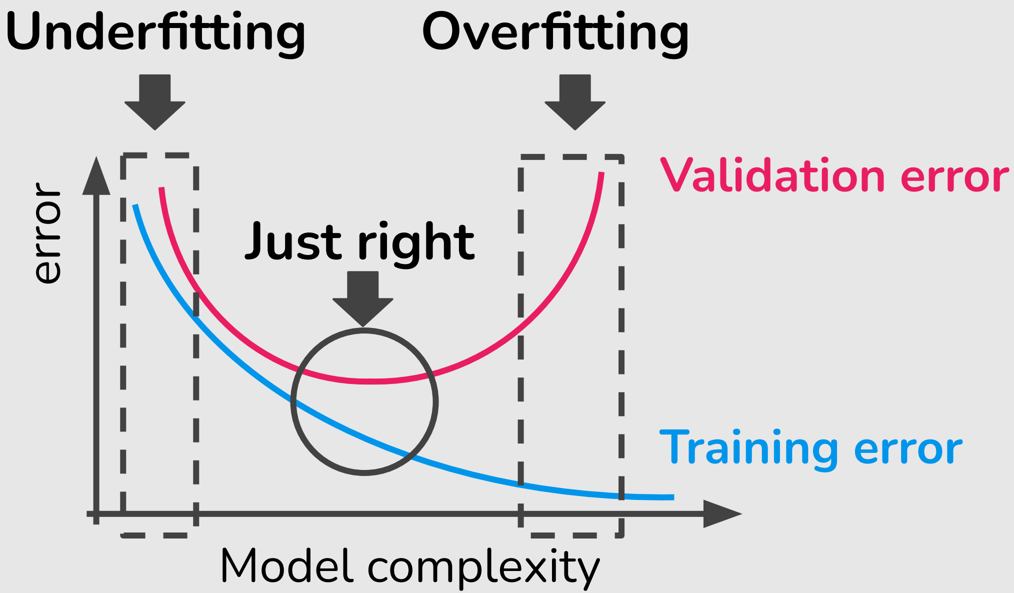
$J_{cv}(\theta) \approx J_{train}(\theta)$

Variance (overfit):

$J_{train}(\theta)$ will be low

$J_{cv}(\theta) \gg J_{train}(\theta)$

Diagnosing Bias vs. Variance



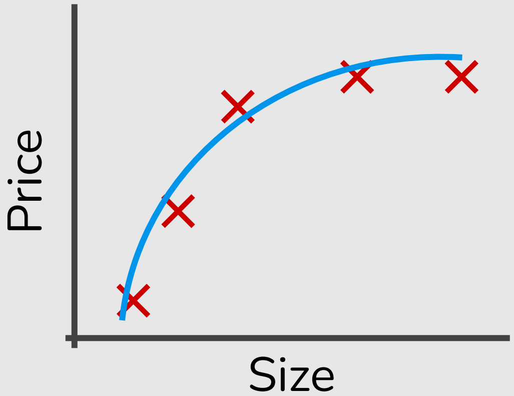
Understanding the Bias-Variance Tradeoff: http://scott.fortmann-roe.com/docs/BiasVariance.html

Underfitting

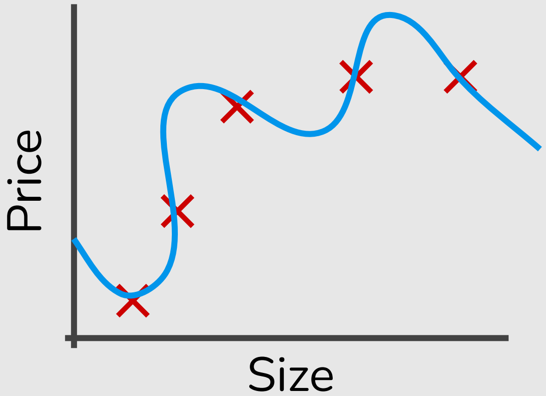
Overfitting

Cost Function

Intuition

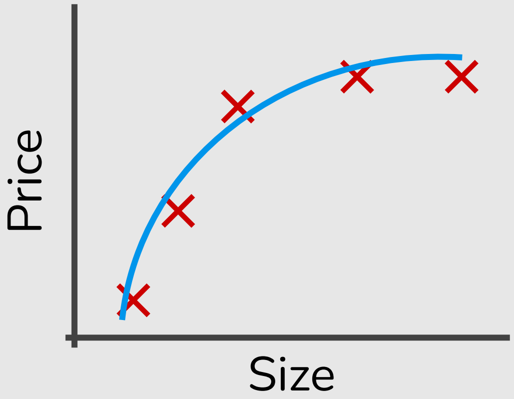


$\theta_0 + \theta_1 x + \theta_2 x^2$

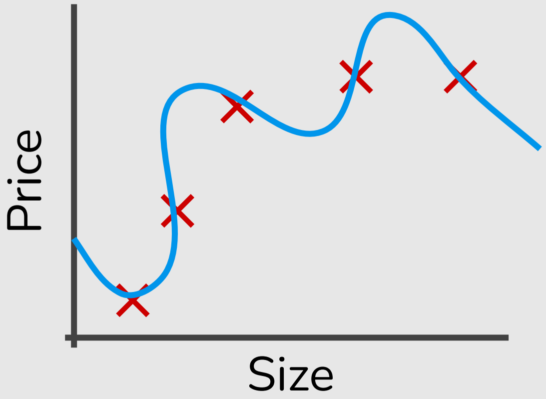


$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$

Intuition



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

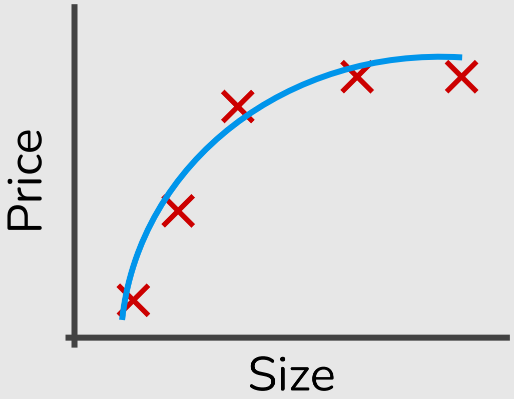


$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$

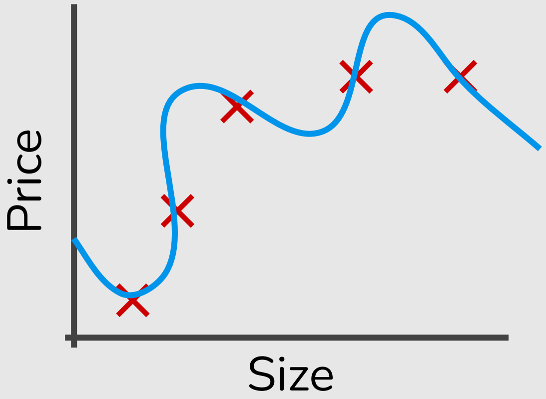
Suppose we penalize and make θ_3, θ_4 really small.

$$\min_{\theta} \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$$

Intuition



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

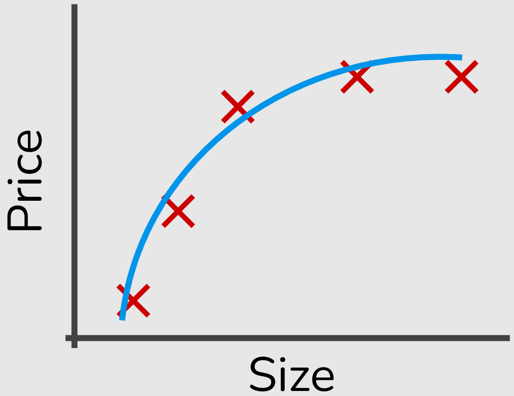


$$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$$

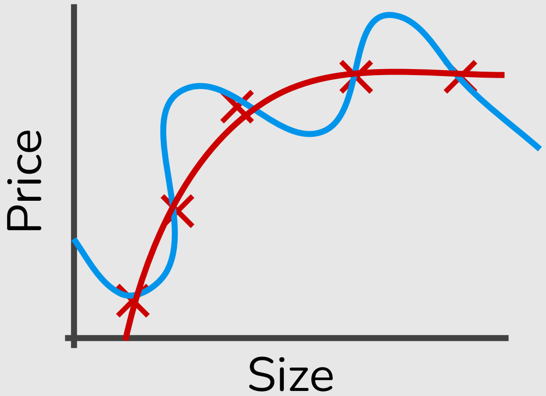
Suppose we penalize and make θ_3, θ_4 really small.

$$\min_\theta \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2 + 1000\ \theta_3^2 + 1000\ \theta_4^2$$

Intuition



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



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

$\theta_3 \approx 0$

$\theta_4 \approx 0$

Suppose we penalize and make θ_3, θ_4 really small.

$$\min_{\theta} \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2 + 1000\ \theta_3^2 + 1000\ \theta_4^2$$

Regularization

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

- "Simpler" hypothesis
- Less prone to overfitting

Regularization

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

- "Simpler" hypothesis
- Less prone to overfitting

Housing

- Features: $x_0, x_1, \ldots, x_{100}$
- Parameters: $\theta_0, \theta_1, \theta_2, \ldots, \theta_{100}$

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

Regularization

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

- "Simpler" hypothesis
- Less prone to overfitting

Housing

- Features: $x_0, x_1, \ldots, x_{100}$
- Parameters: $\theta_0, \theta_1, \theta_2, \ldots, \theta_{100}$

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

Regularization

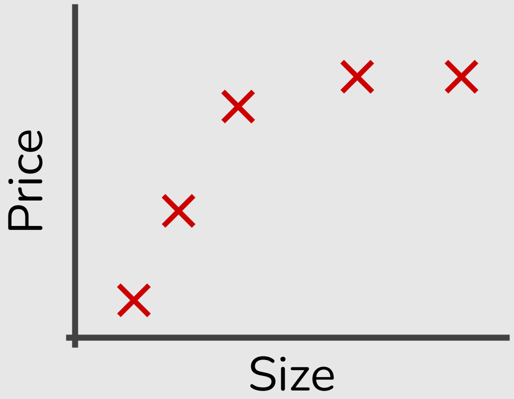
Regularization parameter

$$J(\theta) = \frac{1}{2m}\Bigg[\underbrace{\sum_{i=1}^{m}\big(h_\theta(x^{(i)}) - y^{(i)}\big)^2}_{\text{to fit the training data well}} + \boxed{\lambda}\underbrace{\sum_{j=1}^{n}\theta_j^2}_{\text{to keep the parameters small}}\Bigg]$$

In regularized linear regression, we choose θ to minimize

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

What if λ is set to an extremely large value (perhaps for too large for our problem, say $\lambda = 10^{10}$)?

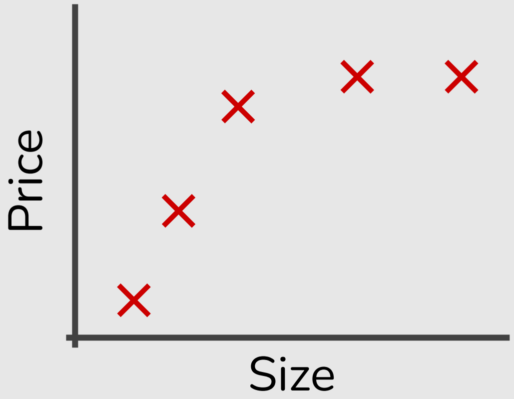


$$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$$

In regularized linear regression, we choose θ to minimize

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

What if λ is set to an extremely large value (perhaps for too large for our problem, say $\lambda = 10^{10}$)?

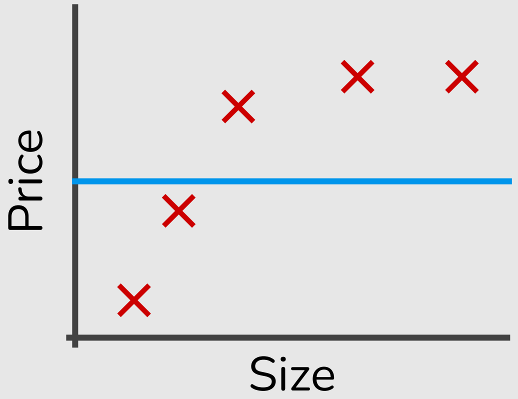


$$\theta_0 + \cancel{\theta_1 x} + \cancel{\theta_2 x^2} + \cancel{\theta_3 x^3} + \cancel{\theta_4 x^4}$$

In regularized linear regression, we choose θ to minimize

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

What if λ is set to an extremely large value (perhaps for too large for our problem, say $\lambda = 10^{10}$)?



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

Regularization

- ℓ_2 or Ridge Regression (also called Tikhonov regularization)

$$J(\theta) = \frac{1}{m} \sum_{i=1}^{m} Cost(h_\theta(x^{(i)}), y^{(i)}) + \frac{\lambda}{2m} \sum_{j=1}^{n} \theta_j^2$$

- ℓ_1 or Least Absolute Shrinkage and Selection Operator Regression (usually simply called Lasso Regression)

$$J(\theta) = \frac{1}{m} \sum_{i=1}^{m} Cost(h_\theta(x^{(i)}), y^{(i)}) + \frac{\lambda}{m} \sum_{j=1}^{n} |\theta_j|$$

Regularized Linear Function

Gradient Descent

repeat {

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

(simultaneously update θ_j for $j = 0, 1, \ldots, n$)

}

Gradient Descent

repeat {

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

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

} (simultaneously update θ_j for $j = \cancel{0}, 1, \ldots, n$)

Gradient Descent

repeat {

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

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

} (simultaneously update θ_j for $j = \cancel{0,}\ 1, \ldots, n$)

Gradient Descent

repeat {

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

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

} (simultaneously update θ_j for $j = \cancel{0}, 1, \ldots, n$)

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

Gradient Descent

repeat {

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

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

} (simultaneously update θ_j for $j =$ $1, \ldots, n$)

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

Normal Equation

$$X = \begin{bmatrix} — (x^{(1)})^{\mathrm{T}} — \\ — (x^{(2)})^{\mathrm{T}} — \\ — \quad \vdots \quad — \\ — (x^{(m)})^{\mathrm{T}} — \end{bmatrix} \qquad y = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(m)} \end{bmatrix} \qquad \theta = (X^T X)^{-1} X^T y$$

Normal Equation

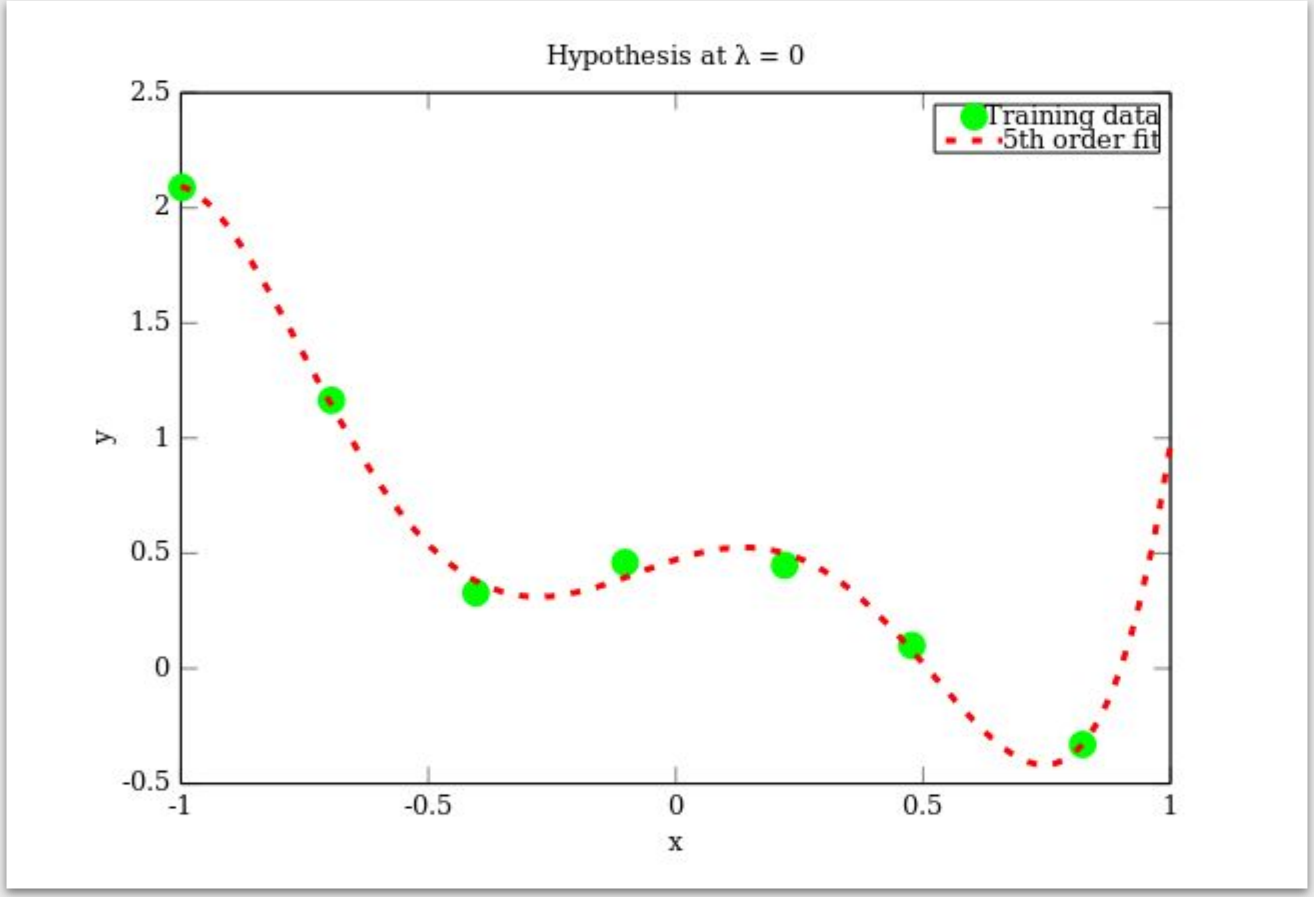
$$X = \begin{bmatrix} — (x^{(1)})^{\mathrm{T}} — \\ — (x^{(2)})^{\mathrm{T}} — \\ — \vdots — \\ — (x^{(m)})^{\mathrm{T}} — \end{bmatrix} \qquad y = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(m)} \end{bmatrix} \qquad \theta = (X^T X)^{-1} X^T y$$

$$\theta = \left(X^T X \qquad\qquad\qquad \right)^{-1} X^T y$$

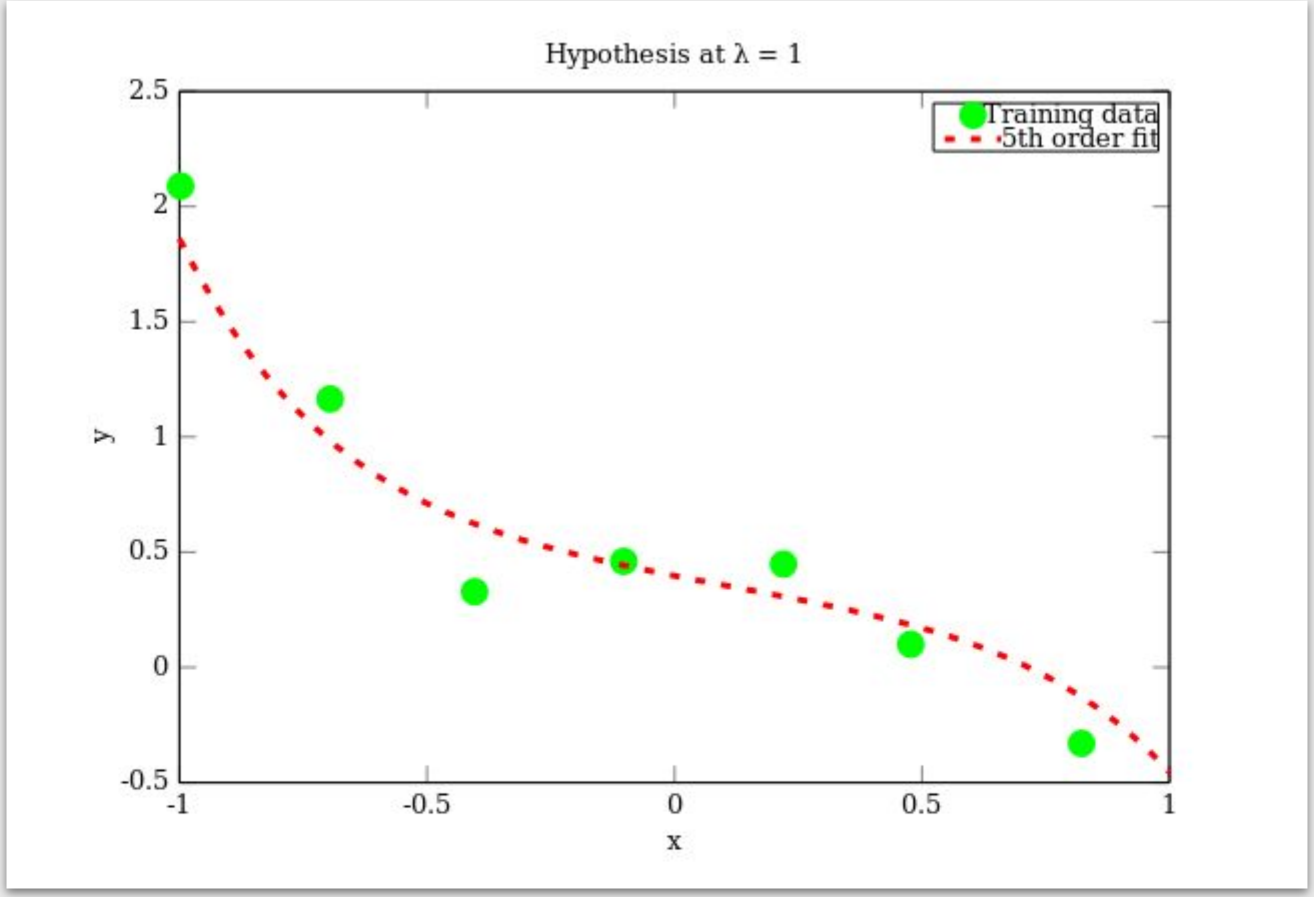
Normal Equation

$$X = \begin{bmatrix} — & (x^{(1)})^{\mathrm{T}} & — \\ — & (x^{(2)})^{\mathrm{T}} & — \\ — & \vdots & — \\ — & (x^{(m)})^{\mathrm{T}} & — \end{bmatrix} \quad y = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(m)} \end{bmatrix} \quad \theta = (X^T X)^{-1} X^T y$$

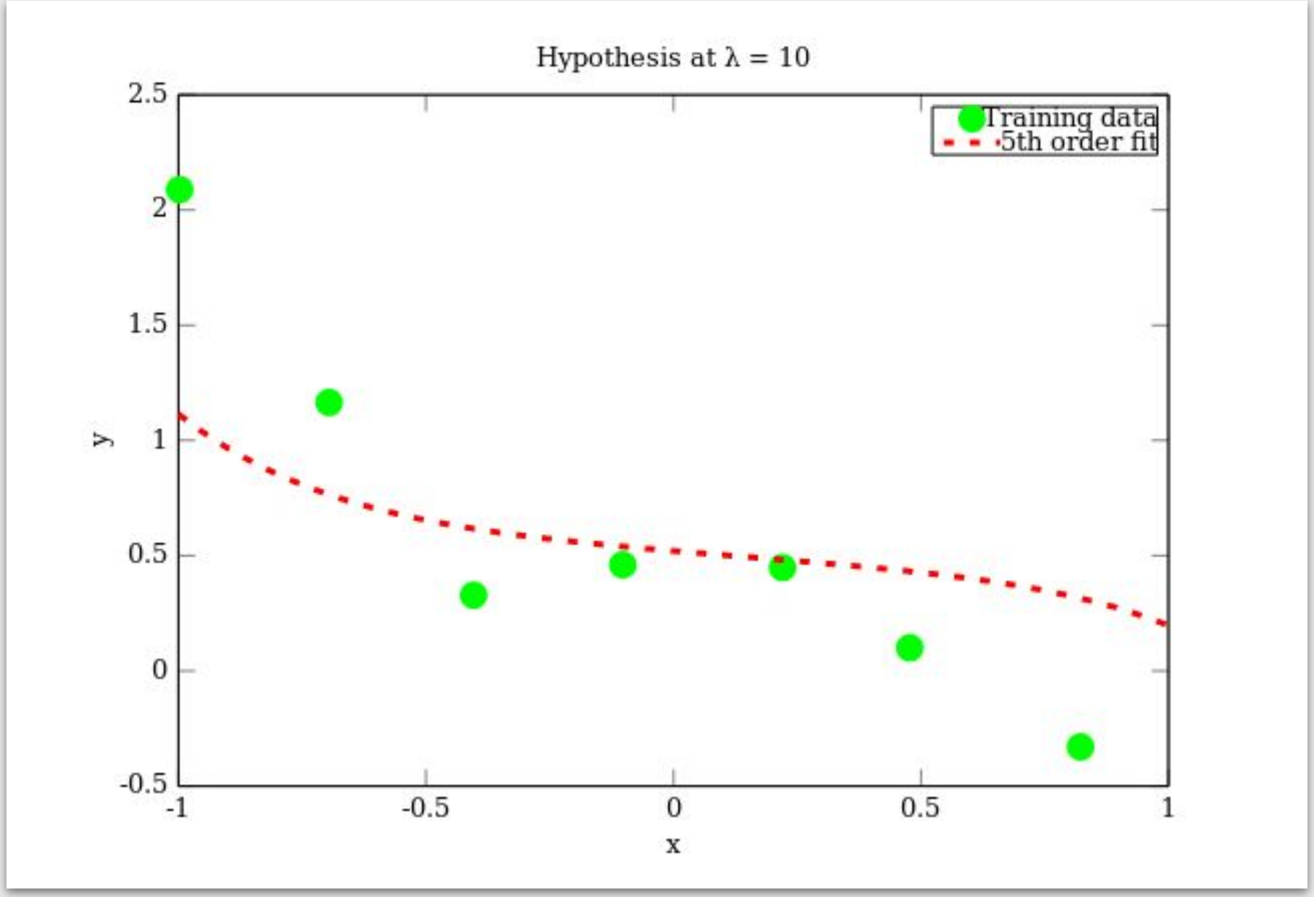
$$\theta = \left(X^T X + \lambda \begin{bmatrix} 0 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & \ddots & \\ & & & & 1 \end{bmatrix} \right)^{-1} X^T y$$



http://melvincabatuan.github.io/Machine-Learning-Activity-4/



http://melvincabatuan.github.io/Machine-Learning-Activity-4/



http://melvincabatuan.github.io/Machine-Learning-Activity-4/

Regularized Logistic Function

Gradient Descent

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

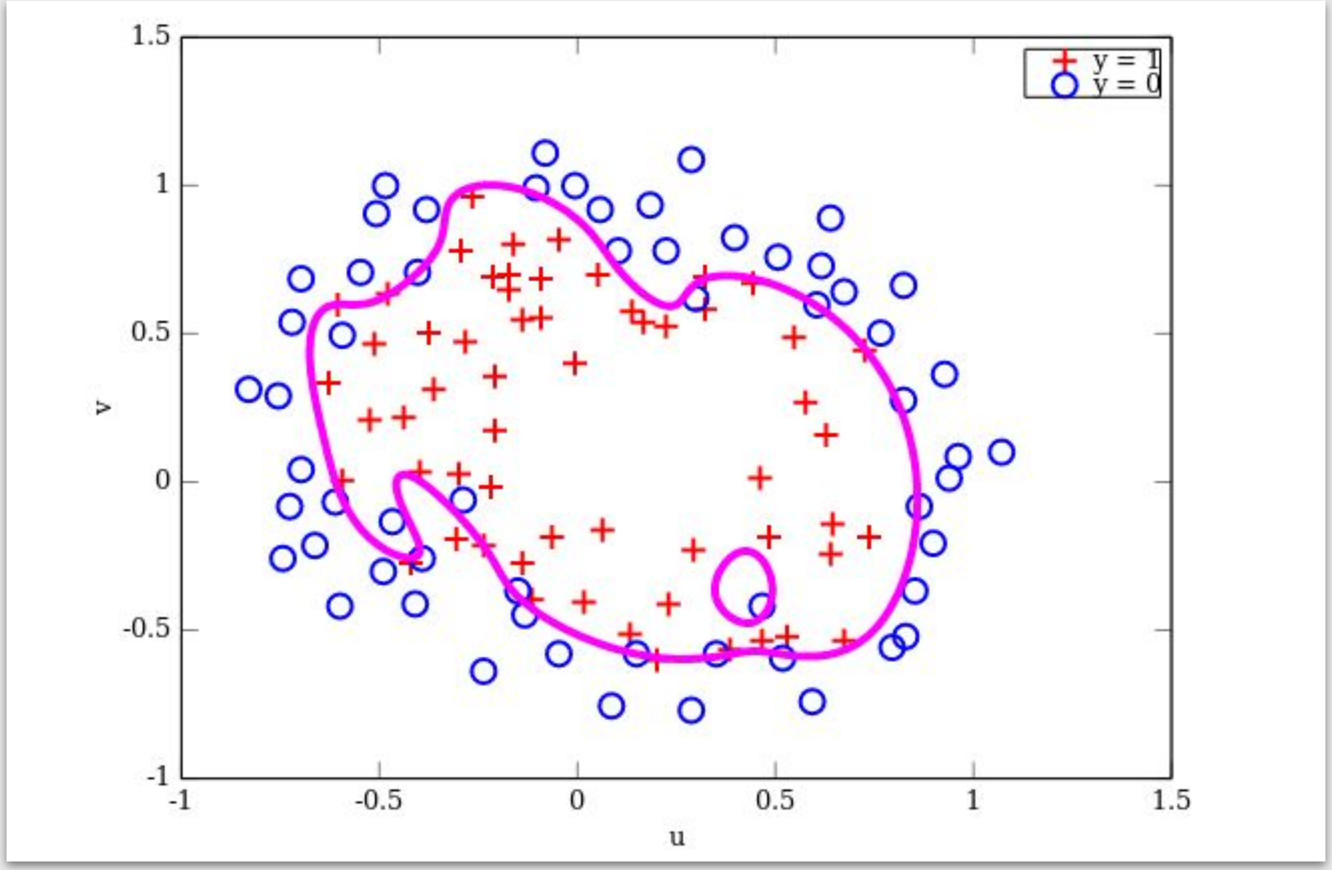
repeat {

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

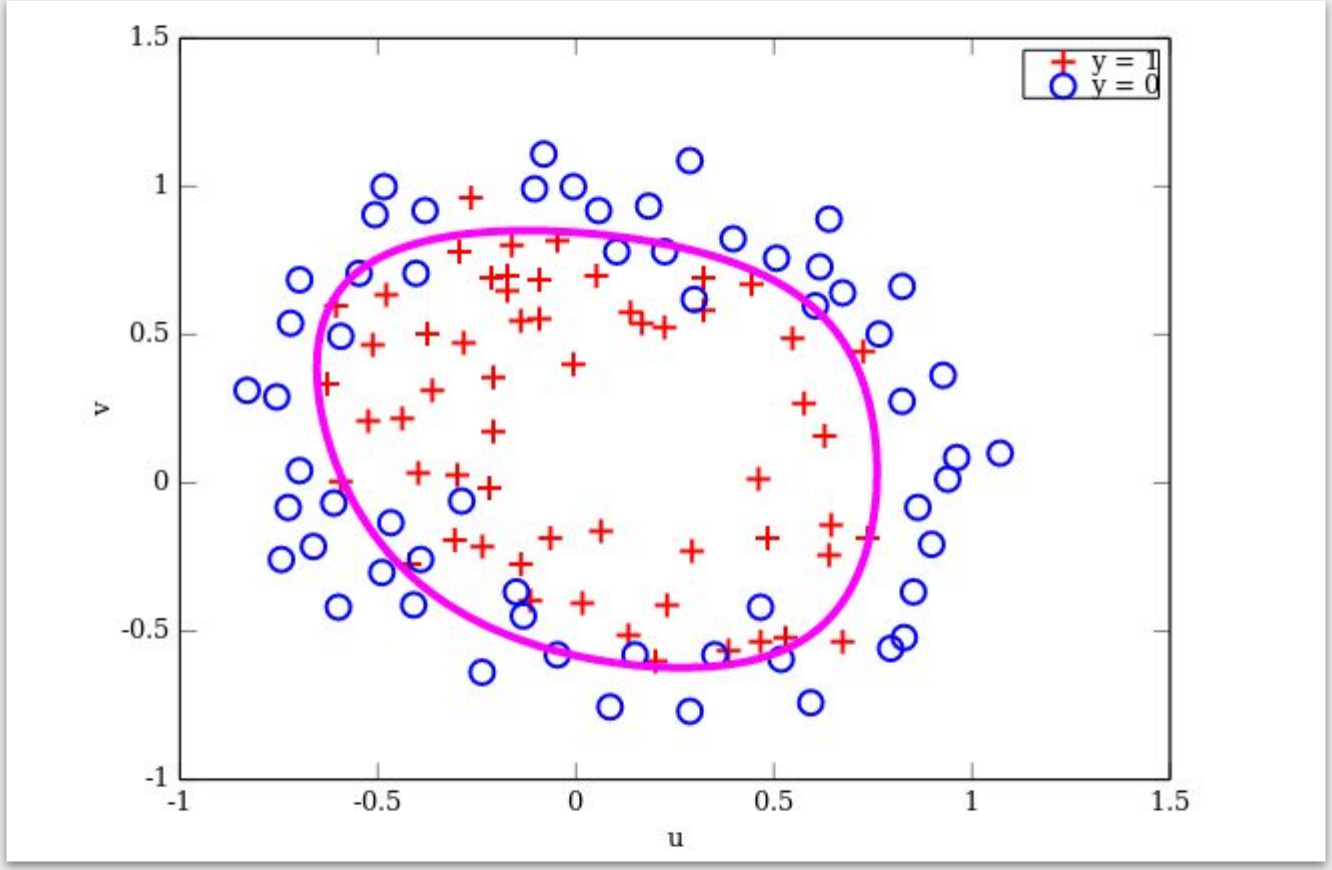
$$\theta_j := \theta_j - \alpha \left[\frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)} + \frac{\lambda}{m} \theta_j \right]$$

} (simultaneously update θ_j for $j =$ $1, \ldots, n$)

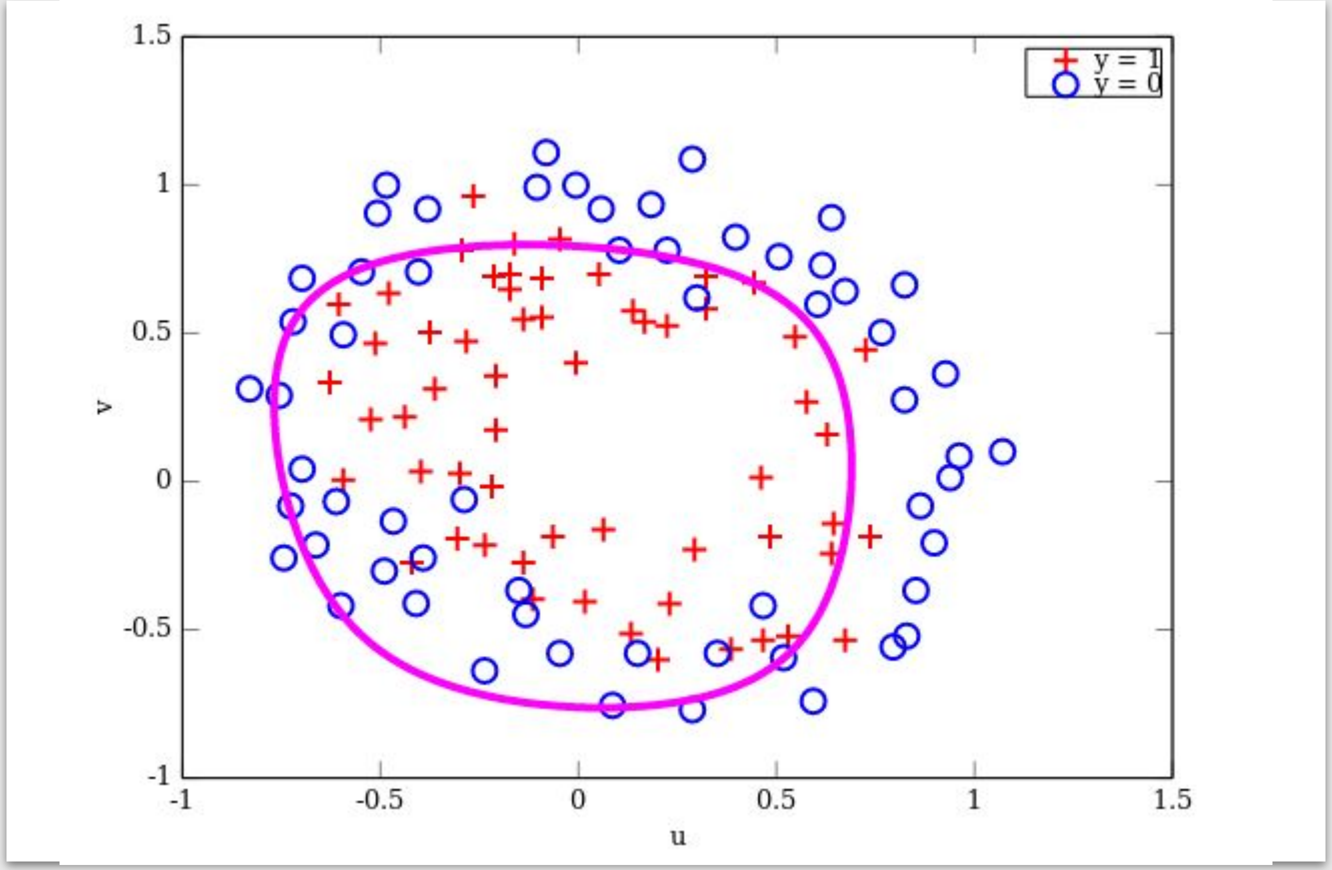
$$\theta_j := \theta_j (1 - \alpha \frac{\lambda}{m}) - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)}$$



http://melvincabatuan.github.io/Machine-Learning-Activity-4/



http://melvincabatuan.github.io/Machine-Learning-Activity-4/



http://melvincabatuan.github.io/Machine-Learning-Activity-4/

References

Machine Learning Books

- Hands-On Machine Learning with Scikit-Learn and TensorFlow, Chap. 4
- Pattern Recognition and Machine Learning, Chap. 3

Machine Learning Courses

- https://www.coursera.org/learn/machine-learning, Week 3 & 6