Introduction to Data Science

(Lecture 14)

Dr. Mohammad Pourhomayoun

Assistant Professor

Computer Science Department

California State University, Los Angeles

Review

- So far we just talked about **Linear** Models (with one or more features)!
- We learned **Linear Regression** and **Logistic Regression**.
- In Linear Regression and Logistic Regression, we only use the **first order of the features**.

Linear Regression Model:

$$h_\theta(\boldsymbol{x}) = \boldsymbol{\theta}^T \boldsymbol{x} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \ldots + \theta_n x_n$$

Logistic Regression Model:

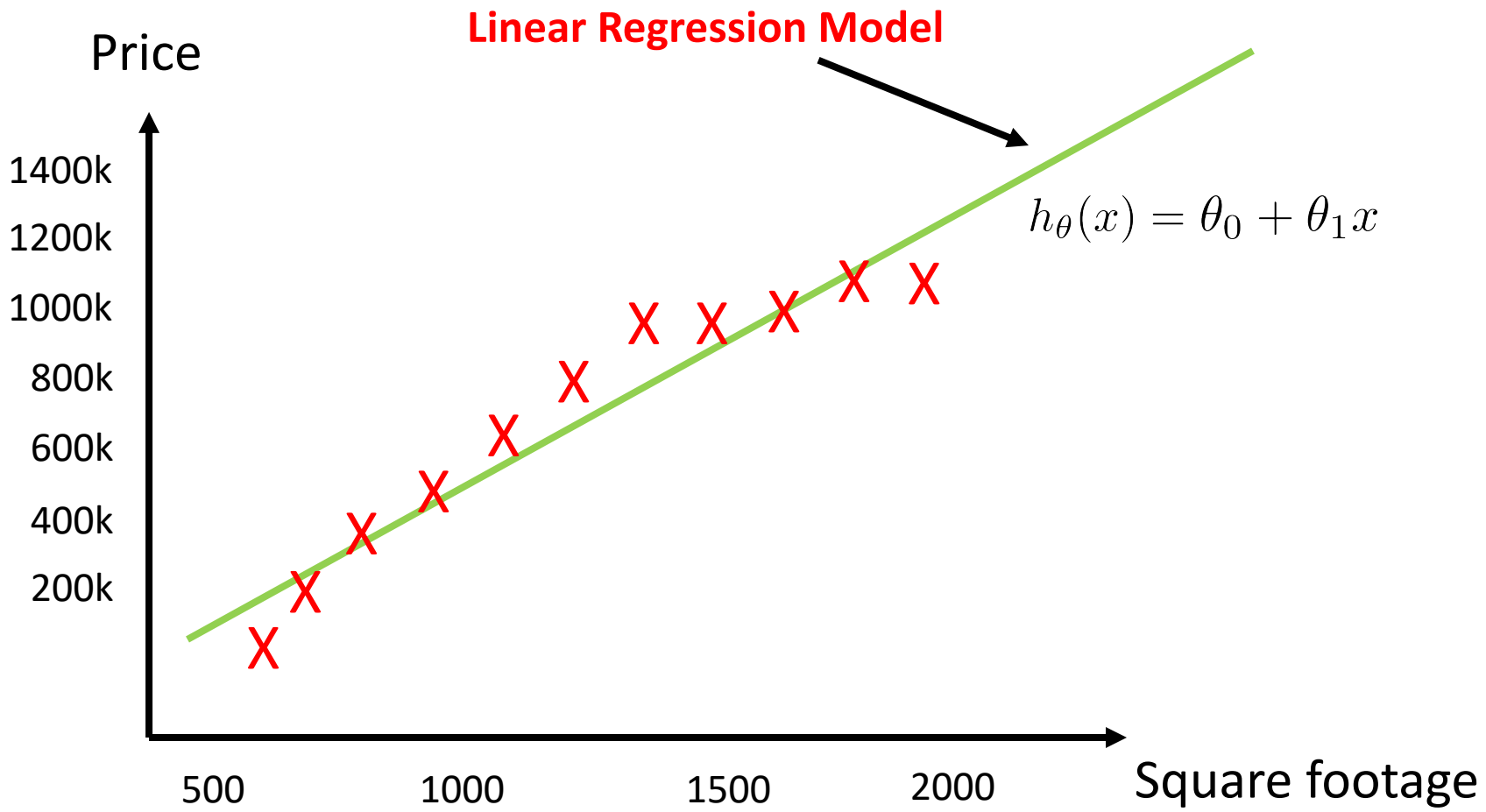
$$h_\theta(\boldsymbol{x}) = g(\boldsymbol{\theta}^T \boldsymbol{x}) = \frac{1}{1 + e^{-\boldsymbol{\theta}^T \boldsymbol{x}}}$$

Polynomial Regression

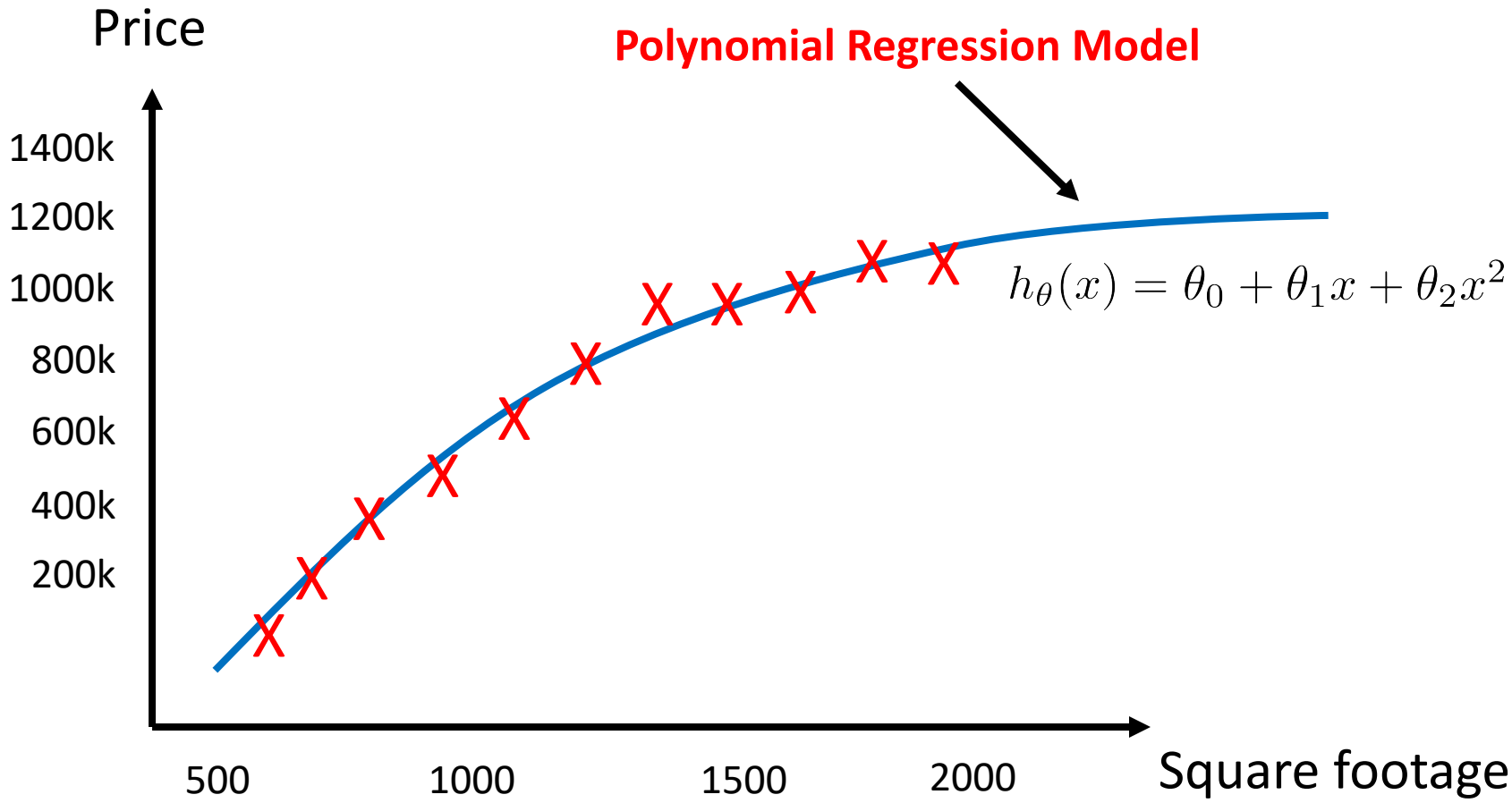
Polynomial Model

- Sometimes, for some specific data, using a model with higher degrees (i.e. polynomial) may achieve a better accuracy than linear regression or logistic regression with order one (first degree).

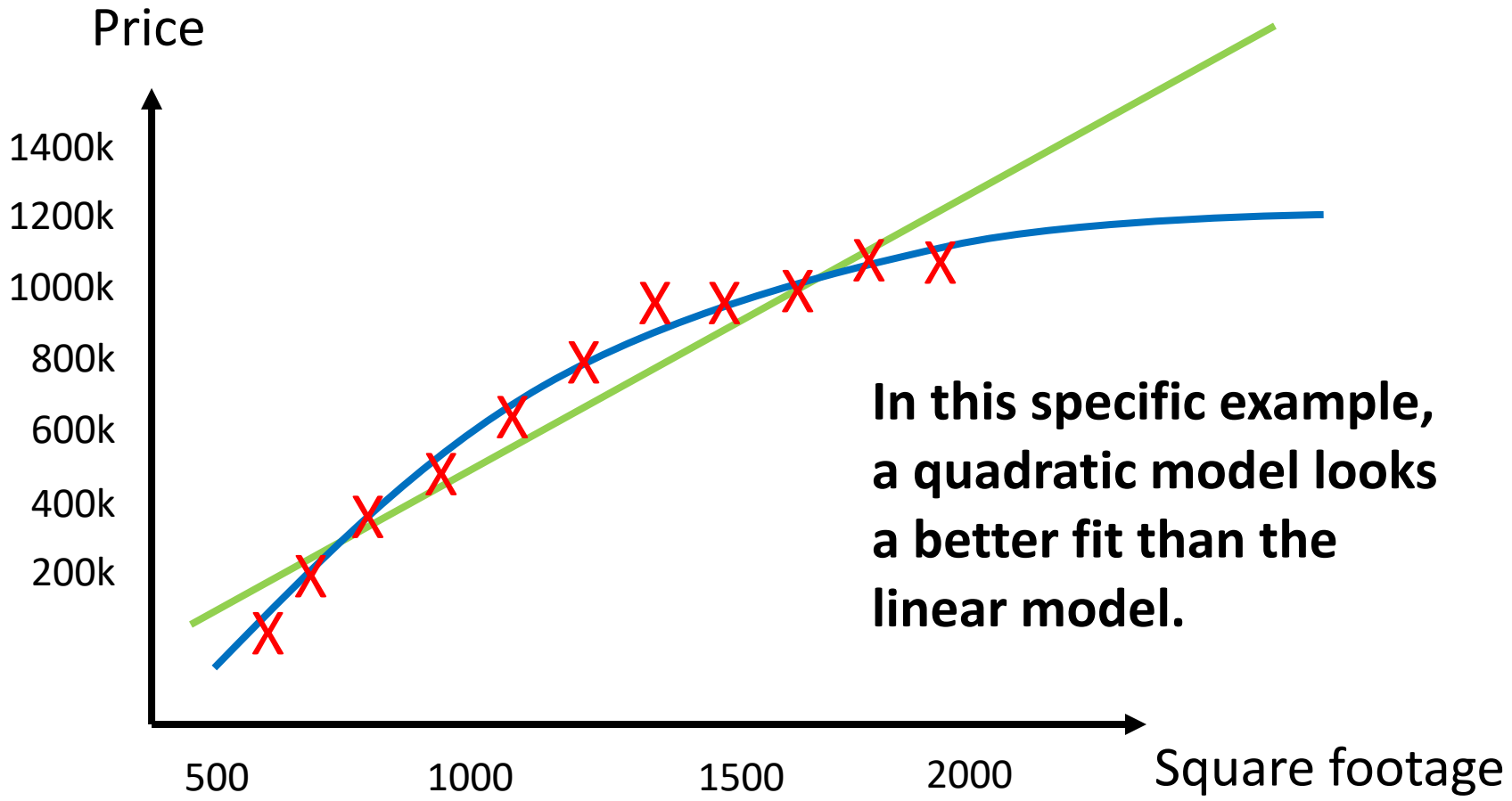
Regression Example



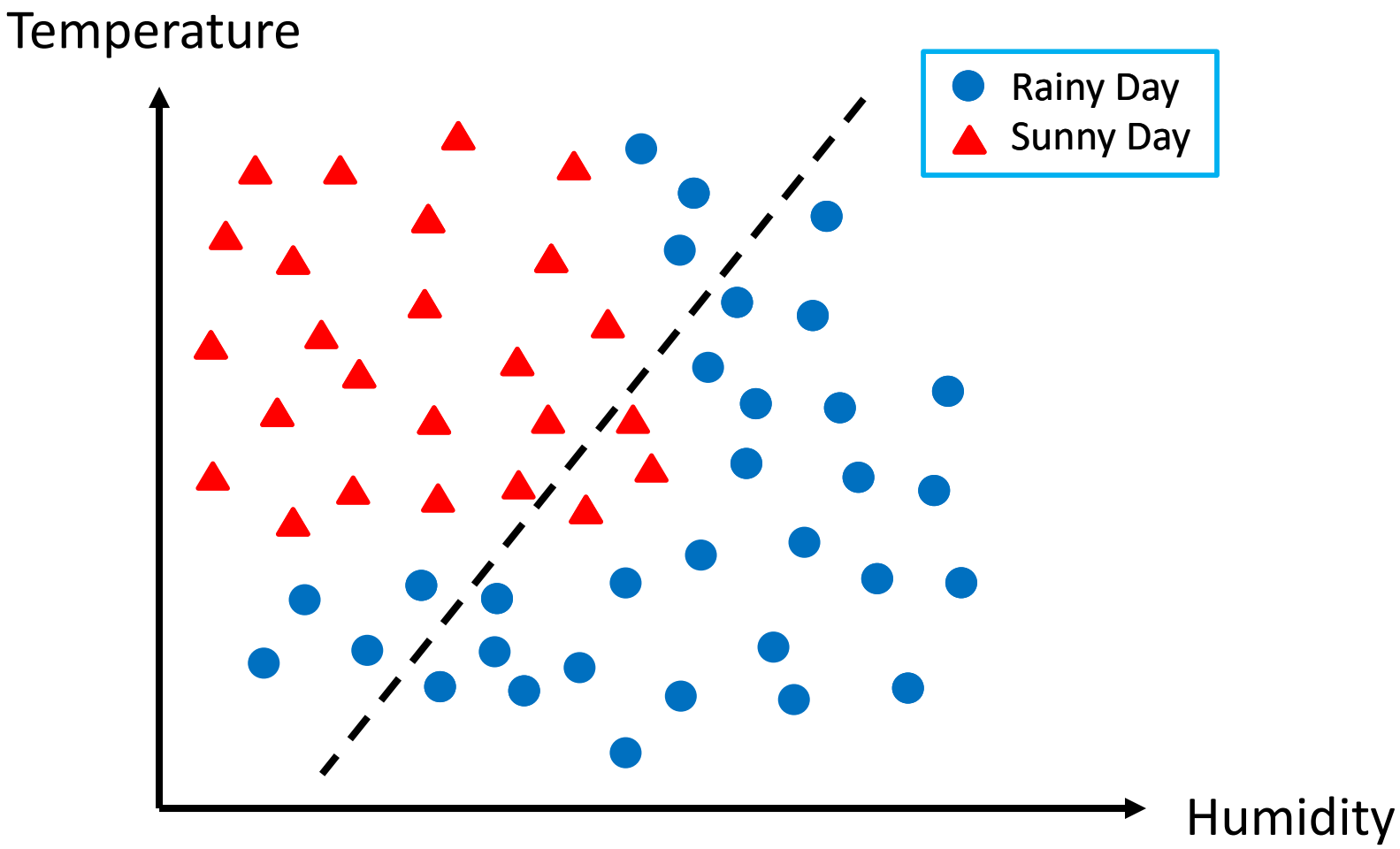
Regression Example



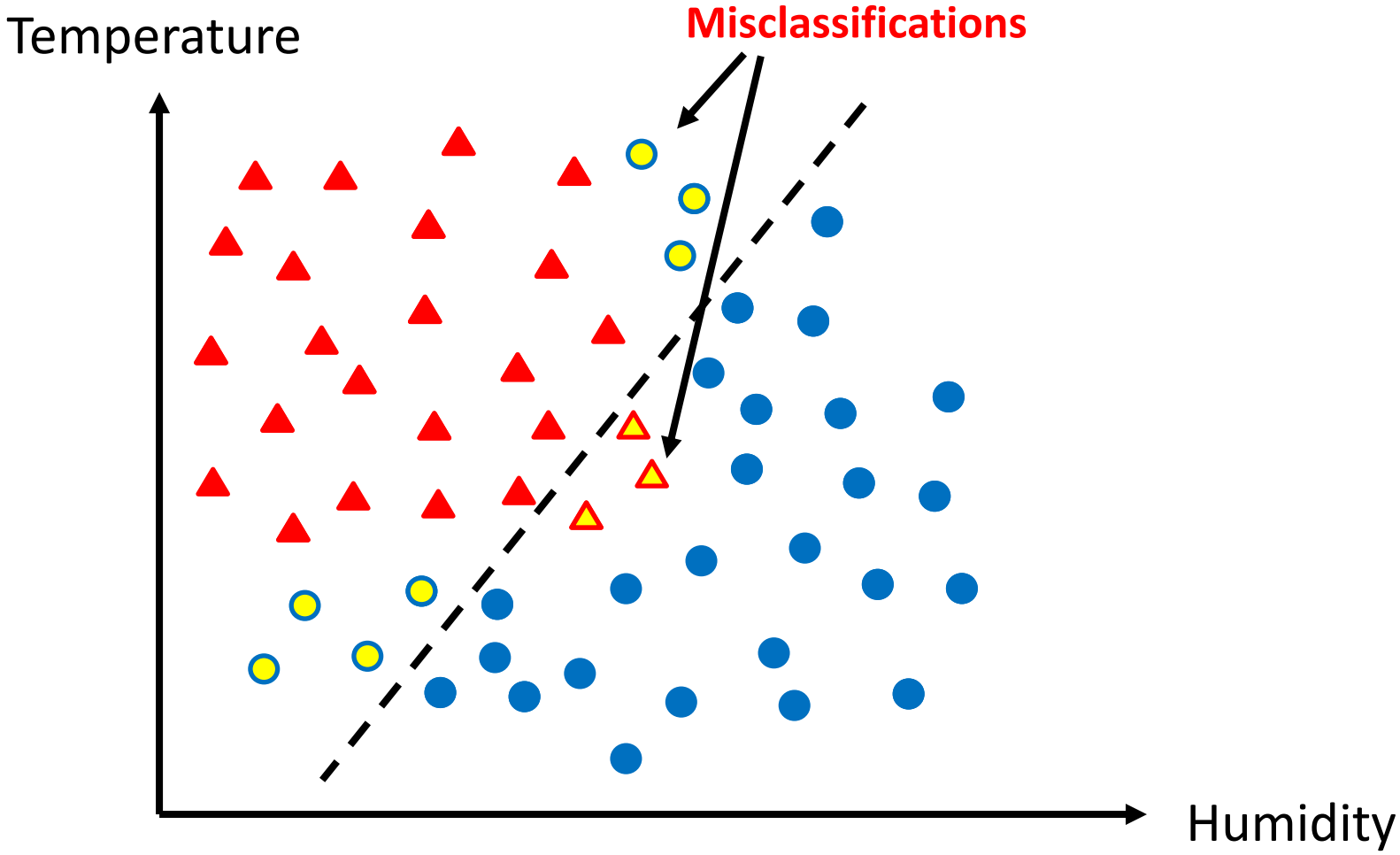
Regression Example



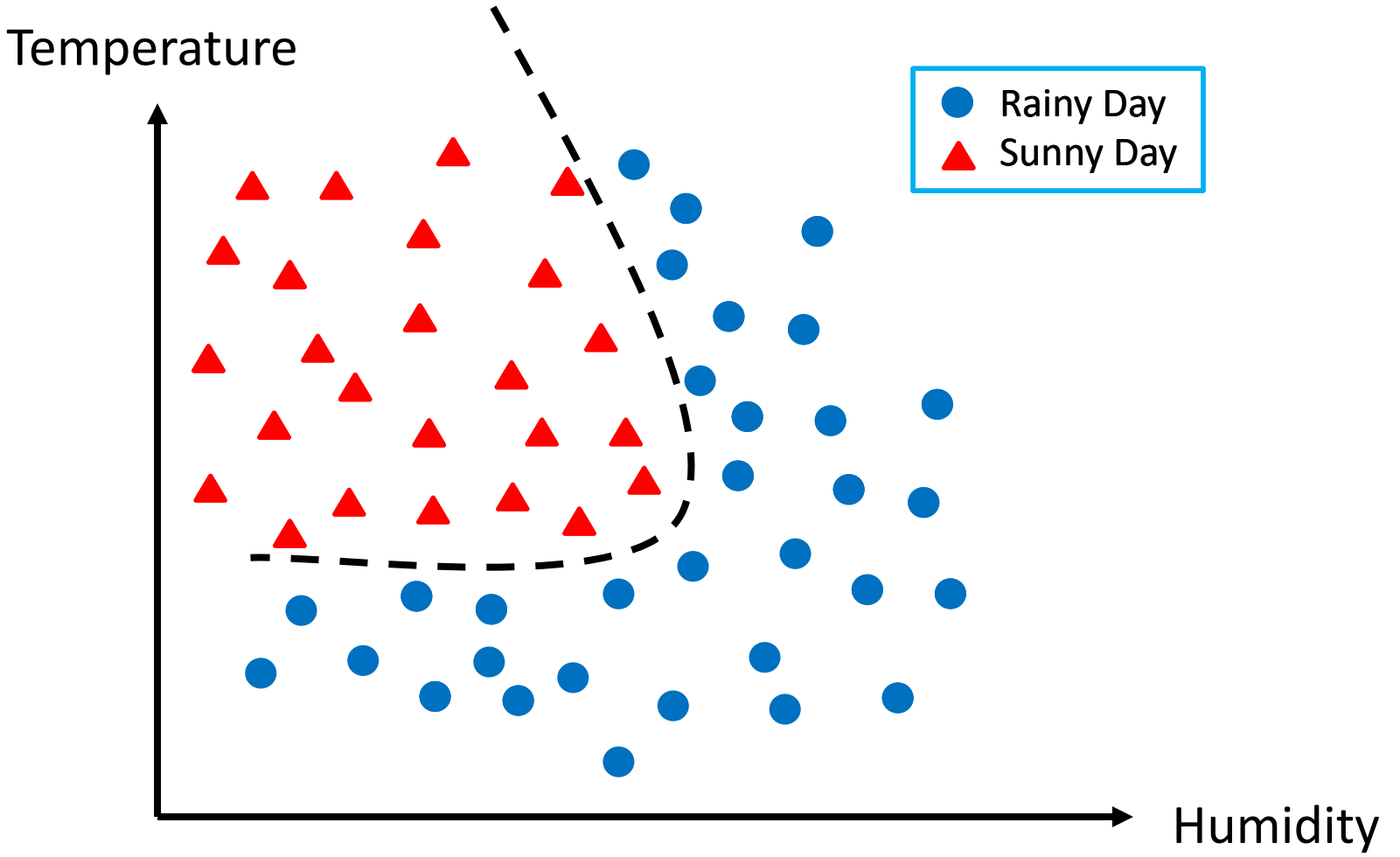
Classification Example



Classification Example



Classification Example



Example

- Linear Regression with one feature x:

$$h_\theta(x) = \theta_0 + \theta_1 x$$

- Polynomial Regression with one feature x and order 2 (quadratic):

$$h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2$$

- Polynomial Regression with one feature x and order 4:

$$h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$$

Example

- Regular Logistic Regression with two features x_1 , x_2 :

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

- Polynomial Classifier with two features x_1 , x_2 and order 2 (quadratic):

$$\begin{aligned} h_\theta(x) = \; & g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 \\ & + \theta_3 x_1^2 + \theta_4 x_2^2 \\ & + \theta_5 x_1 x_2) \end{aligned}$$

- Polynomial Classifier with two features x_1 , x_2 and higher order:

$$\begin{aligned} h_\theta(x) = \; & g(\theta_0 + \theta_1 x_1 + \theta_2 x_1^2 \\ & + \theta_3 x_1^2 x_2 + \theta_4 x_1^2 x_2^2 \\ & + \theta_5 x_1^2 x_2^3 + \theta_6 x_1^3 x_2 + \dots) \end{aligned}$$

The Problem of Overfitting

The Problem of Overfitting

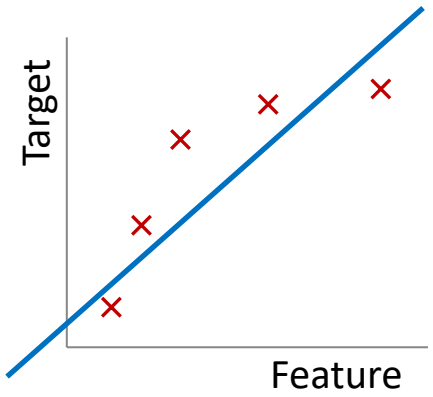
- **Overfitting** happens when the predictive model (classification model or regression model) **fits too much** with the **Training Samples** so that it starts learning, capturing and representing the **noise and randomness or outlier samples** available in the training dataset.

- Overfitting provides *excellent accuracy for training data*, but *poor results* for future data samples (*testing set*)!

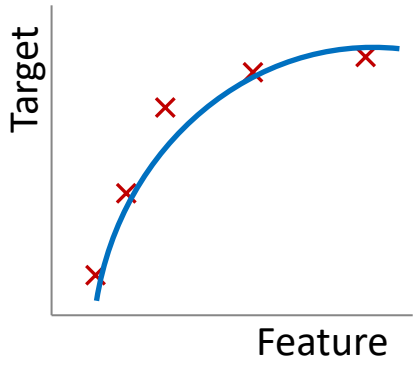
The Problem of Overfitting

- **Overfitting** usually occurs when a model is excessively complex. The two main reasons that makes a model too complex are:
  1. having **too many input features**.
  2. having a **complex model with very high order**.

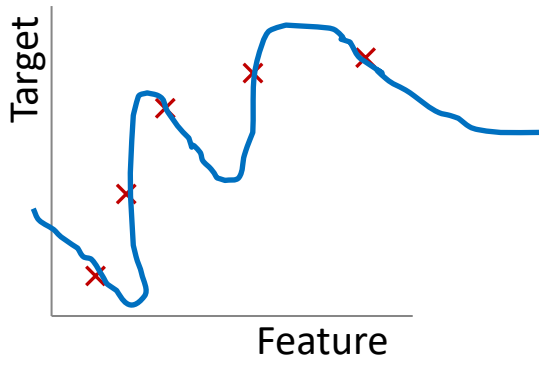
Example of Overfitting for Regression



Under-fit



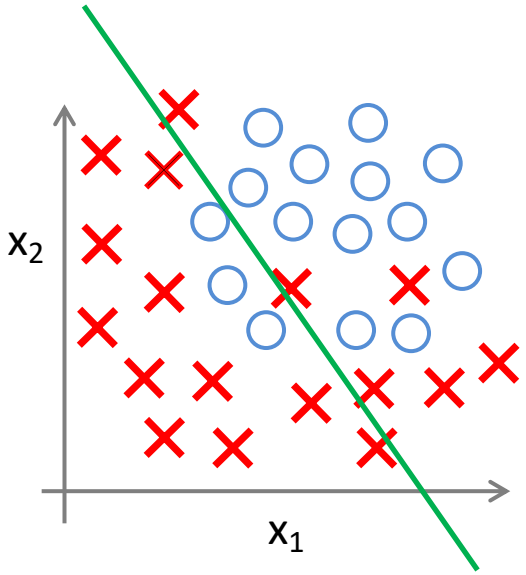
Ideal fit



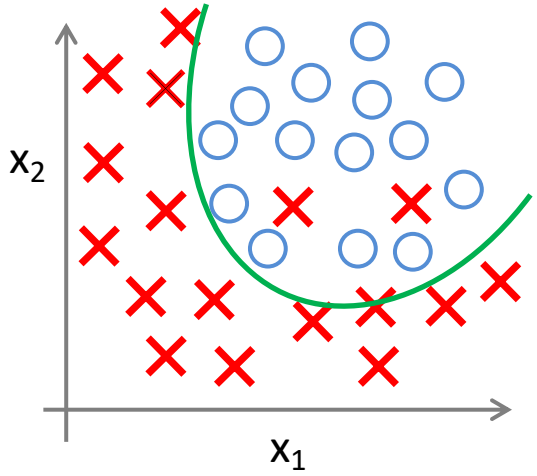
Over-fit

*Reference: Andrew Ng, Stanford University.

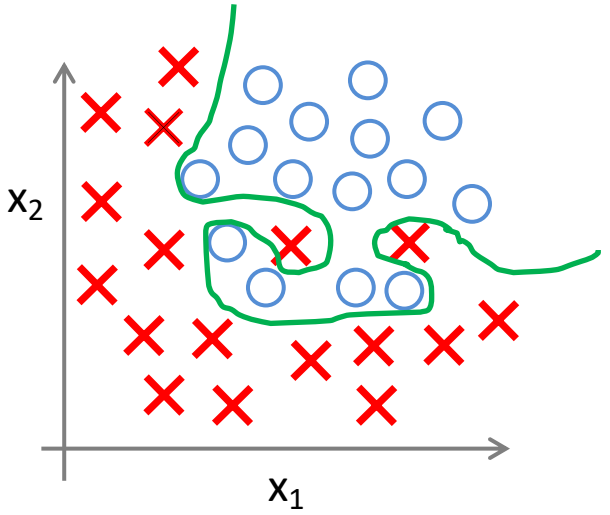
Example of Overfitting for Classification



Under-fit



Ideal fit



Over-fit

*Reference: Andrew Ng, Stanford University.

Addressing the Overfitting: Approach 1: Dimensionality Reduction

- **Approach 1**: **Dimensionality Reduction** (Reduce the number of features):

$$\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \ldots + \theta_{20} x_{20} \quad \rightarrow \quad \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3$$

a) Manually select which features to keep.

b) Detecting the **best features** using automated **Feature Selection** and/or **Dimensionality Reduction** algorithms (will be covered in CS4662/CS5661).

Example: Best Features for Iris Flowers Dataset

- According to your homework results, what was the best single feature?
- What was the second best feature?
- What was the best pair of features?
- Is the best pair equal to the combination of the best first and best second features?
- Why?
- What was the accuracy of classification based on the best pair?
- Do we really need to use all 4 features?
- According to your results, is always a larger K better for KNN?
- What is the best K?

Feature Selection

- **Feature selection** is an important field of research in data science. The conventional feature selection algorithms usually focus on specific metrics to quantify the relevance and/or redundancy of each feature with the goal of finding **the smallest subset of features that provides the maximum amount of useful information** for prediction.
- Thus, the **main goal of feature selection algorithms is to eliminate redundant or irrelevant features** in a given feature set.
- Applying an effective feature selection algorithm not only decreases the complexity of the system by reducing the dimensionality, but also increases the performance of the classifier by avoiding overfitting and also removing irrelevant and confusing features.

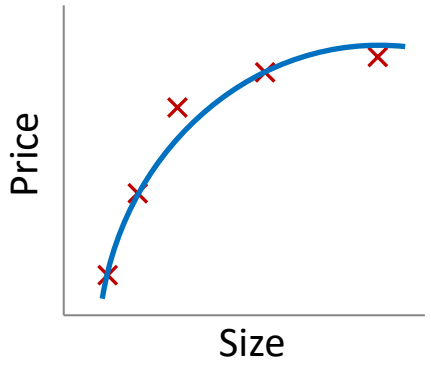
Addressing the Overfitting: Approach 2: Regularization

- **Approach 2**: **Regularization**:
  - Keep all features, but reduce the magnitude/values of parameters θ_j to simplify the model.

$$\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \theta_4 x_1^2 + \theta_5 x_2^2 + \theta_6 x_2 x_3 + \theta_6 x_2 x_3^2 + \ldots$$

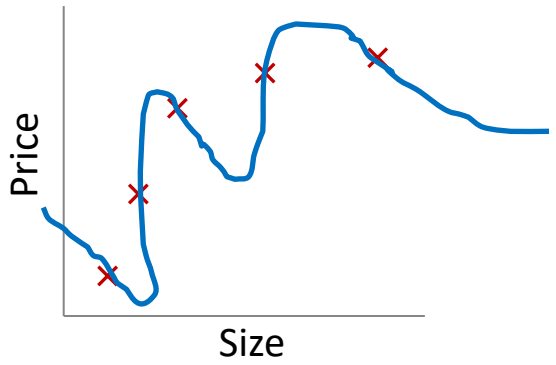
$$\rightarrow \quad \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \theta_4 x_1^2 + \theta_6 x_2 x_3$$

Regularization



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

Ideal fit



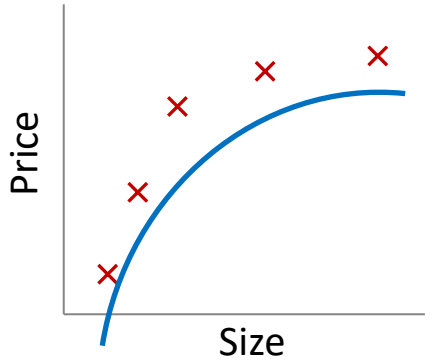
$$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$$

Over-fit

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

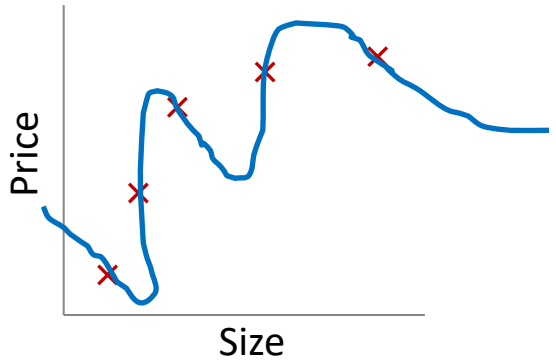
Goal: $\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) \;=\; \min[\frac{1}{2m}\sum_{i=1}^{m}(h_\theta(\boldsymbol{x}^{(i)}) - y^{(i)})^2]$

Regularization



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

Ideal fit



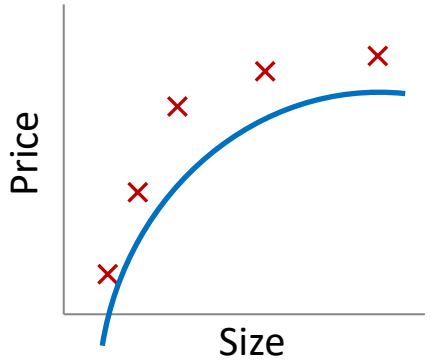
$$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$$

Over-fit

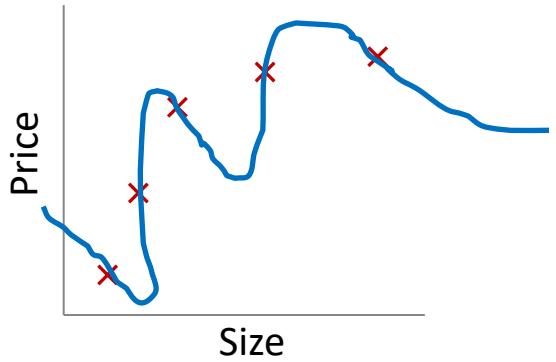
Alternative Cost Function: $J(\boldsymbol{\theta}) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(\boldsymbol{x}^{(i)}) - y^{(i)})^2 + \underbrace{\theta_3^{\,2} + \theta_4^{\,2}}_{\text{Regularization}}$

Goal: $\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = \min[\frac{1}{2m}\sum_{i=1}^{m}(h_\theta(\boldsymbol{x}^{(i)}) - y^{(i)})^2 + \underbrace{\theta_3^{\,2} + \theta_4^{\,2}}_{\text{Regularization}}]$

Regularization



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



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

New Cost Function: $$J(\boldsymbol{\theta}) = \frac{1}{2m}\left[\sum_{i=1}^{m}(h_\theta(\boldsymbol{x}^{(i)}) - y^{(i)})^2 + \lambda\underbrace{\sum_{j=1}^{n}\theta_j^{\,2}}_{\text{Regularization}}\right]$$

Goal: $$\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = \min \frac{1}{2m}\left[\sum_{i=1}^{m}(h_\theta(\boldsymbol{x}^{(i)}) - y^{(i)})^2 + \lambda\underbrace{\sum_{j=1}^{n}\theta_j^{\,2}}_{\text{Regularization}}\right]$$

Regularization

New Cost Function:

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

Regularization parameter

λ (Regularization parameter) controls the trade-off between regularization term (to keep the parameters small to keep the model simple), and fitting term (to achieve acceptable fitting with the training data).

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

Regularization

Regularization for Linear Regression (Optional)

Regularization for Linear Regression:

- **Gradient descent:**

Repeat $\{$

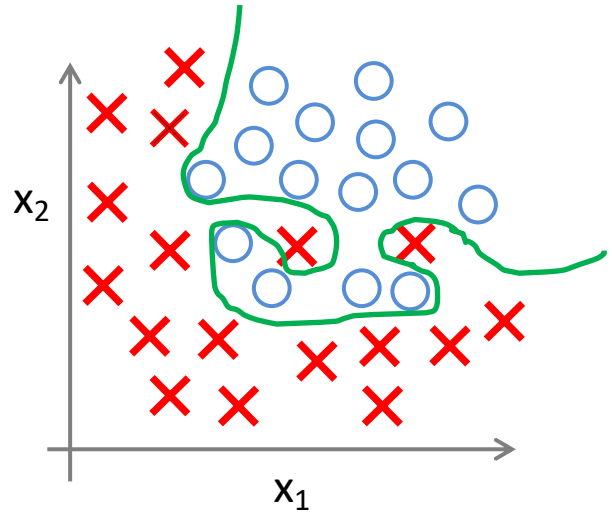
$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_0^{(i)}$$

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

$$(j = 1,2,3,\ldots,n)$$

$\}$

Regularization for Logistic Regression



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

New Cost function:

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

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

Regularization for Logistic Regression (Optional)

Regularization for Logistic Regression:

- **Gradient descent:**

Repeat $\{$

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

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

$$(j = 1,2,3,...,n)$$

$\}$

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

Summary

- In the past four sessions of class, we covered:
  - Linear Regression with one feature
  - Linear Regression with multiple features
  - Logistic Regression Classifier with multiple features
  - Polynomial model for Regression with multiple features
  - Polynomial model for Classification with multiple features
- The problem of OverFitting!
- Solutions to deal with OverFitting
  - Dimensionality Reduction (or Feature Selection) to reduce the number of features
  - Regularization to simplify the model

Thank You!

Questions?