ieeta instituto de engenharia electrónica e telemática de aveiro

universidade de aveiro

Foundations of Machine Learning

(winter semester 2022/2023)

LECTURE 1: INTRODUCTION & REGRESSION

Petia Georgieva
(petia@ua.pt)

universidade de aveiro

LECTURE OUTLINE

1. ML Overview

2. Linear Regression
   - Cost (loss) function
   - Gradient descent optimization

3. Overfitting problem. Regularization

Machine Learning Approaches

Supervised Learning

Unsupervised Learning

Deep Learning

Reinforcement Learning

Machine Learning Approaches

Supervised Learning

Given examples with "correct answer" (labeled examples)

(e.g. given dataset with spam/not-spam labeled emails)

Unsupervised Learning

Given examples without answers (no labels).

Deep Learning

Automatically extract hidden features (in contrast to hand-crafted features). Need a lot of data (Big data) . Need for very high computational resources (GPUs).

Reinforcement Learning

On-line (on the fly) learning, by trial and error.

Supervised Learning

Requires labeled data (examples with “correct answer”).

Regression: The Labels are real numbers.

Ex. Predict the house price (output) based on data for the house area and number of bedrooms (features).

Classification: The Labels are integer numbers (class 1, class 2, etc.)

Ex. Predict normal (0) or abnormal (1) state of data center computers:

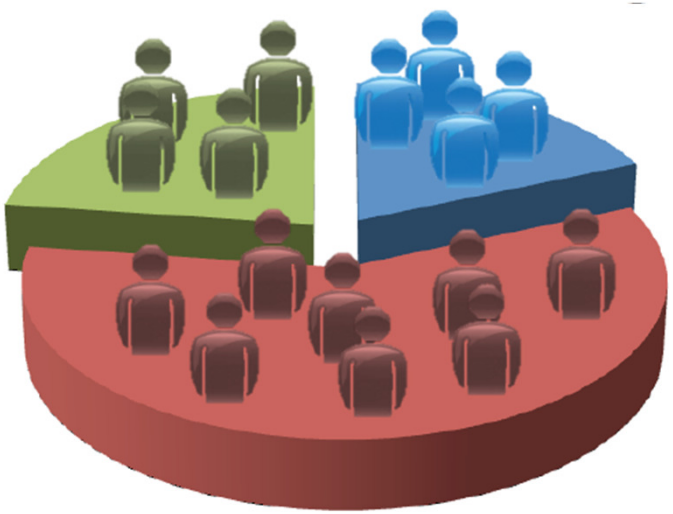
Features: memory use of computer ; number of disc accesses /sec; CPU load ; network traffic; silence

Unsupervised Learning

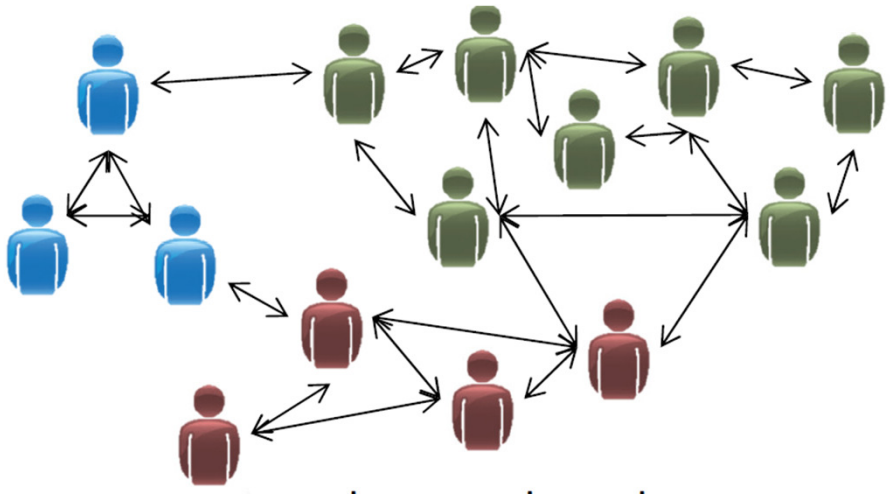
Given unlabeled data (NO answers)

Features: education, job, age, marital status, etc.

Market segmentation



Social network analysis



Clustering: Given a collection of examples (e.g. user profiles with a number of features). Each example is a point in the multidimensional space of features. Find a similarity measure that separates the points into clusters.

-K-means clustering

Deep Learning

Hardware get smaller.
Sensors get cheaper, widely available IoT devices with high sample-rate.
Data sources: sound, vibration, image, electrical signals, accelerometer, temperature, pressure, LIDAR, etc.

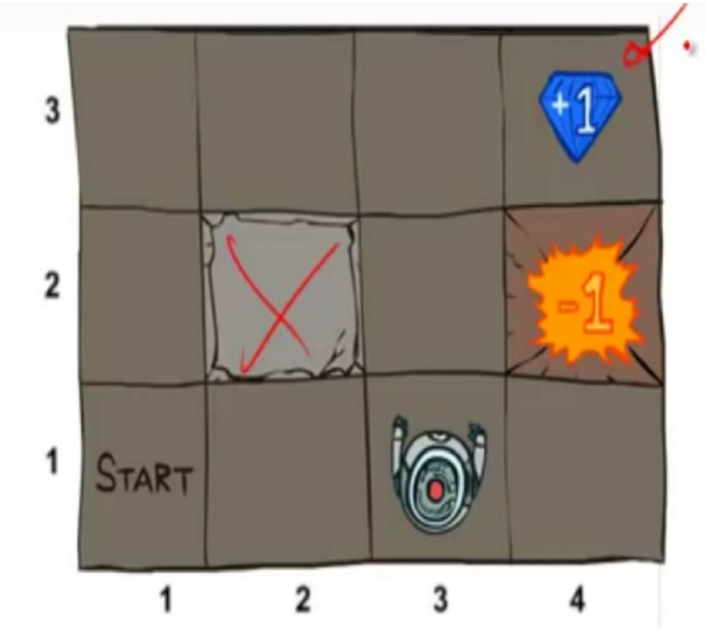
Big Data: Exponential growth of data, (IoT, medical records, biology, engineering, etc.)

How to deals with **unstructured data** (image, voice, text, EEG, ECG, etc.) => What are the best feature ?

Deep Neural Networks: first extract (automatically) the hidden features, then solve ML tasks (classification, regression)

Reinforcement Learning

On-line learning by taking actions and getting rewards/penalties.
intelligent robotics =>



Evaluation

Lectures & labs: 3 hours per week.

Practical component - 50% of the final grade

Practical component consists of 2 projects, developed in a group of two students.

Project 1 is evaluated based on a submitted report (IEEE format) and a short (10-15 min.) oral presentation.

Project 2 is evaluated based on a submitted report (IEEE format).

The students are encouraged to use Latex text editor.
Overleaf is a convenient platform for collaborative writing and publishing using Latex (https://www.overleaf.com/) .

Theoretical Component – 50% of the final grade

Final individual exam.

All starts with data collection

- **Plan data recording** to cover all sources of variation related to

✓ target (different events, anomalies)
✓ background (noise, different environments and conditions)
✓ equipment (different sensors, placement).

- **Collect iteratively**

ML works best as an iterative process.

✓ Collect data to build a basic model that proves the effectiveness of the technique, even if not yet for the full range of expected variations.
✓ Take more data and build a model to get acceptable accuracy in wide range of variations.
✓ Take more data and focus on model optimization to get the best possible performance.

All starts with data collection

Data collection and labelling is the most expensive, most time-consuming aspect of any ML project.

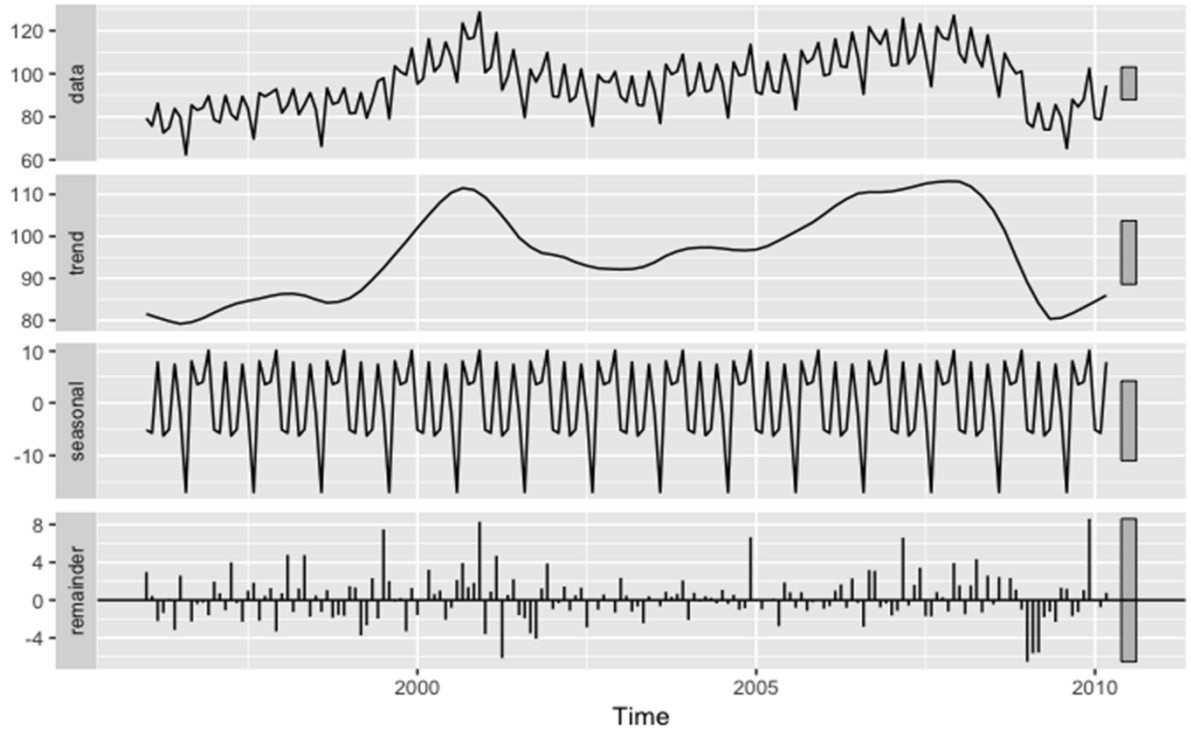
- ✓ **Use rich data.** Collect the least pre-processed signal data from sensors. Compression algorithms often discard important information.

- ✓ **Use too high sample rates.** Collect at the highest sample rate possible in the early stages. Down-sampling in software is easy and cheap. Going back and recollecting data is difficult and expensive.

Data Preprocessing & Feature Extraction

Data preprocessing
Detect outliers, impute missing data, normalize data

Feature extraction (hand-crafted features by domain experts)
Traditional statistics: mean, median, standard deviation, auto-correlation
More parameters : Skewness, Kurtosis
Frequency domain: power spectrum, dominant frequency
Time series: time related features trend, seasonality, residuals per month, day of week, hour.



universidade de aveiro

Feature Engineering

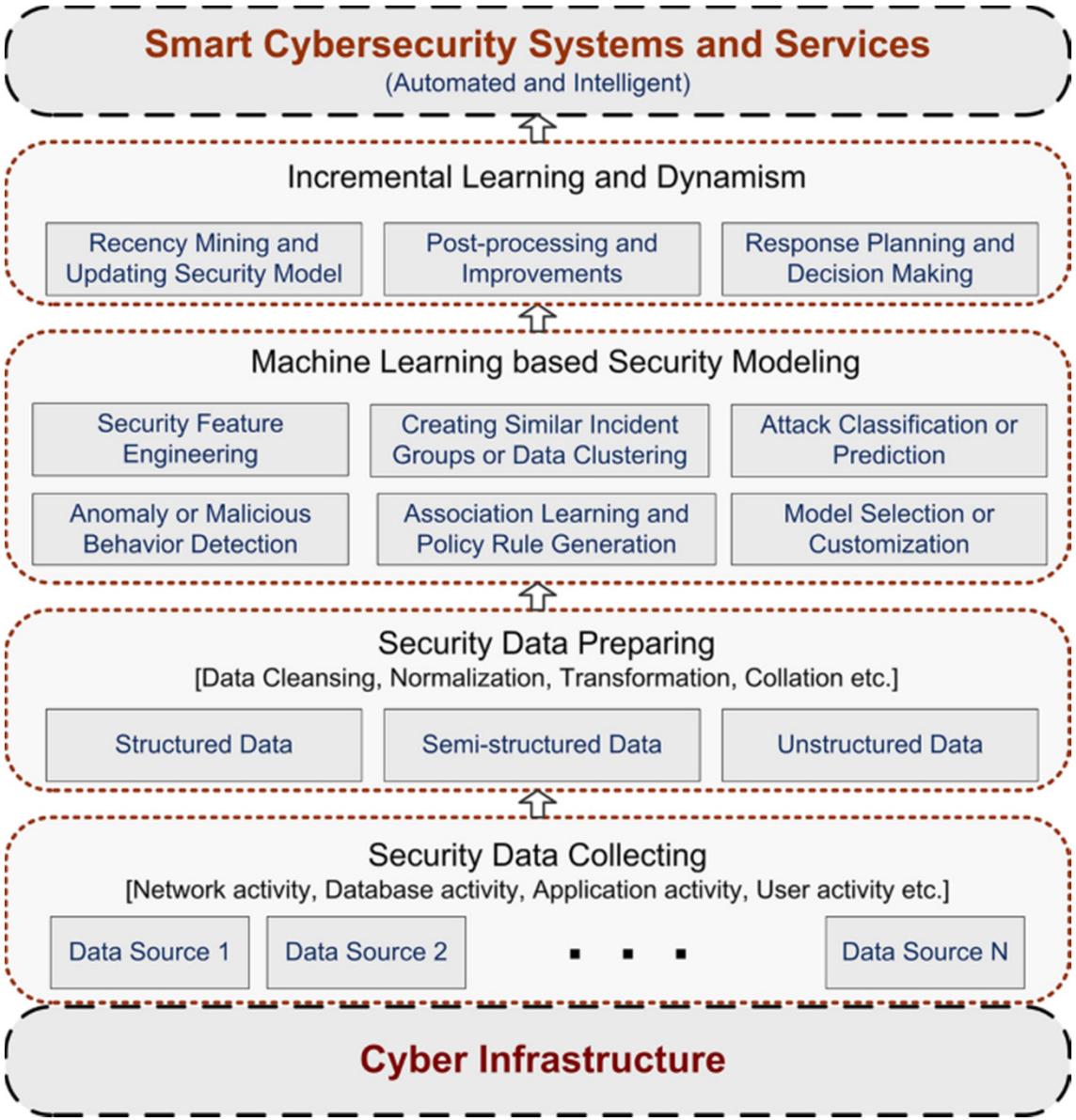
Ex. Monitoring computers in a data center

Basic features:

- memory use of computer
- number of disc accesses /sec
- CPU load
- network traffic

New features:

- CPU load /network traffic
- (CPU load)^2 /network traffic



Sarker, I.H., Kayes, A.S.M., Badsha, S. *et al.* Cybersecurity data science: an overview from machine learning perspective. *Journal of Big Data* 7, 41 (2020). https://doi.org/10.1186/s40537-020-00318-5

Supervised Learning - LINEAR REGRESSION

Univariate Regression

Problem: Learning to predict the housing prices (output) as a function of the living area (input/feature)

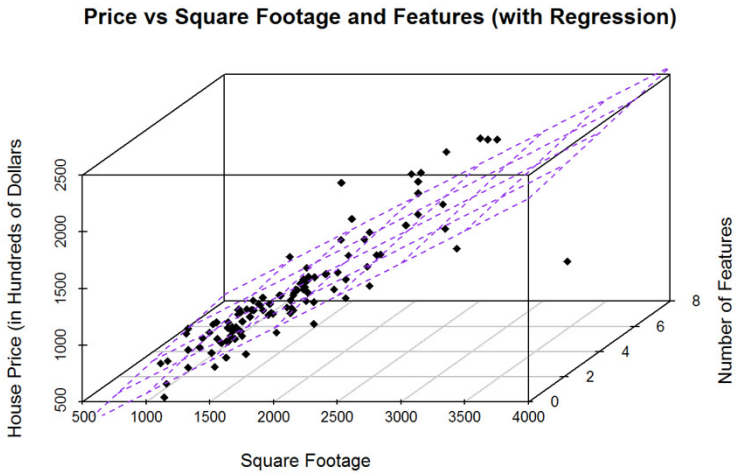
Living area (feet^2)	Price (1000$s)
2104	400
1600	330
2400	369
1416	232
3000	540
⋮	⋮

$$h_\theta(x) = \theta_0 + \theta_1 x_1 = [\theta_0 \quad \theta_1] \begin{bmatrix} x_0 = 1 \\ x_1 \end{bmatrix} = \vec{\theta}^T \vec{x}$$

Multivariate Regression

Problem: Learning to predict the housing price as a function of living area & number of bedrooms.

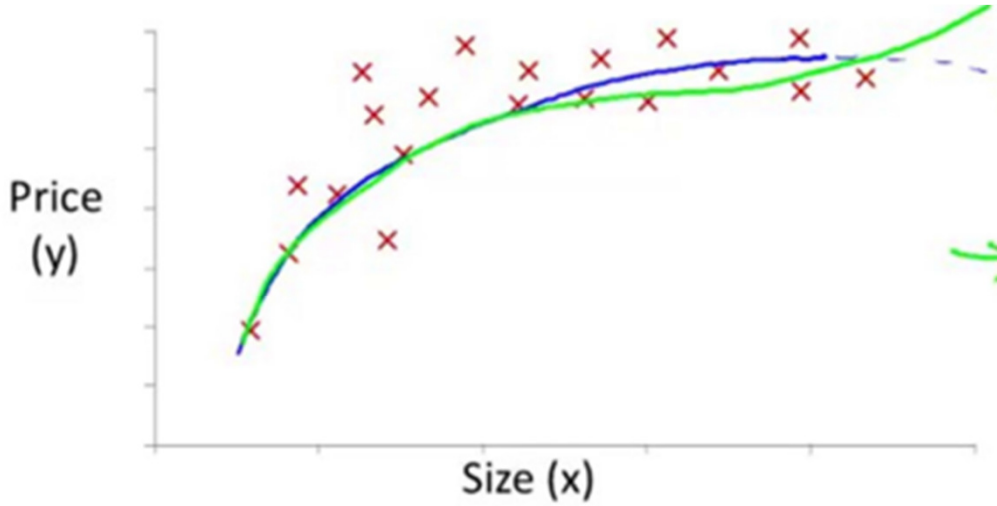
Living area (feet^2)	#bedrooms	Price (1000$s)
2104	3	400
1600	3	330
2400	3	369
1416	2	232
3000	4	540
⋮	⋮	⋮



$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 = [\theta_0 \quad \theta_1 \quad \theta_2] \begin{bmatrix} x_0 = 1 \\ x_1 \\ x_2 \end{bmatrix} = \vec{\theta}^T \vec{x}$$

Polynomial Regression

Univariate (x1=size) housing price problem transformed into multivariate (still linear !!!) regression model x=[x1=size, x2=size^2, x3=size^3]



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

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

$$\begin{aligned} h_\theta(x) &= \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 \\ &= \theta_0 + \theta_1 (size) + \theta_2 (size)^2 + \theta_3 (size)^3 \end{aligned}$$

$$\begin{aligned} x_1 &= (size) \\ x_2 &= (size)^2 \\ x_3 &= (size)^3 \end{aligned}$$

Mean Square Error (MSE)

Linear Model (hypothesis) => $h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \ldots . \theta_n x_n = \theta^T x$

Cost (loss) function => $J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right)^2$
(Mean Square Error)

m – number of training examples

Goal **=>** $\min_\theta J(\theta)$

Gradient descent algorithm =>

Linear Regression – iterative gradient descent algorithm

Inicialize model parameters *(e.g. θ =0)*
Repeat until J converge {

Compute Linear Regression Model =>

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

Compute cost function =>

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

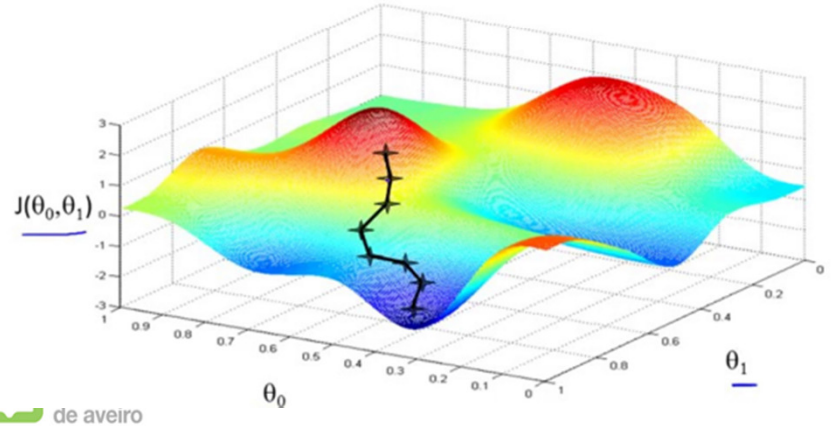
Compute cost function gradients =>

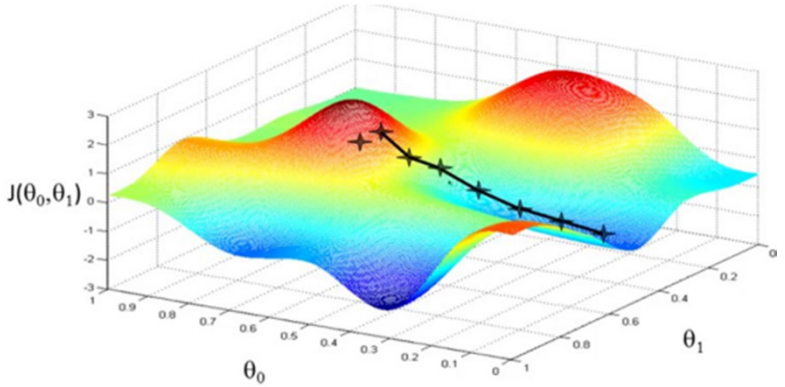
$$\frac{\partial J(\theta)}{\partial \theta_j} = \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)}$$

Update parameters =>

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

}

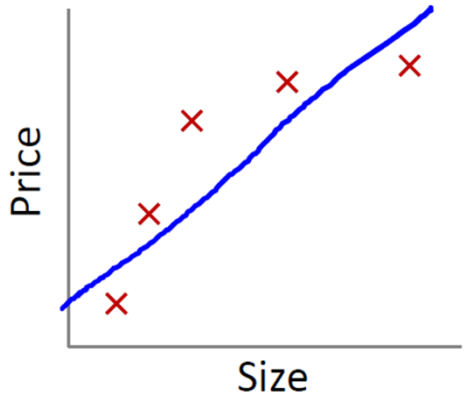




de aveiro

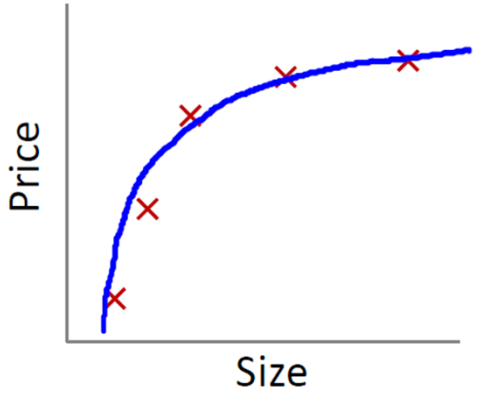
Overfitting problem

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



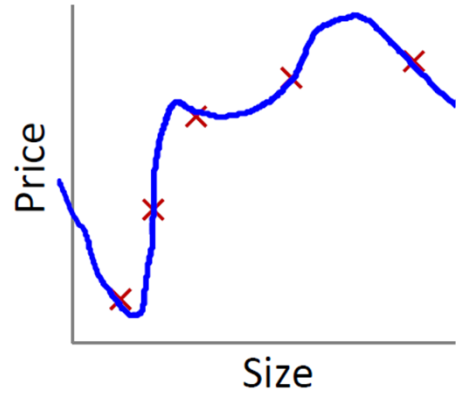
underfit
(1st order polin. model)

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



just right
(3rd order polinom. model)

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



overfit
(higher ord. polinom. Model)

$$h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + ... + \theta_{16} x^n$$

Overfitting problem

Overfitting: If we have too many features (x1,...x100) the learned model may fit the training data very well but fails to generalize to new examples.

$x_1 =$ size of house
$x_2 =$ no. of bedrooms
$x_3 =$ no. of floors
$x_4 =$ age of house
$x_5 =$ average income in neighborhood
$x_6 =$ kitchen size
$\vdots$
x_{100}

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

Regularized Logistic Regression

Regularization is a popular method in ML to prevent overfitting by reducing the model parameters θ towards zero

1 Ridge Regression (L2 norm)

— Keep all the features, but reduces the magnitude of θ.
— Works well when each of the features contributes a bit to predict y.

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

2 Lasso Regression (L1 norm)

- May shrink some coefficients of θ to exactly zero.
- Serve as a feature selection tools (reduces the number of features).

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

Small λ => lower bias, higher variance
High λ => higher bias, lower variance

universidade de aveiro

RECOMMENDED BIBLIOGRAPHY

Books:

- Tom Mitchell, Machine Learning. McGraw-Hill, 1997.
- Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems. Aurélien Géron. O'Reilly
- Python Machine Learning. Sebastian Raschka. Packt Publishing
- F. Chollet. Deep Learning in Python, Manning, 2018.
- I. Goodfellow, Y. Bengio, A. Courville. Deep Learning, MIT Press, 2016.

Courses:

- http://cs229.stanford.edu/
- MOOC (Massive Open Online Courses) e.g. https://www.coursera.org/