Mathematical techniques in data science

Vu Dinh

Departments of Mathematical Sciences
University of Delaware

February 13rd, 2019

General information

- Classes:

  MWF 9:05am-9:55am, ISE Lab 222
- Office hours: Ewing Hall 312
  - Tuesday 3pm-4pm
  - Wednesday 4:30pm-5:30pm
  - By appointment
- Website: `http://vucdinh.github.io/m637s19`

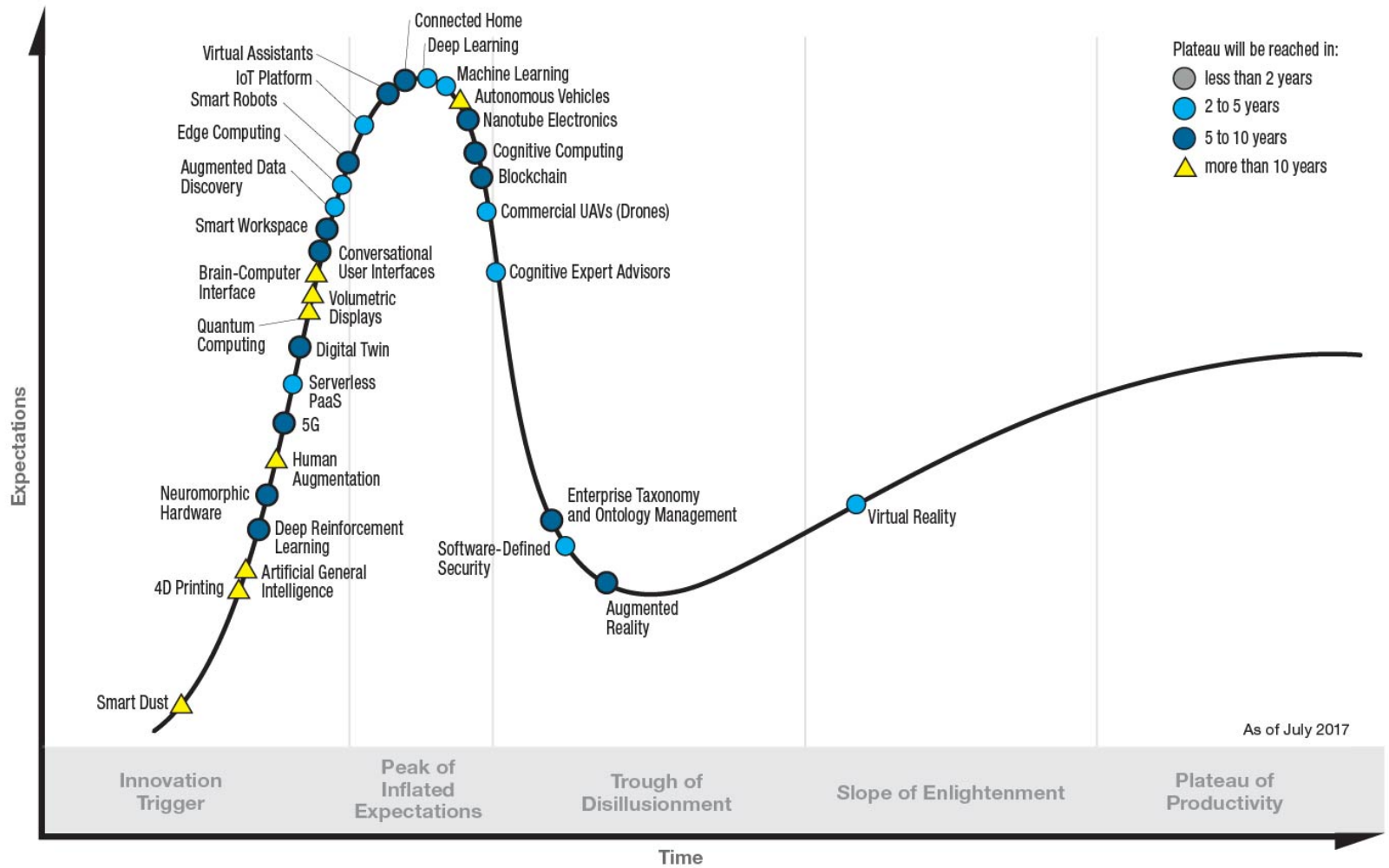
Data science

- is an interdisciplinary field that uses scientific methods, processes, algorithms and systems to extract knowledge and insights from data in various forms, both structured and unstructured
- is a concept to unify statistics, data analysis, machine learning and their related methods in order to understand and analyze actual phenomena with data
- employs techniques and theories drawn from many fields within the context of mathematics, statistics, information science, and computer science
- is a buzzword with no clear meaning

Goals of the course

- Become familiar with the basic methods used to analyze modern datasets.
- Understand the mathematical theory and the standard models used in data science
- Understand how to select a good model for data
- Be able to analyze datasets using a modern programming language such as Python or R

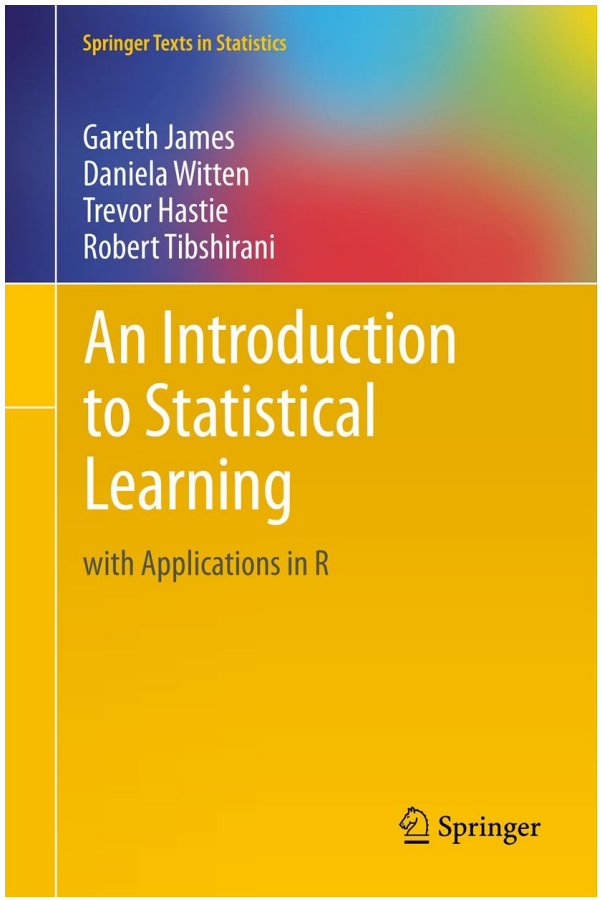
Further goals



Further goals

- Sketch the current landscape of data science and position yourself in the field
- Understand the strengths and weaknesses of current popular approaches in data science
- Think about how the field would change and which abstract ideas would become useful

Textbook



An Introduction to Statistical Learning.
James, Witten, Hastie, and Tibshirani.

The pdf of the book is available for free at
`http://www-bcf.usc.edu/~gareth/ISL/`

Topics

The materials of the course can be organized

- By problems:
  - Classification
  - Regression
  - Clustering
  - Manifold learning
- By methods:
  - Regression-based methods
  - Kernel methods
  - Tree-based methods
  - Network-based methods
- By learning settings:
  - Standard setting
  - Online learning
  - Reinforcement learning
  - Active learning
- By meta-level techniques:
  - Regularization
  - Boosting
  - Bootstrapping
  - Bayesian learning

Tentative schedule

Week	Chapter
1	Chapter 2: Intro to statistical learning
2	Chapter 4: Classification
3	Chapter 9: Support vector machine and kernels
4, 5	Chapter 3: Linear regression
6	Chapter 8: Tree-based methods + Random forrest
7	Neural network
8	
9	Bootstrap and cross-validation + Bayesian methods + UQ
10	Clustering: K-means –> Spectral Clustering
11	PCA –> Manifold learning
12, 13	Reinforcement learning/Online learning/Active learning
14	Project presentation

Data analysis

We will use Python during the course. A good Python tutorial is available at

http://www.scipy-lectures.org/

If you have never used Python before, I recommend using Anaconda Python 3.7

https://www.continuum.io/

It contains all the packages we will need.

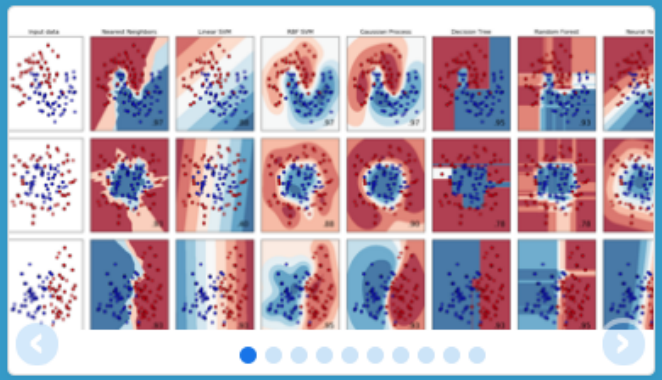
scikit-learn

scikit learn

Home Installation Documentation Examples

Google Custom Search

Fork me on GitHub



scikit-learn

Machine Learning in Python

- Simple and efficient tools for data mining and data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license

Classification

Identifying to which category an object belongs to.

Applications: Spam detection, Image recognition.
Algorithms: SVM, nearest neighbors, random forest, ... — Examples

Regression

Predicting a continuous-valued attribute associated with an object.

Applications: Drug response, Stock prices.
Algorithms: SVR, ridge regression, Lasso, ... — Examples

Clustering

Automatic grouping of similar objects into sets.

Applications: Customer segmentation, Grouping experiment outcomes
Algorithms: k-Means, spectral clustering, mean-shift, ... — Examples

Dimensionality reduction

Reducing the number of random variables to consider.

Applications: Visualization, Increased efficiency
Algorithms: PCA, feature selection, non-negative matrix factorization. — Examples

Model selection

Comparing, validating and choosing parameters and models.

Goal: Improved accuracy via parameter tuning
Modules: grid search, cross validation, metrics. — Examples

Preprocessing

Feature extraction and normalization.

Application: Transforming input data such as text for use with machine learning algorithms.
Modules: preprocessing, feature extraction. — Examples

Evaluation

- Overall scores will be computed as follows:
  - Homework (theoretical + programming problems): 60%
  - Class project: 40%
- Here are the letter grades you can achieve according to your overall score.

$\geq 95\%$ At least A
$\geq 90\%$ At least A-
$\geq 80\%$ At least B-
$\geq 70\%$ At least C-
$\geq 60\%$ At least D-
$< 60\%$ F

Class project

- Group projects: 3-4 people
- The groups should be formed by the end of Week 4
- There are two different tracks for class projects
  - Data-oriented projects
    - Pick a practical learning problem with a dataset
    - Analyze the dataset
    - Present the topic at the end of the semester
  - Theory-oriented projects
    - Pick an advanced concept/algorithm/learning setting
    - Read related literatures
    - Provide a demo of the topic using Python
    - Coordinate with me to present the topic in a lecture
    - Topics: learning complexity, manifold learning, spectral clustering, boosting, online learning, active learning, reinforcement learning, density estimation,...

An introduction to statistical learning

What is Machine Learning?

- Machine learning studies computer algorithms for learning to do stuffs
- The learning that is being done is always based on some sort of observations or data, such as examples, direct experience, or instructions
- The emphasis of machine learning is on automatic methods.

Examples of machine learning problems

- optical character recognition: categorize images of handwritten characters by the letters represented
- spam filtering: identify email messages as spam or non-spam
- topic spotting: categorize news articles as to whether they are about politics, sports, entertainment, etc
- weather prediction: predict, for instance, whether or not it will rain tomorrow

A tiny learning problem

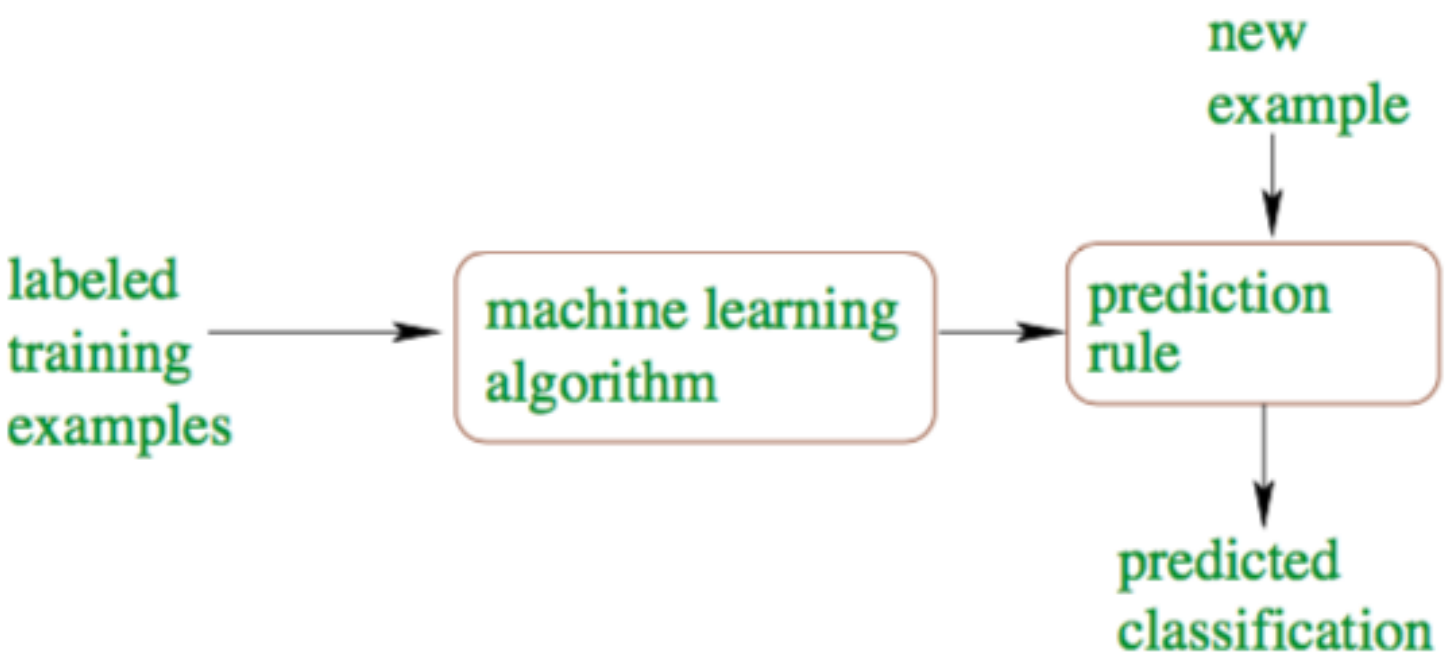
example	label
train	
ant	−
bat	+
dolphin	−
leopard	+
sea lion	−
zebra	+
shark	−
mouse	+
chicken	−
test	
tiger	
tuna	
platypus	

Our thinking process

example	label
train	
ant	−
bat	+
dolphin	−
leopard	+
sea lion	−
zebra	+
shark	−
mouse	+
chicken	−
test	
tiger	
tuna	
platypus	

- Quickly go through (a set) of many possible decision rules
- Find a simple rule that fits the data, use it for prediction

Diagram of a typical supervised learning problem



Supervised learning: learning a function that maps an input to an output based on example input-output pairs

Supervised learning: standard setting

- Given: a sequence of label data $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ sampled (independently and identically) from an unknown distribution $P_{X,Y}$
- x_i is the feature vector of the i^{th} example and y_i is its label
- a learning algorithm seeks a function $h : \mathcal{X} \to \mathcal{Y}$, where $\mathcal{X}$ is the input space and $\mathcal{Y}$ is the output space
- The function h is an element of some space of possible functions $\mathcal{H}$, usually called the *hypothesis space*.

Supervised learning: standard setting

- The function h is an element of some space of possible functions $\mathcal{H}$, usually called the *hypothesis space*
- In order to measure how well a function fits the training data, a *loss function*

$$L : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}^{\geq 0}$$

  is defined
- e.g.: in the previous example, we can define

$$L(+,+) = L(-,-) = 0, \quad L(+,-) = L(-,+) = 1$$

- For training example (x_i, y_i) and a hypothesis h, the loss of predicting the value $h(x_i)$ is $L(y_i, h(x_i))$

Risk and empirical risk

- With a pre-defined loss function, the "optimal hypothesis" is the minimizer over $\mathcal{H}$ of the risk function

$$R(h) = E_{(X,Y)\sim P}[L(Y, h(X))]$$

- Since P is unknown, the simplest approach is to approximate the risk function by the empirical risk

$$R_n(h) = \frac{1}{n}\sum_{i=1}^{n} L(y_i, h(x_i))$$

- The empirical risk minimizer (ERM): minimizer of the empirical risk function

Learning complexity

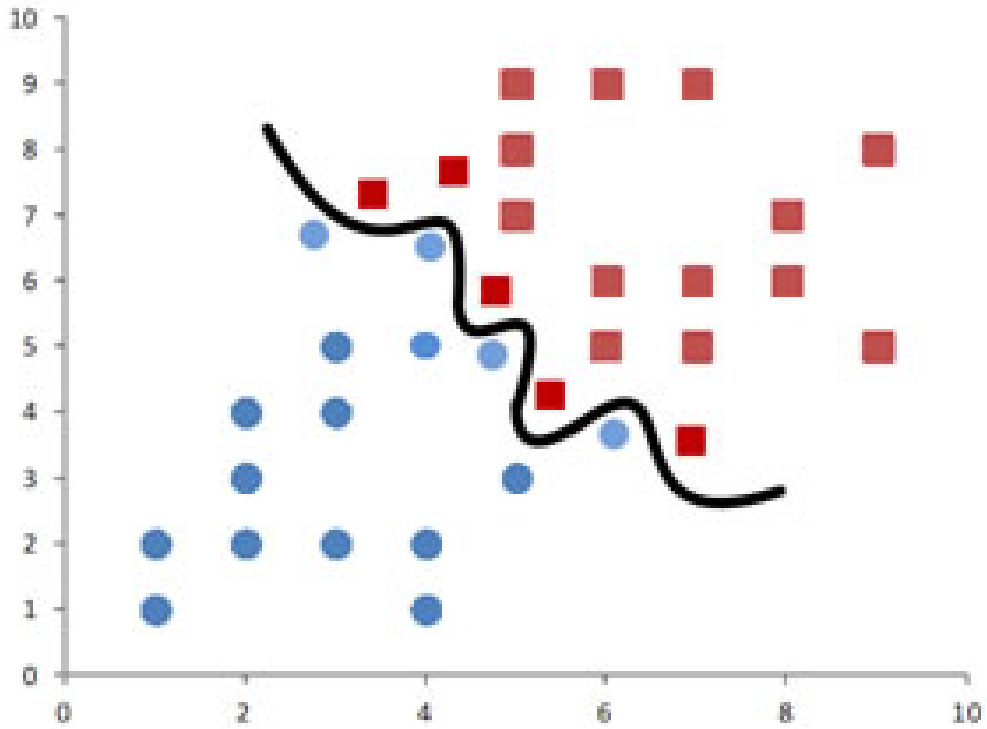
- The central idea of machine learning is that *the past informs the future*, which means that in general

$$R_n(h) = \frac{1}{n}\sum_{i=1}^{n} L(y_i, h(x_i)) \approx E_{(X,Y)\sim P}[L(Y, h(X))] = R(h)$$

  uniformly on $\mathcal{H}$

- This requires that the hypothesis space $\mathcal{H}$ needs not be too large, otherwise overfitting will occur

Overfitting



Choosing a proper hypothesis space plays a central role in statistical learning