An introduction to Machine Learning

E. Rachelson

isae
SUPAERO

Outline

1. General introduction and motivation
2. A geometrical approach to ML
3. A probabilistic approach to ML
4. From mimicking the human brain to Artificial Neural Networks
5. Deep Learning
6. Commitee-based methods: Decision Trees and Boosting
7. Commitee-based methods: Bagging and Random Forests

Syllabus

This course offers a discovery of the landscape of Machine Learning through its key algorithms. Although the first session tries to cover the full span of Machine Learning techniques, the subsequent sessions will focus on the Supervized Learning problem and will categorize the algorithms from four distinct points of view (the Bayesian perspective, linear separation, neural networks and ensemble methods). The approach taken mixes voluntarily hands-on practice in Python with theoretical and mathematical understanding of the methods. At the end of the course you will be able to make an informed choice between the main families of ML algorithms depending on the problem at hand, you will have an understanding of the algorithmic and mathematical properties of each family of methods and you will have a basic practical knowledge of the scikit-learn and keras Python libraries.

Course goals

By the end of the class, you should be able to:

- implement a generic workflow of data analysis for your application field;
- know the main bottlenecks and challenges of data-driven approaches;
- link some field problems to their formal Machine Learning counterparts;
- know the main categories of Machine Learning algorithms and which formal problem they solve;
- know the name and principles of some key methods in Machine Learning:
  - SVM and kernel methods,
  - Naive Bayes Classification,
  - Gaussian Processes,
  - Artificial Neural Networks and Deep Learning,
  - Decision Trees,
  - Ensemble methods: Boosting, Bagging, Random Forests;
- know the basics of scikit-learn and keras.

Course material

`https://github.com/erachelson/MLclass`

Coming soon.

Machine Learning

Let's talk about Machine Learning.

Keywords?
Applications?
Purpose?
Algorithms?

From tasks, to data, to ML

For all these keywords, let's fill the table below, to build a common understanding of:

- the nature of data at stake
- the different tasks to automate
- the difficulties

Use case	Type of data	Properties of data	Task to automate	Difficulties	Comments

The Data Scientist perspective

Identified needs

Let's take the example of Predictive Maintenance.

We would like to build automated tools for the following tasks:

- Visualize system state
- Identify anomalies
- Predict Remaining Useful Life (RUL) / Time To Failure (TTF)
- Predict failure occurrence or probability at a given horizon

All this, in order to base our maintenance strategy on the (inferred) system state, rather than a general statistical trend.

Can you relate this task decomposition to the other use-cases we've seen earlier?

Traditionally, all this is based on user expertise.
Let's take a data-driven approach.

Data analysis workflow

1. Collect

- Sensors deployment
- Historical data collection
- Integrated storage (datawarehouses) and retrieval issues

→ Extract-Transform-Load (ETL) process
More on ETL: [link].
The *data engineer*'s job: data quality, management, availability.

Data analysis workflow

1. Collect
2. Analyze

- data cleaning
- feature selection / engineering
- performance criteria
- algorithm selection
- parameters tuning

The *data analyst* or *data scientist*'s job.
But can't be disconnected from field engineers on the task.

Data analysis workflow

1. Collect
2. Analyze
3. Deploy

- Deploy solution in your operational process
- Make things usable

Data analysis workflow

1. Collect
2. Analyze
3. Deploy
4. Decide

- Improve your decisions

End-user.
Job title depends on your professional field.

Data analysis workflow

1. Collect
2. Analyze
3. Deploy
4. Decide

Need to automate as many steps as possible in this workflow
→ data-driven approaches
→ Machine Learning for step 2

A word on data quality

- amount of data: data is often abundant but crucial data is often scarce
- noise, errors, missing data, outdated data: reliability
- high-dimensional data
- class imbalance
- heterogeneous data (scalars, booleans, time series, images, text, . . .)

All these will influence your algorithmic design or choices.

So let's talk about algorithms to see how we can solve the problems listed earlier.

Machine Learning

Machines that learn?

Let's try to give a general definition.

Machine Learning

Machines that learn?
Let's try to give a general definition.

Machine learning is a field of computer science that gives computer systems the ability to "learn" (i.e. progressively improve performance on a specific task) with data, without being explicitly programmed.

(Wikipedia)

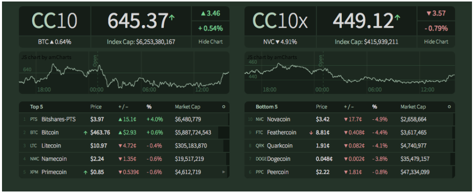
ML examples

- Given 20 years of clinical data, will this patient have a second heart attack in the next 5 years?

Image sources: Wikimedia commons

ML examples

- Given 20 years of clinical data, will this patient have a second heart attack in the next 5 years?
- What price for this stock, 6 months from now?



CC10 645.37↑ ▲3.46 +0.54%

BTC ▲0.64% Index Cap: $6,253,380,167 Hide Chart

		Top 5		Price	+/–	%	Market Cap
1	PTS	Bitshares-PTS		$3.97	▲15.1¢	+4.0%	$6,480,779
2	BTC	Bitcoin	↑	$463.76	▲$2.93	+0.6%	$5,887,724,543
3	LTC	Litecoin		$10.97	▼4.72¢	-0.4%	$305,183,870
4	NMC	Namecoin		$2.24	▼1.35¢	-0.6%	$19,517,219
5	XPM	Primecoin	↑	$0.85	▼0.539¢	-0.6%	$4,612,719

CC10x 449.12↑ ▼3.57 -0.79%

NVC ▼4.91% Index Cap: $415,939,211 Hide Chart

		Bottom 5		Price	+/–	%	Market Cap
10	NVC	Novacoin		$3.42	▼17.7¢	-4.9%	$2,658,664
9	FTC	Feathercoin	↓	8.81¢	▼0.408¢	-4.4%	$3,617,465
8	QRK	Quarkcoin		1.91¢	▼0.082¢	-4.1%	$4,740,977
7	DOGE	Dogecoin		0.048¢	▼0.002¢	-3.8%	$35,479,157
6	PPC	Peercoin		$2.22	▼1.81¢	-0.8%	$47,334,099

Image sources: Wikimedia commons

ML examples

- Given 20 years of clinical data, will this patient have a second heart attack in the next 5 years?
- What price for this stock, 6 months from now?
- Is this handwritten number a 7?

Image sources: Seven.jpg

ML examples

- Given 20 years of clinical data, will this patient have a second heart attack in the next 5 years?
- What price for this stock, 6 months from now?
- Is this handwritten number a 7?
- Is this e-mail a spam?

Enlarge your thesis!

Image sources: Iconfinder

ML examples

- Given 20 years of clinical data, will this patient have a second heart attack in the next 5 years?
- What price for this stock, 6 months from now?
- Is this handwritten number a 7?
- Is this e-mail a spam?
- Can I cluster together customers? press articles? genes?

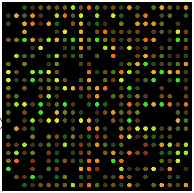


Image sources: People.jpg / Writing to Discuss: Use of a Clustering Technique / DNA microarray

ML examples

- Given 20 years of clinical data, will this patient have a second heart attack in the next 5 years?
- What price for this stock, 6 months from now?
- Is this handwritten number a 7?
- Is this e-mail a spam?
- Can I cluster together customers? press articles? genes?
- What is the best strategy when playing video games? or poker?

COUNTER STRIKE
SOURCE
THE MOST POPULAR ONLINE ACTION GAME OF ALL TIME!
ALSO INCLUDES: HALF-LIFE 2: DEATHMATCH
DAY OF DEFEAT: SOURCE
M
VALVE

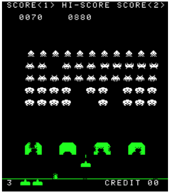


Image sources: CS:source / Space invaders / poker

ML tasks

What does ML do? 3 main tasks.

Task	**Supervized Learning**	**Unsupervized Learning**	**Reinforcement Learning**
Goal	Learn a function, $f(x) = y$	Find groups and correlations, $x \in C$	Optimal control, $f(x) = u \; / \; \max \sum r$
Data	$\{(x, y)\}$	$\{x\}$	$\{(x, u, r, x')\}$
Sub-task	Classification, Regression	Clustering, Density estimation, Dimensionnality reduction	Value estimation, Policy optimization
Algo ex.	Neural Networks, SVM, Random Forests	k-means, PCA, HCA	Q-learning
Appli ex.	Spam filtering, load inference	Topic models, dataviz	Atari games, robotics

Vocabulary

inputs	outputs
independent variables	dependent variables
predictors	responses
features	targets
X (random variables)	Y (random variables)
x_i (observation of X)	y_i (observation of Y)

Evaluation criteria 1/3

Evaluating supervized ML methods: what do we really want?

Ability to fit the training data (regression):

- Mean Square Error, coeffient of determination.

$$MSE = \frac{1}{N} \sum_i (y_i - f(x_i))^2$$

$$R^2 = 1 - \frac{\sum_i (y_i - f(x_i))^2}{\sum_i (y_i - \bar{y})^2}$$

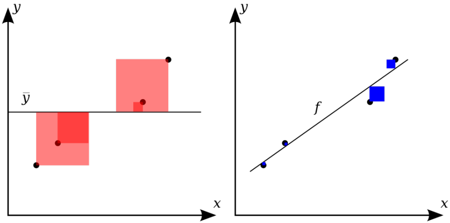


Image source: Wikimedia commons

Evaluation criteria 2/3

Evaluating supervized ML methods: what do we really want?

Ability to fit the training data (classification):

- Accuracy, TP, FP, confusion matrix [link]

$$Acc = \frac{TP + TN}{TP + FP + TN + FN}$$

- ROC, AUC, [link]

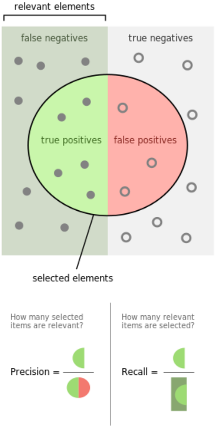


Image source: Wikimedia commons

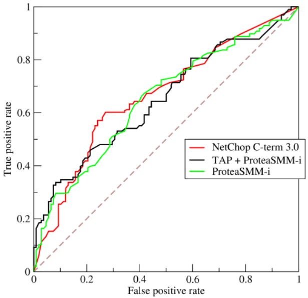


Image source: Wikimedia commons

Evaluation criteria 3/3

Evaluating supervized ML methods: what do we really want?

Ability to generalize:

- Goal: filter out noise, avoid overfitting, generalize to unseen cases.
- ML Notions:
  - maximize margin
  - minimize difference btw class distributions (cross-entropy)

$$H(p,\hat{p}) = \sum_i p(x_i)\log(\hat{p}(x_i)) = \mathbb{E}_p(\log(\hat{p}))$$

Learning contexts

Different kinds of learning contexts:

Context	Sample source
▸ Offline, batch, non-interactive	all samples are given at once
▸ Online, incremental	samples arrive one after the other
▸ Active	the alg. asks for the next sample

Misconceptions and clarifications

AI ML is only a small (currently fashionable) part of Artificial Intelligence.

BD Big Data refers to working with datasets that have large Volume, Variety, Velocity (, Veracity, and Value).

DL Deep Learning is Machine Learning with Deep Neural Networks.

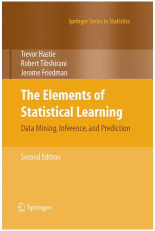
threat ML / Data Science / Big Data are as much of a threat (to jobs, the society, the economy. . .) as the combustion engine was in the XIXth century.

ML software

Software:

- Many free libraries: scikit-learn, tensorflow, caffe. . . check `www.mloss.org` if you're curious.
- Free environments: Weka, RStudio. . .
- Commercial embedded solutions (more or less specialized): Matlab, IBM, Microsoft. . .

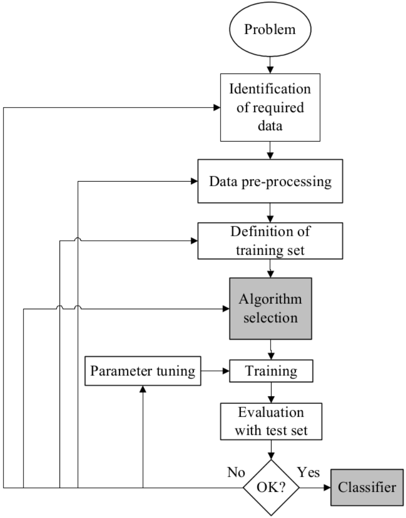
Reference textbooks


Springer Series in Statistics
Trevor Hastie
Robert Tibshirani
Jerome Friedman
The Elements of Statistical Learning
Data Mining, Inference, and Prediction
Second Edition
Springer

The Elements of Statistical Learning, second edition.
Trevor Hastie, Robert Tibshirani, Jerome Friedman.
Springer series in Statistics, 2009.

Other (excellent) references:
Machine Learning. T. M. Mitchell.
Pattern Recognition and Machine Learning. C. Bishop.
Deep Learning. I. Goodfellow, Y. Bengio, A. Courville.
Hands-on ML with Scikit-Learn and Tensorflow. A. Géron.

The process of (Un)Supervized Learning



From **Supervized Machine Learning: A Review of Classification Techniques**, S. B. Kotsiantis, *Informatica*, 31:249–268, 2007.

Relating your needs and ML

Back to the example of Predictive Maintenance tasks.

- Visualizing system state

  → Dimensionnality reduction (Unsupervized learning)
- Detecting anomalies

  → Density estimation (Unsupervized learning)
- Predicting RUL or TTF

  → Regression (Supervized learning)
- Predicting failure in *N* cycles

  → Classification (Supervized learning)

Relating your needs and ML

Back to the example of Predictive Maintenance tasks.

- Visualizing system state
  → Dimensionnality reduction (Unsupervized learning)
- Detecting anomalies
  → Density estimation (Unsupervized learning)
- Predicting RUL or TTF
  → Regression (Supervized learning)
- Predicting failure in *N* cycles
  → Classification (Supervized learning)

Thinking like a Maintenance Engineer:
How can I monitor my system to manage my maintenance operations?
Thinking like a Data Scientist:
Is this a supervized or an unsupervized problem? What available data?

Relate this example to your own field.
Now you can start discussing with data scientists to design together the most appropriate method for your data and your problem.

A word on scikit-learn

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
- Well documented, with lots of examples

`http://scikit-learn.org`

Let's take a look at the documentation's table of contents to grasp a few more keywords.

Extra things

- Installing anaconda, jupyter, etc.
- ML research: arXiv, JMLR, MLJ, IEEE PAMI, NIPS, ICML, ICLR...
- datasets: Kaggle, UCI, ImageNet, CIFAR
- Other algorithms? (scikit-learn documentation or other notebooks)
- Dataviz: upstream methods (PCA...) and storytelling (Tableau...)
- What should I look for in a data scientist's CV?

Course goals

By the end of the class, you should be able to:

- implement a generic workflow of data analysis for your application field;
- know the main bottlenecks and challenges of data-driven approaches;
- link some field problems to their formal Machine Learning counterparts;
- know the main categories of Machine Learning algorithms and which formal problem they solve;
- know the name and principles of some key methods in Machine Learning:
  - SVM and kernel methods,
  - Naive Bayes Classification,
  - Gaussian Processes,
  - Artificial Neural Networks and Deep Learning,
  - Decision Trees,
  - Ensemble methods: Boosting, Bagging, Random Forests;
- know the basics of scikit-learn and keras.

What you should expect in the remainder of this class

- As many intuitive notions as possible,
- . . . but also quite a bit of (hopefully painless) math,
- . . . and a fair amount of hands-on manipulations and demos.