From Field problems to Machine Learning

An introduction to the Data Science workflow and a motivation to understand Machine Learning

E. Rachelson

ISAE SUPAERO

Schedule

1. General introduction and motivation
   How does ML fit within your business process.
   Why you should take time to understand what's under the hood in ML.
2. The importance of data pre-processing
   A practical illustration.
3. A geometrical approach to ML
   Support Vector Machines and a bit of kernel theory.
4. A probabilistic approach to ML
   Naive Bayes Classification and Gaussian Processes.
5. From mimicking the human brain to Artificial Neural Networks
   Artificial Neural Networks.
   Deep Learning.
6. Ensembles of explainable models
   Decision Trees, Bagging and Random Forests.
7. Practical session, discussion and conclusion

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,
  - kernel methods,
  - Naive Bayes Classification,
  - Gaussian Processes,
  - Artificial Neural Networks,
  - Deep Learning,
  - Random Forests;
- know the existence of scikit-learn and its API.

Course material

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

Case studies

Let's list and discuss some cases from your experience and from the literature.

- [Your cases here!]
- Predictive maintenance
- Market segmentation
- Demand forecast
- Preliminary design studies
- Clinical diagnosis
- Documentation management.
- Satellite imaging

From tasks, to data, to ML

For all these cases, 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?

Traditionaly, 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

$\rightarrow$ 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. Predict

- Make predictions on new test cases
- Deploy solution in your operational process
- Make things usable

Data analysis workflow

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

- Improve your decisions

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

Data analysis workflow

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

Need to automate as many steps as possible in this workflow
$\rightarrow$ data-driven approaches
$\rightarrow$ Machine Learning for step 2 (and 3)

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.

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?

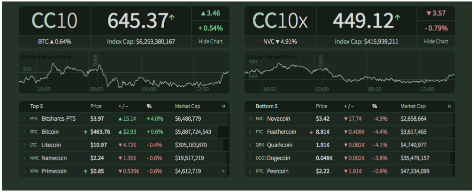


Image sources: Wikimedia commons

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?

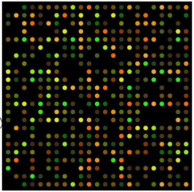


Image sources: People.jpg / Writing to Discuss: Use of a Clustering Technique / DNA microarray

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 Counter Strike? or poker?

Image sources: CS:source / 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

Evaluation criteria

Evaluating ML methods? What do we really want?

Ability to fit the training data:

- Regression: Mean Square Error
- Classification: Accuracy, TP, FP, ROC, AUC. . .
  cf. this Wikipedia article
- Clustering: similarity scores

Ability to generalize:

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

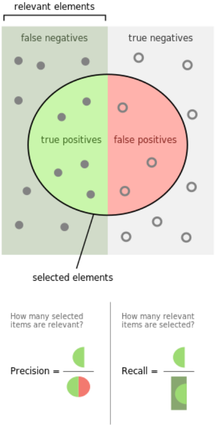


Image source: Wikimedia commons

Misconceptions and clarifications

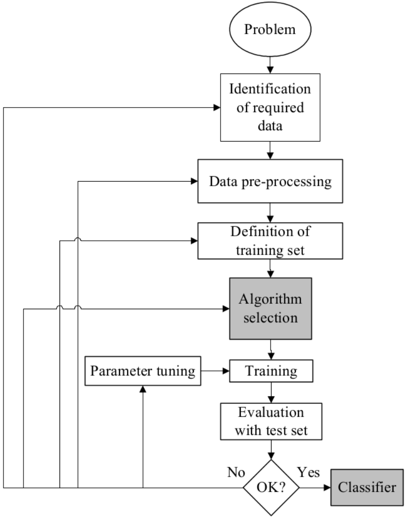
- **AI** ML is only a small (currently fashionable) part of Artificial Intelligence.
- **BD** Big Data refers to working with datasets that have large Volume, Variety, Velocity (, Veracity, and Value).
- **DL** Deep Learning is Machine Learning with Deep Neural Networks.
- **threat** ML / Data Science / 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…

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
  $\rightarrow$ Dimensionnality reduction (Unsupervized learning)
- Detecting anomalies
  $\rightarrow$ Density estimation (Unsupervized learning)
- Predicting RUL or TTF
  $\rightarrow$ Regression (Supervized learning)
- Predicting failure in N cycles
  $\rightarrow$ Classification (Supervized learning)

Relating your needs and ML

Back to the example of Predictive Maintenance tasks.

- Visualizing system state
  $\rightarrow$ Dimensionnality reduction (Unsupervized learning)
- Detecting anomalies
  $\rightarrow$ Density estimation (Unsupervized learning)
- Predicting RUL or TTF
  $\rightarrow$ Regression (Supervized learning)
- Predicting failure in N cycles
  $\rightarrow$ 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.

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,
  - kernel methods,
  - Naive Bayes Classification,
  - Gaussian Processes,
  - Artificial Neural Networks,
  - Deep Learning,
  - Random Forests;
- know the existence of scikit-learn and its API.

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.

Schedule

1. General introduction and motivation ✓
   How does ML fit within your business process.
   Why you should take time to understand what's under the hood in ML.
2. The importance of data pre-processing
   A practical illustration.
3. A geometrical approach to ML
   Support Vector Machines and a bit of kernel theory.
4. A probabilistic approach to ML
   Naive Bayes Classification and Gaussian Processes.
5. From mimicking the human brain to Artificial Neural Networks
   Artificial Neural Networks.
   Deep Learning.
6. Ensembles of explainable models
   Decision Trees, Bagging and Random Forests.
7. Practical session, discussion and conclusion

The importance of data pre-processing

Images, text, video, sound, measurement time series, continuous or discrete variables, missing data...

- $\rightarrow$ filtering out noise and irrelevant data.
  scaling, filtering, reducing...
- $\rightarrow$ data- and application-specific procedures.
  domain knowledge leverages non-representative datasets.

$\Rightarrow$ Crucial elements for a good start.

Never neglect the pre-processing.

Schedule

1. General introduction and motivation ✓
   How does ML fit within your business process.
   Why you should take time to understand what's under the hood in ML.
2. The importance of data pre-processing ✓
   A practical illustration.
3. A geometrical approach to ML
   Support Vector Machines and a bit of kernel theory.
4. A probabilistic approach to ML
   Naive Bayes Classification and Gaussian Processes.
5. From mimicking the human brain to Artificial Neural Networks
   Artificial Neural Networks.
   Deep Learning.
6. Ensembles of explainable models
   Decision Trees, Bagging and Random Forests.
7. Practical session, discussion and conclusion

A geometrical approach to ML

1. Draw a line that sits as far as possible from the data points $\rightarrow$ Support Vector Machines
2. Send all data points in a higher dimension space where they are linearly separable $\rightarrow$ kernel trick

$\Rightarrow$ SVM + kernel trick = Find the optimal separating hyperplane in this higher dimension space, without ever computing the mapping.

- SVM try to separate data by maximizing a geometrical margin
- They are computed offline
- They offer a sparse, robust to class imbalance, and easy to evaluate predictor
- Kernels are a way of enriching (lifting) the data representation so that it becomes linearly separable
- SVMs + kernels offer a versatile method for classification, regression and density estimation
- Link to documentation in scikit-learn

Schedule

1. General introduction and motivation ✓
   How does ML fit within your business process.
   Why you should take time to understand what's under the hood in ML.
2. The importance of data pre-processing ✓
   A practical illustration.
3. A geometrical approach to ML ✓
   Support Vector Machines and a bit of kernel theory.
4. A probabilistic approach to ML
   Naive Bayes Classification and Gaussian Processes.
5. From mimicking the human brain to Artificial Neural Networks
   Artificial Neural Networks.
   Deep Learning.
6. Ensembles of explainable models
   Decision Trees, Bagging and Random Forests.
7. Practical session, discussion and conclusion

A probabilistic approach to ML

Bayesian approach: find y that maximizes $\mathbb{P}(Y = y|\mathrm{data}, X = x)$

This problem of Bayesian inference is hard to solve without additional hypothesis.

A probabilistic approach to ML

Naive Bayes classifiers

- Make a naive, counter-intuitive hypothesis of conditional independence of the feature variables;
- Compute each class' probability for a new example using this hypothesis and picks the most probable one;
- Are a simple, scalable, online method;
- Despite their simplicity, perform surprisingly well and are competitive in many applications.

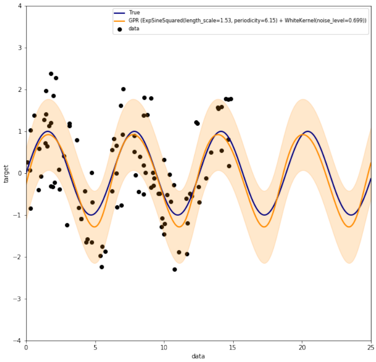
A probabilistic approach to ML

Gaussian Processes

- Compute the most probable function that passes through the data points, given a priori information about how related two data points are (through a covariance kernel);
- Also provide a measure of prediction uncertainty in each point;
- Are computed offline and require an $N \times N$ matrix inversion for N data points in the training set (computationnally costly);
- Careful engineering of covariance kernels can help incorporate priori knowledge into Gaussian Processes;
- Are suitable both for regression and classification.

A probabilistic approach to ML

Note that Gaussian Processes are widely used in preliminary design phases, especially as surrogate models that replace physics computations.



Schedule

1. General introduction and motivation ✓
   How does ML fit within your business process.
   Why you should take time to understand what's under the hood in ML.
2. The importance of data pre-processing ✓
   A practical illustration.
3. A geometrical approach to ML ✓
   Support Vector Machines and a bit of kernel theory.
4. A probabilistic approach to ML ✓
   Naive Bayes Classification and Gaussian Processes.
5. From mimicking the human brain to Artificial Neural Networks
   Artificial Neural Networks.
   Deep Learning.
6. Ensembles of explainable models
   Decision Trees, Bagging and Random Forests.
7. Practical session, discussion and conclusion

Artificial Neural Networks

Schedule

1. General introduction and motivation ✓
   How does ML fit within your business process.
   Why you should take time to understand what's under the hood in ML.
2. The importance of data pre-processing ✓
   A practical illustration.
3. A geometrical approach to ML ✓
   Support Vector Machines and a bit of kernel theory.
4. A probabilistic approach to ML ✓
   Naive Bayes Classification and Gaussian Processes.
5. From mimicking the human brain to Artificial Neural Networks ✓
   Artificial Neural Networks.
   Deep Learning.
6. Ensembles of explainable models
   Decision Trees, Bagging and Random Forests.
7. Practical session, discussion and conclusion

Decision trees

- Easy to interpret and to explain
- Poor representative power
- Greedy growth procedure $\Rightarrow$ suboptimal resulting tree
- Offline training
- Very sensitive to noise in the input data

Random Forests

- RF = decision trees + random feature selection + Bagging
- Robust, scalable, out-of-the-box classifier

$\Rightarrow$ excellent multi-purpose benchmarking algorithm!

Schedule

1. General introduction and motivation ✓
   How does ML fit within your business process.
   Why you should take time to understand what's under the hood in ML.
2. The importance of data pre-processing ✓
   A practical illustration.
3. A geometrical approach to ML ✓
   Support Vector Machines and a bit of kernel theory.
4. A probabilistic approach to ML ✓
   Naive Bayes Classification and Gaussian Processes.
5. From mimicking the human brain to Artificial Neural Networks ✓
   Artificial Neural Networks.
   Deep Learning.
6. Ensembles of explainable models ✓
   Decision Trees, Bagging and Random Forests.
7. Practical session, discussion and conclusion