

---

---

# Machine Learning and AI

- Methods and Algorithms -

---

---

Personnal Notes  
François Bouvier d'Yvoire

CentraleSupélec & Imperial College



# Contents

<b>1</b>	<b>Common Machine Learning algorithms</b>	<b>3</b>
1.1	Linear Regression . . . . .	3
1.1.1	Maximum Likelihood Estimation (MLE) . . . . .	3
1.2	Gradient Descent . . . . .	3
1.2.1	Simple Gradient Descent . . . . .	3
1.2.2	Gradient Descent with Momentum . . . . .	3
1.2.3	Stochastic Gradient Descent . . . . .	3
1.3	Model Selection and Validation . . . . .	3
1.3.1	Cross-Validation . . . . .	3
1.3.2	Marginal Likelihood . . . . .	3
1.4	Bayesian Linear Regression . . . . .	3
1.4.1	Mean and Variance . . . . .	3
1.4.2	Sample function . . . . .	3
<b>2</b>	<b>Argumentation Framework</b>	<b>5</b>
2.1	AA . . . . .	5
2.1.1	Simple AA . . . . .	5
2.1.2	Algorithms for AA . . . . .	6
2.1.3	AA + Suppport . . . . .	7
2.1.4	Argument Mining . . . . .	7
2.1.5	AA with PrefProb . . . . .	7
2.2	ABA . . . . .	7
2.2.1	Simple ABA . . . . .	7
2.2.2	ABA more DDs . . . . .	7
2.2.3	p-acyclic ABA . . . . .	7
2.3	ArgGame . . . . .	7



# Todo list

■ Add bibtex reference . . . . .	3
■ Find better paragraph layout . . . . .	3
■ add ref . . . . .	5
■ add ref to ASPARTIX and CONARG . . . . .	6
■ add algos of computing dispute tree + def of semantics . . . . .	6



# Intro

This document will use the following classification for the machine learning algorithms. However their might be some changes. For exemple, some of them will be part of the commons algorithms and not from their real class.

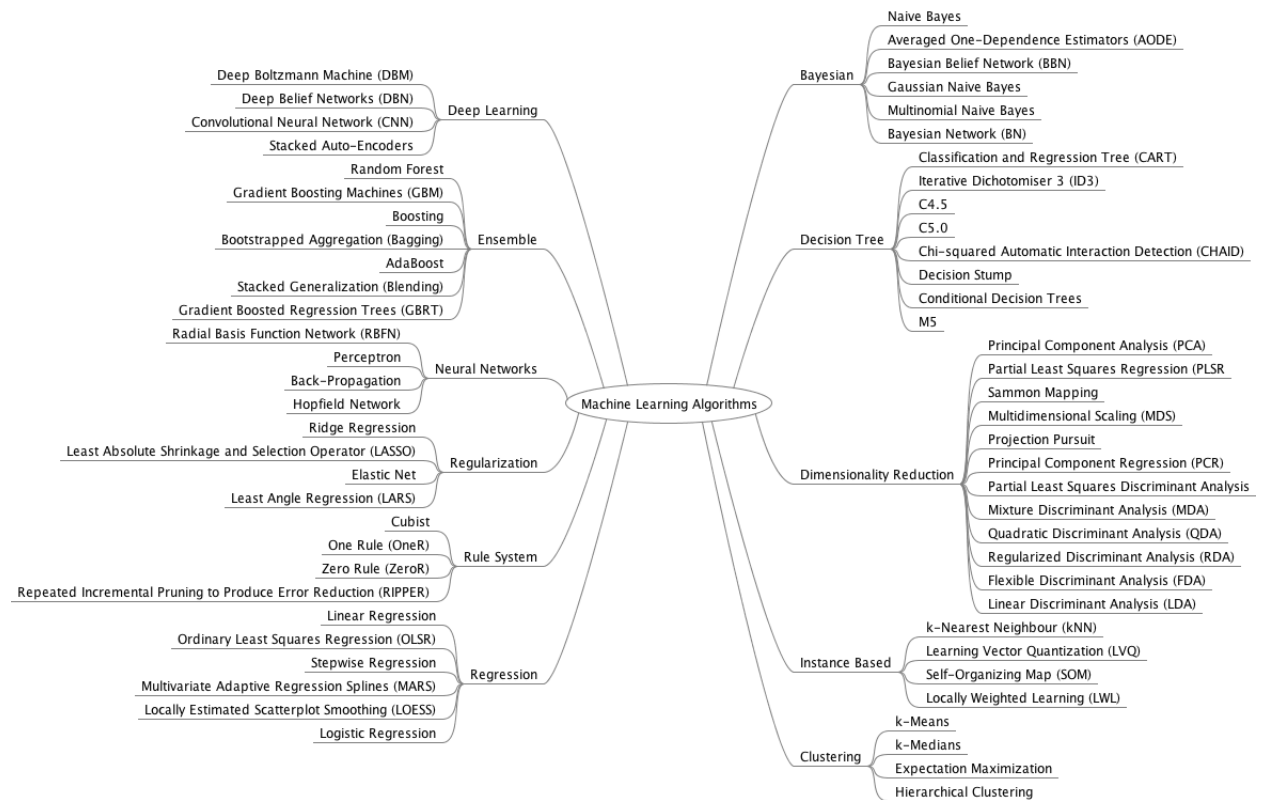


Figure 1 – Simple graph for algorithms classification in ML





# Chapter 1

## Common Machine Learning algorithms

This chapter is dedicated to the most common ML algorithms, a major part of the notes come from the [mml-books.com](http://mml-books.com)

Find better paragraph layout

Add bibtex reference

### 1.1 Linear Regression

#### 1.1.1 Maximum Likelihood Estimation (MLE)

Closed-Form Solution

Maximum A Posteriori Estimation (MAP)

### 1.2 Gradient Descent

#### 1.2.1 Simple Gradient Descent

#### 1.2.2 Gradient Descent with Momentum

#### 1.2.3 Stochastic Gradient Descent

### 1.3 Model Selection and Validation

#### 1.3.1 Cross-Validation

#### 1.3.2 Marginal Likelihood

### 1.4 Bayesian Linear Regression

#### 1.4.1 Mean and Variance

#### 1.4.2 Sample function



## Chapter 2

# Argumentation Framework

This chapter are notes from the Imperial Course Machine Arguing from Francesca Toni.

add ref

**introduction** Argument Framework are a field in AI which provide way of evaluate any debate problem. It is useful to resolve conflict, to explain decision or to deal with incomplete information.

## 2.1 AA

### 2.1.1 Simple AA

#### Definition 1

*an AA framework is a set Args of arguments and a binary relation attacks.  $(\alpha, \beta) \in \text{attacks}$  means  $\alpha$  attacks  $\beta$ .*

**Semantics in AA** In order to define a "winning" set of argument, we need to provide semantics over the the framework. This is like recipes which determine good set of arguments.

#### Definition 2

- *conflict-free*
- *admissible: c-f and attacks each attacking argument.*
- *preferred: maximally admissible.*
- *complete: admissible + contains each argument it defends.*
- *stable: c-f + attacks each argument not in it.*
- *grounded: minimally complete.*

- **sceptically preferred:** Intersection of all preferred.
- **ideal:** maximal admissible and containing all preferred.

### Definition 3

**Semi-stable extension:** complete such as  $A \cup A^+$  is maximal.  $A^+$  is the set of attacked argument by  $A$ .

add ref to ASPARTIX and CONARG

### 2.1.2 Algorithms for AA

**Computing the grounded labelling:** Here is an algorithm to compute a grounded labelling

**Data:** An AA Framework  
**Result:** The grounded Labelling  
 Label all unattacked argument with IN ;  
**while** The IN and the OUT are not stable **do**  
     Label OUT the arguments attacks by IN ;  
     Label IN the arguments only attacked by OUT;  
**end**  
 Label the still unlabelled UNDEC;

**Algorithm 1:** Computing the grounded labelling

**Computing membership in preferred/grounded/ideal extensions;** In order to compute membership, we use Dispute Tree.

We compute a dispute tree for an argument, and apply the different semantics which are easier to compute on a tree than on a graph.

add algos of computing dispute tree + def of semantics

**Computing stable extension:** We use answer set programming with logical program.

**2.1.3 AA + Suppport**

**2.1.4 Argument Mining**

**2.1.5 AA with PrefProb**

**2.2 ABA**

**2.2.1 Simple ABA**

**2.2.2 ABA more DDs**

**2.2.3 p-acyclic ABA**

**2.3 ArgGame**