Machine Learning and AI

- Methods and Algorithms -

Personnal Notes
François Bouvier d'Yvoire

CentraleSupélec & Imperial College

Contents

Todo list

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

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 Abstract Argumentation

2.1.1 Simple AA

Definition 1
*an **AA framework** is a set* Args *of arguments and a binary relation* attacks. $(\alpha, \beta) \in$ attacks *means α attacks β.*

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 prefered.*
- ***ideal:*** *maximal admissible and contain in all prefered (i.e. in the sceptically 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 Grounded extensions Use the same algorithms as grounded labelling, but only output the IN arguments as the grounded extensions.
the grounded extensions in unique.

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

```
Data: An AA Framework
Result: The grounded Labelling
Label all unatacked 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

Data: An AA Framework, and a arg α
Result: Finite or infinite dispute tree corresponding to α
add a root node Label α as proponent ;
while The tree is not stable **do**
  for every $\beta \in$ proponent, and for every (γ, β) add a child-node γ as opponent;
  for every $\beta \in$ opponent, and for every (γ, β) add a node γ as proponent in the limit of one child-node per opponent ;
end

Algorithm 2: Computing dispute Tree

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

2.1.3 AA with Suppport

Bipolar Abstract Argumentation

We add a **Support** relation to a classic AA Framework (BAA = $\langle Args, Attacks, Supports \rangle$). There are different semantics :

Semantics in BAA with deductive support We can deduce an AA Framework from a BAA with $Attacks' = Attacks \cup Attacks_{sup} \cup Attacks_{s-med}$ with

Definition 4

- *$Attacks_{sup}$ is **the supported attacks** $\implies$ α attacks every argument that its supports attacks (supports of supports are supports)*
- *$Attacks_{s-med}$ is the super mediated attacks $\implies$ α attacks every argument whose supports an argument attacked or $attacked_{sup}$ by α*

Then, we apply the AA semantics to $\langle Args, Attacks' \rangle$. Those semantics are the d-X semantics, where X replace every semantic from AA (grounded, complete, etc.)

QuAD Framework We focus here one the QuAD (**Quantitative Argument Debate**) which add a numerical strenght to any argument, and give rule for updating strenght regarding the supporters or attackers.

Definition 5

- *We define the QuAD Framework as $\langle \mathcal{A}, \mathcal{C}, \mathcal{P}, \mathcal{R}, \mathcal{BS} \rangle$ Where $\mathcal{A}$ is the set of answer arguments, $\mathcal{C}$ is the set of con args, $\mathcal{P}$ are the pro args, which are disjoint, $\mathcal{R}$ the relation (with $\mathcal{R}^+$ the supporters and $\mathcal{R}^-$ the attackers) and $\mathcal{BS}$ the Base Score, which give score to each args.*

- *The **Base funcion** is* $f(v_1, v_2) = v_1 + (1 - v_1)v_2$
- *the **aggregation function** as a recursive definition :*

$$\begin{aligned} n = 0 &\implies \mathcal{F}(S) = 0 \\ n = 1 &\implies \mathcal{F}(S) = v_1 \\ n = 2 &\implies \mathcal{F}(S) = f(v_1, v_2) \\ n > 2 &\implies \mathcal{F}(S) = f(\mathcal{F}(S \backslash \{v_n\}, v_n)) \end{aligned}$$

- *the **Combination function** is defined by:*
  $c(v_0, v_a, v_s) = v_0 - v_0 \cdot |v_s - v_a|$ *if* $v_a \geq v_s$
  $c(v_0, v_a, v_s) = v_0 + (1 - v_0) \cdot |v_s - v_a|$ *if* $v_a < v_s$
- *the **Strength Function** is defined as :* $\mathcal{S}(\alpha) = c(\mathcal{BS}(\alpha), \mathcal{F}(SEQ_S(\mathcal{R}^-(\alpha))), \mathcal{F}(SEQ_S(\mathcal{R}^+(\alpha))))$

The DF-QuAD (for **Discontinuity free**-QuAD) algorithm is simply to apply these semantics to a BAA framework, in setting initial weights to any arguments.

2.1.4 Argument Mining

2.1.5 AA with Preference

When taking into account preference in a AA, (in the form $X < Y = X$ is less prefered to Y) we can still use the classic AA semantics but we can improved it by using the preferences informations :

- By **deleting attacks** (an (x, y) attacks succeed iff $x \not< y$)
- By **reversing attacks** (if (x, y) and $x < y$ then we reverse by replacing this attack by (y, x)).
- By **Selecting Amongst extension**, using the preference to select extension with the same extension (ex : differentiate 2 stable extensions). Look at **Rich PAF** for some formal definition.

2.1.6 AA with Probabilities

2.2 Assumption-Based Argumentation

Assumption Based Argumentation introduce arguments as Assumptions $a \in \mathcal{A}$, or deduction σ with premises and conclusions achevied with Rules $R \subseteq \mathcal{R}$(deductive rules, as $q \wedge a \implies p$) defined in a language $\mathcal{L}$

2.2.1 Simple ABA

We can define **attacks** in ABA : an argument $A_1 \vdash \sigma_1$ attacks $A_2 \vdash \sigma_2$ iff σ_1 is the contrary of one of the assumptions in A_2.

2.2.2 ABA more DDs

2.2.3 p-acyclic ABA

Definition 6
*a **positive-acyclic ABA** is a AB framework where the dependancy graph of AB is acyclic.*
*The **dependancy graph** is a graph of all the rules, where assumption ($\mathcal{A}$) are deleted.*

2.3 ArgGame