Bruno Agard

WORKSHOP #3
MACHINE LEARNING FOR PREDICTIVE ANALYSIS

PARTNER IN GROWTH HEC MONTRÉAL

1

Bruno Agard

PARTNER IN GROWTH HEC MONTRÉAL

WELCOME

Workshop#3
Machine Learning for Predictive Analysis

Session#	Day	Date
11	Tuesday	sept. 15th
12	Wednesday	sept. 16th
13	Thursday	sept. 17th
14	Tuesday	sept. 22nd

Start: 1:30 pm

End: 4:30

ERICSSON

2

Bruno Agard

YOUR INSTRUCTOR

Bruno Agard
DEA, Ph.D
Professor Polytechnique Montréal,
Department of Mathematics and industrial engineering

EXPERTISE

- Industrial engineering
- Information technology
- Artificial Intelligence
- Industry 4.0

OTHER PROFESSIONAL IMPLICATIONS:

- Director, laboratory in data intelligence
- Member, laboratory Poly-Industries 4.0
- Member, interuniversity Research Centre CIRRELT
- Member, Canada Research Chair on personal mobility Member , Institute for Data Valorization (IVADO)
- Member, Centre interdisciplinaire de recherche en opérationnalisation du développement durable (CIRODD)

PARTNER IN GROWTH HEC MONTRÉAL

3

Bruno Agard

Workshop#3
Machine Learning for Predictive Analysis

Session#	Day	Date	
11	Tuesday	sept. 15th	Start: 1:30 pm
12	Wednesday	sept. 16th	End: 4:30
13	Thursday	sept. 17th	
14	Tuesday	sept. 22nd	

- Essentials of linear and logistic regression models
- Methods for evaluating the performance of predictive models (learning-validation-test data; cross-validation; ROC curve)
- Regression and classification tree
- Random forest
- « Gradient boosting »
- Neural network – Introduction to multilayer perceptron neural networks

Workshop#4
Non supervised Machine Learning

Session#	Day	Date	
15	Wednesday	sept.23rd	Start: 1:30 pm
16	Thursday	sept. 25th	End: 4:30
17	Tuesday	sept. 29th	
18	Wednesday	sept. 30th	

- Grouping / segmentation analysis: hierarchical and non-hierarchical methods, mixed methods
- Introduction to association rules, methods based on nearest neighbors, filtering methods
- Applications of non-supervised methods to anomaly detection
- Data preparation

4

Bruno Agard

ONLINE TRAINING TIPS

HEC MONTRÉAL

5

Bruno Agard

MAKE THE MOST OF YOUR EXPERIENCE

Keep your microphone on mute when you are not participating

HEC MONTRÉAL

6

PARTICIPATE THROUGHOUT THE TRAINING

Use the **Raise Hand** feature to ask a question

Interact in the **Chat** window

7

MATERIAL YOU SHOULD HAVE

- One pdf file with actual presentation
  - HEC #3.pdf
- One Python Jupyter Notebook file
  - HEC #3.ipynb
- Some data files
  - Reg_1.csv, Class_1.csv
- One computer with
  - Python Jupyter installed
  - The following libraries installed : pandas, numpy, sklearn

8

HOW TO START

1. Start a Python Jupyter Notebook session
2. Load the Python file provided for the Workshop
   - HEC #3.ipynb
3. Identify the path where the data files are located
   - Reg_1.csv, Class_1.csv
4. Update the path variable to find your data file
   - path='../usermane/Desktop/Data/'
5. Run the first block of code, if the head of the file "Reg_1.csv » appers, your in ☺

Step -1

Before to start

9

INTRODUCTION

10

TABLE

- Lines : object / item / customers
- Columns : attributes / variables
- Data :
  - Type : discrete / continuous
  - Value : N/A, U, missing,

Customer	Sexe	Age	Adress	City	Paiement mode	Active
#1	M	35	« 34 St. Denis »	Montréal	Interact	yes
#2	F	27	« 15 bis. Main St. »	Toronto	Credit Card	yes
…	…	…	…	…	…	…
#654332	U	56	« 250 Av. Champlain »	Québec	Interact	no

11

DATABASE

- Set of tables

Customer table

Customer	Sexe	Age	Adress	City	Paiement mode	Active
#1	M	35	« 34 St. Denis »	Montréal	Interact	yes
#2	F	27	« 15 bis. Main St. »	Toronto	Credit Card	yes
…	…	…	…	…	…	…
#654332	U	56	« 250 Av. Champlain »	Québec	Interact	no

Product ref table

Cust.	Intern ref.	Name	Date
#1	#P1564	Phone model 1	date 1
#1	#P3684	Phone model 5	date 3
#1	#MG3685	Modem G	date 3
#1	#VF543699	Video recorder F	date 3
#2	#P985	Phone model 3	date 4
#2	#VF29696	Video recorder F	date 5
…	…	…	
#65432	#VA1236	Video recorder A	date 16965

Phone calls table

Inter ref.	Date	Time	Duration
#P1564	date 1	07:32:45 (UTC-4)	00:02:13
#P65465	date 1	07:32:46 (UTC-4)	00:13:24
#P1564	…	…	…
#P455	date 2	08:13:34 (UTC-2)	00:06:56
#P1564	date 2	08:13:35 (UTC-4)	00:34:28
#P654369	date 2	08:13:35 (UTC-4)	01:27:45
…	…	…	…
#P65432	date n	22:56:12 (UTC-2)	00:12:52

Internet connections table

VideoRec.	Date	Name	Mb	…	…
#MG3685	date 1	07:32:45 (UTC-4)	42	…	…
#ME1236	date 1	07:32:46 (UTC-4)	8	…	…
#MG3685	…	…	…	…	…
#MF5896	date 2	08:13:34 (UTC-2)	2	…	…
#MB65435	date 2	08:13:35 (UTC-4)	12	…	…
#MG3685	date 2	08:13:35 (UTC-4)	27	…	…
…	…	…	…	…	…
#MH236	date n	22:56:12 (UTC-2)	0.7	…	…

12

DATABASE

- Set of tables with relations

Customer table

Customer	Sexe	Age	Adress	City	Paiement mode	Active
#1	M	35	« 34 St. Denis »	Montréal	Interact	yes
#2	F	27	« 15 bis. Main St. »	Toronto	Credit Card	yes
...	...	...	...	...	...	...
#654332	U	56	« 250 Av. Champlain »	Québec	Interact	no

Product ref table

Cust.	Intern ref.	Name	Date
#1	#P1564	Phone model 1	date 1
#1	#P3684	Phone model 5	date 3
#1	#MG3685	Modem G	date 3
#1	#VF543699	Video recorder F	date 3
#2	#P985	Phone model 3	date 4
#2	#VF29696	Video recorder F	date 5
...	...	...	
#65432	#VA1236	Video recorder A	date 16965

Phone calls table

Inter ref.	Date	Time	Duration
#P1564	date 1	07:32:45 (UTC-4)	00:02:13
#P65465	date 1	07:32:46 (UTC-4)	00:13:24
#P1564	...	...	...
#P455	date 2	08:13:34 (UTC-2)	00:06:56
#P1564	date 2	08:13:35 (UTC-4)	00:34:28
#P654369	date 2	08:13:35 (UTC-4)	01:27:45
...	...	...	...
#P65432	date n	22:56:12 (UTC-2)	00:12:52

Internet connections table

VideoRec.	Date	Name	Mb	...	...
#MG3685	date 1	07:32:45 (UTC-4)	42	...	...
#ME1236	date 1	07:32:46 (UTC-4)	8	...	...
#MG3685	...	...	...	...	...
#MF5896	date 2	08:13:34 (UTC-2)	2	...	...
#MB65435	date 2	08:13:35 (UTC-4)	12	...	...
#MG3685	date 2	08:13:35 (UTC-4)	27	...	...
...	...	...	...	...	...
#MH236	date n	22:56:12 (UTC-2)	0.7	...	...

13

DATABASE

- Set of tables with relations

Customer table

Customer	Sexe	Age	Adress	City	Paiement mode	Active
#1	M	35	« 34 St. Denis »	Montréal	Interact	yes
#2	F	27	« 15 bis. Main St. »	Toronto	Credit Card	yes
...	...	...	...	...	...	...
#654332	U	56	« 250 Av. Champlain »	Québec	Interact	no

Product ref table

Cust.	Intern ref.	Name	Date
#1	#P1564	Phone model 1	date 1
#1	#P3684	Phone model 5	date 3
#1	#MG3685	Modem G	date 3
#1	#VF543699	Video recorder F	date 3
#2	#P985	Phone model 3	date 4
#2	#VF29696	Video recorder F	date 5
...	...	...	
#65432	#VA1236	Video recorder A	date 16965

Phone calls table

Inter ref.	Date	Time	Duration
#P1564	date 1	07:32:45 (UTC-4)	00:02:13
#P65465	date 1	07:32:46 (UTC-4)	00:13:24
#P1564	...	...	...
#P455	date 2	08:13:34 (UTC-2)	00:06:56
#P1564	date 2	08:13:35 (UTC-4)	00:34:28
#P654369	date 2	08:13:35 (UTC-4)	01:27:45
...	...	...	...
#P65432	date n	22:56:12 (UTC-2)	00:12:52

Internet connections table

VideoRec.	Date	Name	Mb	...	...
#MG3685	date 1	07:32:45 (UTC-4)	42	...	...
#ME1236	date 1	07:32:46 (UTC-4)	8	...	...
#MG3685	...	...	...	...	...
#MF5896	date 2	08:13:34 (UTC-2)	2	...	...
#MB65435	date 2	08:13:35 (UTC-4)	12	...	...
#MG3685	date 2	08:13:35 (UTC-4)	27	...	...
...	...	...	...	...	...
#MH236	date n	22:56:12 (UTC-2)	0.7	...	...

14

DATABASE

- Set of tables with relations

Customer table

Customer	Sexe	Age	Adress	City	Paiement mode	Active
#1	M	35	« 34 St. Denis »	Montréal	Interact	yes
#2	F	27	« 15 bis. Main St. »	Toronto	Credit Card	yes
...	...	...	...	...	...	...
#654332	U	56	« 250 Av. Champlain »	Québec	Interact	no

Product ref table

Cust.	Intern ref.	Name	Date
#1	#P1564	Phone model 1	date 1
#1	#P3684	Phone model 5	date 3
#1	#MG3685	Modem G	date 3
#1	#VF543699	Video recorder F	date 3
#2	#P985	Phone model 3	date 4
#2	#VF29696	Video recorder F	date 5
...	...	...	
#65432	#VA1236	Video recorder A	date 16965

Phone calls table

Inter ref.	Date	Time	Duration
#P1564	date 1	07:32:45 (UTC-4)	00:02:13
#P65465	date 1	07:32:46 (UTC-4)	00:13:24
#P1564	...	...	...
#P455	date 2	08:13:34 (UTC-2)	00:06:56
#P1564	date 2	08:13:35 (UTC-4)	00:34:28
#P654369	date 2	08:13:35 (UTC-4)	01:27:45
...	...	...	...
#P65432	date n	22:56:12 (UTC-2)	00:12:52

Internet connections table

VideoRec.	Date	Name	Mb	...	...
#MG3685	date 1	07:32:45 (UTC-4)	42	...	...
#ME1236	date 1	07:32:46 (UTC-4)	8	...	...
#MG3685	...	...	...	...	...
#MF5896	date 2	08:13:34 (UTC-2)	2	...	...
#MB65435	date 2	08:13:35 (UTC-4)	12	...	...
#MG3685	date 2	08:13:35 (UTC-4)	27	...	...
...	...	...	...	...	...
#MH236	date n	22:56:12 (UTC-2)	0.7	...	...

15

LARGE SOURCE OF ANALYSES

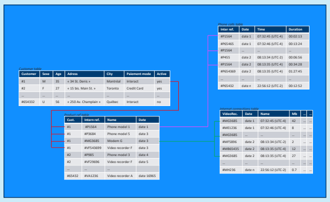


Table of phone calls per customer
Table of Internet connexions per customer
Table of material use per customer / region
Table of customers / region / material ...

	X1	X2	X3	Y1
C1	1	0	1	2
C2	1	1	0	3
C3	1	0	1	2
C4	1	0	1	..
C5	1	0	..	2
C6	..	1	0	3
C7	1	1	0	4

16

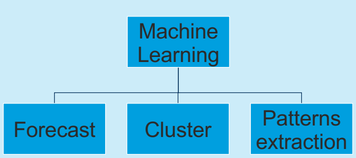
CONTENTS

- Essentials of linear and logistic regression models
- Methods for evaluating the performance of predictive models (learning-validation-test data; cross-validation; ROC curve)
- Regression and classification tree
- Random forest
- « Gradient boosting »
- Neural network – Introduction to multilayer perceptron neural networks

17

MACHINE LEARNING

- Objectives of ML



	X1	X2	X3	Y1
C1	1	0	1	2
C2	1	1	0	3
C3	1	0	1	2
C4	1	0	1	..
C5	1	0	..	2
C6	..	1	0	3
C7	1	1	0	4

18

OBJECTIVES 1/3

- **Forecast**
  - Obj : Determine outputs in function of inputs. Outputs = f(inputs).
  - **Supervised** methods: the analyst selects which variables are inputs / outputs.
    - *Discret* outputs : **Classification methods**
    - *Continuous* outputs : **Estimation methods**
  - Different methods : Regressions, decision trees, decision rules, neural networks…

	X1	X2	X3	Y1
C1	1	0	1	2
C2	1	1	0	3
C3	1	0	1	2
C4	1	0	1	..
C5	1	0	..	2
C6	..	1	0	3
C7	1	1	0	4

	X1	X2	X3	Y1
C1	1	0	1	2
C2	1	1	0	3
C3	1	0	1	2
C4	1	0	1	2
C5	1	0	..	2
C6	..	1	0	3
C7	1	1	0	4

19

OBJECTIVES 2/3

- **Cluster**
  - Obj: identify groups of similar objects
  - **Unsupervised** methods
    - Maximize similarity within each group
    - Maximize dissimilarity between groups
  - Different metrics / methods : k-median, hirarchical algorithms, neural networks…

	X1	X2	X3	Y1
C1	1	0	1	2
C2	1	1	0	3
C3	1	0	1	2
C4	1	0	1	..
C5	1	0	..	2
C6	..	1	0	3
C7	1	1	0	4

	X1	X2	X3	Y1
C1	1	0	1	2
C2	1	1	0	3
C3	1	0	1	2
C4	1	0	1	..
C5	1	0	..	2
C6	..	1	0	3
C7	1	1	0	4

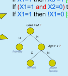
20

OBJECTIVES 3/3

- **Patterns extraction**
  - Obj: Explain/highlight **relations** existing in the data, **associations** between variables.
  - **Supervised and unsupervised** methods
    - Links analysis :
      - Association rules : A $\Rightarrow$ B (support, confiance)
    - Explanatory models (trees)
    - Visualization : simplify data undertanding.

	X1	X2	X3	Y1
C1	1	0	1	2
C2	1	1	0	3
C3	1	0	1	2
C4	1	0	1	..
C5	1	0	..	2
C6	..	1	0	3
C7	1	1	0	4

If X1=1 then X3 =1 [5, 3/5]
If (X1=1 and X2=0) then X3=1 [3, 3/3]
If X1=1 then !X1=0 [5, 100%]



21

Bruno Agard

IN BRIEF…

	X1	X2	X3	Y1
C1	1	0	1	2
C2	1	1	0	3
C3	1	0	1	2
C4	1	0	1	..
C5	1	0	..	2
C6	..	1	0	3
C7	1	1	0	4

Forecasting

	X1	X2	X3	Y1
C1	1	0	1	2
C2	1	1	0	3
C3	1	0	1	2
C4	1	0	1	2
C5	1	0	..	2
C6	..	1	0	3
C7	1	1	0	4

Clustering

	X1	X2	X3	Y1
C1	1	0	1	2
C2	1	1	0	3
C3	1	0	1	2
C4	1	0	1	..
C5	1	0	..	2
C6	..	1	0	3
C7	1	1	0	4

Description

If X1=1 then X3 =1 [5, 3/5]
If (X1=1 and X2=0) then X3=1 [3, 3/3]
If X1=1 then !X1=0 [5, 100%]
…

22

PARTNER IN GROWTH HEC MONTRÉAL

22

Bruno Agard

MACHINE LEARNING

ML is not a magical tool, ML is not new, ML is not recent. ☺

But the combination of :
- Improved power of computers,
- Large accessible sets of data
- « new » computing methods (use of libraries instead of developing them) and philosophy (code sharing),

Permitted the development of ML by the possible use of (many) past and (some) recent mathematical tools, on large (and cheap) accessible amount of data.

ML considers:
- Statistical tools,
- Data manipulation tools,
- Visualisation tools,
- Neural networks,
- And others

Machine Learning
Supervised → Workshop #3
Unsupervised → Workshop #4

ML considers individual data instead of characteristics of populations

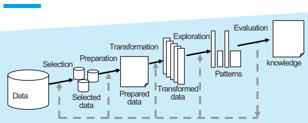
23

PARTNER IN GROWTH HEC MONTRÉAL

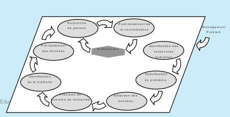
23

Bruno Agard

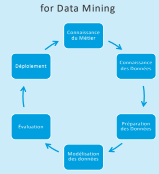
MACHINE LEARNING IS A PROCESS



+



CRISP – DM
Cross Industry Standard Process for Data Mining



24

PARTNER IN GROWTH HEC MONTRÉAL

24

Bruno Agard

ESSENTIALS OF LINEAR AND LOGISTIC REGRESSION MODELS

25

Bruno Agard

IN BRIEF…

	X1	X2	X3	Y1
C1	1	0	1	2
C2	1	1	0	3
C3	1	0	1	2
C4	1	0	1	..
C5	1	0	..	2
C6	..	1	0	3
C7	1	1	0	4

Forecasting

	X1	X2	X3	Y1
C1	1	0	1	2
C2	1	1	0	3
C3	1	0	1	2
C4	1	0	1	2
C5	1	0	..	2
C6	..	1	0	3
C7	1	1	0	4

Clustering

	X1	X2	X3	Y1
C1	1	0	1	2
C2	1	1	0	3
C3	1	0	1	2
C4	1	0	1	..
C5	1	0	..	2
C6	..	1	0	3
C7	1	1	0	4

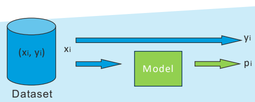
Description

If X1=1 then X3 =1 [5, 3/5]
If (X1=1 and X2=0) then X3=1 [3, 3/3]
If X1=1 then !X1=0 [5, 100%]
…

26

Bruno Agard

MODELS



	X1	X2	X3	Y1
C1	1	0	1	2
C2	1	1	0	3
C3	1	0	1	2
C4	1	0	1	..
C5	1	0	..	2
C6	..	1	0	3
C7	1	1	0	4

Models are dedicated to represent the reality in « something » that can link real observations y_i to conditions x_i.

A model is not reality, it is just a simplified way, eventually comprehensive, we can use to predict what may occur (p_i) in some circumstances (x_i).

If there is a difference between the model predictions (p_i) and what happens in the real data (y_i): the model is wrong, not the real data…

A model may provide predictions p_i that differ to real situation y_i, but if the difference is small enough, we may use it to make predictions.

27

Bruno Agard

LINEAR REGRESSION

28

Bruno Agard

LINEAR REGRESSION

- We have a set of data (x_1, x_2, …x_p) and corresponding output y
- We assume the data follows a model of the following form (linear)

$$y = a_0 + a_1x_1 + a_2x_2 + \ldots + a_px_p + \varepsilon$$

- We compute a_i coefficients that minimise error between real and predicted y
- We evaluate the model performance (which validates or not initial assumption)

29

Bruno Agard

« SIMPLE » LINEAR REGRESSION

- To compute the a_i, the criterion is to minimise the Mean Squared Error

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(p_i - y_i)^2$$

Ex 0 –
Run a first model

Google any function !
See all parameters

- All a_i are supposed “valid” they may be very small
- If 2 (or more) variables are correlated, one may receive most “influence” on a_i

30

LASSO REGRESSION

- *Least Absolute Shrinkage and Selection Operator*
- Similar to "simple" linear regression, but, to compute the a_i the criterion is to minimise the Mean Squared Error, under constraint.
- If variables have small influence, they are removed from the model
  - Permits to select un sub-set of « strong » variables, making the model easier to interpret
  - Works if the number of parameters (ai) is smaller that the size of the data set (number of examples)
- But :
  - If some variables are highly correlated and important for the model, Lasso will keep only one and not consider the others

31

RIDGE REGRESSION

- Similar to Ridge, but different penality
  - Correlated variables share their relative influence, it is called «shrinkage»
- Rq :
  - x_i have to be centered and reduced (z_i), and y_i have to be centered (and could be reduced)

32

ELASTIC-NET

- A mix of Ridge and Lasso
- Proposes a balance between the selection of variables (Lasso) and the shrinkage of correlated variables (Ridge)

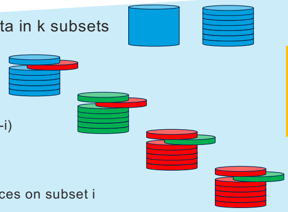
Ex 1-
Run Lasso, Ridge and ElasticNet
Compare the models
Change alphas, observe influence
How to select a « good » alpha ?

33

Bruno Agard

SELECT THE BEST PARAMETER WITH CROSS VALIDATION

- Split the learning data in k subsets
- For i=1 to k
  - Remove subset i
  - Learn on subsets (k-i)
  - Predict for subset i
  - Evaluate performances on subset i
- Keep the parameters from the best iteration



Ex 1- CV
Determine alphas
Compare the models

34

Bruno Agard

Ex 2 -
Load a csv dataset
Learn different prediction models
Set the parameters of the models (alphas)
Compare the models

Ex 2 - CV
Load a csv dataset
Learn different prediction models
Get the best set of parameters (alphas) for the models
Compare the model

35

Bruno Agard

LOGISTIC REGRESSION

36

Bruno Agard

LOGISTIC REGRESSION

Ex 3 -
Classification problem

```
from sklearn.linear_model import LogisticRegression
```

- Similar to linear regression, but for classification
  - We have a set of data (x_1, x_2, …x_p) and corresponding output y
  - We assume the data follows a model of the following form (linear)

$$y = a_0 + a_1x_1 + a_2x_2 + \ldots + a_px_p + \varepsilon$$

- We compute a_i coefficients that minimize misclassifications instead of distance.
- Two steps :
  1. evaluates the probability of being in a given classification
  2. a threshold permits acceptance or rejection of the classification

37

Bruno Agard

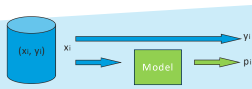
METHODS FOR EVALUATING THE PERFORMANCE OF PREDICTIVE MODELS

- *learning-validation-test data*
- *cross-validation*
- *ROC curve*

38

Bruno Agard

VALIDATION OF ONE PREDICTION



$p_i \approx y_i$?

1. p_i and y_i are numerical values (1, 2, 3, 4.76, -1 120 000)

$$e_i = p_i - y_i$$

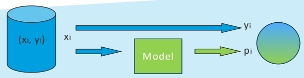
2. p_i and y_i are symbolic values (yes/no, red/green/blue)

$$\begin{cases} p_i = y_i \Rightarrow e_i == 0 \\ p_i <> y_i \Rightarrow e_i == 1 \end{cases}$$

39

Bruno Agard

VALIDATION OF A MODEL



We need to evaluate not on one item but on on the dataset

1. r_i and y_i are numerical values (1, 2, 3, 4.76, -1 120 000)

   $e_i = p_i - y_i$, becomes … see next slide

2. r_i and y_i are symbolic values (yes/no, red/green/blue)

   $\begin{cases} p_i = y_i \Rightarrow e_i == 0 \\ p_i <> y_i \Rightarrow e_i == 1 \end{cases}$, becomes … see the following ones ☺

40

40

Bruno Agard

SOME PERFORMANCE INDICATORS (FOR CONTINUOUS OUTPUTS)

- Mean Error (easy but + vs -)

$$ME = \frac{1}{n}\sum_{i=1}^{n}(p_i - y_i) = \frac{1}{n}\sum_{i=1}^{n} e_i$$

- Mean Square Error (solves + vs – but difficult to explain)

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(p_i - y_i)^2 = \frac{1}{n}\sum_{i=1}^{n} {e_i}^2$$

- Root Mean Square Error (solves + vs –, easier to explain, but sensitive to outliers)

$$RMSE = \frac{1}{n}\sqrt{\sum_{i=1}^{n}(p_i - y_i)^2} = \frac{1}{n}\sqrt{\sum_{i=1}^{n} {e_i}^2}$$

- Mean Absolute Error (solves + vs –, easier to explain, robust to outliers)

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|p_i - y_i| = \frac{1}{n}\sum_{i=1}^{n}|e_i|$$

- Mean absolute percentage error (solves + vs –, easier to explain, robust to outliers)

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\frac{|p_i - y_i|}{r_i} = \frac{1}{n}\sum_{i=1}^{n}\frac{|e_i|}{r_i}$$

Ex 4 -
```
from sklearn.metrics import mean_squared_error
from sklearn.metrics import mean_absolute_error

y_true = [3, -0.5, 2, 7]
y_pred = [2.5, 0.0, 2, 8]

mean_squared_error(y_true, y_pred)
mean_absolute_error(y_true, y_pred)
```

With :
- *r_i real value for i*
- *y_i predicted value for i*
- *e_i prediction error for i*
- *n number of predictions*

Those are just examples, the good metric depends on the problem to solve. Instead on mean errors we may have :

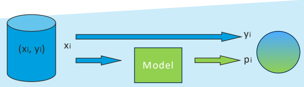
- Cumulative errors
- Mean or cumulated weighted errors
- Mean or cumulated thresholds errors
- …

41

41

Bruno Agard

SOME PERFORMANCE INDICATORS (FOR TWO DISCRETE OUTPUTS)



Confusion matrix

		p_i	
		true	false
r_i	true	TP	FN
	false	FP	TN

Accuracy= (TP+TN)/(TP+FN+FP+TN)
Error Rate = (FP+FN)/(TP+FN+FP+TN)=1-Accuracy

True Positive Rate (=sensitivity=recall) = TP/(TP+FN)
False Positive Rate = FP/(FP+TN)

Precision = TP/(TP+FP)

42

42

Bruno Agard

SOME PERFORMANCE INDICATORS (FOR TWO DISCRETE OUTPUTS)

Confusion matrix

y_i \ p_i	true	false
true	TP	FN
false	FP	TN

Accuracy= (TP+TN)/(TP+FN+FP+TN)

Error Rate = (FP+FN)/(TP+FN+FP+TN)=1-Accuracy

True Positive Rate (=sensitivity=recall) = TP/(TP+FN)

False Positive Rate = FP/(FP+TN)

Precision = TP/(TP+FP)

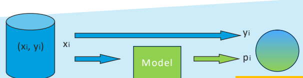
Ex 5 –

```
y_true  = [0, 1, 1, 1, 1, 1, 0, 0]
y_pred = [0, 0, 1, 1, 1, 1, 1, 1]

from sklearn.metrics import confusion_matrix
confusion_matrix(y_true, y_pred)

from sklearn.metrics import accuracy_score
accuracy_score(y_true, y_pred)
5/8

from sklearn.metrics import precision_score
precision_score(y_true, y_pred, average=None)
1/2
4/6

from sklearn.metrics import recall_score
recall_score(y_true, y_pred, average=None)
1/3
4/5
```


43

Bruno Agard

SOME PERFORMANCE INDICATORS (FOR MORE THAN TWO DISCRETE OUTPUTS)



y_i \ p_i	p	o	x	Sum
p	2	0	0	2
o	2	0	0	2
x	1	0	0	1
Sum	5	0	0	5

Accuracy= ( )/( )

Error Rate = (+)/( )

True Positive Rate, for p = $\square$ /

Precision, for p = $\square$ /

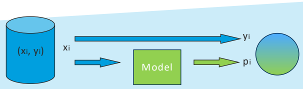
Ex 6 -

```
y_true = [1, 2, 3, 1, 2, 2, 1, 2, 3]
y_pred = [1, 1, 3, 1, 2, 3, 1, 2, 3]

from sklearn.metrics import confusion_matrix
confusion_matrix(y_true, y_pred)

from sklearn.metrics import accuracy_score
accuracy_score(y_true, y_pred)

from sklearn.metrics import precision_score
precision_score(y_true, y_pred, average=None)

from sklearn.metrics import recall_score
recall_score(y_true, y_pred, average=None)
```


44

Bruno Agard

ROC CURVE, AUC



y_i \ p_i	true	false
true	TP	FN
false	FP	TN

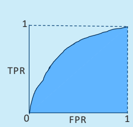
True Positive Rate = TP/(TP+FN)
False Positive Rate = FP/(FP+TN)

When evaluating if an object x_i should be classified true or false, statistical classifiers evaluate a « probability » s to be true and compare it to a threshold (S) :

- If s >= S, the classification is « valid » and returns true
- If s < S, the classification is « unvalid » and returns false

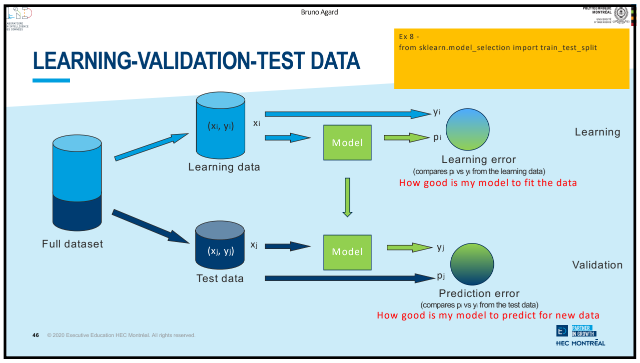
varying s from 0 to 1 gives ROC (receiver operating characteristic) curve

AUC=area under the curve

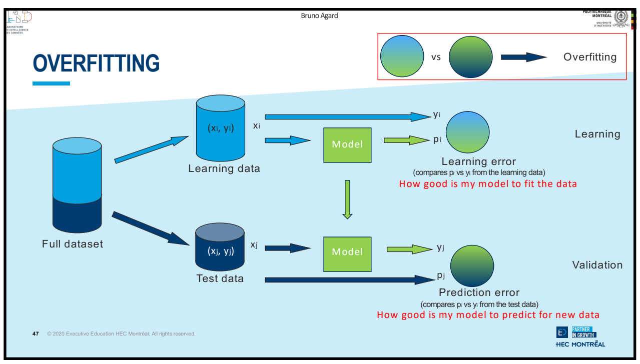


Ex 7 -
Learning performance
ROC and AUC

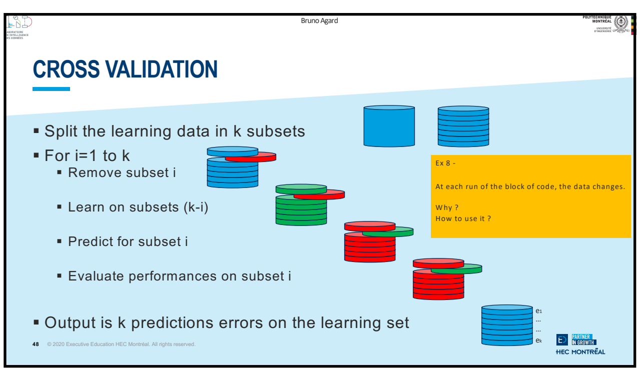
45


Bruno Agard
LEARNING-VALIDATION-TEST DATA
Ex 8 -
from sklearn.model_selection import train_test_split
Full dataset
(xi, yi)
Learning data
xi
Model
yi
pi
Learning
Learning error
(compares pi vs yi from the learning data)
How good is my model to fit the data
(xj, yj)
Test data
xj
Model
yj
pj
Validation
Prediction error
(compares pj vs yj from the test data)
How good is my model to predict for new data
46
© 2020 Executive Education HEC Montréal. All rights reserved.
HEC MONTRÉAL

46


Bruno Agard
OVERFITTING
vs
Overfitting
Full dataset
(xi, yi)
Learning data
xi
Model
yi
pi
Learning
Learning error
(compares pi vs yi from the learning data)
How good is my model to fit the data
(xj, yj)
Test data
xj
Model
yj
pj
Validation
Prediction error
(compares pj vs yj from the test data)
How good is my model to predict for new data
47
© 2020 Executive Education HEC Montréal. All rights reserved.
HEC MONTRÉAL

47


Bruno Agard
CROSS VALIDATION
▪ Split the learning data in k subsets
▪ For i=1 to k
▪ Remove subset i
▪ Learn on subsets (k-i)
▪ Predict for subset i
▪ Evaluate performances on subset i
▪ Output is k predictions errors on the learning set
Ex 8 -
At each run of the block of code, the data changes.
Why ?
How to use it ?
48
© 2020 Executive Education HEC Montréal. All rights reserved.
HEC MONTRÉAL

48

Bruno Agard

CROSS VALIDATION

Ex 9 - Logistic regression

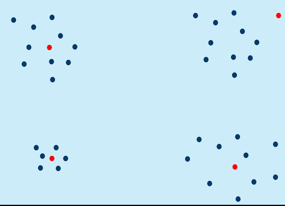
Ex 10 - Linear regressions

- Simply compute
  - Mean error

$$E = \frac{1}{k}\sum_{i=1}^{k} e_i$$

  - Error dispersion

$$\sigma^2 = \frac{1}{k}\sum_{i=1}^{k} (e_i - E)^2$$



49

Bruno Agard

BOOTSTRAP

- Similar to cross validation, but
  - Subsets are built by a random selection of items (x_i, y_i) with replacement
    - A same item (x_i, y_i) may appear more than once on the k subsets
    - Another item (x_i, y_i) may never appear in any subset

50

Bruno Agard

TREES

- *Regression and classification trees*
- *Random Forest*
- *Gradiant Boosting*

51

Bruno Agard

IN BRIEF…

	X1	X2	X3	Y1
C1	1	0	1	2
C2	1	1	0	3
C3	1	0	1	2
C4	1	0	1	..
C5	1	0	..	2
C6	..	1	0	3
C7	1	1	0	4

Forecasting

	X1	X2	X3	Y1
C1	1	0	1	2
C2	1	1	0	3
C3	1	0	1	2
C4	1	0	1	2
C5	1	0	..	2
C6	..	1	0	3
C7	1	1	0	4

Clustering

	X1	X2	X3	Y1
C1	1	0	1	2
C2	1	1	0	3
C3	1	0	1	2
C4	1	0	1	..
C5	1	0	..	2
C6	..	1	0	3
C7	1	1	0	4

Description

If X1=1 then X3 =1 [5, 3/5]
If (X1=1 and X2=0) then X3=1 [3, 3/3]
If X1=1 then !X1=0 [5, 100%]
…

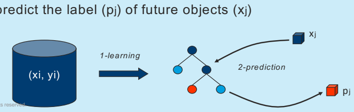
52

52

Bruno Agard

DECISION TREES

- A decision tree allows to classify records (x_i, y_i), by hierarchical division of the whole dataset into subclasses focusing on grouping together items with similar labels (y_i).
- The tree
  - is built from a set of learning objects (x_i, y_i) for which we already know the labels (y_i),
  - will be used to predict the label (p_j) of future objects (x_j)



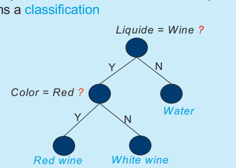
53

53

Bruno Agard

CHARACTERISTICS OF A DECISION TREE

- A decision tree is composed of nodes, branches and leaves :
  - Each node tests an attribute
  - Each branch corresponds to the value of an attribute (in response to the test)
  - Each leaft performs a classification

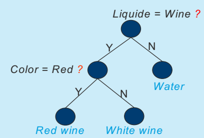


54

54

DECISION TREE

- A decision tree can naturally be translated into deduction rules.



- Rules:
  - If Liquide ≠ Wine then Water
  - If (Liquide = Wine and Color = Red) then Red wine
  - If (Liquide = Wine and Color ≠ Red) then White wine

55

BUILDING A DECISION TREE

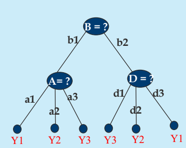
56

TREE PARAMETERS

	A	B	...	K	Y
C1	a1	b2	...	k4	Y1
C2	a3	b1	...	k8	Y2
...	...	...	...	...	...
Ci	a2	b2	...	k3	Y1
...	...	...	...	...	...
Cn	a3	b1	...	k1	Y3

- Objectives:
  - Classify objects in homogeneous classes
  - Cover all the data

- Questions:
  - How to chose the attributes (A, B, …K) ?
  - How to isolate discriminant values (a1, a2 …) ?



57

Bruno Agard

CONSTRUCTION PROCEDURE

- The tree starts at a node P representing all the data
- Partition(P)
  - If all objects are homogeneous, then the node becomes a leaf, labeled with the name of the class.
  - Otherwise, partition the data according to the most discriminating attribute
    - for each attribute A
      - evaluate the quality of partitioning according to A
    - use the best partitioning to divide P into P1, P2, ...Pn
    - for i = 1 to n do Partition(Pi);

58

Bruno Agard

OBSERVATIONS

- The process is recursive
- Recursivity stops when:
  - The objects assigned to a class are homogeneous
  - There are no more attributes for dividing
  - There is no object with the attribute value
- Problem:
  - how to define the "best" partitioning

59

Bruno Agard

EVALUATE THE QUALITY OF A PARTITIONING

60

Bruno Agard

SELECTION OF THE PARTITIONNING ATTRIBUTE

- When partitionning, is it better to select A, B, …K ?
- For each attribute (A, B, …K)
  - Partition according the selected attribute,
  - Evaluate the partition
- Select the best attribute

	A	B	…	K	Y
C1	a1	b2	…	k4	Y1
C2	a3	b1	…	k8	Y2
…	…	…	…	…	…
Ci	a2	b2	…	k3	Y1
…	…	…	…	…	…
Cn	a3	b1	…	k1	Y3

61

61

Bruno Agard

EVALUATE A PARTITIONNING

- The quality of a partitioning differs according to the algorithm :
- The goal is to gain a maximum of information at each partition.
  - Minimizes « disorder » in each class
- Different measures of disorder:
  - Continuous output: based on dispersion (statistical variance)
    Regression tree
  - Discrete output: based on misclassification (Impurity index, Gini index, Entropy,..)
    Classification tree

62

62

Bruno Agard

CONTINUOUS OUTPUT

- Compute variance within each partition P_J

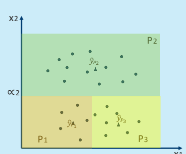
$$V_{P_j} = \frac{1}{|P_J|}\sum_{i \in P_j}\left(y_i - \hat{y}_{P_j}\right)^2$$

- add the variances, weighted by the size of the groups

$$D = |P_J|\sum_{j=i}^{J} V_{P_J}$$

- Or directly

$$D = \sum_{j=i}^{J}\sum_{i \in P_j}\left(y_i - \hat{y}_{P_j}\right)^2$$



- A good score is when all y_i within P_j are close to $\hat{y}_{P_j}$, then D is close to 0

63

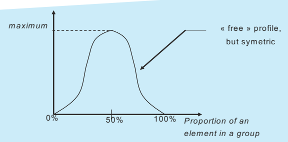
63

DISCRETE OUTPUTS

- The purpose of the indicator is to give a note to a partition
- A partition receives a:
  - good score if all elements in the partition are homogeneous (100% yes or 100% no)
  - bad score if all the elements are mixed (50% yes and 50% no)
- Indicators are based on the proportion of individuals of each type in each class.
  - So, in fact, we measure the degree of mixing

64

DEGREE OF MIXING



- Gini index
- Entropy
- Others…

- Criterion: For all attributes (A, B …K), the division of node j is performed using the variable that ensures the maximum reduction of misclassifications

65

GINI INDEX

- Gini index of a partition p
  - gini(p) = $1 - \sum_i p_i^2$
  - p_i = relative frequency of class i in partition p (% of i in p)
  - Criterion : minimizing gini(p)

ENTROPY

- Entropie of a partition p
  - E(p) = $-\sum p_i \log(p_i)$
  - p_i = relative frequency of class i in partition p
    - Gain = E(before division) - $\sum_j \propto_j$ * E(each resulting partition)
    - $\propto_j$ is the proportion of individuals in each son j
  - Criterion: maximizing gain of entropy

66

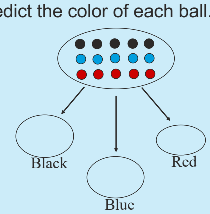
SIMPLE EXAMPLE

Bruno Agard

EXAMPLE: OBJECTIVE

- The goal is to classify balls according to attributes X[0], X[1], X[2], so as to be able to predict the color of each ball.

X[0]	X[1]	X[2]	y
1	2	2	Black
2	1	2	Blue
1	1	1	Red
1	2	2	Red
1	2	2	Red
1	1	2	Black
2	1	2	Blue
1	2	2	Black
2	2	2	Blue
2	2	1	Red
1	1	2	Black
2	2	2	Blue
2	1	2	Blue
2	2	2	Black
1	2	1	Red

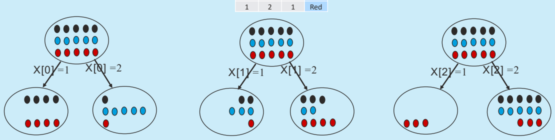


Proportion	C1
Black	0,33
Blue	0,33
Red	0,33
Gini	0,66666667

Bruno Agard

ALTERNATIVES

X[0]	X[1]	X[2]	y
1	2	2	Black
2	1	2	Blue
1	1	1	Red
1	2	2	Red
1	2	2	Red
1	1	2	Black
2	1	2	Blue
1	2	2	Black
2	2	2	Blue
2	2	1	Red
1	1	2	Black
2	2	2	Blue
2	1	2	Blue
2	2	2	Black
1	2	1	Red



Bruno Agard

EXAMPLE : ATTRIBUTE X[0]

Ex 20 –

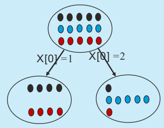
X=X0
max_depth=1

- Gini index of a segment s :
  - i(s) = $1 - \sum_i p_i^2$
  - p_i is the proportion of individuals of class i in s.
- Entropy of a segment s :
  - E(s) = $-\sum_i p_i \log(p_i)$

$1 - \sum_i p_i^2$

Proportion	C1	C2	Sigma
Black	0,50	0,14	
Blue	0,00	0,71	
Red	0,50	0,14	
Gini	0,5	0,44897959	0,476

$\sum_j P_j * i(N_j)$



70

HEC MONTRÉAL

70

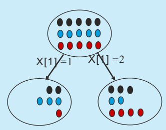
Bruno Agard

EXAMPLE : ATTRIBUTE X[1]

Ex 20 –

X=X1
max_depth=1

Proportion	C1	C2	Sigma
Black	0,33	0,33	
Blue	0,50	0,22	
Red	0,17	0,44	
Gini	0,61111111	0,64197531	0,589



71

HEC MONTRÉAL

71

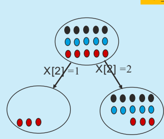
Bruno Agard

EXAMPLE : ATTRIBUTE X[2]

Ex 20 –

X=X2
max_depth=1

Proportion	C1	C2	Sigma
Black	0,00	0,42	
Blue	0,00	0,42	
Red	1,00	0,17	
Gini	0	0,625	0,500



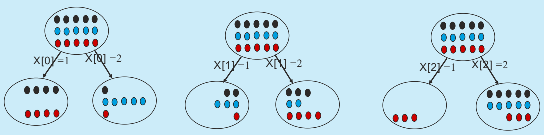
72

HEC MONTRÉAL

72

Bruno Agard

CHOICE



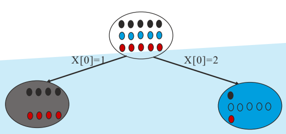
Criterion	X[0]	X[1]	X[2]
Gini	0,476	0,589	0,5

73 © 2020 Executive Education HEC Montréal. All rights reserved.

73

Bruno Agard

STEP 1



Accuracy : 00 %		Predictions		
		Black	Blue	Red
Real	Black	4	1	0
	Blue	0	5	0
	Red	4	1	0

Ex 20 –

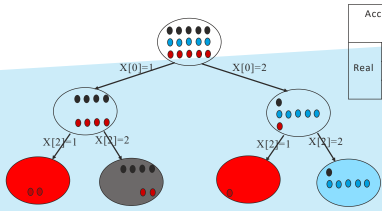
X=X0+X1+X2
max_depth=1

74 © 2020 Executive Education HEC Montréal. All rights reserved.

74

Bruno Agard

STEP 2



Accuracy : 80 %		Predictions		
		Black	Blue	Red
Real	Black	4	1	0
	Blue	0	5	0
	Red	2	0	3

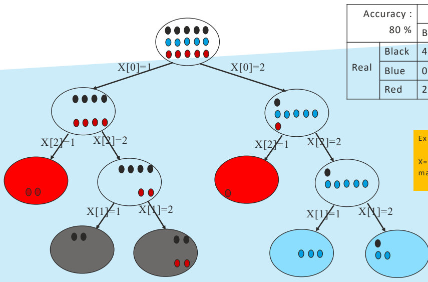
Ex 20 –

X=X0+X1+X2
max_depth=2

75 © 2020 Executive Education HEC Montréal. All rights reserved.

75

STEP 3



Accuracy : 80 %		Predictions		
		Black	Blue	Red
Real	Black	4	1	0
	Blue	0	5	0
	Red	2	0	3

Ex 20 –

X=X0+X1+X2
max_depth=3

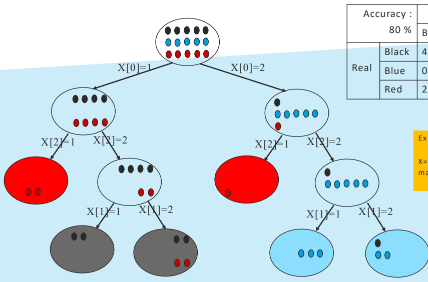
76

PRUNING

- Trees that are too bushy are useless
  - Interest of pruning to simplify the tree
- Possibility to apply it on the whole data set or on a subset reserved for validation
- Can be applied
  - during creation of the tree
  - After creation of the tree
- Examples of pruning criteria:
  - partitioning to be deleted if :
    - Gain < threshold

77

STEP 3



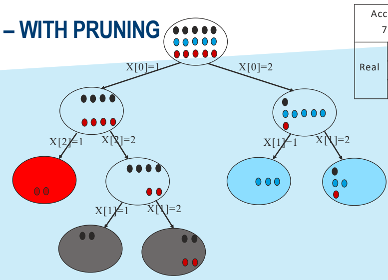
Accuracy : 80 %		Predictions		
		Black	Blue	Red
Real	Black	4	1	0
	Blue	0	5	0
	Red	2	0	3

Ex 20 –

X=X0+X1+X2
max_depth=3

78

STEP 3 – WITH PRUNING



Accuracy : 73.33 %		Predictions		
		Black	Blue	Red
Real	Black	4	1	0
	Blue	0	5	0
	Red	2	1	2

Ex 20 –

X=X0+X1+X2
max_depth=3
min_samples_leaf=2

79

« DIFFERENT » ALGORITHMS

Name	Criterion	Features	Missing values	Pruning	
ID3	Information gain ($\Delta Entropy$)	Discrete	no	no	Classification Binary trees (yes/no)
CART	Gini	Discrete Continuous	yes	yes	Classification and regressions Binary trees (yes/no)
C4.5	Gain ratio ($\frac{\Delta Entropy}{Entropy}$)	Discrete Continuous	yes	yes	Classification and regressions Full trees (yes/no)
C5.5	Gain ratio ($\frac{\Delta Entropy}{Entropy}$)	Discrete Continuous	yes	yes	Applies a boosting method on C4.5

80

LARGE DATABASES

- Previous algorithms assume that the data is stored in memory.
- Easily parallelizable methods
  - SPRINT (VLDB96 -- J. Shafer et al.'96)
  - Scalable PaRallelizable INndution of decision Tree
  - Does not require a resident structure in memory
  - Parallel scaled version

81

Bruno Agard

DATA STRUCTURE (ATTRIBUTE LISTS)

Age	Car Type	Risk
23	family	High
17	sports	High
43	sports	High
68	family	Low
32	truck	Low
20	family	High

Age	Class	rid
17	High	1
20	High	5
23	High	0
32	Low	4
43	High	2
68	Low	3

Car Type	Class	rid
family	High	0
sports	High	1
sports	High	2
family	Low	3
truck	Low	4
family	High	5

Figure 3: Example of attribute lists

HEC MONTRÉAL

82

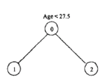
Bruno Agard

EVOLUTION DES LISTES

Attribute lists for node 0

Age	Class	Tid
17	High	1
20	High	5
23	High	0
32	Low	4
43	High	2
68	Low	3

Car Type	Class	Tid
family	High	0
sports	High	1
sports	High	2
family	Low	3
truck	Low	4
family	High	5



Attribute lists for node 1

Age	Class	Tid
17	High	1
20	High	5
23	High	0

Car Type	Class	Tid
family	High	0
sports	High	1
family	High	5

Attribute lists for node 2

Age	Class	Tid
32	Low	4
43	High	2
68	Low	3

Car Type	Class	Tid
sports	High	2
family	Low	3
truck	Low	4

Figure 4: Splitting a node's attribute lists

HEC MONTRÉAL

83

Bruno Agard

Ex 21 -
Classification tree

See influence of parameters

Learning error / prediction error
(misclassifications)

Ex 22 -
See the differences with classification trees

See influence of parameters

Learning error / prediction error
(distances)

84

Bruno Agard

RANDOM FOREST

85

Bruno Agard

RANDOM FOREST

To make a random forest of n trees from base B :

1. Split base B in n subsets
   - Sample n observations (lines) with draw and put back
   - Sample p variables (columns) with and put back (p is about √(number of variables))
2. On each subset, a decision tree is trained
3. You get n trees... that you keep.
4. The prediction of the random forest is the result from simple majority vote from all n trees.

- Advantage: parallel computation, less sensitive to unbalancing, smaller trees, easy to implement, improves the performance of the chosen tree technique, excellent for VERY large problems (compared to other methods)
- Disadvantage: you lose the visual aspect of unique decision trees.

	X1	X2	X3	Y1
C1	1	0	1	2
C2	1	1	0	3
C3	1	0	1	2
C4	1	0	1	3
C5	1	0	..	2
C6	..	1	0	3
C7	1	1	0	4

Ex 23 -

86

Bruno Agard

GRADIENT BOOSTING

87

Bruno Agard

GRADIENT BOOSTING

- Boosting methods are similar to random forest in the way it uses multiple models instead of a unique one.

- All n items (i) receive the same weight (1/n)
- for j=1 to J,
  - A model (j) is learn
  - Each item (i) is tested, and a weighted error rate is computed for (i)
  - Update the weight of each item (i) according to a gradient of the error

- Gives more weight to bad predicted items so as to better consider them in the next model

Ex 24 – Classifier

Ex 25 – Regressor

88

Bruno Agard

NEURAL NETWORK

- *Introduction to multilayer perceptron neural networks*

89

Bruno Agard

IN BRIEF…

	X1	X2	X3	Y1
C1	1	0	1	2
C2	1	1	0	3
C3	1	0	1	2
C4	1	0	1	..
C5	1	0	..	2
C6	..	1	0	3
C7	1	1	0	4

Forecasting

	X1	X2	X3	Y1
C1	1	0	1	2
C2	1	1	0	3
C3	1	0	1	2
C4	1	0	1	2
C5	1	0	..	2
C6	..	1	0	3
C7	1	1	0	4

Clustering

	X1	X2	X3	Y1
C1	1	0	1	2
C2	1	1	0	3
C3	1	0	1	2
C4	1	0	1	..
C5	1	0	..	2
C6	..	1	0	3
C7	1	1	0	4

Description

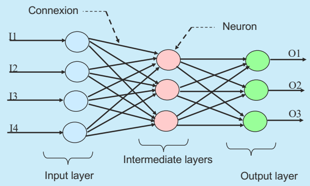
If X1=1 then X3 =1 [5, 3/5]
If (X1=1 and X2=0) then X3=1 [3, 3/3]
If X1=1 then !X1=0 [5, 100%]
…

90

Bruno Agard

INTRODUCTION

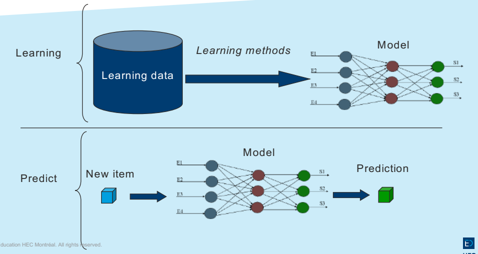
- Neural networks mimic brains, learning process, bla bla bla ☺



91

Bruno Agard

PRINCIPLE



92

Bruno Agard

NEURAL NETWORKS

- Complex networks of elementary, interconnected computing units (neurons).
- A neural network is entirely characterized by
  - its architecture
  - the state transition functions of the neurons.
- Different types of elementary cells
  - Binary or integer values
  - Deterministic or probabilistic output
- Different network architectures
  - With or without feedback (a cell may or may not influence its input)
  - Total or partial feedback
- Different dynamics in the network
  - Synchronous (all cells evolve at the same time)
  - Asynchronous (sequential, random)

93

Bruno Agard

APPLICATIONS

- Learning
  - Supervised
  - Unsupervised
- Learning methods can be:
  - Offline (the entire learning base is processed simultaneously)
  - Online (learning examples are processed one by one when they arrive)

94

Bruno Agard

APPLICATIONS

- Pattern recognition
- Classification
- Prediction
- Vision
- Robotics
- Adaptive control
- For data analysis, the perceptron and the multilayer perceptron are most popular.
  - They are the most "equipped" models
    - Learning Models
    - Learning algorithms
    - Mathematical Evidence

95

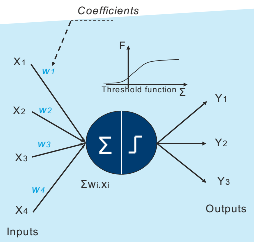
Bruno Agard

PERCEPTRON

96

Bruno Agard

DESIGN



97

Bruno Agard

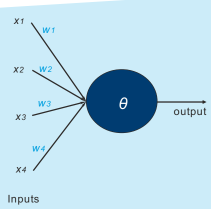
PERCEPTRON

- The perceptron is a neuron model with a learning algorithm created by Frank Rosenblatt in 1958.
- We will see here the simplified version developed by F. Denis and R. Gilleron.

98

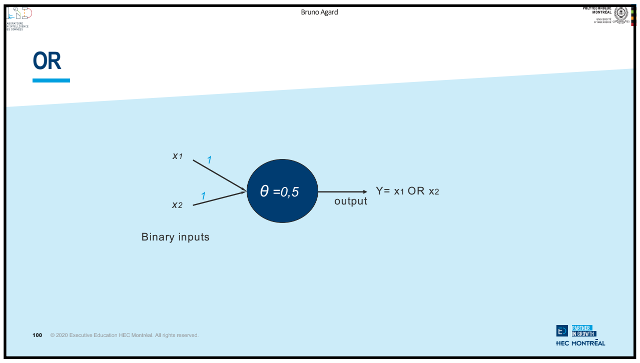
Bruno Agard

DESCRIPTION



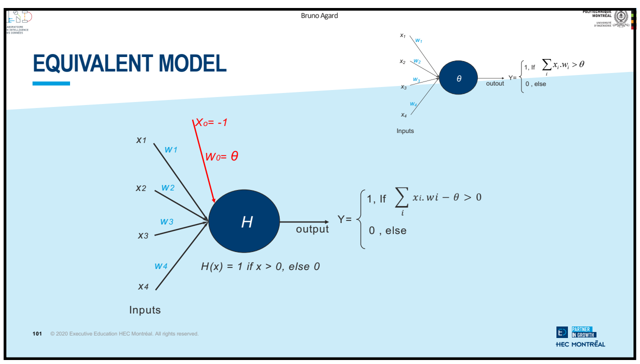
$$Y = \begin{cases} 1, \text{ If } \sum_i x_i . w_i > \theta \\ 0 \text{ , else} \end{cases}$$

99



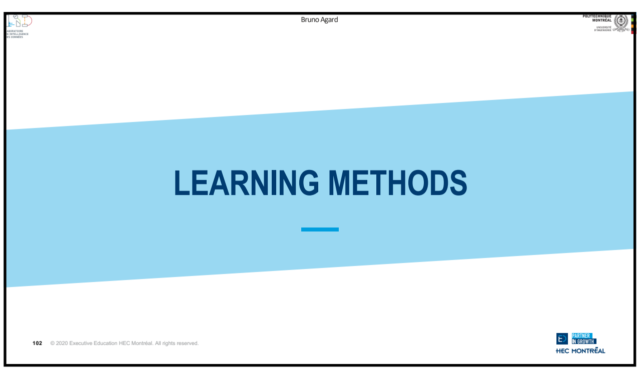
Bruno Agard
OR
x1
1
x2
1
θ =0,5
output
Y= x1 OR x2
Binary inputs
100
© 2020 Executive Education HEC Montréal. All rights reserved.
HEC MONTRÉAL

100



Bruno Agard
EQUIVALENT MODEL
X0= -1
W0= θ
x1
w1
x2
w2
w3
x3
w4
x4
H
output
Y=
1, If $\sum_i x_i . w_i - \theta > 0$
0 , else
H(x) = 1 if x > 0, else 0
Inputs
101
© 2020 Executive Education HEC Montréal. All rights reserved.
HEC MONTRÉAL

101



Bruno Agard
LEARNING METHODS
102
© 2020 Executive Education HEC Montréal. All rights reserved.
HEC MONTRÉAL

102

Bruno Agard

MAIN PRINCIPLE

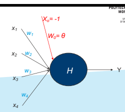
- Discovery of complex models with progressive refinement
- Classic learning process
  - Initialization of weights w_i to random values
  - Repeat, until stop criterion
    - Presentation of an example
    - Signal propagation in the network
    - Error calculation and weights adjustments
- The network adapts during the learning phase
- Several possible algorithms for weights adjustments

103

103

Bruno Agard

ERROR CORRECTION ALGORITHM



- Inputs :
  - a learning dataset D
  - Random initialization of weights w_i for i between 0 and n
- Repeat, until stop criterion
  - Take an example (s, c) in D
  - Calculate the output o of the perceptron for input s
  - For i from 0 to n - - Updating the weights - - -
    - $w_i = w_i + (c-o).x_i$
- Output :
  - A perceptron P defined by (w_0, w_1, ..., w_n)

104

104

Bruno Agard

EXEMPLE: APPRENTISSAGE DU OU

étape	w_0	w_1	w_2	Entrée	$\Sigma_0^2 w_i x_i$	o	c	w_0	w_1	w_2
init								0	1	-1
1	0	1	-1	100	0	0	0	0+0x1	1+0x0	-1+0x0
2	0	1	-1	101	-1	0	1	0+1x1	1+1x 0	-1+1x1
3	1	1	0	110	2	1	1	1	1	0
4	1	1	0	111	2	1	1	1	1	0
5	1	1	0	100	1	1	0	1+(-1)x1	1+(-1)x0	0+(-1)x0
6	0	1	0	101	0	0	1	0+1x1	1+1x0	0+1x1
7	1	1	1	110	2	1	1	1	1	1
8	1	1	1	111	3	1	1	1	1	1
9	1	1	1	100	1	1	0	1+(-1)x1	1+(-1)x0	1 +(-1)x0
10	0	1	1	101	1	1	1	0	1	1

105

105

Bruno Agard

LIMITES

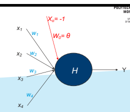
- May not converge
- We don't known if the result will be robust (a new element may differ and reopen gaps)
- No noise tolerance (if an initial, learning information is misfiled, the algorithm will never converge)

106

106

Bruno Agard

WIDROW-HOFF ALGORITHM



- Inputs :
  - a dataset D
  - Step value $e \in [0 ; 1]$
  - Random initialization of weights w_i for i between 0 and n
- Repeat, until stop criterion
  - Take an example (s, c) in D
  - Calculate the output o of the perceptron for input s
  - For i from 0 to n - - Updating the weights - - -
    - $w_i = w_i + e.(c-o).x_i$
- Output :
  - A perceptron P defined by $(w_0, w_1, \ldots, w_n)$

Ex 30 -

107

107

Bruno Agard

GRADIENT DESCENT ALGORITHM

- Rather than considering a perceptron that sould correctly classifies each sample one by one, we will consider the error globally on a subset of data.

108

108

Bruno Agard

GRADIENT DESCENT ALGORITHM

- Inputs :
  - a dataset D, split in subsets S_j
  - Step value $e \in [0 ; 1]$
  - Random initialization of weights w_i for i between 0 and n
- Repeat, until stop criterion
  - For all i, $\Delta w_i = 0$
  - For all samples (s, c) in S
    - Calculate the output o of the perceptron for input s
    - Pour tout i, $\Delta w_i = \Delta w_i + e.(c - o)\, x_i$
  - For i de 1 to n - - Updating the weights - -
    - $w_i = w_i + \Delta w_i$
- Output :
  - Un perceptron P défini par (w_1, ..., w_n)

109

HEC MONTRÉAL

109

Bruno Agard

MULTI-LAYER PERCEPTRON

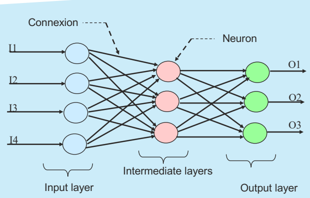
110

HEC MONTRÉAL

110

Bruno Agard

MULTI-LAYER PERCEPTRON



111

HEC MONTRÉAL

111

Bruno Agard

WHY MULTI-LAYERS ?

- Only one neuron:
  - Limited possibilities, only one calculation, only one operator.
- Several interconnected neurons:
  - More flexibility
  - More possibilities
  - More modeling power

112

112

Bruno Agard

TOPOLOGY

- Choice of the number of layers
  - inputs, 1 or more hidden layers, outputs
- Choice of the number of neurons per layer
  - depends on inputs and outputs
  - intermediate hidden layers
- Input variables
  - Discrete values
  - Reduced and centered continuous variable [-1,+1].
- Outputs
  - Continuous (estimations)
  - Discrete (classifications)

113

113

Bruno Agard

DESIGN OF AN MULTI-LAYER PERCEPTRON

- To be able to use multi-layer networks in learning, two things are essential:
  - a method indicating how to choose a network architecture to solve a given problem.
    - how many hidden layers?
    - how many neurons per hidden layer?
- once the architecture is chosen, a learning algorithm that calculates, from the learning samples, the values of the coefficients w_{ij} to build a network adapted to the problem.

114

114

Bruno Agard

HOW

- For the architecture: auto-constructive algorithms. Research is still very active in this domain. Their role is twofold:
  - learning the sample with a current network,
  - modification of the current network, by adding/deleting new neurons or a new layer, in case of learning failure.
- For learning:
  - Corrections methods.

115

Bruno Agard

EXTENTION OF WIDROW-HOFF ALGORITHM FOR MULTI-LAYERS PERCEPTRONS (SIMPLIFIED)

- Inputs :
  - a dataset D
  - Step value e $\in [0 ; 1]$
  - Random initialization of weights w_i for i between 0 and n
- Repeat, until stop criterion
  - Take an example (s, c) in D
  - Calculate the output o of the perceptron for input s
    - Error on the output d=(c-o)
  - For each layer (from the last, to the first)
    - For neuron i in the layer

$$d_i = o_i(1-o_i) \sum_{k \in succ(i)} d_k w_{ki}$$

  - For all i, j, - - Updating the weights - - -
    - $w_{ij} = w_{ij} + e.d_i.x_{ij}$
- Output :
  - A multi-layers perceptron defined by initial structure and all w_{ij}

Ex 31 -

116

Bruno Agard

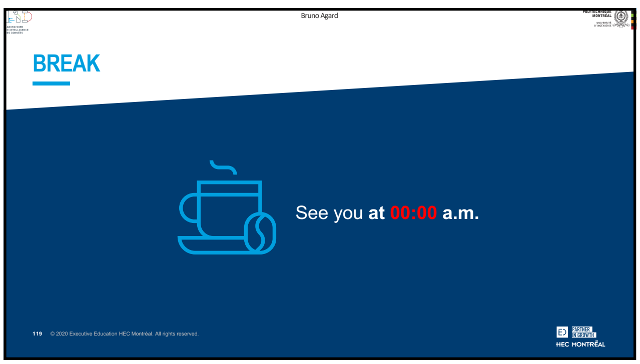
SATISFACTION SURVEY

Please take five minutes to complete the **satisfaction survey**.

117


Bruno Agard
THANK YOU!
PARTNER
IN GROWTH
HEC MONTRĒAL

118


Bruno Agard
BREAK
See you at 00:00 a.m.
119
© 2020 Executive Education HEC Montréal. All rights reserved.
HEC MONTRĒAL

119