# 1 Subject

# Comparing the quickness and memory occupation of various free implementation of the C4.5 (or assimilated) decision tree algorithm on large dataset.

Dealing with large dataset is on of the most important challenge of the Data Mining. In this context, it is interesting to analyze and to compare the performances of various free implementations of the learning methods, especially the computation time and the memory occupation. Most of the programs download all the dataset into memory. The main bottleneck is the available memory.

In this tutorial, we compare the performance of several implementations of the algorithm C4.5 (Quinlan, 1993) when processing a file containing 500,000 observations and 22 variables.

Our main criteria are memory occupation and computation time. So that everyone can replicate the experience and compare the results, here are the characteristics of our machine: a Pentium 4 3.8 GHz with 2 GB of RAM. We have measured the performances of our computer using the Lite version of the SISOFTWARE SANDRA (<u>http://www.sisoftware.net/</u>). We obtain:



Software	Version	URL
KNIME	1.3.5	http://www.knime.org/index.html
ORANGE	1.0b2	http://www.ailab.si/orange/
R (package rpart)	2.6.0	http://www.r-project.org/
RAPIDMINER (YALE)	Community Edition	http://rapid-i.com/
SIPINA	Research	http://eric.univ-lyon2.fr/~ricco/sipina.html
TANAGRA	1.4.27	http://eric.univ-lyon2.fr/~ricco/tanagra/
WEKA	3.5.6	http://www.cs.waikato.ac.nz/ml/weka/

We compare the performances of the following programs in this tutorial.

# 2 The WAVE dataset

Our data file is WAVE500K.ZIP (<u>http://eric.univ-lyon2.fr/~ricco/tanagra/fichiers/wave500k.zip</u>). It is the well-known artificial dataset described in the CART book (Breiman et al., 1984). We have generated a dataset with 500.000 observations. The class attribute has 3 values, there are 21 continuous predictors.

We deal primarily with ARFF WEKA file format. The majority of data mining software can handle this format. Otherwise, text file format with tabulation separator will be used.

# 3 Comparison of performances

📕 Gestionnaire des t	âches de Windows	
Fichier Options Affichag	je Arrêter ?	
Applications Processus	Performances Mise en réseau Utilis	ateurs
Nom de l'image	Nom de l'utilisateur Pr	Util. mémoire 📃 🔼
Tanagra.exe	Maison (	00 45 456 Ko
avgrsx.exe	SYSTEM (	)0 43 368 Ko 🚬 🔤
WINWORD.EXE	Maison (	)0 43208Ko
explorer.exe	Maison (	)1 34 456 Ko
AcroRd32.exe	Maison (	)0 29 408 Ko
svchost.exe	SYSTEM (	)0 24 664 Ko 🔳
RTHDCPL.EXE	Maison (	)0 21208Ko
svchost.exe	SERVICE LOCAL (	)0 7412Ko
spoolsv.exe	SYSTEM (	)0 5444 Ko
VM_STI.EXE	Maison (	)0 5248Ko
rapimgr.exe	Maison (	)0 5176Ko 📃
svchost.exe	SYSTEM (	)0 4 828 Ko
wcescomm.exe	Maison <sub>(</sub> 0	00 4 816 Ko
svchost.exe	SERVICE RESEAU (	00 4 416 Ko
svchost.exe	SYSTEM (	00 4 208 Ko
csrss.exe	SYSTEM (	)1 4 196 Ko
CAPPSWK.EXE	SYSTEM (	00 3 852 Ko
CAPPSWK.EXE	SYSTEM (	)0 3 744 Ko
svchost.exe	SERVICE RÉSEAU (	)0 3 632 Ko
alg.exe	SERVICE LOCAL (	00 3 552 Ko
svchost.exe	SYSTEM (	)0 3432Ko 🤜
Afficher les process	us de tous les utilisateurs	Terminer le processus
Processus : 39 UC utili:	iée : 2% Charge dédiée : 388 M	10/

Figure 1 – Measuring the memory occupation under Windows OS

We utilize the indications of Windows OS about the memory occupation (Figure 1). We measure the memory occupation when the program is launched, when the dataset is loaded, during the learning phase, after the learning phase. The maximum occupation during the learning phase is the most important indicator. This is the main bottleneck of the whole process.

About the computation time, some programs provide it. For the other cases, we use a chronograph. Indeed, we try mainly to obtain an approximated value of the computation time. It is not necessary to measure a very precise value.

#### 3.1 KNIME

KNIME (Konstanz Information Miner -- <u>http://www.knime.org/</u>) describes the treatments as a succession of operations, represented by a workflow.

**KNIME and the JRE**. When we launch the software, the JRE (Java Runtime Environment) is started and the global memory occupation is 92.6 MB.



We create a new project (FILE / NEW). We add the ARFF READER component into the workflow and we click on the CONFIGURE menu in order to select the data file. After the loading, the memory occupation becomes 160.4 MB.

We insert the decision tree LEARNER in the workflow, we see on the panel at the right a description of the method and its parameters. We connect the component ARFF READER, then we click on the menu CONFIGURE in order to define the parameters.

It is not really the C4.5 method, but it is quite similar. We set the minimum number of examples per node at 1000. We set the number of thread used to 1 in order to make the results comparable. We note that Knime is the only one software that can use several threads during the induction of the decision tree.



We launch the calculation by clicking on the EXECUTE AND OPEN VIEW menu. Knime does not give indications about the tree structure (number of nodes and leaves). The processing time is 270 seconds; the maximum memory occupation is 245.8 MB during the learning phase.



It is possible to save the workflow. But, it seems that the software tries to store all the available information, including the dataset. The processing time can be very important for our analysis.

## 3.2 ORANGE

ORANGE can be used through standard programming language (python) or visual programming. We use this last framework in this tutorial.

When we launch ORANGE, the **Python Runtime Environment** is started. The global memory occupation is 24.9 MB. We have a familiar interface. The treatments are defined in a schema. The components are available in the tool bar. We add the FILE (DATA tab) into the schema. We click on the OPEN menu and we select the data file. The loading is automatically started; there is no specific menu for launching the operation.

	🖻 Qt File 📃	
00 Ot Orange Canvas ISchoma 11	Data File	
	Browse documentation data sets 💌 Reload	
	- Data File	
	Symbols for missing values in tab-delimited files (besides defaulted	unes)
	Don't care: Don't know:	
	)pen	? 🛛
	Regarder dans : 📄 big_dataset_decision_tree	- 🖛 🗈 💣 📰 -
	wave500k.arff	
	Mes documents	
Open	Bureau	
File Rename F2		
Remove Del		
4	Mes documents	
	Poste de travail	
	Favoris réseau Nom du fichier : wave500k arff	▼ Duvrir
	Fichiers de tune : Wolk a (* arti)	Annuler

The computation time is 90 seconds. The memory occupation becomes 259.5 MB. We add the SELECT ATTRIBUTE component in order to define the position of the variables.

🥶 Qt Orange Canvas - [Schema 1]		
Eile Options Window Help		_ <u>-</u> -
🛛 🖻 🖓 🕇 🥪	Ct Salaat Attributes	
Data Classify Evaluate Visualize Associate Re	- Available Attributes	- Altributes
		Down C V10
Example:		Class  CLASS  Meta Attributes
File Select Attrib Open		Up
<u>R</u> ename F2 Remove Del		>
		Down
	Apply	Reset

We can insert now the CLASSIFICATION TREE (CLASSIFY tab) component. We configure it **before** the connection. We click on the OPEN MENU. We set the parameters of the algorithm.

🤓 Qt Orange Canvas	
Eile Options Window Help	
	🐯 Qt Classification Tree 📃 🗖 🔀
	Classification Tree
Schema 1	
	Attribute selection criterion
	Gini Index 🔹
	✓ Limit the number of reference examples to 100 ▲
	Number of heighbours in ReliefF 5
Fie Select Attributes Classification	Tree Structure
Remove Del	Pre-Pruning
	Stop splitting nodes with less instances than 5 🚍
	☐ Stop splitting nodes with a majority class of (%) 95 🚔
<	Post-Pruning
	Recursively merge leaves with same majority class
	✓     Pruning with m-estimate, m=     2

We insert the CLASSIFICATION TREE VIEWER (CLASSIFY tab) into the schema. We make the connection between this viewer and the learner. Finally, we make the connection between SELECT ATTRIBUTES and CLASSIFICATION TREE. The computation is automatically launched.



The processing time is 130 seconds; the memory occupation is 795.7 MB. The tree includes 117 leaves.

## 3.3 The R Software with the RPART package

R works with scripts. We use the **system.time(.)** command to obtain the processing time.

```
R RGui - [R Console]
                                                                                                <u>R</u> Fichier Edition Voir Misc Packages Fenêtres Aide
                                                                                                    a ×
🖆 💾 🖪 🔁 🔂 🐵 🎒
R version 2.6.0 (2007-10-03)
Copyright (C) 2007 The R Foundation for Statistical Computing
ISBN 3-900051-07-0
R est un logiciel libre livré sans AUCUNE GARANTIE.
Vous pouvez le redistribuer sous certaines conditions.
Tapez 'license()' ou 'licence()' pour plus de détails.
R est un projet collaboratif avec de nombreux contributeurs.
Tapez 'contributors()' pour plus d'information et
 'citation()' pour la façon de le citer dans les publications.
Tapez 'demo()' pour des démonstrations, 'help()' pour l'aide
en ligne ou 'help.start()' pour obtenir l'aide au format HTML.
Tapez 'q()' pour quitter R.
> #chargement des données
> setwd("D:/DataMining/Databases_for_mining/comparison_TOW/big_dataset_decision_tree")
> system.time(donnees <- read.table(file="wave500k.txt",dec=".",header=T))</pre>
              système
utilisateur
                             écoulé
      24.17
                  0.17
                              24.66
> #construction de l'arbre
> library(rpart)
> parametres <- rpart.control(xval=0,minsplit=1000,minbucket=1000,maxcompete=0,maxsurrogate=0,cp=0)
> system.time(modele <- rpart(CLASS ~., data = donnees, method = "class", control = parametres))</pre>
utilisateur
               système
                             écoulé
     154.62
                   1.67
                              157.09
                                                                                                  >
```

At the beginning, the memory occupation is 18.8 MB. We use the text file format with R, we load the WAVE500K.TXT. The computation is 24 seconds; the memory occupation becomes 184.1MB.

The RPART procedure is rather similar to CART method. But we set the parameters in order to produce a tree similar to the others programs i.e. MINSPLIT = 1000 observations; MINBUCKET = 1000 observations; CP = 0. There is not post-pruning here. But, it does not matter. We know that this part of the processing is fast in the C4.5 approach.

The processing time for the tree construction is 157 seconds. The memory occupation becomes 718.9 MB.

#### 3.4 RAPIDMINER (formerly YALE)

When we launch RAPIDMINER, we create a new project by clicking on FILE / NEW. An empty OPERATOR TREE is available.

We add an ARFFEXAMPLESOURCE component into the tree, in order to load the dataset.

RapidMiner@FUJITSU					
<u>F</u> ile <u>E</u> dit <u>V</u> iew <u>P</u> rocess <u>T</u> ools <u>H</u> elp					
🞦 🎜 📕 🖶 📚 🛛 🛯	'는 🗩 🧯	🕨 🕨 🖌	3	1	1 👮
Coperator Tree		📝 Parameters 🕞 X	ML 📄 Comment	≌ New Operator	
_ Root		logverbosity		init	-
📲 🤷 New Operator	•	Core 🔸			
🥙 New Building Block	•	10 •	Attributes 🕨	l	
Save as Building Block		Learner >	Clustering	-	_
			Examples >	ArffExampleSetWriter	
Rename	F2	Postprocessing +	Models +	ArffExampleSource	
Delete	Supprimer	Preprocessing +	Other 🕨	BibtexExampleSource	
Сору	Ctrl+C	Validation 🕨	Results 🕨	C45ExampleSource	
to Cut	Ctrl+X	Visualization 🕨			
Baste	Ctrl+V	💡 ModelApplier		CSVExampleSetwriter	
Lucio	Out v	🖧 ModelGrouper		CSVExampleSource	
Operator Info	F1	🔊 ModelUngrouper		CachedDatabaseExampleSource	.
📔 Breakpoint Before		ModelUpdater		DBaseExampleSource	
🦲 Breakpoint <u>W</u> ithin		CoperatorChain		📑 DatabaseExampleSetWriter	
属 Bre <u>a</u> kpoint After				📑 DatabaseExampleSource	
Add Breakpoints (Debug Mode)				📑 ExampleSetWriter	57 PM
🔛 📃 🔍 Domovo Brooknointe (Dohus Mada a	en.			European Comments	ST T M

The component appears into the operator tree, we set the right parameters by selecting our data file. We click on the PLAY button of the tool bar. The computation time for the data importation is 7 seconds. The allocated memory increases from 136.6MB to 228.1MB.

RapidMiner@FUJITSU (test.xml)	
<u>F</u> ile <u>E</u> dit <u>V</u> iew <u>P</u> rocess <u>T</u> ools <u>H</u> elp	
🞦 🏳 📗 🖶 🗞 🕜 🔌 🔒	🖛 🧐 🕨 📰 🛷 🚨 🛛 🕎 🖗
Coperator Tree	🕼 Parameters 🕞 XML 📄 Comment 🛸 New Operator
Root     Process     ArrfExampleSource	data_file     D:\DataMining\Databases_for_mining\       label_attribute
ArffExampleSource	Id_attribute
🚭 Open	
big_dataset_decision_tree	
G Sep 20, 2008 6:15:0 B Sep 20, 2008 6:19:0 B Sep 20, 2008 6:19:0	File Name Size Type Last modified
P Sep 20, 2008 6:18:0 Sample Data P Sep 20, 2008 6:18:0 Samples P Sep 20, 2008 6:18:0 Samples P Sep 20, 2008 6:18:0 Workspace P Sep 20, 2008 6:18:0	wave500k.anff 52 MB ARFF Data File Sep 20, 2008
P Sep 20, 2008 6:18:0	
P Sep 20, 2008 6:18:0 P Sep 20, 2008 6:18:0 Wave500k.arff	
P Sep 20, 2008 6:18:0 data_file (*.arff)	
P Sep 20, 2008 6:18:0 P Sep 20, 2008 6:18:0 P Sep 20, 2008 6:18:0 P Sep 20, 2008 6:18:0	Open Cancel
•	6:2 <b>1</b> -21

We must now insert the "decision tree" into the diagram. We operate from the menu NEW OPERATOR on the root of the tree. We use the DECISIONTREE operator. We use parameters similar to the other programs.

*		
🗞 RapidMiner@FUJITSU (test.xml*)		
<u>File Edit V</u> iew <u>P</u> rocess <u>T</u> ools <u>H</u> elp		
🖺 📁 🛃 🖶 🎭 🛷 🔌 🐂 🚒	🍯 🕨 🔳 🎺 🚨	<b>1</b>
- Operator Tree	Parameters 🕞 XML 📄 Comment	🕒 New Operator
E Root	keep_example_set	
- Process	criterion	gain_ratio
	minimal_leaf_size	1000
DecisionTree	minimal_gain	0.0
Ueoision i ree	maximal_depth	-1
	confidence	0.25
	no_pruning	
P Sep 20, 2008 6:24:11 PM: Process initialised P Sep 20, 2008 6:24:11 PM: [NOTE] Process starts P Sep 20, 2008 6:24:11 PM: Process: Root[1] (Process) +- ArfExampleSource[1] (ArfExampleSource) +- DecisionTree[1] (DecisionTree) P Sep 20, 2008 6:29:10 PM: [NOTE] Process finished after 298 seco P Sep 20, 2008 6:29:10 PM: Process:	onds	Max: 1.1 GB Total: 1.1 GB
•		6:32:43 PM

The PLAY button starts again all the computations. We obtain the decision tree after 298 seconds.

Curiously, RAPIDMINER needs much memory: 1274.7MB were necessary during the calculations. It seems we must take with caution this value.

#### 3.5 SIPINA

SIPINA is one of my old projects, specialized to decision tree induction. Unlike the other programs, SIPINA proposes a graphical display of the tree, and above all, possesses interactive functionalities.

This advantage is also a drawback because SIPINA must store much information on each node of the tree: the characteristics of split for the other variables, distributions, list of examples, etc. Thus, the memory occupation will be inevitably high compared to the other programs.

🔉 Sipina Research Version - [Lea	rning set editor]	
🔉 File Edit Data Statistics Inductio	n method Analysis View Window Help	- 8 ×
B New		
Open	Ouvrir	? 🗙
Save as	Regarder dans : 📄 big_dataset_decision_tree 💽 🔶 🖆 🏢 -	
Access foreign databases  Subsample management	Mes documents	
Exit	Bureau	
Learning method	Mes documents	
MethodName=Improved ChAID [1sc MethodClassName=TArbreDecision] Hdl=8 Merae=0.05	Poste de travail	
Split=0.001 TypeBonferroni=1 ValueBonferroni=1		
Sampling=0	Favoris reseau Nom du tichier : [wave500k.arff Fichiers de type : [WEKA File Format (".arff)	Annuler
E	diting NEW.FDM Attribu	ites : 1
Improved ChAID (Tschuprow Goodness of S	iplit)	

The memory occupation is 7.8 MB when we launch SIPINA. It becomes 67.1 MB after the dataset loading. The computation time is 25 seconds.

We select the C4.5 algorithm by clicking on the INDUCTION METHOD / STANDARD ALGORITHM menu. We set the minimal size of leaves to 1000.

🔉 Sipina Research Version - [Lea	rning set editor]							
🔉 File Edit Data Statistics Inductio	on method Analysis	View Window	Help			- 1	⇒ ×	
<u>Ğ </u> 🙀 h 🕒 🛛 Stan	dard algorithm							
	CLASS	Iv1	V2	V3	∀4	∨5	~	
Attni Select an induction method						-0.35		
Induction Graph Rule Induction Neural A limited search induction tree algorithm (Ca	network [Piseriminant an-	alysis Decision list	Other			0.07		
ID3-IV (Quinlan - 1986) GID3 (Cheng, Fauuad, Itani & Qian, 1988)	•					1.76		
ASSISTANT 86 (Cestnik, Konchenko & Bra ChAID (Kass - 1980)	itko - 1986)				C4.	5 paramete	ers	×
C4.5 (Quinlan - 1993)	99)				C.	L. for pessimis	tic pruning	
Improved C4.3 (Nakotomatala & Calich 15 Improved ChAID (Tschuprow Goodness of Cost sensitive C4.5 (Rakotomatala & Chauc	55) Split) hat - 2001)						25 🚖	í
One-Vs-All Decision Tree								
Lear					Siz	e of leaves :	1000	•
Met	C4.5 (Quinlan	- 1993)						
Hdl=	(	,			<b>.</b>	ľ		
Solit				VUK	X A S	amping All dataset		
TypeBonferroni=1	5 A	0.72	2.27	0.42	-1 2	Simple		
ValueBonferroni=1	6 A	1.39	-0.81	2.96	4.	Balanced	Size	:
	7 C	1.66	0.00	-0.04	2.	Stratified	300 -	-
Examples selection	8 A	1.19	-0.99	1.09	3.	Stratilieu	1	-
		0.84	-0.18	1.13	11.1	Г	. <b>/</b> ок	
					_	L	V UK	
I Improved ChAID (Tashuprow Goodpoor of S	alang NEW.FDM							

We must specify the status of the variables. We click on the ANALYSIS / DEFINE CLASS ATTRIBUTE menu. We set CLASS as TARGET, and the other variables (V1 ... V21) as INPUT.

🔉 Sipina Research Ve	rsion - [Learning set editor]	
🔉 File Edit Data Stat	istics Induction method Analysis View Window Help	_ 8 ×
<u>ð 🗎 🚔 </u> 🗎	Define class attribute	/4 \V5 🔨
Attribute selection		
	CLASS CLASS	<u> </u>
<u> </u>		
Learning method MethodName=C4.5 (Quinla		
MethodClassName=TArbrel Hdl=5		
Confidence level=25 Leaf size=1000	V7 V8 V8	
Sampling=0 SamplingPart=300	V9 V10 V10	
Examples selection	IV11 V15 V16	
	C Only discrete V17 V18	
<u> </u>	C Only continuous V19 C Both V20	
C4.5 (Quinlan - 1993)		¥
		🖊 OK 🛛 🗙 Annuler

We can launch the learning phase. The calculation time is 122 seconds. The memory occupation becomes 539.9 MB for a tree with 117 leaves.



The counterpart of this large memory occupation is that we can explore deeply each node. In the screen shot below, we observe the splitting performance for each variable and, when we select a variable, we see the class distribution on the subsequent leaves.



## 3.6 TANAGRA

TANAGRA is my current project. Unlike SIPINA, the implementation of the tree is widely modified. The interactive functionalities are anymore available, but the computation time and the memory occupation are optimized.

When we launch TANAGRA, the memory occupation is 7 MB. We click on the FILE / NEW menu in order to create a new diagram. We import the WAVE500K.ARFF data file.

💯 TANAGRA 1.4.	27	Choose your (	dataset and :	start download			7	
File Diagram Wind	Help							
🛱 Open	Data m	Diagram	tide :					
Save	Data III	Default t	itle					
Save as		Data min	ing diagram fi	ile name :				
Close		D:\DataN	/ining\Databa	ses for miningly	comparison TOWh	in dataset de 🔜		
Exit		Jane and		V				
		Dataset (	*.txt,*.arff,*.xls]	):				
		D:\DataN	/lining\Databa	ses_for_mining\	comparison_TOW\b	ig_dataset_de 🚰		
			Tanagra					? 🗙
			Regarder dans :	向 big dataset deci	sion tree	🔽 🕜 🗊 🗁 📖-		
				wave500k arff	_			
			3					T
Data visualizati	ion	Statistic	Mes documents récents					
Feature select	ion	Regressi						
Spv learning	.	Meta-spv lea						
Correlation sca	tterplot	🥂 Scatterplot	Bureau					
Export dataset		🔣 View datası						
🥂 Scatterplot		😳 View multip		/				
			Mes documents					
L								-
			Poste de travail					
			, osto do travali	•	4			
			<b>(</b>	Nom du fichier :	wave500k.arff	~	Our	vrir
			Favoris réseau	Fichiers de type :	WEKA dataset	~	Ann	uler

The processing time is 11 seconds; the memory occupation becomes 53.1 MB.

Before the learning phase, we must specify the role of each variable. We use the DEFINE STATUS component available into the tool bar. We set CLASS as TARGET, the other columns as INPUT (V1...V21).



We can now insert the C4.5 component and set the adequate parameters.

Pie Diagram Component Window Help       Image: Component Window Help         Image: Default title       Dataset (wave500k.arff)         Image: Dataset (wave500k.arff)       Dataset (wave500k.arff)         Image: Define status 1       Dataset (wave500k.arff)         Image: Dataset (wave500k.arff)       Dataset (wave500k.arff)         Image: Define status 1       Dataset (wave500k.arff)         Image: Define status 1       Dataset (wave500k.arff)         Image: Define status 1       Dataset (wave500k.arff)         Image: Dataset (wave500k.arff)       Dataset (wave500k.arff)         Image: Define status 1       Dataset (wave500k.arff)         Image: Dataset (wave500k.arff)       Dataset (wave500k.arff)
Default title     Dataset (wave500k.arff)       Image: Dataset (wave500k.arff)     Parameters       Image: Define status 1     Database: D:\DataWining\Databases: for mining\comparison_TOW\big_dataset_decision_tree\wave500k       Image: Distance of the status 1     Distate of the status 1       Image: Distance of the status 1     Distate of the status 1       Image: Distance of the status 1     Distate of the status 1       Image: Distate of the status 1     Distate of the status 1       Image: Distate of the status 1     Distate of the status 1       Image: Distate of the status 1     Distate of the status 1       Image: Distate of the status 1     Distate of the status 1       Image: Distate of the status 1     Distate of the status 1
Default title     Dataset (wave500k.arff)       Image: Define status 1     Parameters       Dataset (wave500k.arff)     Dataset (wave500k.arff)       Image: Define status 1     Dataset (wave500k.arff)       Dataset (wave500k.arff)     Dataset (wave500k.arff)       Image: Define status 1     Dataset (wave500k.arff)       Dataset (wave500k.arff)     Dataset (wave500k.arff)       Image: Define status 1     Dataset (wave500k.arff)       Dataset (wave500k.arff)     Dataset (wave500k.arff)       Image: Define status 1     Dataset (wave500k.arff)       Image: Dataset (wave500k.arff)     Dataset (wave500k.arff)       Image: Dataset (wave50k.arff)     Dataset (wave50k.arff)       Image: Dataset (wave50k.a
Supervised Learning 1 (C4.5) D:\Databases_for_mining\comparison_TOW\big_dataset_decision_tree\wave50C Parameters Supervised parameters Brewins
Parameters Supervised parameters Reputs
Supervised parameters
Execute
New Itoad C4.5 parameters
Continuous mis
none Parameters
Datasourc
Computation tim Min size of leaves : 1000
Allocated memor
Compon
Data visualization Statistics Nonparametric st
Feature selection Regression Factorial anal
Spv learning Meta-spv learning Spv learning asse UK Cancel Help
har logistic regression 🖧 C-RT 🎝 103 dars Reg TRIRLS
💑 C4.5 🖄 C-SVC 🔄 K-NN 🏓 Multilayer perceptron
🔀 C-PLS 🗧 🙀 Decision List 📓 Linear discriminant analysis 🔒 Multinomial Logistic Regression
<

The contextual menu VIEW launches the calculations. The processing time is 33 seconds. The maximal memory occupation during the computation is 121.6 MB, it is 73.5MB when the tree is finished. The decision tree includes 117 leaves.

## 3.7 WEKA

We use the EXPLORER module for WEKA. We click on the OPEN button of the PREPROCESS tab. We select the WAVE500K.ARFF data file.



The processing time is 10 seconds; the memory occupation becomes 253.2 MB.

📚 Weka 3.5.6 - Explorer									
Program Applications Tools Visualization Windows Help									
🖆 Explorer									
Preprocess Classify Cluster Associate Select attributes Visualize									
Classifier	ric\bjectEditor								
Choose 348 -C 0.25 -M 1000	weka.classifiers.trees.J48								
Test options	About		h						
⊙ Use training set	Class for generating a pruned or unpruned C4. More								
O Supplied test set Set									
O Cross-validation Folds 10	L								
O Percentage split % 66	binarySplits	False	<u> </u>						
More options	confidenceFactor	0.25							
(Nom) CLASS	debug	False	<b>~</b>						
Start Stop	minNumObj	1000							
Result list (right-click for options)	numFolds	3							
	reducedErrorPruning	False	~						
	saveInstanceData	False	✓						
	seed	1							
	subtreeRaising	True	<b></b>						
	uppruped	False	~						
	usel aplace	Ealco							
	usecapiace	1 disc	U						
OK	Open	Save OK	Cancel						

When the dataset is loaded, we select the CLASSIFY tab. We choose the J48 decision tree algorithm, we set the adequate parameters. We can now click on the START button.

The processing time is 338 seconds. The memory occupation becomes 699.6 MB. The tree includes 123 leaves.

#### 3.8 Summary

The results are summarized into the following table:

Program	Computation time (seconds)		Memory occupation (MB)				
	Data Importation	Tree induction	After launch	After importation	Max during treatment	After induction	
KNIME	47	270	92.6	160.4	245.8	245.8	
ORANGE	90	130	24.9	259.5	795.7	795.7	
R (package rpart)	24	157	18.8	184.1	718.9	718.9	
RAPIDMINER	7	298	136.3	228.1	1274.4	1274.4	
SIPINA	25	122	7.8	67.1	539.9	539.9	
TANAGRA	11	33	7.0	53.1	121.6	73.5	
WEKA	10	338	52.3	253.2	699.6	699.6	

Tanagra is much faster than other programs in the decision tree induction context. This is not surprising when we know the internal data structure of Tanagra. The data are organized in columns in memory. It is very performing when we treated individually variable, when searching the discretization cut point for the split operation during the tree induction for instance.

On the other hand, in an upcoming tutorial, we see that this advantage becomes a drawback when we need to handle simultaneously all the variables during the induction process, when we perform a scalar product during the SVM (Support Vector Machine) induction for instance. In this case, Tanagra is clearly outperformed by other programs such as RAPIDMINER or WEKA.

## 4 Conclusion

This tutorial is one of the possible point views of the decision tree induction with free software in the context of large dataset.

Of course, others point of views are possible: more predictors, a mixture of continuous and discrete predictors, more examples, etc. Because the programs are freely available, everyone can define and evaluate the configuration related to their preoccupation, and use their own dataset. Finally, this is what matters.