



Data mining with WEKA

A use-case to help you get started

Charalampos Mavroforakis BU CS105, Fall 2011

Starting WEKA

Weka Explorer	
Preprocess Classify Cluster Associate Select attributes Visualize	
Open file Open URL Open DB Gen	erate Undo Edit Save
Choose None	Apply
Current relation Relation: None Attributes: None Instances: None Sum of weights: None	Selected attribute Name: None Type: None Missing: None Distinct: None Unique: None
Attributes All None Invert Pattern	Visualize All
Status Welcome to the Weka Explorer	Log x0

Open Weka : Start > All Programs > Weka 3.x.x > Weka 3.x From the "*Weka GUI Chooser*", pick "*Explorer*". This is the main WEKA tool that we are going to use.

Opening a dataset

Open	openoitem		 Luitan	×
Look in	: 🚺 My Docum	ents	• 🤌 📂 🛄 •	
Recent Items	퉬 Aptana Si 퉬 MATLAB	udio 3 Workspace	Invoke options dia	log
Desktop			Some file formats offer options which can be when invoking the opt	er additional customized tions dialog.
My Documents				
Computer				
	File name:			Open
Network	Files of type:	Arff data files (*.arff)	-	Cancel
	IVEIIIG	Arff data files (*.arff) Arff data files (*.arff.gz)		
-		C4.5 data files (*.names)	=	
ome to the Weka I	Explorer	CSV data files (*.csv)	Log	. x
		JSON Instances files (*.json)		
_		JSON Instances files (*.json.gz)		
		libsvm data files (*.libsvm)	-	

To open a dataset (a .csv file in our case), we click "Open file ..." in the *Preprocess* tab and open the file that contains our data. **Remember** that in the open menu you have to choose csv if your file was saved as such. Let's open SPECT.csv

Transforming values to nominal (if needed)

eprocess Classify Cluster	Associate Select attributes	/isualize			
Open file Op	en URL Open DB	Gener	ate Undo	Edit.	Save
ilter					
Choose NumericToNo	ninal -R first-last				Apply
urrent relation Relation: SPECT Instances: 267	Att Sum of v	ributes: 23 veights: 267	Selected attribute Name: diagnosis Missing: 0 (0%)	Distinct: 2	Type: Numeric Unique: 0 (0%)
ttributes	🕝 weka.gui.GenericObject	tEditor			×)
No. Name 1 diagnosis 2 f1 3 f2 4 f3 7	About A filter for turning nume	ric attributes i	nto nominal ones.	More Capabilities	
5 f4 6 f5	attributeIndices first-last				Visualize A
8 77 9 76 10 79	debug False invertSelection False				• <u>212</u>
11 f10 12 f11 13 f12	Open	Save	ОК	Cancel	
14 12	Remove		0	0	D

Weka classified every attribute in our dataset as numeric, so we have to manually transform them to nominal. To do so, we will use a filter. We navigate to *NumericToNominal*, which is in *Unsupervised* > *attribute*. If we click on that, we will get to the options of that filter. Mainly, the most interesting one here is the *attributeIndices*, which enumerates all the attributes that you want the filter to be applied on. To finish, we click *Apply*.

Splitting the dataset

eprocess Classif	y Cluster Associate Select attributes Visualize	
Open file	Open URL Open DB Generate Undo Ed	it Save
ilter		
Choose Ran	domize -5 42	Apply
urrent relation Relation: SPEC Instances: 267	F-weka.filters.unsupervised.a Attributes: 23 Name: diagnosis Sum of weights: 267 Missing: 0 (0%) Distinct: 2	Type: Nominal Unique: 0 (0%)
ttributes	🜍 weka.gui.GenericObjectEditor	Weight
All No. Na	weka.filters.unsupervised.instance.Randomize About	55.0 212.0
1 dia 2 f1 3 f2 4 f3	Capabilities	
5 f4 6 f5 7 f6 8 f7	Open OK Cancel 2	▼ Visualize A
9 f 8	Save the current configured object	
10 f9 11 f10 12 f11 13 f12 14 f12	······································	
	65	

We have to split the dataset into two, 30% testing and 70% training. To do that, we first *Randomize* the dataset (Unsupervised > Instance), so that we create a random permutation.

Splitting the dataset

reprocess Classi	fy Cluster Associate Select attributes Visualize	
Open file	Open URL Open DB Generate Undo Edit	Save
Filter		
Choose Ren	novePercentage -P 30.0	Apply
Current relation Relation: SPEC Instances: 267	T-weka.filters.unsupervised.a Attributes: 23 Name: diagnosis Sum of weights: 267 Missing: 0 (0%) Distinct: 2	Type: Nominal Unique: 0 (0%)
Attributes	🚱 weka.gui.GenericObjectEditor	Weight
Ali No. Na	weka.filters.unsupervised.instance.RemovePercentage About A filter that removes a given percentage of a dataset.	55.0 212.0
2 f1 3 f2 4 f3 5 f4 6 f5 7 f6 8 f7	invertSelection False	▼) Visualize A
2 f1 3 f2 4 f3 5 f4 6 f5 7 f6 8 f7 9 f8 10 f9 11 f10 12 f11 13 f12	Capabilities invertSelection False percentage 30 Whether to invert the selection Open Open OK Cancel	▼ Visualize A

Then we apply *RemovePercentage* (Unsupervised > Instance) with percentage 30 and save the resulting dataset as training.

Splitting the dataset

reprocess Classi	fy Cluster Associate Select attributes Visualize	
Open file	Open URL Open DB Generate Undo Edi	t Save
Filter		
Choose Rer	novePercentage -P 30.0 -V	Apply
Current relation Relation: SPEC Instances: 187	T-weka.filters.unsupervised.a Attributes: 23 Name: diagnosis Sum of weights: 187 Missing: 0 (0%) Distinct: 2	Type: Nominal Unique: 0 (0%)
Attributes	🕝 weka.gui.GenericObjectEditor	Weight
All	weka.filters.unsupervised.instance.RemovePercentage	32.0 155.0
1 dia 2 f1 3 f2 4 f3 5 f4 6 f5	A filter that removes a given percentage of a dataset. More Capabilities invertSelection True	▼ Visualize A
7 f6 8 f7 9 f8	percentage 30.0	56
10 f9 11 f10		
12 f1 13 f12	· · · · · · · · · · · · · · · · · · ·	
	Remove 32	

After that, we undo and apply the same filter choosing *invertSelection* this time. This will pick the rest of the data (30%) so we save them as the testing.

🖉 Weka Explorer	_			_		_ 🗆 🗙
Preprocess Classify Cluster Associate Select attributes V	isualize					
Classifier						
weka	a arror		0.21	66		
🗄 🚂 lazy	uared error		0.33	94		
i⊞ meta	olute error		97.78	25 %		
En la rules	e squared er	ror	100.14	98 8		
DecisionTable	gion size (0	.95 level)	75	* *		
MSDules	of Instance	3	54			
OneR PART PART	Accuracy By	Class ===				
trees	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area
F	1	1	0.704	1	0.826	0.5
G	0	0	0	0	0	0.5
C	0	0	0	0	0	0.5
C C	. 0.704	0.704	0.495	0.704	0.581	0.5
	n Matrix ===					E
<u>q</u>	<pre> class class</pre>	nf_16010 0	7200211			
Close	b = '(16	918.973993	-33464.3579	28]'		
<		404.35/928	-50009.74180	o∠].		
Status OK					Log] 💉 × 0

From now on we will be using the training dataset. We switch to the tab "*Classify*" and we pick a classifier. Let's start with *OneR*, which is the same with the one we saw in the class.

🜍 Weka Explorer		- 0 ×
Preprocess Classify Cluster Associate Select attrib	outes Visualize	
Classifier		
Choose OneR -B 6		
Test options	Classifier output	
🔿 Use training set		^
Supplied test set Set		
Cross-validation Folds 10		
Percentage split % 66		
More options		
(Nom) diagnosis		
Start Stop		
Result list (right-click for options)		
		=
		-
	< III	4
Status		
ОК	Lo	• • • • • • • • • • • • • • • • • • •

We have to specify the attribute that we want to predict and the testing procedure. We first want to see how good OneR is as a model, so we use cross-validation. , and only after that will we go and check what it predicts on the unseen data.

🕝 Weka Explorer							_ 0	x
Preprocess Classify Cluster Associate S	elect attributes Visu	alize						
Classifier								
Choose OneR -B 6								
Test options	Classifier output							
O Use training set	Correctly Clas	ssified Ins	stances	149		79.6791	• 🧲	
Supplied test set	Incorrectly C	lassified 1	Instances	38		20.3209	8	
Supplied test set	Kappa statist:	ic		0				
Cross-validation Test on a user-specifie	d dataset psolute	error		0.20	32			
Percentage split % 66	ROOT mean squa	ared error		0.45	08			
Mara antiana	Relative absol	lute error		62.34	38 %			
More opuoris	Root relative	squared en	cror	111.99	55 8 01 8			
	Mean rel regi	1383 (U.95	Tever)	/9.0/	8 91 - 8			
(Nom) diagnosis 🛛 👻	Total Number (of Instance	2.55 IEVEL) 20	187				
Start Stop	=== Detailed 2	Accuracy By	/ Class ===	-				
Result list (right-click for options)								
02:00:07 - rules.OneR		TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Are	sē 👘
		0	0	0	0	0	0.5	
		1	1	0.797	1	0.887	0.5	
	Weighted Avg.	0.797	0.797	0.635	0.797	0.707	0.5	
	=== Confusion	Matrix ===	-					E
	a b <	- classifie	ed as					
	0 38 4	a = 0						
	0 149 1	b = 1						-
	۰ III						Þ	
Status							_	
ок						Log		, x 0

In the output, we get information about the average accuracy and the confusion matrix of our model.



In order to check how well we do on the unseen data, we select "*supplied test set*", we open the testing dataset that we have created and we specify which attribute is the class. We run the algorithm again and we notice the differences in the confusion matrix and the accuracy.

Association learning

Weka Explorer	
Preprocess Classify Cluster Associate Select attributes	Visualize
Associator	
🔒 weka	I -5 -1.0 -A -c 5
associations	
Apriori	or output
FilteredAssociator	A
FPGrowth	
	ri
	==
	um support: 0.1 (13 instances)
	um metric <confidence>: 0.9</confidence>
	r of cycles performed: 16
	ated sets of large itemsets:
	abea beab of farge formbeab,
	of set of large itemsets L(1): 8
	of set of large itemsets L(2): 1
	rules found:
	egion=Sub-Saharan Africa 33 ==> income-2010='(-inf-16918.973993
	egion=Sub-Saharan Africa income change from 1996='(56.794746-17
	ncome change from 1996='(-inf-56./94/46]' 1/ ==> income-2010='
Close	"ncome change from 1996='(56.794746-1794.332988]' 57 ==> income-
Close	
	4
Status	
OK	Log x0

If all of our attributes are nominal (in case they are not, we can discretize them in the *Preprocess* tab) we can also do association learning. In order to do that, we switch to the *Association* tab and we choose the *Apriori* algorithm. You can play around with its parameters if you want.

Association learning

C Weka Explorer	_ 🗆 X
Preprocess Classify Cluster Associate Select attributes Visualize	
Associator	
Choose Apriori -I -N 10 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.1 -S -1.0 -A -c 1	
Acceptator autout	
Start Stop Associator output	
Result list (right-dick) 1 co	
02:11:44 - Apriori	
Size of set of large itemsets L(7): 1	
Large Itemsets L(7):	
f2=0 f6=0 f7=0 f11=0 f12=0 f17=0 f18=0 89	
1 59	
Best rules found:	
1. f8=1 f13=1 66 ==> diagnosis=1 63 conf:(0.95)	
2. f16=1 61 ==> diagnosis=1 58 conf:(0.95)	
3. f13=1 89 ==> diagnosis=1 84 conf:(0.94)	
4. f7=0 f13=1 63 ==> diagnosis=1 59 conf:(0.94)	
5. f13=1 f15=0 62 ==> diagnosis=1 58 conf: (0.94)	
<pre>b. II3=1 II/=0 76 ==> diagnosis=1 71</pre>	
$\begin{array}{c} 7. 122=1 \ 74 \implies \text{diagnosis=1 69} \text{conf:} (0.93) \\ 8. \ f21=1 \ 72 \implies \text{diagnosis=1 67} \text{conf:} (0.93) \end{array}$	
9. f13=1 f18=0 71 ==> diagnosis=1 66 conf:(0.93)	
10. f6=0 f13=1 70 ==> diagnosis=1 65 conf:(0.93)	
	+
	۲.
Status	
OK	x0

We could set *car* to True (so that it produces rules that predict the class attribute) and specify the index of the attribute that will be considered as class. *minMetric* sets the threshold of confidence and *numRules* limits the number of rules that will be created. The result will be a set of rules that predict the class, together with their confidence.