1 Topic

Induction of fuzzy rules using Knime.

This tutorial is the continuation of the one devoted to the induction of decision rules (<u>Supervised</u> <u>rule induction – Software comparison</u>). I have not included Knime in the comparison because it implements a method which is different compared with the other tools. Knime computes fuzzy rules. It wants that the target variable is continuous. That seems rather mysterious in the supervised learning context where the class attribute is usually discrete. I thought it was more appropriate to detail the implementation of the method in a tutorial that is exclusively devoted to the Knime rule learner (version 2.1.1). Especially, it is important to detail the reason of the data preparation and the reading of the results. To have a reference, we compare the results with those provided by the rule induction tool proposed by Tanagra.

Scientific papers about the method are available on line^{1,2}.

2 Dataset

We use a version of the IRIS dataset with only two descriptors³ (PET_LENGTH, PET_WIDTH). The interest is that we can represent graphically the data and the rules characteristics.

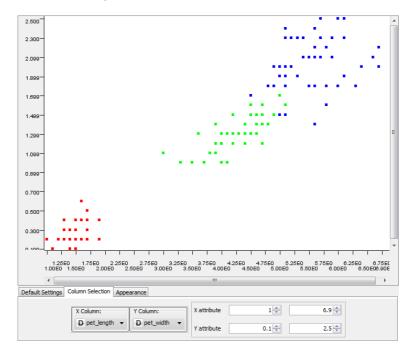


Figure 1 – Iris dataset - Positioning of the groups into a scatter plot (pet.length; pet.width)

¹ M.R. Berthold, « Mixed fuzzy rule formation », International Journal of Approximate Reasonning, 32, pp. 67-84, 2003. http://www.inf.uni-konstanz.de/bioml2/publications/Papers2003/Berto3_mixedFR_ijar.pdf

² T.R. Gabriel, M.R. Berthold, « Influence of fuzzy norms and other heuristics on mixed fuzzy rule formation », International Journal of Approximate Reasoning, 35, pp.195-202, 2004.

http://www.inf.uni-konstanz.de/bioml2/publications/Papers2004/GaBeo4_mixedFRappendix_ijar.pdf

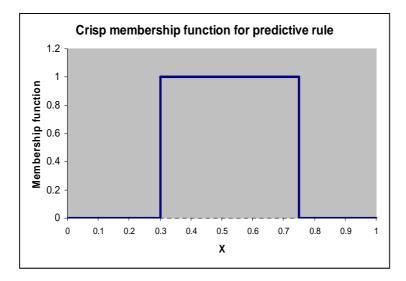
³ <u>http://eric.univ-lyon2.fr/~ricco/tanagra/fichiers/iris2D.txt</u>

3 Induction of fuzzy rules using Knime

Roughly speaking, we say that the fuzzy rules incorporate a gradation in the definition of group membership regions in the representation space.

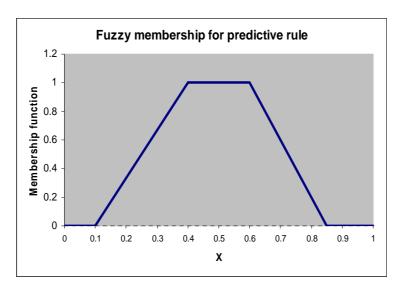
Let us consider the "crisp" decision rule:

The (one-dimensional) region membership can be defined as follows



For an instance ω , if X (ω) \in [0.3; 0.5[, the membership degree is $\mu(\omega) = 1$; if its value is outside of this interval, the membership degree is null [$\mu(\omega) = 0$].

Without going into complicated discussions, it is clear that the thresholds "0.3" and "0.5" seem a bit arbitrary. They are going with some uncertainty because they have estimated using a sample. It is more convenient to include a certain gradation in the definition of the regions. For instance, instead a rectangle, we can use a trapezoid with the following coordinates <0.1, 0.4, 0.6, 0.85>



Thus:

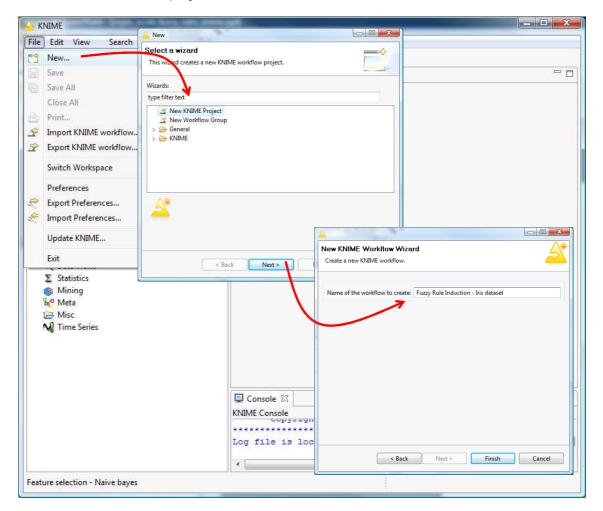
- For an individual with X = 0.5, it fully activates the rule with μ = 1. The conclusion is "Y = +".
- For another instance with X = 0, it is not covered by the rule μ = 0. The conclusion is "Y = -".
- For another instance X = 0.2, it partially activates the rule with $\mu \approx$ 0.3. This reduces the scope of the conclusion.

This gradation is certainly more interesting for the comprehension of the decision. It is less arbitrary. But it is not without problems: (1) the integration of the decision rule into an information system for the deployment is not very easy, the rule cannot be translated directly in a SQL query; (2) many rules can be activated during the classification on an unseen instance, they have not the same weight, sometimes with a contradictory decision, we must implement a strategy for the management of incompatibilities.

Some authors think that fuzzy rules learners outperforms crisp decision rule learners because it reduces the variance of the classifiers.

3.1 Creating a workflow and importing the dataset

We create a new project under Knime by clicking on the FILE / NEW menu. We set «Fuzzy Rule Induction - IRIS dataset » as project name.

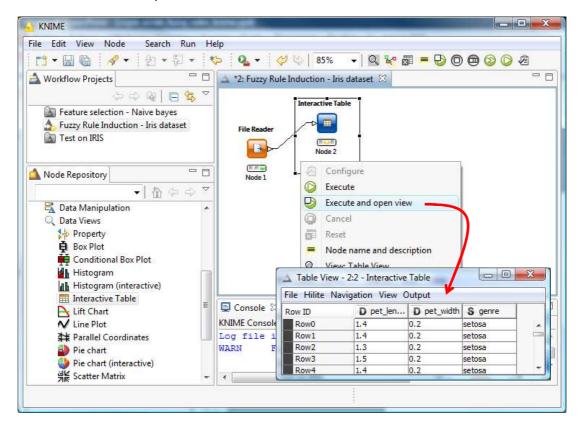


We obtain a new Workflow Projects.

To import the IRIS2D.TXT data file, we use the **FILE READER** component. We select the data file (CONFIGURE menu): the first row corresponds to the name of the variables; the column delimiter is the "tab" character.

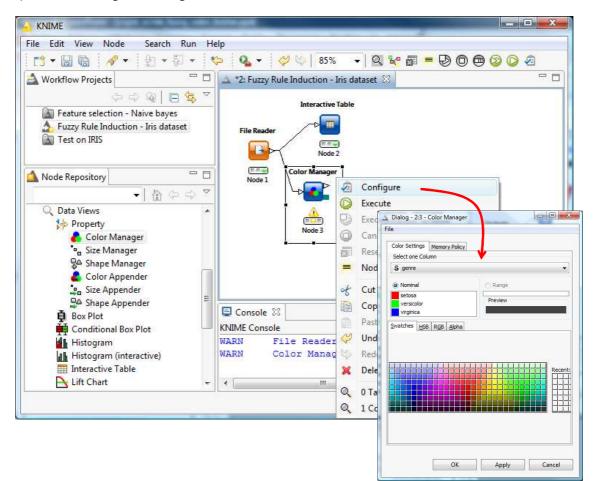
-
riew)
action/fuzzy/iris2d.txt 👻 Browse
Iction/fuzzy/iris2d.txt + Browse
> 👻 Advanced
abs
s Single line comment:
es (* = name/type user settings)
^
ge sa sa sa

We click on the EXECUTE menu to perform the importation. We can visualize the dataset using **INTERACTIVE TABLE** component. We click on the EXECUTE and OPEN VIEW contextual menu.

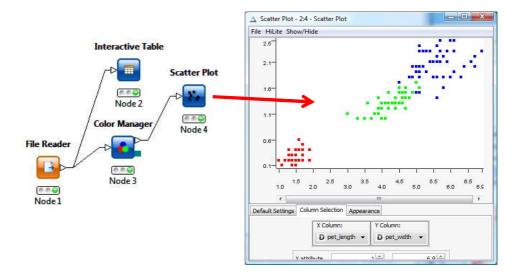


3.2 Graphical representation of the dataset

We want to make a scatter plot of the dataset by distinguishing the group memberships (Figure 1). We add the **COLOR MANAGER** component. We set the parameters as follows in order to colorize the points according to the target attribute GENRE.



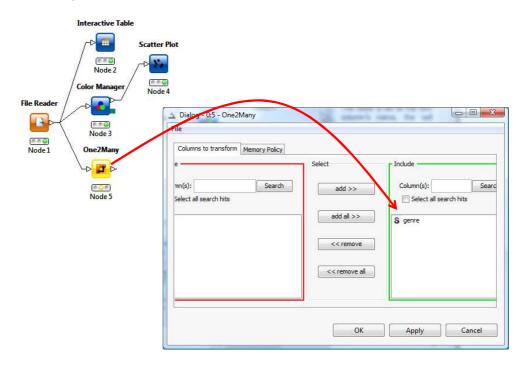
We add the **SCATTER PLOT** component (*Data Views*). We connect the COLOR MANAGER, then we click on the EXECUTE and OPEN VIEW menu. We can specify interactively the attributes on the horizontal and the vertical axes.



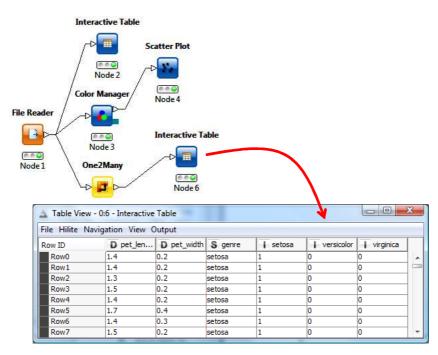
3.3 Transforming the target attribute for the induction process

In the supervised learning context, the target attribute is discrete. Yet, Knime asks one or more continuous variables. In reality, the target columns correspond to the degree of membership to each value of the original target attribute here.

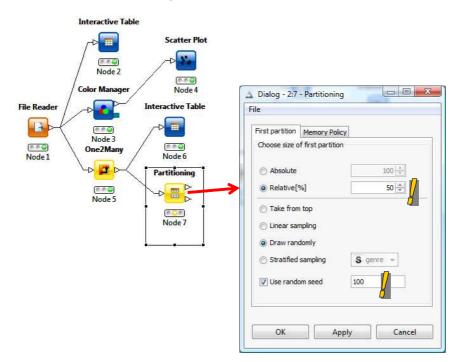
The GENRE attribute consists of 3 values (setosa, versicolor and virginica), we must define 3 new binary columns o/1. We use the **ONE2MANY** component (*Data Manipulation / Column / Transform*). We set the following parameters.



Again, we use the INTERACTIVE TABLE component (Data Views) to visualize the dataset.

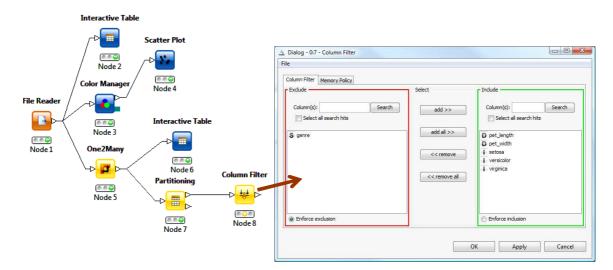


We use a test sample in order to assess the "true" performance of the classifier. Thus, first we must subdivide the dataset using the **PARTITIONNING component** (*DATA MANIPULATION / ROW / TRANSFORM*). We use the half (50%) for the training phase. We set USE RANDOM SEED = 100 to obtain the same results at each launching of the workflow.

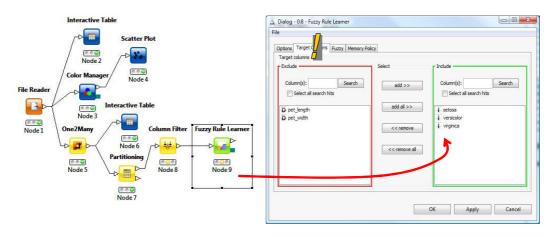


3.5 Induction of fuzzy rules

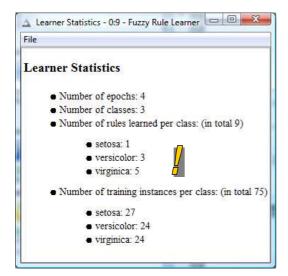
Now, we must specify the attribute to use during the analysis. The GENRE column is not useful here. We use COLUMN FILTER (*Data Manipulation / Column / Filter / Column Filter*).



Then, we add **FUZZY RULE LEARNER** (*Mining / Rule Induction / Fuzzy Rules*). We connect only the train set for the learning phase. We click on the CONFIGURE contextual menu. The most important here is to set appropriately the TARGET settings. We set the binary columns coming from the target attribute GENRE i.e. setosa, versicolor, virginica.

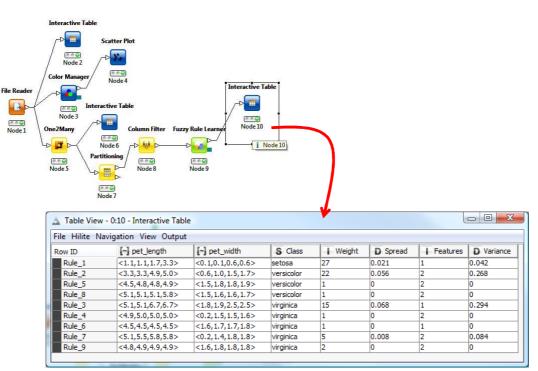


The other columns are the descriptors. We validate and we click on EXECUTE and OPEN VIEW.



The target attribute contains 3 values (number of classes); the learning algorithm supplies 9 rules, 1 for the "setosa" class value, 3 for "versicolor" and 5 for "virginica"; last, we have the number of instances for each class value.

We use INTERACTIVE TABLE to visualize the rules.



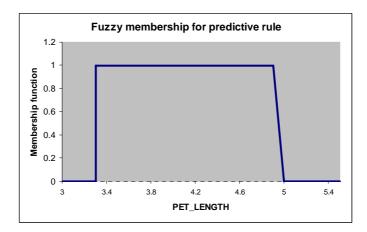
Note 1: FUZZY RULE LEARNER can handle only continuous descriptors. We must transform the discrete attributes, using ONE2MANY component for instance, if we want to use them.

Note 2: The algorithm supplies curiously 9 rules. We see graphically that 3 rules are sufficient. I think we must set carefully the settings of the algorithm before the launching of the learner.

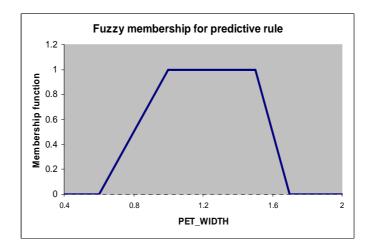
Note 3: The reading of the rules becomes tedious when the number of descriptors increases. We must interpret 4 values for each attribute.

3.6 Interpreting the rules

We study the rule n°2 here: the decision is "versicolor", its weight is 22 i.e. 22 instances are covered by the rule; other indicators are supplied. For each variable, we have the 4 coordinates of the trapezoid describing for the membership function. For PET_LENGTH, we have <3.3, 3.3, 4.9, 5.0>

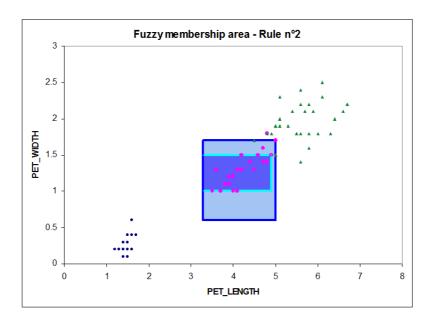


For PET_WIDTH: <0.6, 1.0, 1.5, 1.7>



3.7 Representation of the rule in a scatter plot

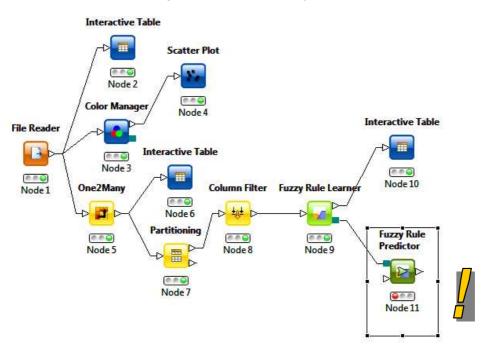
In a two dimensional diagram, we can represent the rule membership as follows.



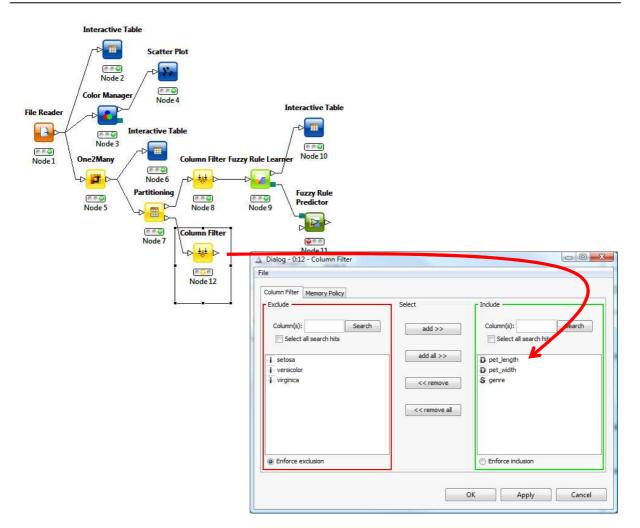
The inner rectangle (dark blue) designates an area of strong membership; the outer rectangle (blue) denotes the limit of not belonging (outside this area, the degree of membership of an individual to the rule is zero); from the inner rectangle to the outer rectangle, there is a gradation of membership. We note that when we consider all the rules of a classifier, some of them are overlapped. That makes the classification of a new individual complicated.

3.8 Generalization error rate – Confusion matrix

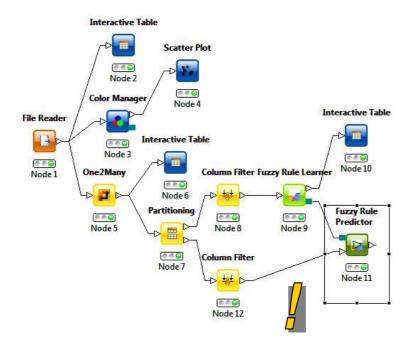
We want to evaluate the generalization capabilities of our classifier. For this, we use the **FUZZY RULE PREDICTOR** component (*Mining / Rule Induction / Fuzzy Rules*).



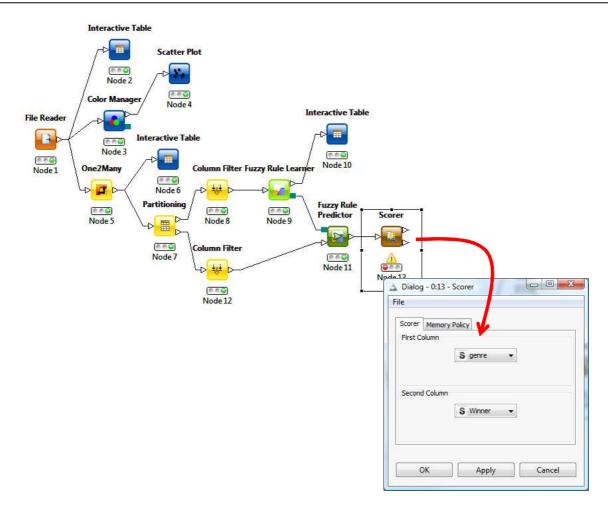
We filter the attributes coming from **PARTITIONING**. We use only the predictive variables and the target variable "GENRE" using **COLUMN FILTER**.



Then we use FUZZY RULE PREDICTOR to perform the prediction on the test set.



SCORER (*Mining / Scoring*) enables to compute the confusion matrix. We compare the target variable (GENRE) with the prediction of the classifier (WINNER).



We click on EXECUTE and OPEN VIEW. The test error rate is 4%.

File Hilite				
genre \Wi	setosa	versicolor	virginica	
setosa	23	0	0	
versicolor	0	25	1	
virginica	0	2	24	

4 Induction of « crisp » rules using Tanagra

We want to analyze the behavior of a "crisp" rule learner on the same dataset, especially about the definition of the region membership. We use the RULE INDUCTION component of Tanagra. *About the comparison of the rule inducer in various tools (Weka, Orange, R), see* "<u>Supervised rule induction –</u> <u>Software comparison</u>".

4.1 Creating a diagram and importing the data file

We launch Tanagra, we click on the FILE / NEW menu to create a new diagram. We select the data file IRIS2D.TXT.

TANAGRA 1.4.35 File Diagram Window	Help			alter a			
New							
Open Dat of Save Save as Close Exit	hoose your dataset and start download Diagram title : Default title Data mining diagram file name : D'Data Mining iDatabases_for_mi Dataset (* bd.* arff,* xla) ; inglidataset_for_soft_dev_and_co		4				
			Tanagra Regarder dans : Emplacements récents Bureau		ie de m Type	Taille	
Data visualization Feature selection Spv learning	Statistics Regression Meta-spv learning	Ce Nonparametric s Factorial ana Spv learning asse	Maison				
 Correlation scatterplo Export dataset Scatterplot 	t 🦉 Scatterplot with label 👿 View dataset 🤃 View multiple scatterp	lot	Ordinateur	Nom du fichier : Types de fichiers :	iris2d.bd Text.file		Ouvrir Annuler
		-					ગોતોનમાં ગોગોગીગી તેમોગી

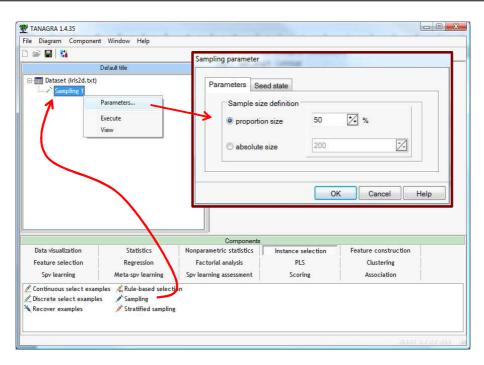
The dataset is loaded, we have 3 variables and 150 instances.

Dataset (ir	is2d.txt)			- • •
Datase 3 attribute(ription		•
150 exampl				
Attribute	Category	Informations		
pet_length	Continue	-		E
pet_width	Continue		<u>U</u>	
genre	Discrete	3 values		
•		III		•

Note: The decimal separator is "." in the IRIS2D.TXT data file. If your computer does not handle this kind of decimal separator, you must modify the file using a text editor.

4.2 Partitioning the dataset

We use the SAMPLING (*Instance Selection* tab) component to create the train and test samples. We set the following parameters (50% for the train sample).

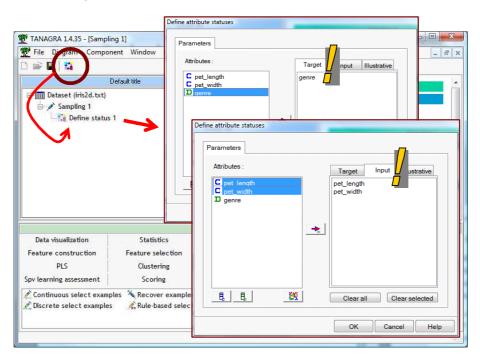


We validate and we click on the VIEW menu: 75 instances are incorporated into the train sample.

4.3 Discretization of continuous attributes

The RULE INDUCTION component can handle discrete descriptors only. We must discretize (subdividing into intervals) PET_LENGTH and PET_WIDTH before the learning process. We use a supervised approach.

We insert the DEFINE STATUS component into the diagram. We set PET_LENGTH and PET_WIDTH as INPUT, GENRE as target.



Then we add MDLPC (*Feature Construction*), a component for a supervised discretization process. We click on the VIEW menu to obtain the cut points. Two new variables are generated.

) 📽 🖪 🗱		
D	efault title	Data description
Dataset (iris2d.txt)		Attributes discretized 2 Examples 75
Define status *	Parameters	Generated attributes
7	Execute	Source New att Intervals Cut points
	View	pet_length d_mdlpc_pet_length_1 3 (2.3500; 4.8500)
	view	pet_width d_mdlpc_pet_width_1 3 (0.7500; 1.7500)
		Computation time : 15 ms. Created at 21/01/2010 08:52:36
	CL II II	Components
Data visualization	Statistics Feature selection Clustering	Nonparametric statistics Instance selection Regression Factorial analysis Spv learning Meta-spv learning
Feature construction PLS Spv learning assessment	Scoring	Association

The cut points are (2.35; 4.85) for PET_LENGTH; (0.75; 1.75) for PET_WIDTH.

Note: It is very important to partitioning the dataset before the discretization process. Indeed, it is already a part of the learning process, the cut points must be determined using the training sample.

4.4 Induction of rules

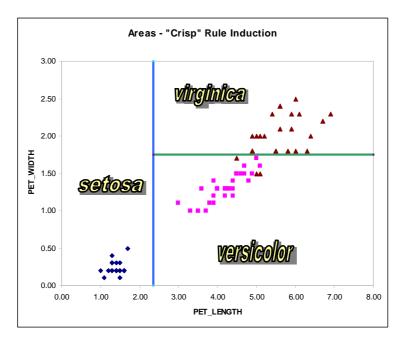
We can launch the rule induction process. We add again the DEFINE STATUS component. We set as INPUT the discretized variables; GENRE is always the target attribute.

TANAGRA 1.4.35 - [MDLPC 1]	Define attribute statuses	×
INVACIA 14:35 - [WIDLPC 1] File Default Window Default Window Hi Hi Hi Default Window Hi Hi Hi <	Parameters Attributes : C pet_length G pet_width D d_mdlpc_pet_length_1 d_mdlpc_pet_width_1 Define attribute statuses Parameters Attributes : C pet_length C pet_length Genre Define attribute statuses Parameters Attributes : C pet_length Genre Image: Status attribute statuses	
Data visualization Statistics Feature construction Feature select PLS Clustering Spv learning assessment Scoring I □ 1_Binarize LL Cont to disc L Eq A Binary binning L EqFreq Disc Fo	Width	

🏆 File Diagram Coi 🗋 🚅 🔚 🌠	mponent Window Help					- 8
Dataset (iris2d.t Dataset (iris2d.t Sampling 1 Sampling 1 Sampling 1 Sampling 1	atus 1	Target attribute # descriptors Number of	rules = 2			
and a second sec	Supervised Learning 1 (Rule In Parameters Supervised parameters Execute View	duction) // IF d_mdpc_pet_le	P-based system ntecedent ngth_1 in [m_<_2.34999990] idth_1 in [m_>=_1.75000000]		Distribution (21; 0; 0) (0; 0; 22) (0; 29; 3)	
		Components				
Data visualization Feature selection Spv learning	Statistics Regression Meta-spy leasning	Nonparametric statistics Factorial analysis Spv learning assessment	Instance selection PLS Scoring	Feature constr Clusterin Associatio	8	
• Multilayer perceptr • Wultinomial Logistic • Naive bayes		De Radial basi A, Rnd Tree ■ Rule Induc	and a second	WM III	.7	

We add the RULE INDUCTION component (*Spv Learning*). We click on the VIEW menu.

We obtain 3 rules. The last one is the "default rule". Here also, the rules define various regions in the representation space. But, the frontiers are "crisp". A new instance is assigned to one and only one region.



4.5 Assessment on the test set

To assess the accuracy of the classifier, we insert a new DEFINE STATUS into the diagram. We set GENRE as target, PRED_SPV_INSTANCE_1, the prediction column generated by the classifier, as input.

	onent Window Help	Define attribute statuses	_ B ×
		Parameters	tion 0; 0) = 122) 9; 3)
Data visualization Feature selection Spv learning Multilayer perceptron Multinomial Logistic Rep Maive bayes	Statistics Regression Meta-spv learning Meta-spv learning PLS-DA gression Meta-Spv Prototype-NN	B genre □ d_mdipc_pet_length_1 □ d_mdipc_pet_width_1 □ pred SpMnstance 1 €, E, E, čX Clear all Clear set	elected

Then we add the TEST component (*Spv Learning Assessment*). It computes automatically the confusion matrix and some indicators on the test set. We click on the VIEW menu. Like for the fuzzy rules, the test error rate is 4%.

Results Results Sampling 1 Sampling 1 Error rate 0.0400 Values prediction Confusion matrix Values prediction Confusion matrix Value Recall 1-Precision setosa versicolor virginica Sum Supervised Learning 1 (Rule Induction) Value Recall 1-Precision setosa versicolor 0 0 29 0 0 29 Parameters Execute Virginica 0 22 24 75 Data visualization Statistics Nonparametric statistics Instance selection Feature construction Spv learning Meta-spv learning Spv learning Spv learning Spv learning Spv learning Spv learning	Defau	ult title									
Sampting 1 Performation Status 1 Error rate O.0400 Values pred_5pvinstance_1 Superised Learning 1 (Rule Induction) Supervised Learning 1 (Rule Induction) Parameters Parameters Parameters Execute Virginica 0 Components View Data visualization Statistics Instance selection Feature selection Regression Regression Supervised Learning 1 (Rule Induction) Visual Sampting 1 (Rule Induction)	🖂 📖 Dataset (iris2d.txt)						Re	esults			
MDLPC 1 Values prediction Confusion matrix Supervised Learning 1 (Rule Induction) Supervised Learning 1 (Rule Induction) Setosa 29 0 0 29 Parameters Parameters Execute View Vignica 0.9200 0.0000 setosa 29 0 0 29 Data visualization Statistics Nonparametric statistics Instance selection Feature selection Feature selection Feature selection Feature selection Clustering							pred_Sp	vinstance_1	i		
Value Precision setosa versicolor virginica Sum Supervised Learning 1 (Rule Induction) Statistics 100000 setosa 1.00000 versicolor 0 0 29 Parameters Parameters Virginica 0.9200 0.0417 virginica 0 2 22 24 75 Data visualization Statistics Nonparametric statistics Instance selection Feature construction Feature selection Regression Factorial analysis PLS Clustering	🗄 🙀 Define status 1				Error rat	e	1		0.0400		
Value Recall Precision setosa versicolor virginica Sum Supervised Learning 1 (Rule Induction Define status 3 Image: Supervised Learning 1 (Rule Induction Define status 3 Image: Supervised Learning 1 (Rule Induction Define status 3 Image: Supervised Learning 1 (Rule Induction Setosa 1.000 0.0000 setosa 2.9 0 0 2.9 Image: Supervised Learning 1 (Rule Induction Define status 3 Parameters Execute Virginica 0.9524 0.0009 versicolor 0 2.0 2.23 2.25 Sum 2.9 0.0417 Virginica 0 0 2.0 2.24 75 View Vi				Valu	es predi	iction	-	C	onfusion mat	rix	
Setosa 1.0000 0.0000 setosa 29 0 0 29 Parameters Parameters versicolor 0.9524 0.0909 versicolor 0 20 1 21 Parameters Execute Virginica 0.920 0.0417 virginica 0 2 23 25 Data visualization Statistics Nonparametric statistics Instance selection Feature construction Feature construction Feature construction Feature construction				Value	Recall	1-Precision		setosa	versicolor	virginica	Sum
Parameters versicolor 0.9524 0.0909 versicolor 0 20 1 21 Parameters virginica 0.9200 0.0417 virginica 0 2 23 25 Sum 2.9 2.2 2.4 75 View Components View View View Data visualization Statistics Nonparametric statistics Instance selection Feature construction Feature selection Regression Factorial analysis PLS Clustering		7.00	nduction)	setosa	1.0000	0.0000	setosa	29	0	0	29
Parameters virginica 0.9200 0.0417 virginica 0 2 23 25 Sum 29 22 24 75 Sum 29 22 24 75 Data visualization Statistics Nonparametric statistics Instance selection Feature construction Feature selection Regression Factorial analysis PLS Clustering					0.9524	0.0909	versicolor	0	20	1	21
Execute Components Data visualization Statistics Nonparametric statistics Instance selection Feature selection Regression Factorial analysis PLS Clustering		The second second		virginica	0.9200	0.0417	virginica	0	2	23	
View Components Data visualization Statistics Nonparametric statistics Instance selection Feature construction Feature selection Regression Factorial analysis PLS Clustering		Everute				~	Sum	29	22	24	75
Components Data visualization Statistics Nonparametric statistics Instance selection Feature construction Feature selection Regression Factorial analysis PLS Clustering		and the second second									
Feature selection Regression Factorial analysis PLS Clustering		1.00		Corr	ponents	ŝ					
	Data visualization	Statistics	Nonparan	netric statistic	s	Instance sel	lection	Feature	constructio	n	
Spv learning Meta-spv learning Spv learning assessment Scoring Association	Feature selection	Regression	Facto	orial analysis		PLS		CI	ustering	1	
	Spv learning	Meta-spv learning	Spv learn	ning assessmen	t	Scorin	Ig	As	sociation	1	
	Bootstrap	Leave-One-Out		Train-test							

Actually, the error rate is not essential in this tutorial. We know that the IRIS problem is easy to learn. The challenge was to understand and to visualize the decision regions according to the rules.

5 Conclusion

The scientific interest of the predictive fuzzy rule induction is undeniable. However, it is little known outside a small circle of researchers. Mainly because it is not available into the popular tools, it restricts the diffusion of technology. Knime is one of the main tools which implement this approach. It seemed interesting to put forward the component to a specific tutorial.