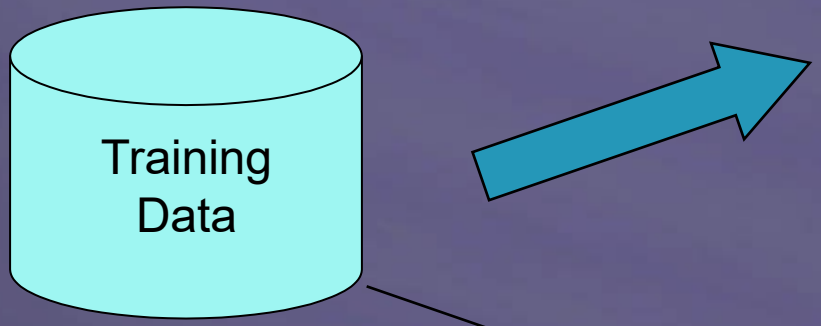
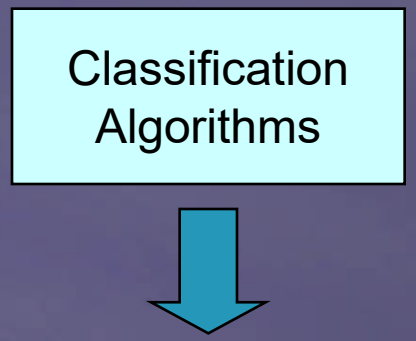
Classification

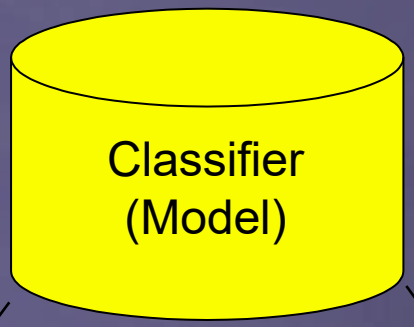
Cristóbal Romero
Córdoba University, Campus Universitario de Rabanales, 14071, Córdoba, Spain
cromero@uco.es

KDIS Lab
KNOWLEDGE DISCOVERY AND INTELLIGENT SYSTEMS

Classification Process: Model Construction



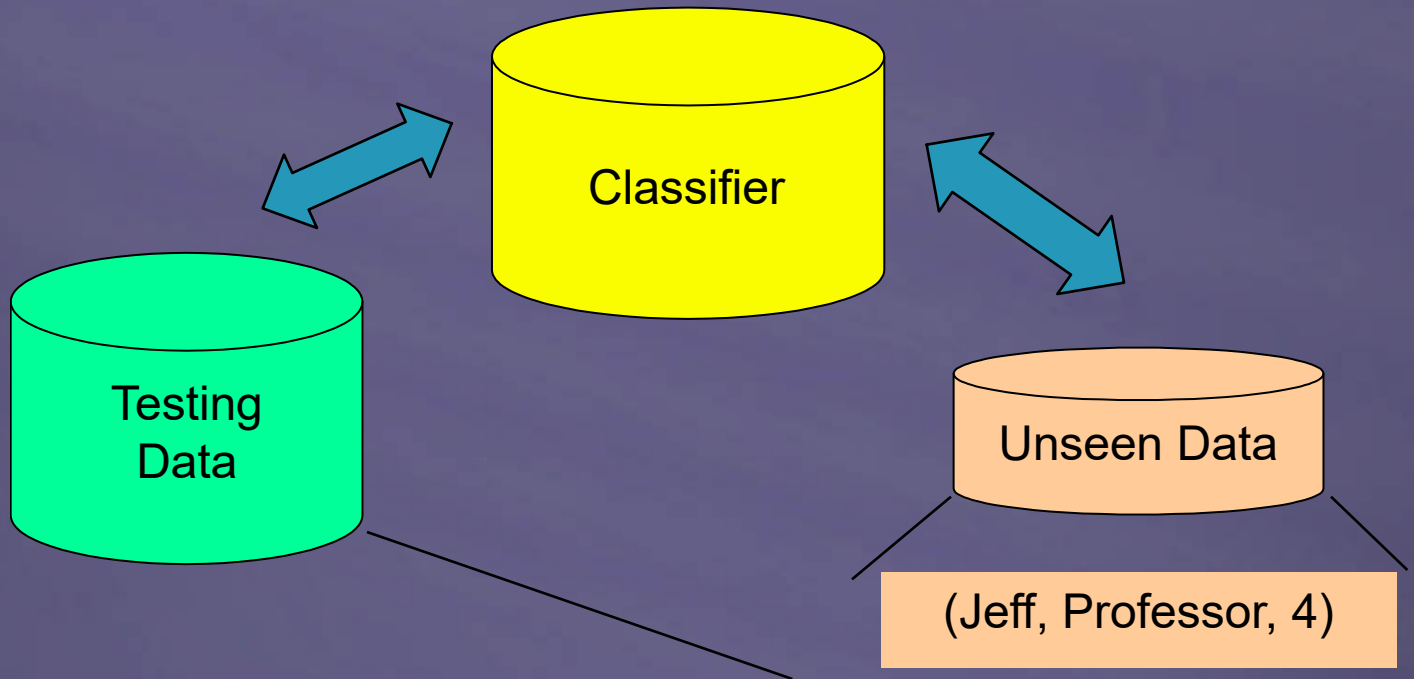




NAME	RANK	YEARS	TENURED
Mike	Assistant Prof	3	no
Mary	Assistant Prof	7	yes
Bill	Professor	2	yes
Jim	Associate Prof	7	yes
Dave	Assistant Prof	6	no
Anne	Associate Prof	3	no

IF rank = ‘professor’
OR years > 6
THEN tenured = ‘yes’

Classification Process: Model usage



NAME	RANK	YEARS	TENURED
Tom	Assistant Prof	2	no
Merlisa	Associate Prof	7	no
George	Professor	5	yes
Joseph	Assistant Prof	7	yes

Tenured?

Yes

Classification Techniques

- Decision Tree based Methods
- Rule-based Methods
- Neural Networks
- Naïve Bayes and Bayesian Belief Networks
- Support Vector Machines

Classifier Accuracy Measures

Confusion Matrix

TP	True Positives
TN	True Negatives
FP	False Positives
FN	False Negatives

		Predicted Label	
		positive	negative
Known Label	positive	TP	FN
	negative	FP	TN

- **Accuracy** *(TP + TN) / (TP + TN + FP + FN)*
  - Percentage of correct predictions
- **Precision** *TP / (TP + FP)*
  - Percentage of correct positive predictions
- **Recall / Sensitivity** *TP / (TP + FN)*
  - Percentage of positively labeled instances, also predicted as positive
- **Specificity** *TN / (TN + FP)*
  - Percentage of negatively labeled instances, also predicted as negative

Evaluating the Accuracy of a Classifier or Predictor

- Holdout method
  - Given data is randomly partitioned into two independent sets
    - Training set (e.g., 2/3) for model construction
    - Test set (e.g., 1/3) for accuracy estimation

- Cross-validation (*k*-fold, where k = 10 is most popular)
  - Randomly partition the data into *k mutually exclusive* subsets, each approximately equal size
  - At *i*-th iteration, use D_i as test set and others as training set

Classification for predicting student performance

Classification for predicting student performance

- We are going to compare different data mining techniques for classifying students based on both students' usage data in a web-based course and the final marks obtained in the course.
- We have used real data from 7 Moodle courses from Cordoba University students. We have also applied discretization and rebalance preprocesing techniques on the original numerical data in order to verify if better classifier models are obtained.
- Finally, we have developed a specific Moodle data mining tool for making this task easier for instructors.

Background

- The task of classification consists of predicting the value of a (categorical) attribute (named the class attribute) based on the values of other attributes (the predicting attributes).
- There are different classification methods:
  - **Statistical classification algorithms** such as linear discriminant analysis, least mean square quadratic, kernel and k nearest neighbours.
  - **Decision tree algorithms** such as C4.5 and CART.
  - **Rule induction algorithms** such as CN2, AprioriC, XCS, SIA, Corcoran and GGP.
  - **Fuzzy rule learning algorithms** such as LogitBoost, MaxLogitBoost, AdaBoost, GP, GAP, SAP and Chi.
  - **Neural network algorithms** such as Multilayer Perceptron, RBFN, incremental RBFN, decremental RBFN, GANN and NNEP.

Experimental Results

- We have carried out some experiments in order to evaluate the performance and usefulness of the previous classification algorithms for predicting students' final marks based on students' usage data.
- Our final objective is to classify students with similar final marks into groups depending on the activities done in Moodle courses.
- To do it, we have used the data of 438 Cordoba University students in 7 Moodle courses.

Experimental Results

- Firstly, we have created a summary table which integrates the most important information for our objective (Moodle activities and the final marks obtained in the course) with numerical data.

Name	Description
course	Identification number of the course.
n_assigment	Number of assignments done.
n_quiz	Number of quizzes taken.
n_quiz_a	Number of quizzes passed.
n_quiz_s	Number of quizzes failed.
n_posts	Number of messages sent to the forum.
n_read	Number or messages read on the forum.
total_time_assignment	Total time used on assignments.
total_time_quiz	Total time used on quizzes.
total_time_forum	Total time used on forum.
mark	Final mark the student obtained in the course.

Experimental Results

- Secondly, we have discretized (to split the numerical data into categorical classes that are easier for the teacher to understand) all the numerical values of the summary table into a new summarization table.
  - We have applied the **manual method** (in which you have to specify the cut-off points) to the mark attribute (from 0 to 10). We have used four intervals and labels (**FAIL**: if value is <5; **PASS**: if value is >=5 and <7; **GOOD**: if value is >=7 and <9; and **EXCELLENT**: if value is >=9).
  - We have applied the automatic **equal-width method** to all the other attributes with three intervals and labels (**LOW**, **MEDIUM** and **HIGH**).

Experimental Results

- Then, we have exported both versions of the summary table (with numerical and categorical values) to text files with KEEL format.
- Next, we have made partitions (10-fold) of whole files into pairs of training and test files.
- We also take into consideration the problem of **learning from imbalanced data** (in which some classes differ significantly from others with respect to the number of instances available). We have used **random over-sampling**, that consisting of copying randomly chosen instances of minority classes in the dataset until all classes have the same number of instances, and we have used the **geometric mean** to measure the quality of the induced classifiers.

Experimental Results

- Finally, we have used three sets of 10-fold data files: the original numerical data, the categorical data and the numerical rebalanced data.
- We have carried out one execution with all the determinist algorithms and 5 executions with the nondeterministic algorithms.
- We have calculated the **global percentage of the accuracy** rate (correctly classified) and the **geometric means**.
- We have used the 25 previous classification algorithms (implemented in KEEL).

Experimental Results

Method	Algorithm	Numerical data	Categorical data	Rebalanced data
Statistical Classifier	ADLinear	59.82 / 0.00	61.66 / 0.00	59.82 / 0.00
Statistical Classifier	PolQuadraticLMS	64.30 / 15.92	63.94 / 18.23	54.33 / 26.23
Statistical Classifier	Kernel	54.79 / 0.00	56.44 / 0.00	54.34 / 0.00
Statistical Classifier	KNN	59.38 / 10.15	59.82 / 7.72	54.34 / 10.21
Decision Tree	C45	64.61 / 41.42	65.29 / 18.10	53.39 / 9.37
Decision Tree	CART	**67.02 / 39,25**	**66.86 / 24,54**	47.51 / 34,65
Rule Induction	AprioriC	60.04 / 0.00	59.82 / 0.00	**61.64 / 0.00**
Rule Induction	CN2	64.17 / 0.00	63.47 / 3.52	50.24 / 15.16
Rule Induction	Corcoran	62.55 / 0.00	64.17 / 0.00	61.42 / 0.00
Rule Induction	XCS	62.80 / 0.00	62.57 / 0.00	60.04 / 23.23
Rule Induction	GGP	65.51 / 1.35	64.97 / 1.16	52.91 / 12.63
Rule Induction	SIA	57.98 / 0.00	60.53 / 0.00	56.61 / 15.41
Fuzzy Rule Learning	MaxLogitBoost	64.85 / 0.00	61.65 / 0.00	62.11 / 8.83
Fuzzy Rule Learning	SAP	63.46 / 0.00	64.40 / 0.00	47.23 / 3.20
Fuzzy Rule Learning	AdaBoost	62.33 / 0.00	60.04 / 0.00	50.47 / 0.00
Fuzzy Rule Learning	LogitBoost	61.17 / 13.05	63.27 / 4.64	55.70 / 13.95
Fuzzy Rule Learning	GAP	65.99 / 0.00	63.02 / 0.00	52.95 / 26.65
Fuzzy Rule Learning	GP	63.69 / 0.00	63.03 / 0.00	53.19 / 11.97
Fuzzy Rule Learning	Chi	57.78 / 10.26	60.24 / 0.00	41.11 / 14.32
Neural Networks	NNEP	65.95 / 0.00	63.49 / 0.00	54.55 / 12.70
Neural Networks	RBFN	55.96 / 3.23	54.60 / 0.00	37.16 / 4.00
Neural Networks	RBFN Incremental	53.65 / 9.87	58.00 / 14.54	30.31 / 18.32
Neural Networks	RBFN Decremental	50.16 / 3.95	53.44 / 5.61	35.32 / 8.41
Neural Networks	GANN	60.28 / 0.00	61.90 / 4.82	53.43 / 17.33
Neural Networks	MLPerceptron	63.91 / 9.65	61.88 / 4.59	53.21 / 17.16

Experimental Results: Comprehensibility

- It is very important that the model obtained to be user friendly, so that teachers can interpret it in order to can make decisions. Some obtained models are more interpretable than others:
  - **Decision trees** are considered easily understood models (they can be transformed into a set of IF-THEN rules).
  - **Rule induction algorithms** are also considered to produce comprehensible models (they discover IF-THEN rules).
  - **Fuzzy rule algorithms** obtain IF-THEN rules that use linguistic terms (more interpretable by humans).
  - **Statistical methods** and **neural networks** are usually considered to be black-box mechanisms.

Experimental Results: Comprehensibility

- **Example Decision tree model obtained:**

```
IF ( n_quiz_a <= 7 ) THEN
{
  IF ( total_time_forum <= 1494 ) THEN { mark = FAIL}
  ELSEIF ( total_time_forum > 1494 ) THEN { mark = PASS }
}
ELSEIF (n_quiz_a > 7) THEN
{
  IF ( n_assignment <= 10 ) THEN { mark = PASS}
 ELSEIF ( n_assignment > 10 ) THEN { mark =  EXCELLENT }
}
ELSEIF ...
```

Experimental Results: Comprehensibility

- **Example Rule Induction model obtained:**

IF n_assignment < 6 THEN mark = FAIL
IF n_assignment > 10 AND n_read > 9 THEN mark = EXCELLENT
IF course = 29 AND n_quiz_a = 0 THEN mark = FAIL
IF course = 110 AND n_quiz_a > 7 THEN mark = GOOD

Conclusions

- We have shown that some algorithms improve their classification performance when we apply such preprocessing tasks as discretization and rebalancing data, but others do not.
- We have also indicated that a good classifier model has to be both accurate and comprehensible for instructors.
- Finally, we want also test the use of the tool by teachers in real pedagogical situations in order to prove on its acceptability.

The end

Thanks for your interest.

Questions? Comments?