Subject

For large simulations, it is more convenient to use BATCH mode capabilities of Tanagra rather than opening interactive session. This is the case for instance when we compare the performance of various algorithms on the same dataset; when we try to find automatically the best parameters for a learning method; when we repeat the same treatment on different datasets, etc. In these contexts, it is more useful to save the diagrams in text mode (.TDM file format). It will be easier to handle it outside TANAGRA, with a text editor for instance.

Organizing experiments for feature selection process

We want to evaluate the efficiency of the FCBF feature selection method[1] in a supervised learning framework. This approach handles only discrete descriptors. It tries to detect the descriptors which are the most correlated with the class attribute and the less correlated among them.

We treat three datasets from the UCI server in this tutorial: VOTE, KR-VS-KP, and SPLICE (http://kdd.ics.uci.edu/). In addition to the original descriptors, we add two kinds of predictive attributes: some are randomly generated; some are generated in a way that they are correlated to the existing descriptors.

We want to compare the performances of the naïve bayes classifier[2] with and without the feature selection process. We know that the naïve bayes classifier is highly sensitive to irrelevant features. The goal of this tutorial is to evaluate the efficiency of the FCBF feature selection method in this context.

Main steps of the experiments

Specifying manually the diagram

In a first step, we create manually the diagram on the VOTE dataset. Then we save the diagram in a ".TDM" file format. It is a text file where all the components with the associated settings are described.

We see below the diagram and the related ".TDM" file. We can open it in a standard text editor if we want to see its contents.

[1] http://www.public.asu.edu/~huanliu

[2] http://en.wikipedia.org/wiki/Naive_Bayes_classifier

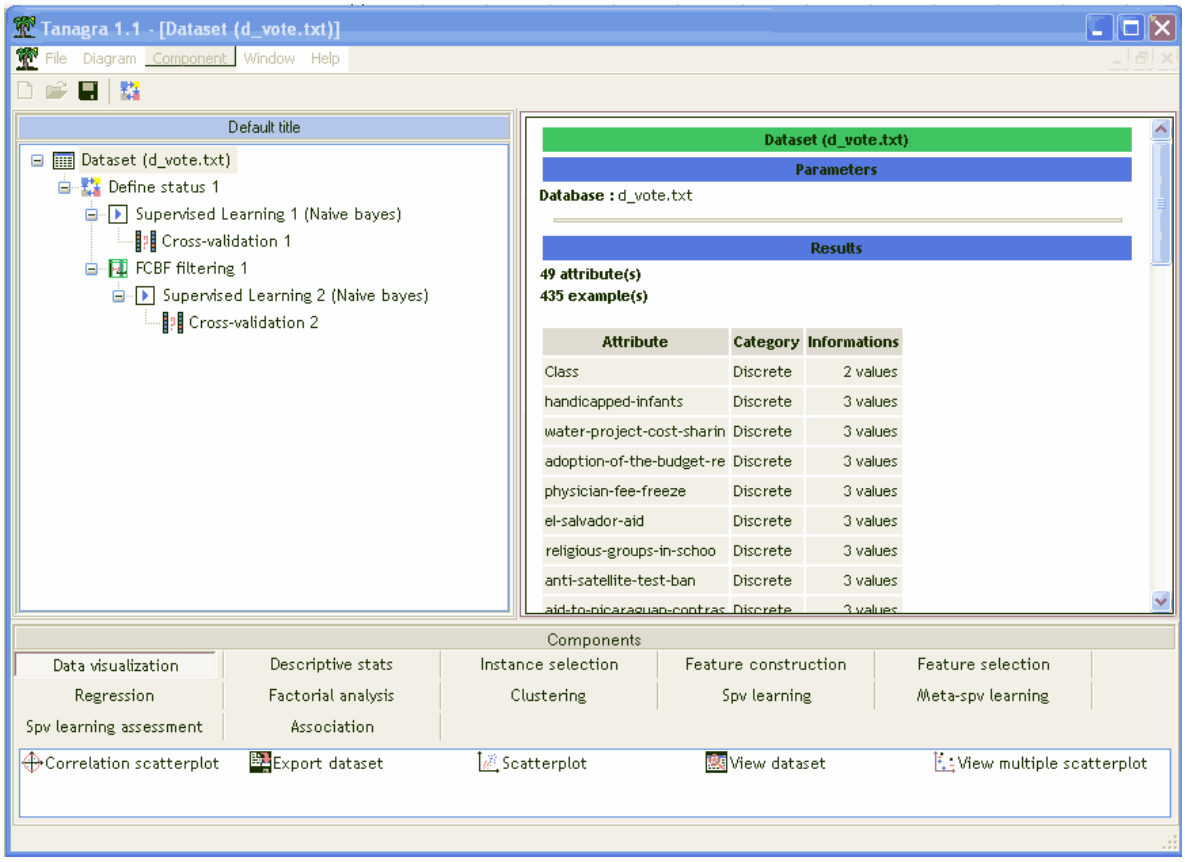

Figure 1 : Comparing the error rate of the naïve bayes classifier with and without FCBF

```
[Diagram]
Title=Default title
Database=d_vote.txt

[Dataset]
MLClassGenerator=TMLGenDataset
successors=1
succ_1=Define status 1

[Define status 1]
MLClassGenerator=TMLGenFSDefStatus
target_count=1
target_1=Class
input_count=48
input_1=handicapped-infants
input_2=water-project-cost-sharin
input_3=adoption-of-the-budget-re
input_4=physician-fee-freeze
input_5=el-salvador-aid
```

```
input_6=religious-groups-in-schoo
input_7=anti-satellite-test-ban
input_8=aid-to-nicaraguan-contras
input_9=mx-missile
input_10=immigration
input_11=synfuels-corporation-cutb
input_12=education-spending
input_13=superfund-right-to-sue
input_14=crime
input_15=duty-free-exports
input_16=export-administration-act
input_17=noise1
input_18=noise2
input_19=noise3
input_20=noise4
input_21=noise5
input_22=noise6
input_23=noise7
input_24=noise8
input_25=noise9
input_26=noise10
input_27=noise11
input_28=noise12
input_29=noise13
input_30=noise14
input_31=noise15
input_32=noise16
input_33=corr1
input_34=corr2
input_35=corr3
input_36=corr4
input_37=corr5
input_38=corr6
input_39=corr7
input_40=corr8
input_41=corr9
input_42=corr10
input_43=corr11
input_44=corr12
input_45=corr13
input_46=corr14
input_47=corr15
```

```
input_48=corr16
illus_count=0
successors=2
succ_1=Supervised Learning 1 (Naive bayes)
succ_2=FCBF filtering 1

[Supervised Learning 1 (Naive bayes)]
MLClassGenerator=TMLGCompOneInstance
embedded_spv=1
embedded_section=Supervised Learning 1 (Naive bayes)--Naive bayes
successors=1
succ_1=Cross-validation 1

[Supervised Learning 1 (Naive bayes)--Naive bayes]
MLClassGenerator=TMLGCompNaiveBayes

[Cross-validation 1]
MLClassGenerator=TMLGenCompAssesCV
isSaveResults=1
results_filename=experiments.txt
nb_repetitions=5
nb_folds=2
successors=0

[FCBF filtering 1]
MLClassGenerator=TMLGenFSFcbf
delta=0
successors=1
succ_1=Supervised Learning 2 (Naive bayes)

[Supervised Learning 2 (Naive bayes)]
MLClassGenerator=TMLGCompOneInstance
embedded_spv=1
embedded_section=Supervised Learning 2 (Naive bayes)--Naive bayes
successors=1
succ_1=Cross-validation 2

[Supervised Learning 2 (Naive bayes)--Naive bayes]
MLClassGenerator=TMLGCompNaiveBayes

[Cross-validation 2]
MLClassGenerator=TMLGenCompAssesCV
```

```
isSaveResults=1
results_filename=experiments.txt
nb_repetitions=5
nb_folds=2
successors=0
```

The TDM file is similar to the INI file format. Each section is related to a component. The settings corresponds to the pairs "name of parameter" = "value".

The cross-validation components play an important role in our analysis. They compute the error rate of the supervised learning method and they write the results in an output file, "experiments.txt" here. The results are thus automatically collected during the execution of the branch of the diagram. We note that Tanagra performs a "true" resampling process i.e. the whole branch is executed for each subsample, if there are other treatments in addition to the supervised learning process, they are launched also.

For our analysis, CROSS VALIDATION 1 computes the error rate of the naive bayes classifier without the feature selection process; CROSS VALIDATION 2, at the opposite, incorporates the FCBF approach. If the selection process is effective, it is expected that the error rate calculated in the second case is lower.

We define the diagrams for each dataset to analyze using the same model (kr-vs-kp.tdm and splice.tdm).

Creating the batch file

The second step is the creation of the batch file which gathers the treatments of the three datasets. Under Windows, we create a very basic ".BAT" file. It is similar to the example below. We must to check the path of the Tanagra executable file (Tanagra.exe).

We set these commands in the "experiments.bat" file (you can specify other .BAT file name of course).

```
d:\temp\exe\tanagra vote.tdm
d:\temp\exe\tanagra splice.tdm
d:\temp\exe\tanagra kr-vs-kp.tdm
```

Reading the results

After we launch the ".BAT" file. A report (HTML format) is automatically generated. We can open it with a browser.

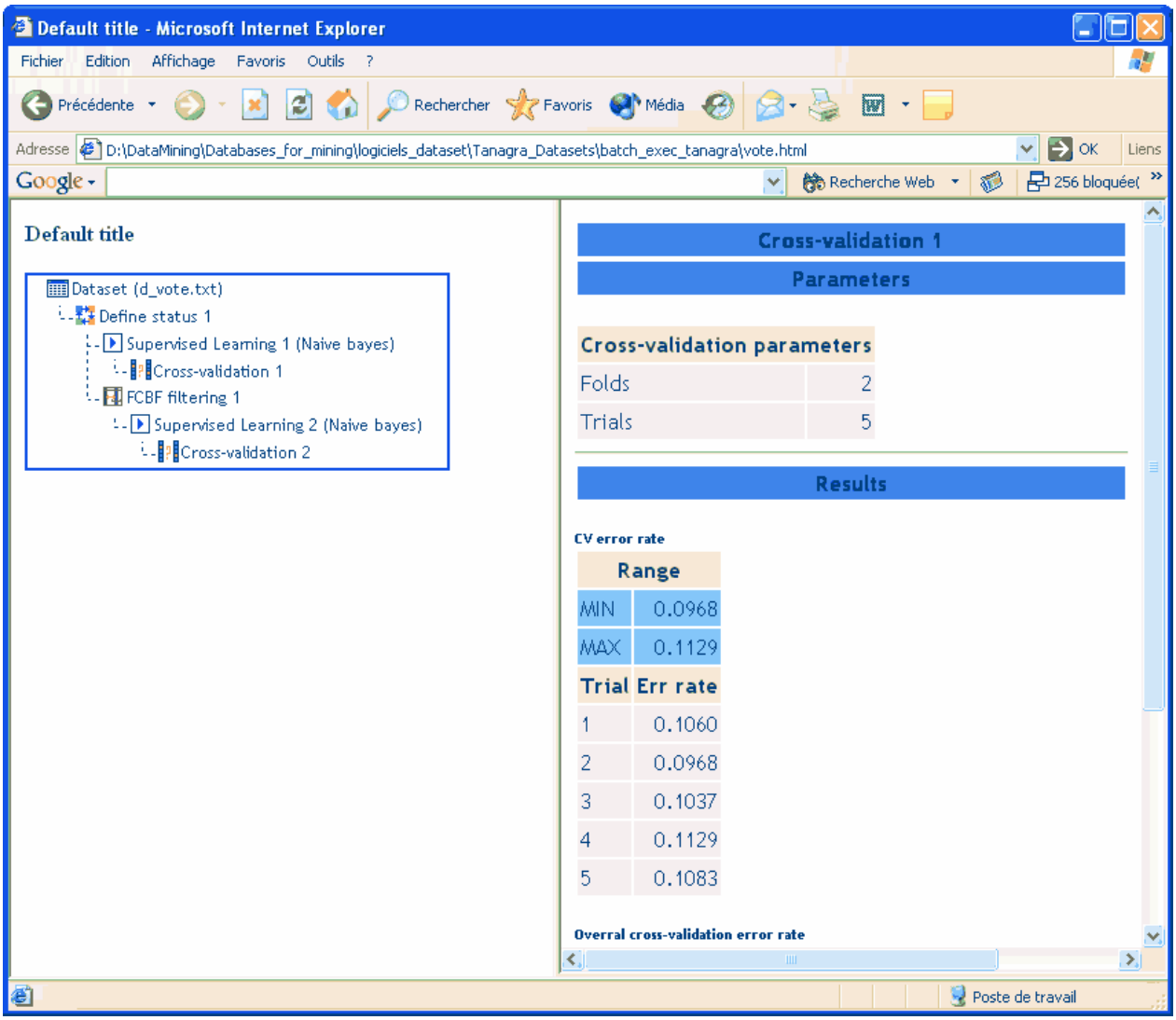

Figure 2 : Report (HTML) for the "vote.tdm" diagram

About our analysis, we have, in addition, the results of the cross-validation in a text file which is automatically generated by the component. Each line corresponds to a measurement of the error rate: CROSS-VALIDATION is the error rate of the naïve bayes classifier without the FCBF feature selection process; CROSS VALIDATION 2, incorporates FCBF.

Diagram	Dataset	Date	Component	Error rate
vote.tdm	d_vote.txt	15/10/2004 19:47	Cross-validation 1	0.094931
vote.tdm	d_vote.txt	15/10/2004 19:47	Cross-validation 2	**0.058986**
splice.tdm	d_splice.txt	15/10/2004 19:47	Cross-validation 1	0.065705

splice.tdm	d_splice.txt	15/10/2004 19:47	Cross-validation 2	**0.04721**
kr-vs-kp.tdm	d_kr-vs-kp.txt	15/10/2004 19:47	Cross-validation 1	0.133229
kr-vs-kp.tdm	d_kr-vs-kp.txt	15/10/2004 19:47	Cross-validation 2	**0.07985**

We note from left to right: the filename of the diagram, the data file used, date and time of execution of the component, the name of the component, and the error rate that has been computed.

On our datasets, that are intentionally selected, we observe that the FCBF feature selection process improves the accuracy of the naive bayes classifier. But, this kind of results is not always true whatever the dataset used.

Extensions

We can easily imagine the possible extensions of such a tool. We could, for example, include in the list of results the number of selected descriptors. In actually, there are as many possible extensions that are of concern to researchers. These developments are specific, accessing to source code of Tanagra will allow everyone to program its procedures.

Similarly, this tool will be very powerful if we can automatically generate the TDM files. This is particularly interesting for example when one wants to vary the parameter of a method in order to determine its optimal value. The TDM format is fairly accessible. Writing a small program that automatically generates this kind of file is rather easy.