TP 1-2: Introduction to KNIME and Association Rules

The purpose of this lab session is to familiarize ourselves with KNIME, a data mining and visualization tool written in Java, as well as to experiment with algorithms for generating frequent itemsets and association rules.

1 Knime Overview

KNIME is a free tool for creating data mining workflows. You can find several tutorials on the web site `https://www.knime.org/`.

KNIME's interface is divided into 5 main parts as shown in the figure below.

— Part 1 allows browsing between elements in the current workflow.
— Part 2 allows browsing between the "nodes" of KNIME, i.e. the predefined bricks that can be used to create a workflow. Simply drag a node from it into part 3.
— Part 3 is the workflow editor, in which you can add nodes and draw links between these nodes.
— Part 4 describes each of the nodes, notably the inputs and outputs of a node and its configuration options.
— Finally, part 5 is the KNIME console, which shows the errors that may occur during the execution of the workflow.

In the next section, we will see how to create a simple workflow using the dataset `iris.csv`, describing flowers (length and width of petal and sepal).

2 My First Workflow

In this section you are going to create a first workflow with the "iris" dataset, so as to discover the most common nodes that are used for data mining and data visualization.

Task 2.1 *Create a new KNIME workflow with the* `File / New...` *menu or the top-left icon. Name it "iris".*

The general method to follow for performing the following tasks is to :

1. select a node in the `Node repository`,
2. add the node to the workflow by drag and drop (several instances of the same node can be created),
3. link the new node to the existing ones if necessary via the nodes' "ports" (inputs/outputs),
4. configure the node with the context menu command `Configure...`,

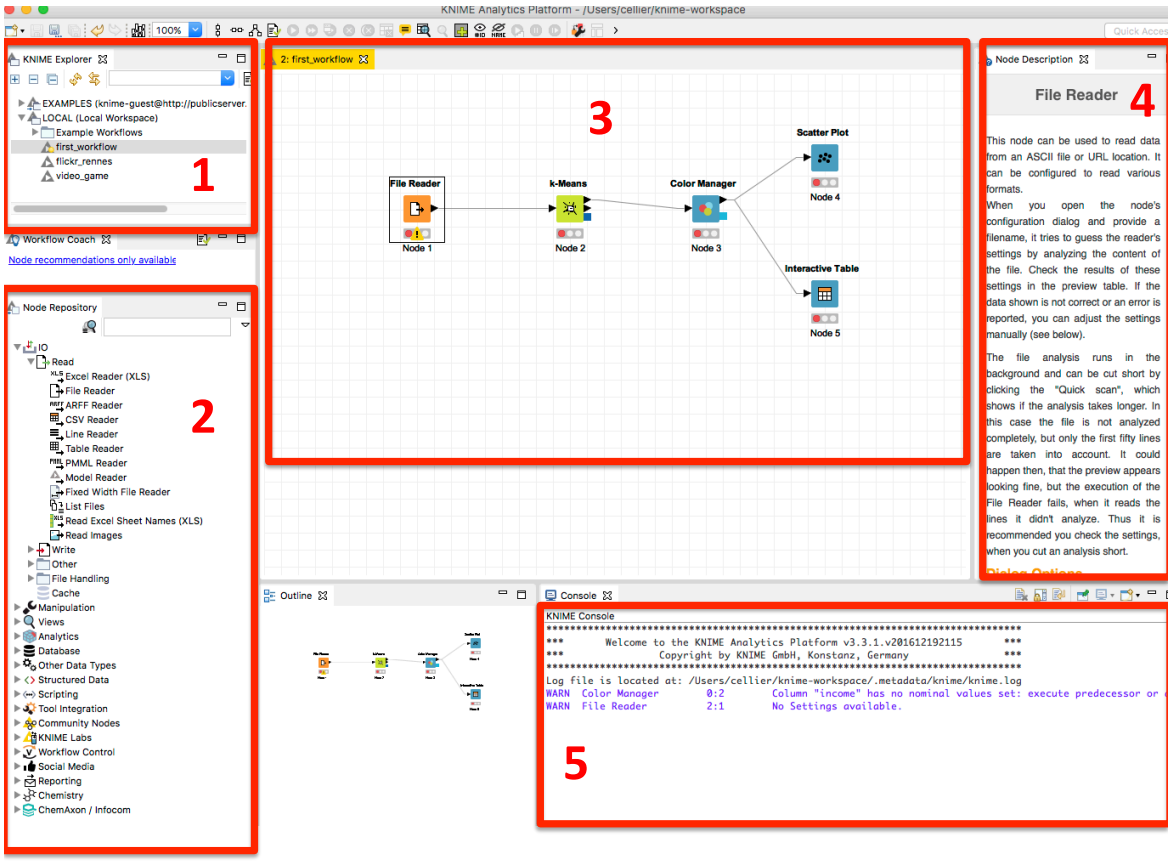

FIGURE 1 – Interface de KNIME

5. Execute the node with the context menu command `Execute` or the F7 key,
6. and, if defined, view the result with the context menu commands whose name begins with `View:` or placed at the bottom of the context menu.

Configuring, executing and visualizing the nodes as soon as they are placed ensures that the workflow conforms to what is expected and allows to detect errors quickly.

There are two ways to search for a node in the node repository : (a) by exploring the node hierarchy, in which nodes are classified by category (b) by entering keywords in the search field. Both ways can be combined. By selecting a node you can see its description in the rightmost part of the interface.

A CSV file containing the data set “iris” is available to you. It contains the description of 150 iris specimens (flowers) of 3 different species. Each description is composed of 4 numerical attributes (length and width of sepals and petals), and a 5th attribute which is the class of the example (i.e. the species of iris to which it belongs).

Task 2.2 *Find a node that allows to read the CSV file* `iris.csv` *and follow all the steps above. Clue : look among the input/output (I/O) nodes. Verify that the file has been correctly loaded with the visualization command* `File table`*.*

Before applying data mining algorithms it is useful to familiarize yourself with the data through simple statistics and visualizations.

Task 2.3 *Apply the* Statistics *node to the data table. Note the distinction between numeric and nominal attributes.*

When the data contains multiple classes, it is useful to associate a color to each class, so as to improve the data visualization.

Task 2.4 *Apply the* Color Manager *to the data table in order to associate a color to each class. The output of this node is a "colored" table.*

Task 2.5 *Apply the* Scatter Plot Matrix *and* Scatter Plot *nodes (from the* Views *node category) to the colored table to observe the correlations between classes and the combinations of pairs of numerical attributes. Play with the resulting views.*

Question 2.6 *Which combination of two attributes (between width/length of sepals/petals) best separates the 3 classes ? Can you state criteria allowing to predict the class of an iris from the values of these these two attributes ?*

Although it is part of the FST course rather than FSY, we will learn a decision tree from the data to automatically discover criteria predicting the class of an iris.

Task 2.7 *Apply the* Decision Tree Learner *to the colored data table to produce a decision tree. Compare the learned criteria to the ones you found.*

3 A First Example of Association Rules

In this section we will build a workflow to search association rules on golf-related data.

Task 3.1 *Create a new workflow and load the file* `weather.nominal.arff`*. It's an already-discretised version of the golf data.*

Task 3.2 *Add nodes to obtain statistics and visualizations of the data. Do not hesitate to play with the nodes of the* Views *category.*

Task 3.3 *Find the node allowing to compute association rules and try to apply it to the data. What is the problem ?*

As it is often the case in data mining, the data needs to be preprocessed according to the algorithm that is applied to it. Here, the association rules node needs a column of type *BitVector*, where each bit corresponds to an item. For this, there is the *Create Bit Vector* node in the same category as the association rules node. But before applying it, you must propositionalize the data, i.e., reduce the values to zeroes/ones.

Task 3.4 *Use the* One to Many *node to propositionalize the data table. Verify the result. Is it correct ? What is the transformation that is applied to the data ?*

Task 3.5 *Now apply the* Create Bit Vector *node to the propositionalized data. Observe the result and explain it.*

Task 3.6 *Now apply the association rule computation and play with the configuration options. Find a way to set the minimum support and the minimum confidence. Find a way to only generate frequent itemsets and not the rules.*

Following the propositionalization, the values have become column names, which can lead to confusion. For example, is the item `yes` about the presence of wind or about going to play ?

Task 3.7 *Find a way to rename the columns in the data to make the itemsets and rules more understandable. Hint : KNIME allows to insert nodes everywhere, not only at the end of a workflow !*

If the goal is to predict whether we can play golf or not based on the other criteria, then rules whose consequent is a value of `play` suffice.

Task 3.8 *Find a way to filter the list of rules to only keep the ones where the consequent is a value of* `play`*. Clue : the keyword is "filter".*

4 A Second Example of Association Rules

The file `adult.csv` contains data extracted from a U.S. population census. The initial purpose of this data is to predict whether someone earns more than 50,000 dollars per year. We will transform the data a bit before extracting association rules.

Task 4.1 *Create a new workflow and charge the* `adult.csv` *file as input data.*

Most of the data transformation nodes are located in the *Manipulation* category. There are nodes for filtering columns or rows, for applying transformations to the values of a column, to merge columns or rows, or to split them.

The data often contain unnecessary attributes : individual number, name, date of entry... It is possible to delete them manually, provided you know the domain. It is also possible to launch a data mining algorithm and look at the attributes that have been used : either they are relevant and it is important to keep them, or they are so related to the class that they alone carry the decision (think of an attribute that would be the copy of the class).

Task 4.2 *Show using attributes statistics that the attribute* `fnlwgt` *can be ignored. Delete it!*
Also delete the attributes `capital-gain` *and* `capital-loss`*.*

Some algorithms need discrete attributes to work (e.g., association rules, concept analysis), others only accept continuous attributes (e.g. neural networks, nearest neighbors). Others also indifferently accept attributes of both types.

Question 4.3 *Which of the remaining attributes need to be discretized ?*

Task 4.4 *Find and apply a node to discretize those attributes. Clue : the keyword is* bin*. Configure the node so as to split the value sets into 3 intervals. Look through the different configuration options and experiment with them.*

Now that the data are discretized, it is possible to generate association rules as we did in the previous section.

Task 4.5 *Complete the workflow so as to produce a set of association rules. Play with the different configuration options so as to find a set of "interesting" rules.*

Task 4.6 *Select classification rules, i.e., rules with the* `class` *attribute. Adapt the options so as to place the most important rules at the top of the table.*

Question 4.7 *What conclusions do you make from the results ?*