A Label Ranking extension for Weka

Alexander Balz

February 26, 2013

1 Introduction

This document describes a modification for the machine learning framework Weka[1], in which Weka will be able to process Label Ranking data sets and to predict rankings by using the currently implemented "*Ranking by Pairwise Comparison*" (RPC) algorithm[2] or the Decision Tree based method *TreeLabelRanking* [3].

In the following, the .xarff file type will be introduced, which simply is an extension of the existing .arff file type supported by Weka. Next, the differences to the original Weka preprocess panel are discussed. Afterwards, it is shown how label rankings can be predicted from a given data set and which statistics are returned.

2 File format of Label Ranking data sets

Label ranking data sets can be saved in .xarff file format. The name already implies that it is an extension of Weka's regular .arff files[1]. By giving an .xarff file fragment, we will explain the differences of this file type to .arff files:

```
1   @relation example
2   @attribute A1 NUMERIC
3   @attribute A2 NUMERIC
4   @attribute A3 NUMERIC
5   @attribute A4 RANKING {L1,L2,L3,L4,L5}
6
7   @data
8   -1.337785, 1.038478, 1.856137, L5>L4>L3>L2>L1
9   -1.237785, 1.033796, 1.956137, L4>L2>L3
10  -1.327803, 1.052523, 0.982119, L5>L4>L1|L3>L1|L2>L4
11  -0.124868, 9.354512, 1.112111, 'L1 > L2 > L3'
                    ⋮
```

In line 5, a ranking attribute has been defined. Its structure is quite similar to nominal

attributes. A ranking attribute needs to have a unique attribute name, followed by the keyword 'RANKING'. After this key, a list of label names is following. Label rankings must only consist of those labels included into the list. For each instance, pairwise comparisons, partial or complete rankings can be assigned. In more detail: Labels are assembled to a Label Ranking by using the '>' character. So Label1>Label2 means that label 1 is preferred over label 2. It is not permitted to state one single label inside of an instance, so there have to be at least two labels separated by the '>' sign. If you wish to use whitespaces inside of the rankings, please make sure that they must be written with apostrophes surrounding, like 'L1 > L2 > L3'. In case of non-existant whitespaces, there is no need to put rankings between apostrophes.

Some examples are given from line 8 on. First, a complete ranking is shown - consisting of all declared labels. Nevertheless, label rankings do not necessarily have to include all labels available in the ranking attribute, so that such rankings like in line 9 are permitted, too. Furthermore, it is possible to insert partial label rankings (line 10). In order to do so, several rankings are separated by the '|' symbol. In line 11, there is an example for a ranking containing whitespaces.

3 Processing .xarff files

.xarff files can be loaded - like the .arff files inside of the preprocess panel of Weka's explorer. After having selected the right file extension and loading a desired file, Weka will display attribute statistics as usual. When clicking on the ranking attribute, one can see a matrix inside of the 'Selected attribute' panel on the right, containing the amount of every pairwise label lanking that occurs inside of the data set. Under this matrix, a visualization of it is displayed. A white field means that no or hardly any pairwise ranking is existant into the loaded file. The darker a matrix field appears, the more pairwise rankings are contained in the data set.
So the preprocess panel could look like in figure 1.

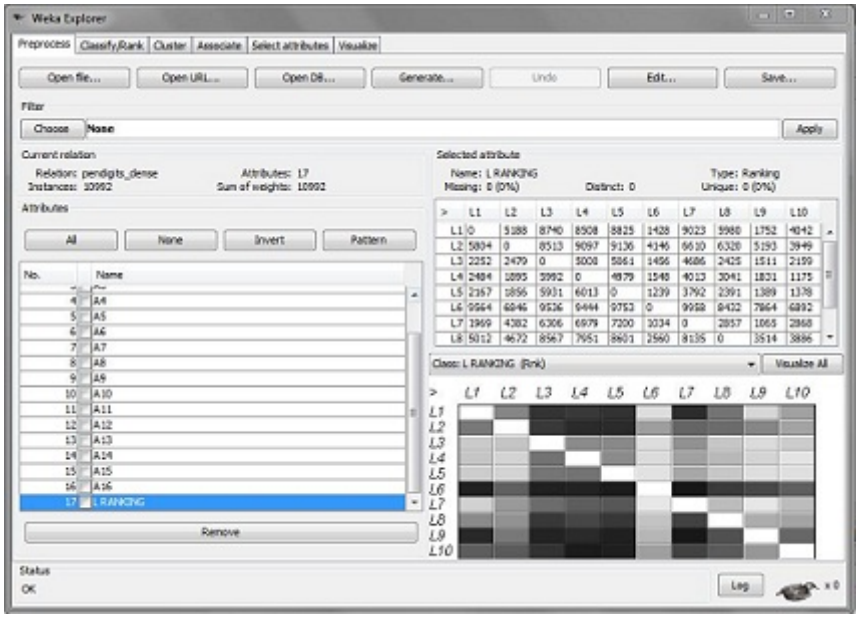

Figure 1: The Preprocess Panel including Label Ranking data

4 Editing data sets manually

In Wekas preprocess panel, it is possible to change entries of the read data set by clicking on the "Edit" button. In order to do so, double click on the value that should be changed and type in the desired value or ranking.
Complete and incomplete rankings may be written into the fields for ranking attributes, see figure 2.

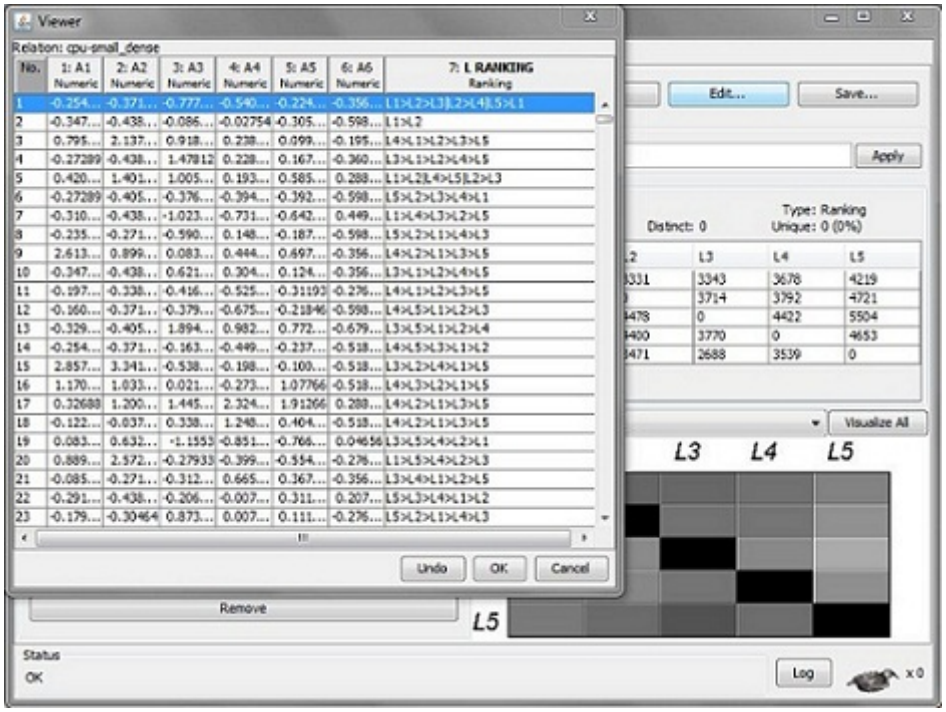

Figure 2: The editing viewer

5 Predicting Label Rankings

After having loaded the data set,you are able to make a prediction over it. A ranking algorithm can be chosen in the "Classify/Rank" panel of the Weka explorer. Within the dialog for choosing a Classifier, we can find a folder named labelranking. Inside of it, the ranking by pairwise comparison (RPC) algorithm can be found [2]. This algorithm has several parameters which can be changed by clicking on its name in the Classifier/Label ranker field (figure 3).

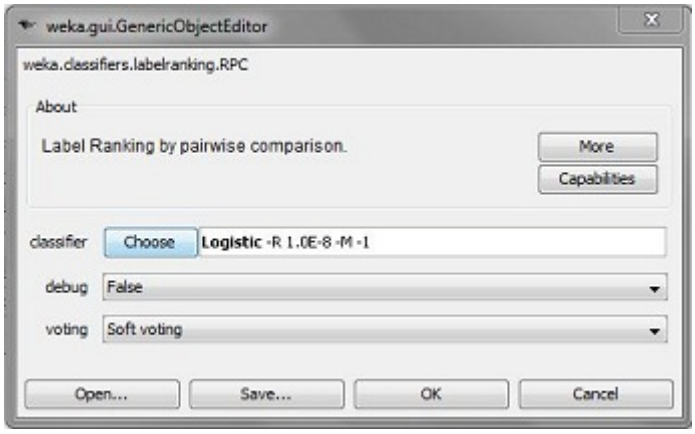

Figure 3: The parameters of RPC

RPC needs a base classifier to work properly. The default base classifier is logistic regression. Next, the debug parameter is contained in most classifiers inside of Weka and provides additional information on the console. RPC does a voting on the base classifiers' results. There are two different ways of voting: The first way of voting is *binary voting*, in which the base classifiers return their predictions for every pairwise label ranking. The values of those predictions may be 0 or 1. Aftwerwards, the labels with the highest amount of votes are proposed as final prediction. The second voting scheme is called *soft voting*. In this voting method, the base classifiers return a real number between 0 and 1 and including for each pairwise ranking. The closer this value to 1, the stronger it is taken into consideration in the voting process. For more details, see [2].
The default value of this parameter is soft voting. The results after a run of the RPC algorithm look similar to figure 4.
Some ranking measures have been implemented to evaluate the quality of prediction. These are Spearman footrule, Spearman rank correlation and Kendall's tau. Please note that those measures will only be shown if all rankings inside of the data set are complete rankings over a subset of the data set's available labels. Thus, if one or several partial rankings appear inside of the data set, the aforementioned measures will not be calculated.
Additionally, a matrix is computed which shows us the relative error of all pairwise label rankings. All pairwise rankings have been derived of the data set and are compared to those of the predictions. Correct and incorrect predictions are summarized into the matrix, separated by the '|' sign. Below, the summed up correct/incorrect values are returned. Unlike the ranking measures, this matrix will be shown even if partial rankings occur inside of the instances.
It is not recommended to use the functions of cluster and associate panels on label ranking data, because no algorithms are provided for this use yet and the existing ones

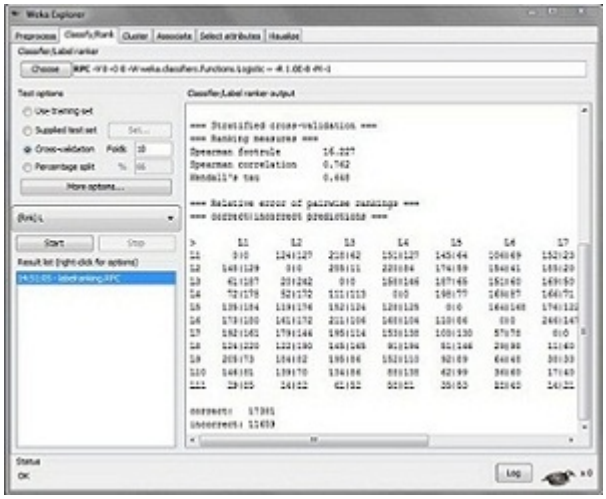

Figure 4: The Classifier Panel after RPC has been executed

inside of these panels will mostly cause errors when trying to evaluate them on Label Ranking data sets.
Using the tree label ranking (TLR) algorithm works quite similar to RPC, but TLR is no meta-classifier, so choosing a base classifier is not necessary.
The output has been slightly extended. A summary of the generated tree will be shown after the classifying process in figure 5.

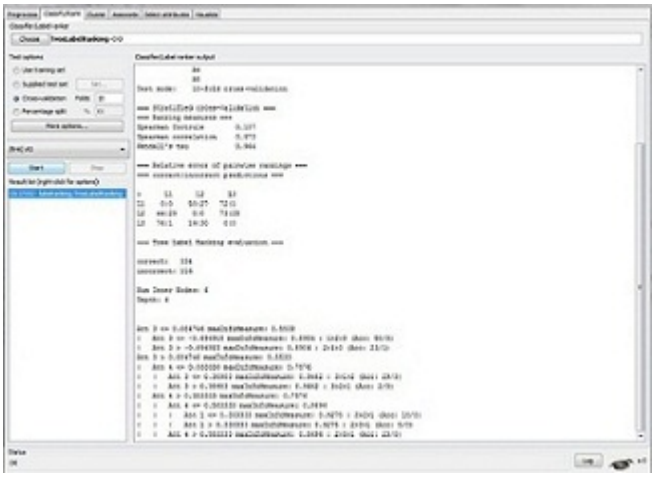

Figure 5: The Classifier Panel with further information about the TLR-generated tree.

References

[1] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, Ian H. Witten (2009); *The WEKA Data Mining Software: An Update;* SIGKDD Explorations, Volume 11, Issue 1.

[2] Weiwei Cheng, Prof. Dr. Hüllermeier, Johannes Fürnkranz, Klaus Brinker *Label ranking by learning pairwise preferences*, Elsevier, 2008

[3] Weiwei Cheng, Jens Hühn, Prof. Dr. Hüllermeier *Decision Tree and Instance-Based Learning for Label Ranking*