WEKA-LR: A Label Ranking Extension for WEKA

Alexander Balz and Robin Senge
Philipps-Universität Marburg

June 29, 2011

This document describes an extension of the machine learning framework WEKA [1], which supports the setting of *label ranking*.

1 File format of label ranking data sets

Label ranking data sets can be saved in .xarff file format, an extension of WEKA's regular .arff files. We explain this format by means of an example:

```
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 is defined. The structure of such attributes is quite similar to nominal attributes. A ranking attribute needs to have a unique attribute name, followed by the keyword 'RANKING' and a list of label names. Label rankings must only consist of those labels included in this list.

For each training instance in the data set (represented by three numeric attributes in our example), information about the ranking of labels can be specified as follows: as complete rankings including all labels (line 8), an incomplete rankings including a

subset of the labels (line 9), or in the form of pairwise comparisons between labels (line 10); obviously, the first and the third option are special cases of the second one. Incomplete rankings and pairwise comparisons can be combined by means of the '|' symbol, thereby enabling the specification of partial order relations.

A preference (order information) between labels is expressed in terms of the '>' character, i.e., $L1 > L2$ means that label $L1$ is preferred to label $L2$. The use of blanks inside a ranking (line 11) is allowed, provided the ranking is put in apostrophes, e.g., '$L1 > L2 > L3$'. In case no blanks are used, apostrophes can be omitted.

2 Processing .xarff files

Just like .arff files, .xarff files can be loaded inside the preprocessing panel of WEKA's explorer. After having selected the right file extension and loading the desired file, WEKA will display attribute statistics as usual. When clicking on the ranking attribute, a matrix inside the 'Selected attribute' panel is displayed on the right. For each pair of labels, this matrix informs about the number of times a preference of the former label over the latter label is observed in the entire data set. This matrix is also visualized graphically, with frequencies represented by levels of gray (the lighter, the more frequent); see Fig. 1 for an example.

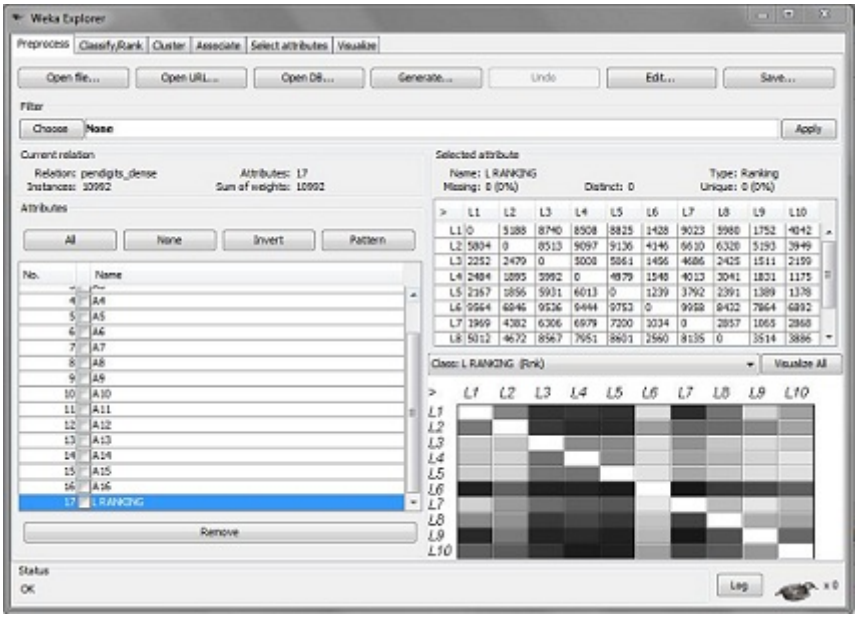

Figure 1: The Preprocess panel including label ranking data.

3 Editing data sets manually

In WEKA's preprocessing panel, it is possible to modify entries in the current 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 Fig. 2.

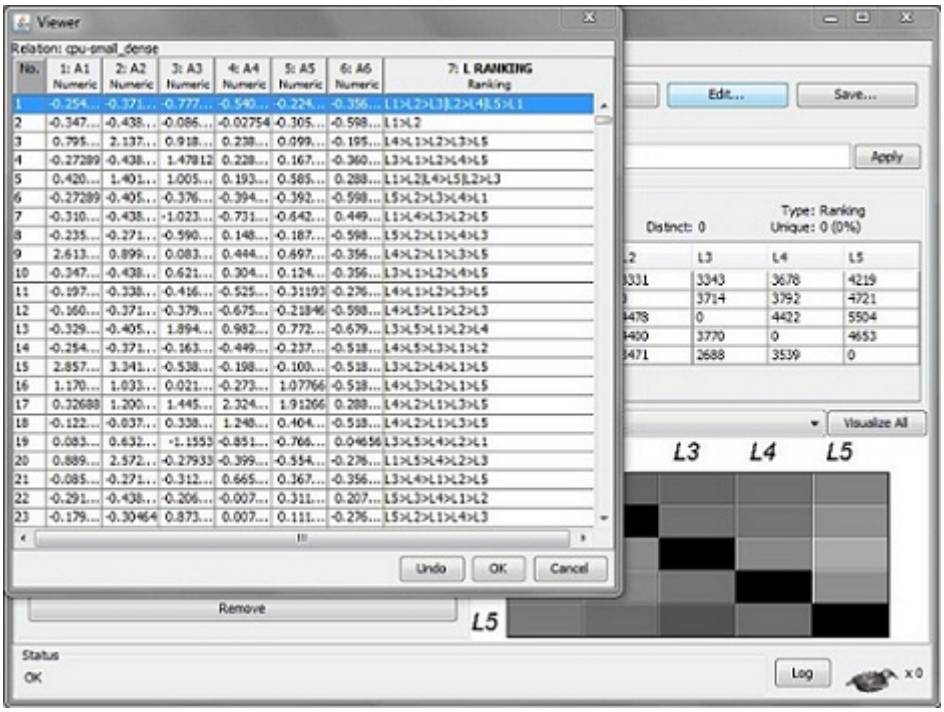

Figure 2: The editing viewer

4 Predicting label rankings

After having loaded the data set, it is possible to train a label ranking model and test its performance. A label ranking algorithm can be chosen in the 'Classify/Rank' panel of the WEKA explorer. Within the dialog for choosing a classifier, there is a folder named 'labelranking'. The parameters of an algorithm can be changed by clicking on its name in the Classifier/Label ranker field (Fig. 3).

Training a model or testing its performance by means of a cross-validation study is done as usual. However, what is of course different in comparison to classification is

the evaluation of predictions. Currently, the following rank correlation measures are implemented: Spearman footrule, Spearman rank correlation and Kendall's tau. These measures, which are all scaled between -1 (worst) and $+1$ (best), are computed on the test data, unless this data contains instances the preference information of which is only specified in the form of a partial order relation. More specifically, the requirement is that the preference information is equivalent to a total order of a subset of the label set; the evaluation is then weighted by the fraction of labels that are ranked.

For example, suppose the label set is given by $\{L1, L2, L3, L4\}$. The preference information $L2 > L1$, $L1 > L4$, $L2 > L4$ is then equivalent to $L2 > L1 > L4$. In order to evaluate the prediction $L1 > L3 > L2 > L4$, the latter would first be reduced to $L1 > L2 > L4$, because $L3$ does not occur in the ground truth. This ranking would then be compared to the ranking $L2 > L1 > L4$ in terms of Spearman and Kendall. Moreover, when computing the average performance on the whole data set, this evaluation would be weighted by 3/4. On the other hand, the preference information $L2 > L1$, $L3 > L4$ (and hence the whole data set) would not be evaluated, because it is not equivalent to a total order.

Additionally, a matrix is computed showing the performance separately for all pairs of labels, i.e., this matrix compares the pairwise preferences as (implicitly or explicitly) specified in the test data with the corresponding predictions. For each pair of labels, the number of correct and incorrect predictions is separated by the '|' sign. The total number of correct/incorrect comparisons is determined by summing over all pairs and shown beneath the matrix. Unlike the ranking measures, this information is always provided, even if the test data contains partial order relations.

Applying the functions in the Cluster and Associate panels to label ranking data is not recommended, because these functions have not yet been extended to the label ranking setting any may hence cause errors.

5 Label ranking algorithms

Currently, two label ranking algorithms are implemented in WEKA-LR: Ranking by pairwise comparison (RPC) and label ranking trees (LRT). If you implement your own algorithms and want them to be included in WEKA-LR, please contact Robin Senge (senge@mathematik.uni-marburg.de).

RPC [2] is a meta-learner that needs a binary classifier as a base learner; by default this is logistic regression. The predictions of the base classifier, one for each pair of labels, are aggregated by means of a voting strategy. Here, the default is *soft voting*, which can be changed into *binary voting*. Running the RPC algorithm will produce results like those shown in Fig. 4.

LRT [3] works quite similar to RPC, but since LRT is not a meta-classifier, there is no need to choose a base classifier. As an additional feature of the output, the generated tree is shown graphically (see Fig. 5 for an example).

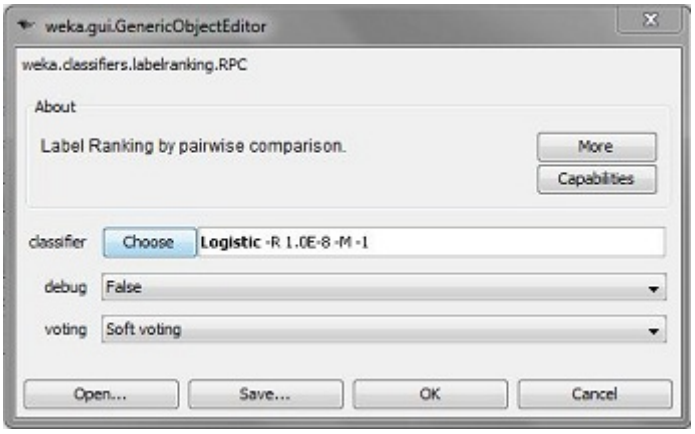

Figure 3: The configuration of RPC.

References

[1] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I.H. Witten. The WEKA data mining software: An update. *SIGKDD Explorations*, 11(1), 2009.

[2] E. Hüllermeier, J. Fürnkranz, W. Cheng, and K. Brinker. Label ranking by learning pairwise preferences. *Artificial Intelligence*, 172:1897–1917, 2008.

[3] W. Cheng, J. Hühn, and E. Hüllermeier. Decision tree and instance-based learning for label ranking. In *Proceedings ICML–2009, 26th International Conference on Machine Learning*, Montreal, Canada, 2009.

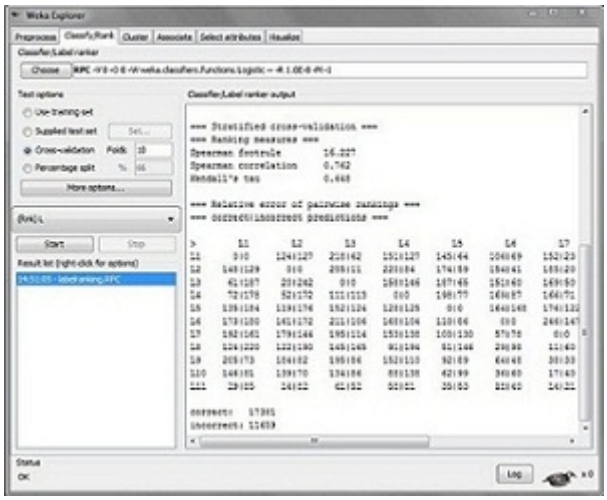

Figure 4: The Classifier panel after RPC has been executed.

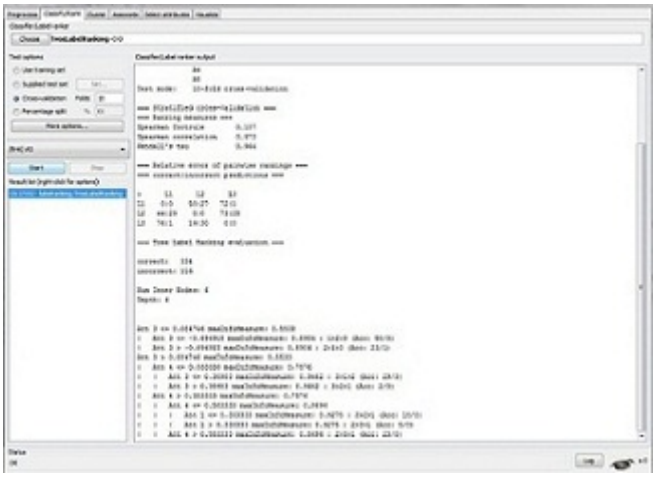

Figure 5: The Classifier panel with further information about the LRT-generated tree.