Distance-based decision tree algorithms for Label Ranking

Cláudio Rebelo de Sá[1,3], Carla Rebelo[3], Carlos Soares[2,3], and Arno Knobbe[1]

[1] LIACS Universiteit Leiden
[2] Faculdade de Engenharia, Universidade do Porto
[3] INESCTEC Porto, Porto, Portugal
c.f.de.sa@liacs.leidenuniv.nl, csoares@fe.up.pt,
a.j.knobbe@liacs.leidenuniv.nl

Abstract. The problem of Label Ranking is receiving increasing attention from several research communities. The algorithms that have developed/adapted to treat rankings as the target object follow two different approaches: distribution-based (e.g., using Mallows model) or correlation-based (e.g., using Spearman's rank correlation coefficient). Decision trees have been adapted for label ranking following both approaches. In this paper we evaluate an existing correlation-based approach and propose a new one, Entropy-based Ranking trees. We then compare and discuss the results with a distribution-based approach. The results clearly indicate that both approaches are competitive.

1 Introduction

Label Ranking (LR) is an increasingly popular topic in the machine learning literature [18,19,7,8,24]. LR studies a problem of learning a mapping from instances to rankings over a finite number of predefined labels. It can be considered as a natural generalization of the conventional classification problem, where only a single label is requested instead of a ranking of all labels [6]. In contrast to a classification setting, where the objective is to assign examples to a specific class, in LR we are interested in assigning a complete preference order of the labels to every example.

There are two main approaches to the problem of LR: methods that transform the ranking problem into multiple binary problems and methods that were developed or adapted to treat the rankings as target objects, without any transformation. An example of the former is the ranking by pairwise comparison of [11]. Examples of algorithms that were adapted to deal with rankings as the target objects include decision trees [23,6], naive Bayes [1] and k-Nearest Neighbor [3,6].

Some of the latter adaptations are based on statistical distribution of rankings (e.g., [5]) while others are based on rank correlation measures (e.g., [23,19]). In this paper we carry out an empirical evaluation of decision tree approaches for LR based on correlation measures and compare it to distribution-based approaches.

We implemented and analyzed the algorithm previously presented in [17]. We also propose a new decision tree approach for LR, based on the previous one, which uses information gain as splitting criterion. The results clearly indicate that both are viable LR methods and are competitive with state of the art methods.

2 Label Ranking

The Label Ranking (LR) task is similar to classification. In classification, given an instance x from the instance space $\mathbb{X}$, the goal is to predict the label (or class) λ to which x belongs, from a pre-defined set $\mathcal{L} = \{\lambda_1, \ldots, \lambda_k\}$. In LR, the goal is to predict the ranking of the labels in $\mathcal{L}$ that are associated with x [11]. A ranking can be represented as a total order over $\mathcal{L}$ defined on the permutation space Ω. In other words, a total order can be seen as a permutation π of the set $\{1, \ldots, k\}$, such that $\pi(a)$ is the position of λ_a in π.

As in classification, we do not assume the existence of a deterministic $\mathbb{X} \to \Omega$ mapping. Instead, every instance is associated with a *probability distribution* over Ω [6]. This means that, for each $x \in \mathbb{X}$, there exists a probability distribution $\mathcal{P}(\cdot|x)$ such that, for every $\pi \in \Omega$, $\mathcal{P}(\pi|x)$ is the probability that π is the ranking associated with x. The goal in LR is to learn the mapping $\mathbb{X} \to \Omega$. The training data is a set of instances $D = \{\langle x_i, \pi_i \rangle\}, i = 1, \ldots, n$, where x_i is a vector containing the values $x_i^j, j = 1, \ldots, m$ of m independent variables describing instance i and π_i is the corresponding target ranking.

Given an instance x_i with label ranking π_i, and the ranking $\hat{\pi}_i$ predicted by an LR model, we evaluate the accuracy of the prediction with a loss function on Ω. One such function is the number of discordant label pairs,

$$\mathcal{D}(\pi, \hat{\pi}) = \#\{(a,b)|\pi(a) > \pi(b) \wedge \hat{\pi}(a) < \hat{\pi}(b)\}$$

If normalized to the interval $[-1, 1]$, this function is equivalent to Kendall's τ coefficient [12], which is a correlation measure where $\mathcal{D}(\pi, \pi) = 1$ and $\mathcal{D}(\pi, \pi^{-1}) = -1$ (π^{-1} denotes the inverse order of π).

The accuracy of a model can be estimated by averaging this function over a set of examples. This measure has been used for evaluation in recent LR studies [6,21] and, thus, we will use it here as well. However, other correlation measures, like Spearman's rank correlation coefficient [22], can also be used.

2.1 Ranking Trees

One of the advantages of tree-based models is how they can clearly express information about the problem because their structure is relatively easy to interpret even for people without a background on learning algorithms. It is also possible to obtain information about the importance of the various attributes for the prediction depending on how close to the root they are used. The Top-Down Induction of Decision Trees (TDIDT) algorithm is commonly used for induction of decision trees [13]. It is a recursive partitioning algorithm that iteratively splits data into smaller subsets which are increasingly more homogeneous in terms of the target variable (Algorithm 1).

It starts by determining the split that optimizes a given splitting criterion. A split is a test on one of the attributes that divides the dataset into two disjoint subsets. For instance, given a numerical attribute x^2, a split could be $x^2 \geq 5$. Without a stopping criterion, the TDIDT algorithm only stops when the nodes are pure, i.e., when the value of the target attribute is the same for all examples in the node. This usually leads the algorithm to overfit, i.e., to generate models that fit not only to the patterns in the data but also to the noise. One approach to address this problem is to introduce a stopping criterion in the algorithm that tests whether the best split is significantly improving the quality of the model. If not, the algorithm stops and returns a leaf node. This node is represented by the prediction that will be made for new examples that fall into that node. This prediction is generated by a rule that solves potential conflicts in the set of training examples that are in the node. In classification, the prediction rule is usually the most frequent class among the training examples. If the stopping criterion is not verified, then the algorithm is executed recursively for the subsets of the data obtained based on the best split.

Algorithm 1 TDIDT algorithm

```
BestSplit = Test of the attributes that optimizes the SPLITTING CRITERION
if STOPPING CRITERION == TRUE then
  Determine the leaf prediction based on the target values of the examples in D
  Return a leaf node with the corresponding LEAF PREDICTION
else
  LeftSubtree = TDIDT($D_{\neg BestSplit}$)
  RightSubtree = TDIDT($D_{BestSplit}$)
end if
```

An adaptation of the TDIDT algorithm for the problem of learning rankings has been proposed [23], called Ranking Trees (RT) which is based on the clustering trees algorithm [2]. Adaptation of this algorithm for label ranking involves an appropriate choice of the splitting criterion, stopping criterion and the prediction rule.

Splitting Criterion The splitting criterion is a measure that quantifies the quality of a given partition of the data. It is usually applied to all the possible splits of the data that can be made based on individual tests of the attributes.

In RT the goal is to obtain leaf nodes that contain examples with target rankings as similar between themselves as possible. To assess the similarity between the rankings of a set of training examples, we compute the mean correlation between them, using Spearman's correlation coefficient. The quality of the split is given by the weighted mean correlation of the values obtained for the subsets, where the weight is given by the number of examples in each subset.

Table 1. Illustration of the splitting criterion

Attribute	Condition		Negated condition	
	values	rank corr.	values	rank corr.
x^1	a	0.3	b, c	-0.2
	b	0.2	a, c	0.1
	c	0.5	a, b	0.2
x^2	< 5	-0.1	≥ 5	0.1

The splitting criterion of ranking trees is illustrated both for nominal and numerical attributes in Table 1. The nominal attribute x^1 has three values (a, b and c). Therefore, three binary splits are possible. For the numerical attribute x^2, a split can be made in between every pair of consecutive values. In this case, the best split is $x^1 = c$, with a mean correlation of 0.5 for the training examples that verify the test and a mean correlation of 0.2 for the remaining, i.e., the training examples for which $x^1 = a$ or $x^1 = b$.

Stopping Criterion The stopping criterion is used to determine if it is worthwhile to make a split to avoid overfitting [13]. A split should only be made if the similarity between examples in the subsets increases substantially. Let $\mathcal{S}_{parent}$ be the similarity between the examples in the parent node and $\mathcal{S}_{split}$ the weighted mean similarity in the subsets obtained with the best split. The stopping criterion is defined in [17] as follows:

$$(1 + \mathcal{S}_{parent}) \geq \gamma(1 + \mathcal{S}_{split}) \tag{1}$$

Note that the significance of the increase in similarity is controlled by the γ parameter.

Prediction Rule The prediction rule is a method to generate a prediction from the (possibly conflicting) target values of the training examples in a leaf node. In RT, the method that is used to aggregate the q rankings that are in the leaves is based on the mean ranks of the items in the training examples that fall into the corresponding leaf. The average rank for each setting is $\overline{\pi\left(j\right)} = \sum_i \pi_i\left(j\right)/n$. The predicted ranking $\hat{\pi}$ will be the average ranking $\overline{\pi}$ after assigning ranks to $\overline{\pi\left(j\right)}$. Table 2 illustrates the prediction rule used in this work.

Table 2. Illustration of the prediction rule.

	λ_1	λ_2	λ_3	λ_4
π_1	1	3	2	4
π_2	2	1	4	3
$\overline{\pi}$	1.5	2	3	3.5
$\hat{\pi}$	1	2	3	4

2.2 Entropy Ranking Trees

Decision trees, like ID3 [15], use Information Gain (IG) as a splitting criterion to look for the best split points.

Information gain IG is a statistical property that measures the difference in entropy, between the prior and actual state relatively to a target variable [13]. In other words, considering a set S of size n_S, as entropy - H - is a measure of disorder, IG is basically how much uncertainty in S is reduced after splitting on attribute A:

$$IG\left(A,T;S\right)=H\left(S\right)-\frac{|S_1|}{n_S}H\left(S_1\right)-\frac{|S_2|}{n_S}H\left(S_2\right)$$

where $|S_1|$ and $|S_2|$ are the number of instances on the left side (S_1) and the number of instances on the right side (S_2), respectively, of the cut point T in attribute A.

Using the same tree generation algorithm, the TDIDT (Section 2.1), we propose an alternative approach of decision trees for ranking data, the Entropy-based Ranking Trees (ERT). The difference is on the splitting and stopping criteria. ERT use IG to assess the splitting points and *MDLPC* [10] as stopping criterion. Using the measure of entropy for rankings [20], the splitting and stopping criteria come in a natural way.

The entropy for rankings [20] is defined as:

$$H_{ranking}\left(S\right)=\sum_{i=1}^{K}P\left(\pi_i,S\right)log\left(P\left(\pi_i,S\right)\right)log\left(\overline{kt}\left(S\right)\right) \tag{2}$$

where K is the number of distinct rankings in S and $\overline{kt}\left(S\right)$ is the average normalized Kendall τ distance in the subset S:

$$\overline{kt}\left(S\right)=\frac{\sum_{i=1}^{K}\sum_{j=1}^{n}\frac{\tau(\pi_i,\pi_j)+1}{2}}{K\times n_S}$$

where K is the number of distinct target values in S.

As in Section 2.1 the leafs of the tree should not be forced to have pure leafs. Instead, they should have a stop criterion to avoid overfitting and be robust to noise in rankings. As shown in [20], the *MDLPC Criterion* can be used as a splitting criterion with the adapted version of entropy $H_{ranking}$. This entropy measure also works with partial orders, however, in this work, we only use total orders.

One other ranking tree approach based in Gini Impurity, which will not be presented in detail in this work, was proposed in [25].

3 Experimental setup

The data sets in this work were taken from KEBI Data Repository in the Philipps University of Marburg [6] (Table 3). Two different transformation methods were used to generate these datasets: (A) the target ranking is a permutation of the classes of the original target attribute, derived from the probabilities generated by a naive Bayes classifier; (B) the target ranking is derived for each example from the order of the values of a set of numerical variables, which are no longer used as independent variables. Although these are somewhat artificial datasets, they are quite useful as benchmarks for LR algorithms.

The statistics of the datasets used in our experiments is presented in Table 3. U_π is the proportion of distinct target rankings for a given dataset.

Table 3. Summary of the datasets

Datasets	type	#examples	#labels	#attributes	U_π
autorship	A	841	4	70	2%
bodyfat	B	252	7	7	94%
calhousing	B	20,640	4	4	0.1%
cpu-small	B	8,192	5	6	1%
elevators	B	16,599	9	9	1%
fried	B	40,769	5	9	0.3%
glass	A	214	6	9	14%
housing	B	506	6	6	22%
iris	A	150	3	4	3%
pendigits	A	10,992	10	16	19%
segment	A	2310	7	18	6%
stock	B	950	5	5	5%
vehicle	A	846	4	18	2%
vowel	A	528	11	10	56%
wine	A	178	3	13	3%
wisconsin	B	194	16	16	100%

The code for all the examples in this paper has been written in R ([16]).

The performance of the LR methods was estimated using a methodology that has been used previously for this purpose [11]. It is based on the ten-fold cross validation performance estimation method. The evaluation measure is Kendall's τ and the performance of the methods was estimated using ten-fold cross-validation.

4 Results

RT uses a parameter, γ, that can affect the accuracy of the model. A $\gamma \geq 1$ does not increase the purity of nodes. On the other hand, small γ values will rarely generate any nodes. We vary γ from 0.50 to 0.99 and measure the accuracy on several KEBI datasets.

To show in what extent γ affects the accuracy of RT we show in Figure 1 the results obtained for some of the datasets in Table 3. From Figure 1 it is clear

Table 4. Results obtained for Ranking Trees on KEBI datasets. (The mean accuracy is represented in terms of Kendall's tau, τ)

	RT	ERT	LRT
authorship	.879	.890	.882
bodyfat	.104	.183	.117
calhousing	.181	.292	.324
cpu-small	.461	.437	.447
elevators	.710	.758	.760
fried	.796	.773	.890
glass	.881	.854	.883
housing	.773	.704	.797
iris	.964	.853	.947
pendigits	.055	.042	.935
segment	.895	.902	.949
stock	.854	.859	.895
vehicle	.813	.786	.827
vowel	.085	.054	.794
wine	.899	.907	.882
wisconsin	-.039	-.035	.343

that γ plays an important role in the accuracy of RT. It seems that the best values lie between 0.95 and 0.98. We will use $\gamma = 0.98$ for the Ranking Tees (RT).

Table 4 presents the results obtained by the two methods presented in comparison to the results for Label Ranking Trees (LRT) obtained in [6]. Even though LRT perform better in the cases presented, given the closer values to it, both RT and ERT give interesting results.

To compare different ranking methods we use a method proposed in [4] which is a combination of Friedmans test and Dunns Multiple Comparison Procedure [14]. First we run the Friedman's test to check whether the results are different or not, with the following hypotheses:

H_0 There is no difference in the mean average correlation coefficients for the 3 methods

H_1 There are some differences in the mean average correlation coefficients for the three methods

Using the *friedman.test* function from the *stats* package [16] we got a p-value $< 1\%$, which shows strong evidence against H_0.

Now that we know that there are some differences between the 3 methods we will test which are different from one another with the Dunns Multiple Comparison Procedure [14]. Using the R package *dunn.test* [9] with a Bonferroni adjustment, as in [4], we tested the following hypotheses for each pair of of methods a and b:

H_0 There is no difference in the mean average correlation coefficients between a and b

Table 5. P-values obtained for the comparison of the 3 methods

	RT	ERT	LRT
RT		1.0000	0.2619
ERTt	1.0000		0.1529
LRT	0.2619	0.1529	

H_1 There is some difference in the mean average correlation coefficients between a and b

The p-values obtained are presented in Table 5. Table 5 indicates that there is no strong statistically evidence that the methods are different. One other conclusion is that both RT and ERT are very equivalent approaches. While RT and ERT does not seem to outperform LRT in most of the cases studied, from the statical tests we can say that both approaches are competitive.

5 Conclusions

In this work we implemented a decision tree method for Label Ranking, *Ranking Trees* (RT) and proposed an alternative approach *Entropy-based Ranking Trees* (ERT). We also present an empirical evaluation on several datasets of correlation-based methods, RT and ERT, and compare with the state of the art distribution-based *Label Ranking Trees* (LRT). The results indicate that both RT and ERT are reliable LR methods.

Our implementation of Ranking Trees (RT) shows that the method is a competitive approach in the LR field. We showed that the input parameter, γ, can have a great impact on the accuracy of the method. The tests performed on KEBI datasets indicate that the best results are obtained when $0.95 < \gamma < 1$.

The method proposed in this paper, ERT, which uses IG as a splitting criterion achieved very similar results to the RT presented in [17]. Statistical tests indicated that there is no strong evidence that the methods (RT, ERT and LRT) are significantly different. This means that both RT and ERT are valid approaches, and, since they are correlation-based methods, we can also say that this kind of approaches is also worth pursuing.

References

1. Aiguzhinov, A., Soares, C., Serra, A.P.: A similarity-based adaptation of naive bayes for label ranking: Application to the metalearning problem of algorithm recommendation. In: Discovery Science - 13th International Conference, DS 2010, Canberra, Australia, October 6-8, 2010. Proceedings. pp. 16–26 (2010)
2. Blockeel, H., Raedt, L.D., Ramon, J.: Top-down induction of clustering trees. CoRR cs.LG/0011032 (2000), `http://arxiv.org/abs/cs.LG/0011032`

3. Brazdil, P., Soares, C., Costa, J.: Ranking Learning Algorithms: Using IBL and Meta-Learning on Accuracy and Time Results. Machine Learning 50(3), 251–277 (2003)
4. Brazdil, P., Soares, C., da Costa, J.P.: Ranking learning algorithms: Using IBL and meta-learning on accuracy and time results. Machine Learning 50(3), 251–277 (2003), http://dx.doi.org/10.1023/A:1021713901879
5. Cheng, W., Dembczynski, K., Hüllermeier, E.: Label ranking methods based on the plackett-luce model. In: ICML. pp. 215–222 (2010)
6. Cheng, W., Huhn, J.C., Hüllermeier, E.: Decision tree and instance-based learning for label ranking. In: Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009, Montreal, Quebec, Canada, June 14-18, 2009. pp. 161–168 (2009)
7. Cheng, W., Hüllermeier, E.: Label ranking with abstention: Predicting partial orders by thresholding probability distributions (extended abstract). Computing Research Repository, CoRR abs/1112.0508 (2011), http://arxiv.org/abs/1112.0508
8. Cheng, W., Hüllermeier, E., Waegeman, W., Welker, V.: Label ranking with partial abstention based on thresholded probabilistic models. In: Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States. pp. 2510–2518 (2012), http://books.nips.cc/papers/files/nips25/NIPS2012_1200.pdf
9. Dinno, A.: dunn.test: Dunn's Test of Multiple Comparisons Using Rank Sums (2015), http://CRAN.R-project.org/package=dunn.test, r package version 1.2.3
10. Fayyad, U.M., Irani, K.B.: Multi-interval discretization of continuous-valued attributes for classification learning. In: Proceedings of the 13th International Joint Conference on Artificial Intelligence. Chambéry, France, August 28 - September 3, 1993. pp. 1022–1029 (1993)
11. Hüllermeier, E., Fürnkranz, J., Cheng, W., Brinker, K.: Label ranking by learning pairwise preferences. Artificial Intelligence 172(16-17), 1897–1916 (2008)
12. Kendall, M., Gibbons, J.: Rank correlation methods. Griffin London (1970)
13. Mitchell, T.: Machine Learning. McGraw-Hill (1997)
14. Neave, H., Worthington, P.: Distribution-free Tests. Routledge (1992), http://books.google.nl/books?id=1Y1QcgAACAAJ
15. Quinlan, J.R.: Induction of decision trees. Machine Learning 1(1), 81–106 (1986), http://dx.doi.org/10.1023/A:1022643204877
16. R Development Core Team: R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria (2010), http://www.R-project.org, ISBN 3-900051-07-0
17. Rebelo, C., Soares, C., Costa, J.: Empirical Evaluation of Ranking Trees on Some Metalearning Problems. In: Chomicki, J., Conitzer, V., Junker, U., Perny, P. (eds.) Proceedings 4th AAAI Multidisciplinary Workshop on Advances in Preference Handling (2008)
18. Ribeiro, G., Duivesteijn, W., Soares, C., Knobbe, A.J.: Multilayer perceptron for label ranking. In: Artificial Neural Networks and Machine Learning - ICANN 2012 - 22nd International Conference on Artificial Neural Networks, Lausanne, Switzerland, September 11-14, 2012, Proceedings, Part II. pp. 25–32 (2012)
19. de Sá, C.R., Soares, C., Jorge, A.M., Azevedo, P.J., da Costa, J.P.: Mining association rules for label ranking. In: Advances in Knowledge Discovery and Data Mining - 15th Pacific-Asia Conference, PAKDD 2011, Shenzhen, China, May 24-27, 2011, Proceedings, Part II. pp. 432–443 (2011)

20. de Sá, C.R., Soares, C., Knobbe, A.: Entropy-based discretization methods for ranking data. Information Sciences in press, in press (2015)
21. de Sá, C.R., Soares, C., Knobbe, A.J., Azevedo, P.J., Jorge, A.M.: Multi-interval discretization of continuous attributes for label ranking. In: Discovery Science - 16th International Conference, DS 2013, Singapore, October 6-9, 2013. Proceedings. pp. 155–169 (2013)
22. Spearman, C.: The proof and measurement of association between two things. American Journal of Psychology 15, 72–101 (1904)
23. Todorovski, L., Blockeel, H., Džeroski, S.: Ranking with Predictive Clustering Trees. In: Elomaa, T., Mannila, H., Toivonen, H. (eds.) Proc. of the 13th European Conf. on Machine Learning. pp. 444–455. No. 2430 in LNAI, Springer-Verlag (2002)
24. Vembu, S., Gärtner, T.: Label ranking algorithms: A survey. In: Fürnkranz, J., Hüllermeier, E. (eds.) Preference Learning, pp. 45–64. Springer-Verlag (2010)
25. Xia, F., Zhang, W., Li, F., Yang, Y.: Ranking with decision tree. Knowl. Inf. Syst. 17(3), 381–395 (2008), `http://dx.doi.org/10.1007/s10115-007-0118-y`

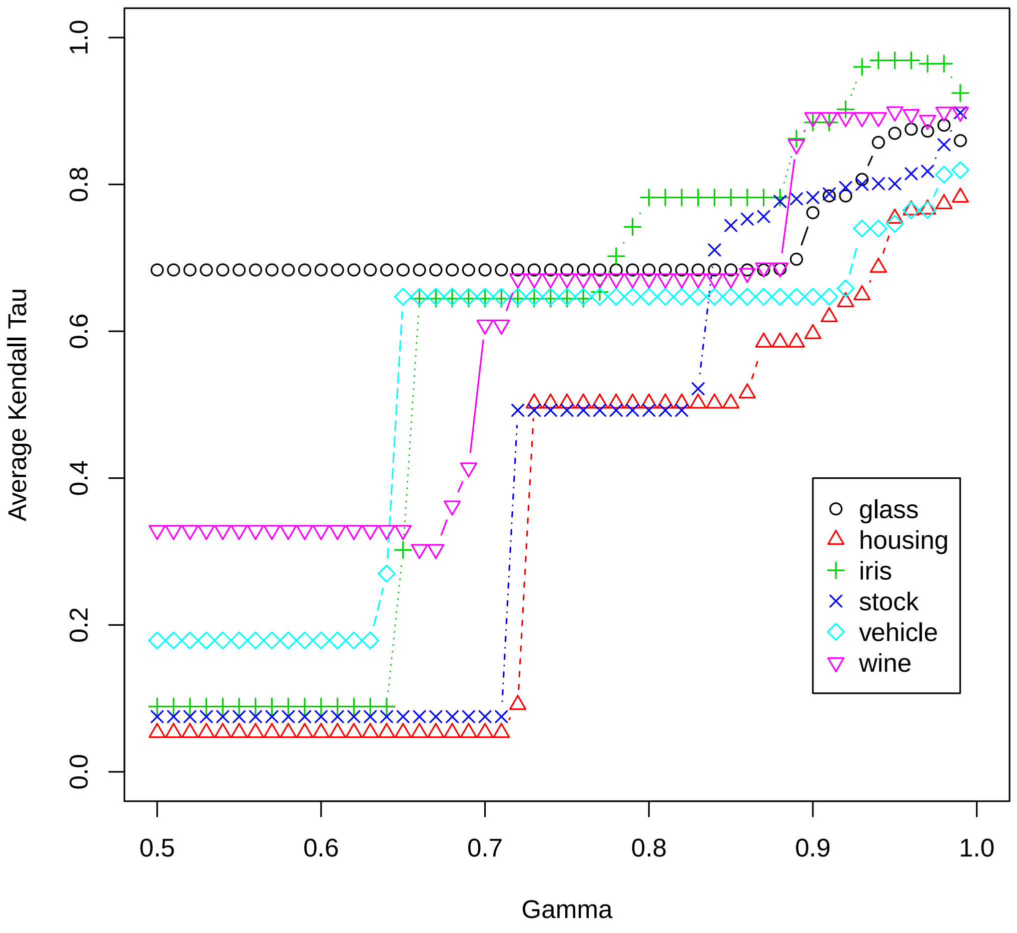

Fig. 1. Comparison of the accuracy obtained on some datasets by RT as γ varies from 0.5 to 0.99.