Dynamic Pruning for Tree-based Ensembles

Mostafa EL HABIB DAHO, Mohammed El Amine LAZOUNI, Mohammed Amine CHIKH

Département d'Informatique
Université de Tlemcen
Ville Tlemcen
Pays Algérie
mostafa.elhabibdaho@mail.univ-tlemcen.dz

ABSTRACT. In this paper, we propose a new dynamic pruning based on trees selection in ensembles methods. This algorithm allows, for each test instance, the selection of the best trees in the forest. This approach is tested on 10 databases from the UCI Machine Learning Repository. Results show that using a few best trees selected by our proposed pruning method, we can improve the performance of each dataset compared to classical ensembles methods and pruning techniques.

KEYWORDS : Random Forests, Bagging, Random Subspaces, Sub_RF, Trees selection, Dynamic Ensemble Pruning, Supervised Classification.

1. Introduction

The principle of ensemble methods (for example [6]) is to build a collection of predictors, and then aggregate all of their predictions. In classification, aggregation returns, for example, a majority vote among the classes provided by each individual predictor.

In this work, tree-based ensemble methods are used. They consist of a set of prediction trees ; each one being capable of producing a response when presented a sub-set of variables. For classification problems, the response takes the form of a class (label).

Using the sets of trees, a significant improvement in prediction compared with the conventional techniques (like CART) is believed to be obtained. Response of each tree depends on the subset of independently selected variables. One of the most used tree-based ensemble methods called RF (Random Forest)[4].

Despite the efficiency of the random forests, several researchers have tried to improve the accuracy using only the best trees of the forest. This improved method is called Trees Selection or Pruning. There are two kinds of Pruning : Static Pruning where a subset of trees is selected once for the whole test set, and Dynamic Pruning where the selection is made for each test sample individually at prediction time.

In this paper, the main interest is therefore to study the ability of tree selection on a modified version of random forests (called Sub_RF) by selecting the best ensemble of trees. Our new proposed method for tree selection attempts at improving accuracy. For that, this work has been framed as follows : in section 2, methods that we use in our algorithm are introduced. After that, related works to the method we made for ensemble pruning is discussed. Then, our results obtained on some benchmarks from the UCI Machine Learning Repository are exposed. At last, a general summary is given that highlights the main properties of our technique.

2. Methods

2.1. Random Forest

In random forests, Breiman proposes to use the Bagging [5], but for each data set generated, the growth of the tree is processed with a random selection of explanatory variables at each node. The word Bagging is a contraction of Bootstrap and Aggregating [1]. The idea of Bagging, is that by applying the basic rule on different bootstrap samples, we modify the predictions, and so we eventually build a collection of various predictors. The aggregation step then allows to obtain a powerful predictor.

The Random Forests algorithm - Random Input (Forest-RI) [4] is one of the most popular achievements of research devoted to the aggregation of randomized trees. Synthesizing the approaches developed respectively by [5] and [1], it generates a set of trees doubly disrupted using a randomization operating both at the training sample and internal partitions. Each tree is thus generated at first from a subsample (a bootstrap sample) of the complete training set, similar to the techniques of bagging. Then the tree is constructed using the CART methodology with the difference that at each node the selection of the best split based on the Gini index is performed not on the complete set of attributes **M** but on a randomly selected subset of it. During the prediction phase, the individual to be

1. A bootstrap sample L is obtained by randomly drawing n observations with replacement from the training sample L_n, each observation has probability $1/n$ to be pulled.

classified is spread in every tree of the forest and labelled according to the CART rules. The whole forest prediction is provided by a simple majority vote of the class assignments of individual trees.

In addition to building a predictor, the algorithm of Random Forests-RI calculates an estimate of its generalization error : the error Out-Of-Bag (OOB). This error was already calculated by the Bagging algorithm ; hence, the presence of "Bag". The calculation procedure of this error is as follows : From a training set "A" of "X" examples , bootstraps samples are generated by drawing "X" samples with replacement from "A". In average, for each bootstrap sample 63.2% are unique examples of "A", the rest being duplicates. So for each sub base, 1/3 samples of "A" are not selected and are called OOB samples. They will be used in internal evaluation of the forest (estimated error classification generalization of forest) or as a measure to calculate the variable of importance to use it in variable selection.

2.2. Subspaces Random Forest

In this method, the creation of a set of classifiers is made by using the method SubBag [17] for the generation of training samples. The classifiers are decision trees generated by using the Forest-RI algorithm [4].This algorithm of tree ensemble creation is called Sub_RF (Subspaces Random Forest) [7].

Algorithm 1 Pseudo code of the Sub_RF algorithm (LearnSubRF)

Input : The Training set L, Number of Random Trees N, SubSpace size S.
Output : $TreesEnsemble$
Process :
for $i = 1 \rightarrow N$ **do**
  $T^i \leftarrow BootstrapSample(T)$
  $T^i \leftarrow SelectRandomSubSpaces(T^i, S)$
  $C^i \leftarrow ConstructRF\_tree(T^i)$
  $E \leftarrow E \cup \{C^i\}$
end for
$ReturnE$

The function $ConstructRF\_tree()$ allows to create trees using the principle of random forests.

3. Related works

Ensemble selection algorithms (also called pruning algorithms) aim at finding the best subset, among the set of all hypotheses space, which may optimize the computation time (as in static Pruning) and / or improve performances (dynamic pruning). The main aim of this experimental work is to fundamentally apply ensemble selection methods for selecting best classifiers from a random forest which is generated using the method SubBag. There exist several studies in the literature that we discuss below according to their types (static or dynamic).

3.1. Static Pruning

Static pruning consists in creating a set of classifiers (random forest or other) and then selecting a part of this set (the best classifiers) that performs as well as, or better than, the original ensemble. The selected set will be used for the classification of test instances. Many researchers have shown in their studies on the tree selection in a random forest, that better subsets of decision trees can be obtained by using sub-optimal methods of classifier selection [29] [20] [26] [15] [3]. Their results affirm that an induction algorithm of classical random forests is not the best approach to produce well performing tree-based classifiers.

Among the most recent works, in this regard, we find the article of Zhao et al. [27] where the authors propose a fast pruning method compared with the existing methods. Their idea is to create a prediction table where each row of the table contains a database instance and each column a classifier. The proposed algorithm chose the best combination of classifiers that minimizes the error.

[13] in their article, propose a heuristic that respects the compromise accuracy / diversity for the evaluation of the contribution of each classifier and thus, choose the best subset. Their results show that the subset chosen by their algorithm EPIC (for Ensemble pruning via indivdual contribution ordering) outperforms the original set.
Other studies present classifiers selection as an optimization problem where we had to look for the best solution in the space. Most of the proposed algorithms have used optimization algorithms such as greedy search [8] [16] [18], hill climbing [25] or even genetic algorithms [28].

In [11], the authors have presented an entropy-inspired ordering ensemble pruning algorithm exploiting an alternative definition of the margin of ensemble methods. This pruning strategy considers the smallest margin instances as the most significant in building reliable classifiers. The algorithm combines best classifiers, which classify correctly smallest margin, for future decisions.

3.2. Dynamic Pruning

Dynamic pruning (also called dynamic ensemble selection or instance-based ensemble selection) aims at selecting the best subset of classifiers dynamically (ie : for each test example) from the original set. The selected classifiers are aggregated afterwards by a majority vote. The subset should lead to a greater accuracy compared to the whole set. This type of selection is best suited for offline problems where we privilege accuracy over computation time because there is an additional cost in the testing phase.

[24] and [10] are said to be among the first authors who were interested in dynamic selection. Their methods consist in using for each instance of the test base, the best classifiers of its neighborhood (using KNN). Authors propose two methods to calculate the performance of classifiers. The first is OLA (Overall local Accuracy) ; this metric calculates the rate of correct classifications of each classifier on instances of the neighborhood. The second metric is called LCA (Local Class Accuracy), it allows to calculate, for each classifier, the rate of correct classification of examples in the neighborhood that have the same given class for the test instance. Best Classifiers are combined to classify this instance.

Two other approaches, dynamic selection (DS) and dynamic voting (DV) have been proposed by [19]. DS uses the same principle as OLA [24] but by weighting selected classifiers by their distance. DV does not use KNN but rather all the classifiers weighted by their local competence. An approach between DS and DV was introduced by [21] where the author proposed to select the 50% best classifiers and then combining them using DV.

Among the most recent works, one may find that of [12]. The authors proposed four different versions of a method called KNORA (K-nearest Oracle). The proposed algorithms use the KNN to select neighbors of each test instance.

[14] modelled the pruning as a multi-label problem called IBEP-MLC (Instance-Based Ensemble Pruning via Multi-label Classification). The idea proposed by the authors is to add, for each instance of the training set, a label with each classifier. If the instance is well classified, a positive label is given (+), otherwise it is a negative one (-). The classification of a new instance is made by taking the classifiers with a positive label in its neighborhood.

In [23] authors developed a probabilistic model method for calculating the classifier competence. The competences calculated for a validation set are generalized to an entire feature space by constructing a competence function based on a potential function model or regression. Three systems based on a dynamic classifier selection and dynamic ensemble selections (DES) were constructed using the method developed.

In [9], they have proposed a dynamic classifier selection strategy for One-vs-One scheme that tries to avoid the non-competent classifiers when their output is probably not of interest. This method considers the neighborhood of each instance to decide which classifier may correctly classify this instance.

4. Proposed Method

It has been noticed that all the works previously cited, in the section dynamic pruning, are based on KNN for the choice of the neighborhood, which is an additional parameter to adjust. Noting that this method is not effective if we do not use all the space of attributes (case of RSM or SubBag). Indeed, two instances may be far in the complete space and close in a part of it.

As a solution to this problem, a method based on a different notion of neighborhood is suggested. In this work, the nodes of the trees are used as a heuristic neighborhood. Indeed, two instances are adjacent if they pass through the same nodes in a given tree. Our algorithm involves three steps :

– Generation of a random tree-based ensemble using Sub_RF method [7].

– For each tree in the forest, the classification of its OOB elements (with this tree) is launched and their paths are saved (step (1) in the Algorithm 4).

– To classify a new instance, the score of each tree for this instance should be calculated and process to a majority vote among the K-best trees. The score of the tree is calculated based on the correct classification of its OOB weighted by their distance with this instance (step (2) in the Algorithm 4.).

For a test instance, the score of a tree, is a value comprised between 0 and 1. A score equal to "1" means that the tree is very efficient and will ensure a correct classification for this test instance. A tree with a score equal to "0", has a high chance to give a false classification for the instance.

The principle of calculating the score of a tree, for an instance, is very simple. It is based on a Boolean function which weights the distance between the test instance and each OOB of this tree. This function returns "1" if the element OOB was well classified by the tree, otherwise "0".

A distance between a test instance and an OOB equal to "1" means they have gone together through all the nodes of the tree. A distance very close to zero means that the two elements have gone through different paths.

The notion of neighborhood based paths was introduced by Vens and Costa in [22]. It is about calculating communes nodes between an OOB and a given instance considering all the paths and not only leafs. The distance of an OOB compared to an instance is a fraction of the number of nodes traversed together over the maximum depth between this two paths.

5. Results and interpretations

To test our algorithm, ten databases from the UCI Machine Learning Repository [2] were used. Databases which have been used in our experiments are described in the Table 1.

Our experiments are to implement seven different ensembles : Sub_RF, Sub_RF with Static Pruning, Sub_RF with Dynamic Pruning, Sub_RF with OEP, Bagging with OEP, Randomized trees with OEP and RF with OEP. The goal is to visualize and study the evolution of the error rate of each method and subsets obtained during the process of tree selection.

First, each database has been divided into two sub-data sets, one for learning and the other for test (using 5-fold cross validation). The separation of the data was carried out by random draw from the whole set.

Databases	Inst	Features	Cl
Breast	699	9	2
Ecoli	366	7	8
Habermann	306	3	2
Isolet	7797	617	26
Liver	345	6	2
Pendigits	10992	16	10
Pima	768	8	2
Segmentation	2310	19	7
Vehicle	846	18	4
Yeast	1484	8	10

Tableau 1. *Used databases*

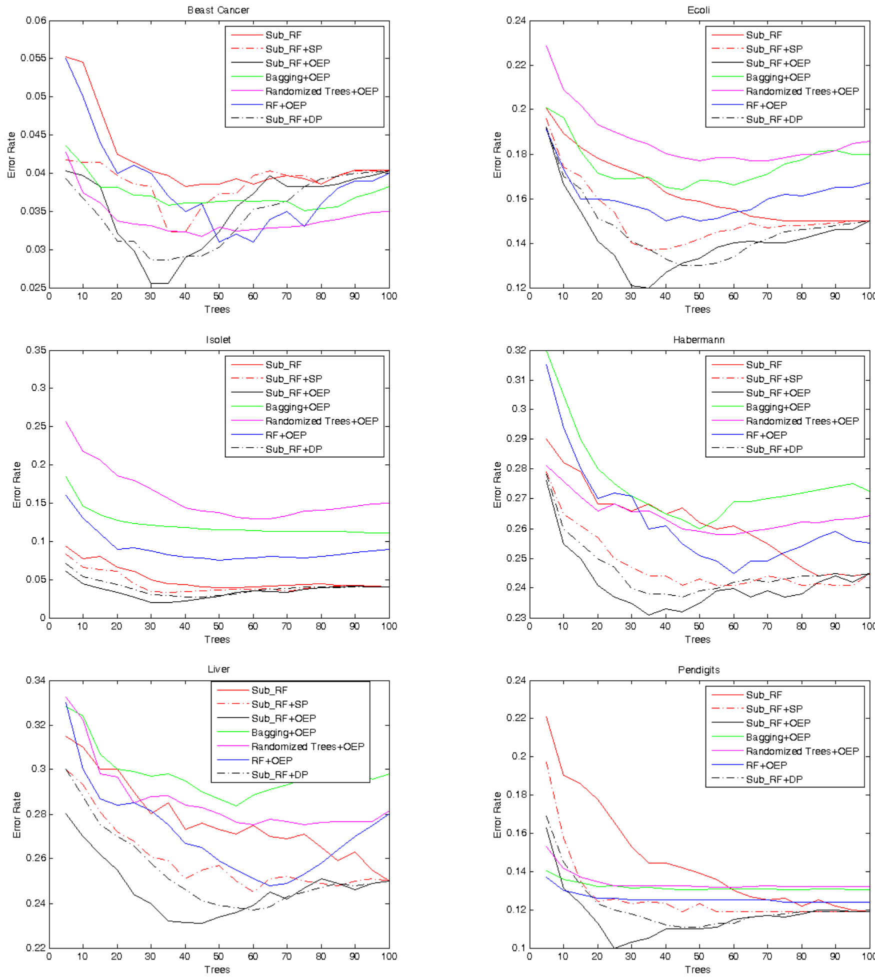
As it has been already explained, our method uses bootstrapping to generate the bag. OOB will be used for selecting classifiers. Several works in the literature bulk have shown that a number of attributes equal to $\sqrt{M}$ is a good compromise to produce an efficient forest [4] [3].

In this experiments, a comparison of our proposed dynamic pruning method OEP (for Out of bag-based Ensemble Pruning), Static Pruning (SP) and Dynamic Pruning (DP) applied on Random Trees (which uses only one random feature), Random forests (RF), Bagging and Sub_RF was established. Groups of selections were organized to which, each time, five trees to the group where added. In the first experiment, a random tree selection for Sub_RF, where trees are selected and aggregated according to their order of appearance and without condition, was processed. For the Static pruning, the OOB database is used like a validation database and the performance of each tree is calculated based on the correct classification rate of its OOB. At each stage, the K-best trees are selected for the classification of the test set. OEP Algorithm is used with all cited methods and compared with the Dynamic Pruning algorithm based on KNN used with Sub_RF (Sub_RF+DP in the figures).

Fig.1 show error rates of different combinations as the number of selected trees increases. It may be observed that our algorithm of dynamic pruning OEP gives best result between 20 and 50 trees for all databases. The best results are obtained with the forest generated by the Sub-RF algorithm. This can be explained by the fact that, unlike Bagging and RF, the Sub-RF trees are very different since they do not use all attributes and, unlike the Random Trees, they choose the best variable. Sub_RF thus provides overall the best tradeoff in terms of randomization in the context of our dynamic pruning algorithm. OEP seems to gives better results than the static pruning and dynamic pruning methods that use KNN : it leads globally a lower error rate than all methods and it also reaches its optimum for a smaller set of trees. Therefore, the neighborhood based on tree nodes is more efficient if we do not use the whole attribute space.

6. Conclusion

To put it in a nutshell ; in this paper, a new instance-based ensemble pruning method which uses the neighborhood in the tree has been essentially hypothesized. This method has, in fact, proven effective on trees that do not use all the attribute space. For this, it sounds quite important to investigate the efficiency of a method of generating tree which is very similar to SubBag (Sub_RF) and gives better results compared to conventional random forests. For that reason, our approach on ten UCI databases was experimentally tested. Results display that our suggested approach is almost competitive with pruning methods (static and dynamic) which are based on KNN.


Beast Cancer
Ecoli
Isolet
Habermann
Liver
Pendigits
Sub_RF
Sub_RF+SP
Sub_RF+OEP
Bagging+OEP
Randomized Trees+OEP
RF+OEP
Sub_RF+DP
Error Rate
Trees

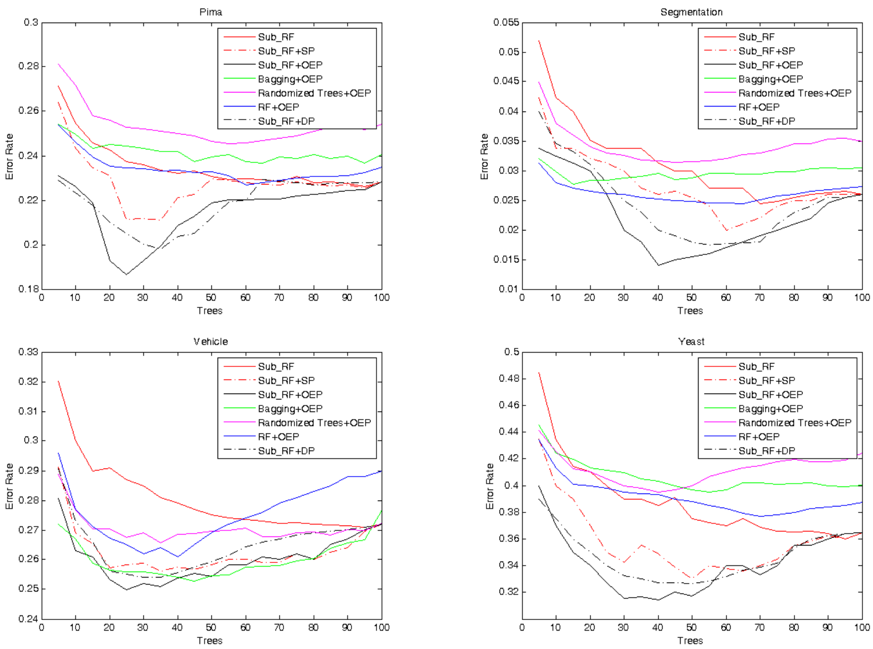


Figure 1. *Error rates of different algorithms*

7. Bibliographie

Yali Amit and Donald Geman. Shape quantization and recognition with randomized trees. *Neural Computation*, 9(7) :1545–1588, 1997.

K. Bache and M. Lichman. UCI machine learning repository, 2013.

Simon Bernard, Laurent Heutte, and Sebastien Adam. Influence of hyperparameters on random forest accuracy. In *Multiple Classifier Systems*, volume 5519 of *Lecture Notes in Computer Science*, pages 171–180. Springer, 2009.

L. Breiman. Random forests. *Machine Learning*, 45 :5–32, 2001.

Leo Breiman. Bagging predictors. *Machine Learning*, 24(2) :123–140, 1996.

T. Dietterich. Ensemble methods in machine learning. *Lecture Notes in Computer Science*, 1857 :1–15, 2000.

Mostafa EL HABIB DAHO and Mohammed El Amine CHIKH. Combining bootstrapping samples, random subspaces and random forests to build classifiers. *Journal of Medical Imaging and Health Informatics*, 5(3) :539–544, 2015.

Wei Fan, Haixun Wang, Philip S. Yu, and Sheng Ma. Is random model better ? on its accuracy and efficiency. In *Third IEEE Int. Conf. on Data Mining*, ICDM '03, pages 51–, Washington, DC, USA, 2003. IEEE Computer Society.

Mikel Galar, Alberto Fernández, Edurne Barrenechea, Humberto Bustince, and Francisco Herrera. Dynamic classifier selection for one-vs-one strategy : Avoiding non-competent classifiers. *Pattern Recognition*, 46(12) :3412–3424, December 2013.

Giorgio Giacinto and Fabio Roli. Adaptive selection of image classifiers. In *Image Analysis and Processing*, volume 1310, pages 38–45. Springer Berlin Heidelberg, 1997.

Li Guo and Samia Boukir. Margin-based ordered aggregation for ensemble pruning. *Pattern Recognition Letters*, 34(6) :603–609, April 2013.

Albert H.R. Ko, Robert Sabourin, Alceu Souza Britto, and Jr. From dynamic classifier selection to dynamic ensemble selection. *Pattern Recognition*, 41(5) :1718 – 1731, 2008.

Zhenyu Lu, Xindong Wu, Xingquan Zhu, and Josh Bongard. Ensemble pruning via individual contribution ordering. In *16th ACM SIGKDD Int. conf. on Knowledge discovery and data mining*, KDD '10, pages 871–880, New York, NY, USA, 2010. ACM.

F. Markatopoulou, G. Tsoumakas, and L. Vlahavas. Instance-based ensemble pruning via multi-label classification. In *22nd IEEE Int. Conf. on Tools with Artificial Intelligence*, volume 1, pages 401–408, 2010.

Gonzalo Martinez-Munoz, Alberto Suarez, Gonzalo Martínez-muñoz, and Alberto Suárez. Using boosting to prune bagging ensembles. *Pattern Recognition Letters*, 28 :156–165, 2007.

Gonzalo Martínez-muñoz and Alberto Suárez. Aggregation ordering in bagging. In *Int. Conf. on Artificial Intelligence and Applications*, pages 258–263. Acta Press, 2004.

Panče Panov and Sašo Džeroski. Combining bagging and random subspaces to create better ensembles. In *7th Int. Conf. on Intelligent data analysis*, IDA'07, pages 118–129, Berlin, Heidelberg, 2007. Springer-Verlag.

Ioannis Partalas, Grigorios Tsoumakas, and Ioannis Vlahavas. A study on greedy algorithms for ensemble pruning. Technical report, Aristotle University of Thessaloniki, 2012.

Seppo J. Puuronen, Vagan Terziyan, Artyom Katasonov, and Alexey Tsymbal. Dynamic integration of multiple data mining techniques in a knowledge discovery management system. *Data Mining and Knowledge Discovery : Theory, Tools, and Technology*, 3695 :128–139, 1999.

Grigorios Tsoumakas, Lefteris Angelis, and Ioannis Vlahavas. Selective fusion of heterogeneous classifiers. *Intelligent Data Analysis*, 9(6) :511–525, November 2005.

Alexey Tsymbal. Decision committee learning with dynamic integration of classifiers. In *East-European Conf. on Advances in Databases and Information Systems*, ADBIS-DASFAA '00, pages 265–278, London, UK, UK, 2000. Springer-Verlag.

Celine Vens and Fabrizio Costa. Random forest based feature induction. *IEEE Int. Conf. on Data Mining*, 0 :744–753, 2011.

Tomasz Woloszynski and Marek Kurzynski. A probabilistic model of classifier competence for dynamic ensemble selection. *Pattern Recognition*, 44(1011) :2656 – 2668, 2011.

Kevin Woods, W. Philip Kegelmeyer, Jr., and Kevin Bowyer. Combination of multiple classifiers using local accuracy estimates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(4) :405–410, April 1997.

Ying Yang, Kevin Korb, KaiMing Ting, and GeoffreyI. Webb. Ensemble selection for superparent-one-dependence estimators. In *Advances in Artificial Intelligence*, volume 3809, pages 102–112. 2005.

Yi Zhang, Samuel Burer, and W. Nick Street. Ensemble pruning via semi-definite programming. *Machine Learning Research*, 7 :1315–1338, December 2006.

Qiang-Li Zhao, Yan-Huang Jiang, and Ming Xu. A fast ensemble pruning algorithm based on pattern mining process. *Data Mining and Knowledge Discovery*, 19(2) :277–292, October 2009.

Z.-H. Zhou, W. Tang, Zhi hua Zhou, and Wei Tang. Selective ensemble of decision trees. In *Lecture Notes in Artificial Intelligence*, pages 476–483. Springer, 2003.

Zhi-Hua Zhou, Jianxin Wu, and Wei Tang. Ensembling neural networks : many could be better than all. *Artificial Intelligence*, 137(1-2) :239–263, May 2002.