Ensembles of Extremely Randomized Trees for Multi-target Regression

Dragi Kocev[1,2(✉)] and Michelangelo Ceci[1]

[1] Department of Computer Science, University of Bari Aldo Moro, Bari, Italy
dragi.kocev@ijs.si, michelangelo.ceci@uniba.it
[2] Department of Knowledge Technologies, Jožef Stefan Institute, Ljubljana, Slovenia

Abstract. In this work, we address the task of learning ensembles of predictive models for predicting multiple continuous variables, i.e., multi-target regression (MTR). In contrast to standard regression, where the output is a single scalar value, in MTR the output is a data structure – a tuple/vector of continuous variables. The task of MTR is recently gaining increasing interest by the research community due to its applicability in a practically relevant domains. More specifically, we consider the Extra-Tree ensembles – the overall top performer in the DREAM4 and DREAM5 challenges for gene network reconstruction. We extend this method for the task of multi-target regression and call the extension Extra-PCTs ensembles. As base predictive models, we propose to use predictive clustering trees (PCTs) – a generalization of decision trees for predicting structured outputs, including multiple continuous variables. We consider both global and local prediction of the multiple variables, the former based on a single model that predicts all of the target variables simultaneously and the latter based on a collection of models, each predicting a single target variable. We conduct an experimental evaluation of the proposed method on a collection of 10 benchmark datasets for with multiple continuous targets and compare its performance to random forests of PCTs. The results reveal that a multi-target Extra-PCTs ensemble performs statistically significantly better than a single multi-target or single-target PCT. Next, the performance among the different ensemble learning methods is not statistically significantly different, while multi-target Extra-PCTs ensembles are the best performing method. Finally, in terms of efficiency (running times and model complexity), both multi-target variants of the ensemble methods are more efficient and produce smaller models as compared to the single-target ensembles.

Keywords: Multi-target regression · Ensembles · Extremely randomized trees · Predictive clustering trees

1 Introduction

Supervised learning is one of the most widely researched and investigated areas of machine learning. The goal in supervised learning is to learn, from a set of examples with known class, a function that outputs a prediction for the class of

N. Japkowicz and S. Matwin (Eds.): DS 2015, LNAI 9356, pp. 86–100, 2015.
DOI: 10.1007/978-3-319-24282-8_9

a previously unseen example. However, in many real life problems of predictive modelling the output (i.e., the target) is structured, meaning that there can be dependencies between classes (e.g., classes are organized into a tree-shaped hierarchy or a directed acyclic graph) or some internal relations between the classes (e.g., sequences).

In this work, we concentrate on the task of predicting multiple continuous variables. Examples thus take the form $(\mathbf{x_i}, \mathbf{y_i})$, where $\mathbf{x_i} = (x_{i1}, \ldots, x_{ik})$ is a vector of k input variables and $\mathbf{y_i} = (y_{i1}, \ldots, y_{it})$ is a vector of t target variables. This task is known under the name of *multi-target regression* (MTR) [1] (also known as multi-output or multivariate regression).

MTR is a type of structured output prediction task which has application in many real life problems where we are interested in simultaneously predicting multiple continuous variables. Prominent examples come from ecology: predicting abundance of different species living in the same habitat [2], or predicting properties of forests [3,4]. Due to its applicability in a wide range of domains, this task is recently gaining increasing interest in the research community.

Several methods for addressing the task of MTR have been proposed [1,5]. These methods can be categorized into two groups of methods [6]: (1) local methods that predict each of the target variable separately and then combine the individual models to get the overall model and (2) global methods that predict all of the variables simultaneously (also known as 'big-bang' approaches). In the case of local models, for a domain with t target variables one needs to construct t predictive models – each predicting a single target. The prediction vector (that consists of t components) of an unseen example is then obtained by concatenating the predictions of the multiple single-target predictive models. Conversely, in the case of global models, for the same domain, one needs to construct 1 model. The prediction vector of an unseen example here is then obtained by passing the example through the model and getting its prediction.

In the past, several researchers proposed methods for solving the task of MTR directly and demonstrated their effectiveness [1,4,7,8]. The global methods have several advantages over the local methods. First, they exploit and use the dependencies that exist between the components of the structured output in the model learning phase, which can result in better predictive performance. Next, they are typically more efficient: it can easily happen that the number of components in the output is very large (e.g., hierarchies in functional genomics can have several thousands of components), in which case executing a basic method for each component is not feasible. Furthermore, they produce models that are typically smaller than the sum of the sizes of the models built for each of the components.

In [1,9], we evaluated the construction of local and global models for MTR in the context of ensemble learning. More specifically, we focus on two most widely used ensemble learning techniques: bagging [10] and random forests [11]. We show that both global and local tree ensembles perform better than the single model counterparts in terms of predictive power. Global and local tree ensembles perform equally well, with global ensembles being more efficient and producing

smaller models, as well as needing fewer trees in the ensemble to achieve the maximal performance.

In this paper, we investigate a new strategy for learning MTR global models through ensemble learning. In particular, we extend the Extra-Trees algorithm to the context of MTR. The Extra-Trees algorithm, proposed by Geurts et al. [12], is an algorithm for tree ensemble construction based on an extreme randomization of the tree construction algorithm. The algorithm at each node of the tree randomly selects k attributes and, on each of them, randomly selects a split. The k candidate splits are then evaluated and the best split is put in the node. Here, we propose an extension of the Extra-Trees algorithm for the task of predicting multiple continuous variables.

Geurts et al. evaluated their approach in the context of single-target regression and classification problems containing only numerical attributes. The bias/variance analysis of the error revealed that Extra-Trees decrease the variance while at the same time they increase the bias. If the level of randomization is well adjusted, then the variance almost disappears at the cost of a slight increase of the bias with respect to that of standard trees. In this study, we perform an empirical evaluation of the Extra-Trees algorithm extension in domains where the descriptive attributes can be continuous, categorical or mixed.

The Extra-Trees algorithm has been successfully applied to several practically relevant domains including computer vision [13] and gene network inference [14,15]. Especially noticeable are the applications in the latter domain: a variant of the method that exploits its feature ranking mechanism (GENIE3 algorithm) has been overall top performer in the DREAM4 and DREAM5 challenges[1] for gene network inference. All of these considerations strongly motivate this study.

In this paper, we propose an extension of the Extra-Trees algorithm based on the predictive clustering trees (PCTs) framework [1,16]. We call this extension Extra-PCTs algorithm. PCTs belong to the group of global methods and can be considered as a generalization of standard decision trees towards predicting structured outputs. They offer a unifying approach for dealing with different types of structured outputs and construct the predictive models very efficiently. They are able to make predictions for several types of structured outputs: tuples of continuous/discrete variables, hierarchies of classes [17], and time series.

The remainder of this paper is organized as follows. Section 2 presents the proposed Extra-PCTs algorithm for MTR. Next, Sect. 3 outlines the design of the experimental evaluation. Furthermore, Sect. 4 discusses the results. Finally, Sect. 5 concludes and provides directions for further work.

2 MTR with Ensembles of Predictive Clustering Trees

The predictive clustering trees framework views a decision tree as a hierarchy of clusters. The top-node corresponds to one cluster containing all data, which

[1] For more information, visit http://dreamchallenges.org/.

is recursively partitioned into smaller clusters while moving down the tree. The PCT framework is implemented in the CLUS system [18], which is available for download at http://clus.sourceforge.net.

PCTs are induced with a standard *top-down induction of decision trees* (TDIDT) algorithm [19]. Table 1 outlines the general algorithm for PCT induction. It takes as input a set of examples (E) and outputs a tree. The heuristic (h) that is used for selecting the tests (t), in a regular PCT, is the reduction in variance caused by the partitioning ($\mathcal{P}$) of the instances corresponding to the tests (t) (see line 7 of the BestTest procedure in Table 2). Intuitively, by maximizing the variance reduction, the cluster homogeneity is maximized and the predictive performance is improved.

Table 1. The top-down induction algorithm for PCTs.

procedure ExtremelyRnd PCT
Input: A dataset E, size of attribute subset k
Output: A predictive clustering tree
1: $(t^*, h^*, \mathcal{P}^*) = \text{FindTest}(E)$
2: **if** $t^* \neq none$ **then**
3: **for each** $E_i \in \mathcal{P}^*$ **do**
4: $tree_i = \text{PCT}(E_i)$
5: **return** $\text{node}(t^*, \bigcup_i \{tree_i\})$
6: **else**
7: **return** $\text{leaf}(\text{Prototype}(E))$

The extremely randomized variant of PCTs introduces a randomization in the test selection (Table 2). More specifically, it requires an input parameter (k) that controls the number of attributes considered at each node of the tree. The test selection procedure randomly selects k attributes and from each attribute randomly selects a split. For each of the k selected attributes, the algorithm selects the split in two different ways, depending on the type of the attribute. If the attribute is numeric the splitting point is selected randomly from the set of possible splitting points. Possible splitting points are found in the set of values of the attribute in the training set associated to the specific node. If the attribute is categorical (i.e., nominal) then a non-empty subset of values of the attribute in the training set associated to the specific node are randomly selected.

The k-candidate tests are then evaluated using the variance reduction heuristic and the best test is selected. In order to take multiple target variables into account simultaneously, variance used to initialize h is defined as follows:

$$Var(E) = \sum_{j=1}^{t} Var(E, Y_j),$$

Table 2. Extremely randomized test selection for PCTs.

procedure FindTest
Input: A dataset E
Output: the selected test (t^*), its heuristic score (h^*) and the partition ($\mathcal{P}^*$) it induces on the dataset (E)

1: $(t^*, h^*, \mathcal{P}^*) = (none, 0, \emptyset)$
2: $A = getAttributeList(E)$
3: $A_s = selectAttributes(E, k)$
4: **for each** attribute $a \in A_s$ **do**
5: $t = selectRandomTest(a)$
6: $\mathcal{P}$ = partition induced by t on E
7: $h = Var(E) - \sum_{E_i \in \mathcal{P}} \frac{|E_i|}{|E|} Var(E_i)$
8: **if** $(h > h^*)$ **then**
9: $(t^*, h^*, \mathcal{P}^*) = (t, h, \mathcal{P})$
10: **return** $(t^*, h^*, \mathcal{P}^*)$

procedure selectRandomTest
Input: Attribute a and partition $\mathcal{P}$
Output: A test t

1: $t = none$
2: $A_v = getAttributeValues(a, \mathcal{P})$
3: **if** a is numerical **then**
4: $a_M = getMaxValue(A_v)$
5: $a_m = getMinValue(A_v)$
6: $a_c = rndCutPoint(a_m, a_M)$
7: $t = a < a_c$
8: **if** a is categorical **then**
9: $A_s = rndNonEmptySet(A_v, \mathcal{P})$
10: $t = a \in A_s$
11: **return** t

where $Var(E, Y_j)$ is the normalized variance (according to the $min - max$ normalization function) of the variable Y_j in the set E. The variances of the target variables are normalized so that each target variable contributes equally to the overall variance. This is due to the fact that the target variables can have completely different ranges.

Obviously, the smaller the variance reduction (h in the procedure $FindTest$ - see Table 2) the better the split. If we set the value of k to 1, this algorithm works in the same way of the Random Tree algorithm proposed in [20]. The advantage with respect to the Random Tree algorithm is that in the approach we adopt there is still a non-random selection based on some evaluation measure (i.e., variance reduction).

The extremely randomized PCTs are very unstable predictive models because of the intense randomization at each node. Consequently, such PCTs are only meaningful when used in combination with an ensemble learning framework. In this work, we construct ensembles of extremely randomized PCTs (Extra-PCTs) by following the same ensemble learning approach proposed in [12] where, however, PCTs are not used and, consequently, it is not possible to directly follow a global approach and naturally consider the multi-target regression task.

Each of the base predictive models is constructed using the complete training set and each of them uses different, randomly selected, attributes in the nodes. The number of attributes (k) that are retained is given by a function of the total number of descriptive attributes D (e.g., $k = 1$, $k = \lfloor \sqrt{D} + 1 \rfloor$, $f(D) = \lfloor log_2(D) + 1 \rfloor$, $k = D \ldots$). Depending on the application, one can select to use different values for k. In this study, we investigate the effect of the function used to initialize k on the performance of the ensemble for MTR.

In the Extra-PCTs ensemble, the prediction for a new instance is obtained by combining the predictions of all the base predictive models. For the MTR task, the prediction for each target variable is computed as the average of the predictions obtained from each tree. Note that this solution exploits possible dependencies in the output space since clusters used for prediction (and their hierarchical organization, i.e., the tree) have been built by taking into account the whole output space.

One of the strong advantages of the Extra-PCTs ensembles is their computational efficiency. In [1], we discuss the computational cost of an ordinary PCTs and ensembles of PCTs extensively. The computational cost of constructing an ordinary PCT for predicting multiple target variables can be summarized as

$$\mathcal{O}(DN \log^2 N) + \mathcal{O}(SDN \log N) + \mathcal{O}(N \log N),$$

where D is the number of descriptive attributes, N is the number of examples and S is the number of target variables.

The cost of constructing a Extra-PCTs can be derived as follows.Two procedures are executed at each node of the tree and they include: calculating the best split out of the k randomly selected, candidate splits at a cost of $\mathcal{O}(kSN)$, and applying the split to the training instances with a cost of $\mathcal{O}(N)$. Furthermore, we assume, as in [20], that the tree is balanced and bushy. This means that the depth of the tree is in the order of $\log N$, i.e., $\mathcal{O}(\log N)$. Having this in mind, the total computational cost of constructing a single tree is

$$\mathcal{O}(kS \log N) + \mathcal{O}(N \log N).$$

Comparing the two costs, we can note that Extra-PCTs have much lower computational complexity as compared to regular PCTs. The ensembles usually amplify the computational cost of the base predictive models linearly with the number of base models. Consequently, the cost of an Extra-PCTs ensemble will be much lower than the cost of a regular ensemble.

3 Experimental Design

We construct several types of trees and ensembles thereof. First, we construct PCTs that predict a separate tree for each variable from the multiple target variables. Second, we learn PCTs that predict all of the target variables simultaneously. Finally, we construct the ensemble models in the same manner by using both random forests and the Extra-PCTs algorithm.

3.1 Experimental Questions

We consider three aspects of constructing tree ensembles with the Extra-PCTs algorithm for predicting multiple target variables: convergence, predictive performance and efficiency. We first investigate the saturation/convergence of the predictive performance of global and local ensembles with respect to the number of base predictive models they consist of. Namely, we inspect the predictive

performance of the ensembles at different ensemble sizes (i.e., we construct saturation curves). The goal is to check which type of Extra-PCTs ensembles, global or local, saturates at a smaller number of trees.

We next assess the predictive performance of global and local Extra-PCTs ensembles and investigate whether global and local ensembles have better predictive performance than the respective single model counterparts. Moreover, we check whether the exploitation of the multiple targets can lift the predictive performance of an Extra-PCTs ensemble (i.e., global versus local ensembles). Furthermore, we compare the performance of the Extra-PCTs ensembles with the performance of a random forest ensemble of PCTs. Random forests of PCTs are considered among the state-of-the-art predictive modelling techniques [1]. Finally, we assess the efficiency of both global and local single predictive models and ensembles thereof by comparing the running times for and the sizes of the models obtained by the different approaches.

3.2 Data Description

The datasets with multiple continuous targets used in this study are 13 in total and are mainly from the domain of ecological modelling. Table 3 outlines the properties of the datasets. The selection of the datasets contain datasets with various number of examples described with various number of attributes. For more details on the datasets, we refer the reader to the referenced literature.

Table 3. Properties of the datasets with multiple continuous targets (regression datasets); N is the number of instances, $\overline{D/C}$ the number of descriptive attributes (discrete/continuous), and T the number of target attributes.

Name of dataset	N	$\lvert D\rvert/\lvert C\rvert$	T
Collembola [21]	393	8/39	3
EDM [22]	154	0/16	2
Forestry-Slivnica-LandSat [23]	6218	0/150	2
Forestry-Slivnica-IRS [23]	2731	0/29	2
Forestry-Slivnica-SPOT [23]	2731	0/49	2
Sigmea real [24]	817	0/4	2
Soil quality [2]	1944	0/142	3
Solar-flare 1 [25]	323	10/0	3
Solar-flare 2 [25]	1066	10/0	3
Water quality [26]	1060	0/16	14

3.3 Experimental Setup

Empirical evaluation is the most widely used approach for assessing the performance of machine learning algorithms, that is based on the 10-fold cross-

validation evaluation strategy. The performance of the algorithms are assessed using some evaluation measures and, in particular, since the task we consider is that of MTR, we employed three well known measures: the correlation coefficient (CC), root mean squared error ($RMSE$) and relative root mean squared error ($RRMSE$). We present here only the results in terms of $RRMSE$, but similar conclusions hold for the other two measures.

Next, we define the parameter values used in the algorithms for constructing the single trees and the ensembles of PCTs. The single trees (both for multi-target and single-target regression) are obtained using F-test pruning. This pruning procedure uses the exact Fisher test to check whether a given split/test in an internal node of the tree results in a reduction in variance that is statistically significant at a given significance level. If there is no split/test that can satisfy this, then the node is converted to a leaf. An optimal significance level was selected by using internal 3-fold cross validation, from the following values: 0.125, 0.1, 0.05, 0.01, 0.005 and 0.001.

The construction of both ensemble methods takes, as an input parameter, the size of the ensemble, i.e., number of base predictive models to be constructed. We constructed ensembles with 10, 25, 50, 75, 100, 150 and 250 base predictive models. Following the findings from the study conducted by Bauer and Kohavi [27], the trees in the ensembles were not pruned.

Both the Extra-PCTs ensemble and the random forests algorithm take as input the size of the feature subset that is randomly selected at each node. For the Extra-PCTs ensemble, we follow the recommendations from Geurts et al. [12], and set the value of k to the number of descriptive attributes, i.e., $k = D$. For the random forests of PCTs, we apply the logarithmic function of the number of descriptive attributes $\lfloor \log_2 |D| \rfloor + 1$, which is recommended by Breiman [11].

In order to assess the statistical significance of the differences in performance of the studied algorithms, we adopt the recommendations by Demšar [28] for the statistical evaluation of the results. In particular, we use the Friedman test for statistical significance. Afterwards, to check where the statistically significant differences appear (between which algorithms), we use two post-hoc tests. First, we use Bonferroni-Dunn test to compare the best performing method with the remaining methods. Second, we use Nemenyi post-hoc test when we compare all of the methods among each other. We present the results from the statistical analysis with *average ranks diagrams* [28]. The diagrams plot the average ranks of the algorithms and connect the ones whose average ranks are smaller than a given value, called critical distance. The critical distance depends on the level of the statistical significance, in our case 0.05. The difference in the performance of the algorithms connected with a line is not statistically significant at the given significance level.

4 Results and Discussion

We analyze the results from the experiments along three dimensions. First, we present the saturation curves of the ensemble methods (both for multi-target

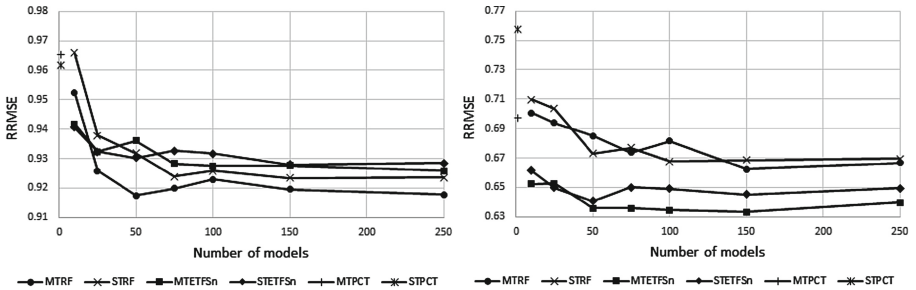
and single-target regression). We also compare single trees vs. ensembles of trees. Next, we compare models that predict the complete structured output vs. models that predict components of the structured output. Finally, we evaluate the algorithms by their efficiency in terms of running time and model size.

In Fig. 1, we present the saturation curves for the ensemble methods for multi-target regression. Although these curves are averaged across all target variables for a given dataset, they still provide useful insight into the performance of the algorithms. First, the curves show that for part of the datasets the ensembles reach their optimal performance when just as few as 25 base predictive models are constructed.

Second, we note that on majority of the datasets the proposed Extra-PCTs ensembles outperform the random forests of PCTs across all ensemble sizes. The most notable improvements are for the following datasets: *EDM*, *Forestry-Slivnica-LandSat*, *Forestry-Slivnica-SPOT* and *Soil quality*. The worst performance of the Extra-PCTs ensembles as compared with the random forests is for the dataset *Sigmea real*. For this, dataset the Extra-PCTs ensembles perform worse even than a single PCT. This may be due to the fact that this dataset has only 4 descriptive variables and the extreme randomization used in the Extra-PCTs ensembles hurts the predictive performance of the ensemble and misses on a crucial information. More specifically, the extreme randomization in this case decreases the variance only slightly while it increases the bias significantly (similarly as observed in [12]). Furthermore, on the datasets containing only categorical descriptive variables (*Solar-flare1* and *Solar-flare2*) both the Extra-PCTs ensembles and random forests perform poorly and their performance is worse than the performance of a single tree.Finally, in the case of mixed numeric and categorical variables (*Collembola* dataset) the multi-target random forests are the best performing method. The application of the proposed Extra-PCTs ensembles on datasets with categorical variables prompts further investigation.

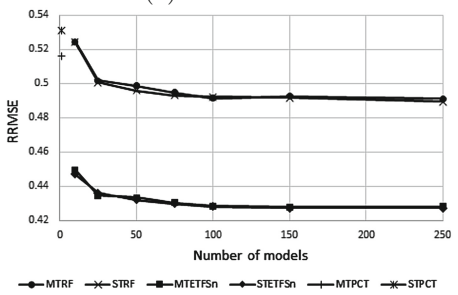
Next, we perform statistical tests to detect up to which point the improvement is no longer statistically significant. To this end, we used Friedman test with Bonferroni-Dunn post-hoc test. We center the Bonferroni-Dunn test around the best performing ensemble size and check until which size the performance does not degrade statistically significantly. The results are presented in Fig. 2. From the diagrams, we can note that the multi-target Extra-PCTs ensembles achieve optimal performance with 75 base predictive models added to the ensemble. The remaining methods, multi-target and single-target random forests and single-target Extra-PCTs ensembles, require 100 base predictive models to achieve their optimal performance. This means that the global Extra-PCTs ensembles achieve their optimal performance with fewer trees added as compared with the local Extra-PCTs ensembles. Considering this, we perform the statistical analysis on ensembles with both 75 and 100 base predictive models.

Figure 3 gives the average rank diagrams of the different ensemble methods and the single-tree models. The results for ensembles with both 75 and 100 base predictive models show that the differences in predictive performance among

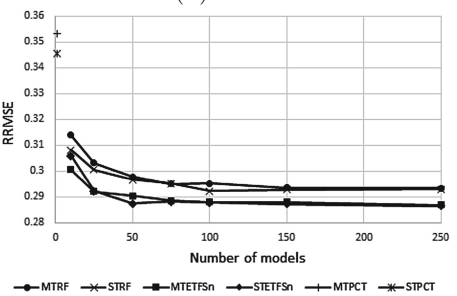


(a) Collembola

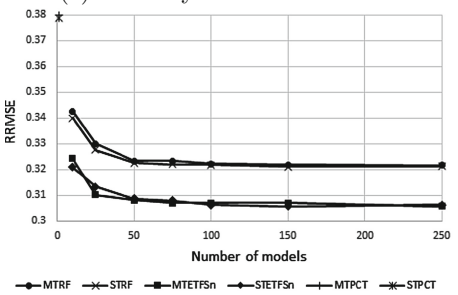
(b) EDM



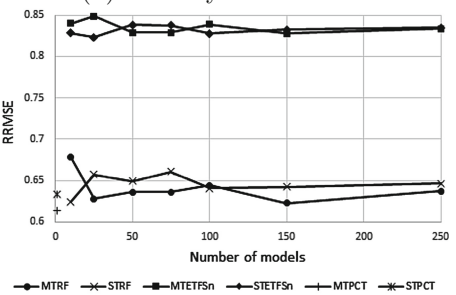
(c) Forestry-Slivnica-LandSat



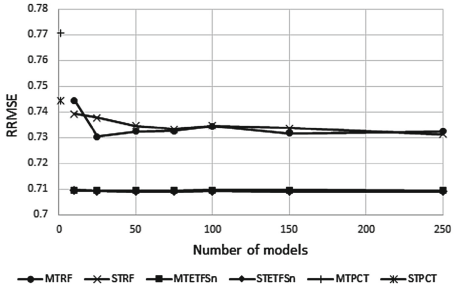
(d) Forestry-Slivnica-IRS



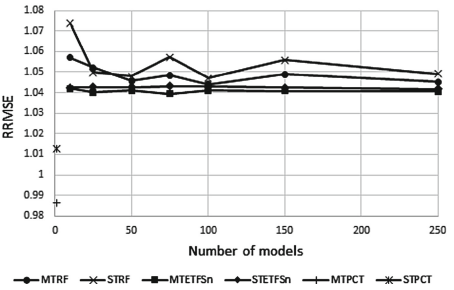
(e) Forestry-Slivnica-SPOT



(f) Sigmea real

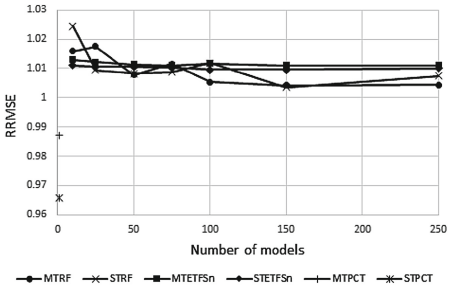


(g) Soil quality

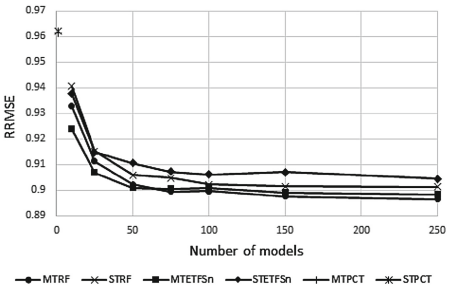


(h) Solar-flare 1

Fig. 1. Saturation curves for the two ensemble methods for MTR. Note that the scale of the y-axis is adapted for each curve. The algorithm names are abbreviated as follows: Predictive clustering trees - *PCT*, Extra-PCTs - *ET*, random forests - *RF*, multi-target prediction - *MT* and single-target prediction - *ST*.



(i) Solar-flare 2

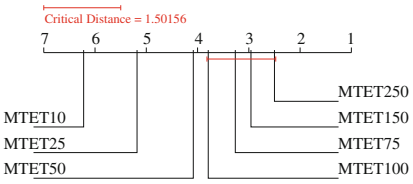


(j) Water quality

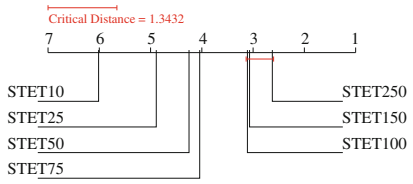
Fig. 1. (*continued*)

the different ensemble methods are not statistically significant at the level of 0.05. However, the multi-target Extra-PCTs ensembles are the best performing method. Furthermore, the difference in performance between ensembles and single multi-target and single-target PCTs is statistically significant.

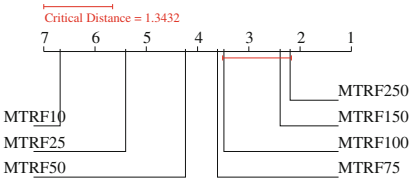
Finally, we compare the algorithms by their running time and the size of the models for ensembles of 50 trees (see Fig. 4). The statistical tests show that, in terms of the time efficiency, the multi-target Extra-PCTs ensembles are the fastest method. Moreover, they significantly outperform both ensemble methods predicting the targets separately. The diagram also shows that the global (multi-target) ensembles are clearly more efficient than the local (single-target) ensembles. The multi-target Extra-PCTs are faster than multi-target random



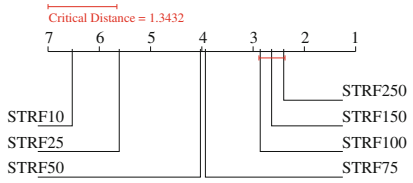
(a) Multi-target Extra-PCTs



(b) Single-target Extra-PCTs

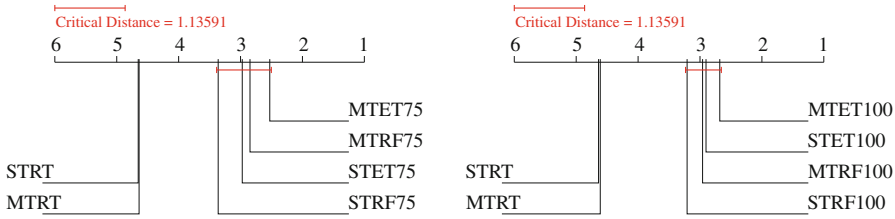


(c) Multi-target random forest of PCTs



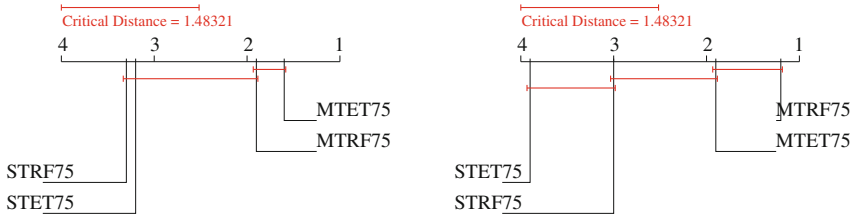
(d) Single-target random forest of PCTs

Fig. 2. Average rank diagrams for the ensembles constructed with varying number of base predictive models. The critical distance is set for a significance level at 0.05. The differences in performance of the algorithms connected with a line are not statistically significant.



(a) 75 base predictive models (b) 100 base predictive models

Fig. 3. Average rank diagrams for the various ensembles consisting of (a) 75 and (b) 100 base predictive models. The critical distance is set for a significance level at 0.05. The differences in performance of the algorithms connected with a line are not statistically significant.



(a) Time efficiency (b) Size of the models

Fig. 4. Efficiency (running time and model size) of the ensembles for MTR. The size of the ensembles is 75 trees, however the same conclusions hold across all ensemble sizes. The critical distance is set for a significance level at 0.05. The differences in performance of the algorithms connected with a line are not statistically significant.

forests ~ 1.77 times. The computational advantage is even more pronounced in the datasets with more examples. In terms of model size, the multi-target random forests are the best performing method. Both global ensembles are clearly better than their local counterparts. The results for the efficiency of the methods given in Fig. 4 show that the computational efficiency of the multi-target Extra-PCTs ensembles comes at the price of constructing slightly larger models. Namely, due to the increased randomness as compared to the random forests method fewer test are being evaluated (i.e., smaller computational cost) but, in the same time, this means that the constructed (PCT) models will grow larger.

5 Conclusions

In this work, we address the task of learning ensembles of predictive models for predicting multiple continuous variables, i.e., multi-target regression. In contrast to standard regression, where the output is a single scalar value, in MTR the output is a data structure – a tuple/vector of continuous variables. We consider both global and local prediction of the multiple variables, the former based on

a single model that predicts all of the target variables simultaneously and the latter based on a collection of models, each predicting a single target variable.

Ensembles have proved to be highly effective methods for improving the predictive performance of their constituent models, especially for classification and regression tree models. In particular, we consider the Extra-Tree ensembles as predictive models. Extra-Tree ensembles are a well established method for predictive modelling that has been successfully applied to computer vision and, especially, gene network inference. This approach has been the overall top performer in the DREAM4 and DREAM5 challenges for gene network reconstruction. Following this, we extend this method for the task of multi-target regression and call the Extra-PCTs ensembles. As base predictive models, we propose to use predictive clustering trees (PCTs). These can be considered as a generalization of decision trees for predicting structured outputs, including multiple continuous variables.

We conduct an experimental evaluation of the proposed method on a collection of 10 benchmark datasets for with multiple continuous targets. We make several comparisons. First, we investigate the influence of the number of base predictive models in an ensemble to its predictive performance. Second, we compare the performance of multi-target Extra-PCTs ensembles with the performance of single-target Extra-PCTs ensembles. Next, we compare the multi-target Extra-PCTs ensembles with multi-target and single-target random forests of PCTs. Random forests are considered among the state-of-the-art modelling techniques. Furthermore, we compare the efficiency of the different approaches.

The results reveal the following. First, the performance of the multi-target Extra-PCTs ensembles starts to saturate as soon as even only 25 base predictive models are added to the ensemble. Moreover, after adding 75 base predictive models, the performance of a multi-target Extra-PCTs ensemble does not change statistically significantly. Second, a multi-target Extra-PCTs ensemble performs statistically significantly better than a single multi-target or single-target PCT. Next, the performance among the different ensemble learning methods is not statistically significantly different, while multi-target Extra-PCTs ensembles are the best performing method. Finally, in terms of efficiency (running times and model complexity), both multi-target variants of the ensemble methods are more efficient and produce smaller models as compared to the single-target ensembles.

We plan to extend the work along four major dimensions. First, we will extend the proposed algorithm to cover other tasks of structured output prediction, such as multi-target classification, multi-label classification and hierarchical multi-label classification. Second, we will adapt the feature ranking mechanism of the Extra-Trees algorithm for different types of structured outputs. Next, we will perform a more extensive study on the sensitivity of the algorithm of its parameter k and the influence of categorical variables in the dataset to the ensembles' performance. Finally, we will perform a more extensive experimental evaluation by using a larger number of benchmarking datasets.

Acknowledgments. We acknowledge the financial support of the European Commission through the grants ICT-2013-612944 MAESTRA and ICT-2013-604102 HBP.

References

1. Kocev, D., Vens, C., Struyf, J., Džeroski, S.: Tree ensembles for predicting structured outputs. Pattern Recogn. **46**(3), 817–833 (2013)
2. Demšar, D., Džeroski, S., Larsen, T., Struyf, J., Axelsen, J., Bruns-Pedersen, M., Krogh, P.H.: Using multi-objective classification to model communities of soil. Ecol. Model. **191**(1), 131–143 (2006)
3. Stojanova, D., Panov, P., Gjorgjioski, V., Kobler, A., Džeroski, S.: Estimating vegetation height and canopy cover from remotely sensed data with machine learning. Ecol. Inform. **5**(4), 256–266 (2010)
4. Kocev, D., Džeroski, S., White, M., Newell, G., Griffioen, P.: Using single- and multi-target regression trees and ensembles to model a compound index of vegetation condition. Ecol. Model. **220**(8), 1159–1168 (2009)
5. Tsoumakas, G., Spyromitros-Xioufis, E., Vrekou, A., Vlahavas, I.: Multi-target regression via random linear target combinations. In: Calders, T., Esposito, F., Hüllermeier, E., Meo, R. (eds.) ECML PKDD 2014, Part III. LNCS, vol. 8726, pp. 225–240. Springer, Heidelberg (2014)
6. Bakır, G.H., Hofmann, T., Schölkopf, B., Smola, A.J., Taskar, B., Vishwanathan, S.V.N.: Predicting Structured Data. Neural Information Processing. The MIT Press, Cambridge (2007)
7. Struyf, J., Džeroski, S.: Constraint based induction of multi-objective regression trees. In: Bonchi, F., Boulicaut, J.-F. (eds.) KDID 2005. LNCS, vol. 3933, pp. 222–233. Springer, Heidelberg (2006)
8. Appice, A., Džeroski, S.: Stepwise induction of multi-target model trees. In: Kok, J.N., Koronacki, J., Lopez de Mantaras, R., Matwin, S., Mladenič, D., Skowron, A. (eds.) ECML 2007. LNCS (LNAI), vol. 4701, pp. 502–509. Springer, Heidelberg (2007)
9. Kocev, D., Vens, C., Struyf, J., Džeroski, S.: Ensembles of multi-objective decision trees. In: Kok, J.N., Koronacki, J., Lopez de Mantaras, R., Matwin, S., Mladenič, D., Skowron, A. (eds.) ECML 2007. LNCS (LNAI), vol. 4701, pp. 624–631. Springer, Heidelberg (2007)
10. Breiman, L.: Bagging predictors. Mach. Learn. **24**(2), 123–140 (1996)
11. Breiman, L.: Random forests. Mach. Learn. **45**(1), 5–32 (2001)
12. Geurts, P., Ernst, D., Wehenkel, L.: Extremely randomized trees. Mach. Learn. **36**(1), 3–42 (2006)
13. Maree, R., Geurts, P., Piater, J., Wehenkel, L.: Random subwindows for robust image classification. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), vol. 1, pp. 34–40 (2005)
14. Ruyssinck, J., Huynh-Thu, V.A., Geurts, P., Dhaene, T., Demeester, P., Saeys, Y.: NIMEFI: gene regulatory network inference using multiple ensemble feature importance algorithms. PLoS ONE **9**(3), 1–13 (2014)
15. Huynh-Thu, V.A., Irrthum, A., Wehenkel, L., Geurts, P.: Inferring regulatory networks from expression data using tree-based methods. PLoS ONE **5**(9), 1–10 (2010)
16. Kocev, D.: Ensembles for Predicting Structured Outputs. Ph.D. thesis, Jožef Stefan International Postgraduate School, Ljubljana, Slovenia (2011)

17. Stojanova, D., Ceci, M., Malerba, D., Deroski, S.: Using PPI network autocorrelation in hierarchical multi-label classification trees for gene function prediction. BMC Bioinform. **14**, 285 (2013)
18. Blockeel, H., Struyf, J.: Efficient algorithms for decision tree cross-validation. J. Mach. Learn. Res. **3**, 621–650 (2002)
19. Breiman, L., Friedman, J., Olshen, R., Stone, C.J.: Classification and Regression Trees. Chapman & Hall/CRC, Boca Raton (1984)
20. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann, San Francisco (2005)
21. Kampichler, C., Džeroski, S., Wieland, R.: Application of machine learning techniques to the analysis of soil ecological data bases: relationships between habitat features and Collembolan community characteristics. Soil Biol. Biochem. **32**(2), 197–209 (2000)
22. Karalič, A.: First order regression. Ph.D. thesis, Faculty of Computer Science, University of Ljubljana, Ljubljana, Slovenia (1995)
23. Stojanova, D.: Estimating forest properties from remotely sensed data by using machine learning. Master's thesis, Jožef Stefan International Postgraduate School, Ljubljana, Slovenia (2009)
24. Demšar, D., Debeljak, M., Džeroski, S., Lavigne, C.: Modelling pollen dispersal of genetically modified oilseed rape within the field. In: The Annual Meeting of the Ecological Society of America, p. 152 (2005)
25. Asuncion, A., Newman, D.: UCI - Machine Learning Repository (2007). http://www.ics.uci.edu/mlearn/MLRepository.html
26. Džeroski, S., Demšar, D., Grbović, J.: Predicting chemical parameters of river water quality from bioindicator data. Appl. Intell. **13**(1), 7–17 (2000)
27. Bauer, E., Kohavi, R.: An empirical comparison of voting classification algorithms: bagging, boosting, and variants. Mach. Learn. **36**(1), 105–139 (1999)
28. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. J. Mach. Learn. Res. **7**, 1–30 (2006)