Semi-supervised learning for multi-target regression

Jurica Levatić[1,2], Michelangelo Ceci[3], Dragi Kocev[1], and Sašo Džeroski[1,2]

[1] Department of Knowledge Technologies, Jožef Stefan Institute, Ljubljana, Slovenia
[2] Jožef Stefan International Postgraduate School, Ljubljana, Slovenia
[3] Department of Computer Science, University of Bari Aldo Moro, Bari, Italy
name.surname@ijs.si, michelangelo.ceci@uniba.it

Abstract. The most common machine learning approach is supervised learning, which uses labeled data for building predictive models. However, in many practical problems, the availability of annotated data is limited due to the expensive, tedious and time-consuming annotation procedure. At the same, unlabeled data can be easily available in large amounts. This is especially pronounced for predictive modelling problems with structured output space. Semi-supervised learning (SSL) aims to use unlabeled data as an additional source of information in order to build better predictive models than can be learned from labeled data alone. The majority of work in SSL considers the simple tasks of classification and regression where the output space consists of a single variable. Much less work has been done on SSL for structured output prediction. In this study, we address the task of multi-target regression (MTR), a type of structured output where the output space consists of multiple numerical values. Our main objective is to investigate whether we can improve over supervised methods for MTR by using unlabeled data. We use ensembles of predictive clustering trees in a self-training fashion: most reliable predictions on unlabeled data are iteratively used to re-train the model. We use variance of an ensemble models as an indicator of the reliability of predictions. Our results provide a proof-of-concept: Unlabeled data improves predictive performance of ensembles for multi-target regression, however further efforts are needed to automatically select the optimal threshold for reliability of predictions.

Keywords: semi-supervised learning, self-training, multi-target, multi-output, multivariate, regression, ensembles, structured outputs, PCTs

1 Introduction

The major machine learning paradigms are supervised learning (e.g., classification, regression), where all the data are labeled, and unsupervised learning (e.g., clustering) where all the data are unlabeled. Semi-supervised learning (SSL) [1] examines how to combine both paradigms and exploit both labeled and unlabeled data, aiming to benefit from the information that unlabeled data can bring. SSL is of important practical value since the following scenario can often

be encountered: labeled data are scarce and hard to get because they require human experts, expensive devices or time-consuming experiments, while, at the same time, unlabeled data abound and are easily obtainable. Real-world classification problems of this type include: phonetic annotation of human speech, protein 3D structure prediction, and spam filtering. Intuitively, SSL yields best results when there are few labeled examples as compared to unlabeled ones (i.e., large-scale labelling is not affordable). Such a scenario is in particular relevant for machine learning tasks with complex (structured) outputs, where providing the labels of data is a laborious and/or an expensive process, while at the same time large amounts of unlabeled data are readily available.

In this study, we are concerned with the semi-supervised learning for *multi-target regression* (MTR). MTR is a type of structured output prediction task where the goal is to predict multiple continuous target variables (also known as multi-output or multivariate regression). In many real life problems, 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 forest [3]. There are several advantages of learning a multi-target (i.e., global) model over learning a separate (i.e., local) model for each target variable: Global models are typically easier to interpret, perform better and overfit less than collection of single-target models [4]. In the past, classical (single-target) regression received much more attention than MTR, however several researchers proposed methods for solving the task of MTR directly and demonstrated their effectiveness [5–8].

Semi-supervised methods able to solve MTR problems are scarce. Most commonly, SSL methods for structured output prediction are dealing with discrete outputs. Here, prominent work was done by Brefeld [9], who used co-training paradigm and the principle of maximizing the consensus among multiple independent hypotheses to develop semi-supervised support vector learning algorithm for joint input-output spaces and arbitrary loss. Zhang and Yeung [10] proposed a semi-supervised method based on Gaussian processes for a task related to MTR: multi-task regression. In multi-task learning the aim is to predict multiple single-target variables with different training sets (in general, with different descriptive attributes) at the same time. Also related, Navaratnam et al. [11] proposed a semi-supervised method for multivariate regression specialized for computer vision. The goal is to relate features of images (z) to joint angles (θ). Unlabeled examples are used to help the fitting of the joint density $p(z, \theta)$.

In this work, we propose a self-training approach [12] (also called self-teaching or bootstrapping) for performing SSL for MTR. As a base predictive model, we use predictive clustering trees (PCTs), or more precisely, random forest of predictive clustering trees for MTR [8]. PCTs are a generalization of standard decision trees towards predicting structured outputs. They are able to make predictions for several types of structured outputs [8]: tuples of continuous/discrete variables, hierarchies of classes and time series.

The main feature of self-training is that it iteratively uses its own most reliable predictions in the learning process. The most reliable predictions are

selected by using a threshold on the reliability scores. The main advantage of the iterative semi-supervised learning approach is that it can be "wrapped" around any existing (supervised) method. The only prerequisite is that the underlying method is able to provide a reliability score for its predictions. With our base predictive models, i.e., random forest of PCTs for MTR, this score is estimated by using the variance of the votes from the ensemble members of an example.

The concept of self-training was first proposed by Yarowsky [13] for word sense disambiguation, i.e., deciding the meaning of a homonym in a given context. Other successful applications of self training include: detection of objects on image [14], identification of subjective nouns [15] and learning human motion over time [16]. There are several examples of methods based on self-training (or based on closely related co-training) implemented for solving the task of (single-target) regression [17–21]. To the best of our knowledge, self-training was not implemented yet for the problem of multi-target regression.

The main purpose of this study is to investigate the following question: Can unlabeled data improve predictive performance of the models for MTR in a self-training setting? To this end, we compared our semi-supervised method to its supervised counterpart in the following evaluation scenario: We consider the best result (considering different thresholds for reliability score) of semi-supervised method. Results show that the proposed semi-supervised method is able to improve over supervised random forest in 4 out of 6 considered datasets. Thus, the evaluation provides a positive answer to our research question posed above, and motivates further research efforts in this direction.

2 Semi-supervised learning with ensembles of PCTs

The basis of the semi-supervised method proposed in this study is the use, in an ensemble learning fashion, of predictive clustering trees (PCTs). In this section, we first briefly describe PCTs for multi-target regression, followed by a description of the method for learning random forest. We then present in details the adaptation of semi-supervised self-training approach for multi-target regression with random forest of PCTs.

2.1 Predictive clustering trees for MTR

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 is recursively partitioned into smaller clusters while moving down the tree. The PCT framework is implemented in the CLUS system [22], which is available for download at `http://clus.sourceforge.net`.

PCTs are induced with a standard *top-down induction of decision trees* (TDIDT) algorithm [23] (see Table 1). It takes as input a set of examples (E) and outputs a tree. The heuristic (h) that is used for selecting the tests (t) is the reduction in variance caused by the partitioning ($\mathcal{P}$) of the instances corresponding to the tests (t) (see line 4 of the BestTest procedure in Table 1). By

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

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

procedure BestTest
Input: A dataset E
Output: the best 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: **for each** possible test t **do**
3: $\mathcal{P}$ = partition induced by t on E
4: $h = Var(E) - \sum_{E_i \in \mathcal{P}} \frac{|E_i|}{|E|} Var(E_i)$
5: **if** $(h > h^*) \wedge \text{Acceptable}(t, \mathcal{P})$ **then**
6: $(t^*, h^*, \mathcal{P}^*) = (t, h, \mathcal{P})$
7: **return** $(t^*, h^*, \mathcal{P}^*)$

maximizing the variance reduction, the cluster homogeneity is maximized and the predictive performance is improved.

The main difference between the algorithm for learning PCTs and a standard decision tree learner is that the former considers the variance function and the prototype function (that computes a label for each leaf) as *parameters* that can be instantiated for a given learning task. So far, PCTs have been instantiated for the following tasks [8]: multi-target prediction (which includes multi-target regression), hierarchical multi-label classification and prediction of time-series. In this article, we focus on the task of multi-target regression (MTR).

The variance and prototype functions of PCTs for MTR are instantiated as follows. The variance (used in line 4 of the procedure BestTest in Table 1) is calculated as the sum of the variances of the target variables, i.e., $Var(E) = \sum_{i=1}^{T} Var(Y_i)$, where T is the number of target variables, and $Var(Y_i)$ is the variance of target variable Y_i. The variances of the targets are normalized, so each target contributes equally to the overall variance. The normalization is performed by dividing with the estimates with the standard deviation for each target variable on the available training set. The prototype function (calculated at each leaf) returns as a prediction the mean values of the target variables, calculated by using the training instances that belong to the given leaf.

2.2 Ensembles of PCTs

We consider random forest of PCTs for structured prediction, as suggested by Kocev et al. [8] in the CLUS system. The PCTs in the random forest are constructed by using the random forests method given by Breiman [24]. The algorithm of this ensemble learning method is presented in Table 2, left.

A random forest (Table 2, left) is an ensemble of trees, where diversity among the predictors is obtained by using bootstrap replicates and additionally by changing the set of descriptive attributes during learning. Bootstrap samples are obtained by randomly sampling training instances, with replacement, from the original training set, until an equal number of instances as in the training

Table 2. The learning algorithms for random forests and semi-supervised self-training (CLUS-SSL). Here, E_l is the set of the labeled training examples, E_u is a set of unlabeled examples, k is the number of trees in the forest, $f(D)$ is the size of the feature subset considered at each node during tree construction for random forests and τ is the threshold for reliability of predictions.

procedure RForest($E_l, k, f(D)$)
returns Forest
1: $F = \emptyset$
2: **for** $i = 1$ **to** k **do**
3: $E_i = bootstrap(E_l)$
4: $T_i = PCT\_rnd(E_i, f(D))$
5: $F = F \bigcup \{T_i\}$
6: **return** F

procedure CLUS-SSL(E_l, E_u, τ, k, f(D))
returns Forest
1: **repeat**
2: F = RForest($E_l, k, f(D)$)
3: predict(F, E_u)
4: **for each** $e_u \in E_u$ **do**
5: **if** Reliability(F, e_u) $\geq \tau$ **then**
6: move e_u from E_u to E_l
7: **until** No example e_u is moved from E_u to E_l

set is obtained. Breiman [25] showed that bagging can give substantial gains in predictive performance, when applied to an unstable learner (i.e., a learner for which small changes in the training set result in large changes in the predictions), such as classification and regression tree learners.

To learn a random forest, the classical PCT algorithm for tree construction (Table 1) is replaced by PCT_rnd which replaces the standard selection of attributes with a randomized selection. More precisely, at each node in the decision trees, a random subset of the descriptive attributes is taken, and the best attribute is selected from this subset. The number of attributes that are retained is given by a function f of the total number of descriptive attributes D (e.g., $f(D) = 1$, $f(D) = \lfloor \sqrt{D} + 1 \rfloor$, $f(D) = \lfloor log_2(D) + 1 \rfloor \ldots$). The reason for random selection of attributes is to avoid (possible) correlation of the trees in a bootstrap sample. Namely, if only few of the descriptive attributes are important for prediction of target variables, these will be selected by many of the bootstrap tress, generating highly correlated trees.

In the random forest of PCTs, the prediction for a new instance is obtained by combining the predictions of all the base predictive models. For the MTR task, the predictions for each target variable is computed as the average of the predictions obtained from each tree.

2.3 Self-training for MTR

To perform semi-supervised learning with ensembles of PCTs for MTR, we consider a self-training approach. In self-training, first a predictive model (i.e., a random forest of PCTs) is constructed by using the available labeled instances. The unlabeled instances are then labeled by using the obtained predictive model. Next, the examples with the most reliable predictions are selected and then added to the training set. A predictive model is again constructed and the procedure is repeated until a stopping criterion is satisfied.

To adapt the self-training procedure to the MTR task, we need to define: *i)* a reliability measure of the predictions, *ii)* a criterion to select the most reliable predictions and *iii)* a stopping criterion. Since self-training relies on the assumptions that its own (most reliable) predictions are correct, the most crucial part of the algorithm is the definition of a good reliability measure. This measure should be able to discern correct (with high reliability score) from wrong (with low reliability score) predictions. At this purpose, we exploit a solution provided directly with the ensemble learning – we use the variance of the votes of an ensemble as an indicator of reliability.

When an unlabeled example is predicted by a random forest, we consider the prediction reliable if predictions of individual trees (i.e., votes) in the ensemble are coherent. Otherwise, if the predictions by individual trees in the ensemble are very heterogeneous, we consider the prediction unreliable. The variance has been previously used in bagging where it has been found to perform the best in an extensive empirical comparison of various approaches for estimating reliability of regression predictions [26].

Here we present the procedure for calculation of reliability score in more detail. Formally, for each iteration of the self-training algorithm, we have to solve an MTR problem with m continuous target variables by learning a random forest ensemble F with k trees. These trees are trained on a set of labeled examples E_l and applied on a set of unlabeled examples E_u. First, for each unlabeled example $e_u \in E_u$, per-target standard deviation of votes of ensemble r_u^i is calculated as:

$$r_u^i = \sqrt{\frac{1}{k-1}\sum_{j=1}^{k}\left(tree_j^i(e_u) - F^i(e_u)\right)^2}, \quad i = 1 \dots m,$$

where $tree_j^i$ is a vote (i.e., a prediction score) for e_u returned by the j^{th} tree for the i^{th} target. F^i is the prediction for e_u returned by the ensemble for the i^{th} target (i.e., the average of the votes of each tree).

In order to equally weight the contribution of each target attribute in the reliability of the prediction obtained for each unlabeled example, we normalize per-target standard deviations in the interval $[0, 1]$ as follows:

$$\bar{r}_u^i = \frac{r_u^i - \min\limits_{j=1\dots|E_u|} r_j^i}{\max\limits_{j=1\dots|E_u|} r_j^i - \min\limits_{j=1\dots|E_u|} r_j^i}, \qquad i = 1 \dots m.$$

After normalization, the reliability score for an example e_u can be computed by considering the average of the normalized per-target standard deviations:

$$\mathit{Reliability}(F, e_u) = 1 - \frac{1}{m}\sum_{i=1}^{m}\left(\bar{r}_u^i\right)$$

In this formula we have that, a small standard deviation leads to a high score (high reliability).

Table 3. Characteristics of the datasets. N: number of instances, D/C: number of descriptive attributes (discrete/continuous), T: number of target variables.

Dataset	N	D/C	T
Forestry LIDAR IRS [27]	2731	0/29	2
Sigmea real [28]	817	0/4	2
Soil quality [2]	1944	0/142	3
Solar flare-2 [29]	1066	10/0	3
Vegetation clustering [30]	29679	0/65	11
Water quality [31]	1060	0/16	14

The self-training algorithm for MTR with ensembles of PCTs (named CLUS-SSL) is presented in Table 2 (right). To choose which unlabeled examples should be added to the training set we use a user-defined threshold for the reliability score: $\tau \in [0, 1]$. If the reliability of the prediction of an unlabeled example is greater than τ, the example is moved from the unlabeled set (E_u) to the training set (E_l), together with its multi-target predictions. The iterations are repeated until no unlabeled example is moved from the set E_u to the set E_l. This can happen for two reasons, either the set E_u becomes empty, or the reliability score for all the unlabeled examples is smaller than τ.

It is noteworthy that, the combination of random forest and self-training can be considered as a variant of the co-training learning schema where, at each iteration, we do not keep the same views used in the previous iteration and independence among the views is (partially) guaranteed by the ensemble learning approach. This guarantees that the semi-supervised approach can still improve prediction even if, at each iteration, it considers the same features.

3 Experimental design

The semi-supervised method for MTR proposed in this study (CLUS-SSL) iteratively trains random forest tree ensemble for MTR. Thus, we compare the predictive performance of the CLUS-SSL to the performance of a supervised random forest, which is considered as baseline for comparison. The exact evaluation procedure is presented in more details in the remainder of this section.

3.1 Data description

To evaluate the predictive performance of the methods, we use six dataset with multiple continuous target variables. The selected datasets are mainly from the domain of ecological modelling. The main characteristics of the datasets are provided in Table 3. We can observe that the datasets vary in the size, number of attributes and number of target variables.

3.2 Experimental setup and evaluation procedure

Random forests used in the experimental evaluation were constructed with 100 trees. Trees were not pruned and the number of random features used in random forest was set to $\lfloor \log_2(D) + 1 \rfloor$, where D is the total number of features, as recommended by Breiman [24].

To evaluate the predictive performance of the models, we use a procedure similar to 5-fold cross validation, with the difference that the training folds are further partitioned into labeled and unlabeled. More specifically, the data are first randomly divided into 5 folds. Each fold is used once as a test set, and the remaining four folds are used for training. From the training folds, we randomly chose a percentage of the data which serve as labeled examples. We remove the labels of other examples and provide them to the algorithm to serve as unlabeled data during training. Supervised random forests were trained only on the labeled part of the data. The predictive performance reported in the results is the average obtained on the 5 test sets.

To investigate the influence of the amount of labeled data, for each dataset we vary the ratio of labeled versus unlabeled data, where percentage of labeled relative to unlabeled data ranges in the following set: [1%, 3%, 5%, 7%, 10%, 15%, 20%, 30%, 50%].

For the CLUS-SSL algorithm, we need to set the threshold τ for the reliability score, which is used throughout the iterations. For each percentage of labeled data, we tested 15 different thresholds:

$$\tau = \{0.1, 0.2, 0.3, 0.4, 0.5, 0.55, 0.6, 0.65, 0.7, 0.75, 0.8, 0.85, 0.9, 0.95, 0.99\}\,.$$

Therefore, 15 predictive models were built (one model corresponding to one threshold) for each percentage of labeled data. Among these, we report the predictive performance of the best model.

We evaluate the algorithms by using the *root mean square error* (RMSE):

$$RMSE = \sqrt{1/(N * m) * \sum_{i=1}^{m} \sum_{j=1}^{N} \left(a_j^i - p_j^i\right)^2},$$

where m is the number of target variables, N is the number of examples, a_j^i is the real value of the i^{th} target of the j^{th} example, and p_j^i is the predicted value of the i^{th} target of the j^{th} example.

In order to make results comparable across different percentages of labeled examples, we opted to use an evaluation procedure where the test sets are consistent for all the settings. In the results reported in this paper, we consider that the optimal threshold is provided by an 'oracle'. Such threshold selection procedure suffices for answering the research question investigated in this work: *Can unlabeled data potentially improve the predictive performance of models for MTR?* A more general solution for selecting the threshold, would be to use a cross-validation procedure or by implementing smarter thresholding system in self-training which tries to automatically detect the optimal threshold. This aspect is left as future work.

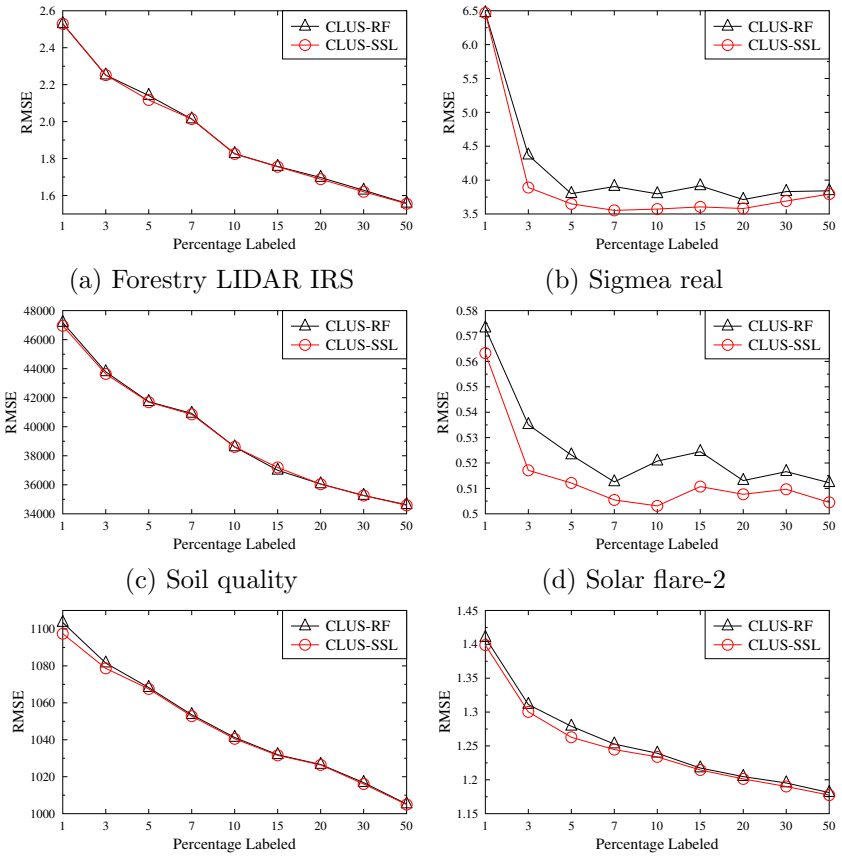

Fig. 1. Comparison of predictive performance of random forest (CLUS-RF) and semi-supervised self-training (CLUS-SSL). Percentage of labeled data varies from 1% to 50%. For each pertentage of labeled data, the best result for CLUS-SSL is presented, considering the different thresholds for confidence of predictions. Optimal threshold is indicated on the plot. CLUS-SSL performs very similar to CLUS-RF (a and c) or improves over CLUS-RF (b, d, e and f).

4 Results and discussion

The results of the experimental evaluation are presented in Figure 1. Their analysis reveals that the proposed semi-supervised method (CLUS-SSL) outperforms its supervised counterpart (CLUS-RF) on 4 out of 6 datasets: Sigmea real, Solar flare-2, Water quality and Vegetation clustering. On the other two datasets (Forestry LIDAR IRS and Soil quality), the two methods perform very similar, with small improvements or degradations in performance made by CLUS-SSL. It was noted before that the success of SSL is domain depended, i.e., methods can behave very differently depending on the nature of the datasets, and that no single SSL method consistently performs better than supervised learning [32].

Table 4. Optimal threshold for reliability of predictions (τ), the percentage of unlabeled examples added to the training set after the completion of the self-training procedure ($\mathcal{E}$) and the number of iterations performed ($\mathcal{I}$) of the CLUS-SSL method.

Dataset		Percentage of labeled data								
		1%	3%	5%	7%	10%	15%	20%	30%	50%
Forestry LIDAR IRS	τ	0.99	0.99	0.9	0.99	0.95	0.95	0.9	0.95	0.95
	$\mathcal{E}$	0%	0%	28%	0%	5%	3%	26%	3%	2%
	$\mathcal{I}$	1	1	80.8	1	26.4	18	57.4	14	8.2
Sigmea real	τ	0.4	0.85	0.6	0.95	0.85	0.55	0.7	0.8	0.65
	$\mathcal{E}$	100%	100%	100%	100%	100%	100%	100%	99%	100%
	$\mathcal{I}$	6.8	9.8	7	16	8.8	6.4	8	5.6	5.2
Soil quality	τ	0.9	0.95	0.7	0.9	0.99	0.95	0.9	0.99	0.95
	$\mathcal{E}$	4%	1%	95%	26%	0%	2%	34%	0%	4%
	$\mathcal{I}$	3.8	2.6	8.8	15	1	2.4	11	1	5.2
Solar flare-2	τ	0.9	0.7	0.8	0.55	0.65	0.75	0.75	0.55	0.9
	$\mathcal{E}$	99%	98%	91%	100%	100%	97%	96%	100%	83%
	$\mathcal{I}$	15.2	5	11	4	5.4	9	7	4.4	9.2
Vegetation clustering	τ	0.1	0.5	0.4	0.95	0.9	0.85	0.9	0.9	0.95
	$\mathcal{E}$	100%	100%	100%	0%	1%	7%	2%	2%	0%
	$\mathcal{I}$	2	7.4	4.6	1.2	32.8	82.6	43.6	52.4	7.2
Water quality	τ	0.3	0.65	0.65	0.5	0.4	0.65	0.5	0.4	0.55
	$\mathcal{E}$	100%	100%	100%	100%	100%	99%	100%	100%	99%
	$\mathcal{I}$	3.2	36.2	32.2	8	3.8	29.8	8.2	4.2	16.8

Results reported in this paper are, thus, consistent with results obtained in previous research on tasks which are different from MTR.

The analysis of the results by varying the percentage of labeled data shows that, as expected, RMSE error decreases with the increase of the percentage of labeled data used to construct the predictive model (better models are learned with more data). However, these trends are not observed across all of the datasets. We can observe the saturation in performance for Sigmea real and Solar flare-2 datasets. There, from about 5% to 7% percent of labeled data, both methods (CLUS-SSL and CLUS-RF) were not able to improve much in the absolute terms. In spite of that, CLUS-SSL is consistently performing better than CLUS-RF, meaning that even in situations where supervised models reached saturation, unlabeled data can further boost the performance. On the other two datasets where unlabeled data helps (Vegetation clustering and Water quality), the improvements of CLUS-SSL over CLUS-RF are more notable with smaller percentages of labeled data. Such behavior is expected, since SSL has the best potential when not much labeled examples are available.

In Table 4, the specific conditions used to obtain and evaluate the CLUS-SSL models (whose performances are depicted in Fig. 1) are given. When observing the variability of the optimal thresholds for the reliability score, we cannot detect regularities. They vary greatly from one dataset to another, and from one percentage of labeled data to another, meaning that it is hard to tell in advance

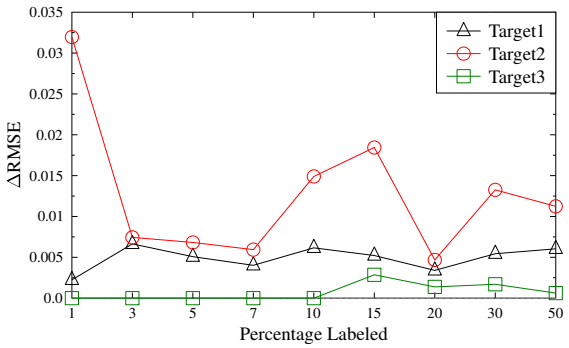

Fig. 2. Analysis of per target performance for the Solar flare-2 dataset, in terms of difference in performance between CLUS-RF and CLUS-SSL (ΔRMSE). Positive values suggest that CLUS-SSL is better, while negative that CLUS-RF is better. Zero means that there is no difference in performance.

which threshold should be used. Self-training can also degrade performance of the underlying method if a sub-optimal threshold is chosen. In particular, if a too permissive threshold is selected, it can allow wrongly predicted examples to enter in the training set. A classification error in the earliest iterations can reinforce itself in the next iterations, leading to a degradation of the performance. On the other hand, if a too stringent threshold is set, it is possible that none, or very few, of the unlabeled examples will enter the training set, meaning that we will miss the opportunity to improve performance with unlabeled data.

Similar observation can be made for the number of performed self-training iterations, they are very heterogeneous regarding the different datasets and percentages of labeled data. Analysis of the number of unlabeled examples added to the training set reveals that, in the cases where semi-supervised learning helps, almost all of the unlabeled examples were moved to the training set at the end of the self-training procedure. This is very consistent across datasets where CLUS-SSL improves over CLUS-RF: Sigmea real, Solar flare-2, Vegetation clustering (for the cases with 1 to 5% percent of labeled data) and Water Quality. The fundamental assumption of self-training is that its most reliable predictions are correct. Thus, the success of this method depends on the ability to learn an accurate model from the data at hand. The assumption is apparently met on the former four cases. Moreover, the (good) predictive ability of the models was retained throughout iterations, as all unlabeled examples were eventually added to the training set. Contrary, if CLUS-SSL was not able to improve over CLUS-RF, then generally very few, or none of the unlabeled examples were added to the training set. This is the case at Forestry LIDAR IRS, Soil Quality and Vegetation Clustering (for more than 5% of labeled data) datasets. The predictive models learnt from these datasets are most probably prone to errors and the

self-training approach would only lead to a propagation of the errors (this is confirmed by the optimal threshold for reliability close to 1).

A different perspective of the results is provided in Figure 2, where per-target RMSE improvements over the baseline are shown. As it is possible to see, these results show that the improvement provided by the semi-supervised setting is not uniform over the different targets. This means that for some target attributes, there is still a large margin for improvement with respect to accuracy reached by the random forest approach.

5 Conclusions and further work

Semi-supervised learning is an intriguing research area because of potential gains in performance for 'free' – labeling of the data is expensive and laborious, while freely available unlabeled data can be used to enhance the performance of traditional, supervised, machine learning methods. Such proposition is even more relevant for learning problems with structured outputs, where labeling of the data is even more expensive and problematic.

We address the task of semi-supervised learning for multi-target regression – a type of structured output, where the goal is to simultaneously predict multiple continuous variables. To the best of our knowledge, semi-supervised methods dealing with this task do not exist thus far. We propose a self-training approach to semi-supervised learning by using a random forest of predictive clustering trees for multi-target regression. In the proposed approach, a model uses its own most reliable predictions in an iterative fashion.

Due to its relative simplicity and intuitiveness, self-training can be considered as a baseline semi-supervised approach, i.e., a starting point for investigation of the influence of unlabeled data. In this study, we wanted to investigate whether unlabeled data can improve predictive performance of the models for MTR in a self-training setting. The results of the experimental evaluation show that the proposed method outperforms its supervised counterpart on 4 out of 6 datasets. These are encouraging results and prompt further investigation.

In future, we plan to extend this work along several directions. First, we plan to implement an intelligent threshold selection procedure. Namely, here we consider a relatively simple implementation of self-training (with respect to the thresholding system and the stopping criteria), but there are several possibilities to implement more sophisticated variants of it. For instance, so-called 'airbag' stopping criteria [33] can detect degradation in performance and stop self-training. Alternatively, we can utilize 'out-of-bag properties' of the random forest to automatically detect the optimal threshold for the reliability score. Second, success of the reliability estimate of regression predictions can vary depending on the domain or the regression model used. The most appropriate estimates can be automatically detected [34] and used during self-training. Third, modularity of predictive clustering trees enables easy extension of the self-training approach to the other types of structured outputs, such as multi-target classification or time-series prediction.

Acknowledgments

We would like to acknowledge the support of the European Commission through the project MAESTRA - Learning from Massive, Incompletely annotated, and Structured Data (Grant number ICT-2013-612944).

References

1. Chapelle, O., Schölkopf, B., Zien, A.: Semi-supervised Learning. Volume 2. MIT Press, Cambridge, MA (2006)
2. Demšar, D., Džeroski, S., Larsen, T., Struyf, J., Axelsen, J., Pedersen, M., Krogh, P.: Using multi-objective classification to model communities of soil. Ecological Modelling **191**(1) (2006) 131–143
3. Stojanova, D., Panov, P., Gjorgjioski, V., Kobler, A., Džeroski, S.: Estimating vegetation height and canopy cover from remotely sensed data with machine learning. Ecological Informatics **5**(4) (2010) 256–266
4. Levatić, J., Kocev, D., Džeroski, S.: The use of the label hierarchy in hierarchical multi-label classification improves performance. In Appice, A., et al., eds.: New Frontiers in Mining Complex Patterns. Volume 8399 of Lecture Notes in Computer Science. Springer International Publishing, Switzerland (2014) 1–16
5. Appice, A., Džeroski, S.: Stepwise induction of multi-target model trees. In Kok, J.N., Koronacki, J., Mantaras, R.L.d., Matwin, S., Mladenić, D., Skowron, A., eds.: Machine Learning: ECML 2007. Volume 4701 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2007) 502–509
6. Struyf, J., Džeroski, S.: Constraint based induction of multi-objective regression trees. In: Proceedings of the 4th International Workshop on Knowledge Discovery in Inductive Databases, LNCS 3933, Springer, Berlin (2006) 222–233
7. Kocev, D., Džeroski, S., White, M.D., Newell, G.R., Griffioen, P.: Using single- and multi-target regression trees and ensembles to model a compound index of vegetation condition. Ecological Modelling **220**(8) 1159–1168
8. Kocev, D., Vens, C., Struyf, J., Džeroski, S.: Tree ensembles for predicting structured outputs. Pattern Recognition **46**(3) (2013) 817–833
9. Brefeld, U.: Semi-supervised Structured Prediction Models. PhD thesis, Humboldt-Universität zu Berlin, Berlin, Germany (2008)
10. Zhang, Y., Yeung, D.Y.: Semi-supervised multi-task regression. In Buntine, W., Grobelnik, M., Mladeni, D., Shawe-Taylor, J., eds.: Machine Learning and Knowledge Discovery in Databases. Volume 5782 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2009) 617–631
11. Navaratnam, R., Fitzgibbon, A., Cipolla, R.: The joint manifold model for semi-supervised multi-valued regression. In: Proceedings of the 11th IEEE International Conference on Computer Vision. (2007) 1–8
12. Zhu, X.: Semi-supervised learning literature survey. Technical report, Computer Sciences, University of Wisconsin-Madison (2008)
13. Yarowsky, D.: Unsupervised word sense disambiguation rivaling supervised methods. In: Proceedings of the 33rd annual meeting on Association for Computational Linguistics. (1995) 189–196
14. Rosenberg, C., Hebert, M., Schneiderman, H.: Semi-supervised self-training of object detection models. In: Proceedings of the 7th IEEE Workshop on Applications of Computer Vision. (2005)
15. Riloff, E., Wiebe, J., Wilson, T.: Learning subjective nouns using extraction pattern bootstrapping. In: Proceedings of the 7th Conference on Natural Language Learning. (2003) 25–32

16. Bandouch, J., Jenkins, O.C., Beetz, M.: A self-training approach for visual tracking and recognition of complex human activity patterns. International Journal of Computer Vision **99**(2) (2012) 166–189
17. Brefeld, U., Grtner, T., Scheffer, T., Wrobel, S.: Efficient co-regularised least squares regression. In: Proceedings of the 23rd international conference on Machine learning. (2006) 137–144
18. Zhou, Z.H., Li, M.: Semi-supervised regression with co-training style algorithms. IEEE Transaction in Knowledge Data Engineering **19**(11) (2007) 1479–1493
19. Appice, A., Ceci, M., Malerba, D.: An iterative learning algorithm for within-network regression in the transductive setting. In Gama, J., Costa, V., Jorge, A., Brazdil, P., eds.: Discovery Science. Volume 5808 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2009) 36–50
20. Appice, A., Ceci, M., Malerba, D.: Transductive learning for spatial regression with co-training. In: Proceedings of the 2010 ACM Symposium on Applied Computing. (2010) 1065–1070
21. Yang, M.C., Wang, Y.C.F.: A self-learning approach to single image super-resolution. IEEE Transactions on Multimedia **15**(3) (2013) 498–508
22. Blockeel, H., Struyf, J.: Efficient algorithms for decision tree cross-validation. Journal of Machine Learning Research **3** (2002) 621–650
23. Breiman, L., Friedman, J., Olshen, R., Stone, C.J.: Classification and Regression Trees. Chapman & Hall/CRC (1984)
24. Breiman, L.: Random forests. Machine Learning **45**(1) (2001) 5–32
25. Breiman, L.: Bagging predictors. Machine Learning **24**(2) (1996) 123–140
26. Bosnić, Z., Kononenko, I.: Comparison of approaches for estimating reliability of individual regression predictions. Data & Knowledge Engineering **67**(3) (2008) 504–516
27. Stojanova, D.: Estimating forest properties from remotely sensed data by using machine learning. Master's thesis, Jožef Stefan International Postgraduate School, Ljubljana, Slovenia (2009)
28. Demšar, D., Debeljak, M., Lavigne, C., Džeroski, S.: Modelling pollen dispersal of genetically modified oilseed rape within the field. In: The Annual Meeting of the Ecological Society of America. (2005)
29. Asuncion, A., Newman, D.: UCI machine learning repository (2007)
30. Gjorgjioski, V., Džeroski, S.: Clustering analysis of vegetation data. Technical report, Jožef Stefan Institute (2003)
31. Blockeel, H., Džeroski, S., Grbović, J.: Simultaneous prediction of multiple chemical parameters of river water quality with tilde. In: Proceedings of the 3rd European Conference on PKDD. Volume 1704 of LNAI. (1999) 32–40
32. Chawla, N., Karakoulas, G.: Learning from labeled and unlabeled data: An empirical study across techniques and domains. Journal of Artificial Intelligence Research **23**(1) (2005) 331–366
33. Leistner, C., Saffari, A., Santner, J., Bischof, H.: Semi-supervised random forests. In: Proceedings of the 12th International Conference on Computer Vision. (2009) 506–513
34. Bosnić, Z., Kononenko, I.: Automatic selection of reliability estimates for individual regression predictions. The Knowledge Engineering Review **25**(1) (2010) 27–47