Comparison of Tree-Based Methods for Multi-target Regression on Data Streams

Aljaž Osojnik[1,2(✉)], Panče Panov[1], and Sašo Džeroski[1,2,3]

[1] Jožef Stefan Institute, Jamova Cesta 39, Ljubljana, Slovenia
{aljaz.osojnik,pance.panov,saso.dzeroski}@ijs.si
[2] Jožef Stefan International Postgraduate School, Jamova Cesta 39, Ljubljana, Slovenia
[3] Centre of Excellence for Integrated Approaches in Chemistry and Biology of Proteins, Jamova Cesta 39, Ljubljana, Slovenia

Abstract. Single-target regression is a classical data mining task that is popular both in the batch and in the streaming setting. Multi-target regression is an extension of the single-target regression task, in which multiple continuous targets have to be predicted together. Recent studies in the batch setting have shown that global approaches, predicting all of the targets at once, tend to outperform local approaches, predicting each target separately. In this paper, we explore how different local and global tree-based approaches for multi-target regression compare in the streaming setting. Specifically, we apply a local method based on the FIMT-DD algorithm and propose a novel global method, named iSOUP-Tree-MTR. Furthermore, we present an experimental evaluation that is mainly oriented towards exploring the differences between the local and the global approach.

1 Introduction

A common approach to complex data mining tasks is to transform them into simpler tasks, for which have known methods that produce appropriate solutions. This problem transformation approach has been used to address predictive modelling tasks, such as the multi-label classification and multi-target regression tasks. A multi-label classification task can thus be transformed into a collection of binary classification tasks, while a multi-target regression task can be decomposed into several single-target regression problems.

There are, however, methods that forego the reduction to simpler tasks and tackle the complexity head-on. Specifically, in the case of multi-target regression, methods that consider and predict all of the continuous targets at once have received considerable coverage in the literature [13,20]. Almost exclusively, though, these methods have been introduced in the batch setting.

Recently, the streaming setting is becoming more and more prominent, in large part, due to the ever increasing presence of the Big Data paradigm. The streaming setting emphasizes several of the characteristics of Big Data, i.e., the

M. Ceci et al. (Eds.): NFMCP 2015, LNAI 9607, pp. 17–31, 2016.
DOI: 10.1007/978-3-319-39315-5_2

"V"s of Big Data. Specifically, streaming methods need to tackle Velocity – data arriving with high speed; Volume – potentially unbounded number of data instances; and Variability – potential changes in the data itself.

Methods that address the task of multi-target regression in a streaming setting are few and far between, especially those that predict all of the targets at the same time. In this paper, we present a new tree-based method, named iSOUP-Tree-MTR, capable of addressing the task of multi-target regression in this manner. We compare it to the simpler problem transformation approach of using single-target tree-based methods in a streaming setting and show that the iSOUP-Tree-MTR method has superior performance. Finally, we explore the performance of ensembles, e.g., online bagging [15], when using the iSOUP-Tree-MTR method as a base learner.

The structure of the paper is as follows. First, we present the background and related work (Sect. 2). Next, we present several tree-based approaches for multi-target regression on data streams (Sect. 3). Furthermore, we present the research questions we address and the experimental design employed to answer them (Sect. 4). Finally, we conclude with a discussion of the results (Sect. 5), followed by conclusions, and directions for further work (Sect. 6).

2 Background and Related Work

In this section, we define the multi-target regression task and present the constraints imposed by the streaming context. Additionally, we briefly review the state-of-the art in multi-target regression, both in the batch and in the streaming setting.

2.1 Multi-target Regression

In essence, we can look at the task of multi-target regression as an extension of the single-target regression task. In the later, only one continuous variable needs to be predicted. The task of multi-target regression (MTR) deals with predicting multiple numeric variables simultaneously, or, formally, with making a prediction $\hat{y}$ from $\mathbb{R}^n$, where n is the number of targets for a given instance $\boldsymbol{x}$ from an input space X. To categorize the different approaches to MTR we use the nomenclature introduced by Silla and Freitas [12] for the task of hierarchical multi-label classification. The task of simultaneous prediction of all targets at the same time (the *global* approach) has been considered in the batch setting by Struyf and Džeroski [20]. In addition, Appice and Džeroski [1] proposed a method for stepwise induction of multi-target model trees.

2.2 Data Streams

Unlike the batch context, where a fixed and complete dataset is given as input to a learning method, the streaming context presents several constraints that a stream learning method must consider. The most relevant are [2]:

1. the examples arrive sequentially in a specific order;
2. the number of examples can be arbitrarily large;

3. the distribution of examples need not be stationary; and
4. after an example is processed, it is discarded or archived.

The fact that the distribution of examples is not assumed to be stationary means that methods learning in a streaming context should be able to detect and adapt to changes in the distribution (*concept drift*).

2.3 Multi-target Regression on Data Streams

In the streaming context, some recent work has already addressed the task of single- and multi-target regression. Ikonomovska et al. [11] introduced an instance-incremental streaming tree-based single-target regressor (FIMT-DD) that utilizes the Hoeffding bound. This work was later extended for the task of multi-target regression (FIMT-MT) [10]. However, both of these methods had the drawback of ignoring nominal input attributes. Recently, there has been some theoretical debate whether the use of the Hoeffding bound is appropriate [16], however, a recent study by Ikonomovska et al. [9] has shown that in practice the use of the Hoeffding bound produces good results. Other related work includes an instance-based system for classification and regression (IBLStreams), introduced by Shaker et al. [17], which can be, in principle, used for multi-target regression.

3 Tree-Based Approaches for Multi-target Regression on Data Streams

Generally, the quickest way of solving a complex task, such as multi-target regression, is to transform it into a set of simpler tasks that we know how to solve. In the case of multi-target regression, specifically, this is achieved by training a regressor for each of the targets separately, essentially resulting in a collection/ensemble of regressors. The other option for addressing the task of multi-target regression is to produce a regressor which gives predictions for all of the targets simultaneously.

To distinguish between these two approaches we refer to them as local and global, respectively [12]. Specifically, a method that uses one regressor per target is using the *local approach*, while a method that uses one regressor to predict all of the targets simultaneously is using the *global approach*. Recent studies show, that in the batch case, the global approaches outperform the local ones [13]. Global methods tend to (implicitly) exploit the dependencies between the targets.

In this section, we present several tree-based methods for multi-target regression, which utilize the local approach, as well as the global approach. Tree-based methods are often used, as they generally provide good results in terms of predictive performance, while also yielding interpretable models. Finally, we present a baseline method that can be viewed as both local and global and is highly relevant to the methods introduced below.

3.1 A Local Approach to MTR

One of the best known single-target tree-based regressors in the stream setting is the FIMT-DD method [11]. It is based on the Hoeffding bound, which allows the generalization of observations from small samples to the underlying distribution. Similarly to Hoeffding trees for classification [4], FIMT-DD uses the Hoeffding bound to determine the best splits at each node of the regression tree.

We have re-implemented the FIMT-DD method in the Java-based MOA stream-mining framework [3] and extended it to use adaptive models in the leaves, similarly as Duarte et al. [5]. Specifically, each leaf of the tree contains a perceptron. The perceptron is a linear function of the values of the input attributes $\boldsymbol{x}$ that produces the prediction, i.e., $\hat{y} = \boldsymbol{w} \cdot \boldsymbol{x} + b$, where $\boldsymbol{w}$ and b are a learned weight vector and a constant, respectively.

In the original implementation of Ikonomovska et al. [11], the perceptron was always used to make the prediction. However, the adaptive model records the errors of the perceptrons and compares them to the errors of the mean target predictors, which predict the value of the target by computing the average value of the target over the examples observed in a given leaf. In essence, each leaf has two learners, the perceptron and the target mean predictor. The prediction of the learner that (at a given point in time) has a lower error is then used as a final prediction.

To monitor the errors, we use the faded absolute error which is calculated as

$$\mathit{fMAE}_{\mathit{learner}}(m) = \frac{\sum_{j=1}^{m} 0.95^{m-j} |\hat{y}(j) - y(j)|}{\sum_{j=1}^{m} 0.95^{m-j}},$$

where m is the number of observed examples in a leaf, $\hat{y}(j)$ and $y(j)$ are the predicted and real value for the j-th example, respectively, and $\mathit{learner} \in \{\mathit{perceptron}, \mathit{targetMean}\}$. In essence, the faded error is weighted towards more recent examples. Intuitively, the numerator of the fraction is the faded sum of absolute errors, while the denominator is the faded count of examples. For example, the most recent (m-th) example contributes with a weight of 1, the previous example with weight 0.95, and the first example with weight 0.95^{m-1}. This places an emphasis on the more recent examples and generally benefits the perceptron, as we expect its errors to decrease as it learns the weight vector.

We have also implemented a meta-learning method in MOA that creates a homogeneous ensemble of single-target regressors, one regressor for each target, and combines their single-target predictions into a multi-target prediction in real-time to facilitate the use of FIMT-DD as a multi-target regressor. This combination of methods is referred to as the **Local FIMT-DD** method.

3.2 A Global Approach to MTR

As noted earlier, the global approach has been shown to yield better predictive performance in the case of tree-based methods in the batch setting. This has motivated the introduction of global tree-based methods for data streams, i.e., the FIMT-MT method introduced by Ikonomovska et al. [10]. FIMT-MT extends FIMT-DD by replacing the use of the variance reduction heuristic with the intra-cluster variance reduction heuristic, which captures some of the dependencies of the targets. However, one of the major downsides of the FIMT-MT method, is the fact that it completely ignores nominal input attributes.

We have extended the FIMT-MT method by adding the support for nominal input attributes. In addition, we have also proposed the use of this extension to address other structured output prediction tasks, e.g., multi-label classification [14]. The new method is named **i**ncremental **S**tructured **Ou**tput **P**rediction **Tree** for **M**ulti-**t**arget **R**egression (**iSOUP-Tree-MTR**). This method, as the one before, is implemented in the MOA environment.

In each leaf, the iSOUP-Tree method uses an adaptive multi-target model, which consists of a multi-target perceptron and a multi-target target mean predictor. As in the single-target case, the multi-target perceptron produces the prediction vector as $\hat{\boldsymbol{y}} = W\boldsymbol{x} + \boldsymbol{b}$, where W is the weight matrix and $\boldsymbol{b}$ is the additive vector of constants. On the other hand, the multi-target target mean predictor computes the prediction as the mean value of each of the targets observed at a given leaf. Individually, these learners can be seen as local, however, in a conjunction with a tree building method, they constitute a global method.

For each target y_i, the errors $fMAE^i_{perceptron}$ and $fMAE^i_{targetMean}$ are recorded and the decision which of the predictions to use is made for each variable separately. Formally, for each $i \in \{1, \ldots, n\}$ the prediction $\hat{y}^i_{perceptron}$ is used when $fMAE^i_{perceptron} < fMAE^i_{targetMean}$, otherwise we use $\hat{y}^i_{targetMean}$. Consequently, a final prediction $\hat{\boldsymbol{y}} = (\hat{y}^1, \ldots, \hat{y}^n)$ can contain predictions made by the perceptron (for some targets), the target mean predictor (for other targets), or both.

3.3 Ensemble of Trees for MTR on Data Streams

In this paper, we also consider constructing ensembles of trees for multi-target regression in the stream setting. For this purpose, we use iSOUP-Trees, discussed in the previous subsection, as base learners. To construct the ensemble, we use the bagging method for introducing diversity among the ensemble members. The bagging method for data streams was introduced by Oza et al. [15] and incorporates a probabilistic variation of how many times each given example is "seen" by each of the base learners. In this paper, we refer to this ensemble method as **iSOUP-Tree-MTR bagging**.

3.4 Baseline Method

For the comparison of tree-based methods for multi-target regression on data streams, we use an adaptive multi-target model as a **baseline** regressor. The baseline regressor conveniently corresponds to both an ensemble of leaves using the local approach, as well as to a single leaf in the global iSOUP-Tree approach. In essence, the adaptive model corresponds to a tree-based model that is not allowed to grow, i.e., with leaves that are never split. The baseline method is specifically implemented as a stripped down version of iSOUP-Tree-MTR, building a tree that consists of a single leaf node.

4 Experimental Setup

In this section, we first present the experimental questions that we want to answer in this paper. Next, we discuss the evaluation measures used in the experiments and present the experimental methodology. Finally, we describe the datasets and conclude with the methods used in the experiments.

4.1 Experimental Questions

The first experimental question, that we wish to address in this paper, is the experimental comparison of local and global approaches. As the streaming context imposes several constraints on the learning process, it is not immediately clear whether the findings from the batch setting will be replicated in the streaming setting. While we are specifically using tree-based methods for multi-target regression on data streams, showing that the global approach increases predictive performance could also suggest that this may be generally true, i.e., applicable to other types of methods and structured outputs in the streaming setting.

When the product of the number of input attributes and target variables, which generally corresponds to the time and memory complexity, is low, we expect the local approach to be competitive with the global approach in terms of time and memory. However, as this product increases, we expect that the training of several distinct, even if simpler, models takes more time and especially more memory than the training of a single, more complex, model.

In a single-target study, Ikonomovska et al. [9] have shown no particular differences in the predictive performance of the basic method and the bagging method (therein referred to as FIMT-DD and OBag, respectively). In this work, we wish to investigate whether similar conclusions can be drawn in the multi-target case. To that end, we explore the differences in predictive performance between the iSOUP-Tree-MTR and the iSOUP-Tree-MTR bagging methods. The resource consumption of the bagging method is trivially extrapolated from the resource consumption of a single tree and the number of trees used. Consequently, for this experimental question we focus on the predictive performance.

4.2 Evaluation Measures and Experimental Methodology

To evaluate the predictive performance we define the *average relative mean absolute error* ($\overline{RMAE}$) on an evaluation dataset D as

$$\overline{RMAE} = \frac{1}{n}\sum_{i=1}^{n}\frac{\sum_{j=1}^{|D|}\left|y_j^i - \hat{y}_j^i\right|}{\sum_{j=1}^{|D|}\left|y_j^i - \overline{y}_j^i\right|},$$

where y_j are the true values of the targets, $\hat{y}_j$ are the predictions of the evaluated model and $\overline{y}_j$ are the predictions of a baseline model (all on D). Specifically,

Table 1. Datasets used in the experiments and their properties. N – number of instances, T – number of targets.

Dataset	Domain	N	Input attr.	T
Bicycles [6]	service prediction	17379	12 numeric	3
EUNITE03	quality prediction	8064	29 numeric	5
Forestry Kras [18]	vegetation prediction	60607	160 numeric	11
Forestry Slivnica [19]	vegetation prediction	6218	149 numeric	2
RF1 [21]	environmental prediction	9005	64 numeric	8
RF2 [21]	environmental prediction	7679	575 numeric	8
SCM1d [21]	price prediction	9803	280 numeric	16
SCM20d [21]	price prediction	8966	61 numeric	16

we use the predictions of the baseline perceptron as described in Sect. 3.4 as the baseline (but note that other candidates for the baseline can be used). By definition, this means that the $\overline{RMAE}$ of the baseline is always equal to 1. Essentially, $\overline{RMAE}$ is the relative error averaged over all of the target variables.

To evaluate the time consumption, we will consider the running time of the methods. The memory consumption is measured using the size (in bytes) of the learned models. Both time and memory consumption, as well as $\overline{RMAE}$, are reported at intervals of 1000 examples.

We are using the *prequential* [7] approach for evaluating data stream mining methods. An incoming instance is first used to make a prediction, which is used in the evaluation. Afterwards, the model is updated using the instance. Since the reported errors on data streams can be volatile if reported on an instance by instance basis, due to, e.g., the sampling of different parts of the input space, we calculate the evaluation measures on batches of 1000 examples. Specifically, we calculate the $\overline{RMAE}$ on the first 1000 examples, report it, and then repeat this for examples 1000 through 2000, etc.

4.3 Datasets

We have selected a total of 8 datasets for our experiments based on their size, looking for diversity in the number of input and target attributes. We consider the datasets under the assumption of no concept drift, given that these datasets are generally considered as batch datasets. A summary of the datasets and their properties is shown in Table 1.

The *Bicycles* dataset [6] is concerned with the prediction of demand for rental bicycles on an hour-by-hour basis. The 3 targets represent the number of casual (non-registered) users, the number of registered users and the total number of users for a given hour, respectively.

The *EUNITE03*[1] dataset was used for the competition at the 3rd European Symposium on Intelligent Technologies, Hybrid Systems and their implementation

[1] http://www.eunite.org/eunite/news/Summary%20Competition.pdf.

on Smart Adaptive Systems in 2003. The data describes a complex process of continuously manufactured glass products, i.e., the input attributes describe various influences which can or can not be changed by an operator, while the target attributes describe the glass quality.

The data used in the *Forestry Kras* dataset [18] was constructed from multi-spectral multi-temporal Landsat satellite images and 3D LiDAR recordings of a part of the Kras region in Slovenia. The input attributes and target variables were extracted from the readings, specifically for spatial units of 25 m by 25 m. For specifics on the data preparation procedure, see Stojanova et al. [18]. The task is to predict 11 target variables which correspond to properties of the vegetation in the observed spatial unit.

The data in the *Forestry Slivnica* [19] dataset was, as in the previous case, constructed from multi-spectral multi-temporal Landsat satellite images and 3D LiDAR recordings of a part of the Slivnica region in Slovenia. In this dataset, the task is to predict only 2 target variables: vegetation height and canopy cover.

The river flow datasets, *RF1* and *RF2* [21], concern the prediction of river network flows for 48 h at 8 locations on the Mississippi River network. Each data example comprises the latest observations for each of the 8 locations as well as time-lagged observations from 6, 12, 18, 24, 36, 48 and 60 h in the past. In RF1, each location contributes 8 input attributes, for a total of 64 input attributes and 8 target variables. The RF2 dataset extends RF1 with the precipitation forecast information for each of the 8 locations and 19 other meteorological sites. Specifically, the precipitation forecast for 6 h windows up to 48 h in the future is added, which nets a total of 280 input attributes.

The *SCM1d* and *SCM20d* are datasets derived form the Trading Agent Competition in Supply Chain Management (TAC SCM) conducted in July 2010. The preparation (preprocessing) of the datasets is described by Xioufis et al. [21]. The data instances correspond to daily updates in a tournament – there are 220 days in each game and 18 games per tournament. The 16 targets are the predictions of the next day and the 20 day mean price for each of the 16 products in the simulation, for the SCM1d and SCM20d datasets, respectively.

The Bicycles dataset is available at the UCI Machine Learning Repository[2] and the RF1, RF2, SCM1d and SCM20d datasets are available at the Mulan multi-target regression dataset repository[3]. The examples with missing values in the RF1 and RF2 datasets were removed, so the resulting datasets were somewhat smaller than reported in the repository.

4.4 Compared Methods

For our experiments, we consider the local and global tree-based methods described in Sect. 3. Specifically, we consider the multi-target perceptron as a baseline, the local FIMT-DD-based approach to MTR, the global iSOUP-Tree-MTR approach, and iSOUP-Tree-MTR bagging.

[2] https://archive.ics.uci.edu/ml/datasets/Bike+Sharing+Dataset.
[3] http://mulan.sourceforge.net/datasets-mtr.html.

The FIMT-DD method is capable of detecting changes in the concept and adapting to them. However, as this study is oriented towards comparing the local and global tree-based approaches on equal grounds, the change detection and adaptation mechanisms in FIMT-DD have been disabled for this study.

When comparing the bagging approach we also study the effect of increasing the number of trees in the ensemble. To do this we train an ensemble of 100 trees, then use an increasing number of trees for prediction (20, 40, 60, 80 and 100 trees). This allows us to directly see the improvement provided by additional trees, as the results for the higher number of trees are in essence an extension of those with a lower number of trees.

5 Results

In this section we present and discuss the results of our experiments and provide insights into several of the observed phenomena. We consider the performance of the compared methods on the 8 datasets in terms of the evaluation measures: predictive performance and time/memory consumption.

5.1 Predictive Performance ($\overline{RMAE}$)

The predictive performance of the iSOUP-Tree-MTR method appears to be generally worse than that of the Local FIMT-DD method (Fig. 1), excluding the results on the EUNITE03 and SCM20d datasets (Fig. 1b and h). However, we observe that for some datasets (RF1, RF2, SCM1d and Forestry Kras) the iSOUP-Tree-MTR method initially has an identical performance as the baseline. This occurs due to the splitting mechanism of the iSOUP-Tree-MTR method: the root node of the tree is not split until a large number of examples accumulates (in our specific results, 3400 examples on all of the affected datasets).

The affected datasets have a high number of input attributes and/or target variables. In the case of a large number of input attributes, many of them can and do have similar values of the heuristic function so the splitting mechanism cannot easily and quickly determine the best candidate among them. On the other hand, when the number of target variables is high, the aggregation part of the heuristic removes the specificity to particular output attributes, again resulting in similar evaluations of different input attributes. The number of input attributes at which this occurs is much higher (100+) than the number of target variables at which it occurs (already at 8 targets, Fig. 1e and f). As examples accumulate, a tie threshold is reached and a split is made with lower confidence (for details on the tie breaking mechanism, see Ikonomovska et al. [8]). On the above datasets, inappropriate splits are apparently often selected and the performance suffers.

This problem affects the local method as well, on a tree by tree basis for each tree predicting a single-target. However, it is generally easier to distinguish among candidate splits when considering only one target. A potential workaround to this problem is the use of option trees [9], which bypass the myopia of the greedy tree building approach. This problem also naturally affects the results of the ensemble method, in a slightly different way as described below.

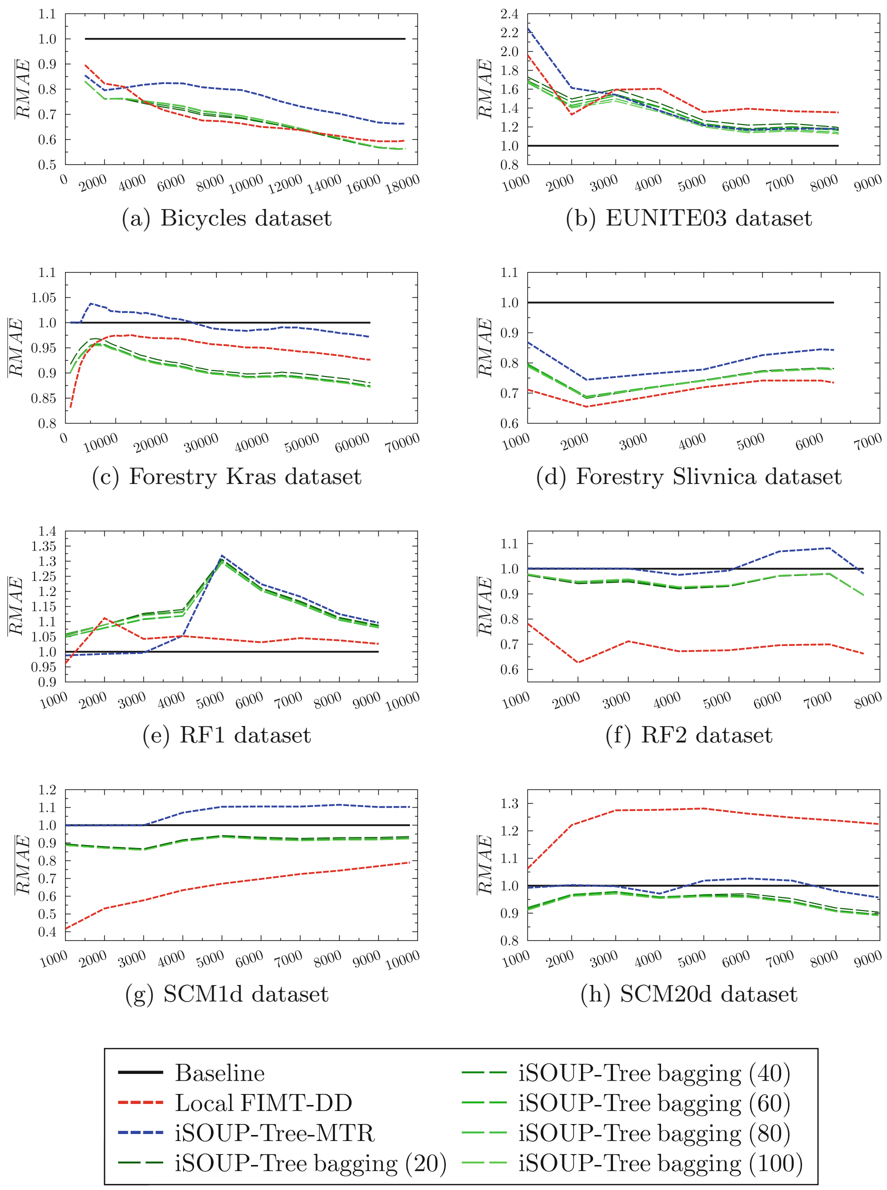
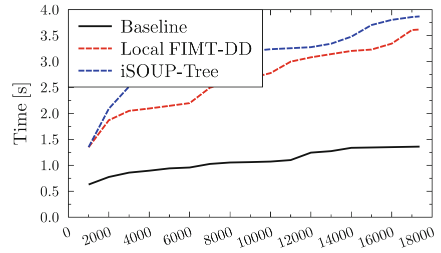
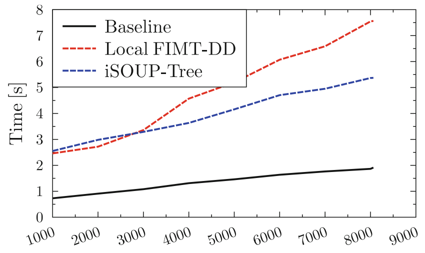


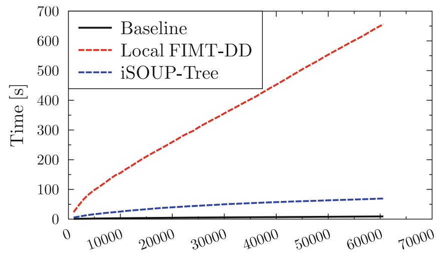
Fig. 1. The evolution of $\overline{RMAE}$ on the selected datasets. Horizontal axes show number of processed examples.



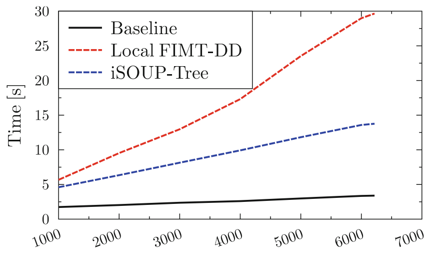
(a) Bicycles dataset



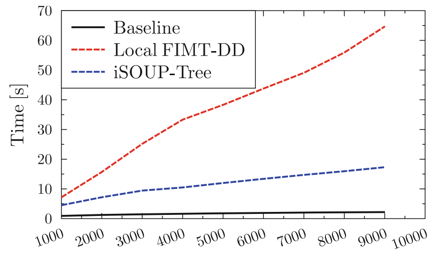
(b) EUNITE03 dataset



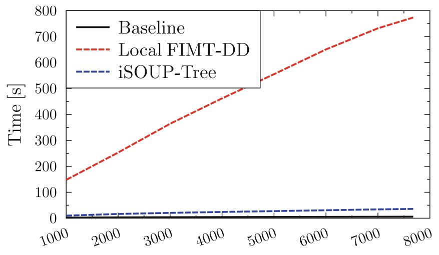
(c) Forestry Kras dataset



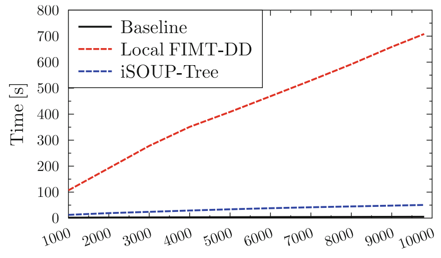
(d) Forestry Slivnica dataset



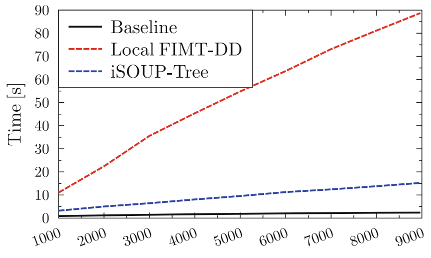
(e) RF1 dataset



(f) RF2 dataset



(g) SCM1d dataset



(h) SCM20d dataset

Fig. 2. The time consumption on the selected datasets. Horizontal axes show number of processed examples.

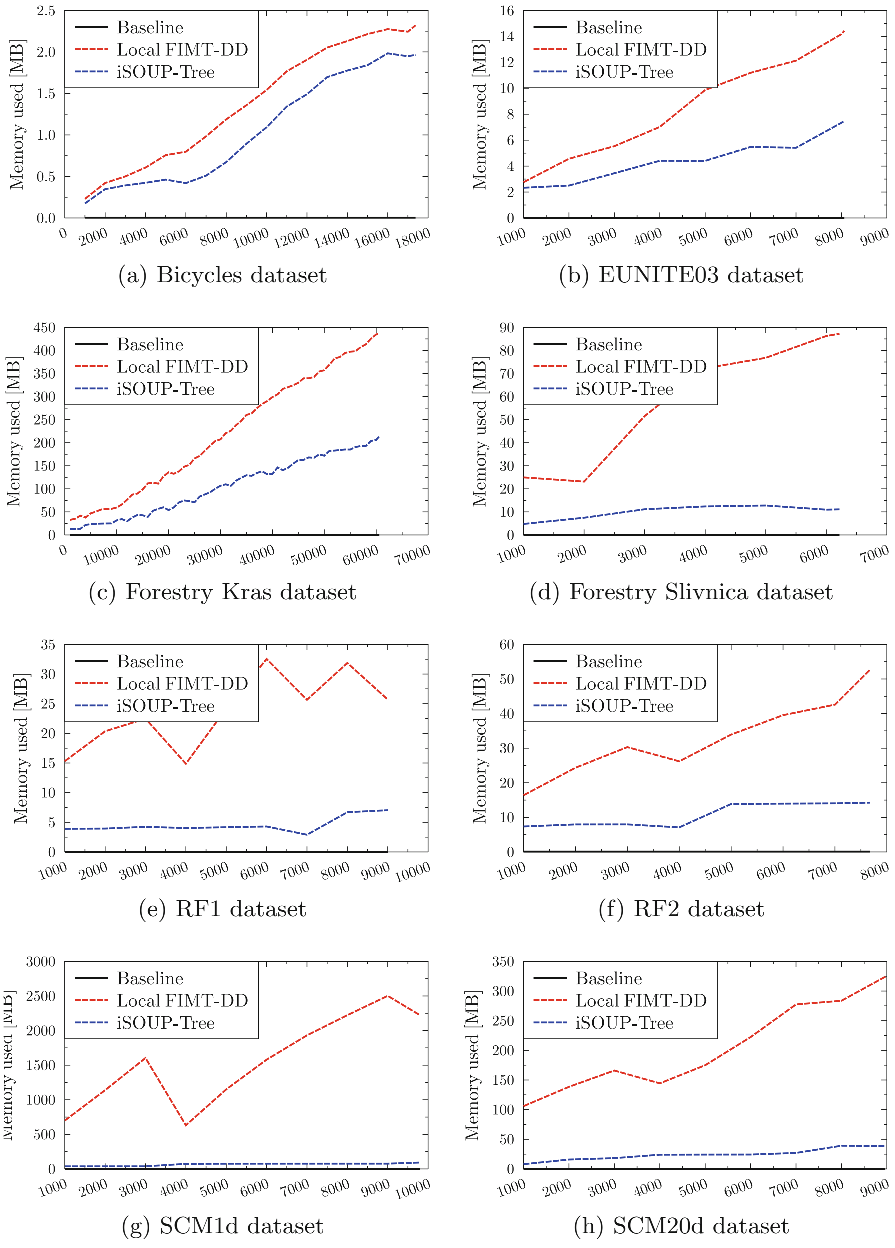


Fig. 3. The memory consumption on the selected datasets. Horizontal axes show number of processed examples.

On the datasets where this does not occur, iSOUP-Tree-MTR and Local FIMT-DD have comparable results. Specifically, Local FIMT-DD outperforms iSOUP-Tree-MTR on the Bicycles and Forestry Kras datasets, while the opposite is true on the EUNITE03 and SCM20d datasets.

When comparing the performance of bagging versus a single tree, bagging generally outperforms the single tree and in some cases even the local method. Interestingly, the variation introduced by the bagging mechanism is sufficient to produce splits (for some of the trees in the ensemble), even for the datasets where iSOUP-Tree-MTR gets "stuck". Therefore, the increase in predictive performance is highest on those datasets. In addition, increasing number of trees generally does not increase the predictive performance much, i.e., the predictive performance of 20 tree ensembles is similar to that of 100 tree ensembles.

5.2 Time Consumption

The results about time consumption (see Fig. 2) are clear. When the number of input attributes and target variables is low, i.e., on the Bicycles and the EUNITE03 dataset, the local FIMT-DD uses less or a similar amount of time as compared to the global iSOUP-Tree-MTR. In these cases, the number of local trees that are traversed and processed is relatively low. In all other cases, iSOUP-Tree-MTR convincingly outpaces the local method, clearly demonstrating that it is a much more scalable approach in terms of time consumption.

5.3 Memory Consumption

The graphs in Fig. 3 indicate that memory consumption of the observed methods is less stable than those of the time consumption. However, the results are more straightforward. The decreases in memory usage occur when a memory intensive leaf node (which stores all the necessary statistics used to evaluate potential splits) is replaced with a split node and new, empty leaf nodes. The iSOUP-Tree-MTR method, which produces only one tree, uses much less memory. As with time, iSOUP-Tree-MTR is more scalable in terms of memory.

6 Conclusions and Further Work

We have conducted an experimental study comparing different approaches for multi-target regression on data streams. The main comparison of this paper considers the local approach (using the FIMT-DD single-target regressor) versus the global approach (using the iSOUP-Tree-MTR multi-target regressor. Additionally, we have compared the performance of iSOUP-Tree-MTR and an ensemble of such trees learned using the online bagging learning method.

Unlike the previous work by Ikonomovska et al. [10], our experiments have highlighted a vulnerability of the greedy, myopic tree building mechanism. Unable to distinguish the best candidate split, the learning method is placed in a staying pattern, where it waits for more information. It finally selects the currently best evaluated split, even though there is not enough support to do so.

More specifically, this problem occurs when the number of input attributes and/or target variables is high. In the first case, when the number of input attributes is high, it is difficult to distinguish them in terms of the employed heuristic. In the second case, when the number of target variables is high, the heuristic values across target variables, results in similar values of the heuristic for many input attributes. The number of input attributes when this occurs is much higher (100+) than the number of target variables (where this phenomena occurs already at 8 targets). This results in bad predictive performance, which occurs on 4 out of 8 of the observed datasets.

On the remaining 4 datasets, we observe that the compared local FIMT-DD and global iSOUP-Tree-MTR have similar results in terms of predictive performance. However, the global method is much more scalable in terms of time and memory consumption. As far as resource consumption is concerned, the local method only outperforms the global one on one dataset – the one with the smallest product of the number of input attributes and target variables.

Furthermore, we have observed that the bagging method generally produces better results in terms of predictive performance. Due to the randomness of the sampling process, it often avoids being trapped into the staying pattern of the single-tree approaches. Note that, the increases in performance due to increasing the number of trees in the ensemble (20 to 100) seem to be negligible.

While the increase in predictive performance brought by the bagging approach can be valuable, it comes at a great cost to both time and memory consumption. While we omitted the specific analysis of resource consumption for the ensemble method, the time and memory demands are linear in the number of trees. This leads to a trade-off between predictive performance and resource usage, if parallelization is not used.

Overall, the global approach does have merit in the streaming context and can produce similar results as the local approach, with a significantly lower resource use footprint. In our future work, we plan to address the problem of insufficient statistical proof when constructing the tree, possibly by implementing option trees [9] within the global iSOUP-Tree-MTR approach. These have been shown to grow faster with fewer available examples, by considering and using multiple attributes for splits, especially in learning the upper levels of the tree.

Acknowledgments. The authors are supported by The Slovenian Research Agency (Grant P2-0103 and a young researcher grant) and the European Commission (Grants ICT-2013-612944 MAESTRA and ICT-2013-604102 HBP).

References

1. Appice, A., Džeroski, S.: Stepwise induction of multi-target model trees. In: 18th European Conference on Machine Learning, pp. 502–509 (2007)
2. Bifet, A., Gavaldà, R.: Adaptive learning from evolving data streams. In: 8th International Symposium on Advances in Intelligent Data Analysis, pp. 249–260 (2009)
3. Bifet, A., Holmes, G., Kirkby, R., Pfahringer, B.: MOA: Massive online analysis. J. Mach. Learn. Res. **11**, 1601–1604 (2010)

4. Domingos, P., Hulten, G.: Mining high-speed data streams. In: 6th ACM SIGKDD, pp. 71–80. ACM, New York (2000)
5. Duarte, J., Gama, J.: Ensembles of adaptive model rules from high-speed data streams. In: 3rd International Workshop on Big Data, Streams and Heterogeneous Source Mining, pp. 198–213 (2014)
6. Fanaee-T, H., Gama, J.: Event labeling combining ensemble detectors and background knowledge. Prog. Artif. Intell. **2**, 1–15 (2013)
7. Gama, J.: Knowledge Discovery from Data Streams. CRC Press, Boca Raton (2010)
8. Ikonomovska, E., Gama, J.: Learning model trees from data streams. In: Boulicaut, J.-F., Berthold, M.R., Horváth, T. (eds.) DS 2008. LNCS (LNAI), vol. 5255, pp. 52–63. Springer, Heidelberg (2008)
9. Ikonomovska, E., Gama, J., Džeroski, S.: Online tree-based ensembles and option trees for regression on evolving data streams. Neurocomputing **150**, 458–470 (2015)
10. Ikonomovska, E., Gama, J., Džeroski, S.: Incremental multi-target model trees for data streams. In: 2011 ACM Symposium on Applied Computing, pp. 988–993. ACM, New York (2011)
11. Ikonomovska, E., Gama, J., Džeroski, S.: Learning model trees from evolving data streams. Data Min. Knowl. Disc. **23**(1), 128–168 (2011)
12. Silla Jr., C.N., Freitas, A.A.: A survey of hierarchical classification across different application domains. Data Min. Knowl. Disc. **22**(1–2), 31–72 (2011)
13. Kocev, D., Vens, C., Struyf, J., Džeroski, S.: Tree ensembles for predicting structured outputs. Pattern Recognit. **46**(3), 817–833 (2013)
14. Osojnik, A., Panov, P., Džeroski, S.: Multi-label classification via multi-target regression on data streams. In: Japkowicz, N., Matwin, S. (eds.) DS 9356. LNCS, vol. 9356, pp. 170–185. Springer, Heidelberg (2015). doi:10.1007/978-3-319-24282-8_15
15. Oza, N.C., Russel, S.J.: Experimental comparisons of online and batch versions of bagging and boosting. In: 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 359–364. ACM, New York (2001)
16. Rutkowski, L., Pietruczuk, L., Duda, P., Jaworski, M.: Decision trees for mining data streams based on the McDiarmid's bound. IEEE Trans. Knowl. Data Eng. **25**(6), 1272–1279 (2013)
17. Shaker, A., Hüllermeier, E.: IBLStreams: a system for instance-based classification and regression on data streams. Evol. Syst. **3**(4), 235–249 (2012)
18. 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)
19. Stojanova, D.: Estimating Forest Properties from Remotely Sensed Data by using Machine Learning. Master's thesis, Jožef Stefan International Postgraduate School, Ljubljana, Slovenia (2009)
20. Struyf, J., Dzeroski, S.: Constraint based induction of multi-objective regression trees. In: 4th International Workshop on Knowledge Discovery in Inductive Databases, pp. 222–233 (2005)
21. Xioufis, E.S., Groves, W., Tsoumakas, G., Vlahavas, I.P.: Multi-label classification methods for multi-target regression. CoRR abs/1211.6581 (2012). http://arxiv.org/abs/1211.6581

New Frontiers in Mining Complex Patterns
4th International Workshop, NFMCP 2015, Held in Conjunction with ECML-PKDD 2015, Porto, Portugal, September 7, 2015, Revised Selected Papers
Ceci, M.; Loglisci, C.; Manco, G.; Masciari, E.; Ras, Z.W. (Eds.)
2016, X, 239 p. 57 illus., Softcover
ISBN: 978-3-319-39314-8