Online semi-supervised learning for multi-target regression in data streams using AMRules

Ricardo Sousa[1] and João Gama[1,2]

[1] LIAAD/INESC TEC, Universidade do Porto, Portugal
rtsousa@inesctec.pt
[2] Faculdade de Economia, Universidade do Porto, Portugal
jgama@fep.up.pt

Abstract. Most data streams systems that use online Multi-target regression yield vast amounts of data which is not targeted. Targeting this data is usually impossible, time consuming and expensive. Semi-supervised algorithms have been proposed to use this untargeted data (input information only) for model improvement. However, most algorithms are adapted to work on batch mode for classification and require huge computational and memory resources.
Therefore, this paper proposes an semi-supervised algorithm for online processing systems based on AMRules algorithm that handle both targeted and untargeted data and improves the regression model. The proposed method was evaluated through a comparison between a scenario where the untargeted examples are not used on the training and a scenario where some untargeted examples are used. Evaluation results indicate that the use of the untargeted examples improved the target predictions by improving the model.

Keywords: $Multi-Target\ Regression \cdot Semi-Supervised\ Learning \cdot AMRules \cdot Data\ Streams$

1 Introduction

Multi-target regression(MTR), also known as Multi-output, Multi-variate or Multi-value, consists of predicting targets of numerical and nominal variables (outputs variables) from a set of other variables (input variables) using a trained relational and functional model [1].

Online data streams systems that use Multi-target regression produce massive amounts of data. Targeting all examples may be impossible, time consuming or expensive. Untargeted examples are abundant due to sensor malfunction, reluctance for sharing sensitive information, high cost of data collection or databases failure [2,3].

As related problems, most of the available methods work on batch mode with high computational and memory requirements [4,5]. Moreover, the literature presents more solutions for classification that cannot be directly applied to regression [6].

This regression methods are applied to forecasting and modelling in a wide range of areas such as Engineering Systems (electrical power consumption) [7], Physics (weather forecasting and ecological models) [8], Biology(model of cellular processes) [9] and Economy/Finance(stock price forecasting) [10]. In most of these areas, data from streams is obtained and processed in real time and most data is untargeted [3].

Several authors proposed the use of untargeted data (input variables information only) to improve regression models and targeted examples prediction that lead to the Semi-Supervised Learning(SSL) methods, also called Partially-Supervised Learning or Bootstrapping techniques [3].

Let $\mathcal{D} = \{..., (\mathbf{x}_1, \mathbf{y}_1),\ (\mathbf{x}_2, \mathbf{y}_2), ..., (\mathbf{x}_i, \mathbf{y}_i), ...\}$ denote an unlimited stream of data examples, where $\mathbf{x}_i = [x_{i,1} \cdots x_{i,j} \cdots x_{i,M}]$ is a vector of data descriptive variables $x_{i,j}$ and $\mathbf{y}_i = [y_{i,1} \cdots y_{i,j} \cdots y_{i,N}]$ is a vector of response output variables $y_{i,j}$ of the i^{th} example (considering one example with the index of reference). The untargeted example is represented with an empty target vector $\mathbf{y}_i = \emptyset$. The objective of SSL consists of using examples $(\mathbf{x}_i, \emptyset)$ to improve the regression model. SSL explores the information for the inference of the targets that is being conveyed by the input variables [6]. These algorithms are very useful if the untargeted data is far more abundant [6]. However, in some scenarios, the algorithm reinforces and propagates the error through the estimations [6].

The objective of this work is to develop an online multi-target algorithm that handle the untarget examples in order to improve the prediction for both target and untargeted examples.

Section 2 briefly reviews semi-supervised learning methods. Section 3 describes the proposed algorithm. Section 4 explains the evaluation method. Finally, the results are presented and discussed in Section 5 and the main conclusions are reported in Section 6.

2 Related work

This section briefly reviews some existent SSL algorithms. In the literature, most semi-supervised algorithms work on batch mode where the models are produced with the knowledge of all data [5]. Classification is the more addressed problem in SSL related literature [6]. For the best of our knowledge, no online semi-supervised multi-target regressor was found. Therefore, this review contains essentially descriptions of batch modes algorithms which are fair starting points for online algorithms development.

SSL techniques are essentially divided into five categories: self-training, co-training, generative models and graphs [6]. The self-training is an method that produces an prediction(based on the current model) and a confidence score for the untargeted example [11]. The prediction is used to target the example and the score is used to measure/predict the benefit of this artificially targeted example on the current model. A threshold is used to select the artificially targeted examples that benefit most [12, 11]. A self-training batch mode algorithm was proposed by Levatic et al [2]. This batch algorithm uses ensemble of Predictive

Clustering Trees(PCT) as underlying regressor. The predicted targets mean is used to target the example and a variability measure is used in the example acceptance for training.

The co-training method divides the input variables of the incoming example into two independent/uncorrelated groups and produce two examples with different input variables but with equal targets(targets of the incoming example). Two complementary regressors of the same type yield a prediction for each example which is used to targeting the example of the complementary regressor and for training [6]. COREG is a batch mode algorithm based on co-training that uses k-Nearest Neighbours(kNN) regressors [3]. This algorithm produces a small dataset of examples that are close(according to a predetermined distance metric) to the untargeted example. Each regressor predicts a target for the untargeted example and uses it to re-train the models with all targeted examples. Mean Squared Error (MSE) variation is computed between the scenarios with and without the artificially targeted examples. If MSE is reduced, the artificially targeted example is joined to the targeted examples set. The process stops when none untargeted example is interchanged between target and untargeted sets.

The generative models assume a distribution model of $p(\mathbf{x}_i|\mathbf{y}_j, \theta)$, where θ represent a set of parameters of the model which is identifiable by a Expectation-Maximization or clustering algorithms. The distribution $p(\mathbf{y}_j|\mathbf{x}_i, \theta)$ is computed using the Bayes rule assuming proportionality to $p(\mathbf{x}_i|\mathbf{y}_j, \theta)$ and $p(\mathbf{y}_j)$. The artificial target is computed by maximising $p(\mathbf{y}_j|\mathbf{x}_i, \theta)$ [12]. However, this method is usually applied to classification problems [6].

The graph based methods create models for the association between the inputs and target variables and between output variables themselves using graphs. The examples are the nodes and the weighted links represent their similarity. These models assume that neighbour input nodes tend to produce the similar targets [13]. This model allows to produce $p(\mathbf{y}_j|\mathbf{x}_i)$ (discriminative property) distribution as function of these parametrized associations and founds the target by optimizing it [14, 15]. The Continuous Conditional Random Fields (CCRF) explores the relations between input and target variables. A model is created for $p(\mathbf{y}_j|\mathbf{x}_i)$ based on graph parameters and optimised as a function of a target. This approach is an adaptation from classification [14].

3 Semi-Supervised Multi-Target AMRules

In this section, the Semi-supervised Multi-Target AMRules (SS-AMRules) algorithm and the underlying principles are presented. This SSL algorithm uses the AMRules regressor due to the modularity property that allows the construction of models for particular input variables regions(defined by the rule). The SSL algorithm is based on the assumption that the most likely input variables will benefit the model by reinforcing it even with the artificially targeted examples. This principle approximates to the semi-supervised smoothness assumption which states that if two points(target representation) are close in a very dense region, then their respective response points are also close [11]. If input variables

$\mathbf{x}_i$ are too frequent an artificial target is predicted and the respective example is targeted. The untargeted example occurrence uses the same principle of anomaly detection.

3.1 Rule Learning

Rule Learning is based on implications $R_r = (\mathcal{A}_r \Rightarrow \mathcal{C}_r)$, called rules, where antecedent $\mathcal{A}_r$ is a conjunction of conditions (called literals) that create partitions in the input variables $\mathbf{x}_i$ space and the consequent $\mathcal{C}_r$ is a predicting function (in this context, it is a basic online multi-target regressor). Literals may present the forms $(X_j \leq v)$ and $(X_j > v)$ for numerical data and $(X_j = v)$ and $(X_j \neq v)$ for nominal data, where X_j represents the j^{th} input variable. Rule R_r is said to cover $\mathbf{x}_i$ if $\mathbf{x}_i$ satisfies all the literals in $\mathcal{A}_r$. Support $S(\mathbf{x}_i)$ corresponds to a set of rules that cover $\mathbf{x}_i$ and $\mathcal{C}_r$ returns a prediction $\hat{\mathbf{y}}_i$ if a rule $R_r \in S(\mathbf{x}_i)$.

Data structure $\mathcal{L}_r$ containing the necessary statistics(about the rule and the examples) to the algorithm training and prediction (rule expansion, changes detection and anomalies detection,...) is associated to each rule R_r. In particular, $\mathcal{L}_r$ contains the input variables statistics $\mathcal{I}_r$. Default rule D exists for initial conditions and for the case of none of the current rules covers the example $(S(\mathbf{x}_i) = \emptyset)$. The antecedent of D and is initially empty. Rule set is formed by a set of U learned rules defined as $\mathcal{R} = \{R_1, \cdots, R_r, \cdots, R_U\}$ and a default rule D as depicted in Figure 1.

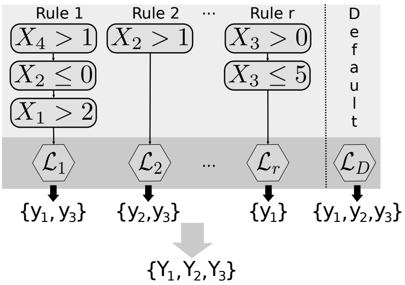


Fig. 1. Rule Learning on regression.

3.2 Untargeted Data Handling

The untargeted examples handling essentially performs likelyhood score and artificial target computation for the untargeted examples inputs $\mathbf{x}_i$. The selected likelyhood score was Odd Ratio (OR) measure defined by

$$OR_i = \frac{1}{M} \sum_{j=1}^{M} \log \left(\frac{P(|X_j - \mu_j| \geq |x_{i,j} - \mu_j|)}{1 - P(|X_j - \mu_j| \geq |x_{i,j} - \mu_j|)} \right), \tag{1}$$

where i is the example index, M is the number of input variables, X_j is the j^{th} input variable treated as a random variable and μ_j is the mean of X_j [4]. OR is the mean of probabilities measures($\frac{P(|X_j-\mu_j|\geq|x_{i,j}-\mu_j|)}{1-P(|X_j-\mu_j|\geq|x_{i,j}-\mu_j|)}$) of each input variable. The higher the OR is the better is the benefit of the artificially targeted example on the current model. For continuous attributes, the Cantelli's inequality [16] is used to estimate $P(|X_j - \mu_j| \geq |x_{i,j} - \mu_j|)$:

$$P(|X_j - \mu_j| \geq k) \leq \begin{cases} \frac{2\sigma_j^2}{\sigma_j^2+k^2} & \text{if } \sigma_j < k \\ 1 & \text{otherwise} \end{cases} \tag{2}$$

where σ_j is the standard deviation of X_j target element and k corresponds to the distance of the actual variable value and its mean $k = |x_{i,j} - \mu_j|$. The artificial target is computed by predicting the target for example using the current model associated to the rules that cover the examples.

3.3 SS-AMRules training

Algorithm 1 presents the pseudo-code of the SS-AMRules algorithm. On the occurrence of an incoming example $(\mathbf{x}_i, \mathbf{y}_i)$, the SS-AMRules searches for rules R_r of the current rule set that covers the example. Considering a $R_r \in S(\mathbf{x}_i)$, anomaly detection is performed to increase resilience to data outliers ($\texttt{isAnom-aly}(\mathcal{I}_r, \mathbf{x}_i)$). In case of anomaly occurrence, the example is simple ignored by the rule otherwise it verifies if the example is untargeted ($\mathbf{y}_i = \emptyset$). For an untargeted example, the OR score is computed from the input variables statistics $\mathcal{I}_r$ and if this score is higher than a predetermined threshold $scoreThreshold$, an artificial target is predicted for the example using the model statistics $\mathcal{L}_r$ and the example become targeted ($\texttt{getPrediction}(\mathcal{L}_r, \mathbf{x}_i)$). This prediction is the artificial target of the untargeted example. In case of targeted example identification, the statistics of likelihood score computation $\mathcal{I}_r$ are updated ($\texttt{updateInputStatistics}(\mathcal{I}_r)$).

Posteriorly, both target and artificially targeted examples are submitted to the change detector and training process. The Page-Winkle algorithm is used for change detection. The algorithm also performs outputs selection in order to create rules that are specialized on sets of output variables ($\texttt{Compute}\ \mathcal{O}_c$). The algorithm presents self-training and generative models features. The OR score is the measure of confidence that is estimated from a generative model which describes the input variables space. As consequent $\mathcal{C}_r$) function, a multi-target perceptron regressor was was used as linear predictor due to its models simplicity, low computational cost and low error rates [4].

4 The evaluation method

The evaluation consisted of a simulation of a data stream using artificial and real worlds datasets. A d percentage of each dataset examples are used to train

Algorithm 1 Semi-Supervised AMRules

```
$\mathcal{R} \leftarrow \emptyset,\ D \leftarrow 0$
for all $(\mathbf{x}_i, \mathbf{y}_i) \in \mathcal{D}$ do
    for all $R_r \in S(\mathbf{x}_i)$ do
        if $\neg$isAnomaly$(\mathcal{I}_r, \mathbf{x}_i)$ then
            if $y_i = \emptyset$ then
                $ORscore =$ getORScore$(\mathcal{I}_r, \mathbf{x}_i)$
                if $ORscore > scoreThreshold$ then
                    $\mathbf{y}_i =$ getPrediction$(\mathcal{L}_r, \mathbf{x}_i)$
                else
                    continue
            else
                updateInputStatistics$(\mathcal{I}_r)$
            if changeDetected$(\mathcal{L}_r, \mathbf{x}_i)$ then
                $\mathcal{R} \leftarrow \mathcal{R} \setminus \{R_r\}$
            else
                $R_c \leftarrow R_r$, update$(\mathcal{L}_r)$, $expanded \leftarrow$ expand$(R_r)$
                if $expanded =$ TRUE then
                    Compute $\mathcal{O}'_c$, $\mathcal{O}_c \leftarrow \mathcal{O}'_c$, $\mathcal{R} \leftarrow \mathcal{R} \cup \{R_c\}$

    if $S(\mathbf{x}_i) = \emptyset$ then
        if $\mathbf{y}_i = \emptyset$ then
            $ORscore =$ getORScore$(\mathcal{I}_r, \mathbf{x}_i)$
            if $ORscore > scoreThreshold$ then
                $\mathbf{y}_i =$ getPrediction$(\mathcal{L}_r, \mathbf{x}_i)$
        else
            updateInputStatistics$(\mathcal{I}_r)$
        update$(\mathcal{L}_D)$
        $expanded \leftarrow$ expand$(D)$
        if $expanded =$ TRUE then
            $\mathcal{R} \leftarrow \mathcal{R} \cup \{D\}$, $D \leftarrow 0$
```

a initial model and the remaining examples are used for testing. Iteratively, a binary random process choose if a test example is untarget with probability p. If the example is imputed as untargeted, the respective target is omitted from the SS-AMRules algorithm perspective. In the experiments, d is 30% in order to ensure initial model consistency. The chosen p probabilities were 50%, 80%, 90%, 95% and 99% which reflect different levels of untargeted examples content on the stream. This evaluation used the prequential mode where the algorithm predicts a target and the error for both targeted and untargeted examples (it uses the hidden targets). Posteriorly, it uses the example for training [17].

The performance measurements used the Error(euclidean norm of the difference between the true and the prediction vector) to measure an example prediction precision (local performance) and the Root Mean Square Error(RMSE) to evaluate global performance. The RMSE benefit was measured by computing the percentage of RMSE reduction between the scenario where no untargeted examples(reference) are used for training and the best scenario (associated to a score threshold) where artificial targeted examples are used. Since the error presented some isolated peaks and noise like aspect in the first experiments, a smoothing median filter was used to better observe the error tendency on the graphs [18]. The window size is 1000 and the window step is 1. Five versions of the Friedman artificial dataset with complex model were generated [19]. These datasets were produced with 128000 examples and each example has 10 inputs and 3 outputs. The models were produced by changing the weights of the complex functions. Regarding the real world datasets, Eunite03, Bicycle, SCM1d and SCM20d were used [4]. The examples of these datasets were replicated ten times and shuffled, since AMRules uses the Hoeffding bound to determine a sufficient number of examples to produce consistent models. Table 1 shows the original features of the real world data sets used in these experiments.

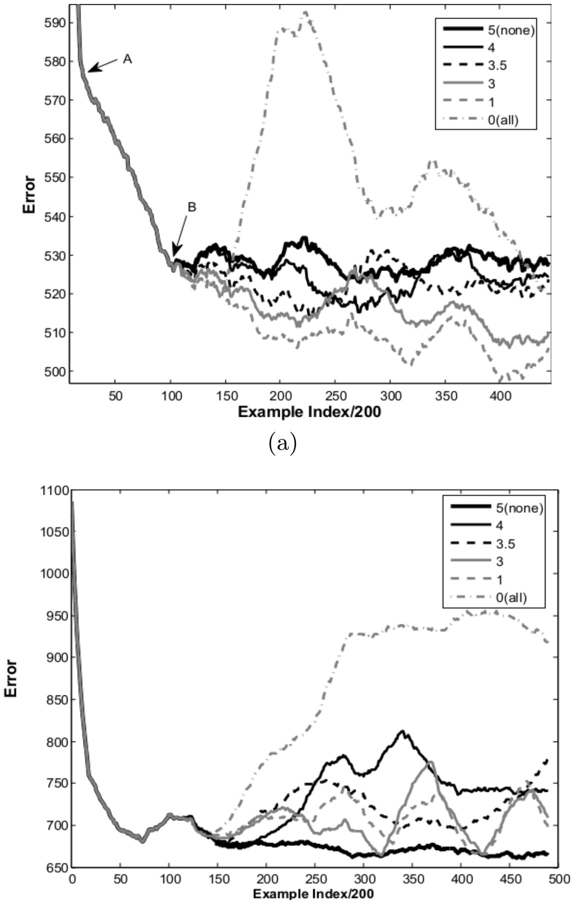
Table 1. Original datasets description

Dataset	# Examples	# Outputs	# Inputs
Eunite30	8064	5	29
Bicycle	17379	3	12
SCM1d	10103	16	280
SCM20d	9047	16	61

5 Results

In this section, the evaluation results are presented and discussed. Figure 2 presents the graphs of error evolution for two cases. The graph at the top reveals a successful error improvement and the graph at the bottom shows an unsuccessful improvement attempt. Each graph presents several curves for several score thresholds that were used to calibrate the algorithm. Curves for score thresholds 0, 1, 3, 3.5, 4 and 5 were selected for clearer plot presentation. The score thresholds 0 and 5 correspond to the scenarios where all and none untargeted examples were used in the training, respectively. The score threshold of 5 is the reference curve. The algorithm starts by training the initial model producing high errors with the first examples.

Referenced by point A on the graph at the top, the algorithm learns the first rules and the error decreases significantly. From point A to B, the algorithm improves the initial model and from point B, the algorithm starts to process untargeted examples. The curves diverge since the score threshold are different and lead to different behaviours on this phase. The graph of the successful case presents error curves (related to artificially targeted examples usage on the training) with lower values than the reference curve(scenario where none untargeted example is used). Moreover, these curves present an error reduction tendency. The score threshold 0 curve presents higher error because some of the accepted untargeted examples damaged the model.



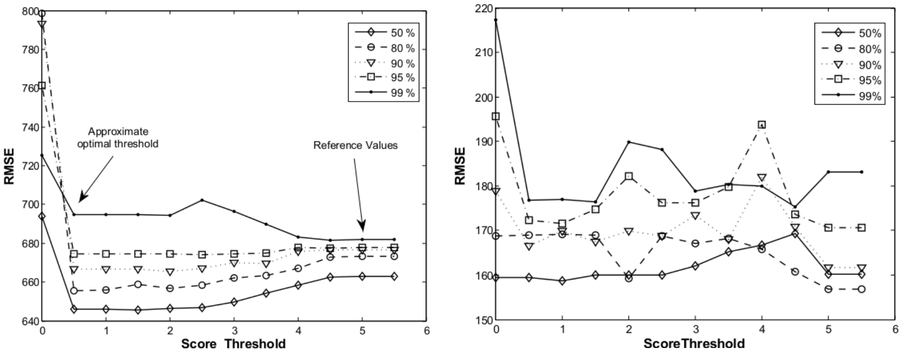
(a)

(b)

Fig. 2. Error evolution curves (for several score thresholds) of a successful(a) and a unsuccessful(b) cases of improvement attempts. The graph at the top shows the error behaviour of the algorithm for untargeted probability p=50% and for SCM20d dataset. The graph at the bottom shows for untargeted probability p=99% and for SCM1d dataset.

The graph at the bottom reveals a case where the algorithm produced less accurate models. This fact is due to error propagation through the model that lead to worst predictions in the artificial targeting. This effect leads to a cycle that reinforce the error on each untargeted example processing. In fact, the more untargeted examples arrive the higher is the error.

Figure 3 shows the RMSE as function of score threshold and the untargeted probability for two cases. The graph on the left indicates an approximate optimal score threshold (score threshold= 0.5) that rejects model damaging examples and accepts model reinforcing examples. This threshold is valid for any untargeted probability but it depends on the dataset characteristics(e.g, inputs variables distributions). The p=99% scenario is an extreme case where the model is trained essentially with artificially targeted examples and the error propagation can easily occur. The graph on the right presents a dataset where any score threshold produced higher error than the reference scenario. This observation means that most untargeted examples contributed to model damage and the artificial targets conveyed significant errors.



(a) (b)

Fig. 3. RMSE as a function of score threshold and untarget example probability. The graph on the left (a) shows a case of successful improvement using the dataset SCM20d and the graph on the right (b) shows a case of unsuccessful improvement attempt using the Bicycle dataset.

Table 2 presents the RMSE benefit for the experiments on artificial datasets for each chosen untargeted examples probabilities. When the value is zero, it means that there was not any score threshold that improved the model.

Table 2. RMSE benefit(%) for artificial datasets.

Datasets	Untargeted examples probabilities				
	50%	80%	90%	95%	99%
FriedModel1	0,13	0,00	0,26	0,00	3,97
FriedModel2	6,83	7,23	6,73	2,72	0,15
FriedModel3	0,01	0,01	0,01	0,00	0,00
FriedModel4	9,11	22,38	21,62	14,65	5,11
FriedModel5	2,31	2,81	1,26	0,00	0,00

RMSE benefit values show that the error decreased in most part of the artificial datasets and for several untargeted examples probabilities. For FriedModel2, FriedModel4 and FriedModel5 datasets, a significant improvement was achieved.

On the other hand, for FriedModel1 and FriedModel3 datasets, the RMSE improvements were very small. Table 3 presents the RMSE benefits for real world datasets in a similar way as the artificial datasets. According to Table 3, the algorithm seem benefit most part of the scenarios. As expected, the more elevated p is the less is the benefit. For Eunite03 and SCM20d datasets, the algorithm produced significant results. But in particular for Bicycle dataset, error reduction did not occur for most untargeted examples probabilities. As general impression, the error evolution graphs, the RMSE graphs and the RMSE benefit tables support the view that the algorithm leads to an online error reduction by using untargeted examples in most cases and in scenarios where untargeted probabilities is high.

Table 3. RMSE benefit (%) for real world datasets.

Datasets	Untargeted examples probabilities				
	50%	80%	90%	95%	99%
Bicycle	0,48	0,00	0,00	0,00	4,23
Eunite03	8,67	14,38	32,04	4,77	0,00
SCM1d	0,87	0,02	0,00	8,65	0,00
SCM20d	2,58	2,60	1,68	0,53	0,00

6 Conclusion

In this paper, an online semi-supervised multi-target regression algorithm is addressed. The algorithm reduces the error of prediction on online mode by using untargeted examples in most of evaluation experimental scenarios. However, the error reduction still very small and the score thresholds depend on the dataset. In fact, the algorithm implies an calibration of the score threshold for each data stream.

As future work, this approach can be improved by combining it with Random Rules based algorithms due to multiple prediction diversity feature. Since it is important to know what are the dataset characteristics that lead to error reduction in a semi-supervised scenario, an analytical framework could be constructed. In order to increase the algorithm validity, the evaluation tests will be performed using a higher number of real world datasets with a significant amount of examples.

7 Acknowledgments

This work is financed by the ERDF European Regional Development Fund through the Operational Programme for Competitiveness and Internationalisation - COMPETE 2020 Programme within project POCI-01-0145-FEDER-006961, and by National Funds through the FCT Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) as part of project UID/EEA/50014/2013.

References

1. Hanen Borchani, Gherardo Varando, Concha Bielza, and Pedro Larrañaga. A survey on multi-output regression. *Wiley Int. Rev. Data Min. and Knowl. Disc.*, 5(5):216–233, September 2015.
2. Jurica Levatic, Michelangelo Ceci, Dragi Kocev, and Saso Dzeroski. Semi-supervised learning for multi-target regression. In *Third International Workshop, NFMCP 2014, Held in Conjunction with ECML-PKDD 2014*, pages 3–18, 2014.
3. Zhi hua Zhou, Senior Member, and Ming Li. Semi-supervised regression with co-training style algorithms. *IEEE Transactions on Knowledge and Data Engineering*, page 2007.
4. João Duarte and João Gama. Multi-Target Regression from High-Speed Data Streams with Adaptive Model Rules. In *IEEE conference on Data Science and Advanced Analytics*, 2015.
5. Andrew B. Goldberg, Xiaojin Zhu, Alex Furger, and Jun-Ming Xu. OASIS: online active semi-supervised learning. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2011, San Francisco, California, USA, August 7-11, 2011*, 2011.
6. Pilsung Kang, Dongil Kim, and Sungzoon Cho. Semi-supervised support vector regression based on self-training with label uncertainty: An application to virtual metrology in semiconductor manufacturing. *Expert Syst. Appl.*, 51:85–106, 2016.
7. P. Ozoh, S. Abd-rahman, J. Labadin, and M. Apperley. Article: A comparative analysis of techniques for forecasting electricity consumption. *International Journal of Computer Applications*, 88(15):8–12, February 2014.
8. Zaid Chalabi Punam Mangtani Masahiro Hashizume Chisato Imai, Ben Armstrong. Article: Time series regression model for infectious disease and weather. *International Journal of Environment Research*, (142):319–327, June 2015.
9. Huseyin Sekerc Volkan Uslana. Article: Quantitative prediction of peptide binding afnity by using hybrid fuzzy support vector regression. *Applied Soft Computing*, (43):210–221, January 2016.
10. Adebiyi A. Ariyo, Adewumi O. Adewumi, and Charles K. Ayo. Stock price prediction using the arima model. In *Proceedings of the 2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation*, UKSIM '14, pages 106–112, Washington, DC, USA, 2014. IEEE Computer Society.
11. Olivier Chapelle, Bernhard Schlkopf, and Alexander Zien. *Semi-Supervised Learning.* The MIT Press, 1st edition, 2010.
12. Amparo Albalate and Wolfgang Minker. *Semi-Supervised and Unsupervised Machine Learning.* ISTE/Wiley, London (United Kingdom), 2011.
13. Jakob J. Verbeek and Nikos Vlassis. Gaussian fields for semi-supervised regression and correspondence learning. *Pattern Recogn.*, 39(10):1864–1875, October 2006.
14. Vladan Radosavljevic, Slobodan Vucetic, and Zoran Obradovic. Continuous conditional random fields for regression in remote sensing. In *Proceedings of the 2010 Conference on ECAI 2010: 19th European Conference on Artificial Intelligence*, pages 809–814, Amsterdam, The Netherlands, The Netherlands, 2010. IOS Press.
15. Jelena Stojanovic, Milos Jovanovic, Djordje Gligorijevic, and Zoran Obradovic. Semi-supervised learning for structured regression on partially observed attributed

graphs. In *2015 SIAM International Conference on Data Mining (SDM 2015)*, 2015.

16. BB Bhattacharyya. One sided Chebyshev inequality when the first four moments are known. *Communications in Statistics-Theory and Methods*, 16(9):2789–2791, 1987.
17. João Gama, Raquel Sebastião, and Pedro Pereira Rodrigues. On evaluating stream learning algorithms. *Machine Learning*, 90(3):317–346, 2013.
18. Wai-Kai Chen. *Passive, Active, and Digital Filters*. CRC Press, third edition, 2009.
19. Jerome H. Friedman. Multivariate adaptive regression splines. *The Annals of Statistics*, 19(1):1–67, 1991.