

Heuristic Rule-Based Regression via Dynamic Reduction to Classification

Frederik Janssen and Johannes Fürnkranz



TECHNISCHE
UNIVERSITÄT
DARMSTADT

1. Motivation
2. Separate-and-conquer Regression Rule Learning
 - ▶ Regression by Classification
 - ▶ Description of the algorithm
3. Dynamic Reduction to Classification
4. Experimental Setup
5. Results
 - ▶ Dynamic vs. Static Regression by Classification
 - ▶ Comparison with other algorithms
6. Conclusions and Future Work

- ▶ lack of separate-and-conquer based rule learning algorithms for regression
- ▶ simple and elegant technique
- ▶ generation of simple rules that are interpretable
- ▶ heuristics for regression are rare and hard to define
- ▶ **but:** heuristics for classification are well researched

- ▶ adaptation to Regression either by discretizing the numeric outcome and afterwards use standard classification algorithms
 - ▶ methods used to discretize: P-CLASS (Weiss and Indurkha, 1995), equal-frequency, equal-width
 - ▶ problem: number of classes has to be known in advance
- ▶ or by adapting the separate-and-conquer technique to Regression tasks
 - ▶ **idea in this work:** dynamically label examples as positive and negative ones
 - ▶ examples that are predicted well are labelled as positives
 - ▶ examples with a higher error are labelled as negatives
 - ▶ **advantage:** classification heuristics can be re-used (they depend on a notion of positive and negative examples)

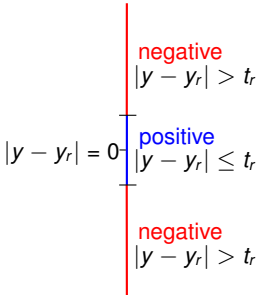


- ▶ we integrated the dynamic reduction in a classification rule learner that is based on the separate-and-conquer strategy
- ▶ the learned rules should predict a single value (not a linear model)
- ▶ a decision list was used for classification (i.e., only one rule is used for prediction)
- ▶ the rule set has to cover at least 90% of the examples
 - ▶ this value was found in a previous work (Janssen and Fürnkranz, 2010b) and was a good choice there
- ▶ the static reduction to classification was used for comparison
 1. discretize the class variable (equal-frequency)
 2. use the classification-version of the rule learner on the discretized data

Dynamic Reduction to Classification

Computation of covered positive/negatives

- ▶ the reduction is implemented with a threshold t_r
- ▶ to compute t_r we used the standard deviation (σ) and also tried to slightly increase or decrease it (by setting factor = 0.95 and factor = 1.05)
- ▶ for each refinement compute the standard deviation and label the covered examples as positive/negative

$$t_r = \text{factor} \cdot \sigma_r$$
$$\text{class}(x) = \begin{cases} \text{positive} & \text{if } |y - y_r| \leq t_r \\ \text{negative} & \text{if } |y - y_r| > t_r \end{cases}$$


$|y - y_r| > t_r$ negative

$|y - y_r| = 0$ positive

$|y - y_r| > t_r$ negative

where x is the current example, y is the true value of the example x , and y_r is the value predicted by rule r .

Dynamic Reduction to Classification

Computation of total positive/negatives

- ▶ some heuristics also need the total statistics of the dataset, i.e., the total positive/negative examples
- ▶ to compute them, the same mechanism as above was used

$$P_r = \sum_{i=1}^m \mathbf{1}(|y_i - y_r| \leq t_r), \quad N_r = m - P_r$$

where m = number of examples, and $\mathbf{1}(\cdot)$ is the indicator function.

Experimental Setup

Heuristics

- ▶ we selected 4 different heuristics with different preferences
- ▶ laplace (lap) = $\frac{p+1}{p+n+2}$ known to overfit
- ▶ weighted relative accuracy (wra) = $\frac{p}{P} - \frac{n}{N}$ known to underfit
- ▶ correlation (corr) = $\frac{p \cdot N - n \cdot P}{\sqrt{P \cdot N \cdot (p+n) \cdot (P-p+N-n)}}$ stable heuristic (cf. (Janssen and Fürnkranz, 2010a))
- ▶ relative cost (rcm) = $c \cdot \frac{p}{P} - (1 - c) \cdot \frac{n}{N}$ with parameter $c = 0.342$

where p and n are the covered positive/negative examples and P and N are the total positive/negative examples

Experimental Setup

Algorithm and Dataset Setup

Algorithm Setup

- ▶ Other Rule-based regression algorithms
 - ▶ M5RULES (Holmes et al., 1999) in default mode and with prediction of single value (-R)
 - ▶ REGENDER (Dembczyński et al., 2008), in default configuration (50 rules), and in setting recommended by the authors (200 rules, different loss function, and different optimization technique)
- ▶ Other standard regression algorithms
 - ▶ LINEAR REGRESSION, MULTILAYER PERCEPTRON, and SVMREG
- ▶ Static reduction to classification
- ▶ bagged versions of our algorithm (in order to reduce its restriction to piecewise constant predictions)

Dataset Setup

- ▶ for comparison 16 datasets from UCI Repository and Luís Torgos website were used

Results

Dynamic vs. Static Regression by Classification

Dynamic Regression by Classification

factor	heuristic	rmse	rank	# rules	# conds
0.95	wra	0.752	8.63	15.06	38.31
0.95	lap	0.784	11.19	11.25	13.88
0.95	corr	0.726	6.50	10.13	24.63
0.95	rcm	0.780	9.81	19.06	34.25
1.00	wra	0.764	10.06	17.06	47.81
1.00	lap	0.774	10.63	10.19	12.50
1.00	corr	0.753	8.38	9.25	22.06
1.00	rcm	0.767	9.50	19.06	35.75
1.05	wra	0.780	13.13	13.19	34.19
1.05	lap	0.772	10.19	9.69	11.81
1.05	corr	0.796	12.88	10.25	33.31
1.05	rcm	0.775	9.75	19.44	37.56

Static Regression by Classification

# classes	heuristic	rmse	rank	# rules	# conds
5	wra	0.883	18.25	5.63	20.75
5	lap	0.857	14.75	84.56	197.44
5	corr	0.844	15.13	28.06	84.00
5	rcm	0.852	16.63	22.88	68.00
10	wra	0.930	18.69	6.06	23.13
10	lap	0.872	17.00	138.44	339.25
10	corr	0.864	15.88	49.31	167.25
10	rcm	0.901	17.94	20.75	67.31
20	wra	0.965	20.81	10.06	36.56
20	lap	0.872	18.06	177.44	423.63
20	corr	0.862	17.81	95.13	295.00
20	rcm	0.928	19.13	33.19	102.13

- ▶ Dynamic Regression outperforms Static Regression significantly (best setting outperforms all but two static approaches with $p = 0.1$)
- ▶ correlation with a factor of 0.95 is the best choice
- ▶ preferences of the heuristics known from classification do not carry over to the dynamic approach, i.e., wra learns more rules than laplace

Results

Comparison with other algorithms I

algorithm	<i>rrmse</i>	rank	# rules	# conds
Dynamic Approach with <i>correlation</i> and $0.95 \cdot \sigma$	0.726	7.06	10.13	24.63
Bagged (10 iterations)	0.671	5.88	97.94	245.81
Bagged (20 iterations)	0.659	4.94	186.75	451.25
Bagged (50 iterations)	0.658	4.63	465.88	1146.6
LR	0.651	4.31	—	—
MLP	0.746	5.88	—	—
SVMreg	0.673	5.19	—	—
RegENDER	0.679	4.50	200.00	1163.6
M5Rules	0.604	2.63	2.94	5.38
RegENDER (50)	0.768	—	50.00	190.00
M5Rules -R	0.773	—	6.19	14.94

- ▶ bagged versions of the algorithm work comparable to state-of-the-art algorithms
- ▶ Dynamic approach outperforms other comparable algorithms (REGENDER with 50 rules and M5RULES-R with prediction of a single value)

Results

Comparison with other algorithms II

- Interpretation of the results by a *CD*-chart as suggested by (Demšar, 2006)

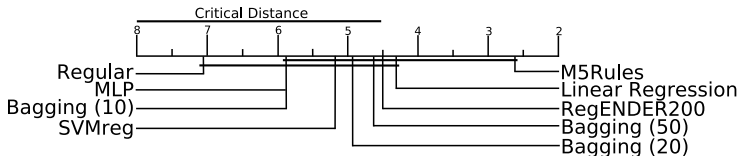


Figure: Comparison of the algorithms against each other with the Nemenyi test. Groups of algorithms that are not significantly different (at $p = 0.01$) are connected.

Conclusions

- ▶ Dynamic Reduction to Classification allows to use classification heuristics directly
- ▶ Dynamic Reduction to Classification outperforms the Static Approach (a priori discretization of class variable)
- ▶ Dynamic Approach is en par with other rule-based regression algorithms

Future Work

- ▶ Systematic evaluation of the factor that is applied to the standard deviation
- ▶ Experiments with the number of examples that remain uncovered
- ▶ apply linear models as in M5RULES (it seems that linear models have the greatest impact)

- ▶ (Dembczyński et al., 2008): K. Dembczyński, W. Kotłowski, and R. Słowiński. Solving Regression by learning an Ensemble of Decision Rules. In *Proc. 9th International Conference on Artificial Intelligence and Soft Computing (ICAISC-08)*, pp. 533–544, Zakopane, Poland, 2008. Springer-Verlag.
- ▶ (Demšar, 2006): Janez Demšar. Statistical Comparisons of Classifiers over Multiple Data Sets. *Machine Learning Research*, Jan(7):1Ú30, 2006.
- ▶ (Holmes et al., 1999): G. Holmes, M. Hall, and E. Frank. Generating rule sets from model trees. In *Proc. 12th Australian Joint Conference on Artificial Intelligence (AI-99)*, pp. 1–12. Springer, 1999.
- ▶ (Janssen and Fürnkranz, 2010a): F. Janssen and J. Fürnkranz. On the Quest for Optimal Rule Learning Heuristics. *Machine Learning*, 78(3): 343-379, 2010.
- ▶ (Janssen and Fürnkranz, 2010b): F. Janssen and J. Fürnkranz. Separate-and-conquer Regression. In *Proc. of the German Workshop on Lernen, Wissen, Adaptivität - LWA2010*, pp. 81–89, Kassel, Germany, 2010.
- ▶ (Weiss and Indurkha, 1995): S. M. Weiss and N. Indurkha. Rule-based Machine Learning Methods for Functional Prediction. *Journal of Artificial Intelligence Research*, 3:383-403, 1995.
- ▶ Luís Torgos website: <http://www.liaad.up.pt/~ltorgo/Regression/DataSets.html>