Histogram-based Probabilistic Rule Lists for Numeric Targets

Lincen Yang[0000−0003−1936−2784], Tim Opdam[0000−0002−9459−2646], and Matthijs van Leeuwen[0000−0002−0510−3549]

Leiden University, Leiden, The Netherlands
l.yang@liacs.leidenuniv.nl; timopdam@hotmail.com;
m.van.leeuwen@liacs.leidenuniv.nl

Abstract. While most rule learning methods focus on categorical targets for tasks including classification and subgroup discovery, rule learning for numeric targets are under-studied, especially using probabilistic rules: the only existing probabilistic rule list method for numeric targets, named SSD^{++}, is based on Gaussian parametric models. To extend this method, we adopt an adaptive histogram model to estimate the probability distribution of the target variable. We formalize the rule list learning as a model selection problem, which we tackle with the minimum description length principle. We demonstrate that our method produces rule lists with better predictive performance than SSD^{++}.

Keywords: Rule list · Regression rule · MDL model selection.

1 Introduction and Related Work

Learning rules from data is a long studied problem in inductive reasoning, data mining, and machine learning. It has been widely used in practice as rules are directly readable by analysts and domain experts, revealing actionable insights for real-world data-driven tasks. However, rule learning for numeric targets is under-studied, especially the task of inducing a rule-based global model for describing the whole dataset.

Classic rule learning algorithms often follow the separate and conquer strategy: learn a single rule from the dataset, remove the covered instances, and repeat the process until some stopping criterion is met. As a result, the separate and conquer strategy leads to an ordered list of rules, also referred to as decision list or rule list. While it is proved to be efficient in classification tasks [6], the extension from categorical to numeric targets is non-trivial.

The criteria used in rule lists for categorical targets are often based on probabilistic estimates (i.e., precision, false positive rate, etc), including 1) heuristics for evaluating individual rules as local patterns, 2) global evaluation metrics for a rule list model, and 3) statistics used for statistical significance testing in order to prevent overfit [5]. However, for numeric targets, performing probabilistic density estimation can be computationally expensive, especially when doing

this a large number of times, i.e., each time when a rule is assessed. Specifically, standard approaches like kernel density estimation (KDE) sutffer from a high computational cost (mainly due to the cross-validation needed for choosing hyper-parameters), which makes it unsuitable for learning rule lists. Parametric models, however, are fast but may lead to mis-specification. One existing method named SSD^{++} [11] assumes Gaussian distribution for the numeric targets of instances covered by a certain individual rule, but this is theoretically sub-optimal and leads to self-contradiction. This is because any individual rule can be regarded as the "union" of its refinements: for instance, data points satisfying some condition (A_1) is the union of the data points satisfying ($A_1 \wedge A_2$) and ($A_1 \wedge \neg A_2$). However, the target variable within the previous three rules cannot be assumed to be Gaussian at the same time, as the first one is by definition the mixture of the other two.

To strike a balance between the computational cost and model expressiveness, we adopt adaptive histograms for density estimation for learning rules for numeric targets. Our contribution are as follows: 1) we formalize the problem of learning rule lists as a model selection task; 2) by showing that the model selection criterion of the rule lists can be decomposed to the sum of local criterion for each individual rule, we demonstrate our model selection criterion to be compatible with the separate and conquer strategy; 3) we empirically showcase that the rule lists obtained by our method outperforms that of SSD^{++} in terms of prediction accuracy, measured by the mean squared error.

Related Work. Far fewer rule learning methods exist for numeric targets than categorical targets. Besides SSD^{++}, PRIM [3] proposes to sequentially search for the rules that deviate from the global mean, which leads to a non-probabilistic rule lists. Further, DSSD [13] proposes to learn a *diverse* set of non-probabilistic rules for numeric targets. Next, instead of learning rules directly from numeric targets, an alternative approach is to transfer regression rule learning to classification rule learning by dynamically cutting the targets [8]. Also, Meeng and Knobbe thoroughly discuss about dealing with numeric targets by discretization in rule learning for *subgroup discovery* [10]. Last, RuleFit [4] can generate rules for numeric targets for constructing an ensemble model, which is less interpretable than a single rule list. Note that some of these methods are developed for the subgroup discovery task instead of predictive rule learning. Specifically, our competitor SSD^{++} is originally proposed for subgroup discovery; however, as pointed out in its original paper, it can be used for regression by formalizing the rule list as a probabilistic model in a slightly different way.

2 Probabilistic Rule Lists for Numeric Targets

Adaptive histogram for numeric targets. Consider a one-dimensional random variable Y taking values in $\mathbb{R}$, a histogram model for Y is a set of cut points (including boundaries), denoted as $\boldsymbol{c} = (c_1, \ldots, c_{K+1})$, where K represents the number of bins. For any value y in the support of Y, the associated probability distribution, denoted by $P_h(.)$, has density function defined and denoted

as $p_h(Y = y) = \sum_{j=1}^{K} \mathbf{1}_{[c_j, c_{j+1})}(y)\beta_j$, where $\mathbf{1}(.)$ is the indicator function and $\beta = (\beta_1, ..., \beta_K)$ is the parameter vector to be estimated from data; i.e., β represents the probability of Y taking values in each of all bins. In practice, β can be estimated by the maximum likelihood estimator: $\hat{\beta}_j = \frac{|\{c_j \leq y < c_{j+1}\}|}{|S|\ (c_{j+1} - c_j)}$, where $|.|$ is the cardinality function.

Histogram-based rule. Consider the d-dimensional feature variables $X = (X_1, \ldots, X_d)$, where each X_i a single dimension of X, and a target variable $Y \in \mathbb{R}$. A histogram-based rule is written as $(X_1 \in R_1 \wedge X_2 \in R_2 \wedge \ldots) \rightarrow P_{h,S}(Y)$, where each $X_i \in R_i$ is called a *literal* of the *condition* of the rule. Specifically, each R_i is a closed interval or a set of categorical levels. A rule in this form describes a subset S of the full sample space of X, such that for any $x \in S$, the conditional distribution $P(Y|X = x)$ is approximated by the probability distribution of Y *conditioned on the event* $\{X \in S\}$, i.e., $P(Y|X \in S)$, which is modelled by an adaptive histogram model, associated with S and hence denoted as $P_{h,S}(Y)$. Thus, a histogram-based rule is a local probabilistic model for all instances that satisfy the rule. To simplify the notation, when clear from the context we use S to refer to the rule itself and the subset of all instances covered by the rule.

Histogram-based rule list. Precisely, a rule list is an ordered list of rules connected by the "If...ElseIf...Else..." statements. For instance, a rule list containing only two rules S_1 and S_2 can be written as *"IF* $x \in S_1$*, THEN* P_{h,S_1}*; ELSE IF* $x \in S_2$*, THEN* P_{h,S_2}*; ELSE:* P_{h,S_0}*"*. Note that we use P_{h,S_0} to represent the histogram that is associated with the instances *not covered by any rule*, and referred to S_0 as the "else rule"[1]. Therefore, a rule list connects multiple rules and hence becomes a global model for the whole dataset: given any instance (x, y), we can calculate the probability (density) by firstly going over the rule list until x satisfies the condition of a certain rule, and then predicting the density function of y conditioned on x by the histogram model associated with that rule.

3 Learning Rule Lists as a Model Selection Task

We formalize the task of learning a rule list as a probabilistic model selection task. We adopt the minimum description length (MDL) principle [7] for the model selection task. The MDL-based model selection has theoretic roots in information theory and can be regarded as an extension of Bayesian model selection. Further, it has a long history of being successfully applied in rule learning, including classic methods like C4.5 and RIPPER [12, 2]. In practice, the MDL-based model selection criterion can be viewed as a form of penalized maximum likelihood, as we will elaborate next. We start with the criterion for optimal histograms of individual rules and then discuss the global criterion for rule lists.

[1] A related and widely used notion is the "default rule". The difference lies in whether the parameters associated with S_0 is estimated by all instances (default rule) or by instances that are not covered by any rule in the rule list (else rule).

3.1 Preliminaries: learning adaptive histograms for individual rules

Consider an individual rule S (with a histogram with fixed cut points) and all the m instances satisfying the condition of S, denoted as (x^m, y^m). Formally, the optimal adaptive histogram h^* among all possible histograms $\mathcal{H}$ is defined as $h^* = \arg\min_{h\in\mathcal{H}} \left(-\log \hat{p}_{h,S}(Y^m = y^m) + \mathcal{R}(m, K)\right)$; note that 1) $\hat{p}_{h,S}$ is the estimated probability density function with the maximum likelihood estimator for $\boldsymbol{\beta}$, 2) $\hat{p}_{h,S}(y)$ is extended to $\hat{p}_{h,S}(y^m)$ with the i.i.d assumption, and 3) $\mathcal{R}(m, K)$ is the so-called regret, the "penalty term" of the MDL model selection criterion, which is a function of sample size m and the number of bins of the histogram K [9]. By definition, $\mathcal{R}(m, K) = \log \int_{z^m} \hat{p}_{h,S}(Y^m = z^m)$ [7]. The seminal work in this research line [9] developed a dynamic programming optimization algorithm with quadratic time complexity, despite the expensive numeric integral in $\mathcal{R}$.

3.2 Model selection for learning histogram-based rule lists

The task of learning a rule list needs to simultaneously select the cut points on features (for constructing the rules' conditions) and on targets (for constructing the adaptive histograms). Formally, denote the (training) dataset as $D = (x^n, y^n)$, then the best histogram-based rule list, denoted as M^*, among all possible rule lists $\mathcal{M}$, is defined as

$$M^* = \arg\min_{M\in\mathcal{M}} L(D, M) = \arg\min_{M\in\mathcal{M}} -\log \hat{P}(y^n|x^n) + \mathcal{R}(M) + L(M), \quad (1)$$

where 1) $\hat{P}(y^n|x^n)$ is the likelihood for the data, based on the maximum likelihood estimator of the parameter $\boldsymbol{\beta}$ for all histograms; 2) $\mathcal{R}(M)$ is the MDL regret term for the rule list as a global model, defined as $\mathcal{R}(M) = \log \int_{z^m} \hat{P}(Y^m = z^m|x^n)$, which can be proved to be the sum of the regret terms $\mathcal{R}$ of all individual rules [14]; and 3) $L(M)$ is the code length, in bits, that is needed to encode the rule list as a model [7]. Note that for a fixed $M \in \mathcal{M}$, not only the conditions of the rule but also the cut points for associated histograms are fixed. To calculate $L(M)$, we sum up the code lengths needed to encode the following: the number of rules in M, the number of literals for each all individual rules in M, and the exact variable and value used for the condition of each literal [7]. The formula for calculating the length is the same as that of the method SSD^{++} [11], and hence we skip the mathematical details here.

3.3 Algorithm: separate and conquer

As exhaustive search in rule learning is known to be computationally prohibitive [6], we resort to the separate and conquer strategy: we iteratively search for the next rule, add it to the rule list, and remove the covered instances, until adding a new rule brings no further decrease in minimizing $L(D, M)$. Since 1) the regret terms $\mathcal{R}$ of the rule list can be written as the sum of the regret terms of each individual rule, 2) $L(M)$ can be decomposed to individual rules by definition, and 3) the log-likelihood can be decomposed to the likelihood of each rule, the

global model selection criterion $L(D, M)$ can be decomposed to the sum of local criterion of each individual rule. Hence, the separate and conquer strategy can be viewed as a greedy manner of optimizing the global criterion.

To search for the next rule based on the current (incomplete) rule list, we grow the rule by iteratively adding literals to an empty rule. Intuitively, we should not search for the next rule by minimizing $L(D, M)$ directly, as our goal is not to minimize $L(D, M)$ by adding one more rule only. Instead, we should search for the next rule such that by adding this rule to the rule list, we take a step towards our final rule list in the "steepest direction". Informally, the "steepest direction" can be thought of the direction that reduces $L(D, M)$ most *per extra covered instance*. That is, we use the heuristic named "normalized gain" [11]. Similar to SSD^{++} we use beam search when growing individual rules to avoid (some) local optima. The rule growing procedure continues as long as the normalized gain is positive, i.e., $L(D, M)$ keeps decreasing. Further, the rule list learning is stopped when adding a new rule will not decrease the global criterion further.

4 Empirical Performance

We benchmark our method against SSD^{++} with widely used UCI datasets for regression tasks, to investigate the predictive performance improvement by adopting the non-parametric histogram models. We also include the results of CART [1] as a baseline interpretable model, for which we use the implementation from scikit-learn with post-pruning.

We report the mean squared error (MSE) and the total number of literals in the rule list in Table 1, and the results are obtained by 10 random train/test splits, in which 80% of the instances are used for training. We show that our histogram-based approach outperforms SSD^{++} in most datasets. Thus, the Gaussian assumption will indeed lead to sub-optimal results for rule learning. Further, we observe that CART outperforms our method in about half the datasets, but CART in general produces more complicated models than our method in terms of the number of literals in each rule (path from root to leaf).

Next, we observe that SSD^{++} produces much simpler models than our method. This can be explained as follows: SSD^{++} and our method both use the MDL principle to control the model complexity, and hence the rule can grow only when the "gain" in likelihood exceeds the "cost" in the regret term and model complexity. Since adaptive histograms are much more expressive than Gaussian parametric models, it is "easier" for the rule to obtain substantial "gain" in likelihoods, which drives the rules to grow longer and hence cover smaller subsets. As a result, more rules are needed to cover the whole dataset and hence our method produces longer rules and a larger number of rules.

5 Conclusion and Discussion

We developed a rule list method for numeric targets with the adaptive histogram model, and we demonstrated that the predictive performance is superior to the

	MSE			# total literals		
data	Hist. Rule List	SSD^{++}	CART	Hist. Rule List	SSD^{++}	CART
cholesterol	3287.72	3286.12	4081.33	12.17	1.92	26.85
autoMPG8	12.26	12.35	13.14	29.64	14.16	163.23
dee	0.64	0.89	0.26	24.85	4.62	63.62
ele-1	458790.85	481793.78	536584.54	26.25	13.36	51.99
forestFires	6520.97	7147.07	4447.27	85.19	61.31	15.29
concrete	72.19	74.86	50.66	126.88	53.98	2215.52
abalone	6.09	6.31	5.69	121.29	62.86	131.84

Table 1. Predictive performance measured by MSE, and model complexity measured by the total number of literals in each model. The results are obtained in 10 random train/test splits for each dataset.

existing method based on parametric Gaussian models. The limitation of our method is scalability, due to the quadratic complexity of searching the optimal adaptive histograms: for datasets with tens of features and thousands of sample sizes, the training process can take a few minutes. The future work may be focused on developing methods for calculating the "score" of each rule list approximately but efficiently.

References

1. Breiman, L., Friedman, J., Stone, C.J., Olshen, R.A.: Classification and regression trees. CRC press (1984)
2. Cohen, W.W.: Fast effective rule induction. In: Machine learning proceedings 1995, pp. 115–123. Elsevier (1995)
3. Friedman, J.H., Fisher, N.I.: Bump hunting in high-dimensional data. Statistics and computing **9**(2), 123–143 (1999)
4. Friedman, J.H., Popescu, B.E.: Predictive learning via rule ensembles. The annals of applied statistics pp. 916–954 (2008)
5. Fürnkranz, J., Flach, P.A.: Roc 'n'rule learning—towards a better understanding of covering algorithms. Machine learning **58**(1), 39–77 (2005)
6. Fürnkranz, J., Gamberger, D., Lavrač, N.: Foundations of rule learning. Springer Science & Business Media (2012)
7. Grünwald, P.D.: The minimum description length principle. MIT press (2007)
8. Janssen, F., Fürnkranz, J.: Heuristic rule-based regression via dynamic reduction to classification. In: Twenty-Second International Joint Conference on Artificial Intelligence (2011)
9. Kontkanen, P., Myllymäki, P.: Mdl histogram density estimation. In: Artificial intelligence and statistics. pp. 219–226. PMLR (2007)
10. Meeng, M., Knobbe, A.: For real: a thorough look at numeric attributes in subgroup discovery. Data Mining and Knowledge Discovery **35**(1), 158–212 (2021)
11. Proença, H.M.e.a.: Discovering outstanding subgroup lists for numeric targets using mdl. In: ECMLPKDD 2020. Springer (2020)
12. Quinlan, J.R.: C4. 5: programs for machine learning. Elsevier (2014)
13. Van Leeuwen, M., Knobbe, A.: Diverse subgroup set discovery. Data Mining and Knowledge Discovery **25**(2), 208–242 (2012)
14. Yang, L., van Leeuwen, M.: Truly unordered probabilistic rule sets for multi-class classification. arXiv preprint arXiv:2206.08804 (2022)