Robust rules for prediction and description

Manuel Proenca, H.

3

MDL for rule lists

In this chapter[1] we formalize the task of finding predictive rule lists and subgroup lists as a model selection problem using the Minimum Description Length (MDL) principle [107, 48, 47].

In the previous chapter, we defined the rule list model, which we recall here in Figure 3.1. Now, the remaining question is how to select adequate models. For that, we resort to the MDL principle, which can be paraphrased as "*induction by compression*" and roughly states that the best model is the one that best compresses the data. The idea of compression can seem unintuitive at first. Still, one should notice that it is intimately connected to the concept of probability, i.e., the model that has the highest probability given the data is the same that maximizes compression. This idea was first formally stated by Shannon [114], which tells us that the optimal length of the encoding for an event A—smaller length corresponds to higher compression—equals the negative logarithm of the probability of that event, thus

$$L(A) = -\log \Pr(A), \tag{3.1}$$

where $L(A)$ is the length of the encoding for the event. To be objective, the MDL principle attempts to make the minimum number of assumptions about the model class. At this point, we should recall that the models we are trying to select from the data are rule lists and that, depending on the type of task, we use predictive rule lists and subgroup lists for machine learning or data mining respectively. Both models have the same model structure (that of Figure 3.1) and only differ in how the parameters of the default rule are estimated.

[1] Parts of this chapter are based on Proença and van Leeuwen [96], Proença et al. [99, 100]

In the case of a subgroup list, the default rule is fixed to the marginal distribution of each target, making its parameters *known* and *fixed* for a certain dataset [99, 100]. In the case of a predictive rule list, however, the last rule is 'free' in the sense that it depends on the estimate of its subset $\mathbf{Y}^d$ [96].

$$
\begin{array}{lllllll}
1: & \text{IF} & a_1 \sqsubseteq \mathbf{x} & \text{THEN} & y_1 \sim Dist(\hat{\Theta}_1^1) & \cdots & y_t \sim Dist(\hat{\Theta}_t^1) \\
 & & \vdots & & & & \\
\omega: & \text{ELSE IF} & a_\omega \sqsubseteq \mathbf{x} & \text{THEN} & y_1 \sim Dist(\hat{\Theta}_1^\omega) & \cdots & y_t \sim Dist(\hat{\Theta}_t^\omega) \\
\text{default:} & \text{ELSE} & & & y_1 \sim Dist(\hat{\Theta}_1^d) & \cdots & y_t \sim Dist(\hat{\Theta}_t^d)
\end{array}
$$

Figure 3.1: Generic rule list model M with ω rules and t (number of target variables) distributions per rule.

This may seem like a subtle difference, but for subgroup lists, it allows to find subgroups that always differentiate themselves from the dataset marginal distribution. In contrast, for predictive rule lists, it will enable finding predictive rules that maximize predictive performance. A theoretical proof of their difference, from an MDL perspective, is given in Chapter 3.7.

Nonetheless, all the data encodings developed in this chapter can be used for both predictive rule lists and subgroup lists. In the case of predictive rule lists, the data encodings were only empirically tested in the classification setting. In contrast, subgroup lists were tested for all the settings, i.e., univariate and multivariate nominal and numeric targets. To not burden the reader, we here present two simple examples of subgroup lists in Figures 3.2 and 3.3, which will be used throughout this chapter to exemplify the MDL encodings.

Structure of the chapter. This chapter is organized as follows. First, in Section 3.1 the MDL principle for supervised datasets is introduced. Next, in Section 3.2 the encoding of the model structure is shown. Then, in Section 3.3 the high-level encoding of the data given the model is presented. After that, the specific encodings of the data given a model for categorical and normal distributions are given in Sections 3.4 and 3.5, respectively. Then, in Section 3.6 a new subgroup set discovery measure is presented. Finally, in Chapter 3.7 the theoretical difference between rule lists and subgroup lists is studied through the MDL lens.

			$\mathbf{Pr}(\boldsymbol{animaltype} = \cdots \mid \boldsymbol{s})$ in %						
s	**description**	n_s	Mammal	Fish	Invert.	Bug	Reptile	Amph.	Bird
1	backbone = no	18	0	0	56	44	0	0	0
2	breathes = no	14	0	93	0	0	7	0	0
3	feathers = yes	20	0	0	0	0	0	0	100
4	milk = no	8	0	0	0	0	50	50	0
5	feathers = no	41	100	0	0	0	0	0	0
dataset distribution		0^*	41	13	10	8	5	4	2

Figure 3.2: *Zoo* dataset subgroup list obtained by RSD algorithm (presented in Chapter 5). *Zoo* contains one nominal target variable with 7 classes, 101 instances, and 15 binary and 1 numeric variables. n_s refers to the number of instances covered by subgroup 's' defined by 'description'. $\Pr(animaltype = * \mid s)$ denotes the estimated probability (in %) of each class label occurring within the subgroup. The bottom row shows the marginal probability distribution of the dataset. $*$ concerns instances not covered by any of the five subgroups. For illustrative purposes the probabilities displayed correspond to the empirical probabilities in the data, not to the probabilities as would be obtained using the appropriate estimator.

			price (K)	
s	**description** of automobile specifications	n_s	$\hat{\mu}$	$\hat{\sigma}$
1	weight = heavy & consumption-city $\leq$ 8 km/L	11	35	8
2	fuel-type = gas & consumption-city $\geq$ 13 km/L	45	7	1
3	weight = light & wheel-base = low	35	9	1
4	length = medium & 13 $\leq$ consumption-city $\leq$ 15 km/L	27	10	2
5	peak-rpm = medium	49	16	3
6	engine-size = medium	12	26	7
dataset overall distribution		18^*	13	8

Figure 3.3: *Automobile import 1985* subgroup list obtained with RSD algorithm (presented in Chapter 5). The dataset contains *price* as numeric target variable, 197 examples, and 17 variables. The dataset was modified, some variables removed and others discretized, for ease of presentation. n_s refers to the number of instances covered by subgroup 's' defined by 'description', $\hat{\mu}$ and $\hat{\sigma}$ its estimated mean and standard deviation for the target variable in thousands of dollars (K). $*$ concerns instances not covered by any of the five subgroups.

3.1 The Minimum Description Length (MDL) principle

As we are interested in finding compact yet good models that are statistically robust, we resort to the Minimum Description Length (MDL) [107, 48] principle. The problem of selecting a concrete model from a large space of possible models is a *point hypothesis selection* problem, for which we should use a two-part code [48].
In contrast to existing pattern-based modeling approaches (e.g., [120, 78]), we deal with a *supervised* setting in which the goal is to learn a mapping from instances to target variables. This implies that we are not looking for structure *within* instance data $\mathbf{X}$, but for structure in $\mathbf{X}$ that helps to *explain* (subgroup lists) or *predict* (predictive rule lists) $\mathbf{Y}$.
That is, to induce a mapping from instances to target variables, we should consider the instance data $\mathbf{X}$ to be given as 'input' to the model and *only encode the target variables* $\mathbf{Y}$. Clearly, this corresponds to the rule lists that we introduced in the previous chapter. Then, given the complete space of models $\mathcal{M}$, uniquely specified by all ordered sets of patterns over $\mathcal{X}$, the optimal model is the model $M \in \mathcal{M}$ that minimizes a two-part code [48], i.e.,

$$M^* = \arg\min_{M \in \mathcal{M}} L(D, M) = \arg\min_{M \in \mathcal{M}} \left[L(\mathbf{Y} \mid \mathbf{X}, M) + L(M) \right], \tag{3.2}$$

where $L(\mathbf{Y} \mid \mathbf{X}, M)$ is the encoded length, in bits[2], of target variables data $\mathbf{Y}$ given explanatory data $\mathbf{X}$ and model M, $L(M)$ is the encoded length, in bits, of the model, and $L(D, M)$ is the total encoded length and the sum of both terms. Note that this definition holds up for different models, such as rule list RL or subgroup list SL. Intuitively, the best model M^* is the model that results in the best trade-off between how well the model compresses the target data and the complexity of that model—thus minimizing redundancy and automatically selecting the best list size. This formulation is similar to that previously used for two-view association discovery [73].

3.2 Model encoding

The next step is to define the two length functions; we start with $L(M)$. Following the MDL principle [48], we need to ensure that: 1) all models in the model class, i.e., all rule lists for a given dataset, can be distinguished; and 2) larger code lengths are assigned to more complex models. To accomplish the former, we encode all elements of a model that can change, while for the latter, we resort to two different codes: when a larger value represents a larger complexity we use the universal code for integers

[2]To obtain code lengths in bits, all logarithms in this paper are to the base 2.

[108], denoted[3] $L_{\mathbb{N}}$; and when we have no prior knowledge but need to encode an element from a set, we choose the uniform code. Note that as predictive rule lists and subgroup lists have the same structure, their model can be defined in the same way, as given all the rules in M, the default rule subset is completely defined.
Specifically, the encoded length of a model M over variables in $\mathbf{X}$ is given by

$$L(M) = L_{\mathbb{N}}(|M|) + \sum_{a_i \in M} \left[L_{\mathbb{N}}(|a_i|) + \log \binom{m}{|a_i|} + \sum_{v \in a_i} L(v) \right], \tag{3.3}$$

where we first encode the number of antecedents $|M|$, which can symbolize predictive rules $|R|$ or subgroups $|S|$, using the universal code for integers, and then encode each rule description individually. For each description, first, the number $|a_i|$ of variables used is encoded, then the set of variables using a uniform code over the set of all possible combinations of $|a_i|$ from all explanatory variables, and finally the specific condition for a given variable. As we allow variables of two types, the latter is further specified by

$$L(v) = \begin{cases} \log |\mathcal{X}_v| & \text{if } v \text{ is nominal} \\ L_{\mathbb{N}|2}(n_{op}) + \log N(n_{op}, n_{cut}) & \text{if } v \text{ is numeric} \end{cases} \tag{3.4}$$

where the code for each variable type assigns code lengths proportional to the number of possible parts the variable's domain can partition the dataset. Note that this seems justified, as having more parts implies more potential spurious associations with the target that we would like to avoid. For `nominal` variables, this is given by the size of the domain, i.e., the number of categories in a nominal variable. For `numeric` variables, it equals the number of operators used $n_{op}|$[4] plus the possible number of outcomes $N(n_{op}, n_{cut})$ given the operators and n_{cut} cut points. The number of operators for numeric variables can be one or two, as there can be conditions with one (e.g., $x \leq 2$) or two operators (e.g., $1 \leq x \leq 2$), which is a function of the number of possible subsets generated by n_{cut} cut points. Note that we here assume that equal frequency binning is used, which means that knowing X and n_{cut} is sufficient to determine the cut points.

Example 5 (continuation): Let us assume that the subgroup list of the *Automobile* example of Figure 3.3 is composed of only the first subgroup. In that case the rule list only has one subgroup with description: {weight = heavy & consumption-city ≤ 8

[3] $L_{\mathbb{N}}(i) = \log k_0 + \log^* i$, where $\log^* i = \log i + \log \log i + \ldots$ and $k_0 \approx 2.865064$.

[4] Note that we use $L_{\mathbb{N}|2}$, which is how we denote the universal code for integers with codes restricted to $n = 1$ or 2. This can be obtained by applying the maximum entropy principle to $L_{\mathbb{N}}$ when it is known that it cannot take values of $n > 2$.

km/L }. Taking into account that the dataset has 17 variables, $|\mathcal{X}_{weight}| = 3$ and only 3 cut points were used for numeric attributes, the model length is given by:

$$\begin{aligned} L(M) &= L_{\mathbb{N}}(1) + L_{\mathbb{N}}(2) + \log \binom{17}{2} + \log |\mathcal{X}_{weight}| + \left[L_{\mathbb{N}|2}(1) + \log 2n_{cut} \right] \\ &= 1.52 + 2.52 + 7.09 + 1.59 + 0.77 + 2.59 \\ &= 16.08 \text{ bits} \end{aligned}$$

It is important to note that the length of the model can (and should) be a real number, as we are only concerned with the idea of compression, not with materialising and transmitting the actually encoded data [48].

3.3 Data encoding

The remaining length function is that of the target data given the explanatory data and model, $L(\mathbf{Y} \mid \mathbf{X}, M)$. In this section, we show how to encode the target data $\mathbf{Y}$ by dividing it into smaller subsets that can be encoded individually and then summed together, and why there are different types of data encoding for each of the subsets. The specifics of encoding nominal and numeric targets are described in Sections 3.4 and 3.5, respectively.

Cover of a rule in a rule list. Let us recall from Chapter 2.4 that for any given rule list of the form of Figure 3.1, *any individual instance* $(\mathbf{x}, \mathbf{y})$ *can only be 'covered' by one rule or subgroup*. That is, the cover of a description in a list a_i, denoted D^i, depends on the order of the list and is given by the instances where its description occurs minus those instances covered by previous descriptions, i.e., $a_j, \forall_{j<i}$.
In case an instance $(\mathbf{x}, \mathbf{y})$ is not covered by any pattern $a \in M$ *then it is 'covered' by the default rule*. The number of instances covered by the default rule D^d are the ones not covered by any description (hence the name default rule). The instances covered by a description, also called *usage*, are denoted by $n_i = |D^i|$, and those covered by the default rule, $n_d = |D^d|$
As every description defines an individual subset, one can estimate the parameters of its target variable distributions using the maximum likelihood estimator described in Section 2.3.2.
Note that this shows us that a rule or subgroup is fully defined by its description a_i in a dataset D, and we will interchangeably refer to rules by their descriptions and to its elements (statistics, parameters, distributions, etc.) by its index i when obvious from context.

As the default rule is the only difference between a rule list and a subgroup list, it is also the difference in their encoding. As a rule list induces a *partition of the data*, the total length of the encoded data can be given by the sum of its *non-overlapping parts*. For a **predictive rule list**, the data encoding is given by:

$$L(\mathbf{Y} \mid \mathbf{X}, M) = L(\mathbf{Y}^d) + \sum_{r_i \in R} L(\mathbf{Y}^i), \tag{3.5}$$

while for a **subgroup list** it is given by:

$$L(\mathbf{Y} \mid \mathbf{X}, M) = L(\mathbf{Y}^d \mid \mathbf{\Theta}^d) + \sum_{s_i \in S} L(\mathbf{Y}^i), \tag{3.6}$$

where $\mathbf{\Theta}^d$ is the vector of parameters for each variable $\Theta_1^d, \ldots, \Theta_t^d$ for the marginal distribution of the target variables. Observe that we dropped $\mathbf{X}^a$ as these are not necessary to encode $\mathbf{Y}^a$ but only to generate the partition of the data, and also dropped the parameters $\mathbf{\Theta}^i$ of the rules and default predictive rule as we do not know what are their parameters until we see the data. This last part will be clarified at the end of this section, where we describe how to encode subsets without knowing their parameters. As can be seen, the difference between the predictive rule list and subgroup list is that the default rule is either encoded as a regular rule, or using the dataset distribution. This amounts to a difference in optimality between predictive rule lists and subgroup lists, which emphasizes the discovery of different types of descriptions for each model class.
As a side-note, note that Eq. (3.5) concerns the encoding of any supervised partition of the data, which allows to directly quantify the quality of any tree learning method—each such tree induces a partition of the data.

Encoding data of t (assumed) independent target variables. As each target variable is assumed independent from each other, the encoding of target data is given by the sum of their individual encodings:

$$L(\mathbf{Y} \mid \mathbf{X}, M) = -\log\left(\prod_{j=1}^{t} \Pr(Y_j \mid \mathbf{X}, M)\right) = \sum_{j=1}^{t} L(Y_j \mid \mathbf{X}, M). \tag{3.7}$$

Joining (3.5) and (3.7), one obtains for **predictive rule lists**:

$$L(\mathbf{Y} \mid \mathbf{X}, M) = \sum_{j=1}^{t} \left(L(Y_j^d) + \sum_{s_i \in S} L(Y_j^i)\right) \tag{3.8}$$

and joining (3.6) and (3.7), one obtains for **subgroup lists**:

$$L(\mathbf{Y} \mid \mathbf{X}, M) = \sum_{j=1}^{t} \left(L(Y_j^d \mid \Theta_j^d) + \sum_{s_i \in S} L(Y_j^i)\right) \tag{3.9}$$

3.3.1 Two types of data encoding

Data encoding can be separated into two different categories: 1) with *known parameters*; and 2) with *unknown parameters*. In our case, *known parameters* correspond to the default rule of a subgroup list, while *unknown parameters* correspond to the predictive rules, subgroups, and default rule of a predictive rule list.

1**) Known parameters**: when the parameters of a distribution are *known*, one can encode the data points directly using the probability for those points given by the distribution with the known parameters. Thus, the encoding of points Y_j^i (j^{th} variable and i^{th} subgroup) is equal to the negative logarithm of their probability given by known parameters $\hat{\Theta}_j^i$:

$$L(Y_j^i \mid \hat{\Theta}_j^i) = \sum_{y \in Y_j^i} -\log \Pr(y \mid \hat{\Theta}_j^i), \tag{3.10}$$

which is just the minus log-likelihood of parameter $\hat{\Theta}_j^i$ given observed data Y_j^i. This type of code is used in the case of the default rule of a subgroup list, as the parameters $\hat{\Theta}_j^d$ are equal to the marginal distribution of variable Y_j and are constant for each dataset. Note that this is the *key difference between a subgroup list and a predictive rule list*: the last rule of a subgroup list is fixed to the marginal distribution, while in the predictive rule list its parameters are unknown and depend on the subset D^d.

2**) Unknown parameters**: when the parameters are *unknown* we need to encode both the parameter values and the data points. We have two possibilities: 1) crude MDL, i.e., encoding the probabilities using a suboptimal probability distribution and then applying the Shannon-Fano code, i.e., the logarithm of the empirical probability [114]; or 2) employ an optimal encoding of both parameters of the distribution and data points together [48]. In this work, we employ optimal encoding of parameters, as it guarantees optimality in the sense that the encoding is the best possible in the worst-case scenario, i.e., in case the sample of the data is not representative of the population. Three types of optimal encodings exist, which are, in increasing order of optimality guarantees: 1) *prequential plug-in*; 2) *Bayesian*; 3) *Normalized Maximum Likelihood (NML)*. While the first two are asymptotically optimal, the NML encoding is optimal for fixed sample sizes.
Depending on the target type, we employ the best encoding possible while being computationally feasible, i.e., we require adequate run-time for our algorithm. For nominal targets, we present a prequential plug-in and an NML encoding for both the probabilities of each class and the data points in Section 3.4, where the second is a theoretical improvement over the first. We resort to a Bayesian encoding for numeric targets as the NML code is not computationally feasible for that case.

3.4 Data encoding: nominal target variables

When the data have one or more nominal targets, the target distributions of the probabilistic rules (2.7) are categorical distributions $Cat(\Theta)$, each with a set of parameters $\Theta = \{p_1, \cdots, p_k\}$ representing the k classes:

$$\Pr(y = c \mid p_1, \cdots, p_k) = p_c, \text{ subject to } \sum_{c=1}^{k} p_c = 1. \tag{3.11}$$

This implies a probabilistic rule of the form:

$$a \mapsto y_1 \sim Cat(p_1, \cdots, p_k), \cdots, y_t \sim Cat(p_{1'}, \cdots, p_{k'}),$$

where k and k' are the number of classes Y_1 and Y_t, respectively. To simplify the introduction of concepts we will assume we only have one target variable in $\mathbf{Y}$, and then generalize the results to multiple variables at the end. Also in line with this simplification, we will only refer to association rules, and then, specialize in the end for both predictive rule lists and subgroup lists. Thus, throughout this section $\mathbf{Y}$ becomes Y, and the parameters of each rule r_i become $\hat{\Theta}^i = \{p_{1|i}, \cdots, p_{k|i}\}$ as there is only one variable with k classes, where $p_{1|i}$ is the probability of class 1 for subgroup i, i.e., $\Pr(c = 1 \mid a_i)$. The general form of a rule list with one nominal target takes the form of Figure 3.4.

$$\begin{array}{rllll}
r_1\text{:} & \text{IF} & a_1 \sqsubseteq \mathbf{x} & \text{THEN} & y \sim Cat(\hat{p}_{1|1}, \cdots, \hat{p}_{k|1}) \\
 & & \vdots & & \\
r_\omega\text{:} & \text{ELSE IF} & a_\omega \sqsubseteq \mathbf{x} & \text{THEN} & y \sim Cat(\hat{p}_{1|\omega}, \cdots, \hat{p}_{k|\omega}) \\
\text{default:} & \text{ELSE} & & & y \sim Cat(\hat{p}_{1|d}, \cdots, \hat{p}_{k|d})
\end{array}$$

Figure 3.4: Generic rule list model M with ω rules $\{r_1, ..., r_\omega\}$ and a single nominal target Y with k categories.

In the following sections, we will derive the data encoding with categorical distributions. First, in Section 3.4.1, it is shown how to encode a categorical distribution when its parameters are known, which is the case for the default rule of a subgroup list. After that, in Section 3.4.2 it is shown how to encode a categorical distribution when the parameters of the distribution are unknown. Then, in Section 3.4.3 the equivalence between MDL-based subgroup lists with only one subgroup and standard (top-1) subgroup discovery with WKL as a quality measure is proven. Finally, in Section 3.4.4, we show the data encoding of subgroup lists is equivalent to a Bayesian test. Note that for the next section we will also use the maximum likelihood expressions of Section 2.3.2.

3.4.1 Encoding categorical distributions with *known* parameters

To encode target values with *known parameters*—as is the case for the default rule of a subgroup list—we can directly use Eq. (3.10) with given parameter estimates $\hat{\Theta}^d = \hat{p}_{1|d}, \cdots, \hat{p}_{k|d}$ (marginal distribution over the whole dataset):

$$L(Y^d \mid \hat{p}_{1|d}, \cdots, \hat{p}_{k|d}) = \sum_{c \in \mathcal{Y}} -n_{c|d} \log \hat{p}_{c|d} = -\ell(\hat{\Theta}^d \mid Y^d), \tag{3.12}$$

where $\ell(\hat{\Theta}^d \mid Y^d)$ is the log-likelihood of the parameter set $\hat{\Theta}^d$, and $n_{c|d}$ denotes the number of points associated with each class c covered by default rule Y^d.
Note that we exemplified this code using the dataset marginal distribution parameters as these are the only *known* parameters used throughout this thesis, however, this encoding can be used with any known parameters.

3.4.2 Encoding categorical distributions with *unknown* parameters

To encode target values for which the parameters are *unknown*—as is the case for each predictive rule, subgroup, and predictive default rule—we need to encode parameters and data together. For that, we have developed two types of codes: 1) the *prequential plug-in code* that is asymptotically optimal; and 2) the *Normalized Maximum Likelihood (NML)* code that is "optimal in the sense that it achieves the minimax optimal codelength regret" [48, Part II]. The prequential plug-in code was developed earlier [96], as it is easier to use and compute. Nonetheless, the NML code enjoys better theoretical properties, and thus should be preferred when possible.

Prequential plug-in encoding. The main idea of the *prequential plug-in* code is to treat each subset of labels Y^i as sequential data and then predict each label as it arrives, starting with no knowledge about their distribution and updating it each time one receives a label. To achieve that, it requires the use of a smoothed version of the ML estimator, as before receiving any point we already need to have a probability distribution

$$\Pr_{\text{plug-in}}(y^u = c \mid Y^{|u-1}) := \frac{|\{y \in Y^{|u-1} \mid y = c\}| + \epsilon}{\sum_{c' \in \mathcal{Y}} |\{y \in Y^{|u-1} \mid y = c'\}| + \epsilon}, \tag{3.13}$$

where $Y^{|u-1}$ represents the ordered sequence of $u-1$ class labels, ϵ the pseudocount which allows us to have probabilities before seeing any label.
Intuitively, this means that one starts with a pseudocount ϵ for each possible element, constructs a code using these pseudocounts, starts encoding/sending/decoding messages one by one, and then *updates the count of each element after sending/receiving*

each individual message. The *prequential plug-in code* is asymptotically optimal even without any prior knowledge of the probabilities [48].
Taking into account that the rule list creates a partition of the data, and applying Eq. (3.13) to each class label in part, we obtain for each part[5] Y^i:

$$\begin{aligned} L_{\text{plug-in}}(Y^i) &= -\log\left(\prod_{u=1}^{n_i} \Pr_{\text{plug-in}}(y^u \mid Y^{i|u-1})\right) \\ &= -\log\left(\frac{\prod_{c=1}^{k}\prod_{u=0}^{n_{c|i}-1}(u+\epsilon)}{\prod_{j=0}^{n_i-1}(u+k\epsilon)}\right) \\ &= -\log\left(\frac{\prod_{c=1}^{k}(n_{c|i}-1+\epsilon)!/(\epsilon-1)!}{(n_i-1+k\epsilon)!/(k\epsilon-1)!}\right) \\ &= -\log\left(\frac{\prod_{c=1}^{k}\Gamma(n_{c|i}+\epsilon)/\Gamma(\epsilon)}{\Gamma(n_i+k\epsilon)/\Gamma(k\epsilon)}\right), \end{aligned} \tag{3.14}$$

where $Y^{i|u}$ is a sequence of class labels of length u in part D^i, and $n_i = |D^i|$ and $n_{c|i} = |D^{c|i}|$. Further, Γ is the gamma function, an extension of the factorial to real and complex numbers that is given by $\Gamma(u) = (u-1)!$. The most common values for ϵ, which takes the role of a prior in the Bayesian literature [125], are the Jeffrey's prior of 0.5 or the uniform prior of 1. For simplicity in our experiments, the value of $\epsilon = 1$ was used as it allows us to obtain natural factorials instead of gamma functions. It is interesting to note two things: 1) we started with a sequential idea, but the final encoding of Eq. (3.14) is independent of the order in which the data is processed; and 2) for the case of categorical and multinomial distributions the prequential plug-in code is equivalent to a Bayesian code with a Dirichelet prior [48, Chapter 9]

NML encoding. The expression of the NML code can be daunting, but its intuition is very clear [65], i.e., the NML code is equivalent to first encoding all maximum likelihood estimates of sequences Z of n_i points based on their likelihoods, and then encoding data Y^i with its maximum likelihood estimate $\hat{\Theta}^i$ as in Eq. (3.12). Formally, the NML code length of the subset Y^i is given by[6]:

$$\begin{aligned} L_{NML}(Y^i) &= -\log \frac{\prod_{y\in Y^i}\Pr(y \mid \hat{\Theta}^i)}{\sum_{Z\in\mathcal{Y}^{n_i}}\prod_{z\in Z}\Pr(z \mid \hat{\Theta}^Z)} \\ &= \sum_{c\in\mathcal{Y}} -n_{c|i}\log\hat{p}_{c|i} + \log\sum_{Z\in\mathcal{Y}^{n_i}}\prod_{z\in Z}\Pr(z \mid \hat{\Theta}^Z) \\ &= -\ell(\hat{\Theta}^i \mid Y^i) + \mathcal{C}(n_i, k) \end{aligned} \tag{3.15}$$

[5]For full details and intuition on the derivations of the prequential plug-in code check Appendix B.
[6]For details on the derivation of Eq. (3.15), please see Appendix C.

where $\mathcal{Y}^{n_i}$ is the space of all possible sequences of n_i points with cardinality $k = |\mathcal{Y}|$ (possible values per point), $\hat{\Theta}^Z$ is the maximum likelihood estimate over Z, $\mathcal{C}(n_i, k)$ is the complexity—as it is called in MDL literature[48]—of the multinomial distribution over n_i points and k categories. Note that this term can be efficiently computed in sub-linear time $\mathcal{O}(\sqrt{dn_i} + k)$ if approximated by a finite floating-point precision of d digits [92].

Predictive rule list encoding. The total data encoding of a predictive rule list, using the NML encoding, is obtained by inserting (3.12) and (3.15) in (3.8):

$$L(\mathbf{Y} \mid \mathbf{X}, M) = \sum_{j=1}^{t} \left(L_{NML}(Y_j^d) + \sum_{\rho_i \in R} L_{NML}(Y_j^i) \right), \tag{3.16}$$

where for the total data encoding using the prequential plug-in code, substitute $L_{NML}(\cdots)$ by $L_{\text{plug-in}}(\cdots)$ of Eq. (3.14).

Subgroup list encoding. The total data encoding of a subgroup list, using the NML encoding, is obtained by inserting (3.12) and (3.15) in (3.9):

$$L(\mathbf{Y} \mid \mathbf{X}, M) = \sum_{j=1}^{t} \left(L(Y_j^d \mid \hat{\mathbf{\Theta}}^d) + \sum_{s_i \in S} L_{NML}(Y_j^i) \right), \tag{3.17}$$

where $\mathbf{\Theta}^d$ is the dataset marginal parameters, and for the total data encoding using the prequential plug-in code, substitute $L_{NML}(\cdots)$ by $L_{\text{plug-in}}(\cdots)$ of Eq. (3.14).

Example 6 (continuation): Let us revisit the *Zoo* subgroup list example of Figure 3.2 and compute the length encoding of the first subgroup subset Y^1 using the NML encoding. To compute it we just need to get the probabilities associated with each category ($\{0; 0; 0.56; 0.44; 0; 0; 0\}$), the number of samples covered by each of them ($\{0; 0; 10; 8; 0; 0; 0\}$), and the total number of categories $k = |\mathcal{Y}| = 7$. Given these, the length of encoding of the data Y^1 is given by:

$$\begin{aligned} L_{NML}(Y^1) &= (-10 \log 0.56 - 8 \log 0.44) + \mathcal{C}(18, 7) \\ &= 17.84 + 10.42 \\ &= 28.26 \text{ bits.} \end{aligned}$$

3.4.3 Relationship of MDL-optimal subgroup lists to WKL-based SD

We now investigate the relationship between finding an MDL-optimal subgroup list and WKL-based top-k subgroup discovery. Remember that WKL is the weighted Kulback-

Leibler (WKL) divergence, an existing subgroup discovery measure [72] that can be seen as an information-theoretic instance of the general form of a subgroup discovery measure as given in Eq. (2.20); we described it in more detail in Section 2.6.2.
Assume that we have a single target variable (Y instead of $\mathbf{Y}$) and a subgroup list consisting of just one subgroup s with description a (and the default rule). Next, let us turn the MDL minimization problem into a maximization problem by multiplying Eq. (3.2) by minus one and adding a constant (for each dataset) $L(Y \mid \mathbf{\Theta}^d)$ to obtain:

$$s^* = \arg\max_{s \in \mathcal{M}} \Big[L(Y^d \mid \mathbf{\Theta}^d) - L(Y \mid \mathbf{X}, M) - L(M) \Big] .$$

In the case of a subgroup list with *one subgroup* and one target, the data encoding of Eq. (3.17) can be substituted by $L(Y \mid \mathbf{X}, M) = L(Y^d \mid \mathbf{\Theta}^d) + L_{NML}(Y^a)$. Also, note that Y^d is given by all the points not covered by the subgroup description a, i.e., $Y^{\neg a}$. Thus, we can further develop the maximization problem to:

$$\begin{aligned}
L(Y \mid \hat{\Theta}^d) - L(Y \mid \mathbf{X}, M) - L(M) = \\
&= L(Y^a \mid \hat{\Theta}^d) + \cancel{L(Y^{\neg a} \mid \hat{\Theta}^d)} - L_{NML}(Y^a) - \cancel{L(Y^{\neg a} \mid \hat{\Theta}^d)} - L(M) \\
&= \sum_{y \in Y^s} \log \frac{\hat{p}_{y|a}}{\hat{p}_{y|d}} - \mathcal{C}(n_a, k) - L(M) \\
&= n_a \sum_{c \in \mathcal{Y}} \hat{p}_{c|a} \log \left(\frac{\hat{p}_{c|a}}{\hat{p}_{c|d}} \right) - \mathcal{C}(n_a, k) - L(M) \\
&= n_a KL(\hat{\Theta}^a; \hat{\Theta}^d) - \mathcal{C}(n_a, k) - L(M),
\end{aligned} \tag{3.18}$$

where $n_a KL(\hat{\Theta}^a; \hat{\Theta}^d)$ is the Weighted Kulback-Leibler divergence from $\hat{\Theta}^a$ to $\hat{\Theta}^d$. *This result shows that finding the MDL-optimal subgroup is equivalent to finding the subgroup that maximizes WKL, plus two extra terms: one that defines the complexity of the distribution $\mathcal{C}(n_a, k)$, and another that defines the complexity of the subgroup $L(M)$.*
When we consider subgroup lists having more than one subgroup, Eq. (3.18) simply expands to:

$$\begin{aligned}
L(Y \mid \hat{\Theta}^d) - L(Y \mid \mathbf{X}, M) - L(M) &= \sum_{a_i \in S} n_i KL(\hat{\Theta}^i; \hat{\Theta}^d) - \sum_{a_i \in S} \mathcal{C}(n_i, k) - L(M) \\
&= \text{SWKL}(S) - \sum_{a_i \in S} \mathcal{C}(n_i, k) - L(M),
\end{aligned}$$

where SWKL(S) is the Sum of Weighted Kulback-Leibler divergences of subgroup set S, a measure for subgroup set quality that we propose later in Section 3.6, and the other terms penalize the complexity of the subgroup list. The fact that the MDL-based objective for the optimal subgroup list can be formulated as subgroup set quality

minus two terms for model complexity demonstrates that our formalization naturally aims for subgroup lists of high quality while penalizing complexity.

3.4.4 Relationship of MDL-optimal subgroup lists to Bayesian testing

We will now show how our MDL criterion is related to Bayesian testing. The Bayesian alternative to statistical testing is the Bayesian factor, denoted here by K [58, 61]. The Bayesian factor compares two models (hypotheses) through the division of the likelihood of the data given each model $\Pr(D \mid M_1)/\Pr(D \mid M_2)$, where the more likely model dominates. Notice that the form that we arrived at in the term $n_a KL(\hat{\Theta}^a; \hat{\Theta}^d) - \mathcal{C}(n_a, k) - L(M)$ of Eq. (3.18) (for a list consisting of one subgroup) is very similar to the logarithm of a Bayes factor, and indeed it can be decomposed into:

$$\begin{aligned} L(Y \mid \hat{\Theta}^d) - L(Y \mid \mathbf{X}, M) - L(M) &= \log\left(\frac{\Pr(Y \mid \mathbf{X}, M)}{\Pr(Y \mid \hat{\Theta}^d)}\right) L(M) \\ &= \log K + L(M), \end{aligned}$$

where we use the Shannon-Fano code [114] to transform code length in bits $L(\cdots)$ to probabilities $\Pr(\cdots)$. In practice, taking into account $L(M)$ (or $\Pr(M)$) is equivalent to using the posterior distributions instead of just the Bayes factor, and in our case amounts to a penalty for multiple hypothesis testing. This tells us that when finding the first subgroup we are indeed maximizing an MDL version of a Bayesian factor, and thus, doing an equivalent Bayesian proportions test (with a binary target) or a multinomial test (with a nominal target). When we consider the problem of finding a subgroup beyond the first, it is straightforward to observe that we are testing each subgroup in S against the marginal distribution of the dataset.

3.5 Data encoding: numeric target variables

When we have one or more numeric target variables, the consequents of probabilistic rules as in Eq. (2.7) are now normal distributions $\mathcal{N}(\Theta)$ with parameters $\Theta = \{\mu, \sigma\}$, and take the following form:

$$\Pr(y \mid \mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y-\mu)^2}{2\sigma^2}\right),$$

where we use $\Pr(y \mid \mu, \sigma)$ to denote the probability density function (pdf), which is a slight abuse of notation that we admit to unify the whole work.

This translates to a probabilistic rule of the form:

$$a \mapsto y_1 \sim \mathcal{N}(\hat{\mu}_{a1}, \hat{\sigma}_{a1}), \cdots, y_t \sim \mathcal{N}(\hat{\mu}_{at}, \hat{\sigma}_{at}) \tag{3.19}$$

To simplify the introduction of concepts we will again assume we only have one target variable in $\mathbf{Y}$, and then generalize the results to multiple variables at the end. Also in line with this simplification, we will only refer to association rules, and then, specialize in the end for both predictive rule lists and subgroup lists. Thus, throughout this section $\mathbf{Y}$ becomes Y, and the parameters of each rule r_i become $\Theta^i = \{\mu_i, \sigma_i\}$ as there is only one variable. The general form of a rule list with normal target distribution is given in Figure 3.5.

$$\begin{array}{rlll} r_1\text{:} & \text{IF} & a_1 \sqsubseteq \mathbf{x} \quad \text{THEN} & y \sim \mathcal{N}(\hat{\mu}_1, \hat{\sigma}_1) \\ & & \vdots & \\ r_\omega\text{:} & \text{ELSE IF} & a_\omega \sqsubseteq \mathbf{x} \quad \text{THEN} & y \sim \mathcal{N}(\hat{\mu}_\omega, \hat{\sigma}_\omega) \\ \text{dataset:} & \text{ELSE} & & y \sim \mathcal{N}(\hat{\mu}_d, \hat{\sigma}_d) \end{array}$$

Figure 3.5: Generic rule list model M with ω rules $\{r_1, ..., r_\omega\}$ and a single numeric target Y.

In the following subsections, we will derive the data encoding with normal distributions. First, in Section 3.5.1 we show how to encode a normal distribution when its parameters μ and σ are known, such as is the case for the default rule of a subgroup list. After that, in Section 3.5.2 we show how to encode a normal distribution using an uninformative prior when the parameters of the distribution are unknown. Then, in Section 3.5.3 the equivalence between MDL-based subgroup lists with only one subgroup and standard (top-1) subgroup discovery with WKL as a quality measure is proven. Finally, in Section 3.5.4, we show the data encoding and corresponding criterion are equivalent to a Bayesian test. Note that for the next section we will also use the maximum likelihood expressions of Section 2.3.2.

3.5.1 Encoding normal distributions with *known* parameters

To encode target values with *known parameters*—as is the case for the default rule of a subgroup list—we can directly use Eq. (3.10) with given parameter estimates

$\hat{\Theta}_d = \{\hat{\mu}_d, \hat{\sigma}_d\}$ (marginal distribution over the whole dataset):

$$\begin{aligned} L(Y^d \mid \hat{\mu}_d, \hat{\sigma}_d) &= -\log\left[\prod_{y\in Y^d} \frac{1}{\sqrt{2\pi\hat{\sigma}_d^2}} \exp\left(\frac{(y-\hat{\mu}_d)^2}{2\hat{\sigma}_d^2}\right)\right] \\ &= \frac{n_d}{2}\log 2\pi + \frac{n_d}{2}\log\hat{\sigma}_d^2 + \left(\frac{1}{2\hat{\sigma}_d^2}\sum_{y\in Y^d}(y-\hat{\mu}_d)^2\right)\log e \\ &= -\ell(\hat{\Theta}^d \mid Y^d), \end{aligned} \tag{3.20}$$

where $\ell(\hat{\Theta}^d \mid Y^d)$ is the log-likelihood of the parameter set $\hat{\Theta}^d$. The first two terms are normalization terms of a normal distribution, while the last term represents the Residual Sum of Squares (RSS) normalized by the variance of the data. Note that when $Y_d = Y$, i.e., the whole dataset target, RSS is equal to $n_d\sigma_d$, and the last term reduces to $n_d/2\log e$.
Note that we exemplified this code using the dataset marginal distribution parameters as these are the only *known* parameters used throughout this thesis, however, this encoding can be used with any known parameters.

3.5.2 Encoding normal distributions with *unknown* parameters

In contrast to the previous case, here *we do not know a priori the statistics defining the probability distribution corresponding to the rule*, i.e., $\hat{\mu}$ and $\hat{\sigma}$ are not given by the model, and thus both need to be encoded. For this, we resort to the Bayesian encoding of a normal distribution with mean μ and standard deviation σ unknown, which was shown to be asymptotically optimal [48]. The optimal code length is given by the negative logarithm of a probability, and the optimal Bayesian probability for Y^i is given by

$$\begin{aligned} &L_{Bayes}(Y^i) = \\ &-\log\int_{-\infty}^{+\infty}\int_{0}^{+\infty}(2\pi\sigma)^{-\frac{n_i}{2}}\exp\left(-\frac{1}{2\sigma^2}\sum_{y\in Y^i}(y-\mu)^2\right)w(\mu,\sigma)\,\mathrm{d}\mu\,\mathrm{d}\sigma, \end{aligned} \tag{3.21}$$

where $w(\mu, \sigma)$ is the prior on the parameters, which needs to be chosen.

Choosing the prior. The MDL principle requires the encoding to be as unbiased as possible for any values of the parameters, which leads to the use of uninformative priors. The most uninformative prior is Jeffrey's prior, which is $1/\sigma^2$ and therefore constant for any value of μ and σ, but unfortunately its integral is undefined, i.e., $\int\int \sigma^{-2}\,\mathrm{d}\sigma\,\mathrm{d}\mu = \infty$. Thus, we need to make the integral finite, which we will do next.

It should be noted that when using normal distributions with Bayes factors—Bayesian equivalent to traditional statistical testing—the authors tend to also add a normal prior on the effect size, as e.g., $\delta = \mu/\sigma \sim \mathcal{N}(0, \tau)$ [58, 44, 110]. Nonetheless, this prior gives a higher probability to values of μ closer to zero, which is a bias that we do not want to impose. Thus we only use Jeffrey's prior, which converges[7] to the Bayes Information Criterion (BIC) for large n.

Now, given the our prior $w(\mu, \sigma) = \frac{1}{\sigma^2\sqrt{2\pi}}$—where $\sqrt{2\pi}$ was added for normalization reasons—the remaining question is how we can make the integral finite. The most common solution, which we also employ, is to use u data points from Y^i, denoted $Y^{i|u}$, to create a proper conditional prior $w(\mu, \sigma \mid Y^{i|u})$. As there are only two unknown parameters, we only need two points hence $u = 2$ [48]; for more on the interpretation of such "priors conditional on initial data points", see [47]. Consequently, we first encode $Y^{i|2}$ with a non-optimal code that is readily available—i.e., the dataset distribution of Eq. (3.20)—and then use the Bayesian rule to derive the total encoded length of Y^i as

$$\begin{aligned} L_{Bayes2.0}(Y^i) &= -\log \frac{P_{Bayes}(Y^i)}{P_{Bayes}(Y^{i|2})} P(Y^{i|2} \mid \mu_d, \sigma_d) \\ &= L_{Bayes}(Y^i) + L_{cost}(Y^{i|2}), \end{aligned} \tag{3.22}$$

where $L_{cost}(Y^{i|2}) = L(Y^{i|2} \mid \mu_d, \sigma_d) - L_{Bayes}(Y^{i|2})$ is the extra cost incurred by encoding two points non-optimally. After some re-writing[8] we obtain the encoded length of the y values covered by a subgroup Y^i as

$$\begin{aligned} &L_{Bayes2.0}(Y^i) = L_{Bayes}(Y^i) + L_{cost}(Y^{i|2}) \\ &= 1 + \frac{n_i}{2}\log\pi - \log\Gamma\left(\frac{n_i}{2}\right) + \frac{1}{2}\log(n_i) + \frac{n_i}{2}\log n_i\hat{\sigma}_i^2 + L_{cost}(Y^{i|2}), \end{aligned} \tag{3.23}$$

where Γ is the Gamma function that extends the factorial to the real numbers ($\Gamma(n) = (n-1)!$ for integer n) and $\hat{\mu}_i$ and $\hat{\sigma}_i$ are the statistics of Eqs. (2.10) and (2.11), respectively. Note that for $Y^{i|2}$ any two unequal values (otherwise $\hat{\sigma}_2 = 0$ and $L_{Bayes}(Y^{i|2}) = \infty$) can be chosen from Y^a, thus we choose them such that they minimize $L_{cost}(Y^{i|2})$.

Predictive rule list encoding. The total data encoding of a predictive rule list, is obtained by inserting Eq. (3.20) and (3.23) in (3.8):

$$L(\mathbf{Y} \mid \mathbf{X}, M) = \sum_{j=1}^{t} \left(L(Y_j^d \mid \mathbf{\Theta}^d) + \sum_{s_i \in S} L_{Bayes2.0}(Y_j^i) \right).$$

[7] See proof in Appendix E.

[8] The full derivation of the Bayesian encoding and an in-depth explanation are given in Appendix D.

Subgroup list encoding. The total data encoding of a subgroup list, is obtained by inserting Eq. (3.20) and (3.23) in (3.9):

$$L(\mathbf{Y} \mid \mathbf{X}, M) = \sum_{j=1}^{t} \left(L(Y_j^d \mid \mathbf{\Theta}^d) + \sum_{s_i \in S} L_{Bayes2.0}(Y_j^i) \right),$$

where $\mathbf{\Theta}^d$ is the dataset marginal parameters.

Example 7 (continuation): We revisit the *Automobile* subgroup list of Figure 3.3 and find the length of the $Bayes2.0$ encoding (Eq. (3.23)) of the first subgroup. To compute it we need to get the statistics of the subgroup ($\hat{\Theta}^1 = \{\hat{\mu}_1 = 35; \hat{\sigma}_1 = 8\}$), the number of samples it covers ($n_1 = 11$), the dataset statistics ($\hat{\Theta}^d = \{\hat{\mu}_d = 13; \hat{\sigma}_d = 8\}$), and the two points closest to the dataset mean $Y^{1|2} = \{14; 31\}$ that make the encoding proper (and which are not available in the example information). Assuming that $L_{cost}(Y^{i|2}) = 0.69$ bits for simplicity, the length of the encoding of Y^1 is given by:

$$\begin{aligned} L_{Bayes2.0}(Y^1) =& 1 + \frac{11}{2}\log\pi - \log\Gamma\left(\frac{11}{2}\right) + \frac{1}{2}\log(11+1) + \frac{11}{2}\log 11 \cdot 8^2 \\ &+ L_{cost}(Y^{i|2}) \\ =& 58.06 + 0.69 \\ =& 58.75 \text{ bits.} \end{aligned}$$

3.5.3 Relationship of MDL-optimal subgroup lists to WKL-based SD

As in Section 3.4 we next investigate the relationship between finding an MDL-optimal subgroup list and WKL-based top-1 subgroup discovery, but now for the numeric case.

First, we show that Eq. (3.23)—with mean and variance unknown—converges, for large n, to Eq. (3.20)—with mean and variance known—plus an additional term. Using the Stirling approximation of $\Gamma(n+1) \sim \sqrt{2\pi n}\left(\frac{n}{e}\right)^n$ leads to[9]

$$L_{Bayes2.0}(Y^a) \sim \frac{n_a}{2}\log 2\pi + \frac{n_a}{2}\log\hat{\sigma}_a^2 + \frac{n_a}{2}\log e + \log\frac{n_a}{e}, \tag{3.24}$$

where $\log\frac{n}{e}$ is equal to the penalty term of BIC and similar to the usual MDL complexity of a distribution [48].

Now, we can show that minimizing our MDL criterion is equivalent to maximizing a subgroup discovery quality function of the form of Eq (2.20). Focusing on the case

[9]The complete derivation can be found in the Appendix E

where $M = \{s\}$ contains only one subgroup with description a and statistics $\hat{\Theta}^a = \{\hat{\mu}_a, \hat{\sigma}_a\}$, we start with $L(Y \mid X, M)$ (Eq. (3.2)), multiply it by minus one to make it a maximization problem, and add a constant $L(Y \mid \hat{\mu}_d, \hat{\sigma}_d)$, i.e., the encoded size of the whole target Y using the overall distribution dataset. We then get

$$s^* = \arg\max_{s \in \mathcal{M}} \Big[L(Y^d \mid \mathbf{\Theta}^d) - L(Y \mid \mathbf{X}, M) - L(M) \Big].$$

Developing this further, the subgroup s that maximizes this expression is equivalent to the one that maximizes

$$\begin{aligned}
&L(Y \mid \hat{\Theta}^d) - L(Y \mid X, M) \\
&= L(Y^a \mid \hat{\Theta}^d) - L_{Bayes2.0}(Y^a \mid \mathbf{X}^a) - L(M) \\
&\sim \frac{n_a}{2} \log \frac{\hat{\sigma}_d^2}{\hat{\sigma}_a^2} + \left[\frac{1}{2\hat{\sigma}_d^2} \sum_{y^i \in Y^a} (y^i - \hat{\mu}_d)^2 \right] \log e - \frac{n_a}{2} \log e - \log n_a - L(M) \\
&= \frac{n_a}{2} \log \frac{\hat{\sigma}_d^2}{\hat{\sigma}_a^2} + \left[\frac{\sum_{y^i \in Y^a} (y^i)^2 - n\hat{\mu}_a^2 + n\hat{\mu}_a^2 - 2n\hat{\mu}_a\hat{\mu}_d - \hat{\mu}_d)^2}{2\hat{\sigma}_d^2} \right] \log e \\
&\quad - \frac{n_a}{2} \log e - \log n_a - L(M) \\
&= n_a \left[\log \frac{\hat{\sigma}_d}{\hat{\sigma}_a} + \frac{\hat{\sigma}_a^2 + (\mu_a - \mu_d)^2}{2\hat{\sigma}_d^2} \log e - \frac{\log e}{2} \right] - \log(n_a) - L(M) \\
&= n_a KL(\hat{\Theta}^a; \hat{\Theta}^d) - \log n_a - L(M),
\end{aligned} \tag{3.25}$$

where $n_a KL(\hat{\Theta}^a; \hat{\Theta}^d)$ is the usage-weighted Kullback-Leibler divergence between the normal distributions specified by the respective parameter vectors. Similar to the result for the nominal target in Section 3.4.3, this shows that *finding the MDL-optimal subgroup is equivalent to finding the subgroup that maximizes the weighted Kullback-Leibler (WKL) divergence*, an existing subgroup discovery quality measure [72], *plus two terms*. The first defines the complexity of the subgroup distribution with two parameters, the second compensates for multiple hypothesis testing (i.e., the number of possible subgroups). When we have a list with multiple subgroups, Eq. (3.18) expands to

$$\begin{aligned}
L(Y \mid \hat{\Theta}^d) - L(Y \mid \mathbf{X}, M) - L(M) &\sim \sum_{a_i \in S} n_i KL(\hat{\Theta}^i; \hat{\Theta}^d) - \sum_{a_i \in S} \log(n_i) - L(M) \\
&= \text{SWKL}(S) - \sum_{a_i \in S} \log(n_i) - L(M),
\end{aligned}$$

where SWKL(S) is the measure of subgroup set qualities that we proposed in Section 3.6, and the other terms penalize the complexity of the subgroup list.

Dispersion-correction quality measure. Importantly, we can observe from Eq. (3.18) that the measure based on the Kullback-Leibler divergence of normal distributions is part of the family of *dispersion-corrected* subgroup quality measures, as it takes into account both the centrality and the spread of the target values [12].

3.5.4 Relationship of MDL-optimal subgroup lists to Bayesian testing

When we have only one subgroup s in a subgroup list, the data encoding for numeric targets of Eq. (3.5.2) is equivalent to the negative logarithm of a Bayes factor [44, 110]. Indeed, the choice of the prior was based on the Bayesian one-sample t-test by Gönen et al. [44], and we effectively perform a one-sample t-test (including two extra terms) for each subgroup. Formally—and similar to the nominal case as described in Section 3.4.4—a Bayes factor K [58, 61] is given by the division of the likelihoods of the data given each hypothesis: $\Pr(D \mid M_1)/\Pr(D \mid M_2)$. If we use the maximization equivalent of Eq. (3.25),

$$\begin{aligned} L(Y \mid \hat{\Theta}^d) - L(Y \mid \mathbf{X}, M) - L(M) &= \log\left(\frac{\Pr(Y \mid \mathbf{X}, M)}{\Pr(Y \mid \hat{\Theta}^d)}\right) L(M) \\ &= \log K + L(M), \end{aligned}$$

we can see that we have the Bayes factor plus the model encoding. To transform code lengths in bits $L(\cdots)$ to probabilities $\Pr(\cdots)$ we used the Shannon-Fano code [114], which states that the best encoding is given by the negative logarithm of its probability for an event A, i.e., $L(A) = -\log \Pr(A)$. Our MDL-based criterion aims at maximizing a one-sample t-test for numeric targets between the subgroup distribution and the marginal distribution of the dataset while taking into account $L(M)$, which is equivalent to using the posterior distribution and penalizes for multiple hypothesis testing. When we aim to find subgroups beyond the first, it is trivial to see that we are testing each subgroup in S against the marginal distribution of the dataset.

3.6 A new measure for subgroup sets: the sum of WKL divergences

As discussed in Section 2.7, there is no SSD measure, to the best of our knowledge, that takes into account the individual quality of subgroups and their global quality over the whole dataset. Therefore, based on the results of Section 3.4.3 and 3.5.3, it is natural to extend the flexible WKL measure in subgroup discovery (described in Section 2.6.2) to subgroup sets.

That is, we propose the *Sum of Weighted Kullback-Leibler divergences* (SWKL), which can be interpreted as the sum of weighted KL divergences for the individual subgroups:

$$\text{SWKL}(S) = \frac{\sum_{i=1}^{\omega} n_i KL(\hat{\Theta}_j^i; \hat{\Theta}_j^d)}{|D|}, \tag{3.26}$$

where i is the index of each subgroup in a subgroup list, ω is the number of subgroups in S, and $|D|$ is the number of instances in D. The latter is used to normalize the measure and make values comparable across datasets. In the case of multiple target variables, the normalization could also include the number of targets, but that is not used in this thesis. The SWKL measure assumes that the data is partitioned per subgroup and that subgroups can be interpreted sequentially as a list, i.e., the second subgroup is interpreted as the description of the second subgroup is active, while the one of the first *is not* active.
An advantage of the SWKL measure is that it can be used for any type of target variable(s), as long as they are described by a probabilistic model. Note that computing SWKL is straightforward for subgroup lists, but not for subgroup *sets* as instances can be covered by multiple subgroups. For subgroup sets, it would be necessary to explicitly define the type of probabilistic overlap, e.g., additive or multiplicative mixtures of the individual subgroup models.

It should be noted that this measure only quantifies how well a list of subgroups capture the deviations in a given dataset and is prone to overfitting: the higher the number of subgroups, the easier it is to obtain a higher value as there is no penalty for the number of subgroups (or their complexities, for that matter). As such, SWKL can be seen as a measure for 'goodness of fit' for subgroup lists. This turns out to not be an issue for our approach though, as our MDL-based criterion naturally penalizes for multiple hypothesis testing and complexity of the individual subgroups. Further, it is neither an issue in our empirical comparisons in Section 5.3, as the number of subgroups found was similar for most algorithms, rendering the SWKL-based comparison valid.

3.7 Theoretical difference between subgroup list and predictive rule list

In this section, we show the difference between the objectives for subgroup discovery and predictive rules. We do this through the comparison of the equivalent maximization MDL scores for subgroup lists and classification rule lists [96] with only one rule—without loss of generality for greater sizes or regression tasks. To differentiate

both model classes SL and RL will be used for subgroup lists and classification rule lists, respectively.
First, let us recall the form of a subgroup list SL as given in:

$$\text{subgroup } 1 : \text{IF } a \sqsubseteq \mathbf{x} \text{ THEN } y \sim Cat(\hat{\Theta}^a)$$
$$\text{dataset} : \text{ELSE } y \sim Cat(\hat{\Theta}^d)$$

where $\hat{\Theta}^a$ are the estimated parameters of subgroup 1 and $\hat{\Theta}^d$ are the estimated parameters of the marginal distribution of the dataset and are thus constant for each dataset. On the other hand, the model form of a classification rule list RL takes the following form:

$$\text{predictive rule } 1 : \text{IF } a \sqsubseteq \mathbf{x} \text{ THEN } Cat(\hat{\Theta}^a)$$
$$\text{default} : \text{ELSE } y \sim Cat(\hat{\Theta}^{\neg a})$$

where $\hat{\Theta}^{\neg a}$ was used to emphasize that the default rule of a predictive rule list is not fixed, and is equivalent to the 'not rule 1'. This is the key difference between these two types of models: for subgroup lists the default rule is fixed to the marginal distribution of the dataset, while for predictive rule lists the default rule has the distribution of the negative set of the rules in the list. It should be noted that there are many definitions of rule lists for classification that use a fixed default rule, however having a variable default rule that maximizes the prediction quality is the best representative of predictive rule lists and of the objective of finding the best machine learning model, i.e., returning the best partition of the data with the smallest error possible. Note that a decision tree also belongs to this family of models, as any path starting at the root of the tree to one of its leaves also forms a rule, and thus, a decision tree is equivalent to a set of disjoint rules, i.e., none of the rules described in this way overlap on a dataset. For the type of classification rule lists defined above, the encoding of the first rule and default rule is given by Eq. (3.15) as for both rules the parameters are unknown.
Thus the MDL score of a predictive rule list can be rewritten as:

$$L(D, RL) = L(Y^a \mid \mathbf{X}^a) + L(Y^{\neg a} \mid \mathbf{X}^{\neg a}) + L(RL), \tag{3.27}$$

and note that the model encoding $L(RL) = L(SL)$ when having the same association rules.
Following the same steps as in Section 3.4.3, turning the MDL score objective from a minimization to maximization by multiplying by minus one and adding the constant $L(Y^d \mid \mathbf{\Theta}^d)$, we obtain the same objective as in Eq. (3.4.3):

$$\rho^* = \underset{s \in \mathcal{M}}{\arg\max} \left[L(Y^d \mid \mathbf{\Theta}^d) - L(Y \mid \mathbf{X}, RL) - L(RL) \right],$$

where ρ is the classification rule that maximizes the objective. Working out this equation, maximization objective of a *classification rule list* for a target variable of k class labels is given by:

$$\begin{aligned}
&L(Y \mid \hat{\Theta}^d) - L(Y \mid \mathbf{X}, M) - L(RL) \\
&= L(Y^a \mid \hat{\Theta}^d) + L_{NML}(Y^{\neg a} \mid \hat{\Theta}^d) - L(Y^a \mid X_a) - L_{NML}(Y^{\neg a} \mid \mathbf{X}^{\neg a}) - L(RL) \\
&= n_a KL(\hat{\Theta}^a; \hat{\Theta}^d) - \mathcal{C}(n_a, k) + n_{\neg a} KL(\hat{\Theta}^{\neg a}; \hat{\Theta}^d) - \mathcal{C}(n_{\neg a}, k) - L(RL),
\end{aligned} \tag{3.28}$$

This should be contrasted with the maximization objective of *subgroup list* of Eq. 3.18, which is given by:

$$\begin{aligned}
&L(Y \mid \hat{\Theta}^d) - L(Y \mid \mathbf{X}, M) - L(SL) = \\
&n_a KL(\hat{\Theta}^a; \hat{\Theta}^d) - \mathcal{C}(n_a, k) - L(SL).
\end{aligned}$$

Comparing the two previous equations, we can notice the most important distinction between subgroup discovery and classification: the *local* nature of subgroup discovery and the *global* nature of the classification task. In other words, subgroup discovery aims at finding subgroups that locally maximize their quality, independently of the rest of the dataset, and even though classification rules try to maximize their *local* quality also, they have to take into account the quality of their negative set, i.e., a classification rule cannot be considered by its quality alone, it has to be considered in terms of its *global* impact in the dataset.

On the other hand, this result also shows the similarity between both tasks and where the confusion sometimes arises, i.e., in particular cases the best subgroup can also be the best predictive rule. An example of this would be a very large dataset (relatively to the number of observations covered by the rule), and the best rule would cover a small number of observations compared to the rule formed by the negative set of that rule, i.e., $D^{\neg a}$, as a similar distribution to $\hat{\Theta}^d$, making $\hat{\Theta}^{\neg a} \sim \hat{\Theta}^d$.