Universiteit
Leiden
The Netherlands

Robust rules for prediction and description

Manuel Proenca, H.

Citation

Manuel Proenca, H. (2021, October 26). *Robust rules for prediction and description*. *SIKS Dissertation Series*. Retrieved from https://hdl.handle.net/1887/3220882

Version:	Publisher's Version
License:	Licence agreement concerning inclusion of doctoral thesis in the Institutional Repository of the University of Leiden
Downloaded from:	https://hdl.handle.net/1887/3220882

Note: To cite this publication please use the final published version (if applicable).

2 Preliminaries

In this chapter predictive rule lists and subgroup lists are presented. To that end, we give a gentle introduction to association rules, what constitutes a rule-based classifier and subgroup discovery, and how to measure the quality of a rule-based model in classification and subgroup discovery.

This chapter is divided as follows. First, in Section 2.1 we give a high-level introduction of association rules, rule-based classifiers, subgroup discovery, and subgroup set discovery. Next, in Section 2.2 the notation for supervised structured *i.i.d.* data is presented together with a formal definition of prediction, subgroup discovery, and subgroup set discovery tasks. Then, in Section 2.3 association rules and their characteristics are introduced. After that, in Section 2.4 the rule list model class is defined in general plus the specific case of the predictive rule lists and the subgroup list. Then, in Sections 2.5, 2.6, and 2.7 performance measures for classification, quality measures for subgroup discovery, and quality measures for subgroup set discovery are introduced, respectively.

2.1 Introduction to rules

The main topics of this thesis, rule-based classification and subgroup discovery, are two paradigms arising from related fields, machine learning and data mining, respectively. Both topics share the fact that they are supervised tasks on structured data that resort to *association rules* to construct their models. Thus, we will now informally introduce what each of these tasks encompasses, starting from what they have in common, and finalizing with their differences.

Association rule. An association rule $a \mapsto b$ is an assertion of a possible relationship between the antecedent a and consequent b, which can be read in the form of "*If* a appears in the data *then* b usually also appears" with a certain level of confidence [2]. A classic example from market basket analysis is that people who buy bread and butter (antecedent), usually, also buy milk (consequent) [2]. In this case, the association rule takes the form of: $\{bread = yes\}\&\{butter = yes\} \mapsto milk = yes$. A probabilistic extension of these rules, deemed a probabilistic association rule [81], associates a parametric probability distribution to the consequent, thus, instead of having a crisp decision, it has a probability associated with each possible case:

$$a \mapsto b \sim Dist(\Theta), \tag{2.1}$$

where Θ are the parameters of the distribution $Dist$ that describe the consequent. In the case of the previous example, where the consequent is a binary variable, this could take the form of: $\{bread = yes\}\&\{butter = yes\} \mapsto milk \sim Bernouilli(p_{yes} = 0.60; p_{no} = 0.40)$; where p_{yes} is the probability of having bought milk, and $p_{no} = 1 - p_{yes}$ the probability of not having bought it. A rule is said to be active in a region of the data D if for a data instance $\mathbf{x} \in D$ its antecedent is present, such as in our example $\{bread = yes\}\&\{butter = yes\}$.

Rule-based classifiers. Classification is the task of predicting an unseen outcome y of a discrete target variable from an instance of explanatory variables $\mathbf{x}$ [36]. In order to learn the relationship between the variables, a classification model is learned from a supervised dataset $D = \{\mathbf{X}, Y\}$, which is composed of paired examples $(\mathbf{x}, y)$. Note that we only talk about rule-based classifiers and not regressors because, to the best of our knowledge, there are no competitive rule-based models for regression.

Rule-based classifiers aggregate several rules together in order to perform classification. Combining rules in different ways leads to different rule-based models, of which two stand out [39]: $1)$ *rule list or sequential activation* [109, 85]—the activation of the rules for prediction follows a pre-determined order of the form *if rule 1 then* $Dist(\Theta^1)$... *else if rule 2 then* $Dist(\Theta^2)$, finishing with a default *else* $Dist(\Theta^m)$ that captures all the data not covered by any of the previous rules; $2)$ *rule set or overlapping rules* [21]—an unordered set of rules where several individual rules can be activated at the same time, overlapping. The key difference between both models is that rule lists are ordered and only one rule is active for one data sample $\mathbf{x}$, while rule sets are unordered and multiple rules can be active for one data sample $\mathbf{x}$.
The objective of a rule-based classifier is to maximize a performance measure, thus each association rule should contribute to that *global* goal, i.e., each rule should

take into account other rules to maximize the overall quality of the classifier. Looking back at our example, we see that $\{bread = yes\}\&\{butter = yes\} \mapsto milk \sim Bernouilli(p_{yes} = 0.60; p_{no} = 0.40)$ does not seem particularly good for prediction as it does not distinguish very well between both classes. On the other hand, the rule $\{yoghurt = yes\} \mapsto milk \sim Bernouilli(p_{yes} = 0.90; p_{no} = 0.10)$ seems well suited for prediction. A note should be made in relation to decision lists, which have the same format as rule lists, but instead of combining probabilistic association rules, they are composed of decision rules, with a crisp decision as in our example $\{bread = yes\}\&\{butter = yes\} \mapsto milk = yes$, and appeared first in the literature [109].

Subgroup Discovery (SD). Subgroup discovery is the data mining task of finding subgroups that stand out with respect to some given target variable(s). The definition of standing out, also known as interestingness, is quantified by a quality measure, which depends on the task at hand [123, 63]. In general, these measures quantify quality by how different the target variable distribution of a subgroup is from what is defined as 'normal' behavior in a dataset. In the case of structured data, a subgroup generally takes the form of an association rule, and the 'normal' behavior is usually measured by the average behavior of the target variable of that dataset [8]. Going back to the market basket analysis example, let us consider a dataset made up of the shopping baskets of different clients, and that has as target variable if a client bought milk (or not). 'Normal' behavior can be given by the percentage of people that buy milk over the whole dataset, and let us assume that this value is 90%. Thus, the subgroup defined by $\{bread = yes\}\&\{butter = yes\} \mapsto milk \sim Bernouilli(p_{yes} = 0.60; p_{no} = 0.40)$ seems interesting, as compared with normal behavior, people that buy bread and butter buy milk 33% times fewer times than an average client. This is in clear contrast with rules for prediction, as subgroups that are interesting do not have to divide well between classes: they need to stand out with respect to what is 'normal' behavior in the data. Sometimes, depending on the dataset and task at hand, a good subgroup will also be a good predictive rule, but both tasks arise from different goals and should thus not be confused. In its standard form, subgroup discovery is called top-k mining, as the goal is to find the k top subgroups that maximize a user-defined quality measure. As the quality measures only quantify the individual quality of a subgroup, top-k mining is a *local* paradigm, as it is only concerned with the independent performance of the k subgroups on the respective data covered by each of their descriptions. Top-k subgroup discovery usually finds subgroups that cover the same region of the data, hence it returns redundant subgroups for many datasets. As a solution to this, *subgroup set discovery* was proposed.

Subgroup Set Discovery (SSD). The task of finding a non-redundant set of subgroups that are individually and collectively interesting at the same time is called Subgroup Set Discovery (SSD) [75]. Contrary to a predictive paradigm, the objective is that the subgroups still abide by the standard subgroup discovery principle of *locally* standing out with respect to the 'normal' behavior, while at the same time, *globally* describing different regions of the dataset. To extend subgroup discovery to its set form, two main models exist: 1) *subgroup lists or ordered sets* [71]—a set of subgroups that should be interpreted sequentially and where no subgroup is allowed to overlap in the same region of data as another, take the form of *if subgroup 1 then* $Dist(\Theta^1)$... *else if subgroup 2 then* $Dist(\Theta^2)$, etc.; and 2) *subgroup sets or overlapping sets* [74]—a set of subgroups where each subgroup can be interpreted individually and overlap is allowed according to a definition of overlap interaction. Both extensions have their advantages and disadvantages: while subgroup lists are less interpretable, they have the advantage of a clear definition of the relevance of each subgroup and which subgroup explains each data point. On the other hand, subgroup sets allow for a (semi)independent interpretation of the subgroups, but they require an extra definition that favors non-redundant sets together with a definition of the interaction of subgroups in the region where they overlap, e.g., as a mixture model.

Rule-based classifiers *versus* Subgroup Set Discovery. As was shown throughout this section, predictive rules and subgroups share a lot of the same characteristics. Rule-based classifiers aggregate association rules to maximize a *global* objective of a good overall classification, while subgroup sets balance both a *local* definition of quality with respect to the 'normal' behavior of the dataset and a *global* objective of covering different regions of the data. It is natural that for some datasets good subgroups will be good predictive rules and vice versa, but this is not always the case and it should be distinguished. Throughout this work, we will be referencing them separately to emphasize the different paradigms: *1) predictive rule* will refer to an association rule as used in rule-based models for classification in machine learning; *2) subgroup* to descriptive rules in subgroup discovery; and *3) association rule* or just *rule* to an association rule in general, i.e., when it refers to either a predictive rule or a subgroup. All their idiosyncrasies may not be apparent yet, but as we progress we will continue to emphasize their similarities and differences.

2.2 Supervised data

Consider a dataset $D = (\mathbf{X}, \mathbf{Y}) = \{(\mathbf{x}^1, \mathbf{y}^1), (\mathbf{x}^2, \mathbf{y}^2), ..., (\mathbf{x}^n, \mathbf{y}^n)\}$ of n *i.i.d.* instances. Each instance $(\mathbf{x}^i, \mathbf{y}^i)$ is composed of a vector of explanatory variable values $\mathbf{x}^i$ and a

vector of target variable values $\mathbf{y}^i$.
Each observed explanatory vector has m values $\mathbf{x} = [x_1, ..., x_m]$, one for each variable $X_1, ..., X_m$. The domain of a variable X_j, denoted $\mathcal{X}_j$, can be one of two types: nominal or numeric. Similarly, each observed target vector is composed of t values $\mathbf{y} = [y_1, ..., y_t]$, one for each target variable $Y_1, ..., Y_t$, with associated domains $\mathcal{Y}_j$. The target variables can be one of two types: nominal, or numeric. In the nominal case it is $\mathcal{Y}_j = \{1, \cdots, k\}$, with $\mathcal{Y}_j$ the set of k classes/categories of variable Y_j, and in the numeric, the domain is $\mathcal{Y}_j = \mathbb{R}$.
Note that we use subscripts on the dataset variables ($D, \mathbf{X}, \mathbf{Y}, X, Y, x, y$) to indicate column subsets and overscripts to subset over rows. In the case of other notation, such as number of elements n or statistics μ, σ we will not use the superscript as it can be confused with the exponentiation of that value. Also, X_i (resp. Y_i) refers to both the properties of the i^{th} explanatory (resp. target) variable and to all the values of this variable for a specific column. When the dataset only contains one target variable $\mathbf{Y}$ is substituted by Y.

Prediction In statistical learning, the task of prediction is to infer unseen values of a target variable from a set of explanatory variables through the use of past evidence that shows the relationship between target and explanatory variables [36]. Formally, this means that we want to find the best mapping g, from a space of possible hypotheses $\mathcal{G}$, between explanatory data $\mathbf{X}$ to target data Y (in the univariate case and without loss of generality). This mapping can be summarized as $g : \mathcal{X}_1 \times \cdots \times \mathcal{X}_m \rightarrow \mathcal{Y}$; and in the case of a probabilistic predictor, such as ours, this mapping is just a conditional probability $g(x) = \Pr(y \mid \mathbf{x} = x)$, and by abuse of notation $g(\mathbf{X}) = \{g(x^1), \cdots, g(x^n)\}$. Assuming that we are dealing with probabilistic mappings, we can now start making predictions $\hat{y}$ for the target variable values for each instance $\mathbf{x}$, by returning the outcome with the largest probability

$$\hat{y} = \arg\max_{y \in \mathcal{Y}} \Pr(y \mid \mathbf{x}) \tag{2.2}$$

The characteristics of a good mapping are: 1) capture the properties in $\mathbf{X}$ that allow predicting Y; and 2) generalize well on previously unseen data $D_{new} = \{\mathbf{X}_{new}, Y_{new}\}$. In order to choose the best possible mapping, we need to introduce a performance measure $meas$ that empirically quantifies the quality of our mappings for a given dataset, formally $meas : \mathcal{Y}^n \times \mathcal{Y}^n \rightarrow \mathbb{R}_{\geq 0}$. Thus the problem of finding the best mapping g in a dataset $D = \{\mathbf{X}, Y\}$ reduces to:

$$g^* = \arg\max_{g \in \mathcal{G}} meas(Y, g(\mathbf{X})), \tag{2.3}$$

but then another, D_{new} is required for evaluation, as this takes into account generalization and avoids overfitting. Some examples of measures $meas$ for classification

are the accuracy or the AUC, described in Section 2.5, or the Mean Squared Error for regression.
Several variations exist, such as using only predictions $\hat{y}$ instead of $g(\mathbf{x})$ or structural measures that add an extra term to $meas$ to penalize for the structural complexity of the mapping [119]. E.g., in the case of nested mappings such as a polynomial regression, the use of higher-order polynomials is "more complex" than lower-order ones, as they have extra terms. The Minimum Description Length (MDL) principle used throughout this dissertation, is a type of probabilistic structural error minimization principle and this mapping g is called a model M or point hypothesis in it [47].

Subgroup discovery Subgroup discovery is the data mining task of discovering *unkown* patterns in the data that stand out with respect to a target variable [116]. In mathematical terms the objective is to find a mapping between descriptions a of the explanatory data $\mathbf{X}$ and the target variable Y (for the univariate case without loss of generality) that stand out in relation to the 'normal' behavior of the target variable Y. Formally, a description is a function $a : \mathcal{X}_1 \times \cdots \times \mathcal{X}_m \mapsto \{false, true\}$. And in our specific case, a description a is a conjunction of conditions on $\mathbf{X}$, each specifying a value or interval on a variable. The domain of possible conditions depends on the type of a variable: numeric variables support *greater and less than* $\{\geq, \leq\}$; nominal support *equal to* $\{=\}$. E.g., from Figure 2.2, where for the *Car import* dataset, a description can be "weight = heavy & consumption-city ≤ 8 km/L", where the variable weight is conditioned to one value (norminal variable) and $consumption - city$ is conditioned to one interval (numeric variable). As the dataset is made of pairs $(\mathbf{x}^i, y^i)$, for each description a there is an associated subset of data $D^a = \{\mathbf{X}^a, Y^a\}$ with $n_a = D^a$ instances, and an associated empirical parameter distribution of the target Y^a given by $\hat{\Theta}^a$—where the parameters depend on the distribution selected by the user. Thus, in the case of *i.i.d.* data, a subgroup is an association rule $s : a \mapsto y \sim Dist(\hat{\Theta}^a)$.
To quantify how interesting a subgroup s with description a is, we need to define a quality measure $q(n_a, \hat{\Theta}^a, \hat{\Theta}^d)$ that is a function of the subgroup empirical distribution $\hat{\Theta}^a$ and the dataset empirical marginal distribution $\hat{\Theta}^d$—'normal' behavior of the dataset.
Formally the best subgroup, or top-1 subgroup, is given by

$$s^* = \underset{s=(a,\hat{\Theta}^a)\in\mathcal{A}}{\arg\max}\ q(n_a, \hat{\Theta}^a, \hat{\Theta}^d), \tag{2.4}$$

where in the case of top-k subgroup discovery, we return the k top ranking subgroups that maximize q. An example of a quality measure for binary targets in the Weighted Relative Accuracy (WRAcc) or the Weighted Kullback-Leibler (WKL) divergence presented in Section 2.6. Contrary to *prediction*, SD does not aim at performing

well on *unseen* data, but on discovering interesting patterns in the *seen* data.

Subgroup set discovery The objective of subgroup set discovery (SSD) is to find a set of subgroups S that are both (individually) high-quality and non-redundant, i.e., cover different regions of the data [75]. Thus, it uses a *local* idea of quality measure from subgroup discovery plus a *global* concept of covering different regions of the dataset. Given this vague trade-off, SSD objectives have only been defined heuristically in different works, either by sequentially covering or by weighting instances [71, 75].
In the sequential approach one iteratively finds subgroups by: $1)$ discovering the top-1 subgroup according to quality measure as in Eq. (2.4); $2)$ removing the data covered by that subgroup—or in some cases, only the data of a certain class in binary SSD[71]—from the dataset, thus getting $D^{\neg a} = D \setminus D^a$; and $3)$ repeating 1 and 2 until the desired number is reached or no more subgroups can be found. The weighting approach follows the same iterative approach as the sequential one, but instead of removing the whole data in step 2, it reweighs each instance if it was already covered by subgroups selected in previous iterations. Formally, SSD can be defined as a dependent system of equations:

$$\begin{aligned} s_1 &= \underset{s=(a,\hat{\Theta}^a)\in\mathcal{A}}{\arg\max}\ q(n_a, \hat{\Theta}^a, \hat{\Theta}^d), \\ s_2 &= \underset{s=(a,\hat{\Theta}^a)\in\mathcal{A}|(s_1)}{\arg\max}\ \tilde{q}(n_a, \hat{\Theta}^a, \hat{\Theta}^d, s_1), \\ &\vdots \\ s_k &= \underset{s=(a,\hat{\Theta}^a)\in\mathcal{A}|(s_1,\cdots,s_{k-1})}{\arg\max}\ \tilde{q}(n_a, \hat{\Theta}^a, \hat{\Theta}^d, s_1, \cdots, s_{k-1}), \end{aligned} \tag{2.5}$$

where $\mathcal{A}|(s_1, \cdots, s_{k-1})$ represents that the space of possible subgroups and their empirical distributions depend on the subgroups found so far, $\tilde{q}$ means that the quality measure can be slightly modified by a weighting, given previous selected subgroups.
As there are no SSD global quality measures, in Section 2.7 we describe what the characteristics of a SSD quality measure $Q(S, Y)$ over a whole dataset should be, and in Section 3.6 we propose the Sum of Weighted Kullback-Leibler (SWKL) divergences as an SSD measure for sequential subgroup sets—that could in the future be adapted for non-sequential sets.

Tasks. Depending on the type (nominal or numeric) and number of targets (one or multiple), and the task at hand—rule-based prediction or subgroup discovery—the type of problem for each task can be divided into *four* categories.
First, for the case of rule-based prediction, it can be divided as: *1) classification:*

univariate nominal target; *2) regression:* univariate numeric target; *3) multi-label or multi-target classification:* multivariate binary or nominal targets, respectively; and *4) multi-target regression:* multivariate numeric targets.
Second, for the case of subgroup discovery the names are themselves self explanatory: *1) single-nominal*; *2) single-numeric*; *3) multi-nominal*; and *4) multi-numeric*.

2.3 Association rules, predictive rules and subgroups

Association rules are the shared building block of rule-based classification and subgroup discovery. To distinguish the tasks, when we mention and association rule r, we are talking about its general form and it can refer to both a predictive rule ρ and a subgroup s.

An association rule r, henceforth, called rule, consists of a *description* (also intent) that defines a *cover* (also extent), i.e., a subset of dataset D. Examples of association rules are given in Figures 2.1 and 2.2

		$\mathbf{Pr}(\boldsymbol{animaltype} = \cdots \mid \boldsymbol{a})$ in %						
description of animal	n	Mammal	Fish	Invert.	Bug	Reptile	Amph.	Bird
backbone = no	18	0	0	56	44	0	0	0

Figure 2.1: Example of one rule from the *Zoo* dataset with coverage n and one nominal target variable characterized by a categorical distribution with parameters p_i for each class in {Mammal; Fish; Invert.; Bug; Reptile; Amph.; Bird}.

		$\boldsymbol{price}$ (K)	
description of automobile specifications	n	μ	σ
weight = heavy & consumption-city $\leq$ 8 km/L	11	35	8

Figure 2.2: Example of one rule from the *Automobile import 1985* with a numeric target variable characterized by a normal distribution with coverage n, mean μ, and standard deviation σ.

Rule description: A description a is a Boolean function over all explanatory variables X. Formally, it is a function $a : \mathcal{X}_1 \times \cdots \times \mathcal{X}_m \mapsto \{false, true\}$. In our case, a description a is a conjunction of conditions on $\mathbf{X}$, each specifying a specific value

or interval on a variable. The domain of possible conditions depends on the type of a variable: numeric variables support *greater and less than* $\{\geq, \leq\}$; nominal support *equal to* $\{=\}$. The size of a description a, denoted $|a|$, is the number of conditioned variables it contains.

Example 1: In Figure 2.2, the rule description size is $|a| = 2$, with one condition on a nominal variable: {weigth $=$ heavy}; and another on a numeric variable: {consumption-city $\leq 8km/L$}.

Rule cover: The cover is the bag of instances from D where the rule description holds true. Formally, it is defined by:

$$D^a = \{(\mathbf{x}, \mathbf{y}) \in D \mid a \sqsubseteq \mathbf{x}\} = \{X_1^a, \cdots, X_m^a, Y_1^a, \cdots, Y_t^a\} = \{\mathbf{X}^a, \mathbf{Y}^a\}, \tag{2.6}$$

where we use $a \sqsubseteq \mathbf{x}$ to denote $a(\mathbf{x}) = true$. Further, let $n_a = |D^a|$ denote the coverage of the subgroup, i.e., the number of instances it covers.
Example 2 (continuation): In Figure 2.2, the rule covers 11 instances in the dataset which can be found by the instances in the data where its description is $true$, and thus its coverage is 11.

2.3.1 Interpretation as probabilistic rule

As D^a encompasses both the explanatory and target variables, the effect of a on the target variables can be interpreted as a probabilistic rule. In this thesis, we will assume that the target variables are *independent* as this simplifies the problem and is a common approach in, e.g., multi-label classification [53]. Thus, the general form of a rule is

$$a \mapsto y_1 \sim Dist(\hat{\Theta}_1^a), \cdots, y_t \sim Dist(\hat{\Theta}_t^a), \tag{2.7}$$

where y_j is a value of variable Y_j, $Dist$ is a probability distribution (defined later) and $\hat{\Theta}_j^a$ is the shorthand for the maximum likelihood estimation of the parameters of $Dist$ over values Y_j^a, i.e., $\hat{\Theta}_j^a = \hat{\Theta}_j(Y^a)$. Thus, $y_j \sim Dist(\hat{\Theta}_j^a)$ tells us that the values of variable Y_j are distributed according to a distribution $Dist$ with parameters $\hat{\Theta}_j^a$ estimated over the values Y_j^a. The vector of all parameter values of a rule is denoted by Θ^a. In our case, $Dist$ is a *categorical* or *normal* distribution for the nominal or numeric target case, respectively. For numeric targets other distributions could have been chosen, however, the *normal* distribution incorporates some of the most relevant information of the data through the mean and variance of the data, it is well studied for regression problems [36], and can be solved in a closed form from a Bayesian [58] and MDL [48] perspective. For an analysis on the direct use of the numeric empirical distribution in subgroup discovery please refer to Meeng et al. [89]. In the numeric

case, the normal distribution is represented as $\mathcal{N}(\hat{\mu}, \hat{\sigma})$, where $\hat{\mu}$ and $\hat{\sigma}$ are the mean and standard deviation of the distribution estimated from the data. In the nominal case, the distribution is $Cat(\hat{p}_1, \cdots, \hat{p}_k)$, where k is the number of classes (or categories) of the corresponding variable and $\hat{p}_c$ the estimated probability for class c.

The categorical distribution is a natural choice for describing the probabilities of classes [81] and the normal distribution captures two properties of interest in numeric variables, i.e., center and spread, while being robust to cases where the data violates the normality assumption [48].
Example 3 (continuation): Revisiting the *Automobile import* example list in Figure 2.2, the description and corresponding statistics are $a = \{\text{weight} = \text{heavy \& consumption-city} \leq 8\,\text{km/L}\}$ and $\hat{\Theta}^a = \{\hat{\mu} = 35; \hat{\sigma} = 8\}$, respectively, where the units are thousands of dollars (K). This corresponds to the following normal probability distribution:

$$\text{price (K)} \sim \mathcal{N}(\hat{\mu} = 35; \hat{\sigma} = 8)$$

Example 4 (continuation): In the case of the *Zoo* rule of Figure 2.1, the description is $a = \{\text{backbone} = \text{no}\}$, and its corresponding statistics are $\hat{\Theta}^a = \{\hat{p}_1 = 0; \hat{p}_2 = 0; \hat{p}_3 = 0.56; \hat{p}_4 = 0.44; \hat{p}_5 = 0; \hat{p}_6 = 0; \hat{p}_7 = 0\}$, where the class labels $1, ..., 7$ correspond to the animal types in the order of Figure 2.1. The target variable follows the following categorical distribution:

$$\text{animal_type} \sim Cat(\hat{p}_1 = \hat{p}_2 = \hat{p}_5 = \hat{p}_6 = \hat{p}_7 = 0.00; \hat{p}_3 = 0.56; \hat{p}_4 = 0.44)$$

2.3.2 Maximum likelihood estimation

The most common way to estimate the parameters of a probability distribution from a dataset is by using the Maximum Likelihood (ML) estimator [93]. In later chapters we also use other estimators, but the ML is still an important building block of these more complex methods.

As shown previously, each description a uniquely defines a subset D^a given by its cover Eq. (2.6). Next, we will show how to estimate the parameters for each type of target variable.
In the **nominal** case, the parameters of the distribution $Cat(\Theta^a)$ are the probabilities associated with each class c, i.e., $\Theta^a = \{p_{c=1|a}, \cdots, p_{c=k|a}\}$, for a domain $\mathcal{Y} = \{1, \cdots, k\}$. Note that we use $\cdot|a$ as a shorthand for conditional on a, as e.g., $p_{c=1|a} = \Pr(c = 1 \mid a \sqsubseteq \mathbf{x})$. Thus, for each class label c, we need to find its subset of

the data $D^{c|a}$, formally given by:

$$D^{c|a} = \{(\mathbf{x}, y) \in D^a \mid y = c\}. \tag{2.8}$$

which allows us to compute the usage over each class $n_{c|a} = |D^{c|a}|$. Now, we are in a position to use the maximum likelihood estimator for the parameters $\hat{\Theta}^a$ of each categorical distribution as:

$$\hat{p}_{c|a} = \frac{n_{c|a}}{n_a}, \tag{2.9}$$

where n_a is the total number of instances and $n_{c|a}$ is the number of instances of class c in the dataset subset D^a.
In the **numeric** case the parameters of the distribution $\mathcal{N}(\Theta^a)$ are the mean and standard deviation, i.e., $\Theta^a = \{\mu, \sigma\}$. They can be directly estimated from the target data Y^a:

$$\hat{\mu}_a = \frac{1}{n_i} \sum_{y \in Y^a} y, \tag{2.10}$$

$$\hat{\sigma}_a^2 = \frac{1}{n_i} \sum_{y \in Y^a} (y - \hat{\mu}_a)^2, \tag{2.11}$$

where $\hat{\sigma}_a^2$ is the biased estimator such that the estimate times n_a equals the Residual Sum of Squares, i.e., $n_a \hat{\sigma}_a^2 = \sum_{y \in Y^a} (y - \hat{\mu}_a)^2 = RSS_a$.

2.4 Rule lists, predictive rule lists, and subgroup lists

Sequentially aggregating rules for prediction and subgroup discovery seamlessly leads to *predictive rule lists* and *subgroup lists*, respectively. They have the same structural format and share the same model class, dubbed *rule list* model class, which takes the format of Figure 2.3, with the *only difference* being how the parameters of the last rule, also known as default rule, are chosen.

The *rule list* is an ordered set of rules, and it contains two parts: $1)$ the part of the rule list that contains the ω ordered rule descriptions $\{a_1, \cdots, a_\omega\}$, which is denoted by R for predictive rule lists and S for subgroup lists; and $2)$ the default rule r_d. Together, both parts form the whole model M. In a rule list, only one rule is activated for each instance, hence each rule only activates in a unique part (subset) of the dataset. If no rule gets activated, that instance will activate the default rule. As an example, to characterize an instance $\mathbf{x}$ with a rule list, one starts by checking the first rule and verify if $a_1 \sqsubseteq \mathbf{x}$ is $true$ or $false$. In case it is $true$, $\mathbf{x}$ belongs to that rule. In case it is $false$, we proceed to check the second rule and so forth, until finding one that returns

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

Figure 2.3: Generic rule list model M with ω rules and t (number of target variables) distributions per rule.

$true$. In case no rule is $true$, that instance activates the default rule.

Cover of a rule in a rule list. We observe that for any given rule list of the form of Figure 2.3, *any individual instance* $(\mathbf{x}^i, \mathbf{y}^i)$ *can only be 'covered' by one rule*. That is, the cover of a_i, denoted D^a, depends on the order of the list and is given by the instances where its description occurs minus those instances covered by previous descriptions:

$$D^i = \{\mathbf{X}^i, \mathbf{Y}^i\} = \{(\mathbf{x}, \mathbf{y}) \in D \mid a_i \sqsubseteq \mathbf{x} \wedge \left(\bigwedge_{\forall_{i'<i}} a_{i'} \not\sqsubseteq \mathbf{x} \right)\}. \tag{2.12}$$

Next, let $n_i = |D^i|$ be the number of instances covered by the i^{th} description (also known as *usage*). In case an instance $(\mathbf{x}^i, \mathbf{y}^i)$ is not covered by any rule *then it is 'covered' by the default rule*. The instances covered by the default rule D^d are the ones not covered by any rule (hence the name default rule), formally defined as:

$$D^d = \{\mathbf{X}^d, \mathbf{Y}^d\} = \{(\mathbf{x}, \mathbf{y}) \in D \mid \forall_{a_i \in M} a_i \not\sqsubseteq \mathbf{x}\}. \tag{2.13}$$

Maximum Likelihood estimator. Given the partition property of the rule lists, it is straightforward to see that the ML estimators of Eq. (2.9), (2.10), and (2.11) still hold if D^a is replaced by D^i.

What predictive rule lists and subgroup lists have in common is that they are interpreted in order and that each predictive rule or subgroup distribution parameter, with the exception of the default rule, is estimated in their respective subsets D^i.

The difference between predictive rule lists and subgroup lists is how the *default rule* distributions parameter are estimated! In the case of a *predictive rule list*, the default rule is just an ordinary rule that characterizes its subset, thus its parameters $\hat{\Theta}_1^d \cdots \hat{\Theta}_t^d$ are estimated in that subset. In the case of a *subgroup list*, the default rule is fixed to the marginal distribution of the target and it is constant for a dataset. If the mean and standard deviation of a numeric target in a dataset are 18 and 13, respectively, the subgroup list default rule will be fixed at those values, independently

of the (number of) subgroups in the list. This may seem like a subtle difference, but it represents a radical difference in what defines an optimal predictive rule list or subgroup list.

The intuition is the following: in a predictive rule list, the goal is to predict as best as possible an unseen data point, thus each rule should represent homogeneous subsets of the data, and the way the default rule predicts best is if its distribution represents well its subset of the data. In a subgroup list, the goal is to find subsets of the data that have different distributions than the marginal distribution of the dataset, and the default rule covers and represents well all data that follows the marginal distribution. This, in turn, incentivizes the optimal subgroup list to have subgroups that follow distributions different than the default rule distribution, as instances well represented by the default will not be covered by subgroups. Structurally both models look very similar, but by having different definitions of optimality, each model type will favor different types of association rules.

As an *example*, one can look at Figures 2.4a and 2.4b. Given that the dataset has few distinguishing variables and few samples both predictive rules and subgroups are mostly the same, as good predictive rules are also good subgroups in this case. Nonetheless, one can see that the default rules are different. In the predictive rule list, the default rule is clearly predictive, while in the subgroup list it is the original distribution of the dataset.

2.5 Classification performance measures

Classification is a *global* predictive paradigm, thus its measures quantify the quality of a model over the whole dataset [36]. As classification is expected to generalize to unseen points, the quality of a classifier should be measured in data that is different than the one used for training and choosing the classifier. In order to achieve this and have some statistical guarantees of the generalization, each dataset is usually divided into different parts, using some of its parts to train and the rest to test. The most well-known of these techniques is called k-fold cross-validation, where one randomly divides the data into k (approximately) equal parts. Then, one uses $k-1$ parts to train the model, and the 1 part left to test the performance of that trained model, repeating the process k times, until all data was used to test. In the end, the performance over k folds is averaged out.

Classification measures can broadly be divided into two types: $1)$ aggregators of classifier decisions, such as accuracy, precision, or recall; $2)$ measures of how a classifier discriminates between classes, taking into account the confidence with which it classifies different classes, such as Area Under the receiver operator Curve (AUC) or

			$\mathbf{Pr}(\boldsymbol{animaltype} = \cdots \mid \boldsymbol{r})$ in %						
ρ	**description**	n_ρ	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
default rule		41*	100	0	0	0	0	0	0

(a) **Predictive rule list** for zoo dataset. Default rule with distribution estimated from its subset.

			$\mathbf{Pr}(\boldsymbol{animaltype} = \cdots \mid \boldsymbol{s})$ 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

(b) **Subgroup list** for zoo dataset. Dataset rule with distribution equal to the marginal distribution of the dataset.

Figure 2.4: Illustrative example of a predictive rule list and subgroup list with the *Zoo* dataset obtained with our method. *Zoo* contains one nominal target variable with 7 classes, 101 instances, and 15 binary and 1 numeric variables. n_i refers to the number of instances covered by i^{th} predictive rule or subgroup defined by 'description'. $\Pr(animaltype = * \mid r)$ denotes the estimated probability (in %) of each class label occurring within the subgroup. *Zoo* is a highly structured dataset, thus both rule list and subgroup list found mostly the same descriptions, as in these case good subgroups are also good predictive rules. However, it is important to check that the 'default rules' are indeed different and thus incite the method to find different types of rules. $*$ 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 estimators.

likelihood. In this thesis, we will focus on accuracy, from the first group, as it gives an overall idea of how the classifier takes decisions and AUC from the second group. It should be noted that likelihood is more related to the MDL theory we use, but its interpretation is not as easy, and so it will not be used for ranking classifiers.

The building blocks for describing the measures are the terms defined by the intersection of the true class label and the predicted class label. We will start with the binary setting, when just two classes exist, the *positive* class, also known as the class of interest, and the *negative* class. With two classes, there are four possible characterizations of decisions based on which class it is and if it is correctly predicted by the classifier or not:

- *True Positives (TP)* - Number of instances (correctly) predicted as positive that are positive.
- *True Negatives (TN)* - Number of instances (correctly) predicted as negative that are negative.
- *False Positives (FP)* - Number of instances (incorrectly) predicted as positive that are negative.
- *False Negatives (FN)* - Number of instances (incorrectly) predicted as negative that are positive.

In the multiclass scenario, we can define all these terms per class c, where the positive class represents the class of interest c and the negative class, either represents all the other classes in a *one-versus-all* or another class in a *one-versus-one* setting. In *one-versus-all*, the performance of each class is measured by creating two classes, positive class (class of interest), and negative class, where the negative class is all classes except the positive one. In the second case, *one-versus-one* requires comparing each class against each class. In this work, we will focus on *one-versus-all* as it is the most common and simpler to understand [106]. Thus, for class c we can define *True Positive of* c (TP_c), *True Negative of* c (TN_c), *False Positives of* c (FP_c), and *False Negatives of* c (FN_c).

Accuracy. Given the previous definitions, Accuracy can be immediately defined as the ratio of correctly identified points to all points, formally:

$$Accuracy = \frac{\sum_{c \in \mathcal{Y}} TP_c}{\sum_{c \in \mathcal{Y}} TP_c + FN_c}, \tag{2.14}$$

where for the binary case we have $TP+TN$ in the numerator and $TP+TN+FP+FN$ in the denominator. Even though accuracy gives us an idea of how the classifier makes

predictions on the data, it has one main problem: if the original data is *imbalanced* it can give an erroneous idea of the quality of the classifier. As an example, if the target class is binary, and the majority class is made of 90% of the points, it is straightforward to see that an accuracy of 90% just requires us to choose all the points as the majority class, and that is not a very interesting classifier. In order to correct this, balanced accuracy was introduced [17].

Balanced Accuracy. Contrary to accuracy, balanced accuracy takes into account the classification of each class separately, by giving the same importance to each class of the positive correctly predicted ratio, True Positive Rate (TPR), or recall. Formally, it is given by:

$$bAcc = \frac{1}{|\mathcal{Y}|} \sum_{c \in \mathcal{Y}} \frac{TP_c}{TP_c + FN_c} \tag{2.15}$$

Area under the ROC Curve (AUC). The area under the receiver operator curve, which is only properly defined for binary problems, is given by the two-dimensional plot of True Positive Rate (TPR) against the False Positive Rate (FPR), as one varies the threshold T of classification. The FPR is just $FP/(FN + TP)$, and the threshold is a value above which a point is classified as belonging to the positive class, i.e., $\mathbf{x}$ is classified as positive $c = 1$ if $\Pr(1 \mid \mathbf{x}) > T$. AUC is not restricted to probabilistic classifiers but for ease of presentation we only consider these. Formally, the AUC is the area under the curve of TPR and FPR as a function of the threshold, which is given by the integral:

$$AUC = \int_0^1 TPR(T)FPR(T)\,\mathrm{d}T = \Pr(\mathbf{x}_{pos.} > \mathbf{x}_{neg.}), \tag{2.16}$$

where $T \in [0, 1]$ as we only consider probabilistic classifiers and $\Pr(\mathbf{x}_1 > \mathbf{x}_0)$ is the probability that a randomly selected positive class example will rank higher than a randomly selected negative class example in terms of [33]. An easier way to interpret the AUC is through the Wilcoxon-Mann-Whitney statistic [20], an unbiased estimator given by:

$$AUC = \frac{\sum_{\mathbf{x}_0 \in D_0} \sum_{\mathbf{x}_1 \in D_1} \mathbb{1}[\Pr(1 \mid \mathbf{x}_0) < \Pr(1 \mid \mathbf{x}_1)]}{|D_0| \cdot |D_1|}, \tag{2.17}$$

where D_0 and D_1 are the negative and positive labeled examples in D, respectively, and $\mathbb{1}$ is the indicator function that is 1 if $\Pr(1 \mid \mathbf{x}_0) < \Pr(1 \mid \mathbf{x}_1)$ and zero otherwise. Contrary to accuracy, and similarly to balanced accuracy, AUC gives the same importance to both classes. Nonetheless, in its current format, AUC is not suited for multiclass.

Multiclass AUC. To extend binary AUC to multiclass two things have to be taken into account: how to compare different classes, and how to average performance per class. To compare different classes, the two most common approaches are *one-versus-all* and *one-versus-one*. As mentioned before, in this work, we will focus on *one-versus-all*. Regarding how to average the performance per class, three methods exist: *micro* average; *macro* average; and *weighted* average. *Micro* average takes into account each example, and in the case of *one-versus-all* transforms the dataset into a binary problem, where 1 is the positive class of interest, and 0 otherwise, and computes the AUC for the whole data. *Macro* average computes one AUC per class by, for each class, transforming the dataset into a positive class versus negative class (all other classes), and then computing the average of all AUCs. *Weighted* average computes one AUC per class like the macro average, but then averages all AUCs weighted with the percentage of class examples in the data. For *macro* and *weighted* AUC (easier to present), the formulas are:

$$AUC_{macro} = \frac{1}{|\mathcal{Y}|} \sum_{c \in \mathcal{Y}} AUC(c), \tag{2.18}$$

and

$$AUC_{weighted} = \sum_{c \in \mathcal{Y}} AUC(c) \frac{|D^c|}{|D|}, \tag{2.19}$$

where $AUC(c)$ is the one-versus-all AUC for class label c and $\frac{|D^c|}{|D|}$ is the frequency of that same class label.

2.6 Subgroup discovery measures

As shown before, subgroup discovery can broadly be divided into two categories: its classic form, also known as top-k mining; and Subgroup Set Discovery (SSD). In the first, only the individual quality of a subgroup is measured, hence quality is quantified independently and *locally* for each subgroup. In the second, SSD takes into account the *local* quality of individual subgroups while also taking into a account how well they cover the whole dataset.

Contrary to classification and prediction in general, the goal of subgroup discovery is to describe the dataset well and not to measure the prediction quality on unseen data points. Thus, the quality of the subgroups or subgroup sets is traditionally measured in the dataset where the model is trained. We will first introduce the quality measures for top-k subgroup discovery, and then in Section 2.7 proceed to generalize for subgroup sets.

2.6.1 Top-k quality measures

Top-k stands for finding the k subgroups that maximize a certain quality measure [8]. To assess the quality (or interestingness) of a subgroup description a, a measure that scores subsets D^a needs to be chosen. The measures used vary depending on the target and task, but in general they have two components: 1) representativeness of the subgroup in the data, based on coverage $n_a = |D^a|$; and 2) a function of the difference between statistics of the empirical target distribution of the pattern, $\hat{\Theta}^a = \hat{\Theta}(\mathbf{Y}^a)$, and the overall empirical target distribution of the dataset, $\hat{\Theta}^d = \hat{\Theta}(\mathbf{Y})$. The latter corresponds to the statistics estimated over the whole data, e.g., in the case of the *Automobile import* subgroup list of Figure 3.3 it is $\hat{\Theta}^d = \{\hat{\mu} = 13; \hat{\sigma} = 8\}$ and it is estimated over all 197 instances of the dataset.
The general form of a quality measure to be maximized is

$$q(a) = (n_a)^{\alpha} f(\hat{\Theta}^a, \hat{\Theta}^d),\ \alpha \in [0, 1], \tag{2.20}$$

where α allows to control the trade-off between coverage and the difference of the distributions, and $f(\hat{\Theta}^a, \hat{\Theta}^d)$ is a function that measures how different the subgroup and dataset distributions are. As an example, the most commonly adopted quality measure for single-numeric targets is Weighted Relative Accuracy (WRAcc)[70], with $\alpha = 1$ and $f(\hat{\Theta}_a, \hat{\Theta}_d) = \hat{\mu}_a - \hat{\mu}_d$ (the difference between subgroup and dataset averages).

2.6.2 Weighted Kullback-Leibler divergence

Another commonly adopted measure is the Weighted-Kullback Leibler divergence (WKL) [74]. This is also the measure that we consider throughout this dissertation because of its: 1) flexibility in terms of (number and types of) supported target variables; and 2) relationship to the MDL principle (see Chapter 3).
WKL is defined as the Kullback-Leibler (KL) divergence [66] between a subgroup's and dataset target distribution $KL(\hat{\Theta}^a; \hat{\Theta}^d)$ linearly weighted by its coverage. Revisiting Eq. (2.20) this corresponds to $f(.) = KL(.)$ and $\alpha = 1$. The definition of WKL for a univariate target variable Y is given by:

$$WKL(\hat{\Theta}^a; \hat{\Theta}^d) = n_a KL(\hat{\Theta}^a; \hat{\Theta}^d), \tag{2.21}$$

where $KL(\hat{\Theta}^a; \hat{\Theta}^d)$ is the Kullback-Leibler divergence between subgroup and dataset for target Y. The KL divergence in Eq. (2.21) depends on the probabilistic model chosen to describe the target variables. In its general form the KL divergence can be defined as

$$KL(\hat{\Theta}^a_j; \hat{\Theta}^d_j) = \sum_{y \in Y^a} \Pr(y \mid \hat{\Theta}^a_j) \log \left(\frac{\Pr(y \mid \hat{\Theta}^a_j)}{\Pr(y \mid \hat{\Theta}^d_j)} \right), \tag{2.22}$$

where the logarithm is to the base two (like all logarithms used in this thesis). Thus the choice of the distribution used to describe the target is of great importance and should reflect what the user deems interesting. Now, depending of the type of target we will see show how to compute $WKL(\hat{\Theta}^a; \hat{\Theta}^d)$. It is easy to see that for multivariate targets we either use a multivariate distribution, e.g., a multivariate normal distribution, or assume that they are *independent* target variables, where the total WKL turns out to be just the sum the WKL for each target variable.

We will now provide the definitions of WKL for univariate categorical and normal distributions.

Weighted Kullback-Leibler for categorical distributions. In the case of a univariate *nominal target* Y, the distribution can be uniquely described by a categorical distribution with the probability of each category $\hat{\Theta}^a = \{\hat{p}_{1|a}, ..., \hat{p}_{k|a}\}$, so that the $KL(\hat{\Theta}^a; \hat{\Theta}^d)$ of Eq. (2.21) takes the form of

$$KL_{Cat}(\hat{\Theta}^a; \hat{\Theta}^d) = \sum_{c \in \mathcal{Y}} \hat{p}_{c|a} \log \left(\frac{\hat{p}_{c|a}}{\hat{p}_c} \right), \tag{2.23}$$

where $\hat{p}_{c|a} = \Pr(c \mid a)$ is the maximum likelihood estimate of the conditional probability of the target c given the subgroup a, and $\hat{p}_c$ is the marginal probability for that category.

Weighted Kullback-Leibler for normal distributions. In the case of a univariate *numeric* target Y, many distributions could be used for modelling. We resort to the normal distribution for its robustness and analytical properties, as mentioned before. Nonetheless, still two possibilities remain: a location distribution $\hat{\Theta}^a = \{\mu_a\}$ that only accounts for the mean, or a 'complete' normal distribution $\hat{\Theta}^a = \{\mu_a, \sigma_a\}$ that accounts for the mean and the variance. With the location distribution $KL(\hat{\Theta}^a; \hat{\Theta}^d)$ equals[1]:

$$KL_{\mu}(s) = \frac{(\hat{\mu}_d - \hat{\mu}_a)^2}{\hat{\sigma}_d}, \tag{2.24}$$

while with the normal distribution one obtains:

$$KL_{\mu,\sigma}(s) = \left[\log \frac{\hat{\sigma}_d}{\hat{\sigma}_a} + \frac{\hat{\sigma}_a^2 + (\hat{\mu}_a - \hat{\mu}_d)^2}{2\hat{\sigma}_d^2} \log e - \frac{\log e}{2} \right]. \tag{2.25}$$

Note that since $\hat{\sigma}_d$ is a constant for each dataset, there is a strong resemblance between $WKL_{\mu}(s)$ and WRAcc, where the only difference is the square of the difference of the means. Also notice that $WKL_{\mu,\sigma}$ directly takes into account the variance of a subgroup and penalizes for a larger variance, while $WKL_{\mu}(s)$ (and also

[1]The derivations of these formulas can be found in Appendix A.

WRAcc) does not take into account the variance, and thus fail to give importance to the spread of subgroup values. This is a key point as this makes a quality measure like $WKL_{\mu,\sigma}(s)$ *dispersion-aware*, while measures like $WKL_{\mu}(s)$ and $WRAcc$ are not [12].

2.7 Subgroup set discovery measures

Subgroup set discovery [75] is the task of finding a set of high-quality, non-redundant subgroups that together describe all substantial deviations in the target distribution. That is, given a quality function Q for subgroup sets and the set of all possible subgroup sets $\mathcal{S}$, the task is to find that subgroup set $S^* = \{s_1, \ldots, s_k\}$ given by $S^* = \arg\max_{S \in \mathcal{S}} Q(S, \mathbf{Y})$. Note that Q should not only take into account the individual quality of subgroups $q(a)$, but also the overlap of their coverages D^a and quantify the contribution of each instance only once, as opposed to top-k mining where only their individual qualities are taken into account, and thus there is no *global* definition of the quality of a set.

Ideally, a quality measure for subgroup sets Q should: 1) *be global*, i.e., for a given dataset it should be possible to compare subgroup set qualities regardless of subgroup set size or coverage; 2) *maximize the individual qualities* of the subgroups; and 3) *minimize redundancy* of the subgroup set, i.e., the subgroups covers should overlap as little as possible while ensuring the previous point.

Subgroup sets quality has mostly only been defined heuristically, by iteratively finding one subgroup at the time and after each discovery removing/weighting the instances covered by these [71, 75].

Nonetheless, there have been attempts to formally define the quality of a subgroup set, although, they are usually not universal, i.e., independent of the number of subgroups, and usually rely on some heuristic definitions. For example, in Knobbe and Ho [64], the authors propose to first mine the top-k patterns with k very large and then filter out a subset k' according to some measure that takes into account overlap and individual quality. This is an effective approach to find a non-redundant set, but by using top-k in the first step and fixing k', they are first, biasing the search to a certain region of the data that is highly redundant, and second, defining optimality dependent on k'. In van Leeuwen and Ukkonen [76], the authors avoid defining one measure, by treating the problem of finding a diverse set as a multi-objective, and thus using two measures instead of one, one for quality and another for diversity of subgroup set of size k. In the case of Belfodil et al. [10], the authors define a subgroup set as a disjunction of subgroups and based on that, propose a *global* measure

for this type of set. However, this subgroup set definition does not match with previous works, as a disjunction of subgroups is just a subgroup that uses both logical conjunctions and disjunctions to form a description. Thus, each set describes only the global behaviour of this definition of subgroup.

Based on this limitation, in Section 3.6 we propose a new subgroup set measure called the Sum of Weighted Kullback-Leibler (SWKL) divergences that is straightforward to compute for a subgroup list. To extend it to subgroup sets, however, it would require defining the overlap of subgroups distributions in a probabilistic format, such as through a mixture model.