Robust rules for prediction and description

Manuel Proenca, H.

Appendices

A

Kullback-Leibler divergence between two normal distributions

Let us assume two normal probability distributions, $p(x) \sim \mathcal{N}(\mu_p, \sigma_p)$ and $q(x) \sim \mathcal{N}(\mu_q, \sigma_q)$. The Kullback-Leibler divergence of q from p is:

$$\begin{aligned} KL_{\mu,\sigma}(p;q) &= \int_{-\infty}^{+\infty} p(x)\log p(x)\,\mathrm{d}x - \int_{-\infty}^{+\infty} p(x)\log q(x)\,\mathrm{d}x \\ &= \mathbb{E}_p\left[\log p(x)\right] - \mathbb{E}_p\left[\log q(x)\right] \\ &= -\frac{1}{2}\left(\log e + \log 2\pi\sigma_p^2\right) + \frac{1}{2}\log 2\pi\sigma_q^2 + \mathbb{E}_p\left[\frac{(x-\mu_q)^2}{2\sigma_q^2}\log e\right] \\ &= -\frac{\log e}{2} + \log\frac{\sigma_p}{\sigma_q} + \mathbb{E}_p\left[\frac{x^2 - 2x\mu_q + \mu_q^2}{2\sigma_q^2}\log e\right] \\ &= -\frac{\log e}{2} + \log\frac{\sigma_q}{\sigma_p} + \frac{\sigma_p^2 + \mu_p^2 - 2\mu_p\mu_q + \mu_q^2}{2\sigma_q^2}\log e \\ &= -\frac{\log e}{2} + \log\frac{\sigma_q}{\sigma_p} + \frac{\sigma_p^2 + (\mu_p - \mu_q)^2}{2\sigma_q^2}\log e. \end{aligned} \tag{A.1}$$

Note that in the specific case where the Kullback-Leibler divergence only takes into account the means and assumes both standard deviations equal, i.e., $p(x) \sim \mathcal{N}(\mu_p, \sigma)$ and $q(x) \sim \mathcal{N}(\mu_q, \sigma)$ one obtains:

$$KL_{\mu}(p;q) = \frac{(\mu_p - \mu_q)^2}{2\sigma^2}\log e, \tag{A.2}$$

and the weighted version of this KL_μ, i.e., $WKL_\mu = nKL_\mu(p;q)$, is similar to the most common subgroup discovery quality functions used for numeric targets that do

not take into account the dispersion of the subgroup, such as the weighted relative accuracy or the mean-test [75], which uses the square root of KL_{μ}. We will call this measure the Weighted Kullback-Leibler without dispersion.

B

Prequential plug-in encoding for rule lists with categorical distributions

For this section, let us assume that we have a dataset $D = \{\mathbf{X}, Y\}$, Y has $k = |\mathcal{Y}|$ class labels and a model M that forms a partition over the whole data. The model M divides the data D in ω parts, of the form $\{(\mathbf{X}^1, Y^1), \cdots, (\mathbf{X}^\omega, Y^\omega)\}$. Each part has an associated categorical distribution with estimated parameters $\hat{\Theta}^i$ over the target part Y^i (as defined in Section 2.4).

Before introducing the *prequential plug-in* code it is necessary to introduce one main building block, the smoothed maximum likelihood estimator for a subset i:

$$\hat{p}_{c|i} = \frac{n_{c|i} + \epsilon}{n_i + |\mathcal{Y}|\epsilon}. \tag{B.1}$$

Unlike the regular maximum likelihood estimator, this smoothed variant—known as Laplace smoothing—adds a (small) pseudocount ϵ to each class-specific usage even when that class has no counts. This avoids zero probabilities for any class label and corresponds in Bayesian statistics to using a symmetric Dirichlet prior ϵ for each class [42].

Now, the main idea of the *prequential plug-in* code is to sequentially predict the points in a subset, starting with no knowledge about their distribution and updating it each time it receives a point using the Equation (B.1). 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 on the

probabilities [48].
Applying this idea to encode the class labels in Y and ignoring the data partition at the moment, initially each class label has a pseudocount of ϵ. Hence, when sending the first class label, y^1, we effectively use a uniform code, i.e., $-\log \frac{\epsilon}{k\epsilon}$. After that, however, we increase the count of that class label by one. Normalizing the updated counts results in a new categorical probability distribution—hence a new code: $-\log \frac{\epsilon+1}{k\epsilon+1}$. This code is the *best possible code given the data seen so far* and is equal to the smoothed maximum likelihood of Eq. (B.1). Formally, the plug-in code for encoding the class labels is defined as

$$\Pr_{\text{plug-in}}(y^u = c \mid Y^{|u-1}) \coloneqq \frac{|\{y \in Y^{|u-1} \mid y = c\}| + \epsilon}{\sum_{c' \in \mathcal{Y}} |\{y \in Y^{|u-1} \mid y = c'\}| + \epsilon}, \tag{B.2}$$

where $u \in \mathbb{N}$, y^u represents the u^{th} class label in Y, $Y^{|u-1} = \{y^1, ..., y^{u-1}\}$ represents the sequence of the $u-1$ first class labels, and ϵ is the pseudocount necessary for $\Pr_{\text{plug-in}}(y^1 = c \mid Y^{|0}) = \epsilon/k\epsilon = 1/k$ to be valid. 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 to obtain natural factorials instead of gamma functions as can be seen next.
We now show how this prequential plug-in code can be used in the encoding of the class labels of a dataset partitioned in ω parts. But assuming no interaction between the parts, the total encoding is equal to the sum of its parts:

$$L_{\text{plug-in}}(Y \mid \mathbf{X}, M) = -\log \prod_{i}^{\omega} \Pr_{\text{plug-in}}(Y^i) = \sum_{i}^{\omega} L_{\text{plug-in}}(Y^i), \tag{B.3}$$

where $L_{\text{plug-in}}(Y^i) = -\log \Pr_{\text{plug-in}}(Y^i)$.
Inserting the prequential plug-in code (B.2) in (B.3) we obtain for each part Y^i:

$$\begin{aligned} L_{\text{plug-in}}(Y^i) &= -\log \left(\prod_{u=1}^{n_i} \Pr_{\text{plug-in}}(y^u \mid Y^{i|u-1}) \right) \\ &= -\log \left(\frac{\prod_{c=1}^{k} \prod_{u=0}^{n_{c|i}-1} (u+\epsilon)}{\prod_{u=0}^{n_i-1} (u+k\epsilon)} \right) \\ &= -\log \left(\frac{\prod_{c=1}^{k} (n_{c|i}-1+\epsilon)!/(\epsilon-1)!}{(n_i-1+k\epsilon)!/(k\epsilon-1)!} \right) \\ &= -\log \left(\frac{\prod_{c=1}^{k} \Gamma(n_{c|i}+\epsilon)/\Gamma(\epsilon)}{\Gamma(n_i+k\epsilon)/\Gamma(k\epsilon)} \right), \end{aligned} \tag{B.4}$$

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 $\Gamma(u) = (u-1)!$.

This code starts from sequential data, but as one can see in Eq. (B.4), the order in which one transmits class labels does not matter. In the end, the formulation is order agnostic and only depends on the counts per class label.

C

Normalized Maximum Likelihood for rule lists with categorical distributions

For this section, let us assume that we have a dataset $D = \{\mathbf{X}, Y\}$ and model M that forms a partition over the whole data. Model M divides the data D in ω parts, of the form $\{(\mathbf{X}^1, Y^1), \cdots, (\mathbf{X}^\omega, Y^\omega)\}$. Each part has an associated categorical distribution with estimated parameters $\hat{\Theta}^i$ over the target part Y^i (as defined in Chapter 2.4).
Here we show that the NML encoding of a partition equals the sum of the NML encoding of its parts:

$$L_{NML}(Y \mid \mathbf{X}, M) = \sum_{i=1}^{\omega} L_{NML}(Y^i). \tag{C.1}$$

Note that in the case of a subgroup list, as the default rule does not require NML encoding, the M used in this section represents the subgroups S, and D represents the data covered by these. In the case of a tree or rule list, M represents the model that partitions the data at the leaves and rules (including default rule), respectively, and D the whole dataset. There is no loss of generality for subgroup lists as the separation property allows us to separate the encoding of the default rule.
First, lets recall the definition of the NML probability distribution [115]:

$$L_{NML}(Y \mid \mathbf{X}, M) = -\log\left(\frac{\Pr(Y \mid \mathbf{X}; \hat{M}(Y \mid \mathbf{X}))}{\sum_{Z \in \mathcal{Y}^n} \Pr(Z \mid \mathbf{X}; \hat{M}(Z \mid \mathbf{X}))}\right),$$

where $\mathcal{Y}^n$ is the set of all possible sequences of n points with $k = |\mathcal{Y}|$ categories, $\hat{M}(Y \mid \mathbf{X})$ and $\hat{M}(Z \mid \mathbf{X})$ are the models with parameters estimated according to the maximum likelihood over the data Y and Z, respectively. Taking into account that

our data is independent and identically distributed (*i.i.d.*), and that our model M partitions the data into ω parts, we can further develop the previous formula to:

$$\begin{aligned} L_{NML}(Y \mid \mathbf{X}, M) &\overset{\text{i.i.d.}}{=} -\log\left(\frac{\prod_{i=1}^{n}\Pr(y^i \mid \mathbf{x}^i; \hat{M}(Y \mid \mathbf{X}))}{\sum_{Z\in\mathcal{Y}^n}\prod_{i=1}^{n}\Pr(z^i \mid \mathbf{x}^i; \hat{M}(Z \mid X))}\right) \\ &= -\log\left(\frac{\prod_{i'=1}^{\omega}\Pr(Y^{i'}; \hat{\Theta}(Y^{i'}))}{\sum_{Z\in\mathcal{Y}^n}\prod_{i'=1}^{\omega}\Pr(Z^{i'}; \hat{\Theta}(Z^{i'}))}\right) \\ &= -\log\left(\frac{\prod_{i'=1}^{\omega} l(\hat{\Theta}^{i'} \mid Y^{i'})}{g(Y, X, M)}\right) \\ &= -\log\left(\sum_{i'=1}^{\omega} l(\hat{\Theta}^{i'} \mid Y^{i'})\right) + \log g(Y, X, M), \end{aligned} \tag{C.2}$$

where $l(\hat{\Theta}^{i'} \mid Y^{i'})$ is the likelihood function for each of the ω parts and $g(Y, X, M)$ is a complexity function that depends on these three variables.
The first term is already independent for each part, although the second is not.
Let us now look at $g(Y, X, M)$ when we only have one part in the dataset, i.e., D^1. We will call this term the NML complexity of a multinomial distribution and denote it by $\mathcal{C}(n_1, k)$ of one part $D^1 = \{Y^1, X^1\}$, with $n_1 = |D^1|$ and $k = \mathcal{Y}$

$$\begin{aligned} \mathcal{C}(n_1, k) &= \log\left(\sum_{Z\in\mathcal{Y}^{n_1}}\Pr(Z^1; \hat{\Theta}(Z^1))\right) \\ &= \log\left(\sum_{Z\in\mathcal{Y}^{n_1}}\prod_{i=1}^{n_1}\Pr(z^i; \hat{\Theta}(Z^1))\right) \\ &= \log\left(\sum_{n_{1|1}+n_{2|1}+\ldots+n_{k|1}=n_1}\frac{n_1!}{n_{1|1}!n_{2|1}!\ldots n_{k|1}!}\prod_{c\in\mathcal{Y}}\left(\frac{n_{c|1}}{n_1}\right)^{n_{c|1}}\right) \end{aligned} \tag{C.3}$$

where $n_{c|1}$ is the number of points of category c in Y^1, and the passage from the second equality to the last is a property of multinomial distributions commonly used to make the computation of $\mathcal{C}(n_a, k)$ simpler [48]. It is interesting to note that $\mathcal{C}(n_a, k)$ only depends on the number of points in Y^1 and its cardinality, not on the actual values. This term, i.e., the complexity of a multinomial distribution over n_1 points with k possible values, measures the likelihood of each possible sequence.
Now we must generalize from a part to the partition of the dataset. To illustrate how to do this, let us first look at Table C.1, which shows an example of all the possible sequences in a fixed-length three-part partition of the data. Taking into account those

Table C.1: All possible sequences of a partition of fixed length of the data in three parts. Fixed length means that all possible parts always have the same amount of points, as e.g. $|A_1| = |A_2| = \cdots = |A_a| = n_A$.

Part 1	Part 2	Part 3
A_1	B_1	C_1
A_1	B_1	C_2
$\vdots$	$\vdots$	$\vdots$
A_1	B_2	C_1
$\vdots$	$\vdots$	$\vdots$
A_a	B_b	C_c

three parts, let us look at how the probabilities of all those sequences could be computed:

$$\begin{aligned}\sum_{\forall a,b,c} \Pr(A_a)\Pr(B_b)\Pr(C_c) &= \left(\sum_{\forall a}\Pr(A_a)\right)\cdot\left(\sum_{\forall b,c}\Pr(B_b)\Pr(C_c)\right)\\ &= \left(\sum_{\forall a}\Pr(A_a)\right)\cdot\left(\sum_{\forall b}\Pr(B_b)\right)\cdot\left(\sum_{\forall c}\Pr(C_c)\right),\end{aligned}$$

where this follows naturally from the distributive property of the multiplication. It is easy to see that this generalizes to partitions of any number of parts. Thus, going back to the complexity term $g(Y, X, M)$, we can see that

$$\begin{aligned}\log g(Y,X,M) &= \log \sum_{Z\in\mathcal{Y}^n}\prod_{i'=1}^{\omega}\Pr(Z^{i'};\hat{\Theta}(Z^{i'}))\\ &= \log\prod_{i'=1}^{\omega}\sum_{Z^{i'}\in\mathcal{Y}^{n_{i'}}}\Pr(Z^{i'};\hat{\Theta}(Z^{i'}))\\ &= \sum_{i'=1}^{\omega}\log\sum_{Z^{i'}\in\mathcal{Y}^{n_{i'}}}\Pr(Z^{i'};\hat{\Theta}(Z^{i'}))\\ &= \sum_{i'=1}^{\omega}\log\mathcal{C}(n_{i'},k)\end{aligned} \tag{C.4}$$

Substituting this back into Eq. (C.2), we obtain what we wanted:

$$\begin{aligned} L_{NML}(Y \mid \mathbf{X}, M) &= -\log\left(\sum_{i=1}^{\omega} l(\hat{\Theta}^i \mid Y^i)\right) + \sum_{i=1}^{\omega} \log \mathcal{C}(n_i, k) \\ &= \sum_{i=1}^{\omega} l(\hat{\Theta}^i \mid Y^i) + \mathcal{C}(n_i, k) \\ &= \sum_{i=1}^{\omega} L_{NML}(Y^i) \end{aligned} \tag{C.5}$$

D

Bayesian encoding of a normal distribution with mean and standard deviation unknown

For encoding a sequence of numeric valued i.i.d. observations such as $Y = \{y_1, ..., y_n\}$, the Bayesian encoding takes the following form:

$$P_{Bayes}(Y) = \int_{\Theta} f(Y \mid \Theta) w(\Theta) \, \mathrm{d}\Theta, \tag{D.1}$$

where f is the probability density function (pdf), Θ is the set of parameters of the distribution, and $w(\Theta)$ the prior over the parameters. In the case of a normal distribution $\Theta = \{\mu, \sigma\}$, with μ and σ being its mean and standard deviation, respectively, the pdf $f(Y \mid \Theta)$ over a sequence Y is the multiplication of the individual pdfs, thus:

$$f(Y \mid \mu, \sigma) = \frac{1}{(2\pi)^{n/2}\sigma^n} \exp\left[-\frac{1}{2\sigma^2}\sum_{i}^{n}(y^i - \mu)^2\right], \tag{D.2}$$

In order not to bias the encoding for specific values of the parameters, we choose to use the constant Jeffrey's prior of $1/\sigma^2$ for the unknown parameters μ and σ, and add an extra. Thus, our prior is given by:

$$w(\mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma^2}, \tag{D.3}$$

where $1/\sqrt{2\pi}$ was added for normalization reasons.

Putting everything together, one obtains:

$$P_{Bayes}(Y) = \\ = (2\pi)^{-\frac{n+1}{2}} \int_{-\infty}^{+\infty} \int_{0}^{+\infty} \frac{1}{\sigma^{n+2}} \exp\left[-\frac{1}{2\sigma^2}\left(\sum_{i}^{n}(y^i-\mu)^2\right)\right] \mathrm{d}\sigma\, \mathrm{d}\mu. \tag{D.4}$$

The integrals over the whole space of the parameters μ and σ allow to penalize the fact that we do not know the statistics *a priori*, thus penalizing the fact that a distribution over n points could, by chance, have the same statistics as the one found in the data.

Note that using an improper prior requires that we somehow make it proper, i.e., we need to find a way to make the integration over the prior finite $\int\int w(\mu,\sigma) = K$, where K is a constant value. The usual way to make an improper prior finite is to condition on the k minimum number observations $Y^{|k} \in Y$ needed to make the integral proper [48], which in the case of two unknowns (μ and σ) is $k = 2$. Thus, instead of using $w(\mu,\sigma)$ we will in practice be using $w(\mu,\sigma \mid Y^{|2})$, and using the the chain rule and the Bayesian formula returns a total encoding of Y equal to

$$P(Y) = P_{Bayes}(Y \mid Y^{|2})P(Y^{|2}) = \frac{P_{Bayes}(Y)}{P_{Bayes}(Y^{|2})}P(Y^{|2}) \tag{D.5}$$

where $P(Y^{|2})$ is a non-optimal probability used to define $Y^{|2} = \{y^1, y^2\}$ that we will define later and y^1, y^2 chosen in a way that maximizes $P(Y)$. Now that we have all the ingredients to define $P(Y)$ we will start by defining $P_{Bayes}(Y)$ and then choose the appropriate probability for $P(Y^{|2})$.

To solve the first integral of $P_{Bayes}(Y)$ in Eq. (D.4), we integrate in σ and note that the formula is an instance of the gamma function,

$$\Gamma(k) = \int_{0}^{+\infty} z^{k-1}e^{-z}\, \mathrm{d}z, \tag{D.6}$$

with the corresponding variable transformation:

$$z = \frac{A}{2\sigma^2};\ \frac{1}{\sigma} = \frac{2^{1/2}z^{1/2}}{A^{1/2}};\ \mathrm{d}\sigma = -\frac{\sigma}{2z}\,\mathrm{d}z;\ A = \left[\sum_{i}^{n}(y^i-\mu)^2\right], \tag{D.7}$$

Performing the variable transformation and noting that the minus sign of $\mathrm{d}z$ cancels with the reversing of the integral limits, we get:

$$P_{Bayes}(Y) = \\ = \Gamma\left(\frac{n+1}{2}\right) 2^{\frac{n+1}{2}-1}(2\pi)^{-\frac{n+1}{2}} \int_{-\infty}^{+\infty} \left[\sum_{i}^{n}(y^i-\mu)^2\right]^{-\frac{n+1}{2}} \mathrm{d}\mu, \tag{D.8}$$

which reveals that the prior on the effect size ρ, and specifically its standard deviation parameter τ, is equivalent to adding $1/\tau^2$ virtual points to the original data.

To solve the integral in μ we need to introduce the statistics $\hat{\mu}$ and $\hat{\sigma}$ as the values estimated from the data. We define these quantities as:

$$\hat{\mu} = \frac{1}{n}\sum_i^n y^i;\ \hat{\sigma}^2 = \frac{1}{n}\sum_i^n (y^i - \hat{\mu})^2\ , \tag{D.9}$$

where $\hat{\mu}$ is the mean estimator over n data points and $\hat{\sigma}^2$ is the estimator of the variance. Note that for the variance the biased version with n was used instead of with $n-1$ as it allows to compute the Residual Sum of Squares (RSS) directly by $RSS = n\hat{\sigma}$.

Focusing now on the interior part of the integral of Eq. D.8 and rewriting it in order to resemble the t-student distribution, we obtain:

$$\begin{aligned}
&\left[\sum_i^n (y^i - \mu)^2\right]^{-(n+1)/2} = \\
&\left[\sum_i^n (y^i)^2 - n\hat{\mu}^2 + n\hat{\mu}^2 - 2n\hat{\mu}\mu + n\mu^2\right]^{-(n+1)/2} = \\
&\left[\sum_i^n (y^i)^2 - n\hat{\mu}^2 + n(\hat{\mu} - \mu)^2\right]^{-(n+1)/2} = \\
&\left[n\hat{\sigma'}^2 + n(\hat{\mu}' - \mu)^2\right]^{-(n+1)/2} = \\
&\left[n\hat{\sigma}^2\right]^{-(n+1)/2}\left[1 + \frac{(\hat{\mu} - \mu)^2}{\hat{\sigma}^2}\right]^{-(n+1)/2} \\
&\left[n\hat{\sigma}^2\right]^{-(n+1)/2}\left[1 + \frac{1}{n}\left(\frac{\hat{\mu} - \mu}{s_s^2}\right)^2\right]^{-(n+1)/2},
\end{aligned} \tag{D.10}$$

where $s_s^2 = \hat{\sigma}^2/n$ is the "sampling" variance. Now, taking into account the fact that the integral of the t-student distribution over the whole space is equal to one, and reshuffling around its terms we get

$$\int_{-\infty}^{+\infty}\left[1 + \frac{1}{n}\left(\frac{\hat{\mu} - \mu}{s_s}\right)^2\right]^{-\frac{n+1}{2}} \mathrm{d}\mu = \frac{\Gamma\left(\frac{n}{2}\right)\sqrt{\pi n}s_s}{\Gamma\left(\frac{n+1}{2}\right)}. \tag{D.11}$$

Inserting this back in Eq. D.4 we obtain:

$$\begin{aligned} &P_{Bayes}(Y) = \\ &= \Gamma\left(\frac{n+1}{2}\right) 2^{\frac{n+1}{2}-1}(2\pi)^{-\frac{n+1}{2}} \frac{\Gamma(\frac{n}{2})\sqrt{\pi n}s_s}{\Gamma(\frac{n+1}{2})} \left[n\hat{\sigma}^2\right]^{-(n+1)/2} \\ &= 2^{-1}\pi^{-\frac{n}{2}}\Gamma\left(\frac{n}{2}\right)\frac{1}{\sqrt{n}}\left[n\hat{\sigma}^2\right]^{-\frac{n}{2}}, \end{aligned} \tag{D.12}$$

Returning to the the conditional probability of Eq. (D.5), we see that we still need to define $P(Y^{|2})$, the non-optimal probability of the first two-points. As in the case of our model class we assume that the dataset overall statistics are known, i.e., $\Theta = \{\hat{\mu}_d, \hat{\sigma}_d\}$, we will use this distribution to find the probability of the points $Y^{|2} = \{y^1, y^2\}$ as :

$$P(Y^{|2}) = \log 2\pi + \log \hat{\sigma}_d + \left[\frac{1}{2\hat{\sigma}_d^2}\sum_i^2 (y^i - \hat{\mu}_d)^2\right]\log e. \tag{D.13}$$

Finally, applying the minus logarithm base 2 to all the terms in Eq (D.5) to obtain the total code length in bits,

$$\begin{aligned} &L_{Bayes2.0}(Y) = -\log P_{Bayes}(Y) + \log P_{Bayes}(Y^{|2}) - \log P(Y^{|2}) \\ &= 1 + \frac{n}{2}\log\pi - \log\Gamma\left(\frac{n}{2}\right) + \frac{1}{2}\log n + \frac{n}{2}\log\left(n\hat{\sigma}_n^2\right) \\ &- 1 - \frac{2}{2}\log\pi + 0 - \frac{1}{2} - \log\left(\sum_i^2 (y^i - \hat{\mu}_2)^2\right) \\ &+ \frac{2}{2}\log\pi + \log\hat{\sigma}_d + \left[\frac{1}{2\hat{\sigma}_d^2}\sum_i^2 (y^i - \hat{\mu}_d)^2\right]\log e \\ &= \frac{n}{2}\log\pi - \log\Gamma\left(\frac{n}{2}\right) + \frac{1}{2}\log n + \frac{n}{2}\log\left(n\hat{\sigma}_n^2\right) + L_{cost}(Y^{|2}), \end{aligned} \tag{D.14}$$

where $\hat{\mu}_2$ is the estimated mean of y^1, y^2 and $L_{cost}(Y^{|2})$ is the extra cost incurred of not being able to use a refined encoding for $Y^{|2}$. Now that the length of the encoding is defined, we just need to choose the two points. i.e., y^1, y^2. Because we want to minimize this length, we notice that there are only two terms that contribute to it in $L_{cost}(Y^{|2})$, and thus by choosing the two observations close to $\hat{\mu}_d$ minimizes both the encoding of $P(Y^{|2})$ and maximize $P_{Bayes}(Y^{|2})$ for most cases. There are exceptions to this, depending on the respective values of μ_d and y^1, y^2 but these are not significant to change the values too much and also requires less computational search to find the points.

E

Bayesian encoding convergence to BIC for large n

This section shows that for a large number of instances n, the Bayesian encoding of Appendix D converges to the Bayesian Information Criterion (BIC). Thus, Eq. (D.14)) converges to the encoding of a normal distribution with mean and standard deviation known plus $\log n$. First, the encoding of a normal distribution with mean and standard deviation known over n *i.i.d.* points is equal to the sum of the individual encodings:

$$L(Y \mid \hat{\Theta}) = \frac{n}{2}\log 2\pi + \frac{n}{2}\log \hat{\sigma}^2 + \left[\frac{1}{2\hat{\sigma}^2}\sum_{i}^{n}(y^i - \hat{\mu})^2\right]\log e. \tag{E.1}$$

Second, we need to use the Stirling's approximation of the Gamma function for large n:

$$\begin{aligned} &-\log\Gamma\left(\frac{n}{2}\right) \\ &\sim -\frac{1}{2}\log\pi - \frac{1}{2}\log\left(n-2\right) - \left(\frac{n}{2}-1\right)\log\left(\frac{n}{2}-1\right) + \left(\frac{n}{2}-1\right)\log e, \end{aligned} \tag{E.2}$$

and finally we insert it into Eq. (D.14) and assume $\tau = 1$ to obtain:

$$
\begin{aligned}
& L(Y) \sim \\
& \sim 1 + \frac{n-1}{2}\log\pi + \frac{1}{2}\log\left(\frac{n}{n-2}\right) + \frac{n}{2}\log\left(\frac{n\hat{\sigma}^2}{n/2-1}\right) + \left(\frac{n}{2}-1\right)\log e \\
& + \log\left(\frac{n}{2}-1\right) + L_{cost}(Y^{|2}) \\
& \sim \frac{n}{2}\log\pi + \frac{n}{2}\log 2\hat{\sigma}^2 + \left[\frac{1}{2\hat{\sigma}^2}\sum_i^n (y^i-\mu)^2\right]\log e + \log n - \log e + L_{cost}(Y^{|2}) \\
& = L(Y \mid \hat{\Theta}) + \log\frac{n}{e} + L_{cost}(Y^{|2}) \\
& \sim \frac{1}{2}\left(2L(Y \mid \hat{\Theta}) + 2\log n - 2\log e\right) \\
& = \frac{1}{2}BIC,
\end{aligned}
\tag{E.3}
$$

where from the second to the third line, we assumed large n, making some of the terms disappear, while the definition $n\hat{\sigma}^2 = \sum_i^n (y^i - \mu)^2$ is used for making the third term of the third expression appear. From the fourth to the fifth expressions, it assumes that $L_{cost}(Y^{|2})$ is negligible, as it is the cost of not being able to encode the first two points optimally. For the Bayes information criterion, we used its standard definition,

$$BIC = -2\ln\ell(\Theta \mid Y) + k\ln n, \tag{E.4}$$

where $\ell(\Theta \mid Y)$ is the likelihood as estimated from the data, and k is the number of parameters, which in our case is 2.

F

Datasets used for classification experiments

The 17 datasets used for classification are shown in Table F.1, and were retrieved from LUCS/KDD[1] repository. The datasets all have *binary* explanatory variables.

[1] http://cgi.csc.liv.ac.uk/~frans/KDD/Software/LUCS-KDD-DN/DataSets/dataSets.html

Table F.1: Dataset properties: number of {samples, binary variables, classes, average number of candidate patterns per fold for CLASSY with $n_{min.} = 5\%$ and $d_{max} = 4$}. The datasets are ordered first by number of classes and then by the number of samples.

Dataset	$\|D\|$	$\|V\|$	$\|\mathcal{Y}\|$	$\|Cands\|$
hepatitis	155	48	2	39137
ionosphere	351	155	2	332560
horsecolic	368	81	2	23552
cylBands	540	120	2	304749
breast	699	14	2	299
pima	768	34	2	543
tictactoe	958	26	2	1907
mushroom	8124	84	2	79602
adult	48842	96	2	7231
iris	150	14	3	144
wine	178	63	3	13439
waveform	5000	96	3	86889
heart	303	46	5	21876
pageblocks	5473	39	5	2902
led7	3200	22	10	2507
pendigits	10992	81	10	107001
chessbig	28056	54	18	1384

G RSD supplementary empirical evaluation

G.1 Datasets used for subgroup discovery experiments

The datasets selected are commonly used in machine learning and subgroup discovery, and were retrieved from UCI [29], Keel [4], MULAN [117] repositories. The datasets for nominal and numeric targets experiments are in Table G.1 and G.2, respectively.

Table G.1: Nominal targets datasets for subgroup discovery: single-binary, single-nominal and multi-label. Dataset properties: number of {target variables T; target labels $|\mathcal{Y}|$; samples $|D|$; type of variables (nominal/numeric)}.

Dataset	T	$\lvert\mathcal{Y}\rvert$	$\lvert D\rvert$	$V(nom./num.)$
sonar	1	2	208	(0/60)
haberman	1	2	306	(0/3)
breastCancer	1	2	683	(0/9)
australian	1	2	690	(0/14)
TicTacToe	1	2	958	(9/0)
german	1	2	1 000	(13/7)
chess	1	2	3 196	(36/0)
mushrooms	1	2	8 124	(22/0)
magic	1	2	19 020	(0/10)
adult	1	2	45 222	(8/6)
iris	1	3	150	(0/4)
balance	1	3	625	(0/4)
CMC	1	3	1 473	(0/9)
page-blocks	1	5	5 472	(0/10)
nursery	1	5	12 960	(7/1)
automobile	1	6	159	(10/15)
glass	1	6	214	(0/10)
dermatology	1	6	358	(0/34)
kr-vs-k	1	18	28 056	(6/0)
abalone	1	28	4 174	(1/7)
emotions	6	2	593	(0/72)
scene	6	2	2407	(0/294)
flags	7	2	194	(9/10)
yeast	14	2	2417	(0/103)
birds	19	2	645	(/258)
genbase	27	2	662	(1186/0)
mediamill	101	2	43 907	(0/120)
CAL500	174	2	502	(0/68)
Corel5k	374	2	5000	(499/0)

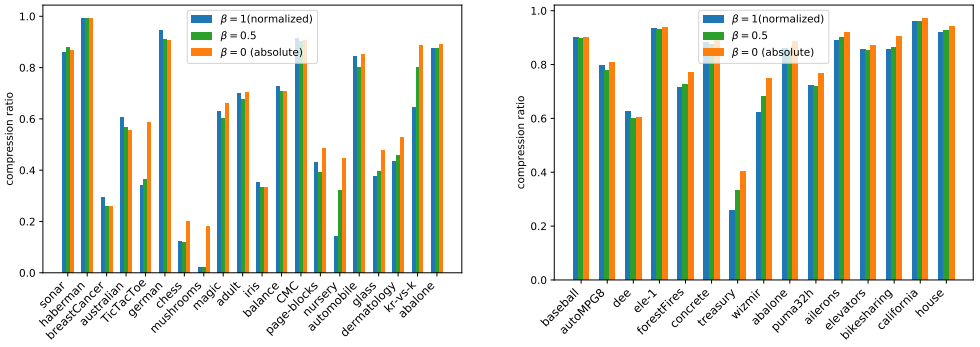
Table G.2: Numeric targets datasets for subgroup discovery: single-numeric and multi-numeric. Dataset properties: {number of target variables T; minimum and maximum target values $[min., max.]$; number of samples $|D|$; number of type of variables (nominal/numeric)}.

Dataset	T	$[min.; max.]$	$\|D\|$	$V(nom./num.)$
baseball	1	$[109; 6100]$	337	$(4/12)$
autoMPG8	1	$[9; 46.6]$	392	$(0/6)$
dee	1	$[0.8; 5.1]$	365	$(0/6)$
ele-1	1	$[80; 7675]$	495	$(0/2)$
forestFires	1	$[0; 1091]$	517	$(0/12)$
concrete	1	$[3; 21]$	1030	$(0/8)$
treasury	1	$[29; 90]$	1049	$(0/15)$
wizmir	1	$[29; 90]$	1461	$(0/9)$
abalone	1	$[1; 29]$	4177	$(0/8)$
puma32h	1	$[-0.0867; 0.0898]$	8192	$(0/32)$
ailerons	1	$[-0.0036; 0]$	13750	$(0/40)$
elevators	1	$[0.012; 0.078]$	16599	$(0/18)$
bikesharing	1	$[1; 977]$	17379	$(2/10)$
california	1	$[14999; 500001]$	20640	$(0/8)$
house	1	$[0; 500001]$	22784	$(0/16)$
edm	2	$[-1; 1]$	154	$(0/16)$
enb	2	$[6.01; 48.03]$	768	$(0/8)$
slump	3	$[0; 78]$	103	$(0/7)$
sf1	3	$[0; 4]$	323	$(0/10)$
sf2	3	$[0; 8]$	1066	$(0/10)$
jura	3	$[0.135; 166.4]$	359	$(0/15)$
osales	12	$[500; 795000]$	639	$(0/413)$
wq	14	$[0; 5]$	1060	$(0/16)$
oes97	16	$[30; 48890]$	334	$(0/263)$
oes10	16	$[30; 64560]$	403	$(0/298)$

G.2 Analysis of RSD compression gain hyperparameter

In this section, we present a thorough comparison of the normalization term β of RSD, where $\beta = 1$ is the *normalized* gain and $\beta = 0$ the *absolute* gain. RSD is executed with the same hyperparameters (beam width, number of cut points for numerical variables, and maximum depth of search) as in the experiments section, i.e., $w_b = 100$, $n_{cut} = 5$, $d_{max} = 5$. The different types of gain are compared for all the benchmark datasets described in the paper in terms of their compression ratio (defined later) in Figure G.1, Sum of Weighted Kullback-Leibler divergency (SWKL) in Figure G.2, and number of rules in Figure G.3. The compression ratio is the length of the found model $L(D, M)$ divided by the length of encoding the data with the dataset distribution (a model without subroups) $L(D \mid \hat{\Theta}^d)$

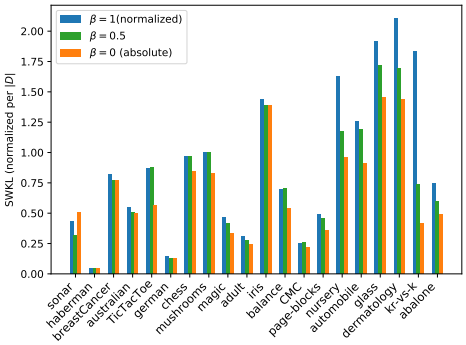
$$L\% = \frac{L(D, M)}{L(D \mid \hat{\Theta}^d)} \tag{G.1}$$



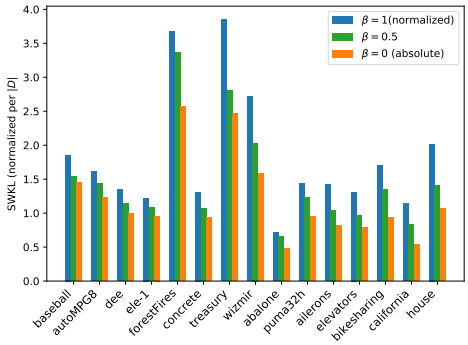
(a) Univariate nominal target.

(b) Univariate numeric target.

Figure G.1: Compression ratio obtained with $\beta = 0$ (absolute gain), $\beta = 0.5$, and $\beta = 1$ (normalized gain).

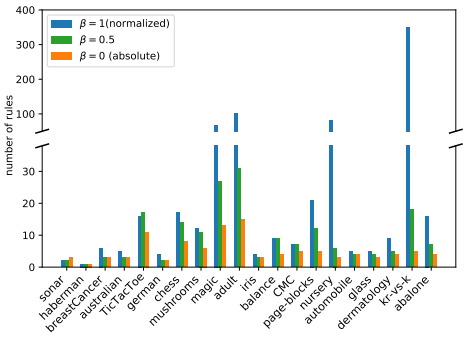


(a) Univariate nominal target.

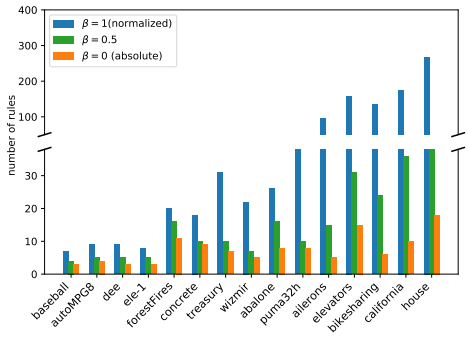


(b) Univariate numeric target.

Figure G.2: Normalized SWKL obtained with $\beta = 0$ (absolute gain), $\beta = 0.5$, and $\beta = 1$ (normalized gain).



(a) Univariate nominal target.



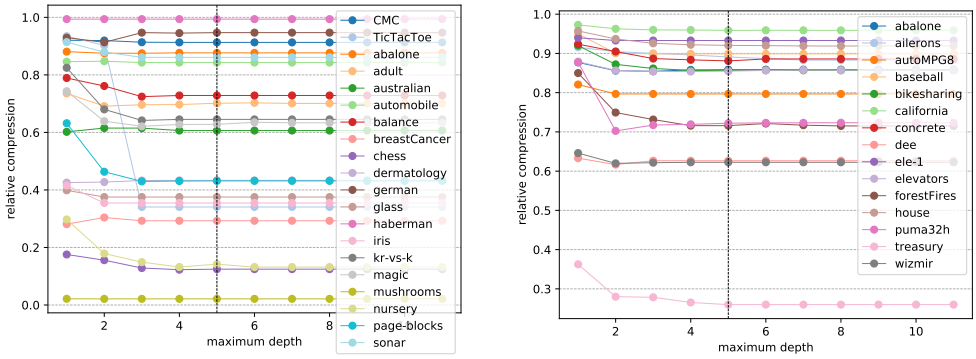
(b) Univariate numeric target.

Figure G.3: Number subgroups obtained with $\beta = 0$ (absolute gain), $\beta = 0.5$, and $\beta = 1$ (normalized gain).

G.3 Analysis of RSD beam search hyperparameters

In this section, we present a thorough comparison of the beam search hyperparameters influence on RSD output. As a complete search over the whole combination of hyperparameters is unfeasible, we present here an exploration over the hyperparameters used for the experimental comparison in the paper ($w_b = 100$, $n_{cut} = 5$, $d_{max} = 5$), i.e., we fix two of the parameters on the aforementioned values and then proceed to change the selected hyperparameter of interest, and we do this for all the 3 parameters. The line between the dots of the same colour does not represent an interpolation and is merely used to aid visualization and suggest trends.

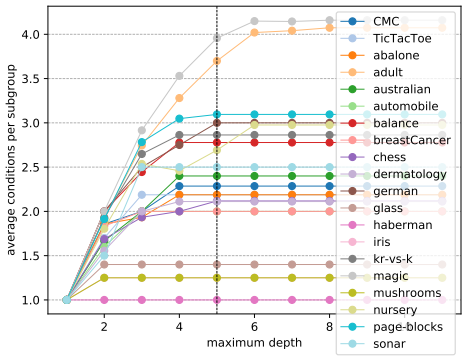
Note on relative compression. It may seem that the values of the relative compression remain constant but that is an illusion due to the scale of the y axis. As the compression ratio is given by the division of large values (usually above the thousands) its value with two decimal digits can be misleading. Nonetheless, in general, when zooming over the figures one can discern a slight improvement (smaller values) for larger values of the hyperparameters.



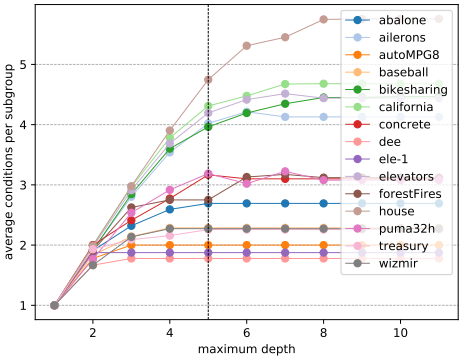
(a) Univariate nominal target.

(b) Univariate numeric target.

Figure G.4: Compression ratio obtained by varying the maximum search depth fixing $w_b = 100$, $n_{cut} = 5$ and $\beta = 1$ (normalized gain). The black vertical line represents the value used in the experiments section for subgroup lists (Section 5.3).

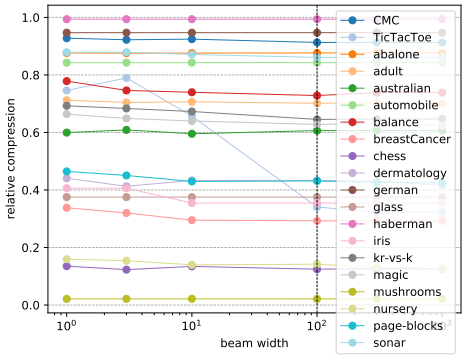


(a) Univariate nominal target.

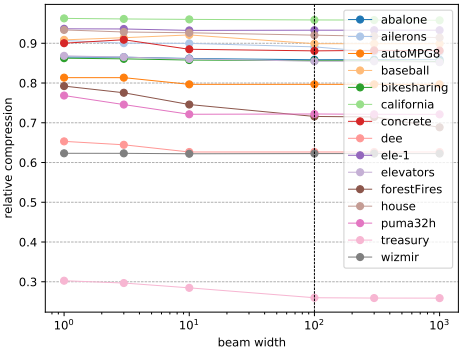


(b) Univariate numeric target.

Figure G.5: Average number of conditions per subgroup obtained by varying the maximum search depth fixing $w_b = 100$, $n_{cut} = 5$ and $\beta = 1$ (normalized gain). The black vertical line represents the value used in the experiments section for subgroup lists (Section 5.3).

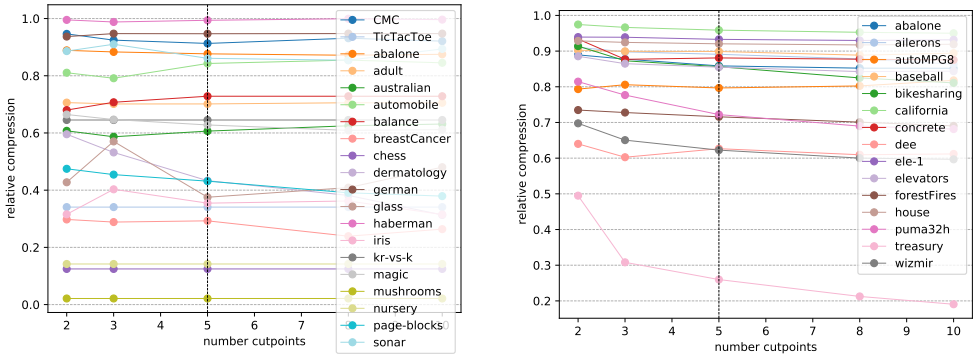


(a) Univariate nominal target.



(b) Univariate numeric target.

Figure G.6: Compression ratio obtained by varying the beam width and fixing $d_{max} = 5$, $n_{cut} = 5$ and $\beta = 1$ (normalized gain). The black vertical line represents the value used in the experiments section for subgroup lists (Section 5.3).



(a) Univariate nominal target.

(b) Univariate numeric target.

Figure G.7: Compression ratio obtained by varying the number of cut points and fixing $w_b = 100$, $d_{max} = 5$ and $\beta = 1$ (normalized gain). The black vertical line represents the value used in the experiments section for subgroup lists (Section 5.3).

G.4 Results of non-sequential subgroup set discovery algorithms

The comparison of RSD with subgroup set discovery algorithms that return sets (and not lists) can be seen in Table G.3.

Table G.3: Single nominal target results for non-sequential methods plus RSD. This includes single-binary, single-nominal, respectively separated by an horizontal line in the table. The properties of the datasets can be seen in Table G.1, and are ordered by number target variables, number of classes, and number of samples, in this order. The evaluation measures are {quality of the subgroup set swkl; number of subgroups $|S|$; and average number of conditions $|a|$}. Note that FSSD does not work for single-nominal case and MCTS4DM only works for datasets with the same type of explanatory variables and thus the empty values $-$. *as DSSD has as stopping criteria the maximum number of subgroups was selected as the number of subgroups found by RSD, and total overlapping subgroups were posteriorly removed.

	DSSD			MCTS4DM			FSSD			RSD		
datasets	swkl	$\|S\|*$	$\|a\|$	swkl	$\|S\|$	$\|a\|$	swkl	$\|S\|$	$\|a\|$	swkl	$\|S\|$	$\|a\|$
sonar	0.33	2	5	–	–	–	0.05	1	43	**0.43**	2	3
haberman	**0.08**	1	4	**0.08**	1	3	0.04	11	3	0.04	1	1
breastCancer	0.79	6	3	0.81	6	4	0.35	6	9	**0.82**	6	2
australian	0.50	3	3	0.54	7	6	0.33	15	12	**0.55**	5	2
tictactoe	0.50	4	3	–	–	–	0.20	5	3	**0.87**	16	2
german	**0.15**	4	5	–	–	–	0.10	6	11	0.14	4	3
chess	0.76	11	4	–	–	–	0.34	4	15	**0.97**	17	2
mushrooms	0.97	3	4	–	–	–	0.40	5	20	**1.00**	12	1
magic	0.30	40	3	–	–	–	0.06	3	10	**0.47**	69	4
adult	0.24	31	5	–	–	–	0.00	1	10	**0.31**	103	4
avg. rank	1.8	1.7	2.0	–	–	–	3.0	1.9	2.9	**1.2**	2.5	1.1
iris	1.44	3	2	**1.45**	4	3	–	–	–	1.44	4	1
balance	0.63	9	3	–	–	–	–	–	–	**0.69**	9	3
CMC	0.18	7	3	0.16	20	4	–	–	–	**0.25**	7	2
page-blocks	0.36	19	3	–	–	–	–	–	–	**0.49**	21	3
nursery	0.92	2	3	–	–	–	–	–	–	**1.63**	81	3
automobile	0.85	5	5	–	–	–	–	–	–	**1.25**	5	2
glass	1.55	3	1	1.12	5	6	–	–	–	**1.92**	5	1
dermatology	1.85	6	3	1.02	9	6	–	–	–	**2.11**	9	2
kr-vs-k	0.62	13	3	–	–	–	–	–	–	**1.83**	351	3
abalone	0.53	14	3	–	–	–	–	–	–	**0.74**	16	2
avg. rank	1.9	1.2	1.7		–	–	–	–	–	**1.1**	1.9	1.3

Bibliography

[1] The Bureau of Transportation Statistics (BTS), 2021. URL `https://www.bts.gov/`.

[2] R. Agrawal, T. Imieliński, and A. Swami. Mining association rules between sets of items in large databases. In *Acm sigmod record*, volume 22, pages 207–216. ACM, 1993.

[3] J. Alcala-Fdez, R. Alcala, and F. Herrera. A fuzzy association rule-based classification model for high-dimensional problems with genetic rule selection and lateral tuning. *IEEE Transactions on Fuzzy systems*, 19(5):857–872, 2011.

[4] J. Alcalá-Fdez, A. Fernández, J. Luengo, J. Derrac, S. García, L. Sánchez, and F. Herrera. Keel data-mining software tool: data set repository, integration of algorithms and experimental analysis framework. *Journal of Multiple-Valued Logic & Soft Computing*, 17, 2011.

[5] E. Angelino, N. Larus-Stone, D. Alabi, M. Seltzer, and C. Rudin. Learning certifiably optimal rule lists. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 35–44. ACM, 2017.

[6] N. Antonio, A. de Almeida, and L. Nunes. Hotel booking demand datasets. *Data in brief*, 22:41–49, 2019.

[7] J. O. Aoga, T. Guns, S. Nijssen, and P. Schaus. Finding probabilistic rule lists using the minimum description length principle. In *International Conference on Discovery Science*, pages 66–82. Springer, 2018.

[8] M. Atzmueller. Subgroup discovery. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 5(1):35–49, 2015.

[9] A. Belfodil, A. Belfodil, and M. Kaytoue. Anytime subgroup discovery in numerical domains with guarantees. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 500–516. Springer, 2018.

[10] A. Belfodil, A. Belfodil, A. Bendimerad, P. Lamarre, C. Robardet, M. Kaytoue, and M. Plantevit. Fssd-a fast and efficient algorithm for subgroup set discovery. In *Proceedings of DSAA 2019*, 2019.

[11] E. Bellodi and F. Riguzzi. Structure learning of probabilistic logic programs by searching the clause space. *Theory and Practice of Logic Programming*, 15(2): 169–212, 2015.

[12] M. Boley, B. R. Goldsmith, L. M. Ghiringhelli, and J. Vreeken. Identifying consistent statements about numerical data with dispersion-corrected subgroup discovery. *Data Mining and Knowledge Discovery*, 31(5):1391–1418, 2017.

[13] C. Borgelt. Efficient implementations of apriori and eclat. In *FIMI'03: Proceedings of the IEEE ICDM workshop on frequent itemset mining implementations*, 2003.

[14] G. Bosc, J.-F. Boulicaut, C. Raïssi, and M. Kaytoue. Anytime discovery of a diverse set of patterns with monte carlo tree search. *Data Mining and Knowledge Discovery*, 32(3):604–650, 2018.

[15] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen. *Classification and regression trees*. CRC press, 1984.

[16] B. Bringmann and A. Zimmermann. The chosen few: On identifying valuable patterns. In *Seventh IEEE International Conference on Data Mining (ICDM 2007)*, pages 63–72. IEEE, 2007.

[17] K. H. Brodersen, C. S. Ong, K. E. Stephan, and J. M. Buhmann. The balanced accuracy and its posterior distribution. In *2010 20th international conference on pattern recognition*, pages 3121–3124. IEEE, 2010.

[18] K. Budhathoki and J. Vreeken. The difference and the norm—characterising similarities and differences between databases. In *Proceedings of ECMLP-KDD'15*, pages 206–223. Springer, 2015.

[19] K. Budhathoki, M. Boley, and J. Vreeken. Discovering reliable causal rules. *arXiv preprint arXiv:2009.02728*, 2020.

[20] T. Calders and S. Jaroszewicz. Efficient auc optimization for classification. In *European Conference on Principles of Data Mining and Knowledge Discovery*, pages 42–53. Springer, 2007.

[21] P. Clark and T. Niblett. The cn2 induction algorithm. *Machine learning*, 3(4): 261–283, 1989.

[22] W. W. Cohen. Fast effective rule induction. In *Proceedings of the twelfth international conference on machine learning*, pages 115–123, 1995.

[23] A.-W. De Leeuw, L. A. Meerhoff, and A. Knobbe. Effects of pacing properties on performance in long-distance running. *Big Data*, 6(4):248–261, 2018.

[24] E. Delahoz-Dominguez, R. Zuluaga, and T. Fontalvo-Herrera. Dataset of academic performance evolution for engineering students. *Data in Brief*, page 105537, 2020.

[25] J. Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine learning research*, 7(Jan):1–30, 2006.

[26] F. Doshi-Velez and B. Kim. Towards a rigorous science of interpretable machine learning. *arXiv:1702.08608 [stat.ML]*, 2017.

[27] F. Doshi-Velez and B. Kim. Considerations for evaluation and generalization in interpretable machine learning. In *Explainable and Interpretable Models in Computer Vision and Machine Learning*, pages 3–17. Springer, 2018.

[28] X. Du, Y. Pei, W. Duivesteijn, and M. Pechenizkiy. Exceptional spatio-temporal behavior mining through bayesian non-parametric modeling. *Data Mining and Knowledge Discovery*, 34(5):1267–1290, 2020.

[29] D. Dua and C. Graff. UCI machine learning repository, 2017. URL `http://archive.ics.uci.edu/ml`.

[30] W. Duivesteijn and A. Knobbe. Exploiting false discoveries–statistical validation of patterns and quality measures in subgroup discovery. In *2011 IEEE 11th International Conference on Data Mining*, pages 151–160. IEEE, 2011.

[31] W. Duivesteijn, A. Knobbe, A. Feelders, and M. van Leeuwen. Subgroup discovery meets bayesian networks–an exceptional model mining approach. In *2010 IEEE International Conference on Data Mining*, pages 158–167. IEEE, 2010.

[32] W. Duivesteijn, A. J. Feelders, and A. Knobbe. Exceptional model mining. *Data Mining and Knowledge Discovery*, 30(1):47–98, 2016.

[33] T. Fawcett. An introduction to roc analysis. *Pattern recognition letters*, 27(8): 861–874, 2006.

[34] A. Fernandez, V. Lopez, M. J. del Jesus, and F. Herrera. Revisiting evolutionary fuzzy systems: Taxonomy, applications, new trends and challenges. *Knowledge-Based Systems*, 80:109–121, 2015.

[35] J. Fischer and J. Vreeken. Sets of robust rules, and how to find them. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 38–54. Springer, 2019.

[36] J. Friedman, T. Hastie, and R. Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics New York, 2001.

[37] M. Friedman. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the american statistical association*, 32 (200):675–701, 1937.

[38] J. Fürnkranz. Separate-and-conquer rule learning. *Artificial Intelligence Review*, 13(1):3–54, 1999.

[39] J. Fürnkranz, D. Gamberger, and N. Lavrač. *Foundations of rule learning*. Springer Science & Business Media, 2012.

[40] E. Galbrun. The minimum description length principle for pattern mining: A survey. *arXiv preprint arXiv:2007.14009*, 2020.

[41] M. García-Borroto, J. F. Martínez-Trinidad, and J. A. Carrasco-Ochoa. A survey of emerging patterns for supervised classification. *Artificial Intelligence Review*, 42(4):705–721, 2014.

[42] A. Gelman, H. S. Stern, J. B. Carlin, D. B. Dunson, A. Vehtari, and D. B. Rubin. *Bayesian data analysis*. Chapman and Hall/CRC, 2013.

[43] B. R. Goldsmith, M. Boley, J. Vreeken, M. Scheffler, and L. M. Ghiringhelli. Uncovering structure-property relationships of materials by subgroup discovery. *New Journal of Physics*, 19(1):013031, 2017.

[44] M. Gönen, W. O. Johnson, Y. Lu, and P. H. Westfall. The bayesian two-sample t test. *The American Statistician*, 59(3):252–257, 2005.

[45] H. Grosskreutz and S. Rüping. On subgroup discovery in numerical domains. *Data Min. Knowl. Discov.*, 19(2):210–226, 2009.

[46] H. Grosskreutz, D. Paurat, and S. Rüping. An enhanced relevance criterion for more concise supervised pattern discovery. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1442–1450, 2012.

[47] P. Grünwald and T. Roos. Minimum description length revisited. *International Journal of Mathematics for Industry*, 11(1), 2019.

[48] P. D. Grünwald. *The minimum description length principle*. MIT press, 2007.

[49] W. Hämäläinen. Kingfisher: an efficient algorithm for searching for both positive and negative dependency rules with statistical significance measures. *Knowledge and information systems*, 32(2):383–414, 2012.

[50] W. Hämäläinen and G. I. Webb. Specious rules: an efficient and effective unifying method for removing misleading and uninformative patterns in association rule mining. In *Proceedings of the 2017 SIAM International Conference on Data Mining*, pages 309–317. SIAM, 2017.

[51] W. Hämäläinen and G. I. Webb. A tutorial on statistically sound pattern discovery. *Data Mining and Knowledge Discovery*, 33(2):325–377, 2019.

[52] F. Herrera, C. J. Carmona, P. González, and M. J. Del Jesus. An overview on subgroup discovery: foundations and applications. *Knowledge and information systems*, 29(3):495–525, 2011.

[53] F. Herrera, F. Charte, A. J. Rivera, and M. J. Del Jesus. Multilabel classification. In *Multilabel Classification*, pages 17–31. Springer, 2016.

[54] S. Holm. A simple sequentially rejective multiple test procedure. *Scandinavian journal of statistics*, pages 65–70, 1979.

[55] J. Hühn and E. Hüllermeier. Furia: an algorithm for unordered fuzzy rule induction. *Data Mining and Knowledge Discovery*, 19(3):293–319, 2009.

[56] J. Huysmans, K. Dejaeger, C. Mues, J. Vanthienen, and B. Baesens. An empirical evaluation of the comprehensibility of decision table, tree and rule based predictive models. *Decision Support Systems*, 51(1):141–154, 2011.

[57] R. L. Iman and J. M. Davenport. Approximations of the critical region of the fbietkan statistic. *Communications in Statistics-Theory and Methods*, 9(6):571–595, 1980.

[58] H. Jeffreys. *The theory of probability*. OUP Oxford, 1998.

[59] F. Jiménez, G. Sánchez, and J. M. Juárez. Multi-objective evolutionary algorithms for fuzzy classification in survival prediction. *Artificial intelligence in medicine*, 60(3):197–219, 2014.

[60] N. Jin, P. Flach, T. Wilcox, R. Sellman, J. Thumim, and A. Knobbe. Subgroup discovery in smart electricity meter data. *IEEE Transactions on Industrial Informatics*, 10(2):1327–1336, 2014.

[61] R. E. Kass and A. E. Raftery. Bayes factors. *Journal of the american statistical association*, 90(430):773–795, 1995.

[62] D. Klabjan. Large-scale models in the airline industry. In *Column generation*, pages 163–195. Springer, 2005.

[63] W. Klösgen. Explora: A multipattern and multistrategy discovery assistant. In *Advances in Knowledge Discovery and Data Mining*, pages 249–271. 1996.

[64] A. J. Knobbe and E. K. Ho. Pattern teams. In *European Conference on Principles of Data Mining and Knowledge Discovery*, pages 577–584. Springer, 2006.

[65] P. Kontkanen, P. Myllymäki, W. Buntine, J. Rissanen, and H. Tirri. An mdl framework for data clustering. *Minimum*, page 323, 2005.

[66] S. Kullback and R. A. Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

[67] H. Lakkaraju and C. Rudin. Learning cost-effective and interpretable treatment regimes for judicial bail decisions. *arXiv preprint arXiv:1610.06972*, 2016.

[68] H. Lakkaraju and C. Rudin. Learning cost-effective and interpretable treatment regimes. In *Artificial Intelligence and Statistics*, 2017.

[69] H. Lakkaraju, S. H. Bach, and J. Leskovec. Interpretable decision sets: A joint framework for description and prediction. In *Proceedings of KDD'16*, pages 1675–1684. ACM, 2016.

[70] N. Lavrač, P. Flach, and B. Zupan. Rule evaluation measures: A unifying view. In *International Conference on Inductive Logic Programming*, pages 174–185. Springer, 1999.

[71] N. Lavrač, B. Kavšek, P. Flach, and L. Todorovski. Subgroup discovery with cn2-sd. *Journal of Machine Learning Research*, 5(Feb):153–188, 2004.

[72] M. van Leeuwen. Maximal exceptions with minimal descriptions. *Data Mining and Knowledge Discovery*, 21(2):259–276, 2010.

[73] M. van Leeuwen and E. Galbrun. Association discovery in two-view data. *IEEE Transactions on Knowledge and Data Engineering*, 27(12):3190–3202, 2015.

[74] M. van Leeuwen and A. Knobbe. Non-redundant subgroup discovery in large and complex data. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 459–474. Springer, 2011.

[75] M. van Leeuwen and A. Knobbe. Diverse subgroup set discovery. *Data Mining and Knowledge Discovery*, 25(2):208–242, 2012.

[76] M. van Leeuwen and A. Ukkonen. Discovering skylines of subgroup sets. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 272–287. Springer, 2013.

[77] M. van Leeuwen and A. Ukkonen. Expect the unexpected–on the significance of subgroups. In *International Conference on Discovery Science*, pages 51–66. Springer, 2016.

[78] M. van Leeuwen and J. Vreeken. Mining and using sets of patterns through compression. In *Frequent Pattern Mining*, pages 165–198. Springer, 2014.

[79] D. Leman, A. Feelders, and A. Knobbe. Exceptional model mining. In *Joint European conference on machine learning and knowledge discovery in databases*, pages 1–16. Springer, 2008.

[80] F. Lemmerich, M. Atzmueller, and F. Puppe. Fast exhaustive subgroup discovery with numerical target concepts. *Data Mining and Knowledge Discovery*, 30 (3):711–762, 2016.

[81] B. Letham, C. Rudin, T. H. McCormick, D. Madigan, et al. Interpretable classifiers using rules and bayesian analysis: Building a better stroke prediction model. *The Annals of Applied Statistics*, 9(3):1350–1371, 2015.

[82] W. Li, J. Han, and J. Pei. Cmar: Accurate and efficient classification based on multiple class-association rules. In *Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on*, pages 369–376. IEEE, 2001.

[83] J. Lijffijt, B. Kang, W. Duivesteijn, K. Puolamaki, E. Oikarinen, and T. De Bie. Subjectively interesting subgroup discovery on real-valued targets. In *2018 IEEE ICDE*, pages 1352–1355. IEEE, 2018.

[84] Y. Lou, R. Caruana, and J. Gehrke. Intelligible models for classification and regression. In *Proceedings KDD'12*, pages 150–158. ACM, 2012.

[85] B. L. W. H. Y. Ma and B. Liu. Integrating classification and association rule mining. In *Proceedings of the fourth international conference on knowledge discovery and data mining*, 1998.

[86] T. Makhalova, S. O. Kuznetsov, and A. Napoli. Mint: Mdl-based approach for mining interesting numerical pattern sets. *arXiv preprint arXiv:2011.14843*, 2020.

[87] M. Meeng and A. Knobbe. Flexible enrichment with cortana–software demo. In *Proceedings of BeneLearn*, pages 117–119, 2011.

[88] M. Meeng and A. Knobbe. For real: a thorough look at numeric attributes in subgroup discovery. *Data Mining and Knowledge Discovery*, pages 1–55, 2020.

[89] M. Meeng, H. de Vries, P. Flach, S. Nijssen, and A. Knobbe. Uni-and multivariate probability density models for numeric subgroup discovery. *Intelligent Data Analysis*, 24(6):1403–1439, 2020.

[90] T. Mielikäinen and H. Mannila. The pattern ordering problem. In *European Conference on Principles of Data Mining and Knowledge Discovery*, pages 327–338. Springer, 2003.

[91] C. Molnar. Interpretable machine learning. *A Guide for Making Black Box Models Explainable*, 2018.

[92] T. Mononen and P. Myllymäki. Computing the multinomial stochastic complexity in sub-linear time. In *Proceedings of the 4th European Workshop on Probabilistic Graphical Models*, pages 209–216, 2008.

[93] I. J. Myung. Tutorial on maximum likelihood estimation. *Journal of mathematical Psychology*, 47(1):90–100, 2003.

[94] E. B. Peterson, K. Neels, N. Barczi, and T. Graham. The economic cost of airline flight delay. *Journal of Transport Economics and Policy (JTEP)*, 47(1):107–121, 2013.

[95] I. Polaka, E. Gašenko, O. Barash, H. Haick, and M. Leja. Constructing interpretable classifiers to diagnose gastric cancer based on breath tests. *Procedia Computer Science*, 104, 2017.

[96] H. M. Proença and M. van Leeuwen. Interpretable multiclass classification by mdl-based rule lists. *Information Sciences*, 512:1372–1393, 2020.

[97] H. M. Proença, S. M. Vieira, U. Kaymak, R. J. Almeida, and J. M. Sousa. Optimizing probabilistic fuzzy systems for classification using metaheuristics. In *2016 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pages 1635–1641. IEEE, 2016.

[98] H. M. Proença, R. Klijn, T. Bäck, and M. van Leeuwen. Identifying flight delay patterns using diverse subgroup discovery. In *2018 IEEE SSCI*, pages 60–67. IEEE, 2018.

[99] H. M. Proença, P. Grünwald, T. Bäck, and M. van Leeuwen. Discovering outstanding subgroup lists for numeric targets using mdl. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 19–35. Springer, 2020.

[100] H. M. Proença, T. Bäck, and M. van Leeuwen. Robust subgroup discovery. *arXiv preprint arXiv:2103.13686*, 2021.

[101] H. M. Proença, T. Bäck, and M. van Leeuwen. Robust subgroup discovery. *Data Mining and Knowledge Discovery (preprint available in arXiv:2103.13686)*, submitted.

[102] F. Provost and P. Domingos. Well-trained pets: Improving probability estimation trees. 2000.

[103] J. R. Quinlan. *C4. 5: programs for machine learning*. Elsevier, 2014.

[104] M. T. Ribeiro, S. Singh, and C. Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

[105] M. T. Ribeiro, S. Singh, and C. Guestrin. Anchors: High-precision model-agnostic explanations. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI)*, 2018.

[106] R. Rifkin and A. Klautau. In defense of one-vs-all classification. *The Journal of Machine Learning Research*, 5:101–141, 2004.

[107] J. Rissanen. Modeling by shortest data description. *Automatica*, 14(5), 1978.

[108] J. Rissanen. A universal prior for integers and estimation by minimum description length. *The Annals of statistics*, pages 416–431, 1983.

[109] R. L. Rivest. Learning decision lists. *Machine learning*, 2(3):229–246, 1987.

[110] J. N. Rouder, P. L. Speckman, D. Sun, R. D. Morey, and G. Iverson. Bayesian t tests for accepting and rejecting the null hypothesis. *Psychonomic bulletin & review*, 16(2):225–237, 2009.

[111] C. Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1 (5):206–215, 2019.

[112] C. M. Salgado, C. Azevedo, H. Proença, and S. M. Vieira. Missing data. *Secondary analysis of electronic health records*, pages 143–162, 2016.

[113] C. M. Salgado, C. Azevedo, H. Proença, and S. M. Vieira. Noise versus outliers. *Secondary analysis of electronic health records*, pages 163–183, 2016.

[114] C. E. Shannon. A mathematical theory of communication. *Bell system technical journal*, 27(3):379–423, 1948.

[115] Y. M. Shtar'kov. Universal sequential coding of single messages. *Problemy Peredachi Informatsii*, 23(3):3–17, 1987.

[116] A. Siebes. Data surveying: Foundations of an inductive query language. In *KDD*, pages 269–274, 1995.

[117] G. Tsoumakas, E. Spyromitros-Xioufis, J. Vilcek, and I. Vlahavas. Mulan: A java library for multi-label learning. *Journal of Machine Learning Research*, 12: 2411–2414, 2011.

[118] J. W. Tukey. *Exploratory data analysis*, volume 2. Reading, MA, 1977.

[119] V. Vapnik. *The nature of statistical learning theory*. Springer science & business media, 2013.

[120] J. Vreeken, M. van Leeuwen, and A. Siebes. Krimp: mining itemsets that compress. *Data Mining and Knowledge Discovery*, 23(1):169–214, 2011.

[121] J. Wang and G. Karypis. Harmony: Efficiently mining the best rules for classification. In *Proceedings of the 2005 SIAM International Conference on Data Mining*, pages 205–216. SIAM, 2005.

[122] T. Wang, C. Rudin, F. Velez-Doshi, Y. Liu, E. Klampfl, and P. MacNeille. Bayesian rule sets for interpretable classification. In *Data Mining (ICDM), 2016 IEEE 16th International Conference on*, pages 1269–1274. IEEE, 2016.

[123] G. I. Webb. Opus: An efficient admissible algorithm for unordered search. *Journal of Artificial Intelligence Research*, 3:431–465, 1995.

[124] G. I. Webb. Discovering significant patterns. *Machine Learning*, 68(1):1–33, 2007.

[125] H. Yang, C. Rudin, and M. Seltzer. Scalable bayesian rule lists. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3921–3930. JMLR. org, 2017.

[126] J. Zeng, B. Ustun, and C. Rudin. Interpretable classification models for recidivism prediction. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 180(3), 2017.

[127] X. Zhang, G. Dong, and K. Ramamohanarao. Information-based classification by aggregating emerging patterns. In *IDEAL*, pages 48–53. Springer, 2000.

[128] A. Zimmermann and S. Nijssen. Supervised pattern mining and applications to classification. In *Frequent Pattern Mining*. Springer, 2014.