Robust rules for prediction and description
Manuel Proenca, H.

Citation

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).
Appendices
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:

\[
KL_{\mu,\sigma}(p; q) = \int_{-\infty}^{+\infty} p(x) \log p(x) \, dx - \int_{-\infty}^{+\infty} p(x) \log q(x) \, dx
= \mathbb{E}_p \left[ \log p(x) \right] - \mathbb{E}_p \left[ \log q(x) \right]
= -\frac{1}{2} \left( \log e + \log 2\pi \sigma_p^2 \right) + \frac{1}{2} \log 2\pi \sigma_q^2 + \mathbb{E}_p \left[ \frac{(x - \mu_q)^2}{2\sigma_q^2} \log e \right]
= -\frac{\log e}{2} + \log \frac{\sigma_p}{\sigma_q} + \mathbb{E}_p \left[ \frac{x^2 - 2x\mu_q + \mu_q^2}{2\sigma_q^2} \log e \right]
= -\frac{\log e}{2} + \log \frac{\sigma_q}{\sigma_p} + \frac{\sigma_p^2 + \mu_p^2 - 2\mu_p\mu_q + \mu_q^2}{2\sigma_q^2} \log e
= -\frac{\log e}{2} + \log \frac{\sigma_q}{\sigma_p} + \frac{\sigma_p^2 + (\mu_p - \mu_q)^2}{2\sigma_q^2} \log e.
\]

(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:

\[
KL_{\mu}(p; q) = \frac{(\mu_p - \mu_q)^2}{2\sigma^2} \log e,
\]

(A.2)

and the weighted version of this \( KL_{\mu} \), 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 $KL_\mu$. 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 $D = \{X, Y\}$, $Y$ has $k = |Y|$ class labels and a model $M$ that forms a partition over the whole data. The model $M$ divides the data $D$ in $\omega$ parts, of the form \{(X^1, Y^1), \cdots, (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 $i$:

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

Unlike the regular maximum likelihood estimator, this smoothed variant—known as Laplace smoothing—adds a (small) pseudocount $\epsilon$ 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 $\epsilon$ 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 $\epsilon$ 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
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 $\epsilon$. 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

$$
\text{Pr}_{\text{plug-in}}(y^u = c \mid Y^{u-1}) := \frac{|\{y \in Y^{u-1} \mid y = c\}| + \epsilon}{\sum_{c' \in Y} |\{y \in Y^{u-1} \mid y = c'\}| + \epsilon}, \quad (B.2)
$$

where $u \in \mathbb{N}$, $y^u$ represents the $u^\text{th}$ class label in $Y$, $Y^{u-1} = \{y^1, ..., y^{u-1}\}$ represents the sequence of the $u - 1$ first class labels, and $\epsilon$ is the pseudocount necessary for $\text{Pr}_{\text{plug-in}}(y^1 = c \mid Y^{10}) = \epsilon / k \epsilon = 1 / k$ to be valid. The most common values for $\epsilon$, 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 $\omega$ parts. But assuming no interaction between the parts, the total encoding is equal to the sum of its parts:

$$
L_{\text{plug-in}}(Y \mid X, M) = -\log \prod_{i=1}^{\omega} \text{Pr}_{\text{plug-in}}(Y^i) = \sum_{i=1}^{\omega} L_{\text{plug-in}}(Y^i), \quad (B.3)
$$

where $L_{\text{plug-in}}(Y^i) = -\log \text{Pr}_{\text{plug-in}}(Y^i)$.

Inserting the prequential plug-in code (B.2) in (B.3) we obtain for each part $Y^i$:

$$
L_{\text{plug-in}}(Y^i) = -\log \left( \prod_{u=1}^{n_i} \prod_{c=1}^{k} \frac{n_{c|u}}{\Gamma(n_{c|u} + k \epsilon)} \right)
= -\log \left( \frac{\prod_{c=1}^{k} \Gamma(n_{c|i} + \epsilon)}{\Gamma(n_{i} + k \epsilon)} \right), \quad (B.4)
$$

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, $\Gamma$ 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.
Normalized Maximum Likelihood for rule lists with categorical distributions

For this section, let us assume that we have a dataset $D = \{X, Y\}$ and model $M$ that forms a partition over the whole data. Model $M$ divides the data $D$ in $\omega$ parts, of the form $\{(X^1, Y^1), \ldots, (X^\omega, Y^\omega)\}$. 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:

$$L_{NML}(Y \mid X, M) = \sum_{i=1}^\omega L_{NML}(Y^i).$$  \hspace{1cm} (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, let’s recall the definition of the NML probability distribution [115]:

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

where $\mathcal{Y}^n$ is the set of all possible sequences of $n$ points with $k = |\mathcal{Y}|$ categories, $\hat{M}(Y \mid X)$ and $\hat{M}(Z \mid X)$ 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 \(\omega\) parts, we can further develop the previous formula to:

\[
L_{\text{NML}}(Y \mid X, M) \overset{\text{i.i.d.}}{=} - \log \left( \prod_{i=1}^{n} \Pr(y^i \mid x^i; \hat{M}(y^i \mid x^i)) \right) \\
= - \log \left( \prod_{i'=1}^{\omega} \Pr(Y^{i'}; \hat{\Theta}(Y^{i'})) \right) \\
= - \log \left( \prod_{i'=1}^{\omega} \frac{\log(l(\hat{\Theta}^{i'} | Y^{i'}))}{g(Y, X, M)} \right) \\
= - \log \left( \sum_{i'=1}^{\omega} \log(l(\hat{\Theta}^{i'} | Y^{i'})) \right) + \log g(Y, X, M),
\]

where \(l(\hat{\Theta}^{i'} | Y^{i'})\) is the likelihood function for each of the \(\omega\) 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 \(C(n_1, k)\) of one part \(D^1 = \{Y^1, X^1\}\), with \(n_1 = |D^1|\) and \(k = \mathcal{Y}\)

\[
C(n_1, k) = \log \left( \sum_{Z \in \mathcal{Y}^{n_1}} \Pr(Z^1; \hat{\Theta}(Z^1)) \right) \\
= \log \left( \sum_{Z \in \mathcal{Y}^{n_1}} \prod_{i=1}^{n_1} \Pr(z^i; \hat{\Theta}(Z^1)) \right) \\
= \log \left( \sum_{n_{c|1}} \frac{n_1!}{n_1! n_{c|1}!^{\frac{n_1}{n_1}} \prod_{c \in \mathcal{Y}} \frac{n_{c|1}}{n_1}} \right)
\]

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 \(C(n, k)\) simpler [48]. It is interesting to note that \(C(n, k)\) 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
Appendix C. Normalized Maximum Likelihood for rule lists with categorical distributions

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. $|A_1| = |A_2| = \cdots = |A_a| = n_A$.

<table>
<thead>
<tr>
<th>Part 1</th>
<th>Part 2</th>
<th>Part 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A_1$</td>
<td>$B_1$</td>
<td>$C_1$</td>
</tr>
<tr>
<td>$A_1$</td>
<td>$B_1$</td>
<td>$C_2$</td>
</tr>
<tr>
<td>$\vdots$</td>
<td>$\vdots$</td>
<td>$\vdots$</td>
</tr>
<tr>
<td>$A_1$</td>
<td>$B_2$</td>
<td>$C_1$</td>
</tr>
<tr>
<td>$\vdots$</td>
<td>$\vdots$</td>
<td>$\vdots$</td>
</tr>
<tr>
<td>$A_a$</td>
<td>$B_b$</td>
<td>$C_c$</td>
</tr>
</tbody>
</table>

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

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

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

$$\log g(Y, X, M) = \log \sum_{Z \in \mathcal{Z}^n} \prod_{i'=1}^{\omega} \Pr(Z^{i'}; \hat{\Theta}(Z^{i'}))$$

$$= \log \prod_{i'=1}^{\omega} \sum_{Z^{i'} \in \mathcal{Z}^{n,i'}} \Pr(Z^{i'}; \hat{\Theta}(Z^{i'})) \quad (C.4)$$

$$= \sum_{i'=1}^{\omega} \log \sum_{Z^{i'} \in \mathcal{Z}^{n,i'}} \Pr(Z^{i'}; \hat{\Theta}(Z^{i'}))$$

$$= \sum_{i'=1}^{\omega} \log C(n_{i'}, k)$$
Substituting this back into Eq. (C.2), we obtain what we wanted:

\[
L_{NML}(Y \mid X, M) = -\log \left( \sum_{i=1}^{\omega} l(\hat{\Theta}^i \mid Y^i) \right) + \sum_{i=1}^{\omega} \log C(n_i, k)
\]

\[
= \sum_{i=1}^{\omega} l(\hat{\Theta}^i \mid Y^i) + C(n_i, k) \tag{C.5}
\]

\[
= \sum_{i=1}^{\omega} L_{NML}(Y^i)
\]
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 $Y = \{y_1, \ldots, y_n\}$, the Bayesian encoding takes the following form:

$$P_{Bayes}(Y) = \int_{\Theta} f(Y | \Theta) w(\Theta) \, d\Theta,$$

where $f$ is the probability density function (pdf), $\Theta$ 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 $\mu$ and $\sigma$ being its mean and standard deviation, respectively, the pdf $f(Y | \Theta)$ over a sequence $Y$ is the multiplication of the individual pdfs, thus:

$$f(Y | \mu, \sigma) = \frac{1}{(2\pi)^{n/2}\sigma^n} \exp \left[ -\frac{1}{2\sigma^2} \sum_{i}^{n} (y_i - \mu)^2 \right],$$

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 $\mu$ and $\sigma$, and add an extra. Thus, our prior is given by:

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

where $1/\sqrt{2\pi}$ was added for normalization reasons.
Putting everything together, one obtains:

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

The integrals over the whole space of the parameters \( \mu \) and \( \sigma \) 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 (\( \mu \) and \( \sigma \)) is \( k = 2 \). Thus, instead of using \( w(\mu, \sigma) \) we will in practice be using \( w(\mu, \sigma | Y|2) \), and using the the chain rule and the Bayesian formula returns a total encoding of \( Y \) equal to

\[ P(Y) = P_{\text{Bayes}}(Y | Y|2) P(Y|2) = \frac{P_{\text{Bayes}}(Y)}{P_{\text{Bayes}}(Y|2)} P(Y|2) \] (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 \( P_{\text{Bayes}}(Y) \) and then choose the appropriate probability for \( P(Y|2) \).

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

\[ \Gamma(k) = \int_{0}^{+\infty} z^{k-1} e^{-z} dz, \] (D.6)

with the corresponding variable transformation:

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

Performing the variable transformation and noting that the minus sign of \( dz \) cancels with the reversing of the integral limits, we get:

\[ P_{\text{Bayes}}(Y) = \Gamma \left( \frac{n+1}{2} \right) 2^{\frac{n+1}{2} - 1} (2\pi)^{-\frac{n+1}{2}} \int_{-\infty}^{+\infty} \left[ \sum_{i}^{n} (y^i - \mu)^2 \right]^{-\frac{n+1}{2}} d\mu, \] (D.8)
which reveals that the prior on the effect size \( \rho \), and specifically its standard deviation parameter \( \tau \), is equivalent to adding \( 1/\tau^2 \) virtual points to the original data.

To solve the integral in \( \mu \) we need to introduce the statistics \( \hat{\mu} \) and \( \hat{\sigma}^2 \) as the values estimated from the data. We define these quantities as:

\[
\hat{\mu} = \frac{1}{n} \sum_{i} y_i; \quad \hat{\sigma}^2 = \frac{1}{n} \sum_{i} (y_i - \hat{\mu})^2 ,
\]

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 \( \text{RSS} = n \hat{\sigma} \).

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:

\[
\int_{-\infty}^{+\infty} \left[ \sum_{i} (y_i^2 - n\hat{\mu}^2 + n\hat{\mu}^2 - 2n\hat{\mu} + n\mu^2) \right]^{-(n+1)/2} \left[ n\hat{\sigma}^2 + n(\hat{\mu} - \mu)^2 \right]^{-(n+1)/2} \, d\mu = \frac{\Gamma \left( \frac{n+1}{2} \right)}{\sqrt{\pi n s_s}} \frac{\sqrt{\pi n s_s}}{\Gamma \left( \frac{n}{2} \right)} ,
\]

where \( s_s^2 = \sigma^2/n \) 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

\[
\int_{-\infty}^{+\infty} \left[ 1 + \frac{1}{n} \left( \frac{\hat{\mu} - \mu}{s_s} \right)^2 \right]^{-(n+1)/2} \, \mu = \Gamma \left( \frac{n}{2} \right) \sqrt{\pi n s_s} \frac{\sqrt{\pi n s_s}}{\Gamma \left( \frac{n+1}{2} \right)} .
\]
Inserting this back in Eq. D.4 we obtain:

\[
P_{\text{Bayes}}(Y) = \\
= \Gamma \left( \frac{n + 1}{2} \right) 2^{\frac{n+1}{2}} (2\pi)^{-\frac{n+1}{2}} \Gamma \left( \frac{n}{2} \sqrt{\pi n s_s} \right) \left[ n\sigma^2 \right]^{-(n+1)/2} \\
= 2^{-1} \pi^{-\frac{n}{2}} \Gamma \left( \frac{n}{2} \right) \frac{1}{\sqrt{n}} \left[ n\hat{\sigma}^2 \right]^{-\frac{n}{2}},
\]  

(D.12)

Returning to 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 \( Y|2 = \{y_1^1, y_2^1\} \) as:

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

(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,

\[
L_{\text{Bayes,2.0}}(Y) = -\log P_{\text{Bayes}}(Y) + \log P_{\text{Bayes}}(Y|2) - \log P(Y|2) \\
= 1 + \frac{n}{2} \log \pi - \log \Gamma \left( \frac{n}{2} \right) + \frac{1}{2} \log n + \frac{n}{2} \log \left( n\hat{\sigma}^2 \right) \\
- 1 - \frac{1}{2} \log \pi + 0 - \frac{1}{2} - \log \left( \sum_i (y_i^1 - \hat{\mu}_2)^2 \right) \\
+ \frac{1}{2} \log \pi + \log \hat{\sigma}_d + \left[ \frac{1}{2\hat{\sigma}_d^2} \sum_i (y_i^1 - \hat{\mu}_d)^2 \right] \log e \\
= \frac{n}{2} \log \pi - \log \Gamma \left( \frac{n}{2} \right) + \frac{1}{2} \log n + \frac{n}{2} \log \left( n\hat{\sigma}^2 \right) + L_{\text{cost}}(Y|2),
\]  

(D.14)

where \( \hat{\mu}_2 \) is the estimated mean of \( y_1^1, y_2^1 \) and \( L_{\text{cost}}(Y|2) \) 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., \( y_1^1, y_2^1 \). Because we want to minimize this length, we notice that there are only two terms that contribute to it in \( L_{\text{cost}}(Y|2) \), and thus by choosing the two observations close to \( \hat{\mu}_d \) minimizes both the encoding of \( P(Y|2) \) and maximize \( P_{\text{Bayes}}(Y|2) \) for most cases. There are exceptions to this, depending on the respective values of \( \mu_d \) and \( y_1^1, y_2^1 \), but these are not significant to change the values too much and also requires less computational search to find the points.
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 \( n \):

\[
-\log \Gamma \left( \frac{n}{2} \right) \\
\sim -\frac{1}{2} \log \pi - \frac{1}{2} \log (n-2) - \left( \frac{n}{2} - 1 \right) \log \left( \frac{n}{2} - 1 \right) + \left( \frac{n}{2} - 1 \right) \log e, \tag{E.2}
\]
and finally we insert it into Eq. (D.14) and assume $\tau = 1$ to obtain:

$$L(Y) \sim$$

$$\sim 1 + \frac{n-1}{2} \log \pi + \frac{1}{2} \log \left( \frac{n}{n-2} \right) + \frac{n}{2} \log \left( \frac{n\sigma^2}{n/2-1} \right) + \left( \frac{n}{2} - 1 \right) \log e$$

$$+ \log \left( \frac{n}{2} - 1 \right) + L_{cost}(Y|2)$$

$$\sim \frac{n}{2} \log \pi + \frac{n}{2} \log 2\sigma^2 + \left[ \frac{1}{2\sigma^2} \sum_i^n (y^i - \mu)^2 \right] \log e + \log n - \log e + L_{cost}(Y|2)$$

$$= L(Y | \hat{\Theta}) + \log \frac{n}{e} + L_{cost}(Y|2)$$

$$\sim \frac{1}{2} \left( 2L(Y | \hat{\Theta}) + 2 \log n - 2 \log e \right)$$

$$= \frac{1}{2} BIC,$$

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,

$$BIC = -2 \ln \ell(\Theta | Y) + k \ln n,$$

(E.4)

where $\ell(\Theta | Y)$ is the likelihood as estimated from the data, and $k$ is the number of parameters, which in our case is 2.
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     | $|D|$ | $|V|$ | $|Y|$ | $|Cands|$ |
|-------------|-----|-----|-----|-------|
| 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.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      | $T$ | $|\mathcal{Y}|$ | $|D|$ | $V_{(\text{nom./num.})}$ |
|--------------|-----|----------------|------|-------------------------|
| 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              | 1000 | (13/7)                  |
| chess        | 1   | 2              | 3196 | (36/0)                  |
| mushrooms    | 1   | 2              | 8124 | (22/0)                  |
| magic        | 1   | 2              | 19020| (0/10)                  |
| adult        | 1   | 2              | 45222| (8/6)                   |
| iris         | 1   | 3              | 150  | (0/4)                   |
| balance      | 1   | 3              | 625  | (0/4)                   |
| CMC          | 1   | 3              | 1473 | (0/9)                   |
| page-blocks  | 1   | 5              | 5472 | (0/10)                  |
| nursery      | 1   | 5              | 12960| (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             | 28056| (6/0)                   |
| abalone      | 1   | 28             | 4174 | (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  | (0/258)                 |
| genbase      | 27  | 2              | 662  | (1186/0)                |
| mediamill    | 101 | 2              | 43907| (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 $[\text{min.}, \text{max.}]$; number of samples $|D|$; number of type of variables (nominal/numeric)\}.

| Dataset   | $T$ | $[\text{min.}; \text{max.}]$ | $|D|$ $V(\text{nom./num.})$ |
|-----------|-----|-------------------------------|-----------------------------|
| 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 $\beta$ 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) $L(D \mid \hat{\Theta}^d)$

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

(a) Univariate nominal target.  \hspace{1cm} (b) Univariate numeric target.

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

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).

(a) Univariate nominal target.  
(b) Univariate numeric target.
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.

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).
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).

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).
G.3. Analysis of RSD beam search hyperparameters

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.

<table>
<thead>
<tr>
<th>datasets</th>
<th>DSSD</th>
<th>MCTS4DM</th>
<th>FSSD</th>
<th>RSD</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>swkl</td>
<td></td>
<td>swkl</td>
<td></td>
</tr>
<tr>
<td>sonar</td>
<td>0.33</td>
<td>2</td>
<td>5</td>
<td>0.05</td>
</tr>
<tr>
<td>haberman</td>
<td><strong>0.08</strong></td>
<td>1</td>
<td>4</td>
<td>0.08</td>
</tr>
<tr>
<td>breastCancer</td>
<td>0.79</td>
<td>6</td>
<td>3</td>
<td>0.81</td>
</tr>
<tr>
<td>australian</td>
<td>0.50</td>
<td>3</td>
<td>3</td>
<td>0.54</td>
</tr>
<tr>
<td>tictactoe</td>
<td>0.50</td>
<td>4</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>german</td>
<td><strong>0.15</strong></td>
<td>4</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>chess</td>
<td>0.76</td>
<td>11</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>mushrooms</td>
<td>0.97</td>
<td>3</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>magic</td>
<td>0.30</td>
<td>40</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>adult</td>
<td>0.24</td>
<td>31</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>avg. rank</td>
<td>1.8</td>
<td>1.7</td>
<td>2.0</td>
<td></td>
</tr>
<tr>
<td>iris</td>
<td>1.44</td>
<td>3</td>
<td>2</td>
<td><strong>1.45</strong></td>
</tr>
<tr>
<td>balance</td>
<td>0.63</td>
<td>9</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>CMC</td>
<td>0.18</td>
<td>7</td>
<td>3</td>
<td>0.16</td>
</tr>
<tr>
<td>page-blocks</td>
<td>0.36</td>
<td>19</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>nursery</td>
<td>0.92</td>
<td>2</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>automobile</td>
<td>0.85</td>
<td>5</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>glass</td>
<td>1.55</td>
<td>3</td>
<td>1</td>
<td>1.12</td>
</tr>
<tr>
<td>dermatology</td>
<td>1.85</td>
<td>6</td>
<td>3</td>
<td>1.02</td>
</tr>
<tr>
<td>kr-vs-k</td>
<td>0.62</td>
<td>13</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>abalone</td>
<td>0.53</td>
<td>14</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>avg. rank</td>
<td>1.9</td>
<td>1.2</td>
<td>1.7</td>
<td></td>
</tr>
</tbody>
</table>


