# Hierarchical two-part MDL code for multinomial distributions

Marc Boullé

Orange Labs - 22300 Lannion - France

**Abstract**

We leverage the Minimum Description Length (MDL) principle as a model selection technique for multinomial distributions and suggest a two-part MDL code based on a hierarchical encoding of the multinomial parameters. We compare this code with the alternative Normalized Maximum Likelihood (NML) code and exhibit large regions of the parameter space where the hierarchical code dominates the NML one. We then present an application of the multinomial distribution to joint density estimation and show that the hierarchical code brings significant improvements.

## 1 Introduction

Industrial companies such as Orange, the main french telecommunication operator, store large amounts of data. They have to deal with many requests for data mining studies, in a wide diversity of application domains and tasks, structure and scale of data, constraints, resource or business requirements. To address these problems in an industrial context, Orange Labs has developed a data mining tool [1], with the following requirements: generic, reliable, accurate, automatic, interpretable and scalable. This tool exploits models for conditional or joint density estimation in the univariate or multivariate cases, with either numerical or categorical variables (Boullé, 2011), for feature selection and construction in the multi-tables context and for modeling (Boullé, 2014). All these models extensively use multinomial distributions as building blocks, and the inference process heavily relies on MDL model selection to meet the tool requirements. Enumerative codes have been used for years, being effective (they are both two-parts, one-part and NML codes) and very simple and efficient to compute at any scale. The objective of this paper is to study whether these codes can be improved in order to detect patterns with fewer instances, with the least possible computational overhead. In particular, we focus on the case of data sets with heavily unbalanced distributions, such as Zipf'law or Pareto distribution, which widely appears in many application domains such as linguistics, physics or economics (Powers, 1998; Newman, 2005).

Model selection is a key problem in statistics and data mining, and the MDL approaches (Rissanen, 1978) to model selection have been extensively studied in the literature (Grünwald, 2007), with successful applications in many practical problems. Simple models such as multinomial distributions are important because they are easy to analyze theoretically and useful in many applications. For example, the multinomial distribution has been used as a building block in more complex models, such as naive Bayes classifiers (Mononen and Myllymäki, 2007), Bayesian networks (Silander et al., 2010; Guo et al., 2017), decision trees (Voisine et al., 2009) or coclustering models (Boullé, 2011; Guigourès et al., 2015). These models involve up to thousands of multinomial blocks, some of them with potentially very large numbers of occurrences and outcomes. For example, the text $\times$ word coclustering of the 20-newsgroup data set described in (Boullé, 2011) exploits a main multinomial block with around two millions words (occurrences) distributed on 200,000 coclusters (outcomes). In (Guigourès et al., 2015), half a billion call detail records (occurrences) are distributed on one million coclusters

[1] This tool, named Khiops, is available as a shareware at `www.khiops.com`

(outcomes). These various and numerous applications critically rely on the use of effective and efficient MDL codes to get a robust and accurate summary of the data.

The MDL approaches come with several flavors, ranging from theoretical but not computable to practical but sub-optimal. Ideal MDL (Vitányi and Li, 2000) relies on the Kolmogorov complexity, that is the ability of compressing data using a computer program. However, it suffers from large constants depending on the description method used and cannot be computed, not even approximated in the case of two-part codes (Adriaans and Vitányi, 2007). Practical MDL leverages description methods that are less expressive than general-purpose computer languages. It has been employed to retrieve the best model given the data in case of families of parametrized statistical distributions. Crude MDL is a basic MDL approach with appealing simplicity. In two-part crude MDL, you just have to encode the model parameters and the data given the parameter, with a focus on the code length only. However, crude MDL suffers from arbitrary coding choices. Modern MDL relies on universal coding resulting in Refined MDL (Grünwald, 2007), with much stronger foundations and interesting theoretical properties. In particularly, the normalized maximum likelihood (NML) (Rissanen, 1996) provides a theoretically solid criterion based on a minimax regret strategy. The NML approach exploits a constant regret: all the distributions are treated on the same footing and the one that best fits the data is chosen. Interestingly, the enumerative two-part MDL code for multinomial models has a strong connection with the NML approach (Boullé et al., 2016). Despite its simplicity, this code is both a two-part and a one-part code, is optimal w.r.t. the NML approach and is parametrization invariant.

In this paper, we investigate on two-part codes for multinomial models based on a hierarchical encoding of the model parameters. Although they loose the appealing theoretical properties of the alternative NML code, they reach a better compression on large regions of the parameter space, namely in case of unbalanced multinomial distributions, with a negligible loss on the rest of the parameter space. We present an application of multinomial models to joint density estimation. We show that using the proposed hierarchical multinomial code significantly improves the quality of the retrieved models in the case of peaked densities, which closely relates to unbalanced distributions.

The rest of the paper is organized as follows. For self-containment reasons, Section 2 presents NML codes for the multinomial distribution. Section 3 introduces a hierarchical code for the multinomial distribution and compares it to alternative enumerative NML code. Section 4 presents an application of these codes to joint density estimation and analyzes the impact of the chosen code, from balanced to unbalanced data. Finally, Section 5 summarizes this paper.

# 2 NML codes for multinomial distribution

Let us consider the multinomial model with parameter $\theta = (\theta_1, \ldots, \theta_m)$, $\sum_{j=1}^{m} \theta_j = 1, \forall j, \theta_j > 0$, such that $P_\theta(X = j) = \theta_j$, in the case of m-ary sequences $x^n \in X^n$ of size $n$. For a given sequence $x_n$, $P_\theta(x_n) = \prod_{j=1}^{m} \theta_j^{n_j}$, where $n_j$ is the number of occurrences of outcome $j$ in sequence $x^n$.

## 2.1 Standard NML approach

Using universal coding, a grounded approach is proposed to evaluate the model complexity, based on the Shtarkov NML code (Shtarkov, 1987), which provides strong theoretical guarantees (Rissanen, 2000).

It exploits the following NML distribution $\overline{P}_{nml}^{(n)}$ on $X^n$:

$$\overline{P}_{nml}^{(n)}(x^n) = \frac{P_{\widehat{\theta}(x^n)}(x^n)}{\sum_{y^n \in X^n} P_{\widehat{\theta}(y^n)}(y^n)} \tag{1}$$

where $\widehat{\theta}(x^n)$ is the model parameter that maximizes the likelihood of $x^n$.

The log of the denominator stands for the *parametric complexity* $COMP^{(n)}(\theta)$ of the model whereas the negative log of the numerator is the *stochastic complexity* of the data given the model. The sum of both terms provides the NML code. It is noteworthy that the

NML code is a one-part rather than two-part code: data is encoded with the help of all the model hypotheses rather than the best hypothesis.

The parametric complexity of the NML universal model with respect to a k-parameter exponential family model is usually approximated by $\frac{k}{2}\log\frac{n}{2\pi}$ (Grünwald, 2007). In the case of the multinomial distribution with $(m-1)$ free parameters, this gives $\frac{m-1}{2}\log\frac{n}{2\pi}$. A better approximation based on Rissanen's asymptotic expansion (Rissanen, 1996) is presented in (Kontkanen, 2009):

$$COMP_{nml}^{(n)}(\theta) = \frac{m-1}{2}\log\frac{n}{2\pi} + \log\frac{\pi^{m/2}}{\Gamma(m/2)} + o(1). \tag{2}$$

where $\Gamma(.)$ is the Euler gamma function. Still in (Kontkanen, 2009), a sharper approximation based on Szpankowski's approximation (Szpankowski, 1998) is presented. This last approximation, far more complex is very accurate w.r.t. $n$, with $o(\frac{1}{n^{3/2}})$ precision. We present below its first terms until $O(\frac{1}{\sqrt{n}})$.

$$COMP_{nml}^{(n)}(\theta) = \frac{m-1}{2}\log\frac{n}{2} + \log\frac{\sqrt{\pi}}{\Gamma(m/2)} + O(\frac{1}{\sqrt{n}}). \tag{3}$$

Finally, (Kontkanen and Myllymäki, 2007; Mononen and Myllymäki, 2008) propose an exact computation of the multinomial parametric complexity, at the expense of sophisticated algorithms with quasilinear computation time, or even sub-linear computation time in case of fixed precision.

## 2.2 Enumerative two-part MDL

We present the enumerative two-part code for multinomial distributions (see for example (Grünwald, 2007) Example 10.1 *Coding by Giving an Index*, also called *conditional two-part code*) using its NML interpretation (Boullé et al., 2016).

### 2.2.1 Enumerative interpretation

Given a sample size $n$, the number of tuples $(n_1, n_2, \ldots, n_m)$ such that $\sum_{j=1}^{m} n_j = n$ is $\binom{n+m-1}{m-1}$. The multinomial model parameters are encoded using a uniform prior

$$P\left(\theta = \left(\frac{n_1}{n}, \frac{n_2}{n}, \ldots, \frac{n_m}{n}\right)\right) = 1/\binom{n+m-1}{m-1},$$

leading to $L(\theta) = \log\binom{n+m-1}{m-1}$.

Second, the data $x^n$ is encoded given the $\theta$ parameter. The following likelihood is exploited, using a discrete probability distribution for encoding the finite size data sample $x^n$.

For $\theta \neq \left(\frac{n_1(x^n)}{n}, \frac{n_2(x^n)}{n}, \ldots, \frac{n_m(x^n)}{n}\right)$, we cannot encode the data and $P(x^n|\theta) = 0$.

For $\theta = \widehat{\theta}(x^n) = \left(\frac{n_1(x^n)}{n}, \frac{n_2(x^n)}{n}, \ldots, \frac{n_m(x^n)}{n}\right)$, the observed data is consistent with the model parameter and all the possible observable data are assumed to be uniformly distributed. The number of m-ary strings where the number of occurrences of outcome $j$ is $n_j$ is given by the multinomial coefficient $\frac{n!}{n_1!n_2!\ldots n_m!}$. Thus the probability of observing one particular m-ary string is $P(x^n|\widehat{\theta}(x^n)) = 1/\frac{n!}{n_1!n_2!\ldots n_m!}$. This gives a total code length of

$$L(\widehat{\theta}(x^n), x^n) = \log\binom{n+m-1}{m-1} + \log\frac{n!}{n_1!n_2!\ldots n_m!}, \tag{4}$$

defined only when $\theta = \widehat{\theta}(x^n)$.

Interestingly, this enumerative MDL approach results in the same code length as that obtained in (Hansen and Yu, 2001) using *Predictive Coding* or *Mixture Coding* with a uniform prior.

### 2.2.2 NML interpretation

Let us compute the NML parametric complexity of this enumerative code on the basis of the discrete likelihood. We call it the *enumerative parameter complexity* and denote $eCOMP$, to distinguish it from the standard NML parametric complexity.

Similarly to the NML approach, we compute the $eCOMP$ using the Shtarkov normalized maximum likelihood, based on the previously defined discrete likelihood.

$$\begin{aligned} eCOMP^{(n)}(\theta) &= \log \sum_{y^n \in X^n} P_{\widehat{\theta}(y^n)}(y^n), && (5)\\ &= \log \sum_{\{n_1+\ldots+n_m=n\}} \frac{n!}{n_1!n_2!\ldots n_m!} \left(1/\frac{n!}{n_1!n_2!\ldots n_m!}\right), && (6)\\ &= \log \binom{n+m-1}{m-1}. && (7) \end{aligned}$$

Interestingly, we find exactly the same complexity term as the coding length of the best hypothesis in the enumerative approach, that simply relies on counting the possibilities for the model parameters. This shows that the enumerative code is both a two-part and a one-part code, min-max regret optimal w.r.t. the NML approach and parametrization invariant (since the integration is independent of the parametrization). We have an exact formula for the complexity term, very simple to compute. Using Stirling's approximation $\log n! = n\log n - n + \frac{1}{2}\log 2\pi n + O(1/n)$, we get the following asymptotic approximation:

$$eCOMP^{(n)}(\theta) = (m-1)(\log n - \log(m-1) + 1) - \frac{1}{2}\log 2\pi(m-1) + O(\frac{1}{n}). \quad (8)$$

# 3 Hierarchical multinomial distributions

Let us introduce a hierarchical way of coding multinomial distributions. Let use assume that the $m$ outcomes are divided into two groups $A$ and $B$ of size $m^A$ and $m^B$, with $m = m^A + m^B$. Let $n^A$ and $n^B$ be the total number of occurrences for the outcomes in each group.

$n^A = \sum_{j=1}^{m_A} n_j,\ n^B = \sum_{j=m_A+1}^{m} n_j,\ n = n^A + n^B$.

Instead of defining a prior for the flat vector of probabilities of outcomes like in Section 2, we exploit the hierarchy of the parameters using a prior for the definition of the two groups of outcomes, then a prior for the sub-vector of probabilities within each group.

## 3.1 Hierarchical enumerative two-part MDL

Let us consider a hierarchical multinomial model with parameter $\theta^H = (\theta^R, \theta^A, \theta^B)$, were $\theta^R$ represents the parameter of a binomial model at the root of the hierarchy, and $\theta^A, \theta^B$ are the parameters of the sub-multinomial models for the groups $A$ and $B$ of outcomes.

We first encode the partition of the $m$ outcomes into two groups $A$ and $B$. We assume that any partition into at most two groups are equiprobable. The number of partition of $m$ values into $k$ groups is given by the Stirling number of the second kind $S(m,k)$. For bi-partitions, we have the following closed formula: $S(m,2) = 2^{m-1} - 1$. Overall, $S(m,1) + S(m,2) = 2^{m-1}$. Assuming uniform probabilities, the probability of one partition is thus $\frac{1}{2^{m-1}}$. This gives a code length of

$$(m-1)\log 2. \quad (9)$$

We use enumerative codes presented in Section 2.2 to encode both the binomial and multinomial distributions involved in the hierarchical multinomial model.

We then encode a binomial with $n$ occurrences and two outcomes: belonging to group $A$ or $B$. The code length for this binomial distribution is

$$L(\theta^R(x^n), x^n) = \log(n+1) + \log \frac{n!}{n^A!n^B!}. \quad (10)$$

Then, within each group, we encode the outcomes using two sub-multinomial distributions, for $n^A$ occurrences with $m^A$ outcomes and $n^B$ occurrences with $m^B$ outcomes. We get

$$\log \binom{n^A + m^A - 1}{m^A - 1} + \log \frac{n^A!}{n_1! n_2! \dots n_{m^A}!} \tag{11}$$

and

$$\log \binom{n^B + m^B - 1}{m^B - 1} + \log \frac{n^B!}{n_{m^A+1}! n_{m^A+2}! \dots n_{m^A+m^B}!}. \tag{12}$$

Adding all contributions and rearranging the terms of the binomial and multinomial coefficients, we get

$$\begin{aligned} L(\theta^H(x^n), x^n) &= (m-1)\log 2 && (13)\\ &\quad + \log(n+1) + \log \binom{n^A + m^A - 1}{m^A - 1} + \log \binom{n^B + m^B - 1}{m^B - 1} && (14)\\ &\quad + \log \frac{n!}{n_1! n_2! \dots n_m!}, && (15) \end{aligned}$$

As expected, we get the same likelihood term as the code length of the enumerative multinomial distribution presented in Section 2.2. Both approaches differ only by the code length of their parameters, that is their enumerative parameter complexity [2].

The enumerative parameter complexity $eCOMP^{(n)}(\theta^H)$ consists of the first four terms of formula 13. The first term encoded the partition of the outcomes into two groups, the second one the number of outcomes per group, the third and fourth ones the multinomial parameters within each group.

## 3.2 Hierarchical versus enumerative multinomial models

Let us introduce the difference of code length between the hierarchical and the enumerative multinomial models, that reduces to the difference of their enumerative parameter complexity.

$$\begin{aligned} \delta eCOMP^{(n)}(\theta^H, \theta) &= (m-1)\log 2 && (16)\\ &\quad + \log(n+1) + \log \binom{n^A + m^A - 1}{m^A - 1} + \log \binom{n^B + m^B - 1}{m^B - 1} && (17)\\ &\quad - \log \binom{n+m-1}{m-1}. && (18) \end{aligned}$$

Let evaluate this difference in the case where $m = 3$, with one first group $A = \{m_1\}$ containing a frequent outcome ($n_1 = 20$) and a second group containing two rare outcomes $B = \{m_2, m_3\}$ ($n_2 = n_3 = 1$). We have $m^A = 1, m^B = 2, n^A = 20, n^B = 2, n = 22$.

$$\begin{aligned} \delta eCOMP^{(n)}(\theta^H, \theta) &= 2\log 2 + \log 23 + \log \binom{20}{0} + \log \binom{3}{1} - \log \binom{24}{2}, && (19)\\ &= 2\log 2 + \log 23 + \log 3 - \log \frac{24 \times 23}{2}, && (20)\\ &= 0. && (21) \end{aligned}$$

In this special case, both approaches result in exactly the same code length. We can check that for a slightly less frequent outcome $m_1$ with $n_1 = 19$, we have $\delta eCOMP^{(n)}(\theta^H, \theta) \approx 0.0426$, and conversely with $n_1 = 21$, we have $\delta eCOMP^{(n)}(\theta^H, \theta) \approx -0.0408$ Therefore, even for multinomial models with only three outcomes, the hierarchical model obtains shorter code lengths for some data sets.

[2] In the paper, the term *parametric complexity* (COMP) is employed for the standard NML multinomial code, whereas the term *enumerative parameter complexity* ($eCOMP$) is used for the enumerative and hierarchical multinomial codes, both considered as two-part codes, for the part that encodes the model parameters.

## 3.3 Alternative hierarchical multinomial prior

An alternative prior is possible for the partition of the $m$ outcomes into two groups $A$ and $B$. The term $(m-1)\log 2$ is based on the enumeration of all the possible partitions of $m$ into two groups, and it exploits a Stirling number of the second kind. Instead, we can first choose the numbers of outcomes $(m^A, m^B), m^A \geq m^B$ in each group.

The number of such distinct pairs $(m^A, m^B)$ has been studied in combinatorics. An *integer partition* is a way of writing an integer $n$ as a sum of positive integers. For example, 4 can be partitioned in five distinct ways: $4, 3+1, 2+2, 2+1+1, 1+1+1+1$. With a restricted number of parts $k$, $p_k(n)$ is the number of partitions of $n$ into exactly $k$ parts. For example, 4 can be partitioned in two parts in two distinct ways: $3+1, 2+2$. In our problem, we are interested in partitioning $m$ as the sum of $m^A$ and $m^B$. We have

$$p_2(m) = \lfloor m/2 \rfloor. \tag{22}$$

Given $(m^A, m^B)$, the number of partitions $\Pi(m^A, m^B)$ of the $m$ outcomes into two groups $A$ and $B$ is given by the binomial coefficient $\binom{m}{m^A}$. For any $(m^A, m^B)$ such than $m^A \neq m^B$, this gives the exact number of partitions, but for $m^A = m^B$, $A$ and $B$ play the same role and the partitions are counted twice. Overall, we get

$$\Pi(m^A, m^B) = \binom{m}{m^A} \frac{1}{1 + \mathbb{1}_{\{m^A = m^B\}}} \tag{23}$$

Overall, assuming that any pair of sizes of the two groups are equiprobable, then that any partition of the $m$ outcomes on the groups of given sizes are equiprobable, we get the following code length

$$c(\Pi) = \log\lfloor m/2 \rfloor + \log \binom{m}{m^A} - \log\left(1 + \mathbb{1}_{\{m^A = m^B\}}\right). \tag{24}$$

For simplicity reasons, in the rest of the paper, we will use a tight upper bound of this code length:

$$c(\Pi) = \log m/2 + \log \binom{m}{m^A}. \tag{25}$$

Overall, we get the following enumerative parameter complexity.

$$\begin{aligned} eCOMP^{(n)}(\theta^{H'}) &= \log m/2 + \log \binom{m}{m^A} && (26) \\ &\quad + \log(n+1) + \log \binom{n^A + m^A - 1}{m^A - 1} + \log \binom{n^B + m^B - 1}{m^B - 1}. && (27) \end{aligned}$$

Compared to the previous enumerative parameter complexity (cf. Section 3.1), the difference is that the partition of the $m$ outcomes into two groups is encoded using a code length of $(m-1)\log 2$ in the first case and $\log m/2 + \log \binom{m}{m^A}$ in the second case.

## 3.4 Asymptotic difference of code length

Let us study the asymptotic difference of code length between:

- the hierarchical multinomial model (Section 3.1): $\theta^H$,
- its alternative version (Section 3.3): $\theta^{H'}$,
- the enumerative multinomial model (Section 2.2): $\theta$.

As seen before, all codes have the same stochastic complexity, so that we focus on their enumerative parameter complexity. Let us introduce the proportion $\alpha$ of occurrences in group $A$ ($\alpha = n^A/n$) and $\beta$ the proportion of outcomes in group $A$ ($\beta = m^A/m$).

For $\theta \in [0;1]$, let us introduce the entropy $H(\theta) = -\theta\log\theta - (1-\theta)\log(1-\theta)$ and the variance $\mathrm{var}(\theta) = \theta(1-\theta)$.

**Theorem 1.** *The difference of enumerative parameter complexity between the hierarchical multinomial model and its alternative version is*

$$\delta eCOMP^{(n)}(\theta^H, \theta^{H'}) = m(\log 2 - H(\beta)) - \frac{1}{2}\log m + \frac{1}{2}\log 2\pi \mathrm{var}(\beta) + O(1/m) \quad (28)$$

*Proof.* See Appendix A. □

As $0 \leq \beta \leq 1$, we have $0 \leq H(\beta) \leq \log 2$. Theorem 1 shows that the alternative hierarchical multinomial model gets the smaller enumerative parameter complexity, except when the groups $A$ and $B$ have about the same number of outcomes ($m^A \approx m/2$). For a given proportion $\beta$ of outcomes, the difference grow linearly with $m$. In the extreme case of two groups with the same number of outcomes ($\beta = 1/2$), the alternative model is dominated only by a small margin of $\frac{1}{2}\log m$.

**Theorem 2.** *The difference of enumerative parameter complexity between the hierarchical multinomial model and the enumerative one is*

$$\delta eCOMP^{(n)}(\theta^H, \theta) = m\left(\log 2 + H(\beta) - H(\alpha) + (\beta - \alpha)\log\frac{\alpha}{1-\alpha}\right) + \frac{1}{2}\log m \quad (29)$$

$$-\frac{1}{2}(2\log \mathrm{var}(\alpha) - \log \mathrm{var}(\beta) + \log 8\pi) \quad (30)$$

$$+\frac{m^2}{n+m}\frac{(\beta-\alpha)^2}{2\alpha(1-\alpha)} + O(\frac{m^3}{n^2}) \quad (31)$$

*Proof.* See Appendix A. □

Theorem 2 shows that the difference of enumerative parameter complexity between the hierarchical multinomial model and the enumerative one grows linearly with $m$, with a complex factor that depends on the proportion $\beta$ of outcomes and $\alpha$ of occurrences. However, the difference heavily depends on the absolute numbers $m$ of outcomes and $n$ of occurrences, and the asymptotic behavior is reached only for $n \gg m$. If we focus on this asymptotic behavior when $n$ goes to infinity, we get the simplified formula

$$\delta eCOMP^{(n)}(\theta^H, \theta) = m\left(\log 2 + H(\beta) - H(\alpha) + (\beta - \alpha)\log\frac{\alpha}{1-\alpha}\right) + O(\log m). \quad (32)$$

With the alternative prior, we have

$$\delta eCOMP^{(n)}(\theta^{H'}, \theta) = m\left(2H(\beta) - H(\alpha) + (\beta - \alpha)\log\frac{\alpha}{1-\alpha}\right) + O(\log m). \quad (33)$$

Figure 1 compares the enumerative parameter complexity of the hierarchical multinomial model and the enumerative one in the asymptotic case ($n \gg m$). When the number of outcomes and occurrences are balanced, the enumerative multinomial gets the smallest enumerative parameter complexity (blue regions in the center of the figures), whereas in the unbalanced case, the best model is the hierarchical one (red regions in the upper left and bottom right corner of each figure). Among the two variants of hierarchical multinomial models, the alternative one (right figure) is clearly the best one, with a far larger region of the parameter space where it dominates the enumerative multinomial model. For example, when the two groups have the same number of occurrences ($\alpha = 0.5$), the first hierarchical multinomial model is never better than the enumerative multinomial model, while the alternative hierarchical multinomial model is better when the proportion of outcomes in the smallest groups if beyond around 11% ($\beta \in [1; 0.11] \cup [0.89; 1]$). In the rest of the paper, we focus on the alternative hierarchical multinomial model.

## 3.5 Non-asymptotic difference of code length

Figure 2 focuses on the non-asymptotic case in the comparison between the enumerative parameter complexity of the alternative hierarchical multinomial model and the enumerative one. Like in Figure 1, the regions of the $(\alpha, \beta)$ parameter space are in blue when the enumerative multinomial code is better (balanced case) and in red when the hierarchical code is

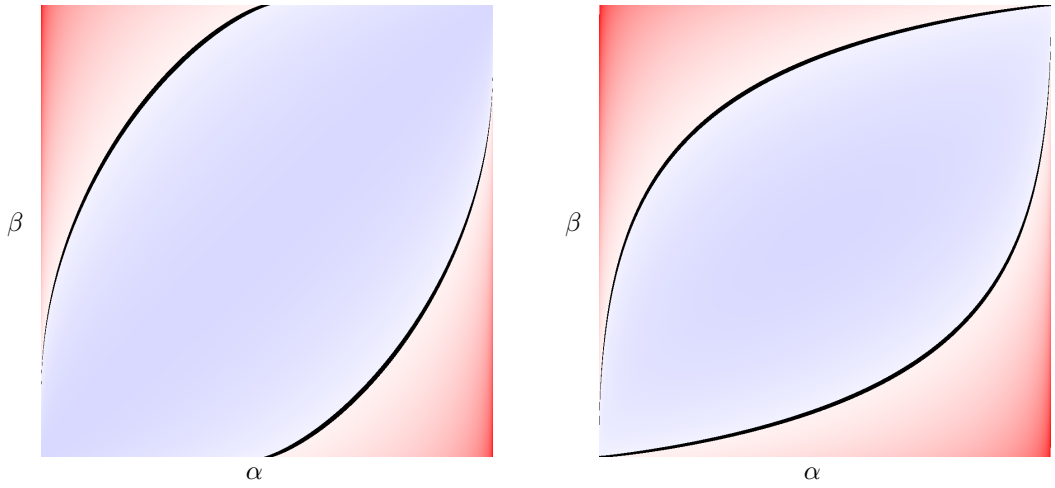



Figure 1: Comparison between the enumerative parameter complexity of the hierarchical multinomial model ($\theta^H$ on the left and $\theta^{H'}$ on the right) and the enumerative one.

better (unbalanced case). The black contour line represents the boundary where each model gets the same code length. The gray contour lines are draw every 10% in the ratio of the minimum of the two code lengths over the other one. The blue and red colors are normalized and represent the same ratio over all panels. Similarly to Figure 1, the regions of dominance of each model are represented by a blue region in the center or a red region in the upper left and bottom right corners. Altogether, the hierarchical multinomial model dominates the enumerative one in case of unbalanced distributions, when a majority of occurrences are related to a minority of outcomes.

The asymptotic behavior is empirically reached for $n/m > 10$. Even when the number of occurrences if about the same as the number of outcomes ($n \approx m$), the hierarchical multinomial model keeps large regions of dominance.

When the number of occurrences is far smaller than the number of outcomes ($n \ll m$), the enumerative multinomial model looks better almost everywhere in the $\alpha \times \beta$ parameter space. However, the figures are misleading: for $n \ll m$, the number of outcomes $j$ such that $n_j > 0$ is bounded by $m$. In this case, one can consider a first group containing all the data ($n^A = n$ and $m^A = m$) while the second group is empty. The interesting part of the $\alpha \times \beta$ parameter space reduces to $\beta \in [0; \frac{n}{m}] \cup [1 - \frac{n}{m}; 1]$, which is precisely the part of the figures where the hierarchical multinomial models exhibits an important region of dominance.

## 3.6 Discussion

In this section, we discuss several points related to the new hierarchical-based code for multinomial models.

### 3.6.1 Combining the hierarchical and enumerative multinomial models

One can easily combine the hierarchical and enumerative multinomial models, by first selecting one of then using a binary choice ($\log 2$ to encode this choice), then encoding the multinomial parameters using either the hierarchical or the enumerative multinomial distribution. Using this combined model, we get a code length that is close (with a $\log 2$ margin) to the minimum of the two code lengths. Let us remember that the enumerative code is a NML code, with a constant regret. The combined code is no longer NML, but its regret is never larger than the NML one with margin of $\log 2$, but there are regions of the parameter space where its regret can be far smaller.

We have investigated a basic hierarchy with a parameter tree of depth one and two leave nodes. This could be extended to parameter trees of any depth with any number of branches per tree node. This is left for future work.

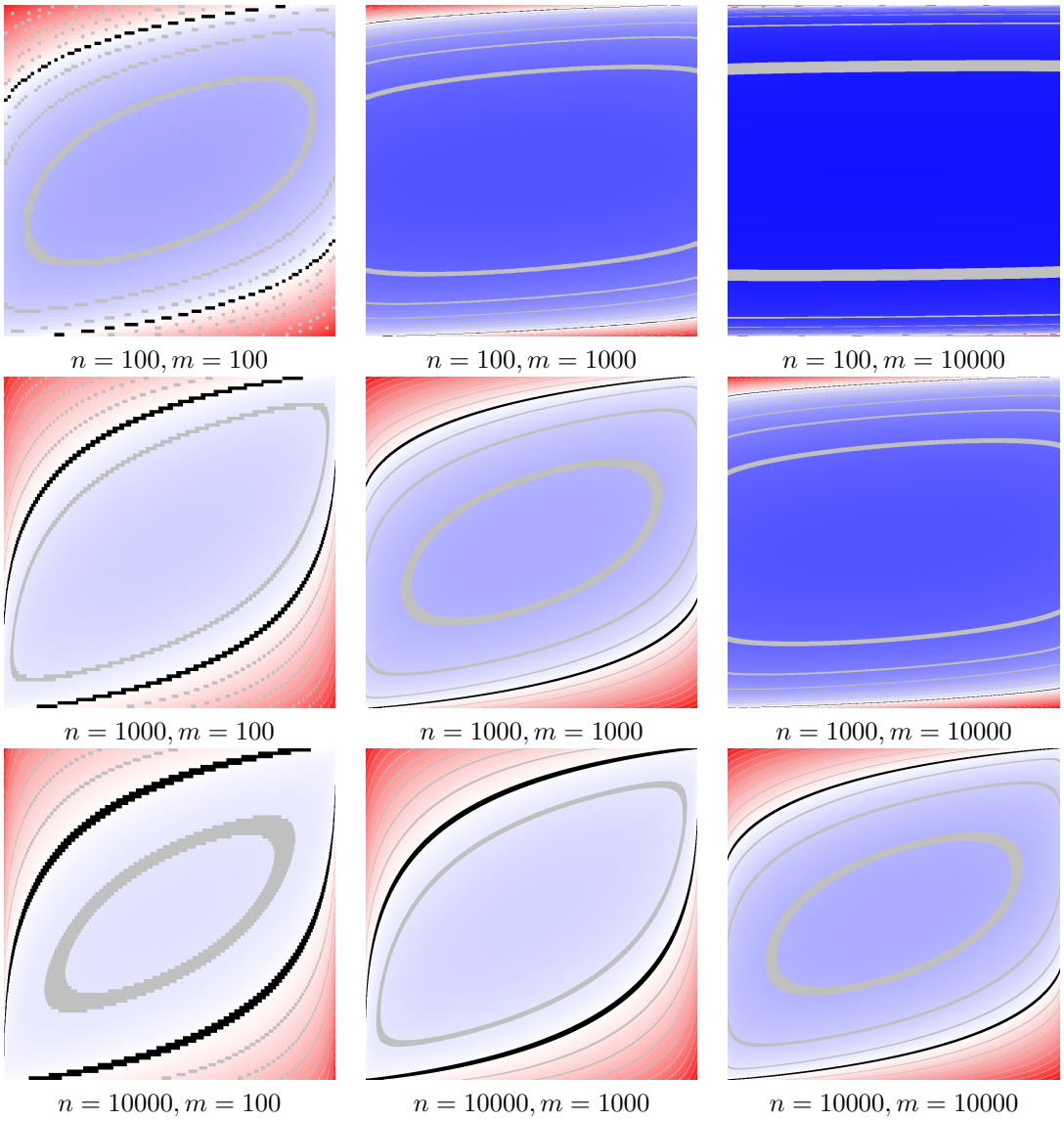



Figure 2: Comparison between the enumerative parameter complexity of the alternative hierarchical multinomial model ($\theta^{H'}$) and the enumerative one, in the non-asymptotic case. The $(X, Y)$ axis of each plot are $(\alpha, \beta)$, as in Figure 1.

### 3.6.2 Selecting the best hierarchical multinomial model

The proposed hierarchical multinomial model is actually a family of models. For a given data set with $m$ outcomes and $n = \sum_j^m n_j$ occurrences, every partition of the $m$ outcomes into two groups can be considered and the one resulting in the smallest code length can be selected. This raises a difficult optimization problem, as the number of bi-partitions of the outcomes is $2^{m-1}$. However, as the parameter complexity of the hierarchical model depends only on $n, n^A, n^B, m, m^A, m^B$ and as the figures in sections 3.4 and 3.5 show a behavior that depends on the unbalance rate of groups, this suggests that the optimal solution should consist in one group containing the least frequent outcomes and the other one containing the most frequent outcomes. In other words, we assume that all the outcomes in the group containing the most frequent outcomes should be more frequent that any outcome in the other group. Proving this assumption is left for future work.

We can then sort the outcomes by decreasing frequency, then perform a linear search

by considering all the bi-partitions where the first group contains the $k, 0 \leq k \leq m$ least frequent outcomes. This simple optimization algorithm requires $\mathrm{O}(n)$ time to compute the frequencies of the outcomes, then $\mathrm{O}(m \log m)$ time to sort the outcomes and finally $\mathrm{O}(m)$ time to compute all the code lengths related to the bi-partitions of the outcomes and select the best one. Overall, this optimization algorithm has a $\mathrm{O}(n + m \log m)$ time complexity to retrieve the best hierarchical multinomial model.

### 3.6.3 Relation with alternative codes

The multinomial distribution is one of most simple and well studied distributions, with either its NML or enumerative code (see Section 2). Although there is no obvious structure in its parameter space, the hierarchical code suggested in this paper manages to produce a new code that is more efficient in large regions of the parameter space, in case of unbalanced multinomial distributions.

Actually, this code can be related to *luckiness NML* (De Rooij and Grünwald, 2009; Grünwald, 2007), with codes that are never much worse, but sometimes much better than the NML code. The hierarchical code benefits from unbalanced distributions to better compress the data. A more Bayesian approach could be considered, by using a beta-binomial distribution for the choice of the frequent or rare events in the hierarchical schema, or more generally a Dirichlet-multinomial distribution over the outcomes of the multinomial distribution. One could achieves better codes using a particular form of luckiness NML, which relates to extending the Shtarkov distribution (Shtarkov, 1987) with an informative prior. However, this raises novel problems, such as the choice of hyper-parameters for the beta-binomial or Dirichlet-multinomial distribution, as well a hard calculation problems, to compute or even approximate the code length of these extended NML codes.

The code suggested in this paper exploits enumerative codes as building blocks for encoding the multinomial distribution at each stage of the hierarchy. Alternatively, standard NML codes could be used in each leaf of the hierarchy. This is not considered in the paper, since enumerative codes are at least as efficient as NML codes (Boullé et al., 2016), while being far simpler to compute (see Section 2.1). As choosing the best hierarchical code relies on a search algorithm (cf. Section 3.6.2) with at least linear complexity, the enumerative codes have the advantage to remain usable in practice for large scale data and models.

### 3.6.4 Interest for model selection in applications

The proposed code for hierarchical multinomial models looks interesting for unbalanced distributions, when one part of the outcomes are significantly more frequent than the other ones. In many domains, such unbalanced distributions are rather usual (Newman, 2005), as for example in text mining where the distribution of words tends to follow a Zipf's law. In these domains, modeling methods have been proposed, that exploit the multinomial distributions as a building block for more complex models. This is the case for example with mixtures of multinomial topic models in text mining (Mei et al., 2007), community detection in social networks (Sun et al., 2007), analysis of gene expression data via biclustering algorithms (Madeira and Oliveira, 2004), analysis of call detail records in large telecommunication networks (Guigourès et al., 2015). In Section 4, we present a preliminary study of the potential benefit of using the proposed code for model selection in case of joint density estimation.

# 4 Application to Joint Density Estimation

Joint density estimation (Scott, 2015) has been studied using a variety of approaches, including histogram based methods (Lugosi and Nobel, 1996) or co-clustering based methods (Seldin and Tishby, 2010) that are suitable for data visualization. In the case of peaked densities, which naturally occur for example in regression problems, histogram based methods are likely to output a set of bins with unbalanced frequencies. This application is then a good candidate to evaluate the interest of using hierarchical codes.

In this section, we present a model family for joint density estimation between two numerical variables, which leverages one of the simplest models that relies on the multinomial

distribution. We suggest an application of these models for exploratory analysis, with detection of correlation across variables and a visualization method. We then illustrate the behavior of the method on some real and artificial data sets. Finally, we study the interest of using the hierarchical multinomial code to select better models.

## 4.1 Joint Density Estimation

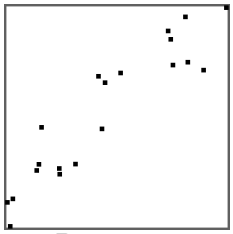
We present a specialization of the functional data clustering method (Boullé, 2012) to the case of one single function, that reduces to density estimation between two numerical variables.

### 4.1.1 Data grid models

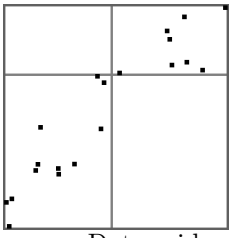
Let $X$ and $Y$ be two numerical variables and $(x, y)^n$ a data set of $n$ points. We choose a model family that describes the ranks of the values in the data set rather than the values themselves. Therefore, the models are invariant w.r.t. any monotonous transformation of the values of $X$ and $Y$ and robust w.r.t. atypical values (outliers). We thus use a rank based transformation of the values of $X$ and $Y$, leading to integer valued variables.

We consider joint density estimation models $M$ where the numerical variables $X$ and $Y$ are partitioned into $I$ and $J$ intervals. The Cartesian product of these two partitions forms a partition of the whole data set into a *data grid* containing $G = I \times J$ cells. Then, the $n$ points are distributed on the $G$ cells according to a multinomial distribution $\theta_{IJ} = \{\theta_{ij}\}_{1 \leq i \leq I, 1 \leq j \leq J}$. A data grid model $M$ is entirely defined by the parameters $I$, $J$ and $\theta_{IJ}$. It can be interpreted as a piecewise-constant (per cell) joint density estimation of the ranks of variables $X$ and $Y$. It is noteworthy that only truly bivariate ($I > 1, J > 1$) models make sense for joint density estimation (see Section 4.1.3).

### 4.1.2 Selection of the best model



Data set points



Data grid

| | | |
|---|---|---|
| $n_{12} = 0$ | $n_{22} = 8$ | $n_{.2} = 8$ |
| $n_{11} = 12$ | $n_{21} = 0$ | $n_{.1} = 12$ |
| $n_{1.} = 12$ | $n_{2.} = 8$ | $n = 20$ |

Data frequencies

Figure 3: Toy data set.

Figure 3 shows a toy data set and gives an example of a data grid with $2 \times 2$ cells, with the notations for all the data frequencies introduced in this section.

We do not want to make any strong assumption to encode $I$ and $J$, and use the universal prior for integer (Rissanen, 1983), which is essentially as close to uniform as possible. We use the enumerative multinomial code (Section 2.2) to encode $\theta_{IJ}$.

The obtained enumerative parameter complexity of $M$ is then

$$eCOMP^{(n)}(I, J, \theta_{IJ}) = \log^\star I + \log^\star J + \log \binom{n + IJ - 1}{IJ - 1}. \tag{34}$$

Let $n_{ij}$ be the frequencies per cell in the data set, $\sum_i \sum_j n_{ij} = n$, Using the enumerative multinomial code, the cell frequencies can be encoded by

$$\log \frac{n!}{\prod_{i=1}^{I} \prod_{j=1}^{J} n_{ij}!}. \tag{35}$$

The frequencies $n_{i.}$ per interval of $X$ and $n_{.j}$ per interval of $Y$ can be obtained by summation over the cells according to

$$n_{i.} = \sum_{j=1}^{J} n_{ij}, \quad n_{.j} = \sum_{i=1}^{I} n_{ij}. \tag{36}$$

Given the model parameters, the data is completely encoded when the ranks of the points are encoded. As the intervals on $X$ and $Y$ are ordered, we need to encode the ranks of the points within each interval to complete the encoding of the data. We assume that within each interval of size $n_{i.}$, all the $n_{i.}!$ possible rankings are equiprobable. We then need $\sum_{i=1}^{I} \log n_{i.}!$ to encode the ranks of the points for variable $X$, and $\sum_{j=1}^{J} \log n_{.j}!$ for variable $Y$.

Overall, the code length of the proposed code is

$$\begin{aligned} L(\widehat{\theta_{IJ}}((x,y)^n),(x,y)^n) &= \log^\star I + \log^\star J + \log \binom{n+IJ-1}{IJ-1} \\ &\quad + \log n! - \sum_{i=1}^{I}\sum_{j=1}^{J} \log n_{ij}! + \sum_{i=1}^{I} \log n_{i.}! + \sum_{j=1}^{J} \log n_{.j}!\,. \end{aligned} \tag{37}$$

The first line stands for the enumerative parameter complexity and is mainly a multinomial model indexed by the size $I \times J$ of the data grid. The second line represents the stochastic complexity: it consists in a multinomial term for the distribution of the points in the cells, then factorial terms for the ranking of the points within each interval of $X$ and $Y$. For the toy data set of Figure 3, the $2 \times 2$ data grid gets a shorter coding length than the $1 \times 1$ data grid:

$$\begin{aligned} L(\widehat{\theta_{11}}((x,y)^n),(x,y)^n) &= 2\log^\star 1 + 2\log 20!\,, \\ &\approx 86.78, \\ L(\widehat{\theta_{22}}((x,y)^n),(x,y)^n) &= 2\log^\star 2 + \log \binom{23}{3} + \log 20! - (\log 12! + \log 0! + \log 0! + \log 8!) \\ &\quad + (\log 12! + \log 8!) + (\log 12! + \log 8!), \\ &\approx 83.90. \end{aligned}$$

The best joint density estimation is selected by optimizing criterion (37). The optimization algorithm is described in (Boullé, 2012) [3], with a time complexity of $O(n\sqrt{n}\log n)$. Mainly, the algorithm is a greedy bottom-up heuristic that starts with fine grained intervals and iteratively merges the adjacent intervals that best improve the criterion.

### 4.1.3 Special cases

We study the behavior of the code length in the case of some particular models or data sets. We show that the best selected model reduces to one single cell in case of two independent variables, whereas the correlation between two identical variables is retrieved by approximating the joint density between the ranks of variables, with a precision that increases with the size of the data set.

**Null model.** Let us consider the null model $M_\emptyset$ ($I = 1, J = 1, \theta_{11}$) that corresponds to the case of one single interval for $X$ and $Y$, resulting in one single cell. We get

$$L(M_\emptyset((x,y)^n),(x,y)^n) = 2\log^\star 1 + 2\log n!. \tag{38}$$

The obtained code length mainly corresponds to encoding the ranks of $X$ and $Y$, independently for each variable. Compared to the null model, a truly bivariate model ($I > 1, J > 1$) encodes the two variables jointly, by encoding the correlation pattern using a data grid, then

[3] Sofware available at www.khiops.com

encoding each variable using the correlation pattern. In case of correlated variables, its coding length is below that of the null model. However, it is lower bounded by half the coding length of the null model, as shown below.

$$
\begin{aligned}
L(\widehat{\theta_{IJ}}((x,y)^n),(x,y)^n) &= \log^\star I + \log^\star J + \log \binom{n+IJ-1}{IJ-1} && (39)\\
&\quad + \log n! - \sum_{i=1}^{I}\sum_{j=1}^{J} \log n_{ij}! + \sum_{i=1}^{I} \log n_{i.}! + \sum_{j=1}^{J} \log n_{.j}!\,, \\
&> \log^\star I + \log n! + \sum_{i=1}^{I} \log \frac{n_{i.}!}{n_{i1}! n_{i2}! \ldots n_{iJ}!}, && (40)\\
&> \frac{L(M_\emptyset((x,y)^n),(x,y)^n)}{2}. && (41)
\end{aligned}
$$

**Maximal model.** Let us consider the maximal model $M_{max}$ ($I = n, J = n, \theta_{nn}$) that corresponds to the finest possible model, with one interval per value for $X$ and $Y$, resulting in $n \times n$ cells. In this case, the stochastic complexity part of the code length is null and the code length reduces to its enumerative parameter complexity part. For $n > 1$, we get

$$
\begin{aligned}
L(M_{max}((x,y)^n),(x,y)^n) &= 2\log^\star n + \log \binom{n+n^2-1}{n^2-1}, && (42)\\
&> 2\log^\star n + (n^2-1)\log n - \log n!, && (43)\\
&> L(M_\emptyset((x,y)^n),(x,y)^n). && (44)
\end{aligned}
$$

The obtained code length is always larger than that of the null model. Therefore, the finest model, with a perfect fit of the training data, can never be selected.

**Univariate model.** Let us consider a univariate model ($I = 1, J > 1, \theta_{1J}$), where there is one single interval for one of the variables and several for the other one. We obtain

$$
\begin{aligned}
L(\widehat{\theta_{1J}}((x,y)^n),(x,y)^n) &= \log^\star 1 + \log^\star J + \log \binom{n+J-1}{J-1}, && (45)\\
&\quad + \log n! - \sum_{j=1}^{J} \log n_{.j}! + \log n! + \sum_{j=1}^{J} \log n_{.j}!, && (46)\\
&= L(M_\emptyset((x,y)^n),(x,y)^n) && (47)\\
&\quad + \log^\star J - \log^\star 1 + \log \binom{n+J-1}{J-1}. && (48)
\end{aligned}
$$

The obtained code length is always larger than that of the null model. Therefore, the univariate model, which cannot describe any correlation pattern between the variables, can never be selected.

**Independent variables.** Let us assume that $X$ and $Y$ are independent and that the data set is partitioned in $I$ and $J$ intervals of equal frequency. Asymptotically, we get $n_{i.} \approx n/I$, $n_{.j} \approx n/J$, $n_{ij} \approx n/IJ$. Using the Stirling approximation $\log n! = n(\log n - 1) + O(\log n)$, we get

$$
\begin{aligned}
L(\widehat{\theta_{IJ}}((x,y)^n),(x,y)^n) &\approx \log^\star I + \log^\star J + \log \binom{n+IJ-1}{IJ-1} && (49)\\
&\quad + \log n! - \sum_{i=1}^{I}\sum_{j=1}^{J} \log \frac{n}{IJ}! + \sum_{i=1}^{I} \log \frac{n}{I}! + \sum_{j=1}^{J} \log \frac{n}{J}!, && (50)\\
&\approx \log^\star I + \log^\star J + \log \binom{n+IJ-1}{IJ-1} && (51)\\
&\quad + n(\log n - 1) - n(\log n - \log IJ - 1) && (52)\\
&\quad + n(\log n - \log I - 1) + n(\log n - \log J - 1) + O(\log n), && (53)\\
&\approx \log^\star I + \log^\star J + \log \binom{n+IJ-1}{IJ-1} + 2\log n! + O(\log n) && (54)
\end{aligned}
$$

Although this relies on an approximation, this shows that in case of two independent variables, any detailed model (with $I > 1, J > 1$) should asymptotically have a code length larger than that of the null model. Extensive experiments, not reported here, show that the null model is actually almost always the selected one in case of two independent variables.

**Identical variables.** Let us assume that $X = Y$ and that we partition the data set using the same number $I = J$ of intervals for both variables. In this case, all the diagonal cells have the same frequency than that of their related $X$ or $Y$ interval and all the non-diagonal cells are empty:

$$
\begin{aligned}
n_{i\cdot} = n_{\cdot i} = n_{ii}, \qquad & \forall i, 1 \leq i \leq I, && (55)\\
n_{ij} = 0, \qquad & \forall i, 1 \leq i \leq I, \forall j, 1 \leq j \leq I, i \neq j. && (56)
\end{aligned}
$$

We obtain

$$
L(\widehat{\theta_{II}}((x,x)^n),(x,x)^n) = 2\log^\star I + \log \binom{n+I^2-1}{I^2-1} + \log n! + \sum_{i=1}^{I} \log n_{i\cdot}! \qquad (57)
$$

For a given $I > 1$, let us analyze the case of a model with equal frequency intervals ($n_{i\cdot} \approx n/i$). Using the Stirling approximation, we get

$$
\begin{aligned}
L(\widehat{\theta_{II}}((x,x)^n),(x,x)^n) &= (I^2-1)\log n + n(\log n - 1) + I\frac{n}{I}(\log \frac{n}{I} - 1) + O(\log n), && (58)\\
&= (I^2-1)\log n + 2n(\log n - 1) - n\log I + O(\log n), && (59)\\
&= L(M_\emptyset((x,y)^n),(x,y)^n) + I^2 \log n - n \log I + O(\log n). && (60)
\end{aligned}
$$

For $I > 1$ and $n$ large enough, we get $I^2 \log n < n \log I$, so that the obtained code length gets smaller than that of the null model. Therefore, in case of identical variables, the diagonal models are preferred to the null model, meaning that the high correlation between $X$ and $Y$ is asymptotically retrieved by the joint density estimation method, whatever be the precision (given by the number $I$ of intervals).

## 4.2 Correlation detection and visualization for exploratory analysis

Density estimation is a key feature, used as a building block in many statistical models, for example to compute the conditional probability tables in a Bayesian networks (Silander et al., 2010; Guo et al., 2017). In this section, we suggest an application of our joint density estimation method for exploratory analysis, with a criterion to evaluate the correlation between two variables and a visualization method.

### 4.2.1 Correlation criterion

Given two numerical variables $X$ and $Y$ and a data set of size $n$, the null model $M_\emptyset$ (cf. Formula 38) correspond to the encoding of the values of $X$ and $Y$ independently. As $M_\emptyset$ encodes the rank of the variables, not their values, its coding length of $L(M_\emptyset)$ depends only on $n$, not on $X$ or $Y$.

Let $\widehat{M}(X,Y)$ be best joint density estimation model between $X$ and $Y$ inferred from the data. We suggest the following $0-1$ normalization of the coding length of $\widehat{M}(X,Y)$ to evaluate the correlation between $X$ and $Y$:

$$Level(X,Y) \quad = \quad 1 - \frac{L(\widehat{M}(X,Y))}{L(M_\emptyset)}. \tag{61}$$

The Level is 0 in case of independent variables, where the best model is no better than the null model, and it is always below 1. It can be interpreted as a compression gain compared to the null model. In our case of joint density models between two variables, the analysis of the null model in Section 4.1.3 provides the following tighter upper bound:

$$0 \leq Level(X,Y) < \frac{1}{2}. \tag{62}$$

In a data set with a set of variables, we can then evaluate the Level of each pair of numerical variables to identify the most correlated ones.

### 4.2.2 Visualization method

Beyond the criterion introduced previously, it is interesting to visualize the correlation pattern between two variables in an exploratory analysis application. We suggest to exploit the mutual information (Cover and Thomas, 1991) between the ranks of the variables to do so.

Given two numerical variables $X$ and $Y$, the best joint density estimation model $\widehat{M}(X,Y)$ provides a piecewise-constant joint probability estimator of the ranks of the variables, with

$$prob(rank(X) \in \text{Interval}_i, rank(Y) \in \text{Interval}_j) \quad = \quad \frac{n_{ij}}{n}, \tag{63}$$

$$prob(rank(X) \in \text{Interval}_i) \quad = \quad \frac{n_{i.}}{n}, \tag{64}$$

$$prob(rank(Y) \in \text{Interval}_j) \quad = \quad \frac{n_{.j}}{n}. \tag{65}$$

We can then estimate the mutual information $MI(X,Y)$ between the ranks of $X$ and $Y$:

$$MI(X,Y) \quad = \quad \sum_{i=1}^{I}\sum_{j=1}^{J} \frac{n_{ij}}{n} \log \frac{\frac{n_{ij}}{n}}{\frac{n_{i.}}{n}\frac{n_{.j}}{n}}, \tag{66}$$

$$= \quad \sum_{i=1}^{I}\sum_{j=1}^{J} \frac{n_{ij}}{n} \log \frac{n\, n_{ij}}{n_{i.} n_{.j}}. \tag{67}$$

Let $MI_{ij}(X,Y) = \frac{n_{ij}}{n} \log \frac{n\, n_{ij}}{n_{i.} n_{.j}}$ be the contribution of cell $(i,j)$ to the mutual information. Whereas the mutual information is 0 in case of independent variables and greater than 0 in case of correlated variables, the contribution $MI_{ij}(X,Y)$ of each cell can be either positive, if the frequency in the cell is above that expected in case of independent variables, or negative otherwise. We propose to visualize the correlation between two variables by drawing a bivariate plot of the data grid cells, using the red color in case of cell frequency higher than expected, the blue color in case of in case of cell frequency lower than expected, and the white color in case of cell frequency as expected or null. This is illustrated in next section.

## 4.3 Illustration on real and artificial data sets

We illustrate the behavior of the density estimation method on real and artificial data sets.

### 4.3.1 Appliances energy prediction data set

The appliance energy prediction data set (Candanedo et al., 2017), available from UCI Irvine (Blake and Merz, 1996), relates to a problem of prediction of the energy use of appliances in a building. Data include measurements of temperature and humidity sensors from a wireless network, weather from a nearby airport station and recorded energy use of lighting fixtures. The data was logged every 10 min during 4.5 months. The data set includes 19,735 record and 31 variables:

- appliance and light energy consumptions,
- temperature and humidity (*T1, ..., T9, RH_1, ..., RH_9*) per room of the building,
- six weather measurements (temperature, humidity, pressure, ...),
- three features extracted from the timestamps (number of second since midnight (*NSM*), day of week, weekend),
- two random variables, used to calibrate feature selection methods.

In (Candanedo et al., 2017), an exploratory analysis is performed using bivariate scatter plots and computing the Pearson correlation coefficient, which is a measure of the linear dependence between two variables. Similarly, we apply our joint density estimation method to the $465 = 31 * 30/2$ pairs of numerical variables of the data set. We obtain 404 correlated pairs of variables, whose *Level* is greater than 0. All the pairs involving one of the two random variables get $Level = 0$, meaning that the method is robust to noise, without needing any calibration. But surprisingly, the two random variables are highly correlated, with $Level = 0.204$ and a $90 \times 90$ grids of cells, with all the points lying in the diagonal cells. Actually, after inspecting the data, the two variables have exactly the same values: they may have been generated using the same random seed.

Focusing on the application variables, we use the *Level* criterion to sort the pairs by decreasing correlation. The *Level* criterion provides a non parametric evaluation of the correlation between any two variables, contrary to the Pearson correlation coefficient that is blind to non-linear dependence. The first family of correlated pairs involves variables related to temperature, followed by variables related to humidity. Then, there is long tail of pairs with decreasing correlation, and a variety of correlation patterns.

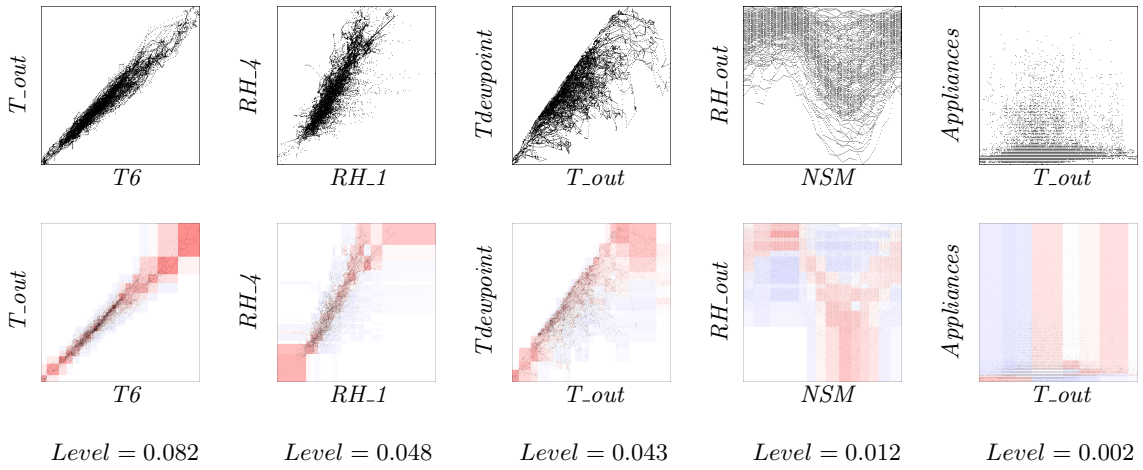



Figure 4: Scatter plots and density estimation models for the 1st, 10th, 21st, 182nd, 331st correlated pairs of variables of the appliances energy prediction data set.

Figure 4 displays the scatter plot and the density estimation model for the pairs ranked 1st, 10th, 21st, 182nd, 331st. The visualization method allows to summarize the joint density per pair of variables, with the red cells including more instances that expected in case of independent variables, and conversely for the blue cells. When the number of points is large and the pattern noisy, the visualization allows an easier understanding of the correlation pattern, that may be unclear using the scatter plot only. Note that in the distribution tails,

few points per cell need large boxes to be represented on the scatter-plots: the interpretation of the importance of the correlation patterns should be based on the intensity of the colors rather than the size of the boxes. The first pair on the left has a quasi-linear correlation pattern. The two next pairs have close values of *Level*, but with different correlations patterns. The two last pairs on the right have complex non linear correlation patterns.

Overall, the exploratory analysis performed using the joint density estimation method allows to reliably and accurately estimate the correlation between all the pairs of variables and to bring many insights using the Level criterion and the visualization method.

### 4.3.2 Challenging artificial data sets

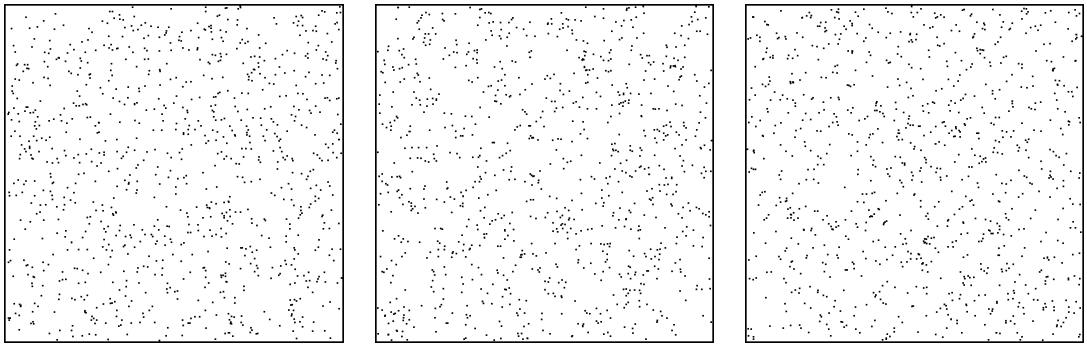



Figure 5: Three artificial data sets.

Figure 5 shows three artificial data set $D_1, D_2, D_3$ containing $n = 1000$ points generated on the $[0; 1] \times [0; 1]$ square, where it looks challenging to decide whether the $X$ and $Y$ variables are independent or correlated, with no simple pattern emerging from a visual inspection. The joint density estimation method is applied to these data sets, and the resulting models are displayed in Figure 6, with the boundaries of the $I \times J$ cells of each retrieved model. The pattern of each generative model is retrieved by the joint density estimation method.

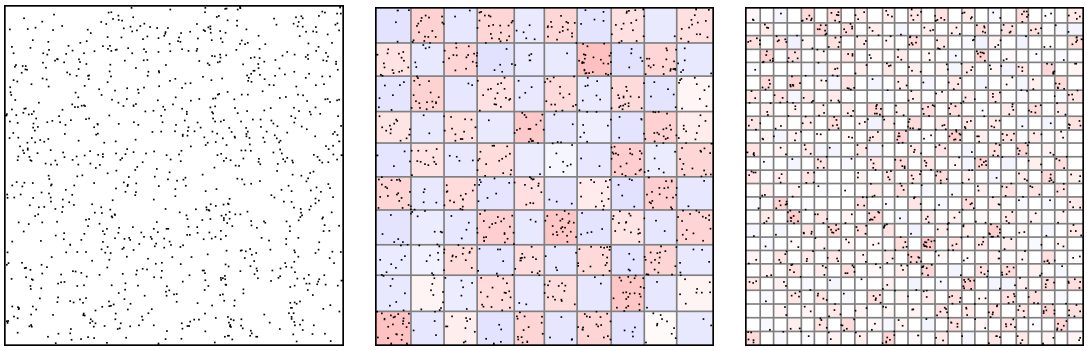
In the $D_1$ data set, the points are generated with independent $X$ and $Y$ coordinates, and the method retrieves one single cell, meaning that the variables are independent. This confirms empirically that the method is resilient to noise.

In the $D_2$ data set, the points are generated according to a noisy $10 \times 10$ checkerboard, where the probabilities of generating points in the checkerboard squares are alternatively $p = 0.75$ and $p = 0.25$. With the help of the cell boundaries drawn in Figure 6, one can actually observe that the density of points is alternatively high or low in the squares of the checkerboard, whereas this pattern was hardly discernible from visual inspection of Figure 5.

In the $D_3$ data set, the points are generated according to a noiseless $25 \times 25$ checkerboard, where the probabilities of generating points in the checkerboard squares are alternatively $p = 1$ and $p = 0$. In the generated data set with $n = 1000$ and $25 \times 25 = 625$ squares, there so few points that the complex pattern is hard to uncover. However, the method selects this complex joint density model, as it is more probable than the null model.

It is noteworthy that the true patterns are not perfectly retrieved with so few points. The obtained Level is 0.001 for the $D_2$ data set and 0.004 for the $D3$ data set. This means that compared to the null model (with 1000 points), the gain in compression is only 1‰ for $D2$ and 4‰ for $D3$. For $D2$ in particularly, the boundaries of the checkerboard are not perfectly retrieved and the inferred densities per cell are sometimes in reverse order of the true ones (see missing red cells in the $D2$ retrieved checkerboard of Figure 6). With fewer points, the hidden patterns are not discernible from noise and the null model is retrieved. With more points, the true patterns are easier to retrieve, both by visual inspection and using the joint density estimation method. This is illustrated in Appendix B, where the same experiment is performed with $10,000$ points, resulting in very accurate retrieved checkerboard patterns.

Altogether, the method is both robust, with a high resilience to noise, and very accurate, with the ability to detect complex patterns with very few points.



$D_1$ $D_2$ $D_3$

Figure 6: Three artificial data sets and the retrieved density estimation models.

## 4.4 Shorter code lengths using the hierarchical code

Instead of using the enumerative code for the multinomial distribution of the $n$ points over the $G$ cells of the data grid, let us use the alternative hierarchical code introduced previously.

### 4.4.1 Code length using the hierarchical multinomial model

Let us split the $G$ cells of the data grid into two groups $A$ and $B$, containing the most frequent and the least frequents cells. For example, $A$ contains all the cells with at most one point and $B$ contains all the empty cells. Let $m^A = |A|$ and $m^B = |B|$ be the number of cells in each groups, $n^A$ and $n^B$ be the number points in each group.

In formula (37), we replace the enumerative multinomial prior term by it hierarchical alternative (cf. Section 3.3) and obtain

$$
\begin{aligned}
\widehat{L(\theta_{IJ}^{HAB}((x,y)^n),(x,y)^n)} &= \log^\star I + \log^\star J && (68)\\
&\quad + \log m/2 + \log \binom{m}{m^A} + \log(n+1) && (69)\\
&\quad + \log \binom{n^A + m^A - 1}{m^A - 1} + \log \binom{n^B + m^B - 1}{m^B - 1} && (70)\\
&\quad + \log n! - \sum_{i=1}^{I}\sum_{j=1}^{J} \log n_{ij}! + \sum_{i=1}^{I} \log n_{i.}! + \sum_{j=1}^{J} \log n_{.j}!. && (71)
\end{aligned}
$$

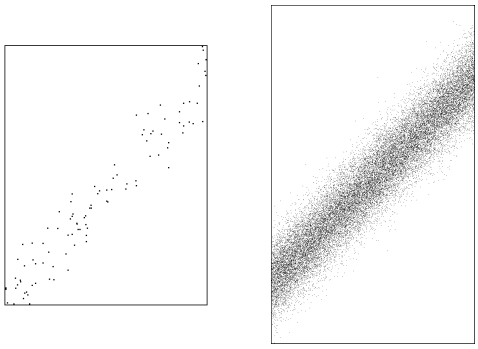
Compared to formula (37), the enumerative parameter complexity (three first lines) has changed while the stochastic complexity (last line) if the same. The first line stands for the choice of the numbers of intervals for $X$ and $Y$. The second line represents the partition of the cells into two groups and the number of points per groups. The third line represents the multinomial distribution of the points on the cells, for each group.

As suggested in Section 3.6.1, we use the combined code choosing the best among the enumerative code and the hierarchical code. In formula 72, the first $\log 2$ term stands for the choice between the two codes, with equal probability. Then, the shorter code length is used, with the enumerative code in the left operand and the hieratchical code in the right operand of the min(,) term. The hierarchical code is optimized by applying the algorithm presented in Section 3.6.2 on all $\{A, B\}$ partitions of the outcomes of the multinomial distribution, where

the outcomes in $A$ are more frequent than those in $B$:

$$L(\widehat{\theta^C_{IJ}}((x,y)^n),(x,y)^n) =$$
$$\log 2 + \min\left(L(\widehat{\theta_{IJ}}((x,y)^n),(x,y)^n), \min_{\{A,B\}}\left(L(\widehat{\theta^{HAB}_{IJ}}((x,y)^n),(x,y)^n))\right)\right). \quad (72)$$

#### 4.4.2 Case of two identical variables with Gaussian noise



$n = 100$ $n = 100,000$

Figure 7: Data sets for $Y = X + \epsilon$.

We first study the potential benefit of using the hierarchical multinomial model for joint density estimation. To understand the relation between the degree of unbalanced data, the size of the data set and the accuracy of the density estimation, we exploit an artificial data set consisting on two almost identical variables. Let $X$ be a random variable uniformly distributed on $[0;1]$ and $Y$ be equal to $X$ with an additional Gaussian noise.

$$X \sim \mathcal{U}([0;1]), \quad (73)$$
$$Y = X + \epsilon, \quad (74)$$
$$\epsilon \sim \mathcal{N}(0,\sigma). \quad (75)$$

The $\sigma$ parameter allows to browse a wide range of cases, from independence between the variables when $\sigma \to \infty$ to functional dependance when $\sigma \to 0$. Figure 7 displays two data sets for $\sigma = 0.1$, with 100 points on the left and 100,000 points on the right. We perform experiments for $\sigma \in \{1, 0.1, 10^{-2}, \ldots, 10^{-6}\}$, with sample sizes $n$ ranging from 2 to $100,000$ according to a geometric progression. We generate 10 random data sets for each $(\sigma, n)$.

To compare the use of the enumerative code (formula 37) and that of the hierarchical one (formula 72), we search the best solution among the data grid models $M_I$ consisting of $I = J$ equal frequency intervals per variable. For each data set, we thus discretize the variables in equal frequency intervals, collect the frequencies of the resulting intervals and cells, and compute the minimum code length obtained using each code. For each code, we keep the model with the minimum code length over all the evaluated interval numbers.

The best model provides a piecewise-constant density estimator, where the density is constant per cell and the cell probabilities are given by $n_{ij}/n$. We then assess the quality of the estimation by computing the Kullback-Leibler divergence (KLD) (Cover and Thomas, 1991) between the estimated density and the true density. The details of this computation are given in Appendix C. For each parameter $\sigma$ and each sample size $n$, we compute the mean of KLD over the ten generated data sets as well as the mean of the optimal number of intervals.

For $\sigma = 1$, the true density is not much different from noise for small sample sizes. Both codes get the same performance and require at least $1,000$ points to start retrieving the true pattern. For all $\sigma \leq 0.1$, the hierarchical code gets better performance than the enumerative code. Figure 8 shows the KLD results for $\sigma = 0.1, 0.01$ and $10^{-6}$ and Figure 9 reports the optimal number of intervals for the related density estimation models. The case of $\sigma = 0.1$

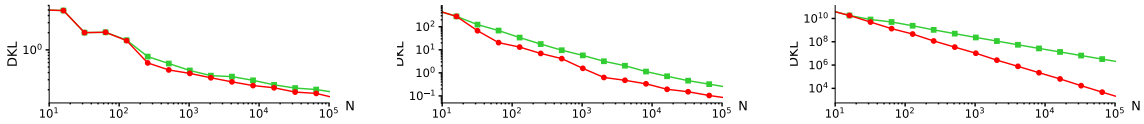



Figure 8: Divergence of Kullback-Leibler between the estimated density and the true density. ■: Enumerative code ●: Hierarchical code

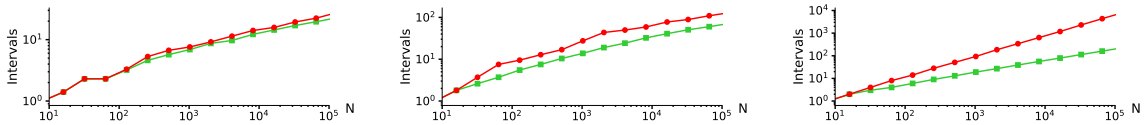



Figure 9: Optimal number of intervals. ■: Enumerative code ●: Hierarchical code

represents the transition where the hierarchical code starts to dominate the enumerative code, with a slight difference. For smaller $\sigma$, the hierarchical code gets superior performance, with a far smaller KLD obtained using a much larger number of intervals. In the extreme case of $\sigma = 10^{-6}$ where the joint density estimation reduces to an almost noiseless regression task, the estimation of the true density is far more accurate. For $n = 100,000$, it is obtained using around $8,000$ instead of 200 intervals, which involves a multinomial distribution with around 64 millions of outcomes. The hierarchical code produces far more intervals than the enumerative one, thanks to its more parsimonious prior for heavily unbalanced multinomial distributions. It is then able to approximate the true joint distribution very accurately with far smaller sample sizes. For example, for $\sigma = 10^{-6}$, the hierarchical code requires 100 times less instances to reach the best performance obtained by the enumerative code with $100,000$ instances.

This experiment demonstrates the interest of using the hierarchical multinomial code, with an important impact on the accuracy of the selected models in case of unbalanced multinomial distributions. Actually, combining the enumerative and hierarchical multinomial codes, as suggested in Section 3.6, allows to improve the selected models for any multinomial distributions. Perfectly balanced ones will be treated using the enumerative multinomial code, whereas the unbalanced distributions will benefit from the hierarchical multinomial code, with a smooth transition between the two kinds of models.

### 4.4.3 Application to the appliances energy prediction data set

We now apply the hierarchical code on the real data set introduced in Section 4.3.1. We extend the algorithm (Boullé, 2012) referenced in Section 4.1.2 by computing the code length of the best hierarchical model using the heuristic suggested in Section 3.6.2.

Figure 10 shows the impact of using the hierarchical code for density estimation, on the 465 pairs of variables of the appliances energy prediction data set. The pair with the best *Level* is still the pair between the two random variables (cf. Section 3.6.2), but its *Level* grows from $Level = 0.204$ and a $90 \times 90$ grids of cells to $Level = 0.315$ and a $356 \times 356$ grids of cells: the *Level* is improved by more than 50%, with a far better estimation of the density. Focusing on the application variables, about half of the pairs are better compressed using the hierarchical code rather than the enumerative code, with a mean improvement of 2%.

In this application to joint density estimation on a real data set, using the hierarchical multinomial code allows to improve the accuracy of the models in many cases, thus to provide more detailed insights in the exploratory analysis task.

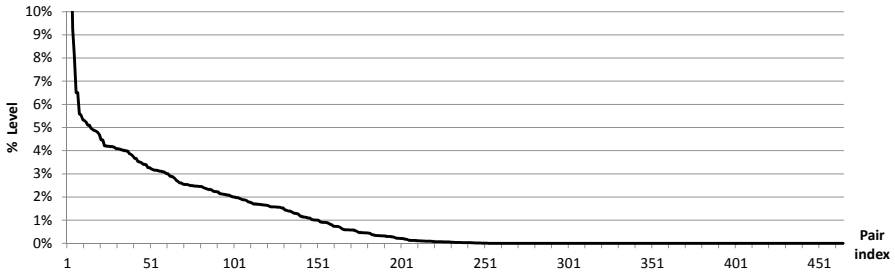



Figure 10: Improvement of the *Level* criterion when using the hierarchical code rather than the enumerative code.

# 5 Conclusion

In this paper, we have suggested a new code for multinomial distributions, based on a hierarchical encoding the model parameters. The outcomes are partitioned into two groups and the occurrences are distributed on these two group using a binomial distribution. Then, within each group of outcomes, the related occurrences are distributed according to a multinomial distribution and encoded using the enumerative code presented in Section 2.2.

Both codes differ only in the way they encode the multinomial parameters: they results in the same stochastic complexity, i.e. the code length for encoding the data given the model. The hierarchical and enumerative multinomial codes are compared both asymptotically and empirically in the non-asymptotic case. The comparison shows that there are vast regions in the parameter space where each code dominates the other one. The enumerative code compresses the model parameters better for balanced multinomial distributions, whereas the hierarchical code is better suitable for unbalanced multinomial distributions.

The hierarchical multinomial code is no longer NML, and thus it looses all the appealing theoretical properties of NML codes. However, one can build a new code that combines the enumerative and hierarchical codes, with a resulting regret that is always within a margin of $\log 2$ above the enumerative code. This combined code comes with a far better regret anywhere the hierarchical code dominates the enumerative one. This code can be seen as an illustration of the principles of luckiness NML (De Rooij and Grünwald, 2009).

The hierarchical code requires an optimization algorithm which time complexity is linear w.r.t. to the number of outcomes of the multinomial model, whereas the enumerative code exploits a simple closed formula. Therefore, potentially better coding lengths come at the expense of longer computation times, and this trade-off has to be studied per application.

The new code can potentially benefit to applications of model selection that uses multinomial distributions as building blocks, in case of unbalanced distributions. Preliminary experiments are performed with a joint density estimation method. They confirm the important impact of using the new code, resulting in far more accurate selected models. Altogether, the theoretical and experimental results suggest that one might use the hierarchical multinomial code in practice. Still, further work is necessary to incorporate this new code in existing methods that exploit multinomial distributions, especially for adapting the optimization algorithms to efficiently explore and retrieve the best parameter hierarchy for each multinomial block.

# A Proof of theorems

**Theorem (1).** *The difference of enumerative parameter complexity between the hierarchical multinomial model and its alternative version is*

$$\delta eCOMP^{(n)}(\theta^H, \theta^{H'}) \quad = \quad m(\log 2 - H(\beta)) - \frac{1}{2}\log m + \frac{1}{2}\log 2\pi \mathrm{var}(\beta) + O(1/m) \quad (76)$$

*Proof.* From Section 3.1, the difference of enumerative parameter complexity between the hierarchical multinomial model and its alternative version reduce to the difference of coding of the partition of the $m$ outcomes into two groups of size $m^A$ and $m^B$.

$$\delta eCOMP^{(n)}(\theta^H, \theta^{H'}) = (m-1)\log 2 - \left(\log m/2 + \log \binom{m}{m^A}\right). \quad (77)$$

We exploit the following approximation given in (Grünwald, 2007) (formula 4.36) with the Bernoulli parameter $\theta$

$$\log \binom{n}{\theta n} = nH(\theta) - \log \sqrt{2\pi n \mathrm{var}(\theta)} + O(1/n), \quad (78)$$

where $H(\theta) = -\theta \log \theta - (1-\theta)\log(1-\theta)$ and $\mathrm{var}(\theta) = \theta(1-\theta)$.

Let $\beta$ be the proportion of outcomes in group $A$: $m^A = \beta m$. We get

$$\begin{aligned}
\delta eCOMP^{(n)}(\theta^H, \theta^{H'}) &= (m-1)\log 2 && (79)\\
&\quad - \log m/2 - mH(\beta) + \log \sqrt{2\pi m \mathrm{var}(\beta)} + O(1/m), && (80)\\
&= m(\log 2 - H(\beta)) - \frac{1}{2}\log m + \frac{1}{2}\log 2\pi \mathrm{var}(\beta) + O(1/m) && (81)
\end{aligned}$$

□

**Theorem (2).** *The difference of enumerative parameter complexity between the hierarchical multinomial code and the enumerative one is*

$$\begin{aligned}
\delta eCOMP^{(n)}(\theta^H, \theta) &= m\left(\log 2 + H(\beta) - H(\alpha) + (\beta-\alpha)\log\frac{\alpha}{1-\alpha}\right) + \frac{1}{2}\log m && (82)\\
&\quad -\frac{1}{2}(2\log \mathrm{var}(\alpha) - \log \mathrm{var}(\beta) + \log 8\pi) && (83)\\
&\quad +\frac{m^2}{n+m}\frac{(\beta-\alpha)^2}{2\alpha(1-\alpha)} + O(\frac{m^3}{n^2}) && (84)
\end{aligned}$$

*Proof.* Starting from formula 16 given in section 3.2 and using

$$\log \binom{n+m-1}{m-1} = \log \binom{n+m}{m} - \log(n+m) + \log m, \quad (85)$$

we get

$$\begin{aligned}
\delta eCOMP^{(n)}(\theta^H, \theta) &= (m-1)\log 2 + \log(n+1) && (86)\\
&\quad + \log \binom{n^A + m^A}{m^A} + \log \binom{n^B + m^B}{m^B} && (87)\\
&\quad - \log \binom{n+m}{m} && (88)\\
&\quad - \log(n^A + m^A) + \log m^A - \log(n^B + m^B) + \log m^B && (89)\\
&\quad + \log(n+m) - \log m && (90)
\end{aligned}$$

We can rearrange the binomial terms, leading to:

$$\begin{aligned}
\delta eCOMP^{(n)}(\theta^H, \theta) &= (m-1)\log 2 + \log(n+1) && (91)\\
&\quad + \log \binom{n}{n^A} + \log \binom{m}{m^A} - \log \binom{n+m}{n^A + m^A} && (92)\\
&\quad - \log(n^A + m^A) + \log m^A - \log(n^B + m^B) + \log m^B && (93)\\
&\quad + \log(n+m) - \log m && (94)
\end{aligned}$$

Let us introduce the proportion $\alpha$ of occurrences in group $A$ ($n^A = \alpha n$) and $\beta$ the proportion of outcomes in group $A$ ($m^A = \beta m$). Let $\gamma$ be such that $(n^A + m^A) = \gamma(n+m)$.

$$\begin{aligned}
\delta eCOMP^{(n)}(\theta^H, \theta) &= (m-1)\log 2 + \log(n+1) && (95)\\
&\quad + \log\binom{n}{\alpha n} + \log\binom{m}{\beta m} - \log\binom{n+m}{\gamma(n+m)} && (96)\\
&\quad + \log(n+m) - \log(\gamma(n+m)) - \log((1-\gamma)(n+m)) && (97)\\
&\quad + \log(\beta m) + \log((1-\beta)m) - \log m && (98)
\end{aligned}$$

We exploit the following approximation given in (Grünwald, 2007) (formula 4.36) with the Bernoulli parameter $\theta$

$$\log\binom{n}{\theta n} = nH(\theta) - \log\sqrt{2\pi n \mathrm{var}(\theta)} + O(1/n), \quad (99)$$

where $H(\theta) = -\theta\log\theta - (1-\theta)\log(1-\theta)$ and $\mathrm{var}(\theta) = \theta(1-\theta)$.

We obtain

$$\begin{aligned}
\delta eCOMP^{(n)}(\theta^H, \theta) &= (m-1)\log 2 + \log n(1+1/n) && (100)\\
&\quad + nH(\alpha) - \log\sqrt{2\pi n \mathrm{var}(\alpha)} + O(1/n) && (101)\\
&\quad + mH(\beta) - \log\sqrt{2\pi m \mathrm{var}(\beta)} + O(1/m) && (102)\\
&\quad - (n+m)H(\gamma) + \log\sqrt{2\pi(n+m)\mathrm{var}(\gamma)} + O(1/(n+m)) && (103)\\
&\quad + \log m - \log(n+m) - \log\mathrm{var}(\gamma) + \log\mathrm{var}(\beta) && (104)
\end{aligned}$$

$$\begin{aligned}
\delta eCOMP^{(n)}(\theta^H, \theta) &= n(H(\alpha) - H(\gamma)) + m(\log 2 + H(\beta) - H(\gamma)) && (105)\\
&\quad - \log\sqrt{2\pi n \mathrm{var}(\alpha)} && (106)\\
&\quad - \log\sqrt{2\pi m \mathrm{var}(\beta)} && (107)\\
&\quad + \log\sqrt{2\pi(n+m)\mathrm{var}(\gamma)} && (108)\\
&\quad - \log((n+m)/nm) - \log\mathrm{var}(\gamma) + \log\mathrm{var}(\beta) - \log 2 && (109)\\
&\quad + O(1/(n+m)) + O(1/n) + O(1/m) && (110)
\end{aligned}$$

$$\begin{aligned}
\delta eCOMP^{(n)}(\theta^H, \theta) &= n(H(\alpha) - H(\gamma)) + m(\log 2 + H(\beta) - H(\gamma)) && (111)\\
&\quad - \frac{1}{2}\log((n+m)/nm) && (112)\\
&\quad - \frac{1}{2}(\log\mathrm{var}(\gamma) + \log\mathrm{var}(\alpha) - \log\mathrm{var}(\beta) + \log 8\pi) && (113)\\
&\quad + O(1/n) + O(1/m) && (114)
\end{aligned}$$

For a given $m$, let us study the limit of $\delta eCOMP^{(n)}(\theta^H, \theta)$ when $n$ goes to infinity. Let us focus on the three first lines of the preceding formula, with a focus on $n \gg m$.

$$\begin{aligned}
\Delta_1^{(n)} &= n(H(\alpha) - H(\gamma)) + m(\log 2 + H(\beta) - H(\gamma)), && (115)\\
\Delta_2^{(n)} &= -\frac{1}{2}\log((n+m)/nm), && (116)\\
\Delta_3^{(n)} &= -\frac{1}{2}(\log\mathrm{var}(\gamma) + \log\mathrm{var}(\alpha) - \log\mathrm{var}(\beta) + \log 8\pi). && (117)
\end{aligned}$$

The second term is:

$$\Delta_2^{(n)} = \frac{1}{2}\log m - \frac{1}{2}\log(1 + m/n). \quad (118)$$

As for $\gamma$, we have

$$\begin{aligned}\gamma &= \frac{n^A + m^A}{n+m}, &(119)\\ &= \frac{\alpha n + \beta m}{n+m}, &(120)\\ &= \frac{n}{n+m}\alpha + \frac{m}{n+m}\beta. &(121)\end{aligned}$$

Let $\epsilon = \frac{m}{n+m}$. We get $\gamma = (1-\epsilon)\alpha + \epsilon\beta$.
We obtain

$$\begin{aligned}\Delta_1^{(n)} &= (n+m)\left((1-\epsilon)H(\alpha) + \epsilon H(\beta) - H((1-\epsilon)\alpha + \epsilon\beta)\right) &(122)\\ &\quad + m\log 2. &(123)\end{aligned}$$

As the entropy function $H(x)$ is concave, the first line of the preceding formula is negative. To approximate $\Delta^{(n)}$, we use the Taylor series of $H(x)$.

We have $H(x+\epsilon) = H(x) + \epsilon H^{'}(x) + \frac{\epsilon^2}{2}H^{''}(x) + O(\epsilon^3)$, with $H^{'}(x) = -\log\frac{x}{1-x}$ and $H^{''}(x) = \frac{-1}{x(1-x)}$.
We get

$$\begin{aligned}H((1-\epsilon)\alpha + \epsilon\beta) &= H(\alpha + \epsilon(\beta-\alpha)), &(124)\\ &= H(\alpha) - \epsilon(\beta-\alpha)\log\frac{\alpha}{1-\alpha} - \frac{\epsilon^2(\beta-\alpha)^2}{2\alpha(1-\alpha)} + O(\epsilon^3). &(125)\end{aligned}$$

Back to $\Delta_1^{(n)}$, we obtain

$$\begin{aligned}\Delta_1^{(n)} &= (n+m)\epsilon\left(H(\beta) - H(\alpha) + (\beta-\alpha)\log\frac{\alpha}{1-\alpha} + \frac{\epsilon(\beta-\alpha)^2}{2\alpha(1-\alpha)} + O(\epsilon^2)\right) &(126)\\ &\quad + m\log 2, &(127)\\ &= m\left(\log 2 + H(\beta) - H(\alpha) + (\beta-\alpha)\log\frac{\alpha}{1-\alpha}\right) &(128)\\ &\quad + \frac{m^2}{n+m}\frac{(\beta-\alpha)^2}{2\alpha(1-\alpha)} + O\left(\frac{m^3}{(n+m)^2}\right). &(129)\end{aligned}$$

For the last term $\Delta_3^{(n)}$, we introduce $g(x) = \log\mathrm{var}(x) = \log x + \log(1-x)$.
We have $g(x+\epsilon) = g(x) + \epsilon g^{'}(x) + O(\epsilon^2)$, with $g^{'}(x) = (1-2x)/(x(1-x))$.
Then,

$$\begin{aligned}\log\mathrm{var}(\gamma) &= g(\alpha + \epsilon(\beta-\alpha)), &(130)\\ &= g(\alpha) - \epsilon(\beta-\alpha)\frac{1-2\alpha}{\alpha(1-\alpha)} + O((\epsilon(\beta-\alpha))^2). &(131)\end{aligned}$$

And

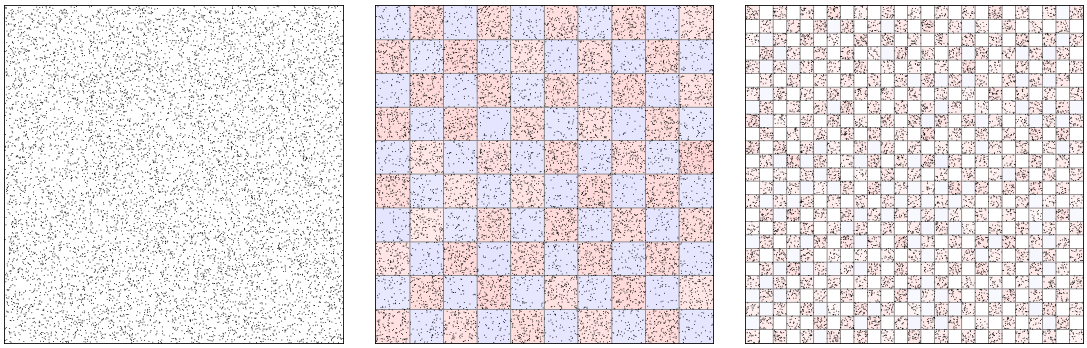
$$\begin{aligned}\Delta_3^{(n)} &= -\frac{1}{2}(\log\mathrm{var}(\gamma) + \log\mathrm{var}(\alpha) - \log\mathrm{var}(\beta) + \log 8\pi), &(132)\\ &= -\frac{1}{2}(2\log\mathrm{var}(\alpha) - \log\mathrm{var}(\beta) + \log 8\pi) + O(\frac{m}{n+m}). &(133)\end{aligned}$$

We finally use these intermediate approximations to get the full approximation of $\delta eCOMP^{(n)}(\theta^H, \theta)$:

$$
\begin{aligned}
\delta eCOMP^{(n)}(\theta^H, \theta) &= m\left(\log 2 + H(\beta) - H(\alpha) + (\beta-\alpha)\log\frac{\alpha}{1-\alpha}\right) && (134)\\
&+\frac{m^2}{n+m}\frac{(\beta-\alpha)^2}{2\alpha(1-\alpha)} + O\left(\frac{m^3}{(n+m)^2}\right) && (135)\\
&+\frac{1}{2}\log m - \frac{1}{2}\log(1+m/n) && (136)\\
&-\frac{1}{2}(2\log \mathrm{var}(\alpha) - \log \mathrm{var}(\beta) + \log 8\pi) + O(\frac{m}{n+m}) && (137)\\
&+O(1/n) + O(1/m) && (138)\\
&= m\left(\log 2 + H(\beta) - H(\alpha) + (\beta-\alpha)\log\frac{\alpha}{1-\alpha}\right) + \frac{1}{2}\log m && (139)\\
&-\frac{1}{2}(2\log \mathrm{var}(\alpha) - \log \mathrm{var}(\beta) + \log 8\pi) && (140)\\
&+\frac{m^2}{n+m}\frac{(\beta-\alpha)^2}{2\alpha(1-\alpha)} && (141)\\
&+O(\frac{m^3}{n^2}) && (142)
\end{aligned}
$$

□

# B Density estimation for the challenging data sets



$D_1$ $D_2$ $D_3$

Figure 11: Three artificial data sets and the retrieved density estimation models, with 10,000 points.

The experiment with the $D1, D2, D3$ challenging data sets introduced in Section 4.3.2 is performed with $10,000$ points instead of 1000 points. Figure 11 shows that the checkerboard patterns are very accurately retrieved with sufficient points.

# C Computation of the Kullback-Leibler divergence

Let $X$ be a random variable uniformly distributed on $[0;1]$ and $Y$ be equal to $X$ with a Gaussian noise.

$$
\begin{aligned}
X &\sim \mathcal{U}([0;1]), && (143)\\
Y &= X + \epsilon, && (144)\\
\epsilon &\sim \mathcal{N}(0,\sigma). && (145)
\end{aligned}
$$

The true density $td(x,y)$ of $(X,Y)$ is

$$td(x,y) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{y-x}{\sigma}\right)^2}. \tag{146}$$

The estimated density is inferred using the method described in Section 4. It is based on a discretization of $X$ into $I$ intervals with lower and upper bounds $lb_i$ and $ub_i$, a discretization of $Y$ into $J$ intervals with lower and upper bounds $lb_j$ and $ub_j$. Each cell $(i,j)$ has a frequency $n_{ij}$, with a probability $p_{ij} = \frac{n_{ij}}{n}$ and a density $d_{ij} = \frac{p_{ij}}{(ub_i - lb_i)(ub_j - lb_j)}$.

Let $KLD(ed,td)$ be the divergence of Kullback-Leibler between the estimated density and the true density.

We have

$$\begin{aligned}
KLD(ed,td) &= \sum_{i=1}^{I}\sum_{j=1}^{J}\int_{lb_i}^{ub_i}\int_{lb_j}^{ub_j} d_{ij}\log\frac{d_{ij}}{td(x,y)}\mathrm{d}y\mathrm{d}x, && (147)\\
&= \sum_{i=1}^{I}\sum_{j=1}^{J}\int_{lb_i}^{ub_i}\int_{lb_j}^{ub_j} d_{ij}\left(\log d_{ij} - \log\frac{1}{\sigma\sqrt{2\pi}} + \frac{1}{2}\left(\frac{y-x}{\sigma}\right)^2\right)\mathrm{d}y\mathrm{d}x, && (148)\\
&= \sum_{i=1}^{I}\sum_{j=1}^{J}\int_{lb_i}^{ub_i}\int_{lb_j}^{ub_j} d_{ij}\left(\log d_{ij} - \log\frac{1}{\sigma\sqrt{2\pi}}\right)\mathrm{d}y\mathrm{d}x + && (149)\\
&\quad \frac{1}{2\sigma^2}\sum_{i=1}^{I}\sum_{j=1}^{J}\int_{lb_i}^{ub_i}\int_{lb_j}^{ub_j} d_{ij}(y-x)^2\mathrm{d}y\mathrm{d}x, && (150)\\
&= \sum_{i=1}^{I}\sum_{j=1}^{J} p_{ij}\log\left(d_{ij}\sigma\sqrt{2\pi}\right) && (151)\\
&\quad + \frac{1}{6\sigma^2}\sum_{i=1}^{I}\sum_{j=1}^{J}\int_{lb_i}^{ub_i} d_{ij}\left((ub_j - x)^3 - (lb_j - x)^3\right)\mathrm{d}x, && (152)\\
&= \sum_{i=1}^{I}\sum_{j=1}^{J} p_{ij}\log\left(d_{ij}\sigma\sqrt{2\pi}\right) && (153)\\
&\quad + \frac{1}{24\sigma^2}\sum_{i=1}^{I}\sum_{j=1}^{J} d_{ij}\left((lb_j - ub_i)^4 + (ub_j - lb_i)^4 - (ub_j - ub_i)^4 - (lb_j - lb_i)^4\right) && (154)
\end{aligned}$$

# References


Adriaans, P. and Vitányi, P. (2007). The power and perils of MDL. In *IEEE International Symposium on Information Theory*, pages 2216–2220.

Blake, C. and Merz, C. (1996). UCI repository of machine learning databases. http://www.ics.uci.edu/mlearn/MLRepository.html.

Boullé, M. (2011). Data grid models for preparation and modeling in supervised learning. In Guyon, I., Cawley, G., Dror, G., and Saffari, A., editors, *Hands-On Pattern Recognition: Challenges in Machine Learning, volume 1*, pages 99–130. Microtome Publishing.

Boullé, M. (2012). Functional data clustering via piecewise constant nonparametric density estimation. *Pattern Recognition*, 45(12):4389–4401.

Boullé, M. (2014). Towards automatic feature construction for supervised classification. In *ECML/PKDD 2014*, pages 181–196. Springer-Verlag.

Boullé, M., Clérot, F., and Hue, C. (2016). Revisiting enumerative two-part crude MDL for Bernoulli and multinomial distributions. In Rissanen, J., Leppä-aho, J., Roos, T., and Myllymäki, P., editors, *Proceedings of the Ninth Workshop on Information Theoretic Methods in Science and Engineering*, pages 12–15.

Candanedo, L., Feldheim, V., and Deramaix, D. (2017). Data driven prediction models of energy use of appliances in a low-energy house. *Energy and Buildings*, 140:81–97.

Cover, T. and Thomas, J. (1991). *Elements of information theory*. Wiley-Interscience, New York, NY, USA.

De Rooij, S. and Grünwald, P. (2009). Luckiness and regret in minimum description length inference.

Grünwald, P. (2007). *The minimum description length principle*. Adaptive computation and machine learning. MIT Press.

Guigourès, R., Gay, D., Boullé, M., Clérot, F., and Rossi, F. (2015). Country-scale exploratory analysis of call detail records through the lens of data grid models. In *ECML/PKDD*, pages 37–52.

Guo, Z., Gao, X., Ren, H., Yang, Y., Di, R., and Chen, D. (2017). Learning bayesian network parameters from small data sets: A further constrained qualitatively maximum a posteriori method. *International Journal of Approximate Reasoning*, 91:22–35.

Hansen, M. and Yu, B. (2001). Model selection and the principle of minimum description length. *J. American Statistical Association*, 96:746–774.

Kontkanen, P. (2009). *Computationally efficient methods for MDL-optimal density estimation and data clustering*. Department of Computer Science, series of publications A, report, 2009-11. University of Helsinki.

Kontkanen, P. and Myllymäki, P. (2007). A linear-time algorithm for computing the multinomial stochastic complexity. *Information Processing Letters*, 103(6):227–233.

Lugosi, G. and Nobel, A. (1996). Consistency of data-driven histogram methods for density estimation and classification. *The Annals of Statistics*, 24(2):687–706.

Madeira, S. and Oliveira, A. (2004). Biclustering algorithms for biological data analysis: A survey. *IEEE/ACM Transactions Computational Biolology and Bioinformatics*, 1(1):24–45.

Mei, Q., Shen, X., and Zhai, C. (2007). Automatic labeling of multinomial topic models. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '07, pages 490–499. ACM.

Mononen, T. and Myllymäki, P. (2007). Fast NML computation for naive bayes models. In *10th International Conference on Discovery Science*, pages 151–160.

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

Newman, M. (2005). Power laws, pareto distributions and zipf's law. *Contemporary Physics*, 46(5):323–351.

Powers, D. (1998). Applications and explanations of zipf's law. In *Proceedings of the Joint Conferences on New Methods in Language Processing and Computational Natural Language Learning*, NeMLaP3/CoNLL '98, pages 151–160. Association for Computational Linguistics.

Rissanen, J. (1978). Modeling by shortest data description. *Automatica*, 14:465–471.

Rissanen, J. (1983). A universal prior for integers and estimation by minimum description length. *Annals of Statistics*, 11(2):416–431.

Rissanen, J. (1996). Fisher information and stochastic complexity. *IEEE Transactions on Information Theory*, 42(1):40–47.

Rissanen, J. (2000). Strong optimality of the normalized ml models as universal codes. *IEEE Transactions on Information Theory*, 47:1712–1717.

Scott, D. (2015). *Multivariate density estimation: theory, practice, and visualization*. John Wiley & Sons, Inc, second edition edition.

Seldin, Y. and Tishby, N. (2010). Pac-bayesian analysis of co-clustering and beyond. *Journal of Machine Learning Research*, 11:3595–3646.

Shtarkov, Y. M. (1987). Universal sequential coding of single messages. *Problems of Information Transmission*, 23(3):3–17.

Silander, T., Roos, T., and Myllymäki, P. (2010). Learning locally minimax optimal bayesian networks. *International Journal of Approximate Reasoning*, 51(5):544–557.

Sun, J., Faloutsos, C., Papadimitriou, S., and Yu, P. (2007). Graphscope: Parameter-free mining of large time-evolving graphs. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '07, pages 687–696. ACM.

Szpankowski, W. (1998). On asymptotics of certain recurrences arising in universal coding. *Problems of Information Transmission*, 34(2):142–146.

Vitányi, P. and Li, M. (2000). Minimum description length induction, Bayesianism, and Kolmogorov complexity. *IEEE Transactions on information theory*, 46:446–464.

Voisine, N., Boullé, M., and Hue, C. (2009). A bayes evaluation criterion for decision trees. *Advances in Knowledge Discovery and Management (AKDM09)*, 292:21–38.