Fast Non-Parametric Bayesian Inference on Infinite Trees

Marcus Hutter

IDSIA, Galleria 2, CH-6928 Manno-Lugano, Switzerland
marcus@idsia.ch, http://www.idsia.ch/~marcus

21 May 2004

Abstract

Given i.i.d. data from an unknown distribution, we consider the problem of predicting future items. An adaptive way to estimate the probability density is to recursively subdivide the domain to an appropriate data-dependent granularity. A Bayesian would assign a data-independent prior probability to "subdivide", which leads to a prior over infinite(ly many) trees. We derive an exact, fast, and simple inference algorithm for such a prior, for the data evidence, the predictive distribution, the effective model dimension, and other quantities.

1 INTRODUCTION

Inference. We consider the problem of inference from i.i.d. data D, in particular of the unknown distribution q the data is sampled from. In case of a continuous domain this means inferring a probability density from data. Without structural assumption on q, this is hard to impossible, since a finite amount of data is never sufficient to uniquely select a density (model) from an infinite-dimensional space of densities (model class).

Methods. In parametric estimation one assumes that q belongs to a finite-dimensional family. The two-dimensional family of Gaussians characterized by mean and variance is prototypical. The maximum likelihood (ML) estimate of q is the distribution that maximizes the data likelihood. Maximum likelihood overfits if the family is too large and especially if it is infinite-dimensional. A remedy is to penalize complex distributions by assigning a prior (2nd order) probability to the densities q. Maximizing the model posterior (MAP), which is proportional to likelihood times the prior, prevents overfitting. Bayesians keep the complete posterior for inference. Typically, summaries like the mean and variance of the posterior are reported.

How to choose the prior? In finite or small compact low-dimensional spaces a uniform prior often works (MAP reduces to ML). In the non-parametric case one typically devises a hierarchy of finite-dimensional model classes of increasing dimension. Selecting the dimension with maximal posterior often works well due to the Bayes factor phenomenon [Goo83, Jay03, Mac03]: In case the true model is low-dimensional, higher-dimensional (complex) model classes are automatically penalized, since they contain fewer "good" models. Full Bayesians would assign a prior probability (e.g. $\frac{1}{d^2}$) to dimension d and mix over dimension.

Interval Bins. The probably simplest and oldest model for an interval domain is to divide the interval (uniformly) into bins, assume a constant distribution within each bin, and take a frequency estimate for the probability in each bin, or a Dirichlet posterior if you are a Bayesian. There are heuristics for choosing the number of bins as a function of the data size. The simplicity and easy computability of the bin model is very appealing to practitioners. Drawbacks are that distributions are discontinuous, its restriction to one dimension (or at most low dimension: curse of dimensionality), the uniform (or more generally fixed) discretization, and the heuristic choice of the number of bins. We present a full Bayesian solution to these problems, except for the non-continuity problem. Polya trees [Lav94] inspired our model.

More advanced model classes. There are plenty of alternative Bayesian models that overcome some or all of the limitations. Examples are continuous Dirichlet process (mixtures) [Fer73], Bernstein polynomials [PW02], Bayesian field theory [Lem03], Bayesian kernel density estimation or other mixture models [EW95], or universal priors [Hut04b], but analytical solutions are infeasible. Markov Chain Monte Carlo sampling or Expectation Maximization algorithms [DLR77] or variational methods can often be used to obtain approximate numerical solutions, but computation time and global convergence remain critical issues. Practitioners usually use (with success) efficient MAP or M(D)L or heuristic methods, e.g. kernel density estimation [GM03], but note that MAP or MDL *can* fail, while Bayes works [PH04].

Our tree mixture model. The idea of the model class discussed in this paper is very simple: With equal probability, we chose q either uniform or split the domain in two parts (of equal volume), and assign a prior to each part, recursively, i.e. in each part again either uniform or split. For finitely many splits, q is a piecewise constant function, for infinitely many splits it is virtually *any* distribution. While the prior over q is neutral about uniform versus split, we will see that the posterior favors a split if and only if the data clearly indicates non-uniformity. The method is a full Bayesian non-heuristic tree approach to adaptive binning for which we present a very simple and fast algorithm for computing all(?) quantities of interest.

Contents. In Section 2 we introduce our model and compare it to Polya trees. We also discuss some example domains, like intervals, strings, volumes, and classification tasks. In Section 3 we present recursions for various quantities of interest, including the data evidence, the predictive distribution, the effective model dimension, the tree size and height, and cell volume. We discuss the qualitative behavior and state convergence of the posterior for finite trees. The proper case of infinite trees is discussed in Section 4, where we analytically solve the infinite recursion at the data separation level. Section 5 collects everything together and presents the algorithm. We also numerically illustrate the behavior of our model on one example distribution. Section 6 contains a brief summary, conclusions, and outlook, including natural generalizations of our model. See [Hut04a] for derivations, proofs, program code, extensions, and more details.

2 THE TREE MIXTURE MODEL

Setup and basic quantities of interest. We are given i.i.d. data $D=(x^1,...,x^n)\in\Gamma^n$ of size n from domain Γ, e.g. $\Gamma\subseteq I\!\!R^d$ sampled from some unknown probability density $q:\Gamma\to I\!\!R$. Standard inference problems are to estimate q from D or to predict the next data item $x^{n+1}\in\Gamma$. By definition, the (objective or aleatoric) data likelihood density under model q is $p(D|q)\equiv q(x_1)\cdot...\cdot q(x_n)$. Note that we consider sorted data, which avoids annoying multinomial coefficients. Otherwise this has no consequences. Results are independent of the order and depend on the counts only, as they should. A Bayesian assumes a (belief or 2^{nd}-order or epistemic or subjective) prior $p(q)$ over models q in some model class Q. The data evidence is $p(D)=\int_Q p(D|q)p(q)dq$. Having the evidence, Bayes' famous rule allows to compute the (belief or 2^{nd}-order or epistemic or subjective) posterior $p(q|D)=p(D|q)p(q)/p(D)$ of q. The predictive or posterior distribution of x is $p(x|D)=p(D,x)/p(D)$, i.e. the conditional probability that the next data item is $x=x^{n+1}$, given D, follows from the evidences of D and (D,x). Since the posterior of q is a complex object, we need summaries like the expected q-probability of x and (co)variances. Fortunately they can also be reduced to computation of evidences: $E[q(x)|D]:=\int q(x)p(q|D)dq=p(x|D)$. In the last equality we used the formulas for the posterior, the likelihood, the evidence, and the predictive distribution, in this order. Similarly for the covariance. We derive and discuss further summaries of q for our particular tree model, like the model complexity or effective dimension, and the tree height or cell size, later.

Hierarchical tree partitioning. Up to now everything has been fairly general. We now introduce the tree representation of domain Γ. We partition Γ into Γ_0 and Γ_1, i.e. $\Gamma=\Gamma_0\cup\Gamma_1$ and $\Gamma_0\cap\Gamma_1=\phi$. Recursively we (sub)partition $\Gamma_z=\Gamma_{z0}\dot\cup\Gamma_{z1}$ for $z\in I\!\!B_0^m$, where $I\!\!B_k^m=\bigcup_{i=k}^m\{0,1\}^i$ is the set of all binary strings of length between k and m, and $\Gamma_\epsilon=\Gamma$, where $\epsilon=\{0,1\}^0$ is the empty string. We are interested in an infinite recursion, but for convenience we assume a finite tree height $m<\infty$ and consider $m\to\infty$ later. Also let $l:=\ell(z)$ be the length of string $z=z_1...z_l=:z_{1:l}$, and $|\Gamma_z|$ the volume or length or cardinality of Γ_z.

Example spaces. *Intervals:* Assume $\Gamma=[0,1)$ is the unit interval, recursively bisected into intervals $\Gamma_z=[0.z,0.z+2^{-l})$ of length $|\Gamma_z|=2^{-l}$, where $0.z$ is the real number in $[0,1)$ with binary expansion $z_1...z_l$.

Strings: Assume $\Gamma_z=\{zy:y\in\{0,1\}^{m-l}\}$ is the set of strings of length m starting with z. Then $\Gamma=\{0,1\}^m$ and $|\Gamma_z|=2^{m-l}$. For $m=\infty$ this set is continuous, for $m<\infty$ finite.

Trees: Let Γ be a complete binary tree of height m and Γ_{z0} (Γ_{z1}) be the left (right) subtree of Γ_z. If $|\Gamma_z|$ is defined as one more than the number of nodes in Γ_z, then $|\Gamma_z|=2^{m+1-l}$.

Volumes: Consider $\Gamma\subset I\!\!R^d$, e.g. the hypercube $\Gamma=[0,1)^d$. We recursively halve Γ_z with a hyperplane orthogonal to dimension $(l \bmod d)+1$, i.e. we sweep through all orthogonal directions. $|\Gamma_z|=2^{-l}|\Gamma|$.

Compactification: We can compactify $\Gamma\subseteq(1,\infty]$ (this includes $\Gamma=I\!\!N\setminus\{1\}$) to the unit interval $\Gamma':=\{\frac{1}{x}:x\in\Gamma\}\subseteq[0,1)$, and similarly $\Gamma\subseteq I\!\!R$ (this includes $\Gamma=Z\!\!\!Z$) to $\Gamma':=\{x\in[0,1):\frac{2x-1}{x(1-x)}\in\Gamma\}$. All reasonable spaces can be reduced to one of the spaces described above.

Classification: Consider an observation $o\in\Gamma'$ (e.g. email) that is classified as $c\in\{0,1\}$ (e.g. good versus spam), where Γ' could be one of the spaces above (e.g. o is a sequence of binary features in decreasing order of importance). Then $x:=(o,c)\in\Gamma:=\Gamma'\times\{0,1\}$ and $\Gamma_{0z}=\Gamma'_z\times\{0\}$ and $\Gamma_{1z}=\Gamma'_z\times\{1\}$. Given D (e.g. pre-classified emails), a new observation o is classified as c with probability $p(c|D,o)\propto p(D,x)$. Similar for more than two classes.

In all these examples (we have chosen) $|\Gamma_{z0}|=|\Gamma_{z1}|=\frac{1}{2}|\Gamma_z|\ \forall z\in I\!B_0^{m-1}$, and this is the only property we need and henceforth assume. W.l.g. we assume/define/rescale $|\Gamma|=1$. Generalizations to non-binary and non-symmetric partitions are straightforward and briefly discussed at the end.

Identification. We assume that $\{\Gamma_z:z\in I\!B_0^m\}$ are (basis) events that generate our σ-algebra. For every $x\in\Gamma$ let x' be the string of length $\ell(x')=m$ such that $x\in\Gamma_{x'}$. We assume that distributions q are σ-measurable, i.e. to be constant on $\Gamma_{x'}\ \forall x'\in I\!B^m$. For $m=\infty$ this assumption is vacuous; we get *all* Borel measures. Hence, we can identify the continuous sample space Γ with the (for $m<\infty$ discrete) space $I\!B^m$ of binary sequences of length m, i.e. in a sense all example spaces are isomorphic. While we have the volume model in mind for real-world applications, the string model will be convenient for mathematical notation, the tree metaphor will be convenient in discussion, and the interval model will be easiest to implement and to present graphically.

Notation. As described above, Γ may also be a tree. This interpretation suggests the following scheme for defining the probability of q on the leaves x'. The probability of the left child node $z0$, given we are in the parent node z, is $P[\Gamma_{z0}|\Gamma_z,q]$, so we have

$$p(x|\Gamma_z,q)=p(x|\Gamma_{z0},q)\cdot P[\Gamma_{z0}|\Gamma_z,q] \quad \text{if} \quad x\in\Gamma_{z0}$$

and similarly for the right child. In the following we often have to consider distributions conditioned to and in the subtree Γ_z, so the following notation will turn out convenient

$$q_{z0}:=P[\Gamma_{z0}|\Gamma_z,q], \quad p_z(x|...):=2^{-l}p(x|\Gamma_z...) \qquad (1)$$

$$\Rightarrow p_z(x|q)=2q_{zx_{l+1}}p_{zx_{l+1}}(x|q)=...=\prod_{i=l+1}^m 2q_{x_{1:i}} \text{ if } x\in\Gamma_z$$

where we have used that $p(x|\Gamma_{x'},q)=|\Gamma_{x'}|^{-1}=2^m$ is uniform. Note that $q_{z0}+q_{z1}=1$. Finally, let $\vec{q}_{z*}:=(q_{zy}:y\in I\!B_1^{m-l})$ be the $(2^{m-l+1}-2)$-dimensional *vector* or *ordered set* or *tree* of all *reals* $q_{zy}\in[0,1]$ in subtree Γ_z. Note that $q_z\notin\vec{q}_{z*}$. The *(non)density* $q_z(x):=p_z(x|q)$ depends on all and only these q_{zy}. For $z\neq\epsilon$, $q_z()$ and $p_z()$ are only proportional to a density due to the factor 2^{-l}, which has been introduced to make $p_{x'}(x|...)\equiv 1$. (They are densities w.r.t. $2^l\lambda_{|\Gamma_z}$, where λ is the Lebesgue measure.) We have to keep this in mind in our derivations, but can ignore this widely in our discussion.

Polya trees. In the Polya tree model one assumes that the $q_{z0}\equiv 1-q_{z1}$ are independent and Beta$(\cdot,\cdot)$ distributed, which defines the prior over q. Polya trees form a conjugate prior class, since the posterior is also a Polya tree, with empirical counts added to the Beta parameters. If the same Beta is chosen in each node, the posterior of x is pathological for $m\to\infty$: The distribution is everywhere discontinuous with probability 1. A cure is to increase the Beta parameters with l, e.g. quadratically, but this results in "underfitting" for large sample sizes, since Beta(large,large) is too informative and strongly favors q_{z0} near $\frac{1}{2}$. It also violates scale invariance, which should hold in the absence of prior knowledge. That is, the p(oste)rior in $\Gamma_0=[0,\frac{1}{2})$ should be the same as for $\Gamma=[0,1)$ (after rescaling all $x\rightsquigarrow x/2$ in D).

The new tree mixture model. The prior $P[q]$ follows from specifying a prior over $\vec{q}_*$, since $q(x)\propto q_{x_1}\cdot...\cdot q_{x_{1:m}}$ by (1). The distribution in each subset $\Gamma_z\subseteq\Gamma$ shall be either *u*niform or non-uniform. A necessary (but not sufficient) condition for uniformity is $q_{z0}=q_{z1}=\frac{1}{2}$.

$$p^u(q_{z0},q_{z1}) \;:=\; \delta(q_{z0}-\tfrac{1}{2})\delta(q_{z1}-\tfrac{1}{2}), \qquad (2)$$

where $\delta()$ is the Dirac delta. To get uniformity on Γ_z we have to recurse the tree down in this way.

$$p^u(\vec{q}_{z*}) \;:=\; p^u(q_{z0},q_{z1})p^u(\vec{q}_{z0*})p^u(\vec{q}_{z1*}) \qquad (3)$$

with the natural recursion termination $p^u(\vec{q}_{z*})=1$ when $\ell(z)=m$, since then $\vec{q}_{z*}=\phi$. For a non-uniform distribution on Γ_z we allow any probability split $q(\Gamma_z)=q(\Gamma_{z0})+q(\Gamma_{z0})$, or equivalently $1=q_{z0}+q_{z1}$. We assume a uniform prior on the *s*plit, i.e.

$$p^s(q_{z0},q_{z1}) \;:=\; \delta(q_{z0}+q_{z1}-1) \qquad (4)$$

We now recurse down the tree

$$p^s(\vec{q}_{z*}) \;:=\; p^s(q_{z0},q_{z1})p(\vec{q}_{z0*})p(\vec{q}_{z1*}) \qquad (5)$$

again with the natural recursion termination $p(\vec{q}_{z*})=p(\phi)=1$ when $\ell(z)=m$. Finally we have to mix the uniform with the non-uniform case.

$$p(\vec{q}_{z*}) \;:=\; p(u)p^u(\vec{q}_{z*})+p(s)p^s(\vec{q}_{z*}) \qquad (6)$$

We choose a 50/50 mixture $p(u)=p(s)=\frac{1}{2}$. This completes the specification of the prior $P[q]=p(\vec{q}_*)$.

For example, if the first bit in x is a class label and the remaining are binary features in decreasing order of importance, then given class and features $z=x_{1:l}$, further features $x_{l+1:m}$ could be relevant for classification ($q_z(x)$ is non-uniform) or irrelevant ($q_z(x)$ is uniform).

Comparison to the Polya tree. Note the important difference in the recursions (3) and (5). Once we decided on a uniform distribution (2) we have to equally split probabilities down the recursion to the end, i.e. we recurse in (3) with p^u, rather than the mixture p (this actually allows to solve the recursion). On the other hand if we decided on a non-uniform split (4), the left and right partition each itself may be uniform or not, i.e. we recurse in (5) with the mixture p, rather than p^s. Inserting (4) in (5) in (6) and recursively (2) in (3) in (6) we get

$$p(\vec{q}_{z*}) = \tfrac{1}{2}\prod_{y\in\mathbb{B}_1^{m-l}}\delta(q_{zy}-\tfrac{1}{2}) + \tfrac{1}{2}\delta(q_{z0}+q_{z1}-1)p(\vec{q}_{z0*})p(\vec{q}_{z1*}) \tag{7}$$

Choosing $p(u)=0$ would lead to the Polya tree model (and its problems) with $q_{z0}\sim$ Beta(1,1). For our choice ($p(u)=\frac{1}{2}$), but with p instead of p^u on the r.h.s. of (3) we would get a quasi-Polya model (same problems) with $q_{z0}\sim\frac{1}{2}$[Beta(∞,∞)+Beta(1,1)].

For $m\to\infty$, our model is scale invariant *and* leads to continuous distributions for $n\to\infty$, unlike the Polya tree model. We also don't have to tune Beta parameters; the model tunes itself by suitably assigning high/low posterior probability to subdividing cells. While Polya trees form a natural conjugate prior class, our prior does not directly, but can be generalized to do so [Hut04a]. The computational complexity for the quantities of interest will be the same (essentially $O(n)$), i.e. as good as it could be.

Formal and effective dimension. Formally our model is $2\cdot(2^m-1)$-dimensional, but the effective dimension can by much smaller, since $\vec{q}_*$ is forced with a non-zero probability to a much smaller polytope, for instance with probability $\frac{1}{2}$ to the zero-dimensional globally uniform distribution. We will compute the effective p(oste)rior dimension.

3 QUANTITIES OF INTEREST

The evidence recursion. At the end of Section 2 we defined our tree mixture model. The next step is to compute the standard quantities of interest defined at the beginning of Section 2. The evidence $p(D)$ is key, the other quantities (posterior, predictive distribution, expected $q(x)$ and its variance) follow then immediately. Let $D_z:=\{x\in D:x\in\Gamma_z\}$ be the $n_z:=|D_z|$ data points that lie in subtree Γ_z. We compute $p_z(D_z)$ recursively for all $z\in\mathbb{B}_0^{m-1}$, which gives $p(D)=p_\epsilon(D_\epsilon)$. Inserting (1) and (7) into

$$p_z(D_z) = \int p_z(D_z|\vec{q}_{z*})p(\vec{q}_{z*})d\vec{q}_{z*} \tag{8}$$

one can derive the following recursion [Hut04a]:

$$p_z(D_z) \;=\; \tfrac{1}{2}\big[1+\frac{p_{z0}(D_{z0})p_{z1}(D_{z1})}{w(n_{z0},n_{z1})}\big] \tag{9}$$

$$w(n_{z0},n_{z1}) \;:=\; 2^{-n_z}\frac{(n_z+1)!}{n_{z0}!n_{z1}!} \;=:\; w_{n_z}(\Delta_z)$$
$$n_z \;=\; n_{z0}+n_{z1},\quad \Delta_z \;:=\; \frac{n_{z0}}{n_z}-\tfrac{1}{2}$$

The recursion terminates with $p_z(D_z)=1$ when $\ell(z)=m$. Recall (1) if you insist on a formal proof: For $\ell(z)=m$ and $x\in\Gamma_z$ we have $\Gamma_{x'}=\Gamma_z \Rightarrow p_z(x|q)=1 \Rightarrow p_z(D_z|q)=1 \Rightarrow p_z(D_z)=1$.

Interpretation of (9): With probability $\frac{1}{2}$, the evidence is uniform in Γ_z. Otherwise data D_z is split into two partitions of size n_{z0} and $n_{z1}=n_z-n_{z0}$. First, choose n_{z0} uniformly in $\{0,...,n_z\}$. Second, given n_z, choose uniformly among the $\binom{n_z}{n_{z0}}$ possibilities of selecting n_{z0} out of n_z data points for Γ_{z0} (the remaining n_{z1} are then in Γ_{z1}). Third, distribute D_{z0} according to $p_{z0}(D_{z0})$ and D_{z1} according to $p_{z1}(D_{z1})$. Then, the evidence in case of a split is the second term in (9). The factor 2^{n_z} is due to our normalization convention (1). This also verifies that the r.h.s. yields the l.h.s. if integrated over all D_z, as it should be.

Discussing the weight. The relative probability of splitting (second term on r.h.s. of (9)) to the uniform case (first term in r.h.s. of (9)) is controlled by the weight w. Large (small) weight indicates a (non) uniform distribution, provided p_{z0} and p_{z1} are $O(1)$. Balance $\Delta_z\approx 0$ ($\not\approx 0$) indicates a (non) symmetric partitioning of the data among the left and right branch of Γ_z. Asymptotically for large n_z (keeping Δ_z fixed), we have

$$w_{n_z}(\Delta_z)\sim\sqrt{\tfrac{2n_z}{\pi}}\,\mathrm{e}^{-2n_z\Delta_z^2}$$

Assume that data D is sampled from the true distribution $\dot{q}$. The probability of the left branch Γ_{z0} of Γ_z is $\dot{q}_{z0}\equiv P[\Gamma_{z0}|\Gamma_z,\dot{q}]=2^l\dot{q}_z(\Gamma_{z0})$. The relative frequencies $\frac{n_{z0}}{n_z}$ asymptotically converge to $\dot{q}_{z0}$. More precisely $\frac{n_{z0}}{n_z}=\dot{q}_{z0}\pm O(n_z^{-1/2})$ with probability 1 (w.p.1). Similarly for the right branch. Assume the probabilities are equal ($\dot{q}_{z0}=\dot{q}_{z1}=\frac{1}{2}$), possibly but not necessarily due to a uniform $\dot{q}_z()$ on Γ_z. Then $\Delta_z=O(n_z^{-1/2})$, which implies

$$w_{n_z}(\Delta_z)\sim\Theta(\sqrt{n_z})\;\underset{w.p.1}{\overset{n_z\to\infty}{\longrightarrow}}\;\infty \quad\text{if}\quad \dot{q}_{z0}=\dot{q}_{z1}=\tfrac{1}{2},$$

consistent with our anticipation. Conversely, for $\dot{q}_{z0}\neq\dot{q}_{z1}$ (which implies non-uniformity of $\dot{q}_z()$) we have $\Delta_z\to c:=\dot{q}_{z0}-\frac{1}{2}\neq 0$, which implies

$$w_{n_z}(\Delta_z)\sim\sqrt{\tfrac{2n_z}{\pi}}\,\mathrm{e}^{-2n_zc^2}\;\underset{w.p.1}{\overset{n_z\to\infty}{\longrightarrow}}\;0 \quad\text{if}\quad \dot{q}_{z0}\neq\dot{q}_{z1},$$

again, consistent with our anticipation.

Asymptotic convergence/consistency ($n\to\infty$). For fixed $m<\infty$, one can show that almost surely the posterior $p_z(\vec{q}_{z*}|D)$ concentrates around the true distribution $\vec{q}_{z*}$ for $n\to\infty$. This implies that the posterior

$p_z(x|D_z)\to\dot q_z(x)$ for all $x\in\Gamma_z$. One can also show that the evidence $p_z(D_z)\to\frac12$ or 1 for uniform $\dot q_z()$, and increases exponentially with n_z for non-uniform $\dot q_z()$ (see [Hut04a] for proofs).

Model dimension and cell number. As discussed in Section 2, the effective dimension of $\vec q_*$ is the number of components that are not forced to $\frac12$ by (2). Note that a component may be "accidentally" $\frac12$ in (4), but since this is an event of probability 0, we don't have to care about this subtlety. So the effective dimension $N_{\vec q_{z*}}=\#\{q\in\vec q_{z*}:q\neq\frac12\}$ of $\vec q_{z*}$ can be given recursively as

$$N_{\vec q_{z*}} \;=\; \begin{cases} 0 & \text{if} \quad \ell(z)=m \quad \text{or} \quad q_{z0}=\frac12 \\ 1+N_{\vec q_{z0*}}+N_{\vec q_{z1*}} & \text{else}\end{cases} \qquad (10)$$

The effective dimension is zero if $q_z=\frac12$, since this implies that the whole tree Γ_z has $q_{zy}=\frac12$ due to (7). If $q_z\neq\frac12$, we add the effective dimensions of subtrees Γ_{z0} and Γ_{z1} to the root degree of freedom $q_{z0}=q_z-q_{z1}$. Bayes' rule allows to represent the posterior probability that $N_{\vec q_{z*}}=k$ as

$$P_z[N_{\vec q_{z*}}=k|D_z]\cdot p_z(D_z)=\int\delta_{N_{\vec q_{z*}}k}\,p_z(D_z|\vec q_{z*})p(\vec q_{z*})d\vec q_{z*}$$

where $P_z[...|...]:=P[...|\Gamma_z...]$, and $\delta_{ab}=1$ for $a=b$ and 0 else. The r.h.s. coincides with (8) except for the extra factor $\delta_{N_{\vec q_{z*}}k}$. Analogous to the evidence (8), using (10) we can prove the following recursion:

$$\begin{aligned} &P_z[N_{\vec q_{z*}}=0|D_z] \;=\; 1-g_z(D_z), \qquad \text{for} \quad l<m, \\ &P_z[N_{\vec q_{z*}}=k+1|D_z] \;= \qquad\qquad (11)\\ &g_z(D_z)\cdot\sum_{i=0}^k P_{z0}[N_{\vec q_{z0*}}=i|D_{z0}]\cdot P_{z1}[N_{\vec q_{z1*}}=k-i|D_{z1}], \\ &P_z[N_{\vec q_{z*}}=k|D_z]=\delta_{k0}:=\left\{\begin{smallmatrix}1\text{ if } k=0\\ 0\text{ if } k>0\end{smallmatrix}\right\} \quad \text{for} \quad l=m. \\ &g_z(D_z):=\tfrac12\frac{p_{z0}(D_{z0})p_{z1}(D_{z1})}{p_z(D_z)w(n_{z0},n_{z1})} \overset{(9)}{=} 1-\frac{1}{2p_z(D_z)} \qquad (12)\end{aligned}$$

Read: The probability that tree Γ_z has dimension $k+1$ equals the posterior probability $g_z(D_z)$ of splitting Γ_z, times the probability that left subtree has dimension i, times the probability that right subtree has dimension $k-i$, summed over all possible i.

Let us define a cell or bin as a maximal volume on which $q()$ is constant. Then the model dimension is 1 less than the number of bins (due to the probability constraint). Hence we also have a recursion for the distribution of the number of cells.

Tree height and cell size. The effective height of tree $\vec q_{z*}$ at $x\in\Gamma_z$ is also an interesting property. If $q_{z0}=\frac12$ or $\ell(z)=m$, then the height $h_{\vec q_{z*}}(x)$ of tree $\vec q_{z*}$ at x is obviously zero. If $q_{z0}\neq\frac12$, we take the height of the subtree $\vec q_{zx_{l+1}*}$ that contains x and add 1:

$$h_{\vec q_{z*}}(x) \;=\; \begin{cases} 0 & \text{if} \quad \ell(z)=m \quad \text{or} \quad q_{z0}=\frac12 \\ 1+h_{\vec q_{zx_{l+1}*}}(x) & \text{else}\end{cases}$$

One can show that the tree height at x averaged over all trees $\vec q_{z*}$ is

$$E_z[h_{\vec q_{z*}}(x)|D_z]=g_z(D_z)\Big[1+E_{zx_{l+1}}[h_{\vec q_{zx_{l+1}*}}(x)|D_{zx_{l+1}}]\Big]$$

where $E_z[f_{\vec q_{z*}}|...]=\int P_z[f_{\vec q_{z*}}|...]p(\vec q_{z*})d\vec q_{z*}$. We may also want to compute the tree height averaged over all $x\in\Gamma_z$. For $\ell(z)<m$ and $q_{z0}\neq\frac12$ we get

$$\bar h_{\vec q_{z*}} \;:=\; \int h_{\vec q_{z*}}(x)q(x|\Gamma_z)dx \;=\; 1+q_{z0}\cdot\bar h_{\vec q_{z0*}}+q_{z1}\cdot\bar h_{\vec q_{z1*}}$$

$$\begin{aligned} E_z[\bar h_{\vec q_{z*}}|D_z] \;=\; g_z(D_z)\Big[1+\frac{n_{z0}+1}{n_z+2}E_{z0}[\bar h_{\vec q_{z0*}}|D_{z0}] \\ +\frac{n_{z1}+1}{n_z+2}E_{z1}[\bar h_{\vec q_{z1*}}|D_{z1}]\Big]\end{aligned}$$

with obvious interpretation: The expected height of a subtree is weighted by its relative importance, that is (an estimate of) its probability. The recursion terminates with $E_z[h_{\vec q_{z*}}|D_z]=0$ when $\ell(z)=m$. We can also compute intra and inter tree height variances.

Finally consider the average cell size or volume v. Maybe more useful is to consider the logarithm $-\log_2|\Gamma_z|=\ell(z)$, since otherwise small volumes can get swamped in the expectation by a single large one. Log-volume $v_{\vec q_{z*}}=\ell(z)$ if $\ell(z)=m$ or $q_z=\frac12$, and else recursively $v_{\vec q_{z*}}=q_{z0}v_{\vec q_{z0*}}+q_{z1}v_{\vec q_{z1*}}$. We can reduce this to the tree height, since $v_{\vec q_{z*}}=\bar h_{\vec q_{z*}}+\ell(z)$, in particular $v_{\vec q_*}=\bar h_{\vec q_*}$

4 INFINITE TREES ($m\to\infty$)

Motivation. We have chosen an (arbitrary) finite tree height m in our setup, needed to have a well-defined recursion start at the leaves of the trees. What we are really interested in are infinite trees ($m=\infty$). Why not feel lucky with finite m? First, for continuous domain Γ (e.g. interval [0,1)), our tree model contains only piecewise constant models. The true distribution $\dot q()$ is typically non-constant and continuous (Beta, normal, ...). Such distributions are outside a finite tree model class (but inside the infinite model), and the posterior $p(x|D)$ cannot converge to the true distribution, since it is also piecewise constant. Hence all other estimators based on the posterior are also not consistent. Second, a finite m violates scale invariance (a non-informative prior on Γ_z should be the same for all z, apart from scaling). Finally, having to choose the "right" m may be worrisome.

For increasing m, the cells Γ_x become smaller and will (normally) eventually contain either only a single data item, or be empty. It should not matter whether we further subdivide empty or singleton cells. So we expect inferences to be independent of m for sufficiently large m, or at least the limit $m\to\infty$ to exist. In this section we show that this is essentially true.

Prior inferences ($\boldsymbol{D=\phi}$). We first consider the prior (zero data) case $D=\phi$. Recall that $z\in I\!B_0^m$ is some node and $x\in I\!B^m$ a leaf node. Normalization implies $p_z(\phi)=1$ for all z, which is independent of m, hence the prior evidence exists for $m\to\infty$. This is nice, but hardly surprising.

The prior effective model dimension $N_{\vec q_*}$ is more interesting. $D=\phi$ implies $D_z=\phi$ implies $n_z=0$ implies $w(n_{z0},n_{z1})=1$ implies a 50/50 prior chance $g_z(\phi)=\frac{1}{2}$ for a split (see (12)). Recursion (11) reads

$$P_z[N_{\vec q_{z*}}=k+1] = \tfrac{1}{2}\sum_{i=0}^k P_{z0}[N_{\vec q_{z0*}}=i]\cdot P_{z1}[N_{\vec q_{z1*}}=k-i]$$

with $P_z[N_{\vec q_{z*}}=k]=\delta_{k0}$ for $l=m$ and $P_z[N_{\vec q_{z*}}=0]=\frac{1}{2}$ for $l<m$. So the recursion terminates in recursion depth $\min\{k+1,m-l\}$. Hence $P_z[N_{\vec q_{z*}}=k+1]$ is the same for all $m>l+k$, which implies that the limit $m\to\infty$ exists. Furthermore, recursion and termination are independent of z, hence also $a_k:=P_z[N_{\vec q_{z*}}=k]$. So we have to solve the recursion

$$a_{k+1} = \tfrac{1}{2}\sum_{i=0}^k a_i\cdot a_{k-i} \quad\text{with}\quad a_0=\tfrac{1}{2} \tag{13}$$

The first few coefficients can be bootstrapped by hand: $(\frac{1}{2},\frac{1}{8},\frac{1}{16},\frac{5}{128},\frac{7}{256},\frac{21}{1024},\frac{33}{2048},...)$. A closed form can also be obtained: Inserting (13) into $f(x):=\sum_{k=0}^\infty a_k x^{k+1}$ we get $f(x)=\frac{1}{2}[x+f^2(x)]$ with solution $f(x)=1-\sqrt{1-x}$, which has Taylor expansion coefficients

$$a_k = (-)^k\binom{1/2}{k+1} = \frac{1}{2(k+1)4^k}\binom{2k}{k} \sim \frac{1}{2\sqrt{\pi}}\,k^{-3/2}$$

$(a_k)_{k\in I\!N_0}$ is a well-behaved distribution. It decreases fast enough to be a proper measure ($\sum_k a_k=f(1)=1<\infty$), but too slow for the expectation $E[N_{\vec q_*}]=\sum_k k\cdot a_k=\infty$ to exist. This is exactly how a proper non-informative prior on $I\!N$ should look like: as uniform as possible, i.e. slowly decreasing. Further, $P[N_{\vec q_*}<\infty]=\sum_k a_k=1$ implies $P[N_{\vec q_*}<\infty|D]=1$, which shows that the effective dimension is almost surely finite, i.e. infinite (Polya) trees have probability zero.

For the tree height we have $E_z[h_{\vec q_{z*}}(x)]=0$ if $l=m$ and otherwise

$$\begin{aligned} E_z[h_{\vec q_{z*}}(x)] &= \tfrac{1}{2}[1+E_{zx_{l+1}}[h_{\vec q_{zx_{l+1}*}}(x)]] \\ &= \;...\;=1-(\tfrac{1}{2})^{m-l}\to 1 \quad\text{for}\quad m\to\infty \end{aligned}$$

This also implies that the expected average height $E_z[\bar h_{\vec q_{z*}}]=1-(\frac{1}{2})^{m-l}\to 1$. This is the first case where the result is not independent of m for large finite m, but it converges for $m\to\infty$, what is enough for our purpose.

Single data item $\boldsymbol{D=(x)}$. Since $p(x)\equiv 1$ (by symmetry and normalization) and $w_1=1$ are the same as for the $n=0$ case, all prior $n=0$, $m\to\infty$ results remain valid for $n=1$: $g(x)=\frac{1}{2}$, $P[N_{\vec q_*}=k|x]=a_k$, and $E[h_{\vec q_*}(x)|x]\to 1$.

General $\boldsymbol{D}$. We now consider general D. For continuous spaces Γ and non-singular distribution $\dot q$, the probability of observing the same point more than once (multi-points) is zero and hence can, to a certain extend, be ignored. See [Hut04a] for a thorough workout of this case. In order to compute $p(D)$ and other quantities, we recurse (9) down the tree until D_z is either empty or a singleton $D_z=(x)\in\Gamma_z$. We call the depth $m_x:=\ell(z)$ at which this happens, the separation level. In this way, the recursion always terminates. For instance, for $\Gamma=[0,1)$, if $\varepsilon:=\min\{|x^i-x^j|:x^i\neq x^j$ with $x^i,x^j\in D\}$ is the shortest distance, then $m_x<\log_2\frac{2}{\varepsilon}=:m_0<\infty$, since $\varepsilon>0$. At the separation level we can insert the derived formulas for evidence, posterior, dimension, and height. Note, there is no approximation here. The procedure is exact, since we analytically computed the infinite recursion for empty and singleton D.

So we have devised a finite procedure, linear in the data size n, for exactly computing all quantities of interest in the infinite Bayes tree. In the worst case, we have to recurse down to level m_0 for each data point, hence our procedure has computational complexity $O(n\cdot m_0)$. For non-singular prior, the time is actually $O(n)$ with probability 1. So, inference in our mixture tree model is *very* fast. Posterior (weak) convergence/consistency for $m=\infty$ can be shown similarly to the $m<\infty$ case [Hut04a].

5 THE ALGORITHM

What it computes. In the last two sections we derived all necessary formulas for making inferences with our tree model. Collecting pieces together we get the exact algorithm for infinite tree mixtures below. It computes the evidence $p(D)$, the expected tree height $E[h_{\vec q_*}(x)|D]$ at x, the average expected tree height $E[\bar h_{\vec q_*}|D]$, and the model dimension distribution $P[N_{\vec q_*}|D]$. It also returns the number of recursive function calls, i.e. the size of the explicitly generated tree. The size is proportional to n for regular distributions $\dot q$.

The BayesTree algorithm (in pseudo C code) takes arguments $(D[],n,x,N)$; data array $D[0..n-1]\in[0,1)^n$, a point $x\in I\!R$, and an integer N. It returns $(p,h,\bar h,\tilde p[],r)$; the logarithmic data evidence $p\widehat{=}\ln p(D)$, the expected tree height $h\widehat{=}E[h_{\vec q_*}(x)|D]$ at x, the average expected tree height $\bar h\widehat{=}E[\bar h_{\vec q_*}|D]$, the model dimension distribution $\tilde p[0..N-1]\widehat{=}P[N_{\vec q_*}=..|D]$, and the number of recursive function calls r i.e. the size of the generated tree. Computation time is about $N^2 n\log n$ nano-seconds on a 1GHz P4 laptop.

BayesTree(D[],n,x,N)

```
⌈ if (n≤1 and (n==0 or D[0]==x or x∉[0,1)))
  ⌈ if (x∈[0,1)) then h=1; else h=0;
    h̄=1; p=ln(1); r=1;
  ⌊ for(k=0,..,N−1) p̃[k]=a_k;              /* see (13) */
  else
  ⌈ n_0=n_1=0;
    for(i=0,..,n−1)
    ⌈ if (D[i]<1/2) then[ D_0[n_0]=2D[i];     n_0=n_0+1;]
    ⌊                else [D_1[n_1]=2D[i]−1; n_1=n_1+1;]
    (p_0,h_0,h̄_0,p̃_0[],r_0)=BayesTree(D_0[],n_0,2x,N−1);
    (p_1,h_1,h̄_1,p̃_1[],r_1)=BayesTree(D_1[],n_1,2x−1,N−1);
    t=p_0+p_1−ln w(n_0,n_1);
    if (t<100) then p=ln(1/2(1+exp(t)));
               else  p=t−ln(2);
    g=1−1/2 exp(−p);
    if (x∈[0,1)) then h=g·(1+h_0+h_1); else h=0;
    h̄=g·(1+(n_0+1)/(n+2) h̄_0+(n_1+1)/(n+2) h̄_1);
    p̃[0]=1−g;
    for(k=0,..,N−1) p̃[k+1]=g·Σ_{i=0}^{k} p̃_0[i]·p̃_1[k−i];
  ⌊ r=1+r_0+r_1;
⌊ return (p,h,h̄,p̃[],r);
```

How algorithm BayesTree() works. Since evidence $p(D)$ and weight $1/w_n$ can grow exponentially with n, we have to store and use their logarithms. So the algorithm returns $p\widehat{=}\ln p(D)$. In the $n\leq 1$ branch, the closed form solutions $p\widehat{=}\ln p(\phi)=\ln(1)$, $h\widehat{=}E[h_{\vec{q}_*}(x)|\phi \text{ or } x]=1$, $\bar{h}\widehat{=}E[\bar{h}_{\vec{q}_*}|D]=1$, and $\tilde{p}[k]=a_k$ have been used to truncate the recursion. If $D=(x^1)\neq x$, we have to recurse further until x falls in an empty interval. In this case or if $n>1$ we partition D into points left and right of $\frac{1}{2}$. Then we rescale the points to [0,1) and store them in D_0 and D_1, respectively. Array D could have been reused (like in quick sort) without allocating two new arrays. Then, algorithm BayesTree() is recursively called for each partition. The results are combined according to the recursions derived in Section 2. $\ln w$ can be computed from (9) via $\ln n!=\sum_{k=1}^{n}\ln k$. (Practically, pre-tabulating a_k or $n!$ does not improve overall performance). For computing p we need to use $\ln(\frac{1}{2}(1+e^t))\doteq t-\ln 2$ to machine precision for large t in order to avoid numerical overflow.

Remarks. Strictly speaking, the algorithm has runtime $O(n\log n)$, since the sorting effectively runs once through all data at each level. If we assume that the data are presorted or the counts n_z are given, then the algorithm is $O(n)$ [Hut04a]. The complete C code, available from [Hut04a], also handles multi-points.

Note that x passed to BayesTree() is *not* and cannot be used to compute $p(x|D)$. For this, one has to call BayesTree() twice, with D and (D,x), respectively. The quadratic order in N is due to the convolution, which could be reduced to $O(N\log N)$ by transforming it to a scalar product in Fourier space with FFT.

Multiply calling BayesTree(), e.g. for computing the predictive density function $p(x|D)$ on a fine x-grid, is inefficient. But it is easy to see that if we once precompute the evidence $p_z(D_z)$ for all z up to the separation level in time $O(n)$, we can compute "local" quantities like $p(x|D)$ at x in time $O(\log n)$. This is because only the branch containing x needs to be recursed, the other branch is immediately available, since it involves the already pre-computed evidence only. The predictive density $p(x|D)=E[q(x)|D]$ and higher moments, the distribution function $P[x\leq a|D]$, updating D by adding or removing one data item, and most other local quantities can be computed in time $O(\log n)$ by such a linear recursion.

A good way of checking correctness of the implementation *and* of the derived formulas, is to force some *minimal* recursion depth m'. The results must be independent of m', since the closed-form speedups are exact and applicable anywhere beyond the separation level.

Numerical example. To get further insight into the behavior of our model, we numerically investigated some example distributions $\dot{q}()$. We have chosen elementary functions, which can be regarded as prototypes for more realistic functions. They include the Beta, linear, a singular, piecewise constant distributions with finite and infinite Bayes trees, and others. These examples on [0,1) also shed light on the other spaces discussed in Section 2, since they are isomorphic. The posteriors, model dimensions, and tree heights, of the singular distribution $\dot{q}(x)=2/\sqrt{1-x}$ are plotted in Figure 1 for random samples D of sizes $n=10^0,...,10^5$. The posterior $p(x|D)$ clearly converges for $n\to\infty$ to the true distribution $\dot{q}()$, accompanied by a (necessary) moderate growth of the effective dimension. For $n=10$ we show the data points. It is visible how each data point pulls the posterior up, as it should be ("one sample seldom comes alone"). The expected tree height $E[h(x)|D]$ correctly reflects the local needs for (non)splits, i.e. is larger near the singularity at $x=1$. The other examples display a similar behavior (see [Hut04a]).

6 DISCUSSION

We presented a Bayesian model on infinite trees, where we split a node into two subtrees with prior probability $\frac{1}{2}$, and uniform choice of the probability assigned to each subtree. We devised closed form expressions

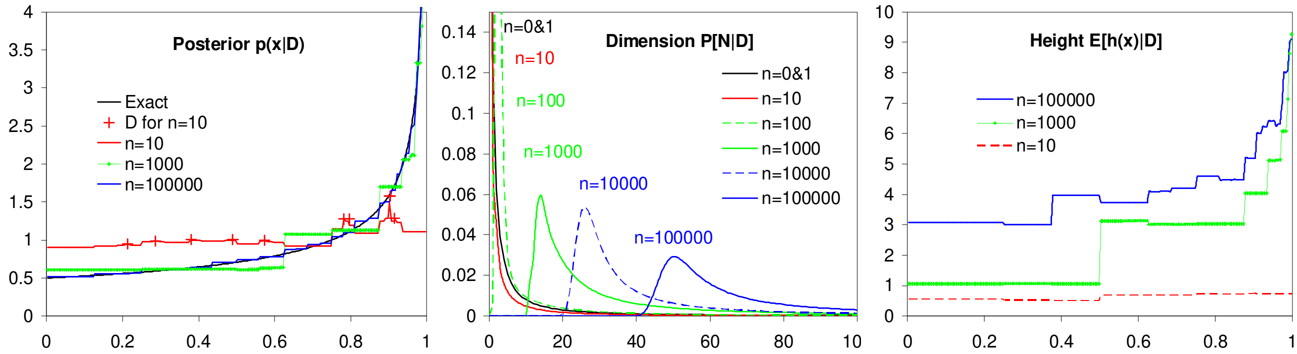

Figure 1: BayesTree() results for a prototypical proper singular distribution $\dot{q}(x) = 2/\sqrt{1-x}$.

for various inferential quantities of interest at the data separation level, which led to an exact algorithm with runtime essentially linear in the data size. The theoretical and numerical model behavior was very reasonable, e.g. consistency (no underfitting) and low finite effective dimension (no overfitting).

There are various natural generalizations of our model. The splitting probability $p(s)$ could be chosen different from $\frac{1}{2}$, k-ary trees could be allowed, and the uniform prior over subtrees could be generalized to Beta/Dirichlet distributions. We were primarily interested in the case of zero prior knowledge, hence zero model (hyper)parameters, but the generalizations above make the model flexible enough, in case prior knowledge needs to be incorporated. The dependency on $p(s)$ is particularly interesting [Hut04a]. The expected entropy can also be computed by allowing fractional counts n_z and noting that $x\ln x = \frac{d}{dx}x^\alpha|_{\alpha=1}$ [Hut02]. A sort of maximum a posteriori (MAP) tree skeleton can also easily be read off from (9). A node Γ_z in the MAP-like tree is a leaf iff $\frac{p_{z0}(D_{z0})p_{z1}(D_{z1})}{w(n_{z0},n_{z1})} < 1$. A challenge is to generalize the model from piecewise constant to piecewise linear continuous functions, at least for $\Gamma = [0,1)$. Independence of subtrees no longer holds, which was key in our analysis.

References

[DLR77] A. P. Dempster, N. Laird, and D. Rubin. Maximum likelihood estimation for incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, Series B 39:1–38, 1977.

[EW95] M. Escobar and M. West. Bayesian density estimation and inference using mixtures. *Journal of the American Statistical Association*, 90:577–588, 1995.

[Fer73] T. S. Ferguson. On the mathematical foundations of theoretical statistics. *Annals of Statistics*, 1(2):209–230, 1973.

[GM03] A. G. Gray and A. W. Moore. Nonparametric density estimation: Toward computational tractability. In *SIAM International Conf. on Data Mining*, volume 3, 2003.

[Goo83] I. J. Good. Explicativity, corroboration, and the relative odds of hypotheses. In *Good thinking: The Foundations of Probability and its applications*. University of Minnesota Press, Minneapolis, MN, 1983.

[Hut02] M. Hutter. Distribution of mutual information. In *Advances in Neural Information Processing Systems 14*, pages 399–406, Cambridge, MA, 2002. MIT Press.

[Hut04a] M. Hutter. Additional material to article. *http://www.idsia.ch/~marcus/ai/bayestreex.htm*, 2004.

[Hut04b] M. Hutter. *Universal Artificial Intelligence: Sequential Decisions based on Algorithmic Probability*. Springer, Berlin, 2004. 300 pages, http://www.idsia.ch/~marcus/ai/uaibook.htm.

[Jay03] E. T. Jaynes. *Probability Theory: The Logic of Science*. Cambridge University Press, Cambridge, MA, 2003.

[Lav94] M. Lavine. More aspects of Polya tree distributions for statistical modelling. *Annals of Statistics*, 22:1161–1176, 1994.

[Lem03] J. C. Lemm. *Bayesian Field Theory and Approximate Symmetries*. Johns Hopkins University Press, 2003.

[Mac03] D. J. C. MacKay. *Information theory, inference and learning algorithms*. Cambridge University Press, Cambridge, MA, 2003.

[PH04] J. Poland and M. Hutter. Convergence of discrete MDL for sequential prediction. In *Proc. 17th Annual Conf. on Learning Theory (COLT-2004)*, volume 3120 of *LNAI*, pages 300–314, Banff, 2004. Springer, Berlin.

[PW02] S. Petrone and L. Wasserman. Consistency of Bernstein polynomial posteriors. *Journal of the Royal Statistical Society*, B 64:79–100, 2002.