Bayesian Bin Distribution Inference and Mutual Information

Dominik Endres and Peter Földiák

***Abstract*—We present an exact Bayesian treatment of a simple, yet sufficiently general probability distribution model. We consider piecewise-constant distributions $P(X)$ with uniform (second-order) prior over location of discontinuity points and assigned chances. The predictive distribution and the model complexity can be determined completely from the data in a computational time that is linear in the number of degrees of freedom and quadratic in the number of possible values of X. Furthermore, exact values of the expectations of entropies and their variances can be computed with polynomial effort. The expectation of the mutual information becomes thus available, too, and a strict upper bound on its variance. The resulting algorithm is particularly useful in experimental research areas where the number of available samples is severely limited (e.g., neurophysiology). Estimates on a simulated data set provide more accurate results than using a previously proposed method.**

***Index Terms*—Bayesian inference, entropy, model selection, mutual information.**

I. Introduction

THE small number of samples available in many areas of experimental science is a serious limitation in calculating distributions and information. For instance, such limitations are typical in the neurophysiological recording of neural activity. Computing entropies and mutual information from data of limited sample size is a difficult task. A related problem is the estimation of probability distributions and/or densities. In fact, once a good density estimate is available, one could also expect the entropy estimates to be satisfactory. One of the several approaches proposed in the past is kernel-based estimation, having the advantage of being able to model virtually any density, but suffering from a heavy bias, as reported by [1]. Another category consists of parametrized estimation methods, which choose some class of density function and then determine a set of parameters that best fit the data by maximizing their likelihood. However, maximum-likelihood approaches are prone to overfitting. A common remedy for this problem is cross validation (see, e.g., [2]), which, while it appears to work in many cases, can at best be regarded as an approximation.

Overfitting occurs especially when the size of the data set is not large enough compared to the number of degrees of freedom of the chosen model. Thus, as a compromise between the two aforementioned methods, *mixture models* have recently attracted considerable attention (see, e.g., [3]). Here, a mixture of simple densities (Gaussians are quite common) are used to model the data. The most popular method for determining its parameters is Expectation Maximization, first described by [4], which, while having nice convergence properties, is still aiming at maximizing the likelihood. The question of how to determine the best number of model parameters therefore remains unanswered in this framework.

This answer can be given by Bayesian inference. In fact, when one is willing to accept some very natural consistency requirements, it provides the only valid answer, as demonstrated by [5]. The difficulty lies in carrying out the necessary computations. More specifically, the integrations involved in computing the posterior densities of the model parameters are hard to tackle. To date, only fairly simple and hence not very general density models could be treated exactly. Thus, a variety of approximation schemes have been devised, e.g., Laplace approximation (where the true density is replaced by a Gaussian), Monte Carlo simulation, or variational Bayes (a good introduction can be found in [6]). As is always the case with approximations, they will work in some cases and fail in others. Two other noteworthy approaches to dealing with the overfitting problem are minimum message length (MML) [7] and minimum description length (MDL) [8], which are similar to Bayesian inference.

In this paper, we will present an exactly tractable model. While still simple in construction, it is sufficiently general to model any distribution (or density). The model's complexity is fully determined by the data, too. Moreover, it also yields exact expectations of entropies and their variances. Thus, an exact expectation of the mutual information and a strict upper bound on its variance can be computed as well.

II. Bayesian Binning

Suppose we had a discrete random variable X, which could take on the values $k = 0, \ldots, K-1$. Furthermore, assume that a notion of similarity between any two instances x_1, x_2 of X could be quantified in a meaningful way by $x_1 - x_2$. It is then natural to try to model the distribution of X by $M+1 \leq K$ contiguous, nonoverlapping bins, such that the bins cover the whole domain of X. Each bin has a probability P_m, $m = 0, \ldots, M$, subject to the normalization constraint $\sum_{m=0}^{M} P_m = 1$. This probability is evenly distributed among the possible values of X in the bin (see Fig. 1). If a bin ends at k_m and the previous one at k_{m-1}, then

$$\text{for all } k_{m-1} < k \leq k_m : \; P(X = k) = \frac{P_m}{\Delta k_m} \qquad (1)$$

Manuscript received August 16, 2004; revised May 12, 2005.

The authors are with the School of Psychology, University of St.Andrews, St. Andrews KY16 9JP, Scotland, U.K. (e-mail: dme2@st-andrews.ac.uk; pf2@st-andrews.ac.uk).

Communicated by P. L. Bartlett, Associate Editor for Pattern Recognition, Statistical Learning and Inference.

Digital Object Identifier 10.1109/TIT.2005.856954

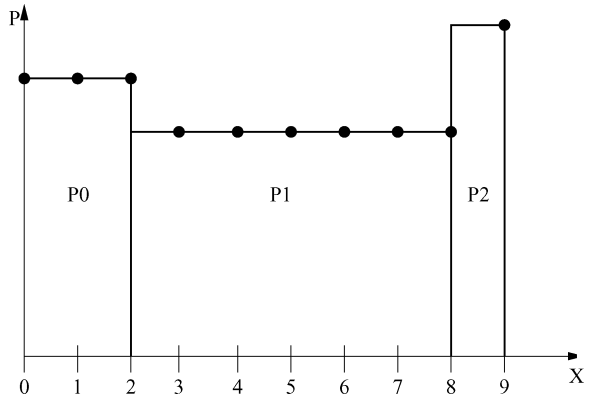

Fig. 1. An example configuration for $K = 10$ values of X, three bins ($M = 2$ interval boundaries), containing the probabilities P_m. Here, $P(0) = P(1) = P(2) = \frac{P_0}{3}$, $P(3) = \cdots = P(8) = \frac{P_1}{6}$, $P(9) = P_2$. The points indicate which values of X belong to a bin. The algorithm iterates over all possible configurations to find the expected probability distribution and other relevant averages.

with $\Delta k_m = k_m - k_{m-1}$, $k_{-1} = -1$, and $k_M = K - 1$, i.e., the first bin starts at $X = 0$ and the last one includes $X = K - 1$. Assume now we had a multiset $D = \{x_1, x_2, \ldots, x_N\}$ of N points drawn independently from the distribution which we would like to infer. Given the model parameterized by M and $\{(P_m, k_m)\}$, the probability of the data then is given (up to a factor) by a multinomial distribution

$$P(D|M, \{(P_m, k_m)\}) = \prod_{m=0}^{M} \left(\frac{P_m}{\Delta k_m}\right)^{n_m} \tag{2}$$

where n_m is the number of points in bin m. One might argue that a multinomial factor should be inserted here, because the data points are not ordered. This would, however, only amount to a constant factor that cancels out when averages (posteriors, expectations of variables, etc.) are computed. It will therefore be dropped. The factors $\Delta k_m^{-n_m}$ express the intention of modeling Δk_m possible values of X by the same probability. From an information-theoretic perspective, we are trying to find a simpler coding scheme for the data while preserving the information present in them: The message "$X = k$" for all $k : k_{m-1} < k \leq k_m$ would be represented by a code element of the same length $\log(\frac{\Delta k_m}{P_m})$. In contrast, were the $\Delta k_m^{-n_m}$ absent, then the message "$X = k$" for each $k : k_{m-1} < k \leq k_m$ would be represented by the same code element, i.e., an information reduction transformation would have been applied to the data.

We now want to compute the evidence of a model with M bins, i.e., $P(D|M)$. It is obtained by multiplying the likelihood (see (2)) with a suitable prior $p(\{(P_m, k_m)\}|M)$ to yield $p(D, \{(P_m, k_m)\}|M)$. This density is then marginalized with respect to (w.r.t.) P_m and k_m, which is done by integration and summation, respectively. The summation boundaries for the k_m have to be chosen so that each bin covers at least one possible value of X. Since the bins may not overlap, $k_0 = 0 \ldots K - 1 - M$, $k_1 = k_0 + 1 \ldots K - M$, etc. Because the P_m represent probabilities, their integrations run from 0 to 1 subject to the normalization constraint, which can be enforced via a Dirac $\delta()$ function

$$P(D|M) = \sum\sum\nolimits_{\{k\}} \int_0^1 d\vec{P} P(D|M, \{(P_m, k_m)\}) \times p(\{(P_m, k_m)\}|M) \tag{3}$$

where

$$\int_0^1 d\vec{P} = \int_0^1 dP_0 \int_0^1 dP_1 \ldots \int_0^1 dP_M \delta(1 - \sum_{m=0}^{M} P_m) \tag{4}$$

and

$$\sum\sum\nolimits_{\{k\}} = \sum_{k_0=0}^{K-1-M} \sum_{k_1=k_0+1}^{K-M} \cdots \sum_{k_{M-1}=k_{M-2}+1}^{K-2} . \tag{5}$$

Note that the prior $p(\{(P_m, k_m)\}|M)$ is a probability density, because the P_m are real numbers.

III. Computing the Prior $p(\{(P_m, k_m)\}|M)$

We shall make a noninformative prior assumption, namely, that all possible configurations of $\{(P_m, k_m)\}$ are equally likely prior to observing the data. The prior can be written as

$$p(\{(P_m, k_m)\}|M) = p(\{P_m\}|\{k_m\}, M) P(\{k_m\}|M) . \tag{6}$$

Note that the second factor on the right-hand side (RHS) is not a probability density, because there is only a finite number of configurations for the k_m. We now make the assumption that the probability contained in a bin is independent of the bin size, i.e.,

$$p(\{P_m\}|\{k_m\}, M) = p(\{P_m\}|M).$$

This is certainly not the only possible choice, but a common one: in the absence of further prior information, independence assumptions can be justified, since they maximize the prior's entropy. Thus, (6) becomes

$$p(\{(P_m, k_m)\}|M) = p(\{P_m\}|M) P(\{k_m\}|M). \tag{7}$$

Since all models are to be equally likely, the prior is constant (denoted by $c(M)$) w.r.t. k_m and P_m, and normalized

$$\sum\sum\nolimits_{\{k\}} \underbrace{\int_0^1 d\vec{P}}_{\frac{1}{M!}} \times c(M) = 1. \tag{8}$$

Carrying out the integrals is straightforward (see, e.g., [9]) and yields the normalization constant of a Dirichlet distribution. The value of the sums is given by

$$\sum\sum\nolimits_{\{k\}} 1 = \binom{K-1}{M}. \tag{9}$$

This is, of course, just the number of possibilities of distributing M ordered bin boundaries across $K - 1$ places. Due to the assumed independence between P_m and k_m, we can write

$$p(\{P_m\}|M) = M! \tag{10}$$

$$P(\{k_m\}|M) = \frac{(K - M - 1)! M!}{(K - 1)!} \tag{11}$$

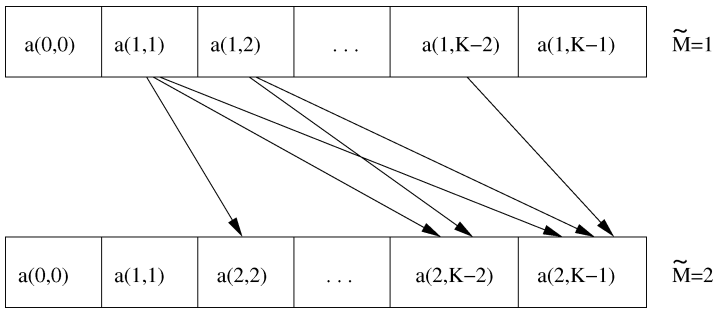

Fig. 2. After all the evidence contributions $a(\bar{M},\bar{K})$ have been evaluated to compute the evidence of a model with $\bar{M}$ intersections, they can be reused to compute the $a(\bar{M}+1,\bar{K})$. The arrows indicate which $a(\bar{M},\bar{K})$ enter into the calculation for an $a(\bar{M}+1,\bar{K})$.

and thus, the prior is

$$c(M)=\frac{(K-M-1)!M!^2}{(K-1)!}. \tag{12}$$

IV. Computing the Evidence

To compute (3), we rewrite it as

$$P(D|M)=\sum\sum_{\{k\}}P(D|\{k_m\},M)P(\{k_m\}|M) \tag{13}$$

where

$$P(D|\{k_m\},M)=\int_0^1 d\vec{P}\prod_{m=0}^{M}\left(\frac{P_m}{\Delta k_m}\right)^{n_m}p(\{P_m\}|M). \tag{14}$$

Given a configuration of the $\{k_m\}$, the $\{n_m\}$ are fixed, and thus the integrals can be carried out (see, e.g., [9]) to yield

$$\int_0^1 d\vec{P}\prod_{m=0}^{M}P_m^{n_m}=\frac{\prod_{m=0}^{M}\Gamma(n_m+1)}{\Gamma(N+M+1)} \tag{15}$$

i.e., the normalization integral of a Dirichlet distribution again. Thus,

$$\begin{aligned}P(D|\{k_m\},M)&=\prod_{m'=0}^{M}(k_{m'}-k_{m'-1})^{-n_{m'}}\times\\&\quad\times\frac{\prod_{m=0}^{M}n_m!}{(N+M)!}M!\\&=\frac{M!}{(N+M)!}\prod_{m=0}^{M}\frac{n_m!}{\Delta k_m^{n_m}}\ .\end{aligned} \tag{16}$$

For a speedy evaluation of the sums in (13), we suggest the following iterative scheme:

let

$$a(0,\tilde{K})=\frac{n_0!}{(\tilde{K}+1)^{n_0}} \tag{17}$$

where n_0 is the total number of data points for which $k\le\tilde{K}$ (i.e., the number of points in the current bin 0). Furthermore, define

$$a(\tilde{M}+1,\tilde{K})=\sum_{\tilde{k}=\tilde{M}}^{\tilde{K}-1}a(\tilde{M},\tilde{k})\frac{n_{\tilde{M}+1}!}{(\tilde{K}-\tilde{k})^{n_{\tilde{M}+1}}} \tag{18}$$

where $n_{\tilde{M}+1}$ is the total number of data points for which $\tilde{k}<k\le\tilde{K}$ (i.e., the number of points in the current bin $\tilde{M}+1$). In other words, to compute the contribution to (13) which has $\tilde{M}+1$ bin boundaries in the interval $0,\ldots,\tilde{K}$, let boundary $\tilde{M}+1$ move between position $\tilde{M}$ (because the previous $\tilde{M}$ boundaries must at least occupy the positions $0,\ldots,\tilde{M}-1$) and $\tilde{K}-1$ (because bin $\tilde{M}+1$ must at least have width 1). For each of these positions, multiply the factor for bin $\tilde{M}+1$ (which ranges from $\tilde{k}+1$ to $\tilde{K}$) with the contribution for $\tilde{M}$ bin boundaries in the interval $0,\ldots,\tilde{k}$ and add.

By induction, we hence obtain

$$\begin{aligned}a(\tilde{M},\tilde{K})&=\sum_{k_{\tilde{M}-1}=\tilde{M}-1}^{\tilde{K}-1}\sum_{k_{\tilde{M}-2}=\tilde{M}-2}^{k_{\tilde{M}-1}-1}\cdots\sum_{k_0=0}^{k_1-1}\prod_{m=0}^{M}\frac{n_m!}{\Delta k_m^{n_m}}\\&=\sum_{k_0=0}^{\tilde{K}-\tilde{M}}\cdots\sum_{k_{\tilde{M}-1}=k_{\tilde{M}-2}+1}^{\tilde{K}-1}\prod_{m=0}^{M}\frac{n_m!}{\Delta k_m^{n_m}}.\end{aligned} \tag{19}$$

Inserting (16) into (13) and using (11) yields

$$P(D|M)=\frac{(K-M-1)!M!^2}{(K-1)!(N+M)!}a(M,K-1). \tag{20}$$

This way of organizing the calculation is computationally similar to the sum–product algorithm [10], the messages being passed from one $\tilde{M}$-level to the next are the $a(\tilde{M},\tilde{K})$, whereas within one level, a sum over the messages from the previous level (times a factor) is performed.

To compute $a(M,K-1)$, we need all the $a(M-1,\tilde{K})$, $M-1\le\tilde{K}<K-1$ (see Fig. 2). While we are at it, we can just as well evaluate $a(M-1,K-1)$, which does not increase the overall computation complexity. Thus, we get the evidences for models with less than M intersections with little extra effort. Moreover, in an implementation it is sufficient to store the $a(M,\tilde{K})$ in a one-dimensional array of length K that is overwritten in reverse order as one proceeds from $M-1$ to M (because $a(M-1,K-2)$ is no longer needed once $a(M,K-1)$ is computed, etc.). In pseudocode (where $\mathrm{getCount(k1,k2)}$ function returns the number of observed points for which $k1<k\le k2$):

1) Initialize $\mathrm{a}[0\ldots\mathrm{K}-1] := 0$, $\mathrm{evidences}[0\ldots\mathrm{M}] := 0$
2) for $\mathrm{k} := 0$ to $\mathrm{K}-1$ do
   a) $\mathrm{n} := \mathrm{getCount}(-1, \mathrm{k})$
   b) $\mathrm{a}[\mathrm{k}] := \frac{n!}{(k+1)^n}$
3) $\mathrm{evidences}[0] := \mathrm{a}[\mathrm{K}-1]/\mathrm{N}!$
4) for $\mathrm{m} := 1$ to M do
   a) if $\mathrm{m} = \mathrm{M}$ then $\mathrm{lb} := \mathrm{K}-1$ else $\mathrm{lb} := \mathrm{m}$
   b) for $\mathrm{k} := \mathrm{K}-1$ downto lb do
      i) $\mathrm{a}[\mathrm{k}] := 0$
      ii) for $\mathrm{kk} := \mathrm{m}-1$ to $\mathrm{k}-1$ do
         A) $\mathrm{n} := \mathrm{getCount}(\mathrm{kk}, \mathrm{k})$
         B) $\mathrm{a}[\mathrm{k}] := \mathrm{a}[\mathrm{k}] + \mathrm{a}[\mathrm{kk}] \times \frac{n!}{(k-kk)^n}$
   c) $\mathrm{evidences}[\mathrm{m}] := \mathrm{a}[\mathrm{K}-1] \times \frac{(K-1-m)!m!^2}{(K-1)!(N+m)!}$

Step 4a) is not essential, but it saves some computational effort: once the main loop 4 reaches M, only $a(M, K-1)$ needs to be calculated. The $a(M, k < K-1)$ would only be necessary if the evidence for $M+1$ bin boundaries were to be evaluated.

In a real-world implementation, it might be advisable to construct a lookup table for

$$\mathrm{getCount}(\mathrm{k1}, \mathrm{k2})!/(k2-k1)^{\mathrm{getCount}(\mathrm{k1},\mathrm{k2})}$$

since these quantities would otherwise be evaluated multiple times in step 4b)ii)B), as soon as $M > 2$.

A look at the main loop 4) shows that the computational complexity of this algorithm is $\mathcal{O}(MK^2)$ (provided that $M \ll K$, i.e., the number of bins is much smaller than the number of possible values of X), or more generally, $\mathcal{O}(K^3)$ (because $M \leq K-1$). This is significantly faster than the naïve approach of simply trying all possible configurations, which would yield $\mathcal{O}(K^M)$.

V. Evaluating the Model Posterior $P(M|D)$, the Distribution $P(k|M, D)$, and Its Variance

Once the evidence is known, we can proceed to determine the relative probabilities of different M, the *model posterior*

$$P(M|D) = \frac{P(D|M)P(M)}{P(D)} \tag{21}$$

where $P(M)$ is the model prior and

$$P(D) = \sum_m P(D|m)P(m).$$

The sum over m includes all models which we choose to include into our inference process, therefore, the conditioning on a particular m is marginalized. It is thus customary to refer to $P(D)$ as "the probability of the data." This is somewhat misleading, because $P(D)$ is still not only conditioned on the set of m we chose, but also on the general model class we consider (here: probability distributions that consist of a number of bins).

The predictive distribution of X can be calculated via the evidence as well. Note that

$$P(k_1, k_2, \ldots, k_R|M, D) = E\left[P(k_1)P(k_2)\ldots P(k_R)|M, D\right] \tag{22}$$

i.e., the joint predictive probability of $k_1, k_2, \ldots, k_R$ is the expectation of the product of their probabilities given M and D. Thus, if we want the value of the predictive distribution of X at a particular k', we can simply add this k' to the multiset D. Call this extended multiset $D' = D \cup k'$, then

$$P(k'|D, M) = \frac{P(D'|M)}{P(D|M)}\,. \tag{23}$$

The choice of $P(M)$ will usually be noninformative (e.g., uniform), unless we have reasons to prefer certain models over others.

Likewise, to obtain the variance, add k' twice: $D'' = D \cup k' \cup k'$. Then

$$\mathrm{Var}(P(k'|D, M)) = \frac{P(D''|M)}{P(D|M)} - \left(\frac{P(D'|M)}{P(D|M)}\right)^2. \tag{24}$$

VI. Inferring Probability Densities

The algorithm can also be used to estimate probability densities. To do so, replace all occurrences of Δk_m with $(\Delta k_m)\Delta x$, where Δx is the interval between k and $k+1$. This yields a discretized approximation to the density. The discretization is just another model parameter which can be determined from the data

$$P(\Delta x|D) = \frac{p(D|\Delta x)P(\Delta x)}{\sum_{\Delta x} p(D|\Delta x)P(\Delta x)} \tag{25}$$

where $\sum_{\Delta x}$ runs over all possible values of Δx which we choose to include. $p(D|\Delta x)$ is computed in the same fashion as $P(D)$ in (21) for a given Δx, except that the data are now assumed to be continuous, hence, the probability turns into a density.

The dependence of $P(D|M, \Delta x)$ on Δx is through K (see (3)): if one tries to find a suitable discretization of the interval $[0, b]$, then $K = \frac{b}{\Delta x}$.

VII. Model Selection Versus Model Averaging

Once $P(M|D)$ is determined, we could use it to select an M'. However, making this decision involves "contriving" information, namely, $-\log(P(M'|D))$ nats. Thus, unless one is forced to do so, one should refrain from it and rather average any predictions over all possible M. Since $M \leq K-1$, computing such an average has a computational cost of $O(K^3)$. If the structure of the data allows for it, it is possible to reduce this cost by finding a range of M that does not increase linearly with K, without a high risk of not including a model even though it provides a good description of the data. In analogy to the significance levels of orthodox statistics, we shall call this risk α. If the posterior of M is unimodal (which it has been in most observed cases, see, e.g., Seection IX), we can then choose the smallest interval of Ms around the maximum of $P(M|D)$ such that

$$P(M_{\min} \leq M \leq M_{\max}|D) \leq 1-\alpha \tag{26}$$

and carry out the averages over this range of M.

VIII. Computing the Entropy and its Variance

To evaluate the entropy of the distribution $P(X)$, we first compute the entropy of (1)

$$H(P(X|M,\{P_m,k_m\})) = -\sum_{m=0}^{M}\sum_{k=k_{m-1}+1}^{k_m}\frac{P_m}{\Delta k_m}\log\left(\frac{P_m}{\Delta k_m}\right) = -\sum_{m=0}^{M}P_m\log(P_m)+\sum_{m=0}^{M}P_m\log(\Delta k_m) \tag{27}$$

where $\log(x)$ is the natural logarithm. This expression must now be averaged over the posterior distributions of $\{(P_m,k_m)\}$ and possibly M to obtain the expectation of $H(P(X|D))$. Instead of carrying this out for (27) as a whole, it is easier to do it term-by-term, i.e., we need to calculate the expectations of $P_m\log(P_m)$ and $P_m\log(k_m-k_{m-1})$. Generally speaking, if we want to compute the average of any quantity that is a function of the probability in a bin m' for a given M, we can proceed in the following fashion: call this function $f(P_{m'})$. Its expectation w.r.t. the $\{P_m\}$ is then

$$E\left[f(P_{m'})|\{k_m\},M,D\right]=\frac{\int_0^1 d\vec{P}f(P_{m'})P(D|\{(P_m,k_m\},M)}{I(\{k_m\})} \tag{28}$$

where, by virtue of (14) and (15)

$$I(\{k_m\}) = \int_0^1 d\vec{P}P(D|\{(P_m,k_m\},M) = \frac{1}{(N+M)!}\prod_{m=0}^{M}\frac{n_m!}{\Delta k_m^{n_m}} \tag{29}$$

because $p(\{P_m\})$ is a constant, otherwise it would have to be included in the integrations. Note that, as far as the counts in the bins n_m are concerned, $E\left[f(P_{m'})|\{k_m\},M,D\right]$ depends only on $n_{m'}$ (and possibly N, the total number of data points). This can be verified by integrating the numerator of (28) over all P_m, $m\neq m'$. Therefore,

$$E[f(P_{m'})|M,D] = \sum\sum_{\{k\}}\int_0^1 d\vec{P}f(P_{m'})p(\{(P_m,k_m)\}|M,D) = \frac{\sum\sum_{\{k\}}P(\{k_m\})p(\{P_m\})E[f(P_{m'})|\{k_m\},M,D]I(\{k_m\})}{P(D|M)}. \tag{30}$$

When (29) is inserted here, it becomes apparent that this expectation has the same form as (13) after inserting (16). Combined with the fact that $E\left[f(P_{m'})|\{k_m\},M,D\right]$ depends only on $n_{m'}$, this means that the above described iterative computation scheme ((17) and (18)) can be employed for its evaluation. All one needs to do is to substitute

$$n_{m'}!\rightarrow n_{m'}!E\left[f(P_{m'})|\{k_m\},M,D\right]$$

i.e., (18) is replaced by

$$a(\tilde{M}+1=m',\tilde{K}) = \sum_{\tilde{k}=\tilde{M}}^{\tilde{K}-1}a(\tilde{M},\tilde{k})\frac{n_{m'}!}{\Delta k_{m'}^{n_{m'}}}E\left[f(P_{m'})|\{k_m\},M,D\right] \tag{31}$$

where $k_{m'-1}=\tilde{k}$ and $k_{m'}=\tilde{K}$. The $a(\tilde{M},\tilde{K})$ for $\tilde{M}\neq m'$ are the same as before. Note that (31) can also be used if $f(P_{m'})$ depends not only on $P_{m'}$, but also on the boundaries of bin m' (i.e., $k_{m'-1}$ and $k_{m'}$). Generally speaking, (31) can be employed whenever the expectation of a function (given M and D) is to be evaluated, if this function depends only on the parameters of one bin.

For fixed $\{k_m\}$, some of these expectations have been computed before in [9].

A. Computing $E\left[P_{m'}\log(P_{m'})|\{k_m\},M,D\right]$

Using (1) and defining $q(\{k_m\})=\prod_{m=0}^{M}\Delta k_m^{-n_m}$, we obtain

$$E\left[P_{m'}\log(P_{m'})|\{k_m\},M,D\right] = \frac{q(\{k_m\})\int_0^1 d\vec{P}P_{m'}^{n_{m'}+1}\log(P_{m'})\prod_{m\neq m'}P_m^{n_m}}{I(\{k_m\})}. \tag{32}$$

To compute the integrals, note that

$$\int_0^1 d\vec{P}P_{m'}^{n_{m'}+1}\log(P_{m'})\prod_{m\neq m'}P_m^{n_m} = \frac{\partial}{\partial n_{m'}}\int_0^1 d\vec{P}P_{m'}^{n_{m'}+1}\prod_{m\neq m'}P_m^{n_m} \tag{33}$$

which is

$$\begin{aligned}
&= \frac{\partial}{\partial n_{m'}}\frac{\Gamma(n_{m'}+2)\prod_{m\neq m'}^{M}\Gamma(n_m+1)}{\Gamma(\sum_{m=0}^{M}n_m+M+2)}\\
&= \frac{\partial}{\partial n_{m'}}\frac{\Gamma(n_{m'}+2)\Gamma(\sum_{m\neq m'}n_m+M)}{\Gamma(\sum_{m=0}^{M}n_m+M+2)}\\
&\quad\times\frac{\prod_{m\neq m'}^{M}\Gamma(n_m+1)}{\Gamma(\sum_{m\neq m'}n_m+M)}\\
&= \frac{\partial B(n_{m'}+2,\sum_{m\neq m'}n_m+M)}{\partial n_{m'}}\frac{\prod_{m\neq m'}^{M}\Gamma(n_m+1)}{\Gamma(\sum_{m\neq m'}n_m+M)}\\
&= \frac{\prod_{m=0}^{M}n_m!}{(N+M)!}\frac{n_{m'}+1}{N+M+1}h_{n_{m'}+2}^{N+M+1}
\end{aligned} \tag{34}$$

where $B(a,b)$ is the beta function (see, e.g., [11] for details on its derivatives and other properties), and

$$h_a^b=\sum_{i=a}^{b}\frac{1}{i} \tag{35}$$

is the difference between the partial sums of the harmonic series with upper limits $a-1$ and b. The first part of (34) is the normalization integral of the density of the P_m. Hence, their entropy for fixed $\{k_m\}$ is given by (34) divided by (15)

$$E\left[H(\{P_m\})|\{k_m\},M,D\right]=\sum_{m=0}^{M}\frac{n_m+1}{N+M+1}h_{n_m+2}^{N+M+1} \tag{36}$$

which is a rational number. In other words, entropy changes in rational increments as we observe new data points.

Thus,

$$E\left[P_{m'}\log(P_{m'})|\{k_m\},M,D\right]=\frac{n_{m'}+1}{N+M+1}h_{n_{m'}+2}^{N+M+1}. \tag{37}$$

B. Computing $E\left[P_{m'}\log(\Delta k_{m'})|\{k_m\},M,D\right]$

Here

$$E\left[P_{m'}\log(\Delta k_{m'})|\{k_m\},M,D\right] = \frac{q(\{k_m\})\log(\Delta k_{m'})\int_0^1 d\vec{P}P_{m'}^{n_{m'}+1}\prod_{m\neq m'}P_m^{n_m}}{I(\{k_m\})} \quad (38)$$

and by using the same identities as above

$$E\left[P_{m'}\log(\Delta k_{m'})|\{k_m\},M,D\right] = \frac{n_{m'}+1}{N+M+1}\log(\Delta k_{m'})\,. \quad (39)$$

C. Computing the Variance

To compute the variance, we need the square of the entropy

$$\begin{aligned}
&H^2(P(X|M,\{P_m,k_m\}))\\
&= \left(-\sum_{k=0}^{K-1}P(k|M,\{P_m,k_m\})\log(P(k|M,\{P_m,k_m\}))\right)^2\\
&= \left(\sum_{m=0}^{M}P_m\log(P_m)-\sum_{m=0}^{M}P_m\log(\Delta k_m)\right)^2\\
&= \sum_{m'=0}^{M}P_{m'}^2\log^2(P_{m'})\\
&\quad+2\sum_{m'=0}^{M}\sum_{m''=m'+1}^{M}P_{m'}P_{m''}\log(P_{m'})\log(P_{m''})\\
&\quad+\sum_{m'=0}^{M}P_{m'}^2\log^2(\Delta k_{m'})\\
&\quad+2\sum_{m'=0}^{M}\sum_{m''=m'+1}^{M}P_{m'}P_{m''}\log(\Delta k_{m'})\log(\Delta k_{m''})\\
&\quad-2\sum_{m'=0}^{M}P_{m'}^2\log(P_{m'})\log(\Delta k_{m'})\\
&\quad-4\sum_{m'=0}^{M}\sum_{m''=m'+1}^{M}P_{m'}P_{m''}\log(P_{m'})\log(\Delta k_{m''})\,. \quad (40)
\end{aligned}$$

Hence, we need to evaluate the expectations of

1) $P_{m'}^2\log^2(P_{m'})$. See Section VIII-D.
2) $P_{m'}P_{m''}\log(P_{m'})\log(P_{m''})$. See Section VIII-E.
3) $P_{m'}^2\log^2(\Delta k_{m'})$. Can be evaluated along the same lines as (39) and yields
$$\frac{(n_{m'}+1)(n_{m'}+2)}{(N+M+1)(N+M+2)}\log^2(\Delta k_{m'})\,. \quad (41)$$
4) $P_{m'}P_{m''}\log(\Delta k_{m'})\log(\Delta k_{m''})$. Follows from (15) by replacing $n_{m'}$ with $n_{m'}+1$, $n_{m''}$ with $n_{m''}+1$, and yields
$$\frac{(n_{m'}+1)(n_{m''}+1)}{(N+M+1)(N+M+2)}\log(\Delta k_{m'})\log(\Delta k_{m''}). \quad (42)$$
5) $P_{m'}^2\log(P_{m'})\log(\Delta k_{m'})$. Can be evaluated along the same lines as (37) and yields
$$\frac{(n_{m'}+1)(n_{m'}+2)}{(N+M+1)(N+M+2)}h_{n_{m'}+3}^{N+M+2}\log(\Delta k_{m'}). \quad (43)$$
6) $P_{m'}P_{m''}\log(P_{m'})\log(\Delta k_{m''})$. Can be evaluated along the same lines as (37) and yields
$$\frac{(n_{m'}+1)(n_{m''}+1)}{(N+M+1)(N+M+2)}h_{n_{m'}+2}^{N+M+2}\log(\Delta k_{m''})\,. \quad (44)$$

In an actual implementation, steps 1), 3), and 5) can be computed in one run, because they contain only terms that refer to the bin m'. Likewise, the remaining three steps can be evaluated together. However, since they depend on the parameters of two distinct bins, the iteration over the bin configuration has to be adapted: assume $m'<m''$. This is no restriction, since m' and m'' can always be exchanged to satisfy this constraint. As we will show later, the quantities of interest can be expressed as products of expectations $E\left[f(P_{m'})|\{k_m\},M,D\right]E\left[f(P_{m''})|\{k_m\},M,D\right]$. The iteration rule (18) then is

$$\begin{aligned}
&a(\tilde{M}+1=m'',\tilde{K})\\
&\quad=\sum_{\tilde{k}=\tilde{M}}^{\tilde{K}-1}a(\tilde{M},\tilde{k})\frac{n_{m''}!}{\Delta k_{m''}^{n_{m''}}}E\left[f(P_{m''})|\{k_m\},M,D\right] \quad (45)
\end{aligned}$$

and

$$\begin{aligned}
&a(\tilde{M}+1=m',\tilde{K})\\
&\quad=\sum_{\tilde{k}=\tilde{M}}^{\tilde{K}-1}a(\tilde{M},\tilde{k})\frac{n_{m'}!}{\Delta k_{m'}^{n_{m'}}}E\left[f(P_{m'})|\{k_m\},M,D\right]\,. \quad (46)
\end{aligned}$$

D. Computing $E\left[P_{m'}^2\log^2(P_{m'})|\{k_m\},M,D\right]$

Since

$$\frac{\partial^2 B(a+1,b+1)}{\partial a^2}=\int_0^1 dx\log^2(x)x^a(1-x)^b \quad (47)$$

we can use the same method of calculation as above. Thus, we obtain

$$\begin{aligned}
&E\left[P_{m'}^2\log^2(P_{m'})|\{k_m\},M,D\right]\\
&\quad=\frac{(n_{m'}+1)(n_{m'}+2)}{(N+M+1)(N+M+2)}\left[\left(h_{n_{m'}+3}^{N+M+2}\right)^2+{}_2h_{n_{m'}+3}^{N+M+2}\right] \quad (48)
\end{aligned}$$

where

$${}_2h_a^b=\sum_{i=a}^{b}\frac{1}{i^2}. \quad (49)$$

E. Computing $E\left[P_{m'}P_{m''}\log(P_{m'})\log(P_{m''})|\{k_m\},M,D\right]$

To compute this expectation, we need to evaluate

$$\begin{aligned}
&\int_0^1 d\vec{P}P_{m'}^{n_{m'}+1}P_{m''}^{n_{m''}+1}\log(P_{m'})\log(P_{m''})\prod_{m\neq m',m''}P_m^{n_m}\\
&\quad=\frac{\partial^2}{\partial n_{m'}\partial n_{m''}}\left(\int_0^1 d\vec{P}P_{m'}^{n_{m'}+1}P_{m''}^{n_{m''}+1}\prod_{m\neq m',m''}P_m^{n_m}\right) \quad (50)
\end{aligned}$$

and then divide it by a normalization constant. The integrals in the last expression can be rewritten, using beta functions, to yield

$$B(n_{m'}+2,n_{m''}+2)B(n_{m'}+n_{m''}+4,N_R+M-1)\times R \quad (51)$$

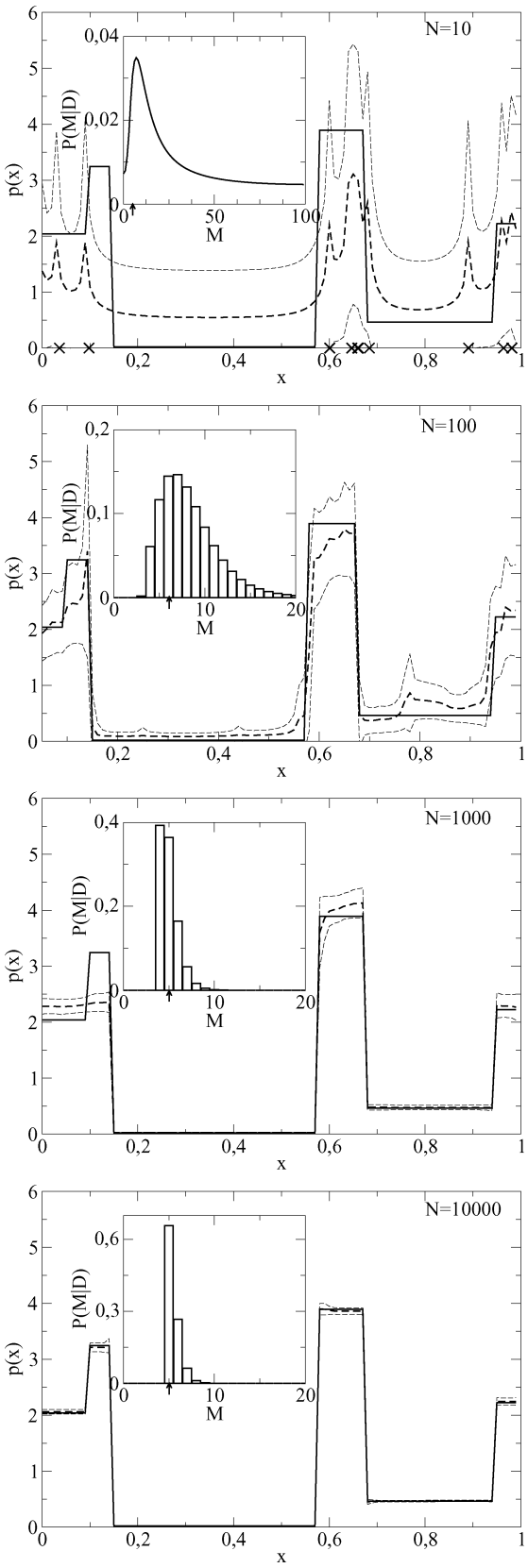

Fig. 3. From top to bottom: densities and posteriors for 10,100,1000 and 10000 data points. Expected probability density (thick dashed line) plus/minus one standard deviation (thin dashed lines), compared to the true density (thick solid line). Insets: posterior distribution of M, the number of intersections. The arrows indicate the true value of $M = 5$. The X's on the abscissa of the graph for 10 data points mark the data points.

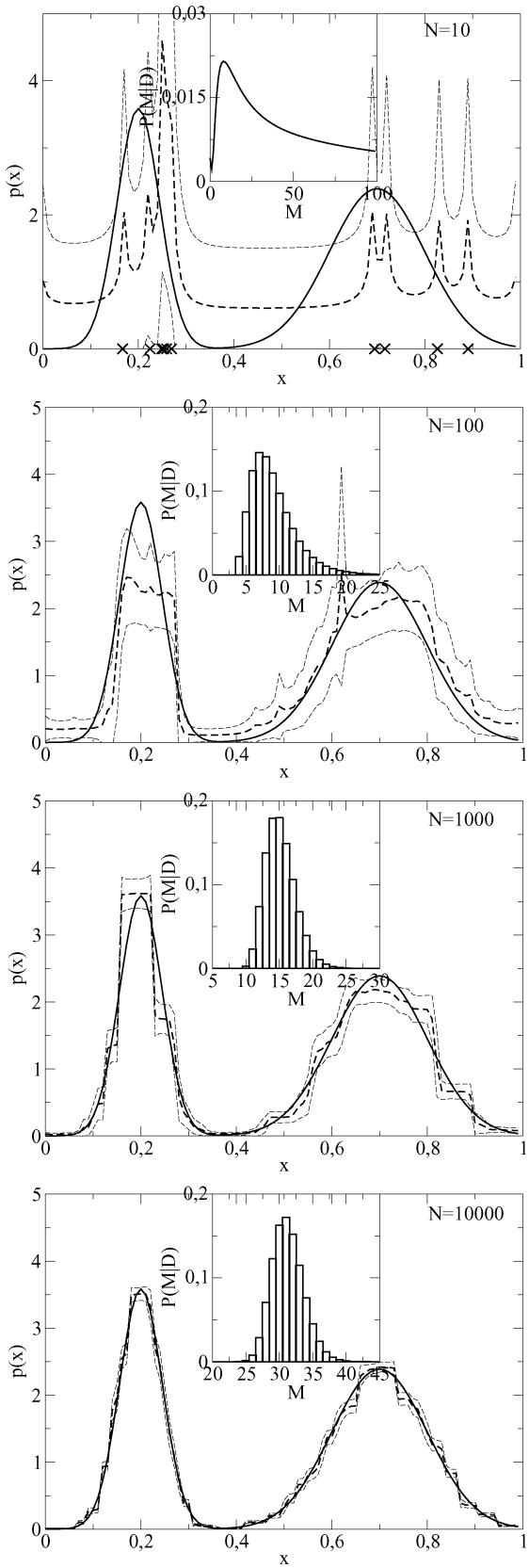

Fig. 4. From top to bottom: densities and posteriors for 10,100,1000 and 10000 data points. Expected probability density (thick dashed line) plus/minus one standard deviation (thin dashed lines), compared to the true density (thick solid line). Insets: posterior distribution of M, the number of intersections. The X's on the abscissa of the graph for 10 data points mark the data points. Note that the maximum of $P(M|D)$ moves upwards with increasing N.

where $N_R = \sum_{m \neq m', m''} n_m$ and R is the part that does not depend upon $n_{m'}$ and $n_{m''}$ and, thus, is not affected by the differentiation

$$R = \frac{\prod_{m \neq m', m''} n_m!}{(N_R + M - 2)!}. \tag{52}$$

We then obtain

$$\frac{\partial^2 B(n_{m'} + 2, n_{m''} + 2) B(n_{m'} + n_{m''} + 4, N_R + M - 1)}{\partial n_{m'} \partial n_{m''}} = \left(h^{N+M+2}_{n_{m'}+2}\right)\left(h^{N+M+2}_{n_{m''}+2}\right) + {}_2h^{N+M+2}_1 - \frac{\pi^2}{6}. \tag{53}$$

The desired expectation can thus be computed in two runs of the iteration: first, let

$$E\left[f(P_{m'})|\{k_m\}, M, D\right] = (n_{m'} + 1) h^{N+M+2}_{n_{m'}+2} \tag{54}$$

$$E\left[f(P_{m''})|\{k_m\}, M, D\right] = (n_{m''} + 1) h^{N+M+2}_{n_{m''}+2} \tag{55}$$

and divide the result by $(N + M + 1)(N + M + 2)$.

Second, let

$$E\left[f(P_{m'})|\{k_m\}, M, D\right] = (n_{m'} + 1) \tag{56}$$

$$E\left[f(P_{m''})|\{k_m\}, M, D\right] = (n_{m''} + 1) \tag{57}$$

and multiply the result with $\frac{{}_2h^{N+M+2}_1 - \frac{\pi^2}{6}}{(N+M+1)(N+M+2)}$.

IX. Examples

Fig. 3 shows examples of the predictive density $p(x)$ and its variance, compared to the density from which the data points were drawn ($K = 100$, $M = 5$, data point abscissas were rounded to the next lower discretization point), as well as the model posteriors $P(M|D)$. The prior $P(M)$ was chosen uniform over the maximum range of M, here $M = 0, \ldots, 99$. Inference was conducted with $\alpha = 0.01$ (see (26)).[1] Note that the curves that represent the expected density plus/minus one standard deviation are not densities any more. The data were produced by first drawing uniform random numbers with the generator sran2() from [12], those were then transformed by the inverse cumulative density (a method also described in [12]) to be distributed according to the desired density. For very small data sets, only the largest structures of the distribution can vaguely be seen (such as the valley between 0.15 and 0.58). Furthermore, the density peaks at each data point (at 0.7, two points were observed very close to each other). One might therefore imagine the process by which the density comes about as similar to kernel-based density estimates. It does, however, differ from a kernel-based procedure insofar as that the number of degrees of freedom does not necessarily grow with the data set, but is determined by the data's structure. The model posterior is very broad, reflecting the large remaining uncertainties due to the small data set. Models with $0 \leq M \leq 97$ were included in the predictions.

With 100 data points, more structures begin to emerge (e.g., the high plateau between 0.58 and 0.68), and the variance decreases, as one would expect. The model posterior exhibits a much narrower maximum, here the included models were those with $4 \leq M \leq 17$.

At 1000 data points, most structures of the original density are modeled, except for the peak at 0.14, and the variance is yet smaller. The algorithm is now fairly certain about the correct number of intersections, $4 \leq M \leq 8$, with a maximum at 4, indicating that the algorithm has not yet discovered the peak at 0.14. Moreover, due to this restriction on the degrees of freedom, the predicted density no longer looks like a kernel-based estimate.

The peak at 0.14 rises out of the background at 10 000 data points, which is also reflected in the fact that the maximum of the model posterior is now at five intersections. All structures of the distribution are now faithfully modeled.

Fig. 4 depicts density estimates from a mixture of two Gaussians. Even though these densities cannot—strictly speaking—be modeled any more by a finite number of bins, the method still produces fairly good estimates of the discretized ($K = 100$) densities. Observe how the maximum of the posterior shifts toward higher values as the number of data points grows. The algorithm thus picks a range of M which, depending on the amount of available information, is best suited for explaining the underlying structure of the data.

For a more quantitative representation of the relationship between N and the "closeness" of expected and true densities, Fig. 5 shows the value of a special instance of the Jensen–Shannon divergence

$$D^2 = \int_0^1 dx \left(p(x) \log \frac{2p(x)}{p(x) + q(x)} + q(x) \log \frac{2q(x)}{p(x) + q(x)} \right). \tag{58}$$

where $p(x)$ is the expected density and $q(x)$ is the discretised true density. D is a bounded metric (see [13]), having a maximum value of $\sqrt{2 \log 2}$ nats. When trying to decide whether a sample is drawn from either $p(x)$ or $q(x)$, D^2 can be interpreted to be the information gain obtained from a sample of length 1. This gain goes to zero, following approximately a power law in N (with exponent ≈ -1 for the bin density and ≈ -0.72 for the mixture of Gaussians). Therefore, given a large enough data set, expected and true density cannot be distinguished any more.

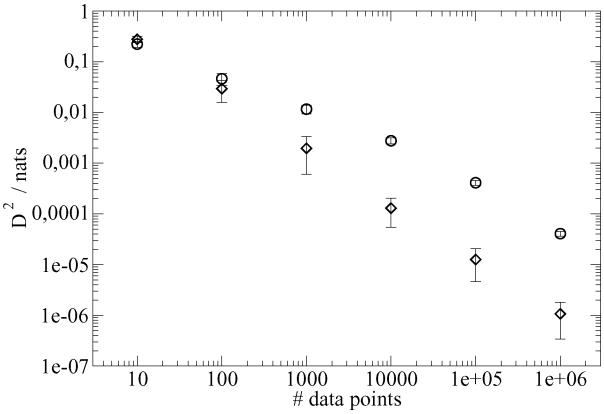

Fig. 5. D^2 between true and expected density as a function of the number of data points. The error bars were obtained from averaging over 100 different data sets drawn from the same density. Diamonds: five-bin density (see Fig. 3), circles: mixture of two Gaussians (see Fig. 4). For details, see text.

[1] On a 2.6-GHz Pentium 4 system running SuSE Linux 8.2, computing the evidence took ≈ 0.26 s (without initializations). The algorithm was implemented in C++ and compiled with the GnU compiler.

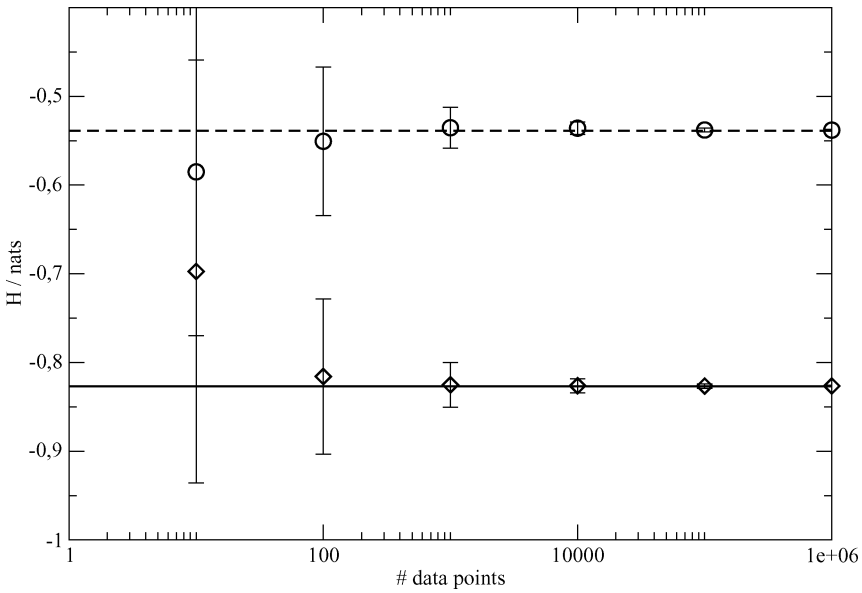

Fig. 6. Expected entropy and expected standard deviation as a function of the number of data points. Both expectations were computed for one data set at a time, then averaged over 100 data sets. The lines represent the true entropies. Diamonds, solid line: five-bin density (see Fig. 3). Circles, dashed line: mixture of two Gaussians (see Fig. 4).

Moreover, in the case of the bin density, the gain is inversely proportional to the number of data points. In other words, the decision as to which density the sample was drawn from becomes twice as hard when N is doubled.

In Fig. 6, the expected entropies are plotted as a function of the number of data points. Error bars represent ± 1 expected standard deviation. Both expectations were computed individually for each data set and then averaged over 100 runs. In every case, the true entropy is well within the error bars. For $N \geq 100$, the expected entropy is fairly close to its true value, thus eliminating the need for finite-size corrections. Note that the standard deviation of the entropy is plotted, *not* the empirical standard deviation of its expectation, i.e., the error bars serve as an indication of the remaining uncertainty in the entropy, which goes to 0 as N increases. This is due to the fact that entropy is treated here as a deterministic quantity, not as a random variable (because samples are drawn from some fixed, albeit possibly unknown, density). To illustrate this point further, look at Fig. 7: For small data sets, the empirical standard deviation of the expectation of the entropy is somewhat smaller than the standard deviation of the entropy. When evaluating experimental results, using the former instead of the latter would thus possibly mislead one to believe in a higher accuracy of the results than can be justified by the data, a danger that is especially preeminent when using methods such as cross validation in place of exact calculations.

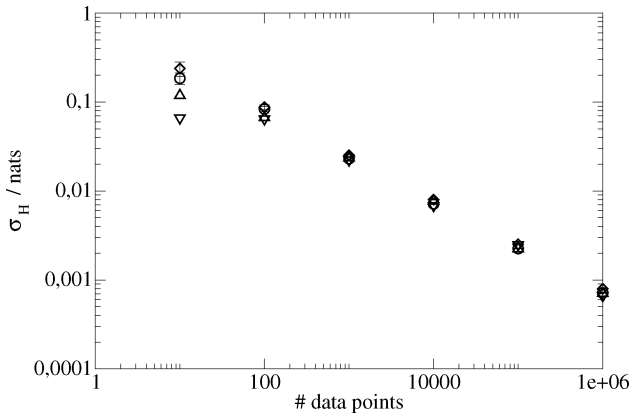

Fig. 7. Expected standard deviation of the entropy (diamonds: five-bin density, circles: mixture of two Gaussians) ± 1 standard deviation and empirical standard deviation (triangle up: five-bin density, triangle down: mixture of two Gaussians) of the expectation of the entropy as functions of the number of data points. Error bars and empirical standard deviations are averages over 100 data sets.

A. The Limit $K \to \infty$

Assume we tried to find the best discretization of the data via (25). If there is no lower limit on Δx (other than 0) given by the problem we are studying (e.g., accuracy limits of the equipment used to record the data), then we might wonder how the discretization posterior behaves as $\Delta x \to 0$ or, conversely, as $K \to \infty$. In this case, given that N stays finite, the data's probability density will diverge whenever a data point is located in an infinitesimally small bin. However, since the number of possible bin configurations also diverges, it is not clear whether $p(D|K)$ stays finite or not. We will not try to give an analytic answer to this question here. Instead, look at Fig. 8, which shows the discretization posterior $P(K|D)$ (normalized on the interval $1 \leq K \leq 1050$, uniform prior over K) for two data sets with $N = 100$ (top) and $N = 1000$ (bottom). The generating density consisted of five bins, with the bin boundaries b_i chosen so that $b_i \times 20$ was an integer. Consequently, the most probable discretization is found to be $\Delta x = \frac{1}{20}$ in both instances. Should the b_i be irrational, or should the generating density not have a bin structure, then we might expect the posterior maximum to move toward smaller Δx (greater K) with increasing N, much like the M-posterior in Fig. 4. For $N = 1000$, the maximum *a posteriori* probability (MAP) choice $K = 20$ can certainly be justified. This is no longer the case for $N = 100$: while $K = 20$ is still the location of the maximum, the probability is now much more

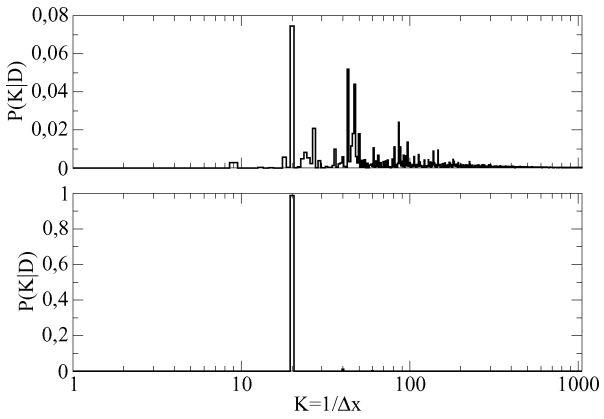

Fig. 8. Discretization posterior $P(K|D)$ for a data set containing 100 (top) and 1000 (bottom) data points. In the latter case, the MAP choice $K = 20$ can certainly be justified, for 100 data points, additional information is necessary before a K can be picked.

evenly spread, and additional prior information is required if a choice is to be made with reasonable certainty.

There is numerical evidence which suggests that $p(D|K)$ actually approaches a (nonzero) constant as $K \to \infty$ (not visible in the graphs, because $p(D|K \to \infty)$ appears to be $\approx 10^{-3}$ ($N = 100$) and $\approx 10^{-11}$ ($N = 1000$) times smaller than the maximum). This implies that the bulk of the probability is concentrated around $\Delta x = 0$. We would, however, argue that since the presented algorithm is not useful in this limit (because it has a computational complexity $\mathcal{O}(K^3)$), one will always have to choose a reasonable upper bound for K, determined by the available computation time, or by the accuracy of the data. Once this bound is set, a K can be determined if enough data are available. Alternatively, a uniform prior over Δx (i.e., $\sim \frac{1}{K^2}$ in K) could also be motivated, which would allow for a cutoff at a moderately large K.

Another quantity of interest is the predictive density $p(x|D, K)$. Its behavior is depicted in Fig. 9, bottom, for $M = 10$ and four different, small data sets. Other values of M give qualitatively similar results. As $K \to \infty$, $p(x|D, K)$ appears to diverge logarithmically with K if evaluated at a data point, and to approach a constant (for given x, D, and M) otherwise. This suggests that the divergence is such that the probability contained in the vicinity of a data point approaches a constant as well. Numerical evidence for this hypothesis is shown in Fig. 9, top. In other words, in the limit $K \to \infty$, the predictive distribution seems to have a δ-peak at each data point. This behavior, which was conjectured by one of the reviewers, is reminiscent of that of a Dirichlet process [14].

X. Extension to Multiple Classes

Assume each data point x_i was labeled, the label being $y_i \in \{1, \ldots, C\}$. In other words, each x_i is drawn from one of C classes. We will now extend the algorithm so as to infer the joint distribution (or density) $P((x, y)|M, D)$, where $D = \{(x_1, y_1), \ldots, (x_n, y_n)\}$.

Instead of $M + 1$ probabilities, we now have $(M+1)C$. P_m^y is the probability mass between k_m and k_{m-1} in class y, i.e., we assume that the bins are located at the same places across

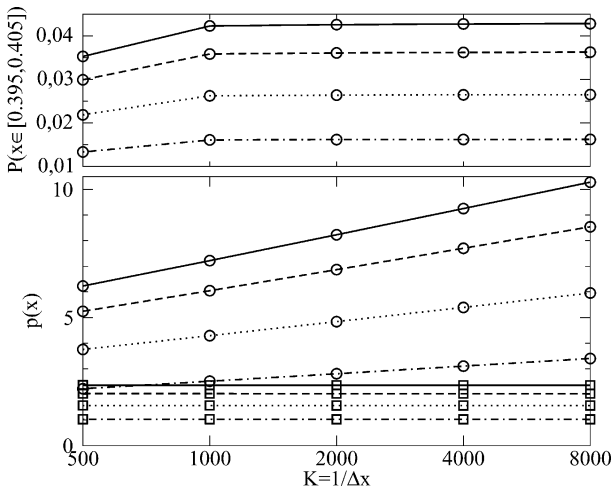

Fig. 9. Bottom: Predictive densities $p(x|D, M = 10)$ as a function of $K = \frac{1}{\Delta x}$. Circles: $p(x|D, M = 10)$ at the location (0.4) of a data point, squares: $p(x|D, M = 10)$ between data points. The lines denote data sets with $N = 1$ (solid), 2 (dashed), 4 (dotted), and 8 (dash-dotted). The predictive density appears to diverge logarithmically with K at the location of a data point, and remains finite elsewhere. Top: Predictive probability $P(x \in [0.395, 0.405]|D, M = 10)$ in the vicinity of a data point as a function of K. This probability seems to asymptote at a constant value for $K \to \infty$, which indicates that the predictive density has a δ-peak at each data point.

classes. This might seem like a rather arbitrary restriction. We shall, however, impose it for two reasons.

1) The algorithm for iterating over the bin configurations keeps a simple form, similar to (17) and (18).
2) If we allowed different sets of k_m for the classes, computing marginal distributions and entropies would become exceedingly difficult. While possible in principle, it would involve confluent hypergeometric functions and a significantly increased computational cost.

Let n_m^y be the number of data points in class y and bin m, and $\tilde{n}_m = \sum_{y=1}^{C} n_m^y$. The likelihood (2) now becomes

$$P(D|M, \{P_m^y, k_m\}) = \prod_{m=0}^{M} \frac{\prod_{y=1}^{C} (P_m^y)^{n_m^y}}{\Delta k_m^{\tilde{n}_m}}. \quad (59)$$

Thus, following the same reasoning as above, the iteration rules now are

$$a(0, \tilde{K}) = \frac{\prod_{y=1}^{C} n_0^y!}{(\tilde{K} + 1)^{\tilde{n}_0}} \quad (60)$$

where n_0^y is the total number of data points in class y for which $k \le \tilde{K}$ and $\tilde{n}_0 = \sum_{y=1}^{C} n_0^y$. Furthermore

$$a(\tilde{M} + 1, \tilde{K}) = \sum_{\tilde{k}=\tilde{M}}^{\tilde{K}-1} a(\tilde{M}, \tilde{k}) \frac{\prod_{y=1}^{C} n_{\tilde{M}+1}^y!}{(\tilde{K} - \tilde{k})^{\tilde{n}_{\tilde{M}+1}}} \quad (61)$$

where $n_{\tilde{M}+1}^y$ is the total number of data points in class y for which $\tilde{k} < k \le \tilde{K}$, and $\tilde{n}_{\tilde{M}+1} = \sum_{y=1}^{C} n_{\tilde{M}+1}^y$.

We can now evaluate the expected joint distribution and its variance at any (k, y), using (23) and (24) as before. To compute the marginal distribution, note that

$$P(k|M, D) = \sum_{y=1}^{C} P((k, y)|M, D) \quad (62)$$

and likewise for its square.

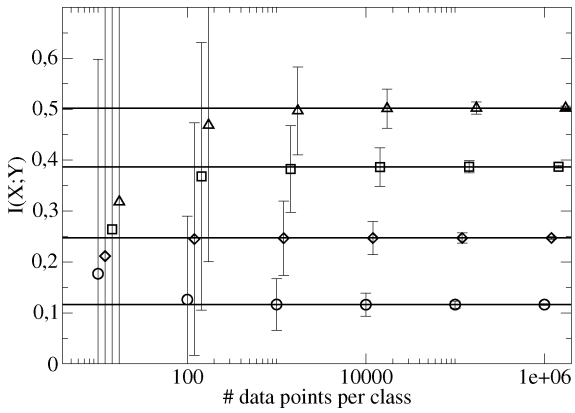

Fig. 10. Expected mutual informations and upper bounds on the standard deviations, computed via (65) for different data set sizes. The lines indicates the true mutual information. Circles: $I(X;Y) = 0.1173$ nat, diamonds: $I(X;Y) = 0.2472$ nat, squares: $I(X;Y) = 0.3870$ nat, triangles: $I(X;Y) = 0.502$ nat. Mutual informations were evaluated at 10, 100, 1000, 10 000, 100 000 and 1 000 000 data points; symbols in the plot are shifted to the right to disentangle the error bars.

XI. Computing the Mutual Information and an Upper Bound on Its Variance

The mutual information between class label and x is given by

$$I(X;Y) = H(X,Y) - H(X) - H(Y) \tag{63}$$

which has to be averaged over the posterior distribution of the model parameters $p(\{P_m^y, k_m\}|M, D)$. This can be accomplished term-by-term, as described above, yielding the exact expectation of the mutual information under the posterior. The evaluation of its variance is somewhat more difficult, due to the mixed terms $\langle H(X)H(Y)\rangle_M$, $\langle H(X)H(X,Y)\rangle_M$, and $\langle H(Y)H(X,Y)\rangle_M$. For the time being, we shall thus be content with an upper bound. Using the identity

$$\mathrm{Var}(\sum_{i=1}^{N} X_i) \leq N \sum_{i=1}^{N} \mathrm{Var}(X_i) \tag{64}$$

we obtain

$$\mathrm{Var}(I(X;Y)) \leq (\mathrm{Var}(H(X)) + \mathrm{Var}(H(Y)) + \mathrm{Var}(H(X,Y))). \tag{65}$$

All terms on the RHS can be computed as above.

Fig. 10 shows the results of some test runs on artificial data. Points were drawn from two classes with equal probability. Within each class, a three-bin distribution was used to generate the data. The probabilities in the bins were varied to create four different values of the mutual information. Before inferring the mutual information from the data, we first determined the best discretization step size (via the maximum of (25); this approach seems to yield good results for the mutual information, even when the discretization posterior probability is not very concentrated at one Δx). The depicted values are individual averages over 100 data sets. In all cases, its true value lies well within the error bars. However, especially for small sample sizes, they seem rather too large—an indication that the upper bound given by (65) needs future refinement. However, observe that the expectation of $I(X;Y)$ is close to its true value from 100 data points per class onwards. This is an indication that the algorithm performs well for moderate sample sizes, without the need for finite-size corrections.

XII. Sparse Priors

The uniform prior (8) is a reasonable choice in circumstances where no information is available about the data *a priori* other than their range. Should more be known, it is sensible to incorporate this knowledge into the algorithm so as to speed up the inference process (i.e., allow for better predictions from small data sets). In the following, we will examine symmetric Dirichlet priors of the form

$$p(\{P_m\}|M) \propto \prod_{m=0}^{M} P_m^{\theta-1} \tag{66}$$

where $\theta > 0$ to avoid divergence of the normalization constant. Fig. 11, left, shows the resulting priors for $M = 1$ (two bins, thus, P_0 is beta-distributed) and two different values of θ. The symmetry arises from the condition $P_0 + P_1 = 1$. The curve for $\theta = 0.5$ (solid line) exhibits the typical behavior expected for $\theta < 1$: Extreme values are favored because $p(P_0) \to \infty$ as P_0 approaches 0 or 1. Thus, this range of θ will generally favor distributions where few bins contain most of the probability mass, i.e., *sparse* distributions. Conversely, the curve for $\theta = 1.5$ (dashed line) indicates the general behavior expected for $\theta > 1$: the prior now promotes moderate values of P. Therefore, the probability mass will tend to be more evenly distributed (dense distributions). For $\theta = 1$, the uniform prior is recovered.

In typical single-cell neurophysiological experiments, sparse distributions are frequently encountered. Consider the following setup: a mammal is presented with a visual stimulus and the firing events produced by one of its visual cortex neurons are recorded in some suitably chosen time window. Assume that the temporal resolution of the recording equipment was 1 ms and the window width 100 ms. A simple model for the firing behavior is the Poisson process, i.e., a constant probability P_{fire} of observing an event at any given time within the window. The expected distribution of the number of observed events is then governed by a binominal distribution. Fig. 11, right, depicts three such distributions for small to medium values of P_{fire}. While up to 100 observed events in the window are possible in principle, those values are extremely unlikely. Hence, a sparseness promoting prior can be expected to speed up convergence when such (or similar) distributions are to be inferred from data.

The algorithm can be generalized to include $\theta \neq 1$. Setting $\theta_M = (M+1)\theta$, (10) now becomes

$$p(\{P_m\}|M) = \Gamma(\theta_M)\,. \tag{67}$$

Likewise, (16) is to be replaced by

$$P(D|\{k_m\}, M) = \frac{\Gamma(\theta_M)}{\Gamma(N+\theta_M)} \prod_{m=0}^{M} \frac{\Gamma(n_m+\theta)}{\Delta k_m^{n_m}}\,. \tag{68}$$

This expression is again a product of the counts observed in different bins and the bin width, times a factor which only depends on the total number of bins and data points. Hence, the same sum–product decomposition scheme as before can be used, with (17) and (18) now being

$$a(0, \tilde{K}) = \frac{\Gamma(n_0+\theta)}{(\tilde{K}+1)^{n_0}} \tag{69}$$

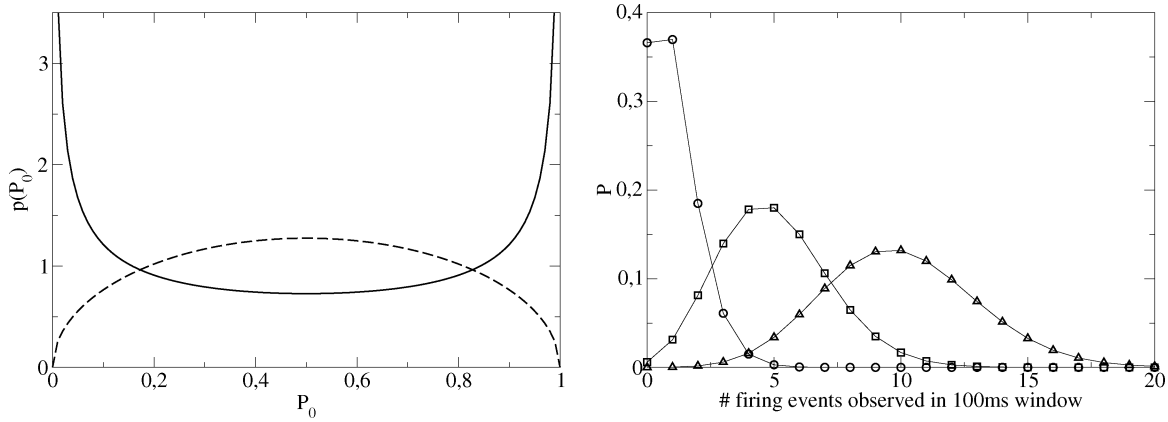

Fig. 11. Left: Prior for $M = 1$ and two choices of θ: $\theta = 0.5$ (solid line) and $\theta = 1.5$ (dashed line). The first prior favors extreme values of P_0, i.e., sparse distributions, the second one promotes moderate values of P_0 (dense distributions). Right: Binominal distributions are simple models for the distributions of spike counts observed in a fixed-time window. The curves were computed for $P_{\text{fire}} = 0.01$ (circles), $P_{\text{fire}} = 0.05$ (squares), and $P_{\text{fire}} = 0.1$ (triangles). Values are plotted only up to a spike count of 20, even though a maximum of 100 would have been possible. However, the probabilities for counts greater than 20 are virtually zero. Thus, most possible values are virtually never assumed, i.e., the distributions are sparse.

$$a(\tilde{M}+1,\tilde{K}) = \sum_{\tilde{k}=\tilde{M}}^{\tilde{K}-1} a(\tilde{M},\tilde{k}) \frac{\Gamma(n_{\tilde{M}+1}+\theta)}{(\tilde{K}-\tilde{k})^{n_{\tilde{M}+1}}} \,. \tag{70}$$

This allows for the computation of evidences and expected probabilities and their variances. To compute entropies and mutual informations and their variances, the relevant averages have to be adapted as well: after some tedious but straightforward calculus, we find that (37) has to be substituted by

$$E\left[P_{m'}\log(P_{m'})|\{k_m\},M,D\right] = \frac{n_{m'}+\theta}{N+\theta_M} \times \left[h_0^N(\theta_M) + \Psi\left(\theta_M\right) - h_0^{n_{m'}}(\theta) - \Psi(\theta)\right] \tag{71}$$

where $\Psi()$ is the digamma function and

$$h_a^b(\theta) = \sum_{i=a}^{b} \frac{1}{i+\theta} \tag{72}$$

$$_2h_a^b(\theta) = \sum_{i=a}^{b} \frac{1}{(i+\theta)^2}. \tag{73}$$

While the term in the square brackets could be written as the difference between two digamma functions, decomposing it into a part that depends both on the prior and on the data and a part that depends only on the prior (i.e., M and θ) allows for a precomputation of the latter. Equation (39) becomes

$$E\left[P_{m'}\log(\Delta k_{m'})|\{k_m\},M,D\right] = \frac{n_{m'}+\theta}{N+\theta_M}\log(\Delta k_{m'}). \tag{74}$$

Likewise, for (48)

$$\begin{aligned} &E\left[P_{m'}^2\log^2(P_{m'})|\{k_m\},M,D\right] \\ &= \frac{(n_{m'}+\theta)(n_{m'}+1+\theta)}{(N+\theta_M)(N+1+\theta_M)} \\ &\quad\times\left[\left(h_0^{N+1}(\theta_M) - h_0^{n_{m'}+1}(\theta) + \Psi(\theta_M) - \Psi(\theta)\right)^2\right. \\ &\quad\left.+{}_2h_0^{N+1}(\theta_M) - {}_2h_0^{n_{m'}+1}(\theta) + \Psi'(\theta) - \Psi'(\theta_M)\right] \end{aligned} \tag{75}$$

and finally, we obtain for (54)

$$\begin{aligned} &E[f(P_{m'})|\{k_m\},M,D] \\ &\quad=(n_{m'}+\theta)\left[h_0^{N+1}(\theta_M)+\Psi(\theta_M)-h_0^{n_{m'}}(\theta)-\Psi(\theta)\right] \end{aligned} \tag{76}$$

$$\begin{aligned} &E[f(P_{m''})|\{k_m\},M,D] \\ &\quad=(n_{m''}+\theta)\left[h_0^{N+1}(\theta_M)+\Psi(\theta_M)-h_0^{n_{m''}}(\theta)-\Psi(\theta)\right] \end{aligned} \tag{77}$$

then divide the results of the averaging by $(N+\theta_M)(N+1+\theta_M)$, and for (56)

$$E\left[f(P_{m'})|\{k_m\},M,D\right] = (n_{m'}+\theta) \tag{78}$$

$$E\left[f(P_{m''})|\{k_m\},M,D\right] = (n_{m''}+\theta) \tag{79}$$

and multiply the averages by $\frac{_2h_0^{N+1}(\theta_M)-\Psi'(\theta_M)}{(N+\theta_M)(N+1+\theta_M)}$.

The question which remains is how to choose θ. Since the sparseness of a distribution will usually not be known *a priori*, a feasible way is to treat it as another hyperparameter to be inferred from the data. However, integrating over θ is computationally rather expensive. Instead, we tried a maximum *a posteriori* approach (the posterior density of θ was unimodal in all observed cases), which seems to work quite well.

To compare the performance of the modified algorithm on sparse distributions with a method proposed in [15], we chose a way similar to that used in that paper: for a set of eight "stimuli," distributions of responses, i.e., "mean firing rates," were generated by drawing random variables from binominal distributions with different values of P_{fire}. These firing probabilities were functions of the stimulus label X. The expected mutual information between X and the responses Y were subsequently computed.

First, the optimal θ was determined by the maximum of its posterior distribution in the interval $[0.0001, 1]$. Using this θ, we then searched the best M in the same fashion. Given those two parameters, the mutual information was computed. The results were averaged over 100 data sets (calculating θ and M anew for each data set), which also allowed for the estimation of error bars. A "data set" is here a collection of (stimulus label,

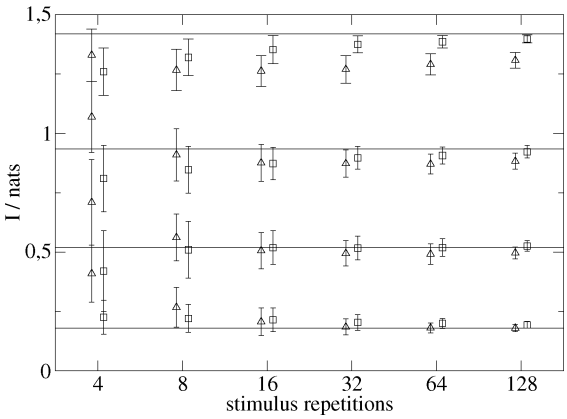

Fig. 12. Results of the modified algorithm on artificially created "neural responses" to eight "stimuli" (squares), compared with the method described in [15] (triangles). Mutual informations were evaluated at 4, 8, 16, 32, 64, and 128 stimulus repetitions for both methods, the symbols in the plot are shifted slightly to the left and to the right to disentangle the error bars. The response distributions were binominals with different P_{fire}'s. Error bars are ± 1 standard deviations, obtained from averaging over 100 data sets. The lines represent the true mutual informations.

response) pairs with a fixed number of stimulus repetitions (between 4 and 128). The error bars are the estimated standard deviations of the mutual information, not the standard errors. Fig. 12 shows the results. The squares are the expectations computed by our algorithm. The triangles depict the mutual information estimates computed from the observed frequencies corrected by the first-order bias term reported in [15]. Those frequencies, as described in [15], were obtained by a simple discretization of the response space. Even for a rather small number of stimulus repetitions, the algorithm performs quite well, getting better as the size of the data set increases. In the great majority of cases, it appears to be more accurate than the finite-size corrected estimates, even though the latter yield reasonable results as well if the number of stimulus repetitions is greater than the number of stimuli. Moreover, the small error bars indicate that reliable results can be expected with our method even if only a single data set is available.

XIII. Conclusion

The presented algorithm computes exact evidences and relevant expectations of probability distributions/densities with polynomial computational effort. This is a significant improvement over the naïve approach, which requires an exponential growth of the number of operations with the degrees of freedom. It is also able to find the optimal number of degrees of freedom necessary to explain the data, without the danger of overfitting. Furthermore, the expectations of entropies and mutual information have been shown to be close to their true values for relatively small sample sizes. In the past, a variety of methods for dealing with systematic errors due to small sample sizes have been proposed, such as stimulus shuffling in [16] or regularization of neural responses by convolving them with Gaussian kernels. While these approaches work to some degree, they lack the sound theoretical foundation of exact Bayesian treatments.

In [15], finite size corrections are given based on the number of effective bins, i.e., the number of bins necessary to explain the data. There, it is also demonstrated that this leads to information estimates which converge much more rapidly to the true value than the other techniques mentioned (shuffling and convolving). However, as authors of [15] themselves admit, their method of choosing the number of effective bins is only "Bayesian-like." Furthermore, their initial regularization applied to the data—choosing a number of bins that is equal to the number of stimuli and then setting the bin boundaries so that all bins contain the same number of data points, a procedure also used by [17]–is debatable (it should, however, be pointed out that this equiprobable binning procedure is not an essential ingredient for the successful application of the finite-size corrections of [15]. It has recently been demonstrated [18] that, given a decent amount of data is available, the methods of [15] can be used to yield reasonably unbiased estimates for $M = K - 1$). On the one hand, one might argue that the posterior of M is still broad when the data set is small (see Fig. 3, graphs for 10 and 100 data points), so choosing the wrong number of bins will do little damage. On the other hand, the bin boundaries must certainly not be chosen in such a way that all bins contain the same number of points. Doing so will destroy the structure present in the data. Consider e.g., Fig. 3, graph for 1000 data points: there are many more data points in the interval $[0.58, 0.68]$ than there are between $[0.15, 0.58]$, which reflects a feature of the distribution from which the data were drawn and should thus be modeled by any good density estimation technique. This will, however, not be the case if the boundaries are chosen as proposed: there would be a boundary somewhere at ≈ 0.63 instead of 0.58, and the step at this point would be replaced by a considerably smaller one at ≈ 0.63, thus misrepresenting the underlying distribution. In other words, that procedure would not even converge to the correct distribution as the data set size grows larger, unless the number of bins was allowed to grow with the data set. But that might introduce unnecessary degrees of freedom. Consequently, mutual information estimates calculated from those estimated distributions must be interpreted with great care.

We believe to have shown that those drawbacks can be overcome by a Bayesian treatment, which also seems to show improved performance over finite-size corrections. Thus, the algorithm should be useful in several areas of research where large data sets are hard to come by, such as neuroscience.

Another interesting Bayesian approach to removing finite-size effects in entropy estimation is the Nemenman–Shafee–Bialek (NSB) method [19]. It exploits the fact that the typical distributions under the symmetric Dirichlet prior (66) have very similar entropies, with a variance that vanishes as K grows large. This observation is then employed to construct a prior which is (almost) uniform in the entropy. The resulting estimator is demonstrated to work very well even for relatively small data sets.

In contrast to the NSB method, our approach deals with finite-size effects by determining the model complexity (i.e., the posterior of M). It might be interesting to combine the two: since the NSB prior depends only on θ and K, the required numerical integration [19, eqn. (9)] could be carried out, with [19, eqn. (10)] replaced by $P(D|\theta)$ (i.e the denominator of (21) for a given θ).

It was proven in [20] that uniformly (over all possible distributions) consistent entropy estimators can be constructed for distributions comprised of any number of bins M, even if $M \gg N$. The above presented results (Fig. 6) suggest that the expected entropies computed with our algorithm are asymptotically unbiased and consistent. Furthermore, the true entropy was usually found within the expected standard deviation. It remains to be determined how the algorithm performs if $M \gg N$.

Since the upper bound (65) on the variance of the mutual information is rather large for small sample sizes, it might be interesting to invest some more work into computing the exact variance of the mutual information. This, however, turns out to be difficult.

ACKNOWLEDGMENT

The authors would like to thank Johannes Schindelin for many stimulating discussions. Furthermore, they are also grateful to the anonymous reviewers for their helpful suggestions and references.

REFERENCES

[1] M. Rosenblatt, "Remarks on some nonparametric estimates of a density function," *Ann. Math. Statist.*, vol. 27, pp. 832–837, 1956.

[2] M. Stone, "Cross-validatory choice and assessment of statistical predictions," *J. Roy. Statist. Soc., Ser. B*, vol. 36, no. 1, pp. 111–147, 1974.

[3] N. Ueda, R. Nakano, Z. Ghahramani, and G. E. Hinton, "SMEM algorithm for mixture models," *Neur. Comput.*, vol. 12, no. 9, pp. 2109–2128, 2000.

[4] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood for incomplete data via the EM algorithm," *J. Roy. Statist. Soc., Ser, B*, vol. 39, no. 1, pp. 1–38, 1977.

[5] R. Cox, "Probability, frequency and reasonable expectation," *Amer. J. Phys.*, vol. 14, no. 1, pp. 1–13, 1946.

[6] M. Jordan, Z. Ghahramani, T. Jaakkola, and L. Saul, "An introduction to variational methods for graphical models," *Mach. Learn.*, vol. 37, no. 2, pp. 183–233, 1999.

[7] C. Wallace and D. Boulton, "An information measure for classification," *Comput. J.*, vol. 11, pp. 185–194, 1968.

[8] J. Rissanen, "Modeling by shortest data description," *Automatica*, vol. 14, pp. 465–471, 1978.

[9] M. Hutter, "Distribution of mutual information," in *Advances in Neural Information Processing Systems*. Cambridge, MA: MIT Press, 2002, vol. 14, pp. 339–406.

[10] F. Kschischang, B. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 498–519, Feb. 2001.

[11] P. Davis, "Gamma function and related functions," in *Handbook of Mathematical Functions*, M. Abramowitz and I. Stegun, Eds. New York: Dover, 1972.

[12] W. Press, B. Flannery, S. Teukolsky, and W. Vetterling, *Numerical Recipes in C: The Art of Scientific Computing*. New York: Cambridge Univ. Press, 1986.

[13] D. Endres and J. Schindelin, "A new metric for probability distributions," *IEEE Trans. Inf. Theory*, vol. 49, no. 7, pp. 1858–1860, Jul. 2003.

[14] T. S. Ferguson, "Prior distributions on spaces of probability measures," *Ann. Statist.*, vol. 2, no. 4, pp. 615–629, 1974.

[15] S. Panzeri and A. Treves, "Analytical estimates of limited sampling biases in different information measures," *Network: Comp. Neur. Syst.*, vol. 7, pp. 87–107, 1996.

[16] L. Optican, T. Gawne, B. Richmond, and P. Joseph, "Unbiased measures of transmitted information and channel capacity from multivariate neuronal data," *Biol. Cybern.*, vol. 65, pp. 305–310, 1991.

[17] E. Rolls, H. Critchley, and A. Treves, "The representation of olfactory information in the primate orbitofrontal cortex," *J. Neurophys.*, vol. 75, pp. 1982–1996, 1995.

[18] E. Arabzadeh, S. Panzeri, and M. Diamond, "Whisker vibration information carried by rat barrel cortex neurons," *J. Neurosci.*, vol. 24, no. 26, pp. 6011–6020, 2004.

[19] I. Nemenman, W. Bialek, and R. van Steveninck, "Entropy and information in neural spike trains: Progress on the sampling problem," *Phys. Rev. E*, vol. 69, no. 5, 2004.

[20] L. Paninski, "Estimating entropy on m bins given fewer than m samples," *IEEE Trans. Inf. Theory*, vol. 50, no. 9, pp. 2200–2203, Sep. 2004.