SEPTEMBER 2025
VOL. 32 NO. 3

INTERNATIONAL SOCIETY FOR BAYESIAN ANALYSIS

THE ISBA BULLETIN

OFFICIAL BULLETIN OF THE INTERNATIONAL SOCIETY FOR BAYESIAN ANALYSIS

SOFTWARE HIGHLIGHT

Luca Porta Mana
Western Norway University of Applied Sciences
pgl@portamana.org

INFERNO: INFERENCE IN R WITH BAYESIAN NONPARAMETRICS

Population inference and Bayesian nonparametrics

A very important kind of inference in research fields such as medicine is *population inference*, also called "density inference" or "density regression". Its general goal is to infer the frequency distribution of some variates in a population. This is different, for instance, from *functional regression*, where the goal is to infer the functional relationship – assumed to exist – between a set of predictor variates and a target or "predictand" variate. In population inference the existence of a functional relation cannot be assumed. In fact there may not even be a clear distinction between predictor and predictand variates. A typical goal is the inference of frequency distributions within particular subpopulations or sub-groups; thus all sorts of conditional probabilities are required. A clinician may be interested in the statistics and probability of a medical condition given a symptom, but also in that of a symptom given a medical condition; and maybe only within subjects of a given sex or age. De Finetti's theorem (see e.g. Bernardo and Smith, 2000, §§ 4.2, 4.3, 4.6) lies at the heart of population-inference methods; a particularly brilliant discussion is given by Lindley & Novick (1981).

Sadly many researchers still approach population-inference problems by means of p-values or other frequentist practices, which only give limited, coarse, and not seldom misleading results about a population's frequency distribution. Some researchers adopt Bayesian methods but limit themselves to *parametric* ones, which make very restrictive and possibly unrealistic assumptions about the population's distribution; as opposed to *nonparametric* ones, which do not.

Until a couple decades ago the use of parametric methods, and maybe even of frequentist practices, had pragmatic reasons. Better methods were computationally too costly or unfeasible. Population-inference problems were low-dimensional; one could *visually* check whether the assumptions were appropriate and the results reasonable. But today these reasons cannot earnestly be given (Walker, 2010). Bayesian nonparametric methods have become computationally feasible for many kinds of inference. Many inference problems today involve from tens to thousands of variates; it is impossible to visually check in such high-dimensional spaces whether frequentist practices or parametric assumptions are acceptable, or by how much they err. Results may be affected by large parametric-modelling errors (Draper, 1995).

But there is still one reason today for why Bayesian nonparametric methods are avoided: *lack of user-friendly software*. Many clinical researchers might like to try a Bayesian nonparametric analysis, but cannot: they would need to study Markov-chain Monte Carlo (MCMC) techniques, programming languages to implement the latter, and a read about plethora of debated practices to "assess convergence". Most clinicians do not have time to learn all this even if they wanted to. Also, available packages for Bayesian nonparametrics are not quite suited to population inference. Some of them focus on functional regression, which as discussed above is not an appropriate assumption. Some make a priori distinctions between predictor and predictand variates, limiting the range of useful inferences. Most still require MCMC programming expertise.

The R-package **inferno**[1] was built to try to remedy the lack of user-friendly software of this kind.

[1] https://pglpm.github.io/inferno/

Use and features

Using the package is simple. The user first provides a data sample S of variates from a population, for instance age, sex, symptom, disease, and kind of treatment of a number of patients that satisfy specific criteria. The package can work with any combination of continuous, discrete ordinal, nominal, and binary variates. Continuous variates can be defined in bounded intervals, and can also have boundary values with finite probability mass, as it may happen with censoring. The package cannot handle periodic variates yet, or variates with complex topology, such as images. The data sample and the variate characteristics (type, domain, possible censoring) are provided by the user as two CSV files to an R function called `learn()`.

The package then runs a MCMC computation by means of the **Nimble** package[2], using parallel CPUs if available, to find the probability distribution over all possible joint frequency distributions of the variates. The result is saved in an R object called `learnt`. The crucial point here is that this computation is automatic and does not require any further control from the user, who is simply informed at regular intervals about the expected end time of the computation. (Optional arguments still allow users expert in MCMC to control many parameters such as number of chains, target effective sample size, thinning and burn-in, and even some hyperprior parameters.)

Once the computation has finished, the user can inquire multiple times about any of the following:

- For a new unit of the population, say a new patient, the conditional probability (density) for the values of *any* set of variates $A{=}a, B{=}b, \dots$ given *any* other set $C{=}c, D{=}d, \dots$:

$$\mathrm{P}(A{=}a, B{=}b, \dots \mid C{=}c, D{=}d, \dots, S) \tag{1}$$

  Such a probability could for instance be used in medical decision making (Sox et al., 2024; Hunink et al., 2014). The conditional can be empty; tail probabilities (e.g. $A \geq a$, $C \leq c$) can also be requested.

- The probability distribution for the conditional *frequencies*, in the whole unsampled population, of the values of any set of variates given any other set (possibly empty). If we denote the frequency distribution by F and a specific value by f, this probability can be written

$$\mathrm{P}\big[F(A{=}a, \dots \mid C{=}c, \dots) = f \;\big|\; S\big] \,\mathrm{d}f \,. \tag{2}$$

  This probability distribution is represented by the MCMC samples f_i drawn from it by `learn()`.

- The mutual information (MacKay, 2005, Ch. 8) between any two sets of quantities.

Probabilities (1) and (2) are connected by a variant of de Finetti's theorem:

$$\mathrm{P}(A{=}a, \dots \mid C{=}c, \dots, S) = \int f \,\mathrm{P}\big[F(A{=}a, \dots \mid C{=}c, \dots) = f \;\big|\; S\big] \,\mathrm{d}f \,. \tag{3}$$

In a manner of speaking, the probability distribution (2) expresses how much the probability value (1) could change, if it were updated by sampling the whole population. It expresses the uncertainty in the statistical results owing to finite sample size.

The package function `Pr(Y, X, learnt)` does the first two kinds of calculations. The user provides a list `Y` of predictand variates and values of interest; an optional, analogous list `X` of predictors; and the `learnt` object produced by `learn()`. The calculation of mutual information is done by the package function `mutualinfo(Y1names, Y2names, X)`, where the first two arguments are the variate sets of interest, and the optional third argument is a set of variate values to conditionalize upon.

The package allows the user to visualize the probabilities above when just one predictand and one predictor variates are involved. If the results of the `Pr()` function are saved in some object, say `probs`, then the visualization is done by simply calling `plot(probs)`.

[2] https://r-nimble.org/

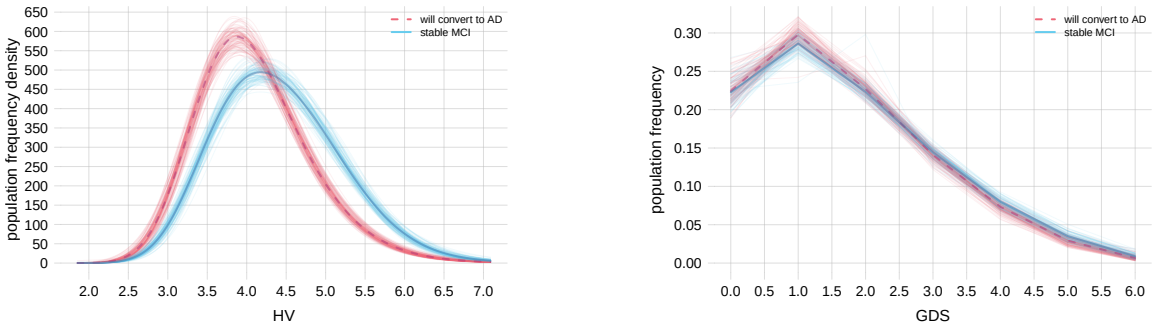

Figure 1

Figure 1 shows two examples from a study (Porta Mana et al., 2023) about conversion from Mild Cognitive Impairment (MCI) to Alzheimer's Disease (AD); they can be used to further illustrate the probability distributions (1) and (2).

In the plot on the left, the predictand variate is Hippocampal Volume (HV); the predictor is the yes/no variate 'will convert to AD' (cAD). The thick solid blue line and thick dashed red line are the conditional probability distributions (omitting the sample-data dependence)

$$\mathrm{P}(HV \mid cAD{=}\mathrm{N}) \,, \qquad \mathrm{P}(HV \mid cAD{=}\mathrm{Y}) \,.$$

The cloud of thinner blue lines surrounding the first distribution above represents the probability distribution

$$\mathrm{P}\bigl[F(HV \mid cAD{=}\mathrm{N})\bigr]$$

of possible frequency distributions; each thin line is a sample from this distribution.

Looking at this plot, a clinician can immediately see that the frequency distributions of hippocampal volume in the sub-populations of patients that will convert to Alzheimer's, and in those who will not, are different. Such a difference is almost certain even accounting for the uncertainty from the finite sample size. The plot on the right is analogous but for the predictand variate 'Geriatric Depression Scale' (GDS) in stead of hippocampal volume. In this case the clinician sees that the frequency distributions for the two sub-populations cannot be distinguished within the finite-sample uncertainty.

It is exactly this kind of differences and uncertainties that clinical researchers often try to clumsily capture through p-values. In private communications, several clinical researchers expressed elation at the capacity to visualize the estimates of different sub-population statistics, and even more the uncertainties they carry because of finite sample size.

The quick analysis above is mostly qualitative, but concrete numbers, such as quantiles and credibility intervals, expected values, and so on, can be easily produced. This becomes necessary when many variates are considered jointly and visualization is impossible. In such high-dimensional cases the package allows the user to compute the credibility intervals for any kind of distance between two frequency distributions (e.g. Hellinger or Kantorovich or Shannon-Jansen distance, or relative entropy). The computation of the mutual information between any two sets of variates, illustrated in a vignette[3], gives moreover a measure of their association that does not depend on assumptions such as linearity or gaussianity.

The ability to swap the "predictor" and "predictand" roles of any variates is illustrated in the plots of fig. 2, parallel to those of fig. 1. In the left plot, the thicker lines show the probabilities

$$\mathrm{P}(cAD{=}\mathrm{Y} \mid HV)$$

[3] https://pglpm.github.io/inferno/articles/mutualinfo.html

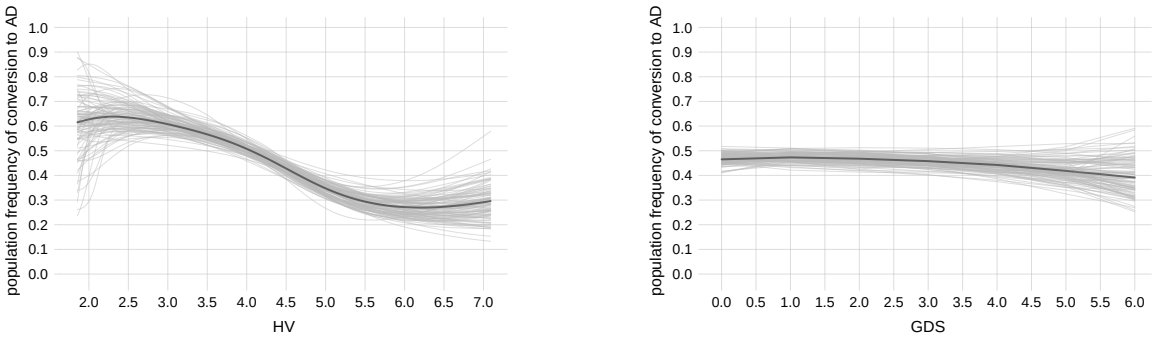

Figure 2

for various values of the HV variate, which is now a predictor. We see for example that among individuals having hippocampal volume around 5.0, between 30% and 40% will convert to Alzheimer's. Among those with volume around 2.0, we can only say that between roughly 45% and 80% will convert; in this case the finite-sample size (few samples with this HV value) leads to a much larger uncertainty of the frequency estimates.

The possibility of swapping predictor and predictand roles makes it also possible to implement, on the fly, corrections for base rates by means of Bayes's theorem (Lindley and Novick, 1981, § 4).

Plots and calculations like the ones above are of course nothing new in Bayesian nonparametrics. The point here is that the user can produce them just by inputting the population sample in the package, waiting for the MCMC computation to finish, and then simply asking about variates of interest and plotting the results. The sequence of commands could be as simple as

```
learnt <- learn(data = 'data.csv', metadata = 'metadata.csv')
## Information about MCMC and expected end time...

probs <- Pr(Y = list(A = seq(0, 8, 0.1)), X = list(C = c('yes', 'no')), learnt)

plot(probs)
```

More extensive use examples are given in the package's main vignette[4]. The package has also been used to "calibrate" the output of machine-learning algorithms (Dyrland et al., 2022), improving their performance. Readers of the ISBA Bulletin are probably interested in the internals of the package, discussed next.

Internals

In nonparametric population inference the prior and posterior probability distributions are over an in-principle infinite-dimensional manifold of frequency distributions. The mathematical representation of this manifold is therefore crucial. The **inferno** package uses the ingenious representation of a distribution as a mixture of product kernels discussed by Dunson & Bhattacharya 2011. For example, a generic frequency distribution for variates A and B is written as

$$F(A, B) = \sum_i w_i \, K_A(A \mid \alpha_i) \, K_B(B \mid \beta_i) \tag{4}$$

[4] https://pglpm.github.io/inferno/articles/inferno_start.html

where w_i are normalized weights, K_A a distribution for variate A depending on parameters α_i, and similarly for K_B. The product is easily generalized to any number of variates. In principle the sum should be countably infinite, but as discussed in Ishwaran & Zarepour 2002 it is possible to truncate it if an appropriate Dirichlet distribution is used for the weights w_i. Thus a frequency distribution F is effectively represented, non-uniquely, by a very large but finite set of parameters (w_i, α_i, β_i).

The representation above was somewhat deprecated in recent works which, however, consider inference problems where variates have clear predictor or predictand roles (e.g. Wade et al., 2014b,a). As previously discussed, in many research fields there are no such a-priori roles; any variate could assume both. A representation that can easily swap the two roles without overemphasizing either is therefore most appropriate. The representation (4) leads to very simple, symmetric, analytical expressions for the conditional of A given B and vice versa, and for any marginal:

$$F(A \mid B) = \sum_i \frac{w_i \, K_A(A \mid \alpha_i) \, K_B(B \mid \beta_i)}{\sum_j w_j K_B(B \mid \beta_j)} \qquad F(B \mid A) = \sum_i \frac{w_i \, K_B(B \mid \beta_i) \, K_A(A \mid \alpha_i)}{\sum_j w_j K_A(A \mid \alpha_j)}$$

$$F(A) = \sum_i w_i \, K_A(A \mid \alpha_i) \; .$$

The (hyper)prior over the parameters representing the frequency distribution is a Dirichlet-process mixture. It allows for distributions having multiple, possibly sharp, peaks. For information about the kernels K for different kinds of variates, and about the prior distribution, see the technical manual[5].

The MCMC computation is based on Gibbs sampling; several chains are run in parallel. The stopping rule implements variations of the methods discussed in Vehtari et al. (2021). Regarding MCMC sampling, the package takes the following stance:

• For this kind of inference problems there is no need for tens of thousands or more independent MCMC samples (effective sample size). The package's default is just 3600. The mean of such samples is the probability (1), and the numerical uncertainty in this mean is thus around $\sqrt{3600} = 60$ times smaller than the "variability" of this probability, given by eq. (2).

• It is acceptable that the sampling has not *fully* converged (emphasis on 'fully'). Why? Consider that machine-learning algorithms such as neural networks are effectively Bayesian nonparametric functional regressors, and their training is essentially an *unconverged* Monte Carlo sampling (see e.g. MacKay, 1992; Gal, 2016; Mandt et al., 2017; Huszár, 2017). This "sampling" stops at a *local* maximum of the posterior, and all samples are discarded. Thanks to this lack of convergence, neural-network training is fast; despite this lack of convergence, inferences can still be impressive. It is this writer's opinion that the same stance can and should at times be adopted in Bayesian nonparametrics: inferences can still be impressive and informative despite lack of full MCMC convergence, and superior to those of many machine-learning algorithms designed for the same task, as we found by using this package. And the uncertainty in the results can moreover be approximately assessed, instead of fully discarded as in neural-network training. So far, in applying **inferno** to concrete problems, we noticed that the MCMC sampling had converged in the majority of cases. In those in which it had not fully converged, the results were still correct to a good approximation, for example with credibility intervals that erred by less than 2% from the converged values. This kind of approximate nonparametric results are still vastly superior to precise but opaque p-values.

These stances are motivated by the need for user-friendliness. The ultimate goal of the package is to let frequentist practitioners try and appreciate Bayesian methods (the package's main vignette is specially written for them); to let Bayesian-parametric practitioners try nonparametrics; and to show that Bayesian methods can be almost as fast and more powerful than many machine-learning approaches. Once the merits are seen, practitioners can slowly move to more complex Bayesian packages that allow for more control and custom problem-solving.

inferno has been already used and tested in several studies, but we'd like more testing before submitting it to CRAN. Testers and feedback are very welcome!

[5] https://github.com/pglpm/inferno/raw/main/development/manual/optimal_predictor_machine.pdf

References

J.-M. Bernardo and A. F. Smith. *Bayesian Theory*. Wiley series in probability and mathematical statistics. Wiley, New York, repr. edition, 2000. doi:10.1002/9780470316870. First publ. 1994.

J. M. Bernardo, M. J. Bayarri, J. O. Berger, A. P. Dawid, D. Heckerman, A. F. M. Smith, and M. West, editors. *Bayesian Statistics 9*. Oxford University Press, Oxford, 2011. doi:10.1093/acprof:oso/9780199694587.001.0001.

D. Draper. Assessment and propagation of model uncertainty. *J. R. Stat. Soc. B*, 57(1):45–70, 1995. doi:10.1111/j.2517-6161.1995.tb02015.x. See also discussion and reply in Spiegelhalter et al. (1995).

D. B. Dunson and A. Bhattacharya. Nonparametric Bayes regression and classification through mixtures of product kernels. In Bernardo et al. (2011), pages 145–158. 2011. doi:10.1093/acprof:oso/9780199694587.003.0005.

K. Dyrland, A. S. Lundervold, and P. G. L. Porta Mana. Don't guess what's true: choose what's optimal. A probability transducer for machine-learning classifiers, 2022. Open Science Framework doi:10.31219/osf.io/vct9y, arXiv doi:10.48550/arXiv.2302.10578.

Y. Gal. *Uncertainty in Deep Learning*. PhD thesis, University of Cambridge, Cambridge, 2016. https://www.cs.ox.ac.uk/people/yarin.gal/website/thesis/thesis.pdf.

N. L. Hjort, C. Holmes, et al., editors. *Bayesian Nonparametrics*. Cambridge series in statistical and probabilistic mathematics. Cambridge University Press, Cambridge, 2010. doi:10.1017/CBO9780511802478.

M. G. M. Hunink, M. C. Weinstein, et al. *Decision Making in Health and Medicine: Integrating Evidence and Values*. Cambridge University Press, Cambridge, 2 edition, 2014. doi:10.1017/CBO9781139506779. First publ. 2001.

F. Huszár. Everything that works works because it's Bayesian: Why deep nets generalize?, 2017. http://www.inference.vc/everything-that-works-works-because-its-bayesian-2.

H. Ishwaran and M. Zarepour. Exact and approximate sum representations for the Dirichlet process. *Can. J. Stat.*, 30(2):269–283, 2002. doi:10.2307/3315951.

D. V. Lindley and M. R. Novick. The role of exchangeability in inference. *Ann. Stat.*, 9(1):45–58, 1981. doi:10.1214/aos/1176345331.

D. J. C. MacKay. A practical Bayesian framework for backpropagation networks. *Neural Comput.*, 4(3):448–472, 1992. https://www.inference.org.uk/mackay/PhD.html, doi:10.1162/neco.1992.4.3.448.

D. J. C. MacKay. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, Cambridge, version 7.2 (4th pr.) edition, 2005. https://www.inference.org.uk/itila/book.html. First publ. 1995. See also video lectures at https://videolectures.net/events/course_information_theory_pattern_recognition.

S. Mandt, M. D. Hoffman, and D. M. Blei. Stochastic gradient descent as approximate Bayesian inference. *J. Mach. Learn. Res.*, 18:134, 2017. https://www.jmlr.org/papers/v18/17-214.html.

P. G. L. Porta Mana, I. Rye, et al. Personalized prognosis & treatment using an optimal predictor machine: An example study on conversion from mild cognitive impairment to Alzheimer's disease, 2023. Open Science Framework doi:10.31219/osf.io/8nr56.

H. C. Sox, M. C. Higgins, et al. *Medical Decision Making*. Wiley, New York, 3 edition, 2024. doi:10.1002/9781119627876. First publ. 1988.

D. J. Spiegelhalter, A. P. Grieve, et al. Discussion of the paper by Draper. *J. R. Stat. Soc. B*, 57(1):71–97, 1995. doi:10.1111/j.2517-6161.1995.tb02016.x. See Draper (1995).

A. Vehtari, A. Gelman, et al. Rank-normalization, folding, and localization: An improved $\hat{R}$ for assessing convergence of MCMC. *Bayesian Anal.*, 16(2):667–718, 2021. doi:10.1214/20-BA1221. With invited discussion.

S. Wade, D. B. Dunson, et al. Improving prediction from Dirichlet process mixtures via enrichment. *J. Mach. Learn. Res.*, 15(30):1041–1071, 2014a. http://jmlr.org/papers/v15/wade14a.html.

S. Wade, S. G. Walker, and S. Petrone. A predictive study of Dirichlet process mixture models for curve fitting. *Scand. J. Stat.*, 41(3):580–605, 2014b. doi:10.1111/sjos.12047.

S. G. Walker. Bayesian nonparametric methods: motivation and ideas. In Hjort et al. (2010), chapter 1, pages 22–34. 2010. doi:10.1017/CBO9780511802478.002.