Bassist — a tool for MCMC simulation of statistical models

Hannu Toivonen, Heikki Mannila, Marko Salmenkivi, and Karri-Pekka Laakso

University of Helsinki, Department of Computer Science, PO Box 26, FIN-00014 University of Helsinki, Finland
Hannu.Toivonen, Heikki.Mannila, Marko.Salmenkivi, Karri-Pekka.Laakso@cs.helsinki.fi

Introduction

Bassist is a simulation tool for the analysis of complex statistical models. Given a high-level specification of a full probability model, Bassist generates a simulator for analysing the model with respect to given data. Bassist follows the Bayesian modeling approach. Model parameters and other unknown quantities are assigned prior distributions by the modeler; the likelihood of observed data follows from the specification of the statistical model. The matter of interest is the posterior probability of the model, i.e., the joint probability distribution of all unknown quantities conditioned on the observed data. Bayesian statistics provide a well-founded means for assigning probabilities for models. For most realistic models, however, analytical methods for the computation of the posterior distribution are not feasible. To solve the problem, Bassist applies Markov chain Monte Carlo (MCMC) simulation techniques to approximate a sample from the joint posterior distribution.

MCMC simulation techniques are gaining popularity in the statistical modeling community, due to the growth of computational power to the level needed for serious simulations, and the success of Bayesian modeling. General purpose MCMC simulation software, however, has been almost non-existent, except for [9] and the Bugs system [12]. Simulator programs have been written specifically for each model, either from scratch of by reusing previous code. In either case, the task is tedious and error-prone. Bassist aims at helping modeler at implementing and analysing their models. In this paper we give a short overview of MCMC simulation and the Bassist system, and describe some of the applications in which Bassist has been used.

Full probability models and MCMC simulation

Bassist is a simulation tool for the analysis of hierarchical Bayesian models, also called full probability models; see, e.g., [6] for an excellent overview. Such a model describes the phenomenon of interest by a set of quantities and their statistical relationships. In a typical setting, some of the quantities are parameters of the model, and the model specifies the probability distribution of the rest of the quantities, or data, given values of the parameters. In a full probability model, the model parameters have prior probability distributions and the result is a distribution on the parameters, not just a point estimate (as in, e.g., maximum likelihood approaches).

In Bassist terminology, all quantities considered in a model are called variables, whether actually observed facts or model or hyper parameters. The full probability model is expressed as the product of densities of all variables in the model. The densities may be conditional on other variables, as long as the chain rule applies and the product of the specified densities equals (or is proportional to) the joint probability of all variables. In a typical simple setting, the density of data depends on model parameters and distributions of model parameters are expressed as unconditional prior probabilities, but there can be an arbitrary number of layers of conditional distributions.

More formally, a hierarchical Bayesian model is defined by giving a directed acyclic dependency graph $(V, \leq)$ where V is the set of model variables. Given a variable $v \in V$ and the set $pred(v) \subset V$ of its immediate predecessors in a dependency graph, the variable v is independent of other variables. The model specification is completed by defining the conditional probability $\Pr(v \mid pred(v))$ for all $v \in V$. The joint probability distribution of all variables is then $\prod_{v \in V} \Pr(v \mid pred(v))$.

In Bayesian analysis some variables are given as data, and the task is to compute the (posterior) distribution of the remaining variables. Let $D \subset V$ denote the set of variables given as data, and let $\theta = V \setminus D$ be the set of variables to be simulated; the task is to find the distribution $\Pr(\theta \mid D)$. Bayesian inference is a central tool used in this task. Denote by $succ(\theta) = \{w \in V \mid \exists v \in \theta \cap pred(w)\}$ the set of immediate successors of θ in the dependency graph. By the application of Bayes' Theorem we have that the posterior density of θ is proportional to the product of the prior density and likelihood of θ:

$$\Pr(\theta \mid D) = \frac{\Pr(\theta \wedge D)}{\Pr(D)} \propto \Pr(\theta \wedge D) \propto \prod_{v \in \theta} \Pr(v \mid pred(v)) \prod_{w \in succ(\theta)} \Pr(w \mid pred(w)). \quad (1)$$

MCMC and the Metropolis–Hastings method

The idea in MCMC techniques is to define a Markov chain with the domain of θ as state space, and the distribution $\Pr(\theta \mid D)$ as stationary distribution. A sample from the posterior distribution $\Pr(\theta \mid D)$ is obtained by simulating the Markov chain and outputting states of the chain as they are visited.

The Metropolis–Hastings method [8] used in Bassist constructs such a Markov chain by repeating, in each state θ_{i-1}, the following two steps. First, a possible next state θ_i is drawn from some distribution $q(\theta_{i-1}, \theta_i)$ (the proposal distribution). Then the new state θ_i is accepted with probability

$$\min\left(1, \frac{\Pr(\theta_i \mid D)\, q(\theta_i, \theta_{i-1})}{\Pr(\theta_{i-1} \mid D)\, q(\theta_{i-1}, \theta_i)}\right). \quad (2)$$

Unlike the more widely used Gibbs sampling [11], the Metropolis-Hastings algorithm is not limited to conjugate distributions but can handle (in principle) arbitrary distributions. Equation (1) shows how the posterior density can be computed from the dependency graph; the only computational requirement is that the user-specified conditional densities can be computed. By the application of the Metropolis–Hastings method (formula (2)), one can simulate samples from extremely complex posterior distributions. Metropolis-Hastings-algorithm can even be applied so that the dimension of the model is also a random variable and thus the amount of parameters can change during simulation [7]. This property makes it superior to Gibbs sampling in many modeling tasks by allowing non-parametric models with an arbitrary amount of parameters.

For computational convenience, usually only one parameter is updated at a time, i.e., vector θ_i differs from θ_{i-1} in one position only. In Bassist, the default for the proposal q is for each variable its own prior distribution unless the distribution is a user-defined function. In such cases the user must specify a suitable q, such as a random walk method. Bassist also supports updates of multiple variables at a time.

Design of Bassist

Bassist is a domain independent, general purpose tool for Bayesian analysis. Bassist compiles a model specification to a simulation program that performs Metropolis-Hastings sampling to obtain an approximation of the posterior distribution of the model. In addition to models that can be expressed as simple and finite graphs, models can also contain non-parametric components, including piece-wise constant functions with a varying number of pieces. Another special feature supported by Bassist is the processing of event sequences and intensity models [1,3]. This makes the tool quite useful for modeling event data ubiquitous in epidemiology, fault management, and quality control.

For Bassist, the major differences when compared to Bugs [12] are the capability of simulating models with complex non-parametric components and with event sequences. In principle, models of arbitrary probability distributions can be simulated as long as the user is able to specify the corresponding density functions.

Bassist allows the user to specify a full probability model using a high-level model description language. The model specification language of Bassist is declarative, and it has been designed to have a clear separation between the model specification, the simulation details, and the particulars of the data. The language is based on a data modeling view of hierarchical Bayesian models; this makes the language easy to use and suitable for data mining applications.

Bassist is implemented as a simulator generator. The model specifications are first transformed into a C++ program which is then compiled and linked with the Bassist library. The produced executable reads the data and performs the actual simulation. Produced programs can simulate any combinations of variables of the model: the input data determines which parts are given as data. This design not only allows easier experimentation but also makes the model specification easier.

The system uses the class mechanisms of C++ to obtain a clean and fast implementation of parameter updates in the Metropolis–Hastings algorithm. Constant parts of simulators are available as components in a library. These include, for instance, classes for variables, standard distributions, and sampling. Depending on the model specification, code is generated to cope with the dependency structures, proposal methods, and user-defined density functions. The density functions are provided by the user as C++ code that is linked to the system, so there are practically no limits for them. While Bassist helps in the implementation of models, the user naturally remains responsible for the correctness and convergence of the model. – Bassist is currently under development. A publicly available version will appear in 1998.

Example applications

Palaeoecological reconstruction

In the palaeoecological reconstruction task of [10] the goal is to estimate past temperatures based on fossil assemblages of chironomid taxa. The idea is to specify an ecologically plausible model for the response of the organisms to lake temperature, fit this model using modern data where both temperatures and taxa abundances are known, and finally apply this model in the reconstruction task where only organism abundances are known.

The central part of our model for this task is the specification of how the abundances of taxa respond to the lake temperature. Following the traditional response-curve model in palaeoecology [2], we model the response function of a taxon k to temperature z as a bell-shaped curve, defined by three parameters $\alpha_k, \beta_k, \gamma_k$:

$$r(\alpha_k, \beta_k, \gamma_k, z) = \alpha_k \exp\left(\frac{(z-\beta_k)^2}{\gamma_k}\right).$$

Here β_k is the optimum temperature for the taxon and the location of the maximum of the response curve, α_k is the height of the curve at the maximum, and γ_k determines how steep the curve is. Given a temperature z, we assume the abundances of taxon k are Poisson distributed with parameter $r(\alpha_k, \beta_k, \gamma_k, z)$. For simplicity, we assume here the taxa are independent. To complete the model, we specify prior distributions for the parameters $\alpha_k, \beta_k, \gamma_k, z$ such that the probability of a value combination reflects the probability of such response function occurring in the nature. Note that throughout the process of designing the model the role of domain experts is essential. In the provisional Bassist language the model is expressed in the following way.

```
record taxon {
     var alpha ~ Unif(0.1, 100),
     var beta ~ Normal(12.1, 4),
     var gamma ~ Gamma(50, 5)
};
record lake {
     var z ~ Normal(12.1, 0.4)
};
record measurement {
     ref taxon,
     ref lake,
     var amount ~ Poisson(r(taxon.alpha, taxon.beta, taxon.gamma, lake.z))
};
```

The function `r` is defined elsewhere in C++. The user may additionally specify proposal methods for parameters and in this way affect the convergence. The actual data, and the number of iterations and burn-in iterations are specified at run time.

Now with the modern calibration set we know the actual responses of each taxon in a number of lakes, and the parameters $\alpha_k, \beta_k, \gamma_k$ are simulated for each taxon k. In the actual reconstruction, temperatures of unknown lakes are simulated based on the current values of response parameters. Within Bassist, every unknown variable is simulated with the same principles and mechanisms. There is no need at all to discriminate, e.g., between model parameters, data, and missing data, and actually the whole simulation process can be performed at once: all parameters $\alpha_k, \beta_k, \gamma_k$ are unknown, whereas some of the temperatures z are known and some unknown. A more detailed description of the model can be found in [13].

Figure 1 below shows the trace of a simulated temperature during the first 1000 iterations. In the setting of the figure, much longer runs are necessary to ensure convergence. Figure 2 below displays a summary of the result for the period of last 10000 years.

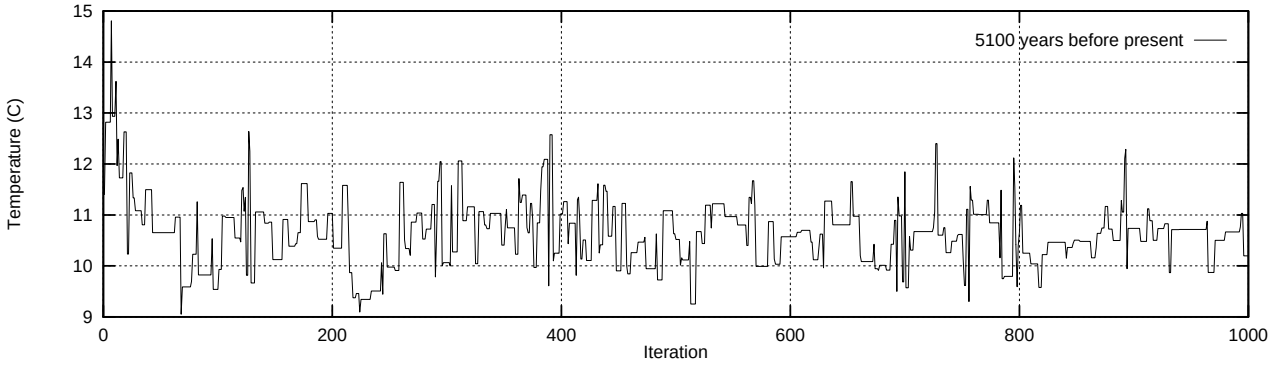

Figure 1: The trace of the first 1000 iterations of a simulated temperature of approximately 5100 years ago.

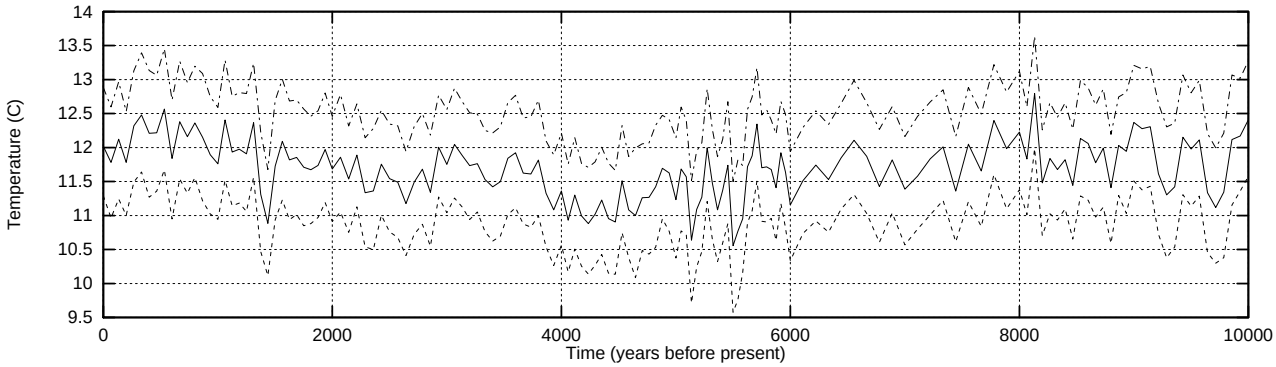

Figure 2: A result of a simulation for the whole reconstruction problem: solid line is posterior mean, dotted lines are 90 % credibility intervals.

Ear infections

As an example of modeling event sequences we discuss analysing occurrences of middle ear infections in small children. In addition to the occurrences of ear infection episodes, our data consists of changes in several risk factors. For illustration purposes we will only consider here the effects of two risk factors: weaning and day care attendance. A more detailed description of the problem is found in [5].

Intensity functions are a common methodological tool in applications of this kind. They are used to describe the instantaneous risk of observing the event of interest over time [1,3]. We modeled the age-dependent intensity by a piece-wise constant function. The levels of the function, the jump times and also the number of

pieces were random variables. We added observed history in terms of time-dependent indicator functions of weaning and day care multiplicatively into the model. To separate the effects for the first and subsequent infections, we used the following parameterisation [4] to model the infection intensity $\lambda_i(t)$ (for individual i):

$$\lambda_i(t) = f(t) \cdot Y_i(t) \cdot \beta_i(t) \cdot \gamma_i(t) \cdot \alpha_i(t) \cdot \beta_i^*(t) \cdot \gamma_i^*(t)$$

Here $f(t)$ is the piece-wise constant baseline function and $Y_i(t)$ is an indicator function of being at risk at time t. The time-dependent functions of weaning ($\beta_i(t)$) and day care ($\gamma_i(t)$) indicate the relative risk of these factors compared to baseline hazard $f(t)$. The value of β_i is defined to be 1 after weaning, otherwise it is a parameter (β) to be simulated. Similarly, the value of γ_i is 1 before day care attendance and a random variable (γ) after that time. The values of β or γ < 1 correspond to protective effect and the values > 1 to increased risk. The indicator function $\alpha_i(t)$ is used to distinguish between those who have had at least one infection already. The functions $\beta_i^*(t)$ and $\gamma_i^*(t)$ correspond to the additional effect of weaning and day care, respectively, on recurrent infections compared to the effect on the first infection. The total relative risk for recurrences is therefore $\beta \cdot \beta^*$ (and $\gamma \cdot \gamma^*$ respectively).

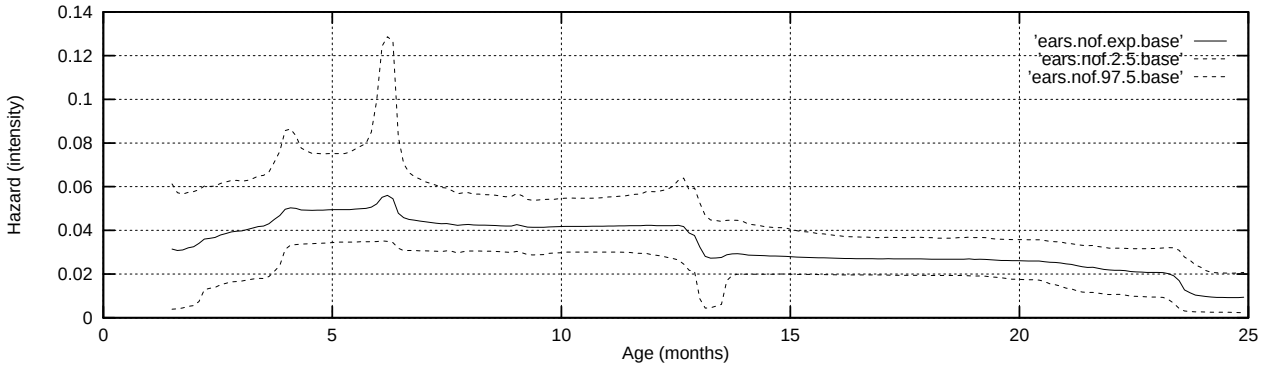

Figure 3. Posterior mean and 95 %-probability intervals of the piece-wise constant baseline function f(t).

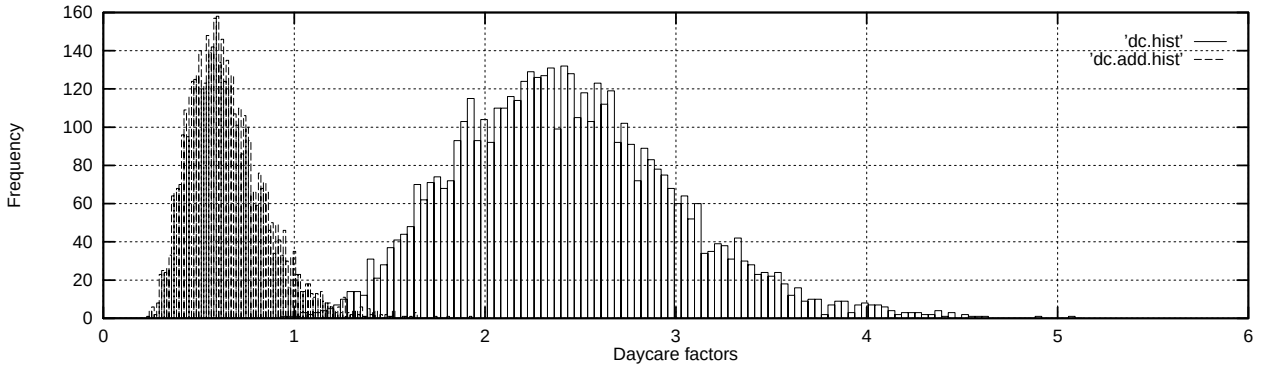

Figure 4. Posterior distributions of the parameters γ and γ^* that were related to the day care attendance.

The point-wise averages and the 95 %-probability intervals of the piece-wise constant posterior baseline function $f(t)$ are shown in Figure 3. The strong risk effect of attending day care for the first infection indicated by high values of parameter γ loses significance for recurrent events, as the interaction parameter γ^* is clearly below 1. The posterior distributions of these factors are presented in Figure 4. The simulation was run for 50000 updates for each parameter and the parameter values were picked up in every tenth round.

Conclusions

We have described Bassist, a tool for the simulation and analysis of statistical models. Bassist is based on the Bayesian full probability paradigm, and it uses MCMC methods to approximate complex posterior

distributions. Bassist is targeted mainly at the statistical modeling community, where Bayesian modeling is gaining popularity but the implementation of simulators is a burden.

We described two applications of Bassist. Bayesian modeling provides sound conceptual and mathematical tools for reasoning with complex probability models, and for incorporating available prior knowledge to the reasoning. In the reconstruction task this means that the models can and should be ecologically realistic and plausible, irrespective of their possible statistical complexity. Reasoning on full distributions is another advantage. Distributions give more information than simple point estimates available, e.g., from maximum a posterior analysis. Distributions are especially useful in the estimation of the uncertainty of the inference.

Acknowledgements

We would like to thank Heikki Olander and Atte Korhola, Laboratory of Physical Geography, University of Helsinki, for providing an extended chironomid data set of [10]. We also thank the Pnc study group from the National Public Health Institute (KTL), Finland, for letting us use the ear infection data.

References

[1] Arjas, E.: Survival models and martingale dynamics. *Scandinavian Journal of Statistics*, 16, p. 177–225, 1989.

[2] Birks, H. J. B.: Quantitative palaeoenvironmental reconstructions. *Statistical modelling of Quaternary science data, Technical Guide 5,* edited by Maddy, D. and Brew, J.S. , p. 161–254, Quaternary Research Association, Cambridge, 1995.

[3] Cox, D. R.; Oakes, D.: Analysis of survival data. *Monographs on Statistics and Applied Probability 21*. Chapman & Hall, 1984.

[4] Eerola, M.: Repeatable events in event-history analysis; the effect of childhood separation on future mental health. *Bull. Int. Statist.Inst. vol LIII(1)*, p. 213-225, 1989.

[5] Eerola, M., Mannila, H., Salmenkivi, M.: Frailty factors and time-dependent hazards in modelling ear infections in children using Bassist. Preliminary version. To appear in *Proceedings of 13th Symposium in Computational Statistics*, Bristol, 1998.

[6] Gelman, A., Carlin, J., Stein, H., Rubin, D.: Bayesian Data Analysis. Texts in Statistical Science. Chapmann & Hall, 1995.

[7] Green P.: Reversible jump Markov chain Monte Carlo computation and Bayesian model determination.. *Biometrica*, 82, 4 , p. 711-732, 1995.

[8] Hastings, W. K.: Monte Carlo sampling methods using Markov chains and their applications. *Biometrica*, 57, p. 97–109, 1970.

[9] Neal, R. M.: Bayesian learning for Neural Networks. Ph.D. thesis, Dept. of Computer Science, University of Toronto, 1994.

[10] Olander, H.; Korhola, A.; Birks, H. J. B.; Blom, T.: Surface sediment Chironomidae (Insecta: Diptera) distributions along an ecotonal transect in subarctic Fennoscandia: developing a tool for palaeotemperature reconstructions. *Journal of Paleolimnology,* 18, p. 45–59, 1997.

[11] Smith, A. F. M.; Roberts, G. O.: Bayesian computation via the Gibbs sampler and related Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society*, Series B, 55, No. 1, p. 3–23.

[12] Spiegelhalter, D. J.; Thomas, A.; Best, N. G.; Gilks, W. R.: BUGS Version 0.5, Bayesian inference using Gibbs sampling, MRC Biostatistics Unit, Institute of Public Health, Cambridge, UK, 1995.

[13] Toivonen, H; Mannila, H; Korhola, A.: Paleoecological reconstruction using Bayesian methods and MCMC simulation. Manuscript, 1998.