A Bayesian reassessment of nearest-neighbour classification

Jean-Michel Marin, Christian Robert, Mike Titterington

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

A Bayesian reassessment of nearest–neighbour classification

Jean-Michel Marin — Christian P. Robert — Mike Titteringtion

N° ????

Avril 2007

Thème COG

Rapport de recherche

ISSN 0249-6399 ISRN INRIA/RR--????--FR+ENG

A Bayesian reassessment of nearest–neighbour classification

Jean-Michel Marin* , Christian P. Robert † , Mike Titterington ‡

Thème COG — Systèmes cognitifs
Projets SELECT

Rapport de recherche n° ???? — Avril 2007 — 25 pages

Abstract: The k-nearest-neighbour procedure is a well-known method used in supervised classification. While it has been superseded by more recent methods developed in machine learning, it remains an essential tool for classifiers. This paper proposes a reassessment of this approach as a statistical technique derived from a proper probabilistic model; in particular, we modify the assessment made in a previous analysis of this method undertaken by Holmes and Adams (2002, 2003) where the underlying probabilistic model is not completely well-defined. Once clear probabilistic bases of the k-nearest-neighbour procedure are established, we proceed to the derivation of practical computational tools to conduct Bayesian inference on the parameters of the corresponding model. In particular, we assess the difficulties inherent to pseudo-likelihood and to path sampling approximations of a missing normalising constant, and propose a perfect sampling strategy to implement a correct MCMC sampler associated with our model. Illustrations of the performance of the corresponding Bayesian classifier are provided for two benchmark datasets, demonstrating in particular the limitations of the pseudo-likelihood approximation in this set-up.

Key-words: Bayesian inference, classification, compatible conditionals, Boltzmann model, normalising constant, pseudo-likelihood, path sampling, perfect sampling

* INRIA Futurs, Projet SELECT, Université Paris-Sud
† CEREMADE, Université Paris Dauphine and CREST-INSEE
‡ University of Glasgow

Unité de recherche INRIA Futurs
Parc Club Orsay Université, ZAC des Vignes,
4, rue Jacques Monod, 91893 ORSAY Cedex (France)
Téléphone : +33 1 72 92 59 00 — Télécopie : +33 1 60 19 66 08

Reformulation bayésienne de la méthode des k-plus-proches-voisins

Résumé : Bien que maintenant supplantée par des méthodes plus récentes, l'heuristique des k-plus-proches-voisins reste essentielle en classification supervisée. Dans cet article, nous en proposons une reformulation sous forme d'une modèle statistique. Nous corrigeons ainsi les reformulations effectuées par Holmes and Adams (2002, 2003) pour lesquelles le modèle sous-jacent n'est pas proprement défini. Le modèle proposé dépend d'une constante de normalisation inconnue. Nous nous plaçons dans le paradigme bayésien et comparons différentes méthodes d'inférence palliant cette difficulté. Nous étudions les limites de l'utilisation de la pseudo-vraisemblance et de la méthode d'Ogata (Ogata, 1989) dans un schéma MCMC et proposons une méthode MCMC exacte basée sur la simulation parfaite par couplage. Nous illustrons les performances de cet algorithme sur divers jeu de données.

Mots-clés : Inférence bayésienne, classification, lois conditionnelles compatibles, modèle de Boltzmann, constante de normalisation, pseudo-vraisemblance, méthode d'Ogata, simulation parfaite par couplage

1 Introduction

1.1 Statistical classification

Supervised classification has been used for quite a while both in machine learning and in Statistics to infer about the functional connection between a group of covariates (or explanatory variables) and a vector of indicators (or classes) (see, e.g., McLachlan, 1992; Ripley, 1994, 1996; Devroye et al., 1996; Hastie et al., 2001). For instance, the method of *boosting* (Freund and Schapire, 1997) has been developed for this very purpose by the Machine Learning community and it has also been assessed and extended by statisticians (Hastie et al., 2001; Bühlmann and Yu, 2002, 2003; Bühlmann, 2004; Zhang and Yu, 2005).

The k-nearest-neighbour method is a well-established straightforward technique in this area with both a long past and a fairly resilient resistance to change (Ripley, 1994, 1996). It nonetheless suffers from the difficulty that, while providing an instrumental device to classify points into two or more classes, it lacks a corresponding assessment of its classification error. While alternative techniques like boosting offer this assessment, it is obviously of interest to provide the original k-nearest-neighbour method with this additional feature. In contrast, statistical classification methods that are based on a model like a mixture of distributions do provide an error assessment along with the most likely classification. This more global perspective thus requires the technique to be embedded within a probabilistic framework in order to give a proper meaning to the notion of classification error. In an earlier paper, Holmes and Adams (2002) proposes a Bayesian analysis of the k-nearest-neighbour-method based on those premises and we refer the reader to this paper for background and references. Holmes and Adams (2003) define another model based on autologistic representations and conduct a likelihood analysis of this model. While we also adopt a Bayesian approach, our paper differs from Holmes and Adams (2002) in two important respects: first, we define a global probabilistic model that encapsulates the k-nearest-neighbour method, rather than working with incompatible conditional distributions, and, second, we derive a fully operational simulation technique adapted to our model and based on perfect sampling, that allows for a reassessment of the pseudo-likelihood approximation often used in those settings.

1.2 The original k-nearest-neighbour method

Given a training set of individuals allocated to one of G classes, the classical k-nearest-neighbour procedure is a method that allocates new individuals to the most common class in their neighbourhood among the training set, the neighbourhood being defined in terms of the covariates. More formally, based on a training dataset $((y_i, x_i))_{i=1}^n$ where $y_i \in \{1, \ldots, G\}$ denotes the class label of the i-th point and $x_i \in \mathbb{R}^p$ is a vector of covariates, an unobserved class y_{n+1} associated with a new set of covariates x_{n+1} is estimated by the most common class among the k nearest neighbours of x_{n+1} in the training set $(x_i)_{i=1}^n$; hence the name k-nearest-neighbour. The neighbourhood is defined in the space of the covariates x_i, namely

$$\mathcal{N}_{n+1}^k = \left\{1 \leq i \leq n;\, d(x_i, x_{n+1}) \leq d(\cdot, x_{n+1})_{(k)}\right\} ,$$

where $d(\cdot, x_{n+1})$ denotes the vector of distances to x_{n+1} and $d(\cdot, x_{n+1})_{(k)}$ denotes the k-th order statistic. The original k-nearest-neighbourmethod usually uses the Euclidean norm, even though the Mahalanobis distance would be more appropriate in that it rescales the covariates. Whenever ties occur, they are resolved by decreasing the number k of neighbours until the problem disappears.

As such, and as also noted in Holmes and Adams (2002), the method is both deterministic, given the training dataset, and not parameterised, even though the choice of k is both non-trivial and relevant to the performance of the method. Usually, k is selected via cross-validation, as the number of neighbours that minimises the cross-validation error rate.

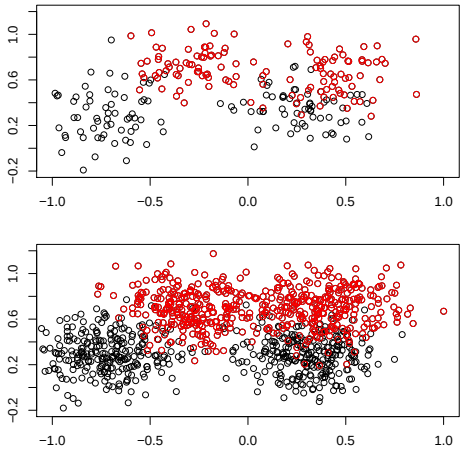

Figure 1: Training *(top)* and test *(bottom)* groups for Ripley's benchmark: the points in red are those for which the label is equal to 1 and the points in blue are those for which the label is equal to 2.

To illustrate the original method and to compare it later with our approach, we use throughout a toy benchmark dataset taken from Ripley (1994). This dataset corresponds to a two-class classification problem where each (sub)population of covariates is simulated from a bivariate normal distribution, both populations being of equal sizes. A sample of $n = 250$ individuals is used as the training set and the model is tested on a second group of $m = 1,000$ points acting as a test dataset. Figure 1 presents the dataset[1] and Table 1 displays the performance of the standard k-nearest-neighbour method on the test dataset for several values of k. However, we want to point out the disturbing feature that, using only the training dataset, there are ten different values of k that achieve the same minimum in the leave-one-out cross validation error rate, namely 17, 18, 35, 36, 45, 46, 51, 52, 53 and 54, which reflects rather negatively on the discriminative abilities of the method!

[1]This dataset is available at http://www.stats.ox.ac.uk/pub/PRNN.

k	Misclassification error rate
1	0.150
3	0.134
15	0.095
17	0.087
31	0.084
54	0.081

Table 1: k-nearest-neighbour performances on the Ripley test dataset

1.3 Goal and plan

As presented above, the k-nearest-neighbour method is merely an allocation technique that does not account for uncertainty and, as such, does not pertain to Statistics. To allow for uncertainty, we need to introduce a probabilistic framework that relates the class label y_i to both the covariates x_i *and* the class labels of the neighbours of x_i. Not only does this perspective provide more information about the variability of the classification, when compared with the point estimate given by the original method, but it also takes advantage of the full (Bayesian) inferential machinery to introduce parameters that measure the strength of the influence of the neighbours, and to analyse the role of the variables, of the metric, of the number k of neighbours and of the classes towards achieving higher efficiency. Once again, this statistical viewpoint was previously adopted by Holmes and Adams (2002, 2003) and we follow suit in this paper, with a modification of their original model geared towards more coherence, while providing new developments in computational model estimation.

The paper is organised as follows. We establish the validity of the new probabilistic k-nearest-neighbour model in Section 2, pointing out the deficiencies of the models advanced by Holmes and Adams (2002, 2003), and then cover the different aspects of running Bayesian inference in this k-nearest-neighbour model in Section 3, addressing in particular the specific issue of evaluating the normalising constant of the probabilistic k-nearest-neighbour model that is necessary for inferring about k and an additional parameter. We take advantage of an exact MCMC approach proposed in Section 3.4 to evaluate the limitations of the pseudo-likelihood alternative in Section 3.5 and illustrate the method on a Pima Indian diabetes benchmark dataset in Section 4.

2 The probabilistic k-nearest-neighbour model

2.1 Markov random field modelling

In order to build a probabilistic structure that reproduces the features of the original k-nearest-neighbour procedure and then to estimate its unknown parameters, we first need to

define a joint distribution of the labels y_i conditional on the covariates x_i, for the training dataset. A natural approach is to take advantage of the spatial structure of the problem and to use a Markov random field model. Although we will show below that this is not possible within a coherent probabilistic setting, we could thus assume that the full conditional distribution of y_i given $\mathbf{y}_{-i} = (y_1, \ldots, y_{i-1}, y_{i+1}, \ldots, y_n)$ and the x_i's only depends on the k nearest neighbours of x_i in the training set. The parameterised structure of this conditional distribution is obviously open but we opt for the most standard choice, namely, like the Potts model, a Boltzmann distribution (Møller and Waagepetersen, 2003) with potential function

$$\sum_{\ell \sim_k i} \delta_{y_i}(y_l)\,,$$

where $\ell \sim_k i$ means that the summation is taken over the observations x_ℓ belonging to the k nearest neighbours of x_i, and $\delta_a(b)$ denotes the Dirac function. This function actually gives the number of points from the same class y_i as the point x_i that are among the k nearest neighbours of x_i. As in Holmes and Adams (2003), the expression for the full conditional is thus

$$f(y_i|\mathbf{y}_{-i}, \mathbf{X}, \beta, k) = \exp\left(\beta \sum_{\ell \sim_k i} \delta_{y_i}(y_l) \Big/ k\right) \Big/ \sum_{g=1}^{G} \exp\left(\beta \sum_{\ell \sim_k i} \delta_g(y_l) \Big/ k\right) \tag{1}$$

where $\beta > 0$ and $\mathbf{X}$ is the (p, n) matrix $\{x_1, \ldots, x_n\}$ of coordinates for the training set.

In this parameterised model, β is a quantity that is obviously missing from the original k-nearest-neighbour procedure. It is only relevant from a statistical point of view as a degree of uncertainty: $\beta = 0$ corresponds to a uniform distribution over all classes, meaning independence from the neighbours, while $\beta = +\infty$ leads to a point mass distribution at the prevalent class, corresponding to the ultimate dependence. The introduction of the scale parameter k in the denominator is useful in making β dimensionless.

There is, however, a difficulty with this expression which is that, for almost all datasets $\mathbf{X}$, there does not exist a joint probability distribution on $\mathbf{y} = (y_1, \ldots, y_n)$ with full conditionals equal to (1). The reason for this lack of a joint distribution is related to the basic property that the k-nearest-neighbour system is usually asymmetric: when x_i is one of the k nearest neighbours of x_j, x_j is not necessarily one of the k nearest neighbours of x_i. Therefore, the pseudo-conditional distribution (1) will not depend on x_j while the equivalent for x_j does depend on x_i: this is obviously impossible in a coherent probabilistic framework (Besag, 1974; Cressie, 1993; Besag and Kooperberg, 1995).

One way to overcome this fundamental difficulty is to follow Holmes and Adams (2002) and to define directly the joint distribution

$$f(\mathbf{y}|\mathbf{y}_{-i}, \mathbf{X}, \beta, k) = \prod_{i=1}^{n} \exp\left(\beta \sum_{\ell \sim_k i} \delta_{y_i}(y_l) \Big/ k\right) \Big/ \sum_{g=1}^{G} \exp\left(\beta \sum_{\ell \sim_k i} \delta_g(y_l) \Big/ k\right). \tag{2}$$

Unfortunately, there are drawbacks to this approach, in that, first, the function (2) is not properly normalised (a fact overlooked by Holmes and Adams, 2002), with an untractable

normalising constant, and, second, the full conditional distributions corresponding to this joint distribution are not given by (1). The first drawback is a common occurrence with Boltzmann models and we will deal with this difficulty in detail below (see Section 3). The second drawback implies that (2) cannot be treated as a pseudo-likelihood (Besag and Kooperberg, 1995) since, as stated above, the conditional distribution (1) cannot be associated with any joint distribution. That (2) misses a normalising constant can be seen from the special case in which $n = 2$, $\mathbf{y} = (y_1, y_2)$ and $G = 2$, since

$$\begin{aligned}
\sum_{y_1=1}^{2}\sum_{y_2=1}^{2}\prod_{i=1}^{2}\exp\left(\beta\sum_{\ell\sim_k i}\delta_{y_i}(y_l)\Big/k\right)\Big/\sum_{g=1}^{2}\exp\left(\beta\sum_{\ell\sim_k i}\delta_{g}(y_l)\Big/k\right)\\
&= \sum_{y_1=1}^{2}\sum_{y_2=1}^{2}\exp\left(\beta\left[\delta_{y_1}(y_2)+\delta_{y_2}(y_1)\right]/k\right)\Big/\left(1+e^{\beta/k}\right)^2\\
&= 2\left(1+e^{2\beta/k}\right)\Big/\left(1+e^{\beta/k}\right)^2,
\end{aligned}$$

which is clearly different from 1 and, more importantly, depends on both β and k.

2.2 A symmetrised Boltzmann modelling

Given these difficulties, we therefore adopt a different strategy and define a joint model on the training set as

$$f(\mathbf{y}|\mathbf{X},\beta,k) = \exp\left(\beta\sum_{i=1}^{n}\sum_{\ell\sim_k i}\delta_{y_i}(y_l)\Big/k\right)\Big/ Z(\beta,k)\,, \tag{3}$$

where $Z(\beta,k)$ is the normalising constant of the distribution. The motivation for this modelling is that the full conditional distributions corresponding to (3) can be obtained as

$$f(y_i|\mathbf{y}_{-i},\mathbf{X},\beta,k) \propto \exp\left\{\beta/k\left(\sum_{\ell\sim_k i}\delta_{y_i}(y_l)+\sum_{i\sim_k \ell}\delta_{y_l}(y_i)\right)\right\}, \tag{4}$$

where $i \sim_k \ell$ means that the summation is taken over the observations x_ℓ for which x_i is a k-nearest neighbour. Obviously, these conditional distributions differ from (1) if only because of the impossibility result mentioned above. The additional term in the potential function corresponds to the observations that are not among the nearest neighbours of x_i but for which x_i is a nearest neighbour. In this model, compared with single neighbours, mutual neighbours are given a double weight. This feature is of importance in that this coherent model defines a new classification criterion that can be treated as a competitor of the standard k-nearest-neighbour objective function. Note also that the original full conditional (1) is recovered as (4) when the system of neighbours is perfectly symmetric (up to a factor 2). Once again, the normalising constant $Z(\beta,k)$ is intractable, except for the most trivial cases.

In the case of unbalanced sampling, that is, if the marginal probabilities $p_1 = \mathbb{P}(y = 1), \ldots, p_G = \mathbb{P}(y = G)$ are known and are different from the sampling probabilities $\tilde{p}_1 = n_1/n, \ldots, \tilde{p}_G = n_G/n$, where n_g is the number of training observations arising from class g, a natural modification of this k-nearest-neighbour model is to reweight the neighbourhood sizes by $a_g = p_g n/n_g$, leading to the modified model

$$f(\mathbf{y}|\mathbf{X}, \beta, k) = \exp\left(\beta \sum_i a_{y_i} \sum_{\ell \sim_k i} \delta_{y_i}(y_l) \Big/ k\right) \Big/ Z(\beta, k)\,.$$

This modification is useful in practice when dealing with stratified surveys. In the following, we assume however that $a_g = 1$ for all $g = 1, \ldots, G$.

2.3 Predictive perspective

When based on (4), the predictive distribution of a new unclassified observation y_{n+1} given its covariate x_{n+1} and the training sample $(\mathbf{y}, \mathbf{X})$ is, for $g = 1, \ldots, G$,

$$\mathbb{P}(y_{n+1} = g|x_{n+1}, \mathbf{y}, \mathbf{X}, \beta, k) \propto \exp\left\{\beta/k \left(\sum_{\ell \sim_k (n+1)} \delta_g(y_l) + \sum_{(n+1) \sim_k \ell} \delta_{y_l}(g)\right)\right\}, \qquad (5)$$

where

$$\sum_{\ell \sim_k (n+1)} \delta_g(y_l) \quad \text{and} \quad \sum_{(n+1) \sim_k \ell} \delta_{y_l}(g)$$

are the numbers of observations in the training dataset from class g among the k nearest neighbours of x_{n+1} and among the observations for which x_{n+1} is a k-nearest neighbour, respectively. This predictive can then be incorporated in the Bayesian inference process by considering the joint posterior of (β, k, y_{n+1}) and by deriving the corresponding marginal posterior distribution on y_{n+1}.

While this model provides a sound statistical basis for the k-nearest-neighbour methodology as well as a means of assessing the uncertainty of the allocations to classes of unclassified observations, and while it corresponds to a true albeit unavailable joint distribution, it can be argued that, from a Bayesian point of view, it can be criticised because it suffers from a lack of statistical coherence when considering multiple classifications. Indeed, the k-nearest-neighbour methodology is overwhelmingly used in a repeated manner, either jointly on a sample $(x_{n+1}, \ldots, x_{n+m})$ or sequentially. Rather than assuming both dependence in the training sample and independence in the unclassified sample, it would be more sensible to consider the whole collection of points as issued from a single joint model of the form given by (3), some of which have their class missing at random. Always reasoning from a Bayesian point of view, addressing simultaneously the inference on the parameters (β, k) and on the missing classes $(y_{n+1}, \ldots, y_{n+m})$—i.e. assuming *exchangeability* between the training and the unclassified datapoints—is certainly more coherent and it does provide a better perspective on the uncertainty about the classifications as well as about the parameters.

Unfortunately, this more global and arguably more coherent perspective is mostly unachievable for computational reasons, if none other, since the set of the missing class vector $(y_{n+1}, \ldots, y_{n+m})$ is of size G^m. It is practically impossible to derive an efficient simulation algorithm that would correctly approximate the joint probability distribution of both parameters and classes, especially when the number m of unclassified points is large. We will thus adopt the slightly less coherent approach of dealing separately with each unclassified point in the analysis, because this simply is the only realistic way. This perspective can also be justified by the fact that, in realistic machine learning set-ups, the unclassified data $(y_{n+1}, \ldots, y_{n+m})$ mostly occurs in a sequential environment with, furthermore, the true value of y_{n+1} being revealed before y_{n+2} is observed.

In the following sections, we consider solely the case $G = 2$ as in Holmes and Adams (2003).

3 Bayesian inference and the normalisation problem

Given the joint model (3) for $(y_1, \ldots, y_{n+1})$, Bayesian inference can be conducted in a standard manner (Robert, 2001), provided computational difficulties related to the unavailability of the normalising constant can be solved. Indeed, as stressed in the previous section, from a Bayesian perspective, the classification of unclassified points can be based on the marginal predictive (or posterior) distribution of y_{n+1} obtained by integration over the conditional posterior distributions of the parameters, namely, for $g = 1, 2$,

$$\mathbb{P}(y_{n+1} = g | x_{n+1}, \mathbf{y}, \mathbf{X}) = \sum_k \int \mathbb{P}(y_{n+1} = g | x_{n+1}, \mathbf{y}, \mathbf{X}, \beta, k) \pi(\beta, k | \mathbf{y}, \mathbf{X}) \, \mathrm{d}\beta \,, \qquad (6)$$

where $\pi(\beta, k|\mathbf{y}, \mathbf{X}) \propto f(\mathbf{y}|\mathbf{X}, \beta, k)\pi(\beta, k)$ is the posterior distribution of (β, k) given the training dataset $(\mathbf{y}, \mathbf{X})$. While other choices of prior distributions are available, we choose a compact support for both k and β, namely a uniform prior on $\{1, \ldots, K\} \times [0, \beta_{\max}]$ for (k, β). The limitation on k is imposed by the structure of the training dataset in that K is at most equal to the minimal class size, $\min(n_1, n_2)$, while the limitation on β, $\beta < \beta_{\max}$, is customary in Boltzmann models, due to transition phase phenomena (Møller, 2003): when β is above a certain value, the model becomes "all black or all white", i.e. all y_i's are either equal to 1 or to 2. (This is illustrated in Figure 3 below by the convergence of the expectation of the number of identical neighbours to k.)

3.1 MCMC steps

Were the posterior distribution $\pi(\beta, k|\mathbf{y}, \mathbf{X})$ available (up to a normalising constant), we could design an MCMC algorithm that would produce a Markov chain approximating a sample from this posterior (Robert and Casella, 2004). First, it would then be possible to use a Gibbs sampling scheme based on the full conditional distributions of both k and β. However, due to the associated representation (4), the conditional distribution of β is non-standard and we need to resort to a hybrid sampling scheme in which the exact simulation

from $\pi(\beta|k,\mathbf{y},\mathbf{X})$ is replaced with a single Metropolis–Hastings step. Furthermore, use of the full conditional distribution for k can impose fairly severe computational constraints. Indeed, for a given value $\beta^{(t)}$, computing the posterior $f(\mathbf{y}|\mathbf{X},\beta^{(t)},i)\pi(\beta^{(t)},i)$, for $i=1,\ldots,K$, requires computations of order $\mathrm{O}(KnG)$, once again because of the likelihood representation. A faster alternative is to use a hybrid step for both β and k: in this way, we only need to compute the full conditional distribution of k for one new value of k, modulo the normalising constant.

An alternative to Gibbs sampling is to use instead a random walk Metropolis–Hastings algorithm: both β and k are then updated using random walk proposals. Since $\beta\in(0,\beta_{\max})$ is constrained, we first introduce a logistic reparameterisation of β,

$$\beta=\beta_{\max}\exp(\theta)\big/(\exp(\theta)+1)\,,$$

and then propose a normal random walk on the θ's, $\theta'\sim\mathcal{N}(\theta^{(t)},\tau^2)$. For k, we use instead a uniform proposal on the $2r$ neighbours of $k^{(t)}$, namely a uniform on $\{k^{(t)}-r,\ldots,k^{(t)}-1,k^{(t)}+1,\ldots k^{(t)}+r\}\bigcap\{1,\ldots,K\}$. This proposal distribution with pdf $Q_r(k,\cdot)$, with $k'\sim Q_r(k^{(t-1)},\cdot)$, thus depends on a parameter $r\in\{1,\ldots,K\}$ that needs to be calibrated so as to aim at optimal acceptance rates, as does τ^2. The acceptance probability in the Metropolis–Hastings algorithm is thus

$$\begin{aligned}\rho \;=\; & \frac{f(\mathbf{y}|\mathbf{X},\beta',k')\pi(\beta',k')\Big/Q_r(k^{(t-1)},k')}{f(\mathbf{y}|\mathbf{X},\beta^{(t-1)},k^{(t-1)})\pi(\beta^{(t-1)},k^{(t-1)})\Big/Q_r(k',k^{(t-1)})}\times\\ & \frac{\exp(\theta')\big/(1+\exp(\theta'))^2}{\exp(\theta^{(t-1)})\big/(1+\exp(\theta^{(t-1)}))^2}\,,\end{aligned}$$

where the second ratio is the ratio of the Jacobians due to the reparameterisation. Note that this algorithm is a formal reversible jump MCMC (Green, 1995) in disguise since changing the parameter k means changing the k-nearest-neighbour model, even though the dimension of β does not change.

Once the Metropolis–Hastings algorithm has produced a satisfactory sequence of (β,k)'s, the Bayesian prediction for an unobserved class y_{n+1} associated with x_{n+1} is derived from (6). In fact, if we use a $0-1$ loss function (Robert, 2001) for predicting y_{n+1}, namely

$$\mathrm{L}(\hat{y}_{n+1},y_{n+1})=\mathbb{I}_{\hat{y}_{n+1}\neq y_{n+1}}\,,$$

the Bayes estimator $\hat{y}^{\pi}_{n+1}$ is then the most probable class g according to the posterior predictive (6). The associated measure of uncertainty is then the posterior expected loss, $\mathbb{P}(y_{n+1}=g|x_{n+1},\mathbf{y},\mathbf{X})$.

Explicit calculation of (6) is obviously impossible and this distribution must be approximated from the MCMC chain $\{(\beta,k)^{(1)},\ldots,(\beta,k)^{(M)}\}$ simulated above, namely by

$$M^{-1}\sum_{i=1}^{M}\mathbb{P}\left(y_{n+1}=g|x_{n+1},\mathbf{y},\mathbf{X},(\beta,k)^{(i)}\right)\,. \tag{7}$$

Unfortunately, since (3) involves the intractable constant $Z(\beta,k)$, the above schemes cannot be implemented as such and we need to replace f with a more manageable target. We proceed below through three different approaches that try to overcome this difficulty, postponing the comparison till Section 3.5.

3.2 Pseudo-likelihood approximation

A first solution, dating back to Besag (1974), is to replace the true joint distribution with the pseudo-likelihood, defined as

$$\hat{f}(\mathbf{y}|\mathbf{X},\beta,k) = \prod_{i=1}^{n} \frac{\exp\left\{\beta/k\left(\sum_{\ell\sim_k i}\delta_{y_i}(y_l)+\sum_{i\sim_k\ell}\delta_{y_l}(y_i)\right)\right\}}{\sum_{g=1}^{2}\exp\left\{\beta/k\left(\sum_{\ell\sim_k i}\delta_{g}(y_l)+\sum_{i\sim_k\ell}\delta_{y_l}(g)\right)\right\}} \tag{8}$$

and made up of the product of the (true) conditional distributions associated with (3). The true posterior distribution $\pi(\beta,k|\mathbf{y},\mathbf{X})$ is then replaced with

$$\hat{\pi}(\beta,k|\mathbf{y},\mathbf{X}) \propto \hat{f}(\mathbf{y}|\mathbf{X},\beta,k)\pi(\beta,k)\,,$$

and used as such in all steps of the MCMC algorithm drafted above. The predictive distribution $\mathbb{P}(y_{n+1}=g|x_{n+1},\mathbf{y},\mathbf{X})$ is correspondingly approximated by (7), based on the pseudo-sample thus produced.

While this replacement of the true distribution with the pseudo-likelihood approximation induces a bias in the estimation of (k,β) and in the predictive performances of the Bayes procedure, it has been intensively used in the past, if only because of its availability and simplicity. For instance, Holmes and Adams (2003) built their pseudo-joint distribution on such a product (with the difficulty that the components of the product were not true conditionals). As noted in Friel and Pettitt (2004), pseudo-likelihood estimation can be very misleading and we will describe its performance in more detail in Section 3.5. (To the best of our knowledge, this Bayesian evaluation has not been conducted before.)

As illustrated on Figure 2 for Ripley's benchmark data, the random walk Metropolis–Hastings algorithm detailed above performs satisfactorily on the pseudo-likelihood approximation, even though the mixing is slow (cycles can be spotted on the bottom left graph). On that dataset, the pseudo-maximum–i.e., the maximum of (8)–is achieved for $\hat{k}=53$ and $\hat{\beta}=2.28$. If we use the last $10,000$ iterations of this MCMC run, the prediction performance of (7) is such that the error rate on the testing set of 1000 points is equal to 8.7%. Figure 2 also indicates how limited the information about k is.

3.3 Path sampling

A now standard approach to the estimation of normalising constants is *path sampling*, described in Gelman and Meng (1998) (see also Chen et al., 2000), and derived from the Ogata

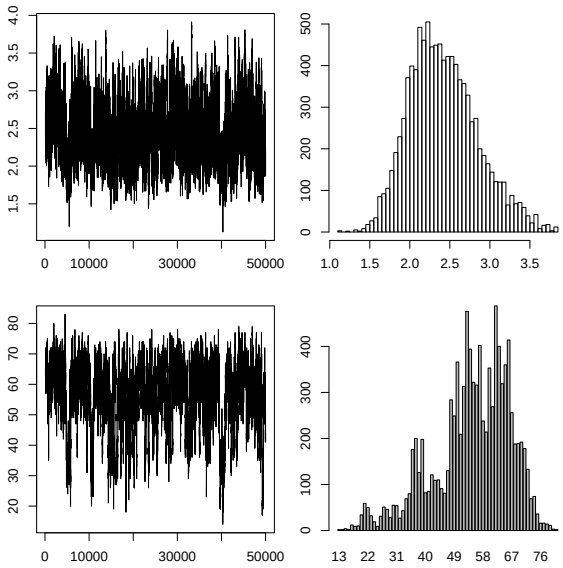

Figure 2: Output of a random walk Metropolis–Hastings algorithm based on the pseudo-likelihood approximation of the normalising constant for $50,000$ iterations, with a $40,000$ iteration burn-in stage, and $\tau^2 = .05$, $r = 3$. *(top)* sequence and marginal histogram for β and *(bottom)* sequence and marginal barplot for k.

(1989) method, in which the ratio of two normalising constants, $Z(\beta',k)/Z(\beta,k)$, can be decomposed as an integral to be approximated by Monte Carlo techniques.

The basic derivation of the path sampling approximation is that, if $S(\mathbf{y})$ denotes the sum

$$S(\mathbf{y}) = \sum_i \sum_{\ell \sim_k i} \delta_{y_i}(y_l)/k\,,$$

then

$$Z(\beta,k) = \sum_{\mathbf{y}} \exp\left[\beta S(\mathbf{y})\right]$$

and

$$\begin{aligned}\frac{\partial Z(\beta,k)}{\partial \beta} &= \sum_{\mathbf{y}} S(\mathbf{y}) \exp[\beta S(\mathbf{y})] \\ &= Z(\beta,k) \sum_{\mathbf{y}} S(\mathbf{y}) \exp(\beta S(\mathbf{y}))\big/ Z(\beta,k) \\ &= Z(\beta,k)\, \mathbb{E}_\beta[S(\mathbf{y})]\,.\end{aligned}$$

Therefore, the ratio $Z(\beta,k)/Z(\beta',k)$ can be derived from an integral, since

$$\log\left\{Z(\beta,k)/Z(\beta',k)\right\} = \int_\beta^{\beta'} \mathbb{E}_{u,k}[S(\mathbf{y})]\,\mathrm{d}u\,,$$

easily evaluated by a numerical approximation.

The practical drawback with this approach is that each time a new ratio is to be computed, that is, at each step of a hybrid Gibbs scheme or of a Metropolis–Hastings proposal, an approximation of the above integral needs to be produced. A further step is thus necessary to use path sampling: we approximate only once the function $Z(\beta,k)$ for each value of k and for a few selected values of β, and later we use numerical interpolation to extend the function to other values of β. The function $Z(\beta,k)$ being very smooth, the degree of additional approximation is quite limited. Given that this approximation is only to be computed once, the resulting Metropolis-Hastings algorithm is very fast, as well as being efficient if enough care is taken with the approximation. (We stress however that the computational cost required to produce those approximations is fairly high.)

We illustrate this approximation using Ripley's benchmark dataset. Figure 3 provides the approximated expectations $\mathbb{E}_{\beta,k}[S(\mathbf{y})]$ for a range of values of β and for two values of k. Within the expectation, the $\mathbf{y}$'s are simulated using a systematic scan Gibbs sampler, because using the perfect sampling scheme elaborated below in Section 3.4 makes little sense when only one expectation needs to be computed. As seen from this comparative graph, when β is small, the Gibbs sampler gives good mixing performances, while, for larger values, it has difficulty in converging, as illustrated by the poor fit on the right-hand plot.

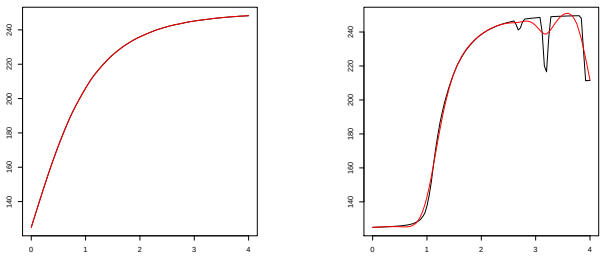

Figure 3: Approximation of the expectation $\mathbb{E}_{\beta,k}[S(\mathbf{y})]$ for Ripley's benchmark, for $k = 1$ *(left)* and $k = 125$ *(right)* (10^4 iterations with 500 burn-in steps for each value of (β, k)). On these graphs, the black curve is the linear interpolation of the expectation and the red curve is a spline interpolation of order 2.

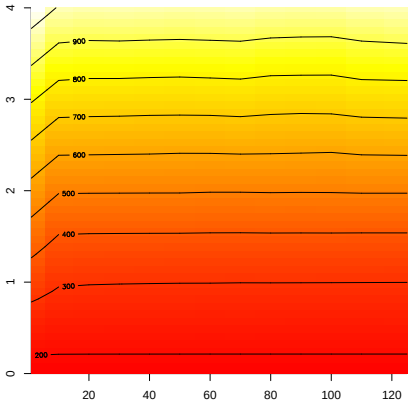

Figure 4: Approximation of the normalising constant $Z(\beta, k)$ for Ripley's dataset where the β's are equally spaced between 0 and 4, and $k = 1, 10, 20, \ldots, 110, 125$ (based on 10^4 Monte Carlo iterations with 500 burn-in steps, and bilinear interpolation).

For the approximation of $Z(\beta, k)$, we use the fact that $\mathbb{E}_{\beta,k}[S(\mathbf{y})]$ is known when $\beta = 0$, namely $\mathbb{E}_{0,k}[S(\mathbf{y})] = n/2$. We can thus represent $\log\{Z(\beta, k)\}$ as

$$n \log 2 + \int_0^\beta \mathbb{E}_{u,k}[S(\mathbf{y})] \, \mathrm{d}u$$

and use numerical integration to approximate the integral. As shown on Figure 4, which uses a bilinear interpolation based on a 50×12 grid of values of (β, k), the approximated constant $Z(\beta, k)$ is mainly constant in k.

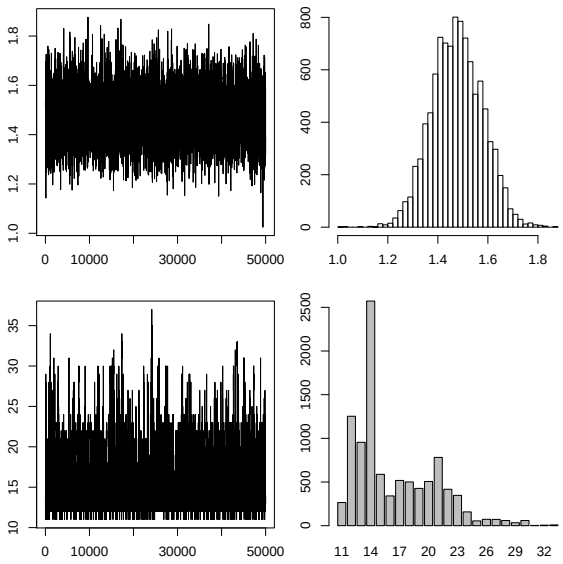

Figure 5: Output of a random walk Metropolis–Hastings algorithm based on the path sampling approximation of the normalising constant for $50,000$ iterations, with a $40,000$ iteration burn-in stage and $\tau^2 = .05$, $r = 3$. *(top)* sequence and marginal histogram for β and *(bottom)* sequence and marginal barplot for k.

Once $Z(\beta, k)$ is approximated, we can use the genuine MCMC algorithm of Section 3.1 fairly easily, the main cost of this approach being thus in the approximation of $Z(\beta, k)$. Figure 5 illustrates the output of the MCMC sampler for Ripley's benchmark, to be compared with Figure 2. A first item of interest is that the chain mixes much more rapidly than its pseudo-likelihood counterpart. A more important point is that the range and shape of the approximations to both marginal posterior distributions differ widely between the two methods, a feature discussed in Section 3.5. When this output of the MCMC sampler is used for prediction purposes in (7), the error rate for Ripley's test set is equal to 8.5%.

3.4 Perfect sampling implementation

A completely different approach to handling missing normalising constants has been developed recently by Møller et al. (2006) and is based on an auxiliary variable idea. If we introduce an auxiliary variable $\mathbf{z}$ on the same state space as $\mathbf{y}$, with arbitrary conditional density $g(\mathbf{z}|\beta, k, \mathbf{y})$, and if we consider the joint posterior

$$\pi(\beta, k, \mathbf{z}|\mathbf{y}) \propto \pi(\beta, k, \mathbf{z}, \mathbf{y}) = g(\mathbf{z}|\beta, k, \mathbf{y}) \times f(\mathbf{y}|\beta, k) \times \pi(\beta, k)\,,$$

then simulating $(\beta, k, \mathbf{z})$ from this posterior is equivalent to simulating (β, k) from the original posterior since $\mathbf{z}$ integrates out. If we now run a Metropolis-Hastings algorithm on this augmented scheme, with q_1 an arbitrary proposal density on (β, k) and with

$$q_2(\beta', k', \mathbf{z}'|\beta, k, \mathbf{z}) = q_1(\beta', k'|\beta, k, \mathbf{y}) f(\mathbf{z}'|\beta', k')\,,$$

as the joint proposal on $(\beta, k, \mathbf{z})$ (i.e., simulating $\mathbf{z}$ directly from the likelihood), the Metropolis-Hastings ratio associated with q_2 is

$$\left(\frac{Z(\beta,k)}{Z(\beta',k)}\right)\left(\frac{\exp\left(\beta' S(\mathbf{y})/k'\right)\pi(\beta',k')}{\exp\left(\beta S(\mathbf{y})/k\right)\pi(\beta,k)}\right)\left(\frac{g(\mathbf{z}'|\beta',k',\mathbf{y})}{g(\mathbf{z}|\beta,k,\mathbf{y})}\right)$$
$$\times\left(\frac{q_1(\beta,k|\beta',k',\mathbf{y})\exp\left(\beta S(\mathbf{z})/k\right)}{q_1(\beta',k'|\beta,k,\mathbf{y})\exp\left(\beta' S(\mathbf{z})/k'\right)}\right)\left(\frac{Z(\beta',k')}{Z(\beta,k)}\right),$$

which means that the constants $Z(\beta, k)$ and $Z(\beta', k')$ cancel out. The method of Møller et al. (2006) can thus be calibrated by the choice of the artificial target $g(\mathbf{z}|\beta, k, \mathbf{y})$ on the auxiliary variable $\mathbf{z}$, and the authors advocate the choice

$$g(\mathbf{z}|\beta, k, \mathbf{y}) = \exp\left(\hat{\beta} S(\mathbf{z})/\hat{k}\right)/Z(\hat{\beta}, \hat{k}),$$

as reasonable, where $(\hat{\beta}, \hat{k})$ is a preliminary estimate, such as the maximum pseudo-likelihood estimate. While we follow this recommendation, we stress that the choice of $(\hat{\beta}, \hat{k})$ is paramount for good performance of the algorithm, as explained below.

Obviously, this approach also has a major drawback, namely that the auxiliary variable $\mathbf{z}$ must be simulated from the distribution $f(\mathbf{z}|\beta, k)$ itself. However, there have been many developments in the simulation of Ising models, from Besag (1974) to Møller and Waagepetersen (2003), and the particular case $G = 2$ allows for exact simulation of $f(\mathbf{z}|\beta, k)$ using perfect sampling. We refer the reader to Häggström (2002), Møller (2003), Møller and Waagepetersen (2003) and Robert and Casella (2004, Chapter 13) for details of this simulation technique and for a discussion of its limitations. Without entering into technical details, we comment that, in the case of model (3) with $G = 2$, there also exists a monotone implementation of the Gibbs sampler that allows for a practical implementation of the perfect sampler (Kendall and Møller, 2000; Berthelsen and Møller, 2003). More precisely, we can use a coupling from the past strategy (Propp and Wilson, 1998): in this setting, starting from the saturated situations in which the components of $\mathbf{z}$ are either all equal to 1 or all equal to 2, it is sufficient to monitor both associated chains further and further in the past until they coalesce by time 0. The sandwiching property of Kendall and Møller (2000) and the monotonicity of the Gibbs sampler ensure that all other chains associated with arbitrary starting values for $\mathbf{z}$ will also have coalesced by then. The only difficulty with this perfect sampler is the phase-transition phenomenon, which means that, for very large values of β, the convergence performance of the coupling from the past sampler deteriorates quite rapidly, a fact also noted in Møller et al. (2006) for the Ising model. We overcome this difficulty by using an additional accept-reject step based on smaller values of β that avoids this explosion in the computational time.

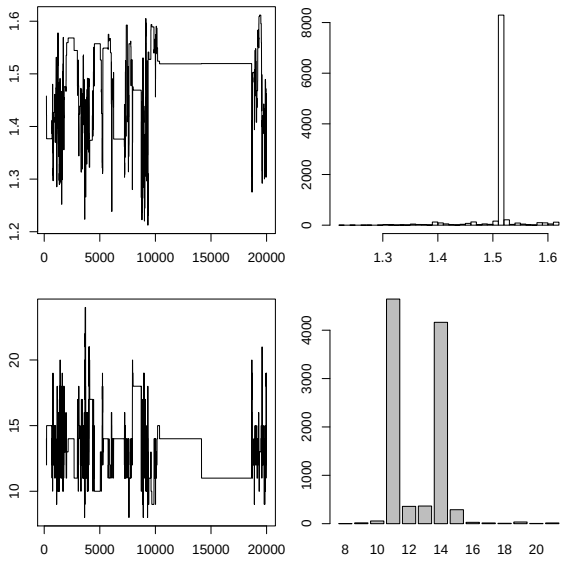

Figure 6: Output of a random walk Metropolis–Hastings algorithm based on the perfect sampling elimination of the normalising constant for a pseudo-likelihood plug-in estimate $(\hat{k}, \hat{\beta}) = (53, 2.28)$ and $20,000$ iterations, with a $10,000$ burn-in stage and $\tau^2 = .05$, $r = 3$: *(top)* sequence and marginal histogram for β and *(bottom)* sequence and marginal barplot for k.

As shown on Figure 6, a poor choice for $(\hat{\beta}, \hat{k})$ leads to very unsatisfactory performance with the algorithm. Starting from the pseudo-likelihood estimate and using this very value for the plug-in value $(\hat{\beta}, \hat{k})$, we obtain a Markov chain with a very low energy and a very high rejection rate. Using the estimate $(\hat{k}, \hat{\beta}) = (13, 1.45)$ resulting from this poor run does however improve considerably the performance of the algorithm, as shown by Figure 7. In this setting, the predictive error rate on the test dataset is equal to 0.084.

3.5 Evaluation of the pseudo-likelihood approximation

Given that the above alternatives can all be implemented for small values of n, it is of direct interest to compare them in order to evaluate the effect of the pseudo-likelihood approximation. As demonstrated in the previous section, using Ripley's benchmark with a training set of 250 points, we are indeed able to run a perfect sampler over the range of possible β's and this implementation gives a sampler in which the only approximation is due to running an MCMC sampler (a feature common to all three versions).

When comparing (for the same dataset), histograms of simulated β's conditional or unconditional on k's show (in Figures 8 and 9) gross misrepresentation of the samples produced

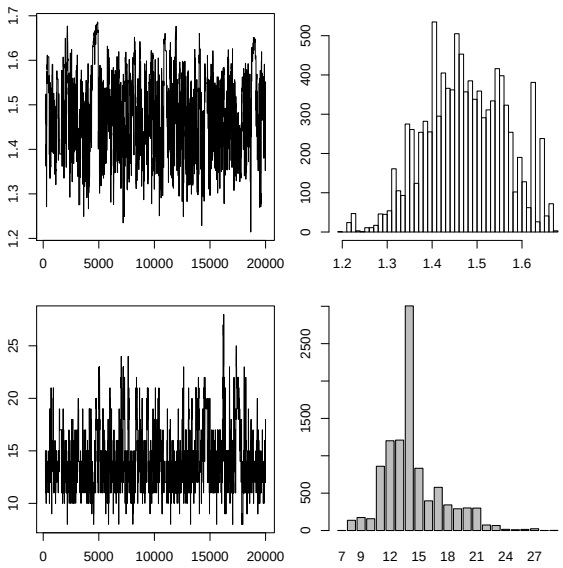

Figure 7: Output of a random walk Metropolis–Hastings algorithm based on the perfect sampling elimination of the normalising constant for a plug-in estimate $(\hat{k}, \hat{\beta}) = (13, 1.45)$ and $20,000$ iterations, with a $10,000$ burn-in stage and $\tau^2 = .05$, $r = 3$: *(top)* sequence and marginal histogram for β and *(bottom)* sequence and marginal barplot for k.

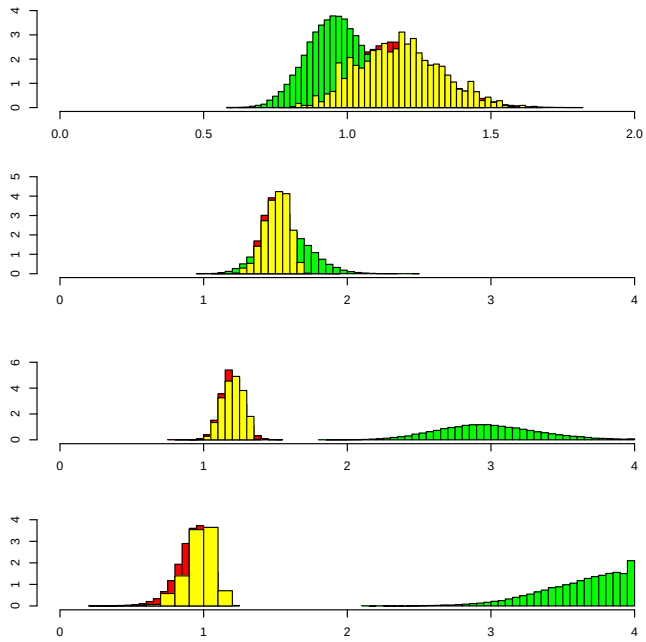

Figure 8: Comparison of the approximations to the posterior distribution of β based on the pseudo *(red)*, the path *(green)* and the perfect *(yellow)* schemes for Ripley's benchmark and $k = 1, 10, 70, 125$, for $20,000$ iterations and $10,000$ burn-in.

by the pseudo-likelihood approximation. (The comparison for a fixed value of k was obtained directly by setting k to a fixed value in all three approaches and running the corresponding MCMC algorithms.) It could of course be argued that the defect lies with the path sampling evaluation of the constant, but this approach strongly coincides with the perfect sampling implementation, as showed on both figures. There is thus a fundamental discrepancy in using the pseudo-likelihood approximation; in other words, the pseudo-likelihood approximation defines a clearly different posterior distribution on (β, k).

As exhibited on Figure 8, the larger k is, the worse is this discrepancy, whereas Figure 9 shows that both β and k are significantly overestimated by the pseudo-likelihood approximation. (It is quite natural to find such a correlation between β and k when we realise that the likelihood mostly depends on β/k.) We can also note that the correspondence between path and perfect approximations is not absolute in the case of k, a difference that may be attributed to slower convergence in one or both samplers.

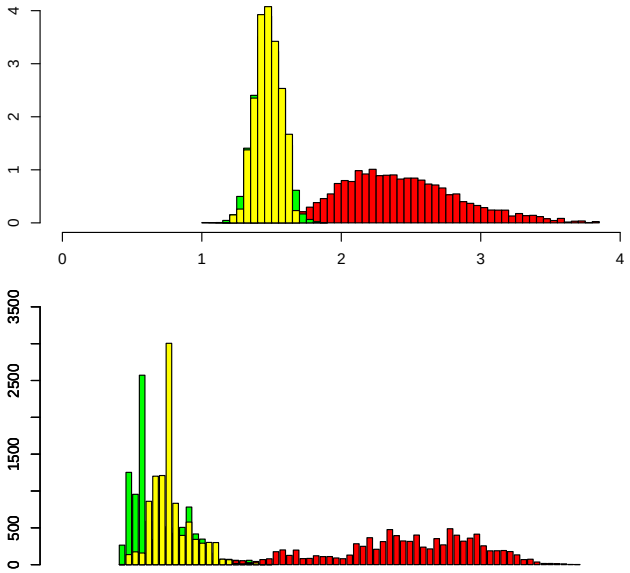

Figure 9: Comparison of posterior distributions of β *(top)* and k *(bottom)* as represented in Figure 2 for the pseudo-likelihood approximation, in Figure 5 for the path sampling approximation and in Figure 7 for the perfect sampling approximation.

4 Illustration on Pima Indian diabetes data

This dataset is borrowed from the MASS library of R. It consists in the records of 532 Pima Indian women who were tested by the U.S. National Institute of Diabetes and Digestive and Kidney Diseases for diabetes. Seven quantitative covariates were recorded, along with the presence or absence of diabetes. The data are split at random into a training set of 200 women, including 109 diagnosed with diabetes, and a test set of the remaining 322 women, including 68 diagnosed with diabetes. The performance for various values of k on the test dataset is given in Table 2. If we use a standard leave-one-out cross-validation for selecting k (using only the training dataset), then there are 10 consecutive values of k leading to the same error rate, namely the range 57–66.

k	Misclassification error rate
1	0.316
3	0.229
15	0.226
31	0.211
57	0.205
66	0.208

Table 2: Performance of k-nearest-neighbour methods on the Pima Indian test dataset.

The maximum of the pseudo-likelihood is obtained for $\hat{k} = 50$ and $\hat{\beta} = 1.338$. A first run of our algorithm with $10,000$ iterations was done using these values as reference values, leading to slow mixing and the new reference values $(\hat{k}, \hat{\beta}) = (40, 1.15)$ and $\tau^2 = .05$. A second run of $20,000$ iterations leads to the results illustrated in Figure 10. While the overall mixing behaviour is quite acceptable, we still observe a phenomenon already pointed out in Møller et al. (2006), namely the occurrence of long breaks in which no new value is accepted. These authors attributed these phenomena to possible discrepancies between the likelihood and the auxiliary variable distribution.

Note that the simulated values of k tend to avoid the region found by the cross-validation procedure. One possible reason for this discrepancy is that, as noted in Section 2.2, the likelihood for our joint model is not directly equivalent to the k-nearest-neighbour objective function, since mutual neighbours are weighted twice as much as single neighbours in this likelihood.

Out of $9,000$ final iterations, the predictive error is 0.209, quite in line with the k-nearest-neighbour solution in Table 2.

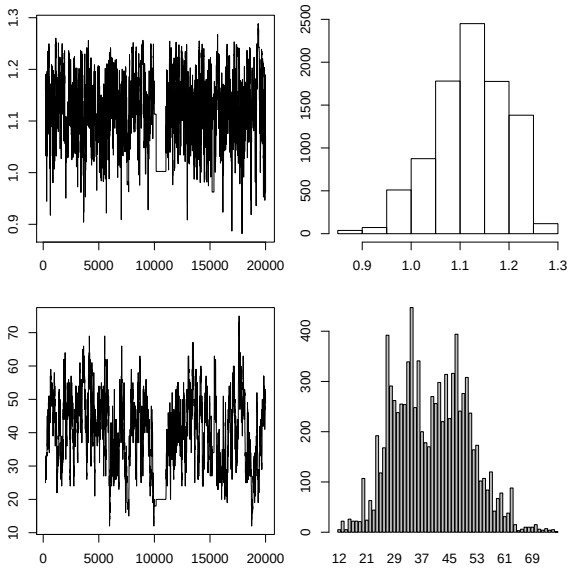

Figure 10: Pima Indian diabetes study using the perfect sampling aproximation with the estimated pluggin $\hat{\beta} = 1.15$ and $\hat{k} = 40$, and $\tau^2 = .05$, $r = 3$, $\beta_{\max} = 1.5$, and $K = 68$, based on $20,000$ iterations.

Acknowledgements

The authors are grateful to Gilles Celeux for his numerous and insightful comments on the different perspectives offered by this probabilistic reassessment. Both first authors are also grateful to the Department of Statistics of the University of Glasgow for its warm welcome during various visits related to this work. This work was supported in part by the IST Programme of the European Community, under the PASCAL Network of Excellence, ST-2002-506778.

References

Berthelsen, K. and Møller, J. (2003). Likelihood and non-parametric Bayesian MCMC inference for spatial point processes based on perfect simulation and path sampling. *Scandinavian J. Statist.*, 30:549–564.

Besag, J. (1974). Spatial interaction and the statistical analysis of lattice systems (with discussion). *J. Roy. Statist. Soc. Ser. B*, 36:192–236.

Besag, J. and Kooperberg, C. (1995). On conditional and intrinsic autoregressions. *Biometrika*, 82(4):733–746.

Bühlmann, P. (2004). Bagging, boosting and ensemble methods. In *Handbook of Computational Statistics*, pages 877–907. Springer, Berlin.

Bühlmann, P. and Yu, B. (2002). Analyzing bagging. *Ann. Statist.*, 30(4):927–961.

Bühlmann, P. and Yu, B. (2003). Boosting with the L_2 loss: regression and classification. *J. Amer. Statist. Assoc.*, 98(462):324–339.

Chen, M., Shao, Q., and Ibrahim, J. (2000). *Monte Carlo Methods in Bayesian Computation*. Springer, New York.

Cressie, N. A. C. (1993). *Statistics for Spatial Data*. Wiley Series in Probability and Mathematical Statistics: Applied Probability and Statistics. John Wiley & Sons Inc., New York.

Devroye, L., Györfi, L., and Lugosi, G. (1996). *A Probabilistic Theory of Pattern Recognition*, volume 31 of *Applications of Mathematics (New York)*. Springer-Verlag, New York.

Freund, Y. and Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. System Sci.*, 55(1, part 2):119–139. Second Annual European Conference on Computational Learning Theory (EuroCOLT '95) (Barcelona, 1995).

Friel, N. and Pettitt, A. N. (2004). Likelihood estimation and inference for the autologistic model. *J. Comput. Graph. Statist.*, 13(1):232–246.

Gelman, A. and Meng, X.-L. (1998). Simulating normalizing constants: from importance sampling to bridge sampling to path sampling. *Statist. Sci.*, 13(2):163–185.

Green, P. J. (1995). Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, 82(4):711–732.

Häggström, O. (2002). *Finite Markov Chains and Algorithmic Applications*, volume 52 of *Student Texts*. London Mathematical Society.

Hastie, T., Tibshirani, R., and Friedman, J. (2001). *The Elements of Statistical Learning*. Springer Series in Statistics. Springer-Verlag, New York.

Holmes, C. C. and Adams, N. M. (2002). A probabilistic nearest neighbour method for statistical pattern recognition. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, 64(2):295–306.

Holmes, C. C. and Adams, N. M. (2003). Likelihood inference in nearest-neighbour classification models. *Biometrika*, 90(1):99–112.

Kendall, W. and Møller, J. (2000). Perfect simulation using dominating processes on ordered spaces, with application to locally stable point processes. *Advances in Applied Probability*, 32:844–865.

McLachlan, G. J. (1992). *Discriminant Analysis and Statistical Pattern Recognition*. Wiley Series in Probability and Mathematical Statistics: Applied Probability and Statistics. John Wiley & Sons Inc., New York.

Møller, J. (2003). *Spatial Statistics and Computational Methods*, volume 173 of *Lecture Notes in Statistics*. Springer-Verlag, New York.

Møller, J., Pettitt, A., Reeves, R., and Berthelsen, K. (2006). An efficient Markov chain Monte Carlo method for distributions with intractable normalising constants. *Biometrika*, 93:451–458.

Møller, J. and Waagepetersen, R. (2003). *Statistical Inference and Simulation for Spatial Point Processes*. Chapman and Hall/CRC, Boca Raton, FL.

Ogata, Y. (1989). A Monte Carlo method for high-dimensional integration. *Numer. Math.*, 55(2):137–157.

Propp, J. and Wilson, D. (1998). Coupling from the past: a user's guide. In *Microsurveys in discrete probability (Princeton, NJ, 1997)*, volume 41 of *DIMACS Ser. Discrete Math. Theoret. Comput. Sci.*, pages 181–192. Amer. Math. Soc., Providence, RI.

Ripley, B. D. (1994). Neural networks and related methods for classification (with discussion). *J. Roy. Statist. Soc. Ser. B*, 56(3):409–456.

Ripley, B. D. (1996). *Pattern Recognition and Neural Networks*. Cambridge University Press, Cambridge.

Robert, C. (2001). *The Bayesian Choice*. Springer Texts in Statistics. Springer-Verlag, New York, second edition.

Robert, C. P. and Casella, G. (2004). *Monte Carlo Statistical Methods*. Springer Texts in Statistics. Springer-Verlag, New York, second edition.

Zhang, T. and Yu, B. (2005). Boosting with early stopping: convergence and consistency. *Ann. Statist.*, 33(4):1538–1579.

Unité de recherche INRIA Futurs
Parc Club Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 ORSAY Cedex (France)

Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)
Unité de recherche INRIA Sophia Antipolis : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399