A Bayesian Reassessment of Nearest-Neighbor Classification

Lionel CUCALA, Jean-Michel MARIN, Christian P. ROBERT, and D. M. TITTERINGTON

The k-nearest-neighbor (knn) procedure is a well-known deterministic method used in supervised classification. This article proposes a reassessment of this approach as a statistical technique derived from a proper probabilistic model; in particular, we modify the assessment found in Holmes and Adams, and evaluated by Manocha and Girolami, where the underlying probabilistic model is not completely well defined. Once provided with a clear probabilistic basis for the knn procedure, we derive computational tools for Bayesian inference on the parameters of the corresponding model. In particular, we assess the difficulties inherent to both pseudo-likelihood and path sampling approximations of an intractable normalizing constant. We implement a correct MCMC sampler based on perfect sampling. When perfect sampling is not available, we use instead a Gibbs sampling approximation. Illustrations of the performance of the corresponding Bayesian classifier are provided for benchmark datasets, demonstrating in particular the limitations of the pseudo-likelihood approximation in this set up.

KEY WORDS: Boltzmann model; Compatible conditionals; Normalizing constant; Path sampling; Markov chain Monte Carlo algorithm; Perfect sampling; Pseudo-likelihood.

1. INTRODUCTION

1.1 Deterministic versus Statistical Classification

Supervised classification has long been used in both Machine Learning and Statistics to infer about the functional connection between a group of covariates (or explanatory variables) and an indicator (or class) variable (e.g., McLachlan 1992; Ripley 1994, 1996; Devroyea, Györfi, and Lugosi 1996; Hastie, Tibshirani, and Friedman 2001). For instance, the method of *boosting* (Freund and Schapire, 1997) has been developed for this very purpose by the Machine Learning community and later assessed and extended by statisticians (Hastie et al. 2001; Bühlmann and Yu 2002, 2003; Bühlmann 2004; Zhang and Yu 2005).

The k-nearest-neighbor (knn) method is a well established and straightforward supervised classification technique (Ripley 1994, 1996). While providing an instrument for classifying points into two or more classes, it lacks a corresponding assessment of its classification error. Although alternative techniques like boosting offer this assessment, it is obviously of interest to provide the original knn method with this additional feature. In contrast, statistical classification methods based on (e.g., mixtures of distributions), do provide an error assessment along with the most likely classification, but this perspective obviously requires a probabilistic framework. Holmes and Adams (2002) proposed a first Bayesian analysis of the knn method based on these premises. In a separate article, Holmes and Adams (2003) conducted a likelihood analysis of another model based on autologistic representations, in particular for selecting the value of k. Although we adopt a Bayesian approach, our article differs from Holmes and Adams (2002) in two important respects. First, we define a global probabilistic model that encapsulates the knn method, rather than working with incompatible conditionals, and, second, we derive an operational simulation technique for our model based either on perfect sampling or on a Gibbs approximation which allows for a reassessment of the pseudo-likelihood approximation often used in those settings.

1.2 The Original knn Method

Given a training set of n individuals allocated each to one of G classes, the classical knn procedure is a method that allocates new individuals to the most common class in their neighborhood among the training set. Formally, based on a training dataset $((y_i, \mathbf{x}_i))^n_{i=1}$, where $y_i \in \{1, \ldots, G\}$ denotes the class label of the ith point and $\mathbf{x}_i \in \mathrm{R}^p$ is a vector of covariates, an unobserved class y_{n+1} associated with a set of covariates $\mathbf{x}_{n+1}$ is estimated by the majority class among the k nearest neighbors of $\mathbf{x}_{n+1}$ in the training set $(\mathbf{x}_i)^n_{i=1}$. The neighborhood is defined in the space of the covariates $\mathbf{x}_i$, namely

$$\mathcal{N}^k_{n+1} = \left\{ 1 \le i \le n;\ d(\mathbf{x}_i, \mathbf{x}_{n+1}) \le d(\cdot, \mathbf{x}_{n+1})_{(k)} \right\},$$

where $d(\cdot, \mathbf{x}_{n+1})$ denotes the vector of distances to $\mathbf{x}_{n+1}$, and $d(\cdot, \mathbf{x}_{n+1})_{(k)}$ denotes its kth order statistic. The original knn method usually uses the Euclidean norm, even though the Mahalanobis distance would be more appropriate in that it rescales the covariates. (Ties are eliminated by decreasing the number k of neighbors until the problem disappears.) When some covariates are categorical, other distances can be used, as in the R package knncat of Buttrey (1998).

As such, the method is both deterministic, given the training dataset, and not parameterized, even though the choice of k is relevant to the performance of the method: Usually, k is selected via cross-validation, as the value that minimizes the cross-validation error rate. In contrast to cluster analysis, the number G of classes in the knn procedure is fixed and given by the training set: the introduction of classes that are not observed in the training set has no effect on the future allocations.

To illustrate the original method and to compare it with our approach, we use throughout a benchmark taken from Ripley

Lionel Cucala Dr., INRIA Saclay, Projet Select, Université Paris-Sud, Laboratoiré de Mathématiques (Bat. 425), 91400 Orsay, France (E-mail: *cucala@cict.fr*). Jean-Michel Marin Pr., INRIA Saclay, Projet Select, Université Paris-Sud Laboratoiré de Mathématiques (Bat. 425), 91400 Orsay, France (E-mail: *Jean-Michel.Marin@inria.fr*). Christian P. Robert Pr., CEREMADE, Université Paris Dauphine 75775 Paris, France (E-mail: *xian@ceremade.dauphine.fr*). D. M. Titterington Pr., University of Glasgow, Department of Statistics, Glasgow, G12 8QW, UK (E-mail: *mike@stats.gla.ac.uk*). This work was supported in part by the IST Programme of the European Community, under the PASCAL Network of Excellence, ST-2002-506778, and by the Agence Nationale de la Recherche (ANR) through the 2007–2009 project SP Bayes. The authors thank Gilles Celeux for his numerous and insightful comments, as well as to the associate editor and referees for their constructive comments. Both JMM and CPR are grateful to the Department of Statistics of the University of Glasgow for its warm welcome during various visits related to this work.

Journal of the American Statistical Association
March 2009, Vol. 104, No. 485, Theory and Methods
DOI 10.1198/jasa.2009.0125

(1994). It corresponds to a two-class classification in which each class of covariates is simulated from a bivariate normal distribution, both populations being of equal sizes. This dataset is available at *http://www.stats.ox.ac.uk/pub/PRNN*. 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, as shown on Figure 1. The performance of the standard knn method on this dataset is such that the misclassification leave-one-out error rates for $k = 1, 3, 15, 17, 31, 54$ are 15%, 13.4%, 9.5%, 8.7%, 8.4%, and 8, 1%, respectively. The overall misclassification leave-one-out error rate is provided as a function of k in Figure 2. It indicates that this criterion is poorly discriminating in this case. There is little variation for a wide range of k's and 10 values of k achieve the same overall minimum, namely 17, 18, 35, 36, 45, 46, 51, 52, 53, and 54. This range of values is an indicator of potential gains when averaging over k, and it hence calls for a Bayesian perspective.

1.3 Goal and Plan

As presented previously, the knn method is an allocation technique that does not provide an assessment for uncertainty. This essential feature requires a probabilistic framework that relates the class label y_i to both the covariates $\mathbf{x}_i$ *and* the class labels of the neighbors of $\mathbf{x}_i$. Not only does this perspective provide information about the variability of the classification, when compared with 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 neighbors, and to analyze the role of the variables, of the metric used, of the number k of neighbors, and of the number of classes toward achieving higher efficiency. This statistical viewpoint is that of Holmes and Adams (2002, 2003) and we follow suit in this article, with a modification of their original model geared toward a coherent probabilistic model, while providing new developments in computational model estimation. Figure 3 illustrates the appeal of adopting a probabilistic perspective through two graphs for Ripley's benchmark. The graph on the left gives a representation of the predictive probabilities to be in class 1, whereas the graph on the right partitions the square into three zones, namely sure allocation to class 1, sure allocation to class 2, and an uncertainty zone. This partition is obtained by first computing 95% credible intervals for the predictive probabilities and then checking those intervals against the borderline value 0.5. Intervals containing 0.5 are ranked as uncertain.

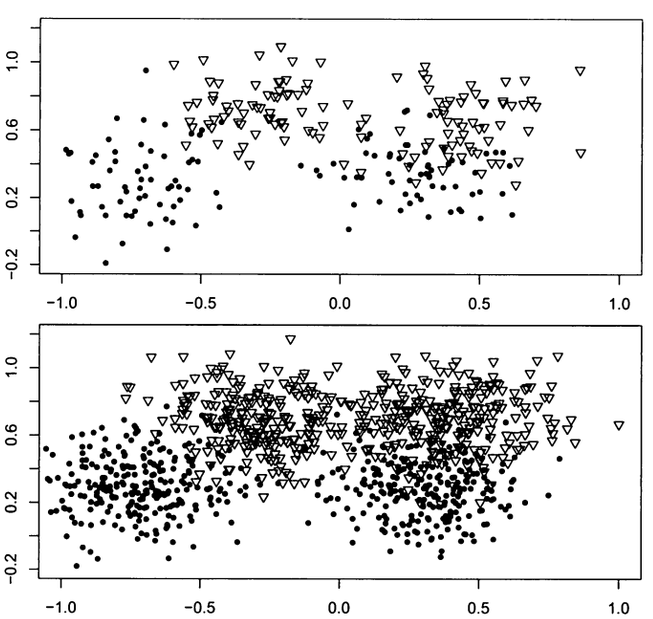

Figure 1. Training (top) and test (bottom) sets for Ripley's benchmark: solid circles correspond to label 1 and unfilled triangles to label 2.

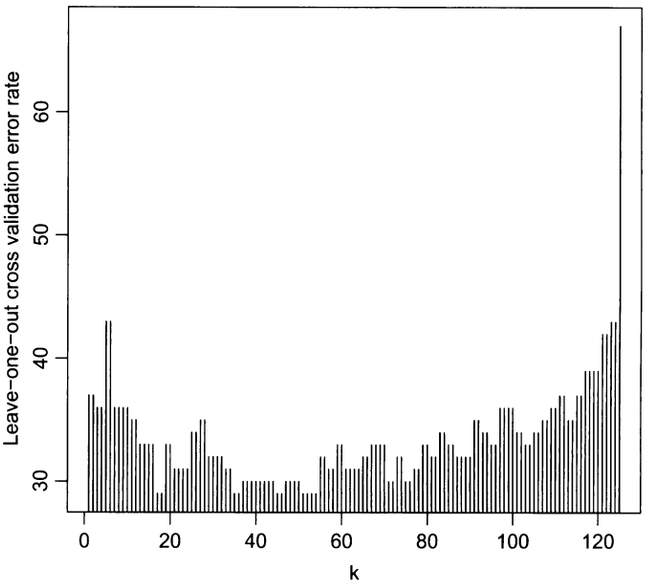

Figure 2. Misclassification leave-one-out error rate of the knn method as a function of k for Ripley's training dataset.

The article is organized as follows. The validity of the new probabilistic knn model is established in Section 2, pointing out the deficiencies of the models advanced by Holmes and Adams (2002, 2003), and we then cover Bayesian inference in this model in Section 3, addressing the specific issue of evaluating the corresponding normalizing constant that is necessary for inferring about k. We take advantage of an exact Markov chain Monte Carlo (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 new benchmarks in Section 4.

2. THE PROBABILISTIC KNN MODEL

2.1 Markov Random Fields

To build a probabilistic structure that reproduces the features of the original knn 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 $\mathbf{x}_i$, for the training dataset. A natural approach is to take advantage of the spatial structure of the problem by using a Markov random field model. Although, as shown later, this is not compatible with a coherent probabilistic setting, we could assume that the full conditional of y_i given $\mathbf{y}_{-i} = (y_1, \ldots, y_{i-1}, y_{i+1}, \ldots, y_n)$ and the

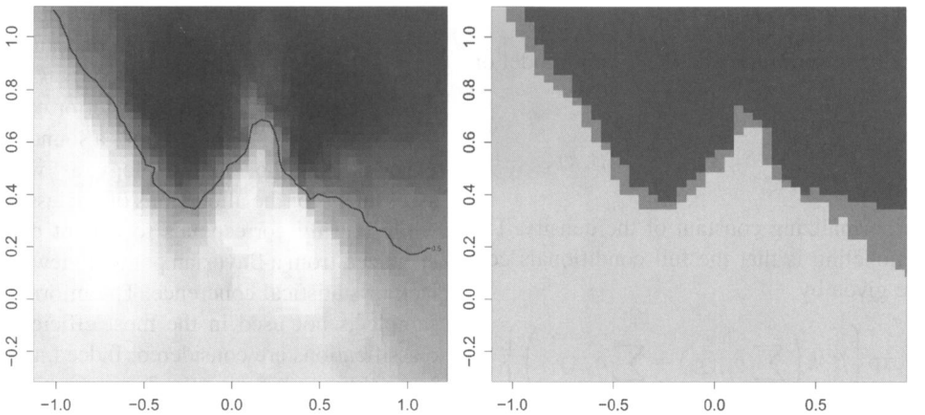

Figure 3. For Ripley's benchmark given in Figure 1, (left) values of the predictive probabilities to be in class 1 when at a given $\mathbf{x}$, ranging from 0 (black) to 1 (white), along with the estimated 0.5 probability separation, and (right) predictive decision regions: dark grey corresponds to sure allocation to class 2, light grey to sure allocation to class 1, and medium grey to the uncertainty zone. (These plots are based on an MCMC sample whose derivation is explained in Section 3.4.)

$\mathbf{x}_i$'s only depends on the k nearest neighbors of $\mathbf{x}_i$ in the training set. The parameterized structure of this conditional can for instance be chosen as a Boltzmann distribution (Møller and Waagepetersen 2003) with potential function $\sum_{\ell\sim_k i}\delta_{yi}(y_\ell)$, where $\ell\sim_k i$ means that the summation is taken over the observations $\mathbf{x}_\ell$ within the k nearest neighbors of $\mathbf{x}_i$, and $\delta_a(b)$ denotes the Dirac function. This function is equal to the number of points from the same class y_i as the point $\mathbf{x}_i$ that are among the k nearest neighbors of $\mathbf{x}_i$. As in Holmes and Adams (2003), the expression for the full conditional is thus

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

where $\beta>0$ and $\mathbf{X}$ is the (p, n) matrix $\{\mathbf{x}_1,\ldots,\mathbf{x}_n\}$ of coordinates for the training set.

In this parameterized model, β is a quantity that is missing from the original knn procedure. Its statistical interpretation is of a degree of uncertainty: $\beta=0$ corresponds to a uniform distribution over all classes, with independence from the neighbors, while $\beta=+\infty$ leads to a point mass distribution at the prevalent class and thus to extreme dependence. The introduction of the parameter k in the denominator makes β dimensionless.

There is, however, a difficulty with this expression in that, for almost all datasets $\mathbf{X}$, there is no joint distribution on $\mathbf{y}=(y_1,\ldots,y_n)$ with full conditionals equal to (1). This happens because the knn system is usually asymmetric: when $\mathbf{x}_i$ is one of the k nearest neighbors of $\mathbf{x}_j$, $\mathbf{x}_j$ is not necessarily one of the k nearest neighbors of $\mathbf{x}_i$. Therefore, the pseudo-conditional distribution (1) will not depend on $\mathbf{x}_j$ while the equivalent for $\mathbf{x}_j$ does depend on $\mathbf{x}_i$: this is obviously impossible in a coherent probabilistic framework (Besag 1974; Cressie 1993)

One way of overcoming this fundamental difficulty is that of Holmes and Adams (2002), who define directly the joint distribution as

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

Unfortunately, the drawbacks to this approach are that first, the function (2) is not properly normalized (a fact overlooked by Holmes and Adams 2002), whereas the necessary normalizing constant is intractable. Second, the full conditionals corresponding to this joint distribution are not given by (1). The first drawback is a common occurrence with Boltzmann models and is dealt with in detail in Section 3. We recall that the most standard approach to this problem is to use pseudo-likelihood following Besag, York, and Mollié (1991), as in Heikkinen and Hogmander (1994) and Hoeting, Madigan, Raftery, and Volinsky (1999), but Section 3.5 shows that this approximation can give poor results. (See Friel, Pettitt, Reeves, and Wit 2005 for a discussion.) The second and more specific drawback implies that (2) cannot be treated as a pseudo-likelihood (Besag 1974; Besag et al. 1991) because the conditional distribution (1) can be associated with no joint distribution.

That (2) misses a normalizing constant can be seen from the special case in which $n=2$, $\mathbf{y}=(y_1, y_2)$, and $G=2$, since

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

which clearly differs from 1 and, more importantly, depends on both β and k. We note that the debate about using a proper joint distribution is reminiscent of the debate between Gaussian conditional autoregressions (CAR) and Gaussian intrinsic autoregressions in Besag and Kooperberg (1995), the latter being associated with no joint distribution.

2.2 A symmetrized Boltzmann Modeling

Given these difficulties, we directly define a joint model on the training set as

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

where $Z(\beta, k)$ is the normalizing constant of the density. The motivation for this modeling is that the full conditionals corresponding to (3) are given by

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

where $i \sim {}_k\ell$ means that the summation is taken over the observations $\mathbf{x}_\ell$ for which $\mathbf{x}_i$ is a k-nearest neighbor. Obviously, these conditionals differ from (1) if only because of the previous impossibility result. The additional term in the potential function corresponds to the observations that are not among the nearest neighbors of $\mathbf{x}_i$ but for which $\mathbf{x}_i$ is a nearest neighbor. In this model, mutual neighbors are weighted twice as much as single neighbors. 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 knn objective function. Note also that the original full conditional (1) is recovered as (4) when the system of neighbors is perfectly symmetric (up to a factor 2).

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 knn model is to reweight the neighborhood sizes by $a_g = p_g n/n_g$. This leads to the modified model

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

This modification is useful in practice when we are dealing with stratified surveys. In the following, however, we assume balanced sampling.

2.3 Predictive Perspective

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

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

where

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

are the numbers of observations in the training dataset from class g among the k nearest neighbors of $\mathbf{x}_{n+1}$ and among the observations for which $\mathbf{x}_{n+1}$ is a k-nearest neighbor, respectively. This predictive distribution can then be processed by considering the joint posterior of (β, k, y_{n+1}) and by deriving the corresponding marginal posterior of y_{n+1}.

Although this model provides a sound statistical basis for the k-nearest-neigbhor methodology as well as an uncertainty assessment for the allocations of unclassified observations, and while it truly corresponds to a joint distribution, (4) can be criticized from a Bayesian point of view in that it suffers from a lack of statistical coherence. The information contained in the sample is not used in the most efficient way when multiple classifications are considered. Indeed, the knn methodology is invariably used in a repeated manner, either jointly on a sample $(\mathbf{x}_{n+1}, \ldots, \mathbf{x}_{n+m})$ or sequentially. Rather than assuming simultaneous dependence within the training sample and independence within the unclassified sample, it would be more sensible to consider the whole collection of points as issuing from a single joint model (3), with some class missing at random. From a Bayesian point of view, addressing jointly 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)—certainly makes sense from a foundational perspective as a correct probabilistic evaluation, and it does provide a better assessment of the uncertainty about the classifications as well as about the parameters.

Unfortunately, this more global perspective is unachievable if only for computational reasons, because 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 distribution of both parameters and classes, especially when m is large. We thus adopt the more ad hoc approach of dealing separately with each unclassified point in the analysis, because this is the only realistic way. This perspective is also justified because in realistic machine learning set ups, the unclassified data $(y_{n+1}, \ldots, y_{n+m})$ mostly occur in a sequential environment with, furthermore, the true value of y_{n+1} being revealed before y_{n+2} is predicted.

In the following sections, we mainly consider the case $G = 2$ as in Holmes and Adams (2003), because we can then conduct a full comparison between different approximation schemes, but we indicate at the end of Section 3.4 how a Gibbs sampling approximation allows for a realistic extension to larger G's, as illustrated in Section 4.

3. BAYESIAN INFERENCE AND THE NORMALIZATION PROBLEM

Given the joint model (3) for $(y_1, \ldots, y_{n+1})$, Bayesian inference can be conducted provided computational difficulties related to the unavailable normalizing constant can be solved. Indeed, as stressed previoiusly, the classification of unclassified points can be based on the marginal predictive of y_{n+1} obtained by integration over the conditional posterior of the parameters, namely, for $g = 1, 2$,

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

where $\pi(\beta, k|\mathbf{y}, \mathbf{X}) \propto f(\mathbf{y}|\mathbf{X}, \beta, k)\pi(\beta, k)$ is the posterior of (β, k) given the training dataset $(\mathbf{y}, \mathbf{X})$. Although other choices of priors are available, we choose for (k, β) a uniform prior on the compact support $\{1, \ldots, K\} \times [0, \beta_{\max}]$. 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, $\min(n_1, n_2)$, whereas the limitation on β, $\beta \leq \beta_{\max}$ is customary in Boltzmann models, because of phase transition 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 4 later by the convergence of the expectation of the number of identical neighbors to k.) The determination of $\beta_{\max}$ is obviously problem-specific and needs to be solved afresh for each new dataset because it depends on the topology of the neighborhood. It is, however, straighforward to implement in that a Gibbs simulation of (3) for different values of β quickly exhibits the "black-or-white" features if β is above the phase transition boundary.

3.1 MCMC Steps

Were the posterior distribution $\pi(\beta, k|\mathbf{y}, \mathbf{X})$ available (up to a normalizing constant), we could design an MCMC algorithm and thus produce an approximate sample from this posterior. However, because of the associated representation (4), the conditional of β is nonstandard and requires hybrid sampling where 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 for k imposes 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 $O(KnG)$, because of the likelihood representation. A faster alternative is to use a hybrid step for both β and k since the full conditional of k is only computed once for one new value of k, modulo the normalizing constant.

An alternative to Gibbs sampling is to use a random walk Metropolis–Hastings algorithm: both β and k are then updated using random walk proposals. Because $\beta \in (0, \beta_{\max})$ is constrained, we consider a logistic reparameterization of β, $\beta = \beta_{\max} \exp(\theta)/(\exp(\theta) + 1)$, and then propose a normal random walk on the θ's, $\theta' \sim \mathrm{N}(\theta^{(t)}, \tau^2)$. For k, we use instead a uniform proposal on the $2r$ neighbors of $k^{(t)}$, namely $\{k^{(t)} - r, \ldots, k^{(t)} - 1, k^{(t)} + 1, \ldots k^{(t)} + r\} \cap \{1, \ldots, K\}$. This proposal distribution with density $Q_r(k, \cdot)$, with $k' \sim Q_r(k^{(t-1)}, \cdot)$, thus depends on a parameter $r \in \{1, \ldots, K\}$ that needs to be calibrated toward optimal acceptance rates, like τ^2. The Metropolis–Hastings acceptance ratio is thus

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

where the second ratio is the ratio of the Jacobians.

Once the Metropolis–Hastings algorithm has produced a sequence of (β, k)'s, the Bayesian prediction for an unobserved class y_{n+1} associated with $\mathbf{x}_{n+1}$ is derived from (6). In fact, if we use a 0–1 loss function (Robert, 2001) for predicting y_{n+1}, $\mathrm{L}(\hat{y}_{n+1}, y_{n+1}) = \mathbb{L}_{\hat{y}_{n+1} \neq y_{n+1}}$, the Bayes' estimator $\hat{y}^{\pi}_{n+1}$ is the most probable class g for the posterior predictive (6). The associated measure of uncertainty is then the posterior expected loss, $\mathbb{P}(y_{n+1} \neq g|\mathbf{x}_{n+1}, \mathbf{y}, \mathbf{X})$.

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

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

Unfortunately, because (3) involves the intractable constant $Z(\beta, k)$, the previous schemes cannot be implemented as such and we need to approximate f. We proceed later 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 the product of the (true) conditionals associated with (3),

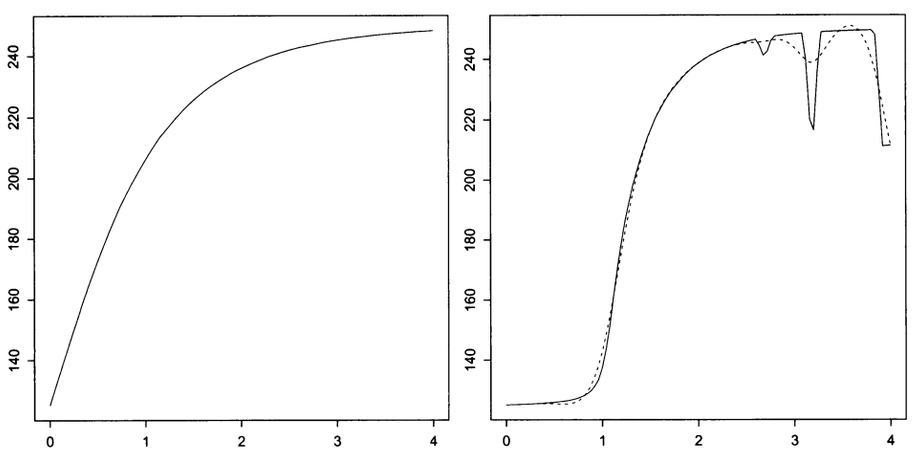

Figure 4. Approximations to the expectation $\mathbb{E}_{\beta, k}[S(\mathbf{y})]$ for Ripley's benchmark, where the β's are equally spaced between 0 and $\beta_{\max} = 4$, and for $k = 1$ (left) and $k = 125$ (right) (10^4 iterations with 500 burn-in steps for each value of β, k). On these graphs, the solid curve is based on linear interpolation of the expectation and the dashed curve on second-order spline interpolation.

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

The true posterior $\pi(\beta, k|\mathbf{y}, \mathbf{X})$ 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 previous MCMC algorithm. The predictive $\mathbb{P}(y_{n+1} = g|\mathbf{x}_{n+1}, \mathbf{y}, \mathbf{X})$ 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 modification of the performance 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 this product (with the difficulty that the components 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.

As illustrated on Figure 6 for Ripley's benchmark, the random walk Metropolis–Hastings algorithm detailed earlier performs satisfactorily with the approximation (8), 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 $\hat{\beta} = 2.28$. Based on the last 10^4 MCMC iterations, the prediction performance of (7) is such that the error rate on the test set of 1,000 points is 8.7%. Figure 6 also indicates how limited the information is about k. We settled on the value $\beta_{\max} = 4$ by trial and error: we started with $\beta_{\max} = 2$, observed that the simulated values of β were stopped by this bound, then we used $\beta_{\max} = 3$, and finally we fixed $\beta_{\max} = 4$, a choice that did not impose a constraint on the simulated values of the Markov chain.

3.3 Path Sampling

A now standard approach to the estimation of normalizing constants is *path sampling*, described in Gelman and Meng (1998) (see also Chen, Shao, and Ibrahim 2000), and derived from the Ogata (1989) method, in which the ratio of two normalizing constants, $Z(\beta', k)/Z(\beta, k)$, can be decomposed as an integral approximated by Monte Carlo techniques.

The basic derivation of the path sampling approximation is that, if

$$S(\mathbf{y}) = \sum_i \sum_{\ell\sim_k i}^{\delta_{y_i}} (y_\ell)/k\,, \quad \text{then} \quad Z(\beta,k) = \sum_{\mathbf{y}} \exp[\beta S(\mathbf{y})]$$

and

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

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

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

is easily evaluated by a numerical approximation.

The practical drawback with this approach is that each time a new ratio is to be computed an approximation of the above integral needs to be produced. A further step is thus necessary: we approximate the function $Z(\beta, k)$ for each value of k and for a few values of β, whose number depends on the slope of $Z(\beta, k)$, and later we use numerical interpolation to extend the approximation. Because the function $Z(\beta, k)$ is very smooth, the additional approximation error is quite limited. Once this approximation is computed, the resulting Metropolis–Hastings algorithm is very fast, as well as being efficient if the approximation of $Z(\beta, k)$ is sufficiently smooth. [We stress, however, that the overall computational cost is very high because of the joint approximation in (β, k).]

We illustrate this approximation on Ripley's benchmark in Figure 4. Within the expectation, the $\mathbf{y}$'s are simulated using a systematic scan Gibbs sampler. As seen from this graph, when β is small, the Gibbs sampler gives good mixing performance, whereas for larger values it has difficulty in stabilizing, as illustrated by the right plot when $k = 125$. The explanation is that the model is getting closer to the phase transition boundary in that case.

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

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

and use numerical integration to approximate the integral. As shown on Figure 5 the approximated constant $Z(\beta, k)$ is mainly constant in k.

Once $Z(\beta, k)$ has been approximated, we can use the genuine MCMC algorithm of Section 3.1 with no difficulty, the main cost of this approach being thus in the approximation of $Z(\beta, k)$. Figure 6 summarizes the output of the MCMC sampler for Ripley's benchmark. A first item of interest is that the chain mixes much more rapidly (in terms of iterations) than its pseudo-likelihood counterpart. A more important point is that the range and shape of the approximations to both marginal posterior distributions widely differ between both methods, a feature discussed in Section 3.5. When this MCMC sample is used for prediction in (7), the error rate for Ripley's test set is equal to 8.5%.

3.4 Perfect Sampling Implementation and Gibbs Approximation

A different approach to handling missing normalizing constants has been recently developed by Møller et al. (2006). They introduced 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 consider the artificial joint posterior

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

Simulating $(\beta, k, \mathbf{z})$ from this posterior is then equivalent to simulating (β, k) from the original posterior. If we run a

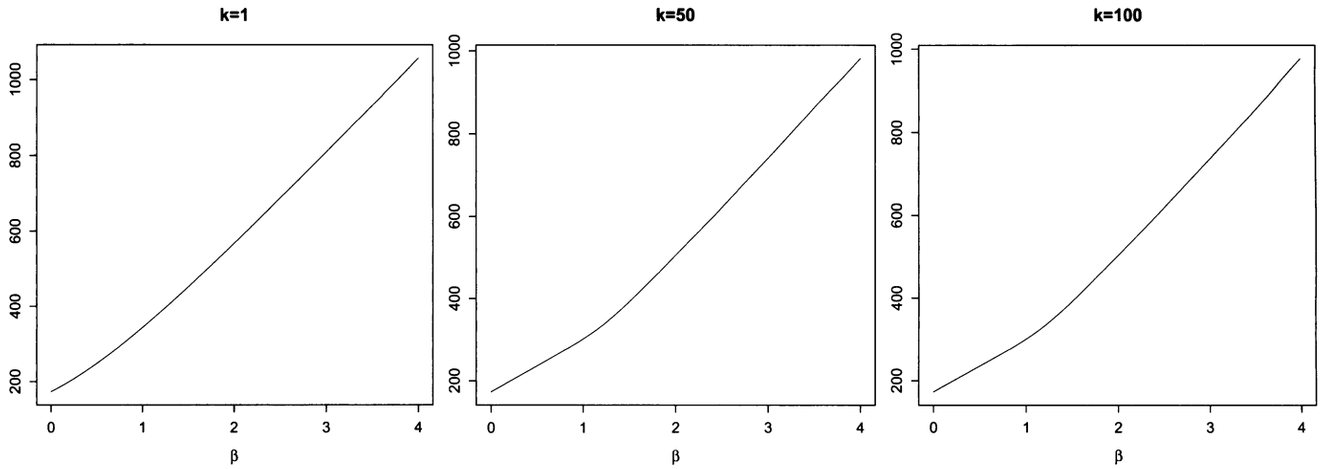

Figure 5. Approximations to the normalizing constant $Z(\beta, k)$ for Ripley's benchmark: the β's are equally spaced between 0 and $\beta_{\max} = 4$, and $k = 1$ (left), $k = 50$ (middle), and $k = 100$ (right) (based on 10^4 Monte Carlo iterations with 500 burn-in steps).

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 $(\beta, k, \mathbf{z})$ (where $\mathbf{z}$ is simulated from the sampling distribution), the associated Metropolis–Hastings ratio is

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

which cancels out the constants $Z(\beta, k)$ and $Z(\beta', k')$. The method of Møller, Pettitt, Reeves, and Berthelsen (2006) is calibrated by the choice of the artificial target $g(\mathbf{z}|\beta, k, \mathbf{y})$ 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 later. The alternative of setting a target $g(\mathbf{z}|\beta, k, \mathbf{y})$ that would depend on β and k is appealing but faces computational

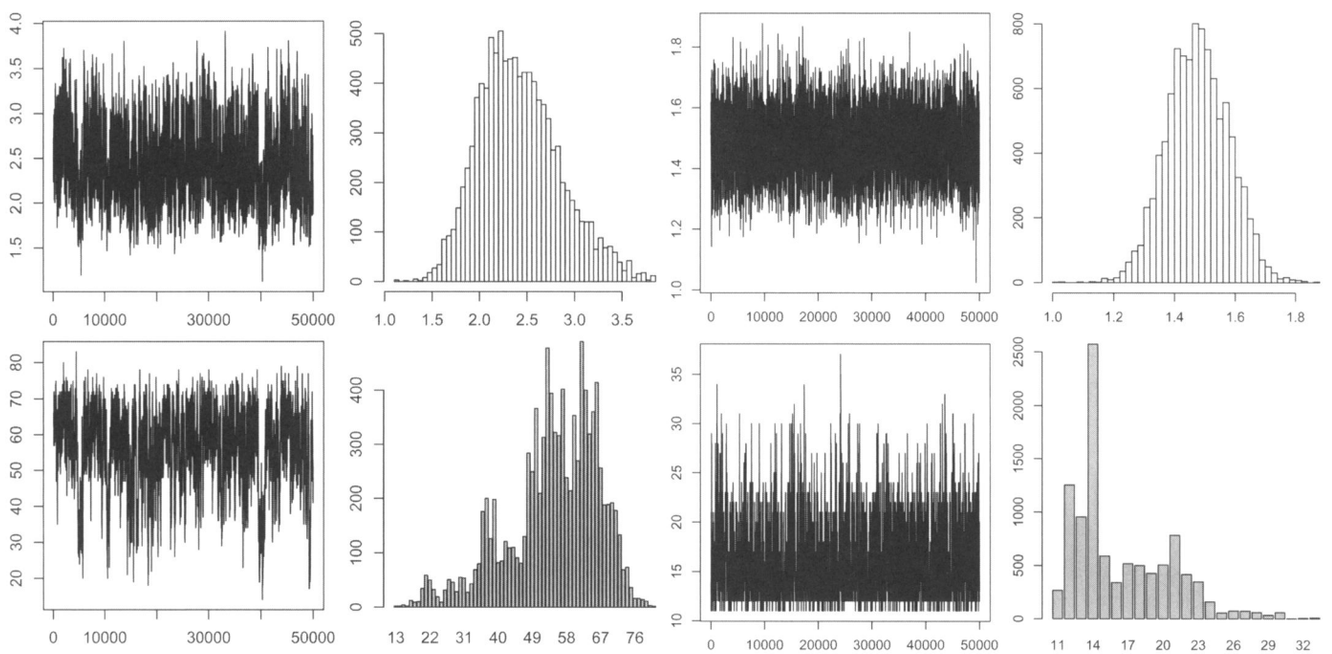

Figure 6. Summary of a random walk Metropolis–Hastings algorithm for Ripley's benchmark: (left) based on the pseudo-likelihood approximation (8) of the normalizing constant and (right) based on the path sampling approximation of the normalizing constants. In each case, there are 5×10^4 iterations, with a 4×10^4 iteration burn-in steps, $\tau^2 = 0.05$ and $r = 3$. (top) Sequence and marginal histogram for β when $\beta_{\max} = 4$ and (bottom) sequence and marginal barplot for k.

difficulties in that natural proposals involve normalizing constants that cannot be computed.

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 $f(\mathbf{z}|\beta, k)$ via perfect sampling. We refer the reader to Häggström (2002), Møller (2003), and Møller and Waagepetersen (2003) for details of this simulation technique and for a discussion of its limitations. Without entering into technical details, we use the fact 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 (Berthelsen and Møller 2003). More precisely, we can use a coupling-from-the-past strategy (Propp and Wilson 1998): in this setting, starting far enough in the past from both 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 into 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: 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 to some extent by using an additional accept-reject step based on smaller values of β that avoids this explosion in the computational time.

During a first run, we have observed that a poor choice for $(\hat{\beta}, \hat{k})$ leads to unsatisfactory performance with the algorithm. Using the pseudo-likelihood estimate as plug-in value $(\hat{\beta}, \hat{k})$, the Markov chain has a low energy and a high rejection rate. However, use of the estimate $(\hat{\beta}, \hat{k}) = (1.45, 13)$ resulting from this first run does improve considerably the performance of the algorithm: the corresponding output is given Figure 7. In this case, the predictive error rate is equal to 0.084.

While this elegant solution based on an auxiliary variable evacuates the issue of the normalizing constant, it faces several computational difficulties. First, as noted previously, the choice of $g(\mathbf{z}|\beta, k, \mathbf{y})$ is driving the algorithm and plug-in estimates $(\hat{k}, \hat{\beta})$ need to be periodically reassessed. Second, perfect simulation from the distribution $f(\mathbf{z}|\beta, k)$ is extremely costly and may fail if β is close to the phase transition boundary. Furthermore, the numerical value of this critical point is not known beforehand and it thus requires an exploration stage. Finally, the extension of the perfect sampling scheme to more than $G = 2$ classes has not yet been achieved.

For these different reasons, we advocate the substitution of a Gibbs sampler for the previous perfect sampler. If we replace the perfect sampling step with 500 (complete) iterations of the Gibbs sampler on $\mathbf{z}$, the computing time is linear in the number n of observations and the results are virtually the same. One has to remember that the simulation of $\mathbf{z}$ is of second order with respect to the original problem of simulating the posterior distribution of (β, k), $\mathbf{z}$ being an auxiliary variable introduced to overcome the difficulty with the normalizing constant. Therefore, the additional uncertainty induced by the Gibbs sampler is limited. Figure 7 compares the Gibbs solution with the perfect sampling implementation and it shows how little loss is incurred by this Gibbs approximation, whereas the gain in computing time is

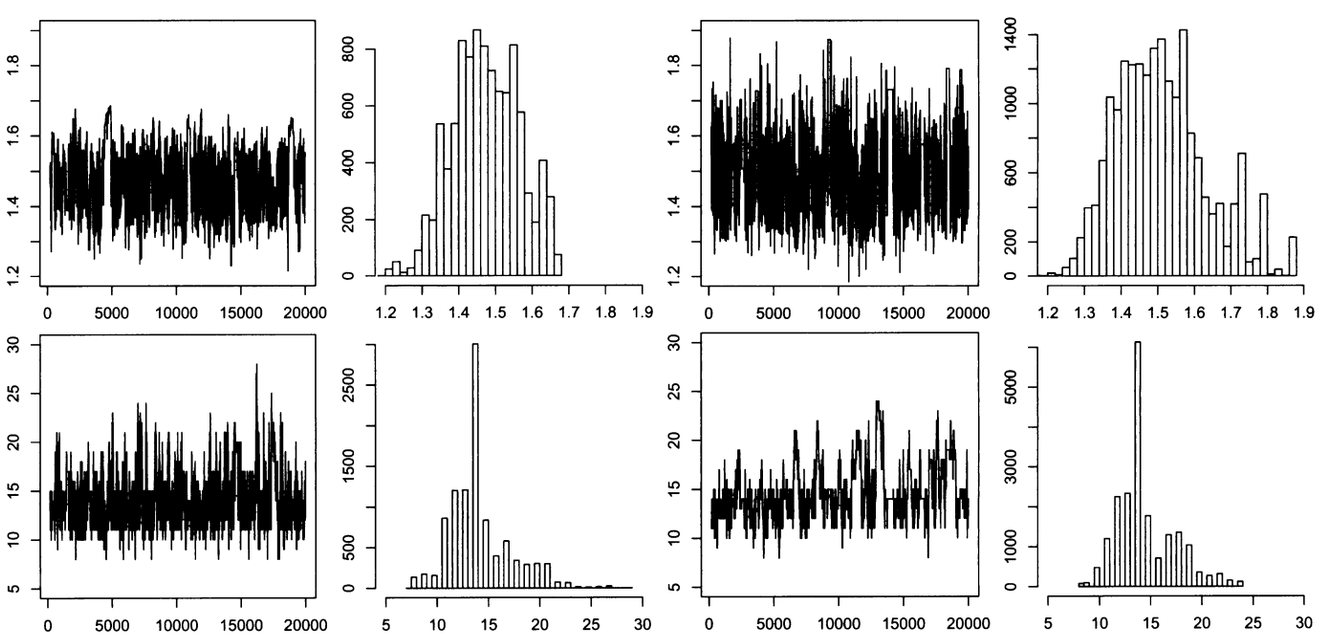

Figure 7. Comparison of (left) the output of a random walk Metropolis–Hastings algorithm for Ripley's benchmark, based on perfect sampling, and of (right) the output of its Gibbs approximation for a plug-in estimate $(\hat{k}, \hat{\beta}) = (13, 1.45)$ and 2×10^4 iterations, with 10^4 burn-in steps and $\tau^2 = 0.05$, $r = 3$: (top) sequence and marginal histogram for β and (bottom) sequence and marginal barplot for k.

enormous. For 5×10^4 iterations, the time required by the Gibbs sampler is approximately 20 min, compared with more than a week for the corresponding perfect sampler (under the same C environment on the same machine).

3.5 Evaluation of the Pseudo-Likelihood Approximation

Given that the previous alternatives can all be implemented for small n's, it is of direct interest to compare them to evaluate the effect of the pseudo-likelihood approximation. As demonstrated in the previous section, we can 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).

Using Ripley's benchmark, histograms of simulated β's, conditional or unconditional on k, show a gross misrepresentation of the samples produced by the pseudo-likelihood approximation, as seen on Figures 8. (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 be argued that the defect lies with the path sampling evaluation of the constant, but this evaluation strongly coincides with the perfect sampling implementation. The left part of Figure 8 shows that, the larger k is, the worse is this discrepancy, whereas the right part of Figure 8 indicates that both β and k are significantly overestimated by the pseudo-likelihood approximation. (It is natural to find such a correlation between β and k when we realize that the likelihood depends on β/k.) We 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 slow convergence in one or both samplers.

To assess the predictive properties of both approaches, we also provide a comparison of the class probabilities $\mathbb{P}(y = 1|x, \mathbf{y}, \mathbf{X})$ for the test sample. As shown by Figure 9, the predictions are quite different for values in the middle of the range, with no clear bias direction in using pseudo-likelihood as an approximation. Note that the discrepancy may be substantial and may result in a large number of different classifications.

4. ILLUSTRATION ON REAL DATASETS

In this section, we illustrate the behavior of the proposed methodology on some benchmarks. We first describe the calibration of the algorithm used on each dataset. As starting value for the Gibbs approximation in the Møller scheme, we use the maximum pseudo-likelihood estimate and we iterate this Gibbs sampler 500 times. After 10^4 iterations of the algorithm, we modify the plug-in estimate using the current average and then we run 5×10^4 more iterations.

The first dataset is borrowed from the MASS library of R. It consists of the records of 532 Pima Indian women who were tested by the U.S. National Institute of Diabetes and Digestive and Kidney Diseases. 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 68 diagnosed with diabetes, and a test set of the remaining 332 women, including 109 diagnosed with diabetes. The performance of the knn classifier on the test dataset for $k =$ 1, 3, 15, 31, 57, 66 gives a misclassification error rate equal to 31.6%, 22.9%, 22.6%, 21.1%, 20.5%, and 20.8%, respectively. If we use 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 misclassification error rate, namely the range 57–66.

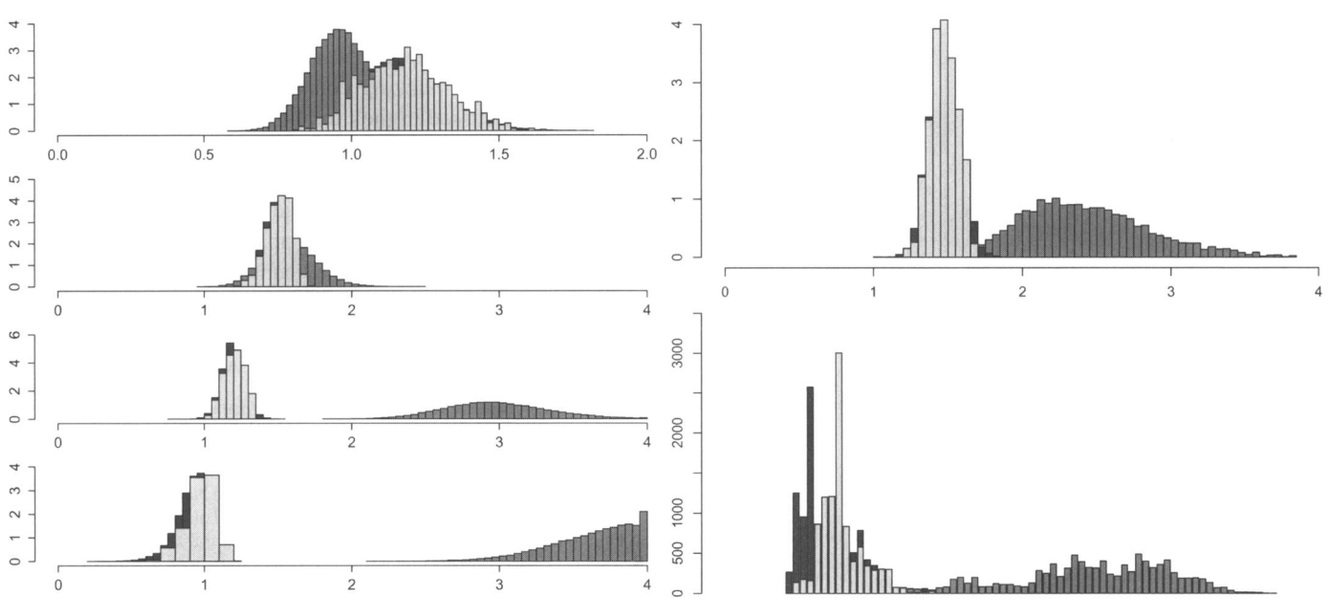

Figure 8. Comparison of the approximations to the posterior distributions of β and k based on the pseudo (medium grey), the path (dark grey), and the perfect (light grey) schemes for Ripley's benchmark: (left) approximations to the posterior distribution of β when the value of k is fixed and equal to 1, 10, 70, and 125 (2×10^4 iterations and 10^4 burn-in); (right) approximations to the posterior distributions of β and k based on the results of Figures 6 and 7.

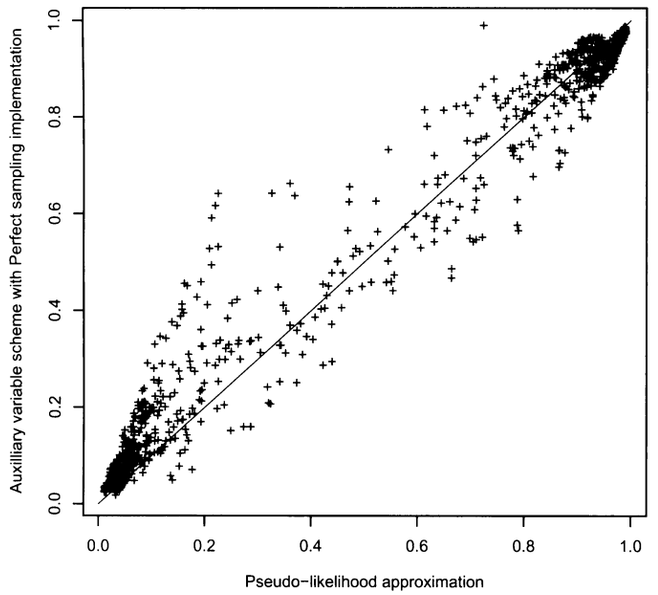

Figure 9. Comparison of the class probabilities $\mathbb{P}(y = 1|x, \mathbf{y}, \mathbf{X})$ estimated at each point of the test sample.

As shown in Figure 10, the mixing on k is slow if reasonable. 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 knn objective function, because mutual neighbors are weighted twice as heavily as single neighbors in this likelihood. Over the final 20,000 iterations, the prediction error is 0.209, quite in line with the knn solution.

To illustrate the ability of our method to consider more than two classes, we also used the benchmark dataset forensic glass fragments, studied in Ripley (1994). This dataset involves nine covariates and six classes, some of which are rather rare. Following the recommendation made in Ripley (1994), we coalesced some classes to reduce the number of classes to four. We then randomly partitioned the dataset to obtain 89 individuals in the training dataset and 96 in the testing dataset. Leave-one-out cross-validation led us to choose the value $k = 17$, a much larger value than those explored by our MCMC sampler. The error rate of the 17 nearest-neighbor procedure on the test dataset is 0.35, whereas using our procedure; we obtain an error rate of 0.29.

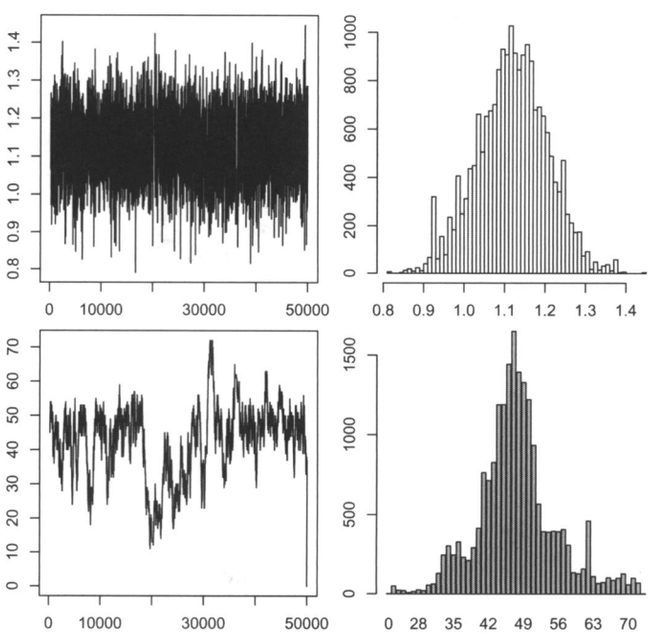

Figure 10. Pima Indian diabetes study based on 5×10^4 iterations of the Gibbs–Møller sampling scheme with $\tau^2 = 0.05$, $r = 3$, $\beta_{\max} = 4$, and $K = 68$, for (top) β and (bottom) k.

5. CONCLUSIONS

While the probabilistic background to a Bayesian analysis of knn methods was initiated by Holmes and Adams (2003), the present article tidies up this setting by defining a coherent probabilistic model on the training dataset. Model (3) provides a sound setting for Bayesian inference and for evaluating both the most likely allocations for the test dataset and the uncertainty that goes with them. The advantages of using a probabilistic environment are clearly demonstrated: it is only within this setting that tools like predictive maps as in Figure 3 can be constructed. This should prove a strong bonus for experimenters, because boundaries between most likely classes can thus be estimated and regions can be established in which allocation to a specific class is uncertain. In addition, the probabilistic framework allows for a natural and integrated analysis of the number of neighbors involved in the class allocation, from standard model-choice perspective. This perspective can be extended to the choice of the most significant components of the covariate $\mathbf{x}$, even though this possibility is not explored in the current paper.

The present article also addresses the computational difficulties related to this approach, namely the well-known issue of the intractable normalizing constant. Although this has been thoroughly discussed in the literature, our comparison of three independent approximations leads to the strong conclusion that the pseudo-likelihood approximation is not to be trusted for training sets of moderate size. Furthermore, while the path sampling and the perfect sampling approximations are useful in establishing this conclusion, they cannot be advocated as such at the operational level, but we also demonstrate that a Gibbs sampling alternative to the perfect sampling scheme of Møller et al. (2006) is both operational and practical.

[Received April 2007. Revised November 2008]

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 Journal of Statistics,* 30, 549–564.

Besag, J. (1974), "Spatial Interaction and the Statistical Analysis of Lattice Systems" (with discussion), *Journal of the Royal Statistical Society, Ser. B,* 36, 192–236.

Besag, J., and Kooperberg, C. (1995), "On Conditional and Intrinsic Autoregressions," *Biometrika,* 82, 733–746.

Besag, J., York, J., and Mollié, A. (1991), "Bayesian Image Restoration, With Two Applications in Spatial Statistics," (with discussion and a reply by Besag), *Annals of the Institute of Statistical Mathematics,* 43, 1–59.

Bühlmann, P. (2004), "Bagging, Boosting and Ensemble Methods, *Handbook of Computational Statistics*, Berlin: Springer, pp. 877–907.

Bühlmann, P., and Yu, B. (2002), "Analyzing Bagging," *Annals of Statistics,* 30, 927–961.

——— (2003), "Boosting With the L_2 Loss: Regression and Classification," *Journal of the American Statistical Association,* 98, 324–339.

Buttrey, S. (1998), "Nearest-Neighbor Classification with Categorical Variables," *Computational Statistics & Data Analysis,* 28, 157–169.

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

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

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

Freund, Y., and Schapire, R. E. (1997), "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting," *Journal of Computer and System Sciences,* 55(1, part 2), 119–139.

Friel, N., and Pettitt, A. N. (2004), "Likelihood Estimation and Inference for the Autologistic Model," *Journal of Computational and Graphical Statistics,* 13, 232–246.

Friel, N., Pettitt, A., Reeves, R., and Wit, E. (2005), "Bayesian Inference in Hidden Markov Random Fields for Binary Data Defined on Large Lattices,, Technical Report, Department of Statistics, University of Glasgow"

Gelman, A., and Meng, X.-L. (1998), "Simulating Normalizing Constants: From Importance Sampling to Bridge Sampling to Path Sampling," *Statistical Science,* 13, 163–185.

Häggström, O. (2002), *Finite Markov Chains and Algorithmic Applications* (Vol. 52), Student Texts, London U.K.: London Mathematical Society.

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

Heikkinen, J., and Hogmander, H. (1994), "Fully Bayesian Approach to Image Restoration With an Application in Biogeography," *Journal of the Royal Statistical Society,* Ser. C, 43, 569–582.

Hoeting, J. A., Madigan, D., Raftery, A., and Volinsky, C. (1999), "Bayesian Model Averaging: A Tutorial" (with discussion)," *Statistical Science,* 14, 382–417.

Holmes, C. C., and Adams, N. M. (2002), "A Probabilistic Nearest Neighbour Method for Statistical Pattern Recognition," *Journal of the Royal Statistical Society,* Ser. B, 64, 295–306.

——— (2003), "Likelihood Inference in Nearest-Neighbour Classification Models," *Biometrika,* 90, 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.

Manocha, J., and Girolami, M. (2007), "An Empirical Analysis of the Probabilistic K-Nearest Neighbour Classifier," *Pattern Recognition Letters,* 28, 1818–1824.

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

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

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*, Boca Raton, FL: Chapman and Hall/CRC.

Ogata, Y. (1989), "A Monte Carlo Method for High-Dimensional Integration," *Numerical Mathematics,* 55, 137–157.

Propp, J., and Wilson, D. (1998), "Coupling From the Past: A User's Guide," *Microsurveys in Discrete Probability (Princeton, NJ, 1997)*, (Vol. 41), *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, Providence RI: American Mathematical Society, pp. 181–192.

Ripley, B. D. (1994), "Neural Networks and Related Methods for Classification" (with discussion)," *Journal of the Royal Statistical Society,* Ser. B, 56, 409–456.

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

Robert, C. (2001), *The Bayesian Choice*, (2nd ed.), Springer Texts in Statistics, New York: Springer-Verlag.

Zhang, T., and Yu, B. (2005), "Boosting With Early Stopping: Convergence and Consistency," *Annals of Statistics,* 33, 1538–1579.