J. R. Statist. Soc. B (2002)
64, *Part* 2, *pp.* 295–306

A probabilistic nearest neighbour method for statistical pattern recognition

C. C. Holmes and N. M. Adams

Imperial College of Science, Technology and Medicine, London, UK

[Received July 2000. Final revision October 2001]

Summary. Nearest neighbour algorithms are among the most popular methods used in statistical pattern recognition. The models are conceptually simple and empirical studies have shown that their performance is highly competitive against other techniques. However, the lack of a formal framework for choosing the size of the neighbourhood k is problematic. Furthermore, the method can only make discrete predictions by reporting the relative frequency of the classes in the neighbourhood of the prediction point. We present a probabilistic framework for the k-nearest-neighbour method that largely overcomes these difficulties. Uncertainty is accommodated via a prior distribution on k as well as in the strength of the interaction between neighbours. These prior distributions propagate uncertainty through to proper probabilistic predictions that have continuous support on (0, 1). The method makes no assumptions about the distribution of the predictor variables. The method is also fully automatic with no user-set parameters and empirically it proves to be highly accurate on many bench-mark data sets.

Keywords: Bayesian k nearest neighbour; Nonparametric classification; Probabilistic nearest neighbour

1. Introduction

The k-nearest-neighbour algorithm is among the most popular methods used in statistical pattern recognition. The method dates back to an unpublished report by Fix and Hodges (1951), with over 900 research articles published on the method since 1981 alone. Dasarathy (1991) has provided a comprehensive collection of around 140 key papers.

The k-nearest-neighbour algorithms are primarily used for predictive pattern recognition (classification) where the task is to label new data as belonging to one of Q distinct classes. This classification is based on a historic data set of examples $\{y_i, \mathbf{x}_i\}_{i=1}^n$, where $y_i \in \{1, \ldots, Q\}$ denotes the class labels and $\mathbf{x}$ denotes a set of predictor variables, $\mathbf{x} \in \mathcal{H}$, where $\mathcal{H}$ is the domain of $\mathbf{x}$.

The k-nearest-neighbour procedure simply predicts a new point y_{n+1} to be the most common class found among the k nearest neighbours of $\mathbf{x}_{n+1}$ in the set $\{\mathbf{x}_i\}_{i=1}^n$. The term 'nearest' is determined by a distance metric $\rho(\mathbf{x}_{n+1}, \mathbf{x}_i)$ which is commonly taken to be the Euclidean norm. This so-called 'vanilla' k-nearest-neighbour algorithm remains the most popular, partly because of its simplicity but also because of empirical evidence that shows that the method is highly accurate at prediction on a variety of data sets (see Michie *et al.* (1994)).

However, some drawbacks exist with the k-nearest-neighbour procedure. The choice of k is not trivial. Theoretical results which state that optimally k should tend to ∞ and $k/n \to 0$ hold only for $n \to \infty$ (Devroye *et al.*, 1996). For small-to-moderate n the choice of k is important

Address for correspondence: C. C. Holmes, Department of Mathematics, Huxley Building, Imperial College of Science, Technology and Medicine, 180 Queen's Gate, London, SW7 2BZ, UK.
E-mail: c.holmes@ic.ac.uk

 1369–7412/02/64295

(McLachlan, 1992; Fukanaga and Hostetler, 1973). Enas and Choi (1986) performed a simulation study and suggested rough guidelines for k scaling as $n^{2/8}$ or $n^{3/8}$. However, no one method dominates the literature although simply setting $k=1$ or choosing k via cross-validation on the misclassification rate appear the most popular (Ripley, 1996), the advantage of cross-validation being that it takes into account the class labels of the data under study. Regardless of how k is chosen, the predictions made by the algorithm have no probabilistic interpretation and the standard approach of simply normalizing the counts for each class leads to a discretization of predictions that depends on k. This lack of a probabilistic interpretation in the predictions makes it difficult to incorporate the k-nearest-neighbour method in a coherent decision process where predictions lead to actions with associated losses.

To counter these difficulties we propose a probabilistic framework for the k-nearest-neighbour model that accommodates uncertainty in k as well as in the strength of the interaction between neighbours. This leads to marginal predictions being given as proper probabilities that are continuous on (0, 1). We formulate the method as a block sequential algorithm where it is assumed that blocks of data arrive over time. We show that this leads to the predictive distribution for a new observation having the form of a probabilistic nearest neighbour (PNN) prior. We find that the predictive class conditional probability fields tend to have smooth contours that spread out in regions of low density of the data. The method makes no assumptions about the distribution of the predictor variables. The method is also fully automatic with just two unknown parameters that are assigned default priors.

The rest of the paper is structured as follows. In Section 2 we describe the model and discuss implementation issues. We illustrate the method on some standard data sets in Section 3 and in Section 4 we provide a brief discussion. A MATLAB version of the algorithm is available on request from the first author.

2. Probabilistic nearest neighbour

2.1. Formulation

Suppose that we observe data $\mathcal{D}$ that arrive sequentially in blocks, with $\mathcal{D} = \{(Y_1, \mathbf{X}_1), (Y_2, \mathbf{X}_2), \ldots, (Y_m, \mathbf{X}_m)\}$, where $Y_s = \{y_1^{(s)}, \ldots, y_{n_s}^{(s)}\}$ and $\mathbf{X}_s = \{\mathbf{x}_1^{(s)}, \ldots, \mathbf{x}_{n_s}^{(s)}\}$. A typical situation might be when $m = 2$ and we have a training set $(Y_1, \mathbf{X}_1)$ of n_1 points and an out-of-sample evaluation set $(\mathbf{X}_2)$ of n_2 points with Y_2 unknown. Another common scenario is when single observations arrive over time, $\mathcal{D} = \{(y_1^{(1)}, \mathbf{x}_1^{(1)}), (y_1^{(2)}, \mathbf{x}_1^{(2)}), \ldots, (y_1^{(m)}, \mathbf{x}_1^{(m)})\}$, with $n_s = 1$ for all s.

Let $Y = (Y_1, \ldots, Y_m) = (y_1, \ldots, y_n)$ denote the set of combined responses and let $\mathbf{X}$ denote the set of combined predictors, $\mathbf{X} = (\mathbf{X}_1, \ldots, \mathbf{X}_m) = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$, where $n = \Sigma_{s=1}^{m} n_s$, and let $T = \{t_1, \ldots, t_n\}$ be a set of indicator variables such that $t_i = j$ if the ith data point in Y came from the jth block, $j \in \{1, \ldots, m\}$.

We now define the joint prior distribution on Y given $\mathbf{X}$, β and k to be

$$p(Y|\mathbf{X}, \beta, k) = \prod_{i=1}^{n} \frac{\exp\left\{\beta(1/k) \sum_{j \overset{k}{\sim} i} \delta_{y_i y_j}\right\}}{\sum_{q=1}^{Q} \exp\left\{\beta(1/k) \sum_{j \overset{k}{\sim} i} \delta_{q y_j}\right\}}, \tag{1}$$

where δ_{ab} is the Dirac function (δ_{ab} equals 1 if $a = b$, and 0 otherwise) β is an interaction parameter that governs the strength of association between the neighbouring y_is and $\Sigma_{j \overset{k}{\sim} i}$

denotes that the summation is over the k nearest neighbours of $\mathbf{x}_i$ in the set $\{\mathbf{X}_1, \ldots, \mathbf{X}_{t_i}\} \setminus \mathbf{x}_i$ given the distance metric $\rho(\cdot, \cdot)$, where as before t_i is the index of the block that contains $\mathbf{x}_i$. The term $(1/k)\Sigma_{j \overset{k}{\sim} i} \delta_{qy_j}$ records the proportion of points of class q in the k nearest neighbours of $\mathbf{x}_i$. In this manner the probability of y_i is conditioned on only those points in blocks up to and including the t_ith block.

The predictive distribution for a new observation is

$$p(y_{n+1}|\mathbf{x}_{n+1}, Y, \mathbf{X}, \beta, k) = \frac{\exp\left\{\beta(1/k) \sum\limits_{j \overset{k}{\sim} n+1} \delta_{y_{n+1} y_j}\right\}}{\sum\limits_{q=1}^{Q} \exp\left\{\beta(1/k) \sum\limits_{j \overset{k}{\sim} n+1} \delta_{q y_j}\right\}}, \tag{2}$$

so the most probable class for y_{n+1} is given by the most common class found among its k nearest neighbours. The interaction parameter β is akin to a regression coefficient and both equation (1) and equation (2) have the form of a local logistic regression. Clearly equation (2) is a proper probability density.

The use of sequential conditioning in the neighbourhood removes the need for an extra term in distribution (2) relating to those points in the original data set $\{\mathbf{x}_i\}_{i=1}^n$ for which $\mathbf{x}_{n+1}$ was now one of their k nearest neighbours. This would require the recomputation of the neighbourhood structure for each prediction point, which is computationally expensive though not prohibitive. For the data sets examined in Section 3 the results were not sensitive to this assumption. However, sequential conditioning is a natural formulation for statistical pattern recognition where we have observed some data and we wish to use this to predict future examples.

The joint distribution (1) is reminiscent of the priors found in Markov random field models used in spatial statistics. This is not surprising as the Markov random field priors are also motivated by the local conditional distributions that they induce (Besag *et al.*, 1991; Besag and Kooperberg, 1995). Our reason for using equation (1) is that it is a normalized distribution whose normalizing constant is independent of β and k. This normalization greatly aids the analysis when we come to consider β and k as random.

Treating β and k as known and fixed *a priori* is unrealistic and fails to account for a key component of uncertainty in the model. To accommodate this we assign prior distributions to β and k, leaving the marginal predictive distribution as

$$p(y_{n+1}|\mathbf{x}_{n+1}, Y, \mathbf{X}) = \sum_k \int p(y_{n+1}|Y, \mathbf{X}, \beta, k)\, p(\beta, k|Y, \mathbf{X})\, \mathrm{d}\beta, \tag{3}$$

where

$$p(\beta, k|Y, \mathbf{X}) \propto p(Y|\mathbf{X}, \beta, k)\, p(\beta, k).$$

We have little prior knowledge about the likely values of k and β, other than the fact that β should be positive. Hence, we adopt the independent default prior densities

$$\begin{aligned} p(k) &= U[1, \ldots, k_{\max}], \qquad & k_{\max} &= n, \\ p(\beta) &= c\, I(\beta > 0), & & \end{aligned} \tag{4}$$

where U denotes the uniform distribution, c is a constant and I is the indicator function, so that the prior on β is uniform on $\Re^+$. If a proper prior is required we recommend

$$p(\beta) = 2N(0, c)\, I(\beta > 0)$$

with c set as large as possible. By allowing β to range over $\Re^+$ we remove the condition that the predictive density (2) takes only a finite set of values on (0,1). For large data sets it is reasonable to set $k_{\max}$ smaller than n (our MATLAB program sets $k_{\max} = \min(n, 1000)$). We note that for fixed k and large β the model produces a classifier whose decision boundary matches that of the conventional k nearest neighbour.

The flat priors in density (4) do not lead to predictive overfitting. This is apparent from equation (1) which has the form of a cross-validation probability measure, i.e. in distribution (1) the ith point is not considered to be a neighbour of itself in the ith term of the product. When single observations arrive over time, $\mathcal{D} = \{(y_1, \mathbf{x}_1), (y_2, \mathbf{x}_2), \ldots, (y_m, \mathbf{x}_m)\}$, our method has the properties of a prequential forecasting system as defined in Dawid (1992). Further evidence of smoothing is present when no neighbours of a new point $\mathbf{x}_{n+1}$ belong to a particular class, say the cth class. In this situation we have

$$p(y_{n+1} = c|\mathbf{x}_{n+1}) = \left[\sum_{q=1}^{Q} \exp\left\{ \beta(1/k) \sum_{j \overset{k}{\sim} n+1} \delta_{qy_j} \right\} \right]^{-1},$$

and hence $p(y_{n+1} = c|\mathbf{x}_{n+1}) > 0$ regardless of the data.

2.2. Implementation

The integral in equation (3) requires an approximation method for its evaluation. For computational reasons we shall distinguish between the case where we have only a small number of points for prediction and the situation when a large number of predictions are required.

Standard quadrature algorithms that are found in conventional statistical software will calculate distribution (3) to a desired level of accuracy and this we strongly recommend. However, the evaluation of equation (3) requires $k_{\max}$ calls to the quadrature routine for each prediction point. This might be computationally prohibitive depending on the number of prediction points and the speed of the computer. Our computer takes around 2 s to evaluate one integral. For example, for the evaluation of the German credit data in Section 3.2 we take $k_{\max} = 200$ and hence with 1000 evaluation points we would require 200000 calls to the quadrature routine.

Hence, when many predictions are required and quadrature is computationally infeasible we propose a Markov chain Monte Carlo (MCMC) simulation which is used to draw M samples from $p(\beta, k|Y, \mathbf{X})$ and then to approximate distribution (3) by

$$p(y_{n+1}|\mathbf{x}_{n+1}, Y, \mathbf{X}) \approx \frac{1}{M} \sum_{i=1}^{M} p(y_{n+1}|\mathbf{x}_{n+1}, Y, \mathbf{X}, \beta^{(i)}, k^{(i)}),$$

where $\beta^{(i)}$ and $k^{(i)}$ represent the ith sample in the converged chain. The advantage of this approach is that it allows a single collection of samples to be used for all predictions that are to be made at the same time point.

The MCMC algorithm that we propose is a random walk Metropolis–Hastings algorithm (see, for example, Gilks *et al.* (1996)). We use a single joint proposal, $(k, \beta) \rightarrow (\hat{k}, \hat{\beta})$, with

$$\begin{aligned} \hat{k} &= k \pm U[0, 1, 2, 3], \\ \hat{\beta} &= \beta + N(0, v), \end{aligned} \tag{5}$$

where N is the normal density with a variance v that is chosen during a burn-in phase to achieve around a 30% acceptance rate. We impose reflection at the boundaries of the prior range in density (4) so that, for example, if $\hat{\beta} < 0$ we reset $\hat{\beta}$ as $|\hat{\beta}|$.

Using the uniform priors in expression (4) these proposals are then accepted with probability

$$J = \min\left\{1, \frac{p(Y|\mathbf{X}, \hat{\beta}, \hat{k})}{p(Y|\mathbf{X}, \beta, k)}\right\};$$

otherwise the current values of k and β are retained. The algorithm is run for a sufficiently long time with an initial portion of samples discarded as being unrepresentative of the target distribution. To choose this burn-in time we run two independent chains, one starting from $k = 1$ and $\beta = 10$ and the other from $k = k_{\max}$ and $\beta = 0.01$, and we begin retaining samples when the mean and variance of k and β become indistinguishable across the two chains within a lag of 1000 samples from the current time.

The posterior distribution $p(\beta, k|Y\mathbf{X})$ will change over time as more data become available. In our implementation detailed above we specifically condition on the current time t, using all information observed up to time t to make predictions. Having made predictions, the actual class labels may later become available, leading to an update to the posterior distribution $p(\beta, k|Y, \mathbf{X}, y_{n+1}, \mathbf{x}_{n+1})$. For the sampling approximation this would require either the resampling of the posterior density as above or the use of a dynamic MCMC scheme (e.g. Gilks and Berzuini (2001)). If a small number of predictions is being made at each time then quadrature is more appealing.

3. Examples

Our examples are chosen to illustrate the features and relative performance of the PNN method. Performance is assessed in terms of the average out-of-sample misclassification cost,

$$C = \frac{1}{n}\sum_{i=1}^{n} L(y_i, q) I[y_i \neq \arg\max_q \{p(y_i = q|\mathbf{x}_i, Y, \mathbf{X})\}],$$

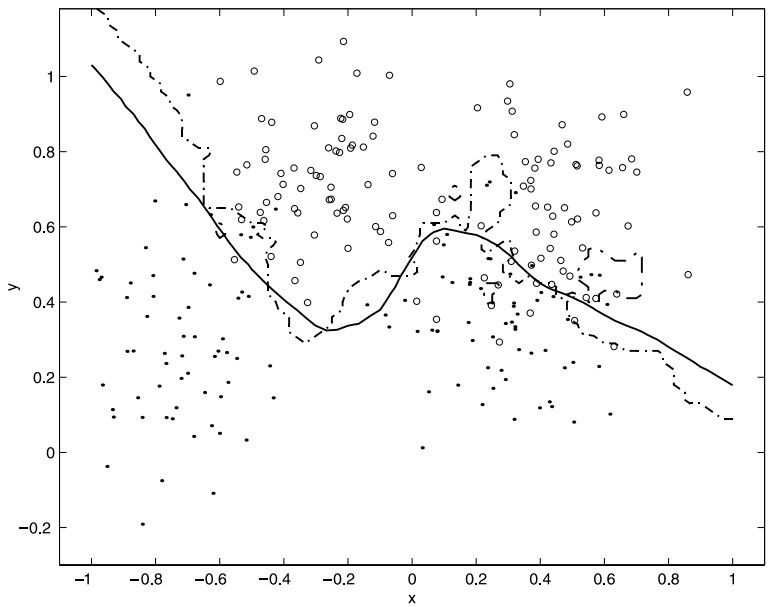

Fig. 1. Training set with decision boundary $p\{(y=1)=0.5\}$ for the PNN (———) and the five-nearest-neighbours (-------) methods, taken from Ripley (1994)

where the summation is over an out-of-sample test set of n points, given the training set $(Y, \mathbf{X})$, $I(\cdot)$ is the indicator function and $L(y_i, q)$ is the cost of misclassifying class y_i as class q.

In all the examples we used the MCMC method with a burn-in of 5000 samples, which we found to be sufficient, after which every 100th sample was retained in the next 50000 samples. The convergence and mixing of the chain were checked using two independent chains as described above and by comparison with the quadrature results for the marginal distribution $p(k|Y)$ and for a small number of predictive densities. The results for quadrature and MCMC sampling were indistinguishable. All the data sets are standardized before the analysis and we use the Euclidean norm as the distance metric $\rho(\cdot)$.

3.1. Synthetic data

Our synthetic data set is taken from Ripley (1994). The data set is a two-class classification problem where each population is an equal mixture of two bivariate normal distributions. A training set of 250 points is used and the model is tested on a set of 1000 points. We use the same data as in Ripley (1994) for ease of comparison. The data can be obtained from `www.stats.ox.ac.uk/pub/PRNN/`.

In Fig. 1 we plot the contour for $p\{(y = 1) = 0.5\}$ produced by the PNN method (full curve) alongside the five-nearest-neighbours results (broken curve) reported in Ripley (1994). We note that the PNN method produces a smooth decision boundary due to the marginalization over β and k. The misclassification rate on the test set was 8.40% for the PNN method which places it second out of the 15 methods that Ripley tested (the standard five-nearest-neighbours method had an error rate of 13.0%, placing it 13th).

The PNN method is a probabilistic method and we plot the full probability contours for $p(y = 1)$ in Fig. 2. We have plotted the contours on a wider scale than in Fig. 1 to highlight some features of the method. The contours in Fig. 2 are seen to vary smoothly over the covariate space $\mathcal{H}$ and spread out in regions of low density of data where there is greater uncertainty in

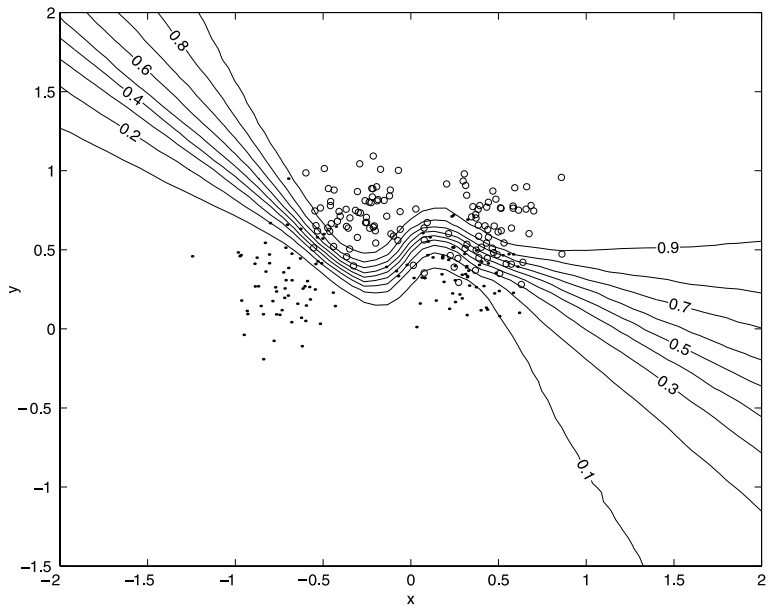

Fig. 2. Contour plot for $p(y=1)$ in the example in Section 3.1

the predictions. This is in contrast with the k-nearest-neighbour method which can only provide discrete non-probabilistic predictions in the set $\{0, 1/k, 2/k, \ldots, k/k\}$ and the contour lines of these values jump jaggedly over $\mathcal{H}$.

We examine the marginal distribution $p(k|Y)$ which is shown in Fig. 3. The distribution of $p(k|Y)$ is striking in two respects. First, relatively high values of k have high probability. Second, the distribution is far from unimodal. The distribution of $p(k|Y)$ is similar to patterns that are often observed in standard k-nearest-neighbour methods when a cost measure such as the cross-validation misclassification rate is plotted against k (see, for example, section 9 of Friedman (1997)). This similarity is not surprising given that $p(k|Y)$, from equation (1), has the form of a marginal cross-validation probability measure for k. Furthermore, the marginalization over β will tend to enhance the multimodality. A slight positive correlation is present in the conditional distribution of β given k as shown in Fig. 4 and the density appears to spread out slightly with increasing k.

3.2. Statlog Data

To compare the predictive performance of the method against other classification models we analysed four two-class data sets chosen from the Statlog project, which compared 24 common classification methods on a variety of data sets. Details and findings of the project are described in Michie *et al.* (1994), and the data sets can be found at the data repository at the University of California, Irvine (Blake and Merz, 1998). In this section we compare the probabilistic version with the standard k-nearest-neighbour method (as used in Statlog) on the four data sets described in Table 1. Our comparisons duplicate the Statlog procedure by standardizing the data and then estimating performances by using cross-validation. Our standardization procedure, as in Statlog, subtracts the average from each predictor and then rescales it to have unit empirical standard deviation. The 'cross-validation fold' in Table 1 refers to the way that the data are partitioned

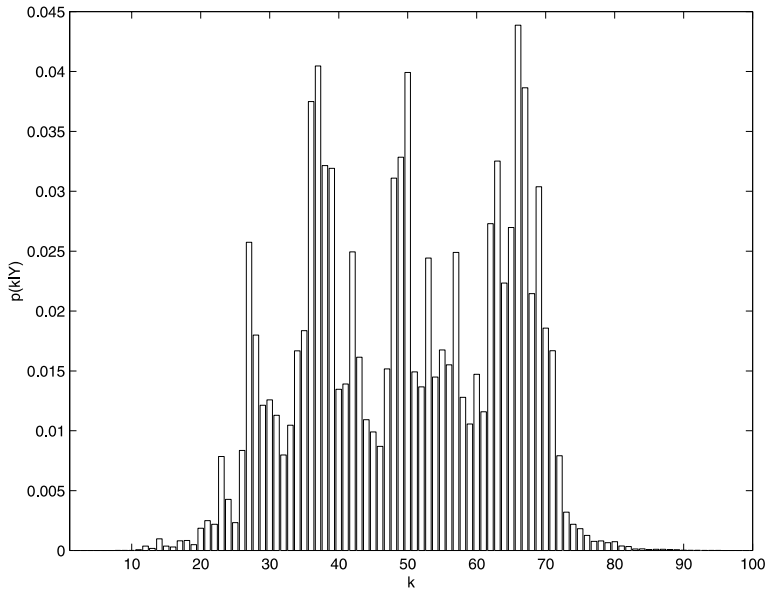

Fig. 3. Plot of $p(k|Y)$ for the example in Section 3.1

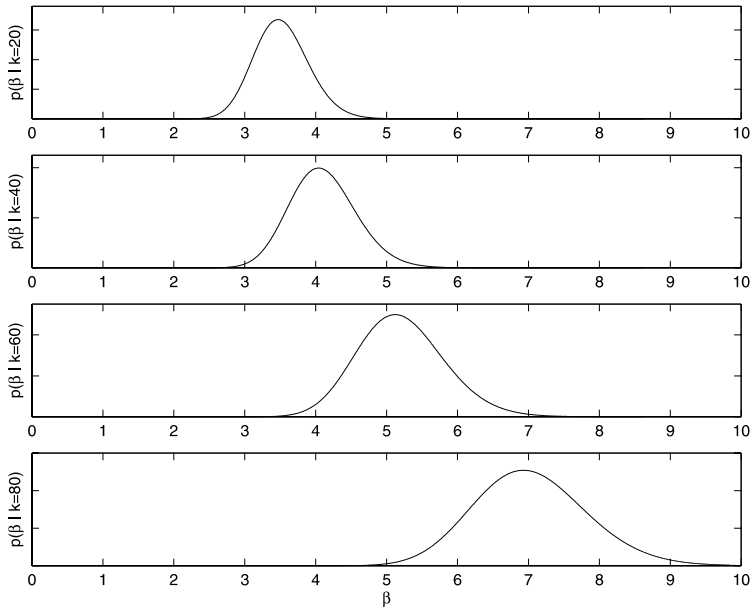

Fig. 4. Conditional distribution of β given k, $p(\beta \mid k, Y)$ for the example in Section 3.1: from top to bottom we have conditioned on $k = 20, 40, 60, 80$

for evaluation. An N-fold procedure will split the data into N disjoint subsets, each used once to test the model built on the remaining $N - 1$ subsets. This is repeated N times to obtain an overall error rate.

The first data set that we consider is the Australian credit data which is concerned with assessing the creditworthiness of applicants for credit cards. For confidentiality, very little information about these data is provided. In credit problems we would typically expect to have different misclassification costs (Hand and Henley, 1997). This information is not given so we assess performances by using the error rate.

The German credit data are also concerned with classifying credit applicants into good and bad risk classes. The 24 recorded predictors are typical for this domain and include the amount of credit, length of employment and reason for the loan. Interestingly, gender is included in these data, although this predictor is never used in practice because of the risk of discrimination. Some categorical variables are recoded as indicators. The German credit data have unequal costs attached to misclassification whereby misclassifying a bad risk applicant as good is quoted as five times as costly as the reverse.

Table 1. Data set characteristics

Data set	*Observations*	*Predictors*	*Cross-validation folds*
Australian credit	690	14	10
Diabetes	768	8	12
German credit	1000	24	10
Heart	270	13	9

The diabetes data consist of observations of adult females of Pima Indian heritage. The classification is a positive or non-positive response for diabetes according to a World Health Organization criterion that partitions a continuous physiological characteristic into the binary response. The Statlog project found that the k-nearest-neighbour method was among the worst classifiers on these data, attributing the method's poor performance to scaling difficulties.

Finally, in the heart data set the class attribute is derived from a five-level categorical variable relating to the degree of heart disease. The problem is to distinguish the absence (value 0) from the presence (values 1–4) of heart disease. The predictor variables are a mixture of physical characteristics and responses to medical tests. The cost of misclassifying a sufferer of heart disease as healthy is five times that of the reverse misallocation.

In the German credit and heart data set examples, where the type 1 errors are five times more costly than type 2 errors, the posterior classification boundary shifts to around the 0.166 probability contour. The results of our comparison are shown in Table 2, along with a conventional k-nearest-neighbour method with k chosen by tenfold cross-validation on each training set. We also include the best result for each data set as reported in Statlog. The PNN method appears consistently more accurate than the k-nearest-neighbour method on these data sets. It is interesting that the performance is particularly good on the data sets with costs. This may be due to random error, though we believe that a proper probabilistic model with a continuous probability field is likely to outperform the standard method whose discrete predictions seem ill matched to the problem at hand.

The marginal distributions $p(k|Y)$ for the four data sets are shown in Fig. 5. It is apparent that the PNN method adjusts the size of the neighbourhood depending on the problem. This data adaption of k is crucial to the ability of the method to fit both simple and complex decision boundaries.

3.3. Spatial Data

Our final example is chosen to illustrate the ability of the method to fit complicated class conditional fields, as well as the simple ones shown in Fig. 2. For this we use the Lancing Wood data described in Diggle (1983). The data originate from a study by Gerrard (1969) who provided the locations of three major species of trees (hickories, maples and oaks) in a 19.6-acre plot in Lancing Woods, Clinton County, Michigan, USA. The data are plotted in Figs 6(a)–6(c).

Diggle (1983) was concerned with investigating spatial dependence between the patterns. From our perspective, the data are interesting because of the strongly overlapping class conditional distributions. Our goal is to provide predictive distributions for the three classes and to answer

Table 2. Statlog test errors†

Data set	*Error rate of PNN*	*Optimum k nearest neighbour*	*Statlog best*
Australian credit	0.147 (7)	0.149 (9)	0.131 (1)
Diabetes	0.247 (7)	0.267 (13)	0.219 (1)
Heart‡	0.377 (2)	0.418 (5)	0.374 (1)
German credit‡	0.583 (3)	0.591 (5)	0.535 (1)

†Numbers in parentheses refer to the relative performance in terms of the position in the Statlog ranking out of the 25 methods tested.
‡The data set has unequal misclassification costs.

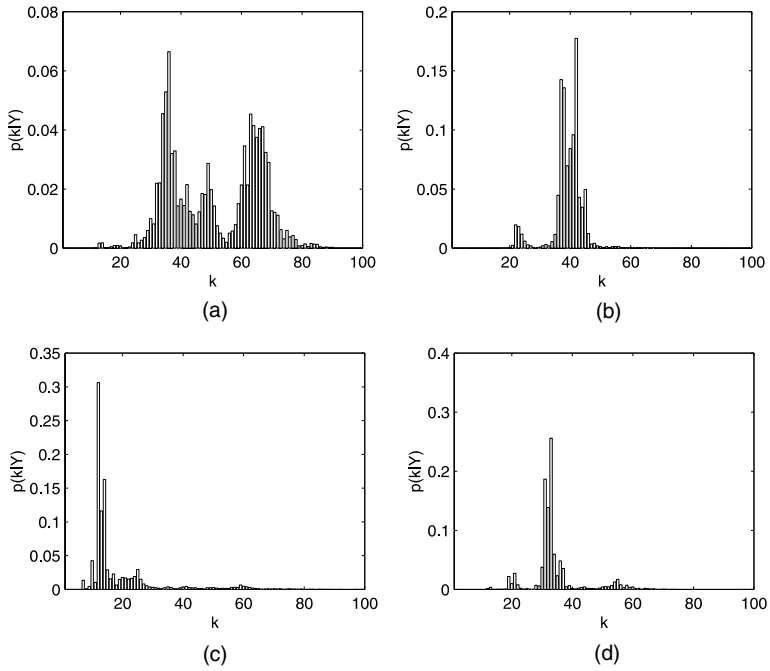

Fig. 5. Plot of $p(k|Y)$ from the example in Section 3.2: (a) Australian credit data ($E(k) = 51.04$; most probable $k = 36$); (b) diabetes data ($E(k) = 39.01$; most probable $k = 42$); (c) heart data ($E(k) = 18.45$; most probable $k = 12$); (d) German credit data ($E(k) = 34.05$; most probable $k = 33$)

questions of the type 'given that I have observed a tree at location (x, y), what is the probability that it is a hickory, maple or oak?'. The posterior class conditional probability fields produced by our method are shown in Figs 6(d)–6(f). The predictive fields in Fig. 6 should be compared with that of Fig. 2. This illustrates that the PNN method, with the same default prior, is capable of fitting very complex probability fields if the classes are strongly overlapping and very simple ones when the classes are reasonably separated.

4. Discussion

The PNN method accounts for a key component of uncertainty that is neglected in the standard k-nearest-neighbour algorithm. The marginalization over the size of the neighbourhood and the level of interaction between neighbours was shown to produce smooth probability fields. The method is nonparametric in that it makes no assumptions about the underlying class conditional distributions $p(\mathbf{x}|y)$. The method is also fully automatic with no user-set parameters and we believe that it is one of the few off-the-shelf Bayesian nonparametric classification methods available. It was shown that the default prior can fit both simple and complex class conditional probability fields depending on the separability of the underlying classes. For many applications we envisage that the MCMC simulation will be unnecessary with all integrals being performed by quadrature methods.

The priors used in this paper were non-informative. However, if knowledge about the likely values of k and β were to exist these could easily be incorporated. Our choice of prior encodes complete ignorance and hence provides a default setting for the automatic use of our model.

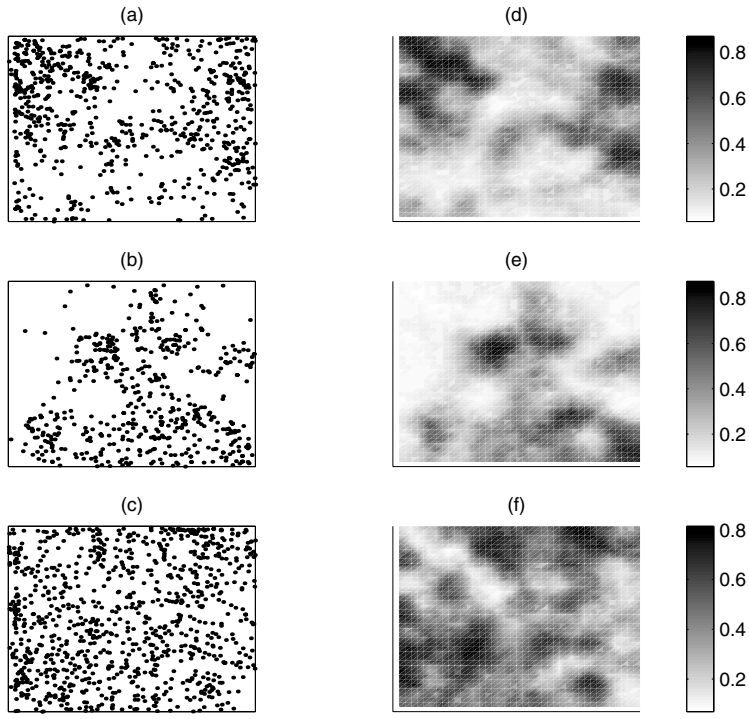

Fig. 6. Location of trees in Lancing Woods: (a) 703 hickories; (b) 514 maples; (c) 929 oaks; (d) predictive distribution p(tree = hickory); (e) predictive distribution p(tree = maple); (f) predictive distribution p(tree = oak) (the posterior expectation of k was 34 for this example)

The posterior distribution of $\{k, \beta\}$ changes as more data become available. In our implementation we specifically condition on t, the current time point, using all information observed up to t. Having made a prediction the actual class labels may be observed leading to an update to $p(k, \beta|Y)$. This would require the re-evaluation of the integrals in equation (3) for future points.

On several bench-mark data sets the method was found empirically to outperform the standard k-nearest-neighbour model. This was especially noticeable for the data sets with unequal misclassification costs where we envisage significant gains in using a proper probabilistic model that can be embedded in a coherent decision process. In on-going research we are investigating the use of various distance metrics, such as the Mahanobolis distance, that can be used to perform the selection of variables or to downweight less relevant predictors. In a subsequent paper Holmes and Adams (2001) investigate a non-Bayesian procedure using multiple values of k and $\mathbf{X}$ within a single model. Holmes and Adams (2001) adopt a penalized likelihood approach and perform a selection of variables to obtain an optimal multiple generalized linear model using a mixture of k-nearest-neighbour terms and original predictor variables.

Acknowledgements

We would like to thank Jane Key for correcting an early version of this paper. Thanks also go to David Denison, two referees and the Joint Editor for their thoughtful comments that led to substantial improvements.

References

Besag, J. E. and Kooperburg, C. (1995) On conditional and intrinsic autoregressions. *Biometrika*, **82**, 733–746.

Besag, J. E., York, J. and Mollié, A. (1991) Bayesian image restoration with two applications in spatial statistics. *Ann. Inst. Math. Statist.*, **43**, 1–59.

Blake, C. L. and Merz, C. J. (1998) UCI repository of machine learning databases. Department of Information and Computer Science, University of California, Irvine. (Available from `http://www.ics.uci.edu/~mlearn/MLRepository.html`.)

Dasarathy, B. V. (1991) *Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques*. Los Alamitos: IEEE Computer Society Press.

Dawid, A. P. (1992) Prequential analysis, stochastic complexity and Bayesian inference (with discussion). In *Bayesian Statistics 4* (eds J. M. Bernardo, J. O. Berger, A. P. Dawid and A. F. M. Smith), pp. 109–125. Oxford: Clarendon.

Devroye, L., Gyorfi, L. and Lugosi, G. (1996) *A Probabilistic Theory of Pattern Recognition*. New York: Springer.

Diggle, P. J. (1983) *Statistical Analysis of Spatial Point Processes*. London: Academic Press.

Enas, G. G. and Choi, S. C. (1986) Choice of the smoothing parameter and efficiency of k-nearest neighbor classification. *Comput. Math. Applic.* A, **12**, 235–244.

Fix, E. and Hodges, J. L. (1951) Discriminatory analysis—nonparametric discrimination: consistency properties. *Project 21-49-004, Report 4*, pp. 261–279. US Air Force School of Aviation Medicine, Randolph Field.

Friedman, J. H. (1997) On bias, variance, 0/1-loss, and the curse-of-dimensionality. *Data Minng Knowl. Discov.*, **1**, 55–77.

Fukunaga, K. and Hostetler, L. D. (1973) Optimisation of k-nearest neighbor density estimates. *IEEE Trans. Inform. Theory*, **19**, 320–326.

Gerrard, D. J. (1969) Competition quotient: a new measure of the competition affecting individual forest trees. *Research Bulletin 20*. Agricultural Experiment Station, Michigan State University, Lansing.

Gilks, W. R. and Berzuini, C. (2001) Following a moving target—Monte Carlo inference for dynamic Bayesian models. *J. R. Statist. Soc.* B, **63**, 127–146.

Gilks, W. R., Richardson, S. and Spiegelhalter, D. J. (eds) (1996) *Markov Chain Monte Carlo in Practice*. London: Chapman and Hall.

Hand, D. J. and Henley, W. E. (1997) Statistical classification methods in consumer credit scoring: a review. *J. R. Statist. Soc.* A, **160**, 523–541.

Holmes, C. C. and Adams, N. M. (2001) Likelihood inference in nearest-neighbour classification models. *Technical Report*. Department of Mathematics, Imperial College of Science, Technology and Medicine, London.

McLachlan, G. J. (1992) *Discriminant Analysis and Statistical Pattern Recognition*. New York: Wiley.

Michie, D., Spiegelhalter, D. J. and Taylor, C. C. (1994) *Machine Learning, Neural and Statistical Classification*. London: Horwood.

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

———(1996) *Pattern Recognition and Neural Networks*. Cambridge: Cambridge University Press.