JIRSS (2011)
Vol. 10, No. 2, pp 167-180

Pseudo-Likelihood Inference Underestimates Model Uncertainty: Evidence from Bayesian Nearest Neighbours

Wanhua Su[1], Hugh Chipman[2], Mu Zhu[3]

[1]Department of Mathematics and Statistics, Grant MacEwan University, Edmonton, Alberta, Canada.
[2]Department of Mathematics and Statistics, Acadia University, Wolfville, Nova Scotia, Canada.
[3] Department of Statistics and Actuarial Science, University of Waterloo, Waterloo, Ontario, Canada.

Abstract. When using the K-nearest neighbours (KNN) method, one often ignores the uncertainty in the choice of K. To account for such uncertainty, Bayesian KNN (BKNN) has been proposed and studied (Holmes and Adams 2002; Cucala *et al.* 2009). We present some evidence to show that the pseudo-likelihood approach for BKNN, even after being corrected by Cucala *et al.* (2009), still significantly underestimates model uncertainty.

Keywords. Bootstrap interval, MCMC, posterior interval, pseudo-likelihood.

MSC: 62C10, 62H30.

1 Introduction

The K-nearest neighbours (KNN) method (e.g., Fix and Hodges 1951; Cover and Hart 1967) is conceptually simple but flexible and useful in

Wanhua Su(SuW3@macewan.ca), Hugh Chipman (✉)(hugh.chipman@acadiau.ca), Mu Zhu(m3zhu@uwaterloo.ca)
Received: May, 2011; August, 2011

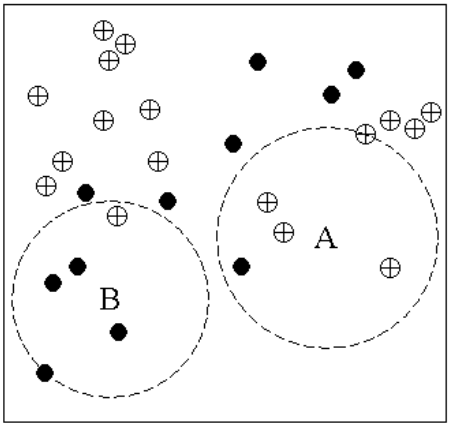

Figure 1: Simulated example illustrating KNN with $K = 5$. Training observations from class 0 are indicated by the symbol "$\oplus$", and those from class 1 are indicated by the symbol "$\bullet$". A and B are two test points.

practice. It can be used for both regression and classification. Like Cucala *et al.* (2009), we focus on classification only.

Under the assumption that points close to one another should have similar responses, KNN classifies a new observation according to the class labels of its K nearest neighbours. In order to identify the neighbours, one must decide how to measure proximity among points and how to define the neighbourhood. The most commonly-used distance metric is the Euclidean distance. The tuning parameter, K, is normally chosen by cross-validation. Figure 1 illustrates how KNN works. Suppose one takes $K = 5$. Then, the possible predicted probabilities are $\{0/5, 1/5, \cdots, 5/5\}$. Among those five nearest neighbours of test point A, four out of five belong to class 0 (C_0) and one out of five belongs to class 1 (C_1). Therefore, A is classified to class 0, and its class probabilities are estimated to be $\hat{P}(A \in C_0) = 4/5$ and $\hat{P}(A \in C_1) = 1/5$. Similarly, test point B is classified to class 1, and its class probabilities are estimated to be $\hat{P}(B \in C_0) = 1/5$ and $\hat{P}(B \in C_1) = 4/5$.

While intuitive and easily implemented, KNN cannot be considered a statistical model. The KNN algorithm lacks a probability model for the observed classes, making inferential statements about parameters such as K or uncertainty statements about predicted probabilities infeasible. Holmes and Adams (2002) developed a pseudo-likelihood representation for the data, enabling BKNN. Cucala *et al.* (2009) identified several shortcomings in Holmes and Adam's approach, proposing a more rig-

orous probability model and several approximate MCMC techniques, including one based on pseudo-likelihood.

Focusing on parameters that are of direct interest in practice, such as $\theta \equiv \Pr(y = 1|\mathbf{x})$, we study the inferential aspect of the pseudo-likelihood approach from a frequentist point of view, that is, by conducting repeated experiments.

Before describing the experiments in Section 3 and results in Section 4, we give a brief review of BKNN model and the pseudo-likelihood approximation. The main result, in Section 4.2, indicates that the coverage of posterior intervals for class probabilities are well below their nominal levels, and the intervals themselves are too narrow. In comparison, bootstrapped KNN gives reasonable coverage and correct interval length. Conclusions in Section 5 offer some hope that the intervals from the pseudo-likelihood approach may still capture some information about uncertainty.

2 Bayesian KNN (BKNN) and a Pseudo-Likelihood Approximation

Holmes and Adams (2002) pointed out that regular KNN does not account for the uncertainty in the choice of K. They presented a Bayesian framework for KNN (BKNN), compared its performance with the regular KNN on several benchmark data sets and concluded that BKNN outperformed KNN in terms of misclassification error.

Cucala *et al.* (2009) presented a more comprehensive treatment of the BKNN model, discussing an exact method for simulation based on perfect sampling. To deal with the computational challenge of evaluating a normalizing constant they propose several computationally feasible approximate simulation methods, including path sampling and Metropolis-Hastings sampling based on a pseudo-likelihood approximation.

To keep this review brief, we focus on the BKNN model of Cucala *et al.* (2009) for Q classes, based on a joint likelihood for all training observations:

$$p(\mathbf{Y}|\mathbf{X}, \beta, K) = \exp\left\{\beta \sum_{i=1}^{n} \sum_{j \in N(\mathbf{x}_i, K)} \mathrm{I}(y_j = y_i)/K\right\} / Z(\beta, K). \quad (1)$$

The indicator function I is 1 whenever its argument is true; the notation "$\sum_{j \in N(\mathbf{x}_i, K)}$" indicates a sum over all observations $\mathbf{x}_j$ that are K-nearest neighbours of $\mathbf{x}_i$; $Z(\beta, K)$ is a normalizing constant.

The central component of (1) is summand $\mathrm{I}(y_j = y_i)/K$, corresponding to the KNN estimate of the probability that observation i takes class y_i. The parameter K controls the number of nearest neighbours and parameter $\beta > 0$ governs the strength of interaction between a data point and its neighbours. A large β tends to enforce strong dependence between neighbours while $\beta = 0$ leads to complete independence. Both Holmes and Adams (2002) and Cucala *et al.* (2009) consider models with interaction parameter β.

Cucala *et al.* (2009) observe that predictive distributions for future y involve integration of a posterior derived from (1). Direct evaluation of such predictive distributions is generally intractable due to the normalizing constant $Z(\beta, K)$. Cucala *et al.* (2009) propose a number of MCMC approximations to the posterior. One approximation involves MCMC with a pseudo-likelihood function replacing (1). This approach is similar to Holmes and Adams (2002), but as pointed out by Cucala *et al.* (2009), the correct pseudo likelihood function should be

$$p(\mathbf{Y}|\mathbf{X}, \beta, K) = \prod_{i=1}^{n} p(y_i|\mathbf{x}_i, \beta, K) = \prod_{i=1}^{n} \frac{\exp\left\{(\beta/K)\left(\sum_{j\in N(\mathbf{x}_i,K)} \mathrm{I}(y_j = y_i) + \sum_{i\in N(\mathbf{x}_j,K)} \mathrm{I}(y_i = y_j)\right)\right\}}{\sum_{q=1}^{Q} \exp\left\{(\beta/K)\left(\sum_{j\in N(\mathbf{x}_i,K)} \mathrm{I}(y_j = q) + \sum_{i\in N(\mathbf{x}_j,K)} \mathrm{I}(q = y_j)\right)\right\}}. \tag{2}$$

The additional notation "$\sum_{i\in N(\mathbf{x}_j,K)}$" indicates a sum over observations $\mathbf{x}_j$ that have $\mathbf{x}_i$ as one of their K-nearest neighbours. This second term was missing in Holmes and Adams (2002), but it is needed because the neighbouring relationship between any two observations is generally not symmetric; the fact that $\mathbf{x}_j$ is the nearest neighbour of $\mathbf{x}_i$ does not imply $\mathbf{x}_i$ is necessarily the nearest neighbour of $\mathbf{x}_j$ (Cucala *et al.* 2009).

Unlike regular likelihood functions, the component for data point y_i depends on the class labels of other data points y_j, for $j \neq i$. Treating β and K as random variables, the marginal predictive distribution for a new data point $(\mathbf{x}_{n+1}, y_{n+1})$ based on the training data $(\mathbf{X}, \mathbf{Y})$ is given by

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

where $p(\beta, K|\mathbf{X}, \mathbf{Y}) \propto p(\mathbf{Y}|\mathbf{X}, \beta, K)p(\beta, K)$ is the posterior distribution of (β, K). Holmes and Adams (2002) adopt uniform prior distributions

for K and β. A random-walk Metropolis-Hastings algorithm is then used to draw M samples from the posterior $p(\beta, K|\mathbf{X}, \mathbf{Y})$, so that (3) can be evaluated. We stress that since pseudo-likelihood approximation (2) is used, these samples are from an approximate posterior, not the correct posterior. The next two sections study the impact of the pseudo-likelihood approximation on predictive intervals.

3 Experiments

We now describe an experiment that shows the pseudo-likelihood approach still significantly underestimates model uncertainty. For the reminder of this article, the acronym "BKNN" will be strictly used to refer to the pseudo-likelihood approach of Bayesian KNN.

The following experiment is repeated 100 times. Each time, we first generate $n = 250$ pairs of training data from a known, two-class model (details in Section 3.1). We then fit BKNN and regular KNN on the training data, and let them make predictions at a set of 160 pre-selected test points (details in Section 3.2). For each test point, say $(\mathbf{x}_{n+1}, y_{n+1})$, our parameter of interest is

$$\theta_{n+1} \equiv \Pr(y_{n+1} = 1|\mathbf{x}_{n+1}). \tag{4}$$

We construct both point estimates (Section 4.1) and interval estimates (Section 4.2) of θ_{n+1}: $\hat{\theta}_{n+1}$ and $\hat{I}_{n+1}$.

To fit BKNN, we use the Matlab code provided by Holmes and Adams (2002) — except we use the corrected expression for $p(\mathbf{Y}|\mathbf{X}, \beta, K)$; see (2). To fit regular KNN, we use the `knn` function in R.

3.1 Simulation Model

Both Holmes and Adams (2002) and Cucala *et al.* (2009) made heavy use of a synthetic dataset consisting of 250 training and 1000 test points, taken from `http://www.stats.ox.ac.uk/pub/PRNN`. These data were originally generated from two classes, each being an equal mixture of two bivariate normal (BVN) distributions. In order to be able to generate slightly different training data every time we repeat our experiment, we imitate this synthetic data set by assuming the underlying distributions of class 1 (C_1) and class 0 (C_0) to be:

$$\begin{aligned}
\mathbf{x}|C_1 &\sim f_1(\mathbf{x}) = 0.5\text{BVN}\left(\boldsymbol{\mu}_{11}, \boldsymbol{\Sigma}\right) + 0.5\text{BVN}\left(\boldsymbol{\mu}_{12}, \boldsymbol{\Sigma}\right) \\
\mathbf{x}|C_0 &\sim f_0(\mathbf{x}) = 0.5\text{BVN}\left(\boldsymbol{\mu}_{01}, \boldsymbol{\Sigma}\right) + 0.5\text{BVN}\left(\boldsymbol{\mu}_{02}, \boldsymbol{\Sigma}\right),
\end{aligned}$$

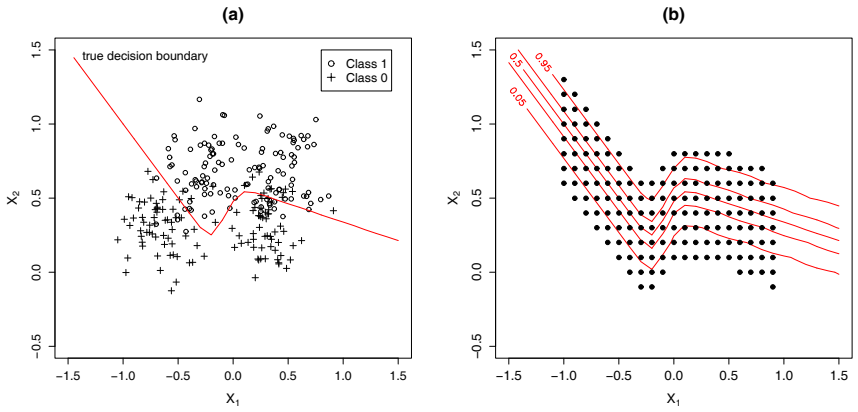

Figure 2: (a) Training data from one experiment, and the decision boundary given by (5). (b) The fixed set of test points, and the true probability contour.

with

$$\boldsymbol{\mu}_{11} = \begin{pmatrix} -0.3 \\ 0.7 \end{pmatrix}, \ \boldsymbol{\mu}_{12} = \begin{pmatrix} 0.4 \\ 0.7 \end{pmatrix}, \ \boldsymbol{\mu}_{01} = \begin{pmatrix} -0.7 \\ 0.3 \end{pmatrix}, \ \boldsymbol{\mu}_{02} = \begin{pmatrix} 0.3 \\ 0.3 \end{pmatrix}$$

and

$$\boldsymbol{\Sigma} = \begin{pmatrix} 0.03 & 0 \\ 0 & 0.03 \end{pmatrix}.$$

The prior class probabilities are taken to be equal, i.e., $\Pr(y = 1) = \Pr(y = 0) = 0.5$. Given any data point $\mathbf{x}$, its posterior probability of being in C_1 can be calculated by Bayes' rule

$$\Pr(y = 1|\mathbf{x}) = \frac{0.5 f_1(\mathbf{x})}{0.5 f_1(\mathbf{x}) + 0.5 f_0(\mathbf{x})}. \tag{5}$$

Figure 2(a) shows the training data from one experiment and the true decision boundary.

The model from which the data are simulated is different from BKNN model (2). Direct simulation from the BKNN model is not possible, since $p(y_i|\mathbf{x}_i, \beta, K)$ depends on the labels of neighbouring points y_j. That is, without any y values, new y values cannot be simulated.

3.2 Test Points

Instead of focusing on the total misclassification error, we focus on predictions made at a *fixed* set of 160 test points. The test points are chosen

to cover the critical part of the true posterior probability contour as indicated in Figure 2(b). We shall refer to θ_{n+1} as the key parameter of interest, but it should be understood that the subscript "$n+1$" is used to refer to any of the 160 test points.

4 Results

We now compare both point estimates and interval estimates produced by the ordinary KNN method and the Bayesian KNN method (using the pseudo-likelihood approach).

4.1 Point Estimates of θ_{n+1}

We begin with point estimates. For BKNN, the point estimate of $\theta_{n+1} \equiv \Pr(y_{n+1} = 1|\mathbf{x}_{n+1})$ is the posterior mean:

$$\hat{\theta}_{n+1}^{BKNN} = \frac{1}{M}\sum_{j=1}^{M} \Pr(y_{n+1} = 1|\mathbf{x}_{n+1}, \mathbf{X}, \mathbf{Y}, \beta^{(j)}, K^{(j)}),$$

where $(K^{(j)}, \beta^{(j)})$ are samples drawn from the posterior distribution, $p(K, \beta|\mathbf{X}, \mathbf{Y})$. For regular KNN, one chooses the parameter K by cross-validation, and normally uses the original KNN score

$$\tilde{\theta}_{n+1}^{KNN} = \frac{1}{K}\sum_{j \in N(\mathbf{x}_{n+1}, K)} \mathrm{I}(y_j = 1) \tag{6}$$

as the point estimate.

Through experimentation in this example we discovered that the KNN scores (6) were not well-calibrated probability estimates, and the use of a logistic transformation improved this accuracy. Thus we report

$$\hat{\theta}_{n+1}^{KNN} = \frac{\exp\{\hat{\alpha} + \hat{\beta}\tilde{\theta}_{n+1}^{KNN}\}}{1 + \exp\{\hat{\alpha} + \hat{\beta}\tilde{\theta}_{n+1}^{KNN}\}} \tag{7}$$

as the point estimate of regular KNN. Let

$$g(y_i) = \frac{1}{K}\sum_{j \in N(\mathbf{x}_i, K)} \mathrm{I}(y_j = y_i), \tag{8}$$

be the output of KNN. In (7), $\hat{\alpha}$ and $\hat{\beta}$ are obtained by running a logistic regression of y_i onto $g(y_i)$ using the training data. Notice that the

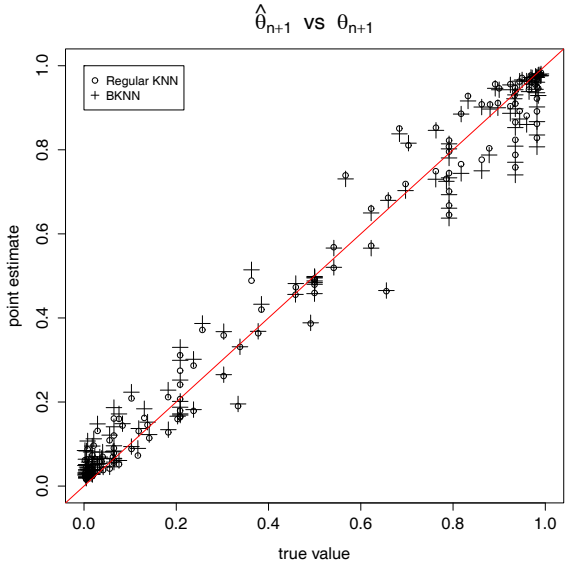

Figure 3: Average of 100 point estimates versus the true parameter value, for all 160 test points. A 45-degree reference line going through the origin is also displayed.

calibrated KNN estimate given by (7) above is also more comparable to the one given by (2). The logistic form (7) is slightly superior to simply using KNN output $g(y_i)$.

After repeating the experiment 100 times, we obtain 100 slightly different point estimates at each $\mathbf{x}_{n+1}$. Figure 3 plots the average of these 100 point estimates against the true value for all 160 test points. We see that both BKNN and regular KNN give very similar point estimates.

4.2 Interval Estimates of θ_{n+1}

The main focus of our experiments is interval estimation. In particular, we are interested in the question of whether these interval estimates adequately capture model uncertainty.

For BKNN, we use the 95% posterior (or credible) interval as our interval estimate, $\hat{I}_{n+1}^{BKNN}$. This is constructed by finding the 2.5th and 97.5th percentiles of the posterior samples. To obtain an interval estimate for regular KNN, $\hat{I}_{n+1}^{KNN}$, we resort to Efron's bootstrap. Given a training set, $\mathcal{D}$, we generate 500 bootstrap samples, $\mathcal{D}_1^*, \mathcal{D}_2^*, \cdots, \mathcal{D}_{500}^*$, and repeat the entire KNN model building process — that is, choosing K by cross-validation and calculating $\hat{\theta}_{n+1,b}$ according to (7) — for every

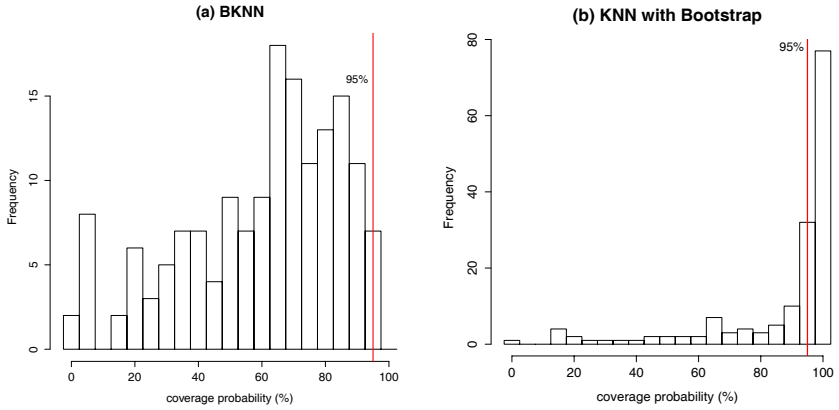

Figure 4: Estimated coverage probabilities of (a) $\hat{I}_{n+1}^{BKNN}$ and (b) $\hat{I}_{n+1}^{KNN}$, for all 160 test points.

$\mathcal{D}_b^*$, $b = 1, 2, \cdots, 500$. The interval estimate of θ_{n+1} is constructed by taking the 2.5th and 97.5th percentiles of the set, $\{\hat{\theta}_{n+1,1}, \cdots, \hat{\theta}_{n+1,500}\}$.

Our first question of interest is: What are the coverage probabilities of $\hat{I}_{n+1}^{KNN}$ and $\hat{I}_{n+1}^{BKNN}$? After repeating the experiment 100 times, we obtain 100 slightly different interval estimates, produced from 100 slightly different training sets. The coverage probability of $\hat{I}_{n+1}^{BKNN}$ (and that of $\hat{I}_{n+1}^{KNN}$) can be estimated easily by counting the number of times θ_{n+1} is included in the interval over the 100 experiments. Histograms of the estimated coverage probabilities for all 160 test points are shown in Figure 4. The posterior intervals produced by BKNN can easily be seen to have fairly poor coverage overall.

For each interval estimate, we also calculate its length, e.g.,

$$\begin{aligned} \text{length}_{n+1}^{BKNN} &= \left| \hat{\theta}_{n+1}^{BKNN,97.5\%} - \hat{\theta}_{n+1}^{BKNN,2.5\%} \right|, \\ \text{length}_{n+1}^{KNN} &= \left| \hat{\theta}_{n+1}^{KNN,97.5\%} - \hat{\theta}_{n+1}^{KNN,2.5\%} \right|. \end{aligned}$$

Let

$$\overline{\text{length}}_{n+1}^{BKNN} \quad \text{and} \quad \overline{\text{length}}_{n+1}^{KNN}$$

be the average lengths of these 100 interval estimates. Our next question of interest is: Are they too long, too short, or just right? In order to answer this question, we need a "gold standard".

The very reason for using these interval estimates is to reflect that there is uncertainty in our estimate of the underlying parameter, θ_{n+1}.

This uncertainty is easy to assess directly when one can repeatedly generate different sets of training data and repeatedly estimate the parameter, which is exactly what we have done. The standard deviations of the 100 point estimates (Section 4.1), which we write as

$$\text{std}(\hat{\theta}_{n+1}^{BKNN}) \quad \text{and} \quad \text{std}(\hat{\theta}_{n+1}^{KNN}),$$

give us a direct assessment of this uncertainty.

If the point estimates, $\hat{\theta}_{n+1}^{BKNN}$ and $\hat{\theta}_{n+1}^{KNN}$, are approximately normally distributed, then the correct lengths of the corresponding interval estimates should be roughly 4 times the aforementioned standard deviation, that is,

$$\overline{\text{length}}_{n+1}^{BKNN} \approx 4 \times \text{std}(\hat{\theta}_{n+1}^{BKNN}), \tag{9}$$

$$\overline{\text{length}}_{n+1}^{KNN} \approx 4 \times \text{std}(\hat{\theta}_{n+1}^{KNN}). \tag{10}$$

We use (9)-(10) as *heuristic* guidelines to assess how well the interval estimates can capture model uncertainty, despite lack of formal justification for the normal approximation. Figure 5 provides a schematic illustration of our assessment protocol.

Figure 6 plots the average lengths of these 100 interval estimates against 4 times the standard deviations of the corresponding point estimates — that is, $\overline{\text{length}}_{n+1}^{BKNN}$ against $4 \times \text{std}(\hat{\theta}_{n+1}^{BKNN})$ and $\overline{\text{length}}_{n+1}^{KNN}$ against $4 \times \text{std}(\hat{\theta}_{n+1}^{KNN})$ — for all 160 test points. Here, it is easy to see that the Bayesian posterior intervals are apparently too short, whereas bootstrapping regular KNN gives a more accurate and slightly more conservative assessment of the amount of uncertainty in the point estimate.

5 Discussion

The results in Section 4 complement the findings of Cucala *et al.* (2009), who observe that MCMC based on pseudo-likelihood delivers inferior inferences for K and β. We demonstrate that this problem is also manifest in intervals for class probabilities, with the intervals being too narrow and providing coverage below the nominal level.

Why does the pseudo-likelihood approach underestimate uncertainty? We believe it may be because the model only accounts for the uncertainty in the *number* of neighbours (i.e., the parameter K), but it is unable to account for the uncertainty in the spatial *locations* of these neighbours. When the same model (fit via a combination of maximum likelihood and cross-validation) is bootstrapped, spatial locations are made part of

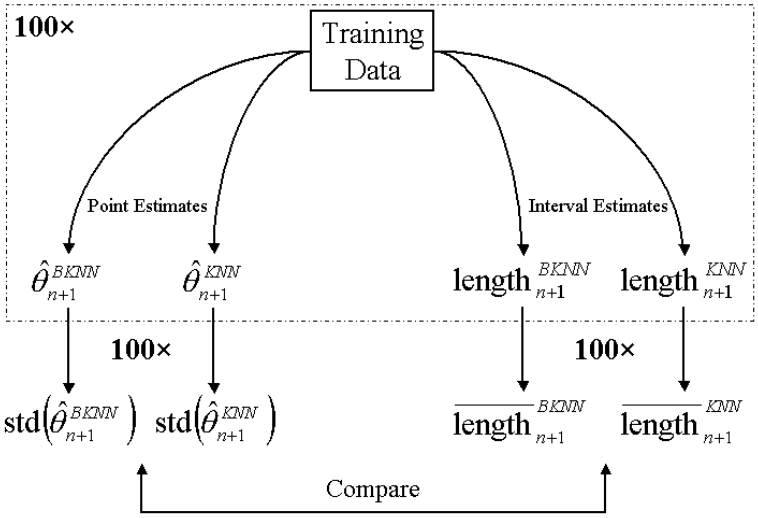

Figure 5: Schematic illustration of our assessment protocol. Variation over 100 point estimates is used as a benchmark to assess the quality of the corresponding interval estimates.

the sampling distribution, and resultant inferences seem more plausible. Perhaps the less tractable correct likelihood (1), by simultaneously considering all training cases, more accurately captures spatial information. Superior results for methods that Cucala *et al.* (2009) characterize as better approximations of (1) hint that this may be the case.

The inability to assess spatial uncertainty is a general phenomenon associated with pseudo-likelihood functions. Pseudo-likelihood functions were first introduced by Besag (1974, 1975) to model spatial interactions in lattice systems. Since then, they have been widely used in image processing (e.g., Besag 1986) and network tomography (e.g., Strauss and Ikeda 1990; Liang and Yu 2003; Robins *et al.* 2007). However, statistical inference based on pseudo-likelihood functions is still in its infancy. Some researchers argue that pseudo-likelihood inference can be problematic since it ignores at least part of the dependence structure in the data. In applications to model social networks, a number of researchers, such as Wasserman and Robins (2005) and Snijders (2002), have pointed out that maximum pseudo-likelihood estimates are substantially biased and the standard errors of the parameters are generally underestimated. For BKNN, the pseudo-likelihood function (2) clearly ignores the fact that

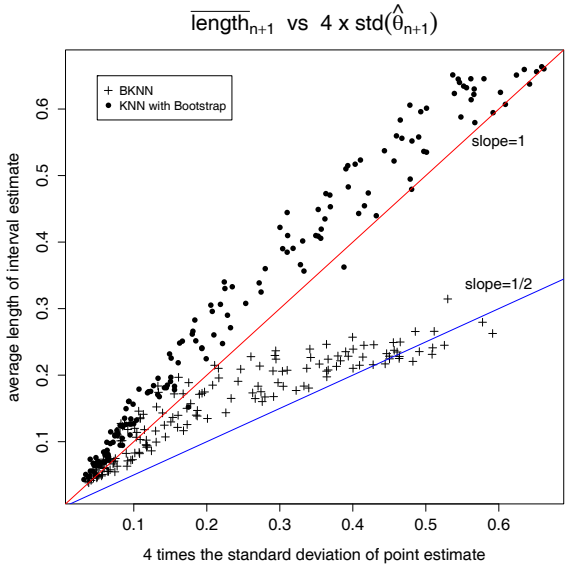

Figure 6: Average length of 100 interval estimates versus 4 times the standard deviation of the corresponding point estimate, for all 160 test points. Two reference lines – both going through the origin, one with slope=1 and another with slope=1/2 — are also displayed.

the locations of one's neighbours are also random, not just the number of neighbours.

One interesting observation from Figure 6 is the fact that

$$\overline{\text{length}}_{n+1}^{BKNN} \approx 2 \times \text{std}(\hat{\theta}_{n+1}^{BKNN}).$$

If we continue to use $4 \times \text{std}(\hat{\theta}_{n+1}^{BKNN})$ as the "gold standard", then these Bayesian posterior intervals are about half as long as they should be. We have observed this phenomenon on other examples, too, but do not yet have an explanation for it. However, this suggests that it *may* be possible to make simple corrections to the standard error estimates produced by the pseudo-likelihood.

Despite the fact that BKNN seems to underestimate overall uncertainty, that $\overline{\text{length}}_{n+1}^{BKNN}$ is still approximately proportional to $\text{std}(\hat{\theta}_{n+1}^{BKNN})$ suggests that we can still rely on it to assess the *relative* uncertainty of its predictions. For many practical problems, this is still very useful. For example, if two accounts, A and B, are both predicted to be fraudulent with a high probability of 0.9 but the posterior interval of A is twice as long as that of B, then it is natural for a financial

institution to spend its limited resources investigating account B rather than account A.

Acknowledgment

This research is partially supported by the Natural Science and Engineering Research Council (NSERC) of Canada, Canada's National Institute for Complex Data Structures (NICDS) and the Mathematics of Information Technology And Complex Systems (MITACS) network.

References

Besag, J. (1974), Spatial interaction and the statistical analysis of lattice systems (with discussion). Journal of Royal Statistical Society, Series. B, **36**(2), 192–236.

Besag, J. (1975), Statistical analysis of non-lattice data. The Statistician, **24**(3), 179–195.

Besag, J. (1986), On the statistical analysis of dirty pictures. Journal of Royal Statistical Society, Series. B, **48**(3), 259–302.

Cover, T. and Hart, P. (1967), Nearest neighbor pattern classification. IEEE Transactions on Information Theory, **IT-13**, 21–27.

Cucala, L., Marin, J. M., Robert, C. P., and Titterington, D. M. (2009), A Bayesian reassessment of nearest-neighbor classification. Journal of the American Statistical Association, **104**(485), 263–273.

Fix, E. and Hodges, J. L. (1951), Discriminatory analysis–nonparametric discrimination: Consistency properties. Technical report, USAF School of Aviation Medicine, Texas: Randolph Field.

Holmes, C. C. and Adams, N. M. (2002), A probabilistic nearest neighbour method for statistical pattern recognition. Journal of Royal Statistical Society, Series. B, **64**(2), 295–306.

Liang, G. and Yu, B. (2003), Maximum pseudo likelihood estimation in network tomography. IEEE Transactions on Signal Processing, **51**, 2043–2053.

Robins, G., Pattison, P., Kalish, Y., and Lusher, D. (2007), An introduction to exponential random graph (P^*) models for social networks. Social Networks, **29**, 173–191.

Snijders, T. A. B. (2002), Markov chain monte carlo estimation of exponential random graph model. Journal of Social Structure, **3**(2).

Strauss, D. and Ikeda, M. (1990), Pseudolikelihood estimation for social networks. Journal of the American Statistical Association, **85**, 204–212.

Wasserman, S. S. and Robins, G. L. (2005), An introduction to random graphs, dependence graphs, and p^*. In J. S. P. Carrrington and S. S. Wasserman, editors, Models and Methods in Social Network Analysis, pages 148–161. Cambridge: Cambridge University Press.