A Bayesian reassessment of nearest-neighbor classification

Maxime Duval-Prevosteau
University of Lille
maxime.duval@centrale.centralelille.com

Karim Zaidi
University of Lille
karim.zaidi@centrale.centralelille.com

Abstract

We aim to reproduce the experiments of the article A Bayesian reassessment of nearest-neighbor classification [1]. In this article, the authors introduce a clear probabilistic basis for the k-nearestneighbour procedure from which they derive computational tools for conducting Bayesian inference on the parameters of the model.

1 Summary

1.1 Introduction, goal and plan

The k-nearest-neighbour method is a well-established and straightforward technique in this area with both a long past and a fairly resilient resistance to change. However, it lacks a corresponding assessment of its classification error.

This paper proposes a global probabilistic model that encapsulates the k-nearest-neighbour method. They then derive a fully operationnal simulation technique adapted to their model.

1.2 Probabilistic model

The first approach consists in taking the advantage of the spatial structure of the problem and to use Markov random field. They uses a Boltzmann distribution with potential:

$$\sum_{\ell \sim_k i} \delta_{y_i}\left(y_\ell\right)$$

where $\ell \sim_k i$ means that the summation is taken over the observations x_ℓ belonging to the k nearest neighbours of x_i, and $\delta_a(b)$ denotes the Dirac function. This function basically gives the number of points from the same class y_i as the point x_i that are among the k nearest neighbours of x_i.

One could be tempted to take the following probability model:

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

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

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

In this parameterised model, β is a parameter that represent the degree of uncertainty of our model:

- $\beta = 0$ corresponds to a uniform distribution over all classes, meaning independence from the neighbours
- $\beta = +\infty$ leads to a point mass distribution at the prevalent class, corresponding to extreme dependence

However, it leads to an incoherent model due to the fact that $\ell \sim_k i$ is not a symmetric relation: if x_i is one of the k nearest neighbours of x_j then x_j is not necessarilty one the k nearest neighbours of x_i.

1.3 A symmetrised Boltzmann modelling

We thus consider the marginal

$$f(\mathbf{y} \mid \mathbf{X}, \beta, k) = \exp\left(\beta \sum_i \sum_{\ell \sim_k i} \delta_{y_i}(y_\ell)/k\right)/Z(\beta, k) \tag{1}$$

Where $Z(\beta, k)$ is the normalising constant.

Thus, the full conditional distribution of 1 can be written as:

$$f(y_i \mid \mathbf{y}_{-i}, \mathbf{X}, \beta, k) \propto \exp\left\{\beta/k\left(\sum_{\ell \sim_k i} \delta_{y_i}(y_\ell) + \sum_{i \sim_k \ell} \delta_{y_\ell}(y_i)\right)\right\} \tag{2}$$

where $\ell \sim_k i$ means that the summation is taken over the observations x_ℓ belonging to the k nearest neighbours of x_i.

However, the normalising constant $Z(\beta, k)$ is intractable (except for the most trivial cases).

1.4 Predictive perspective

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

$$\mathbb{P}(y_{n+1} = g \mid x_{n+1}, \mathbf{y}, \mathbf{X}, \beta, k) \propto \exp\left\{\beta/k\left(\sum_{\ell \sim_k (n+1)} \delta_g(y_\ell) + \sum_{(n+1) \sim_k \ell} \delta_{y_\ell}(g)\right)\right\} \tag{3}$$

where

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

are the numbers of observations in the training dataset from class g among the k nearest neighbours of x_{n+1} and among the observations for which x_{n+1} is a k-nearest neighbour, respectively.

1.5 Bayesian inference and the normalisation problem

The authors has chosen for (k, β) a uniform prior on the compact support $\{1, \cdots, K\} \times [0, \beta_{\max}]$.

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, $\min(n_1, n_2)$ and the limitation on β, $\beta < \beta_{\max}$,is legitimate because above a certain value, the model becomes "all black or all white" (this is known as phase-transition phenomena (Møller, 2003)).

The determination of $\beta_{\max}$ is problem-specific and needs to be solved afresh for each new dataset.

1.6 MCMC steps

The posterior distribution $\pi(\beta, k|\mathbf{y}, \mathbf{X})$ is computed by random walk Metropolis-Hastings.

Since $\beta \in (0, \beta_{\max})$ is constrained, the authors reparametrized β by $\theta \in \mathbb{R}$:

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

and then proposed to do a normal random walk on the θ' s, $\theta' \sim \mathcal{N}\left(\theta^{(t)}, \tau^2\right)$.

For k, the authors proposed a uniform proposal on the $2r$ neighbours of $k^{(t)}$, namely $\left\{k^{(t)} - r, \ldots, k^{(t)} - 1, k^{(t)} + 1, \ldots k^{(t)} + r\right\} \bigcap \{1, \ldots, K\}$.

This proposal distribution with probabiltity density $Q_r(k, \cdot)$, with $k' \sim Q_r\left(k^{(t-1)}, \cdot\right)$, thus depends on a parameter $r \in \{1, \ldots, K\}$ that needs to be calibrated so as to aim at optimal acceptance rates, as does τ^2. The acceptance probability in the Metropolis-Hastings algorithm is thus

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

where the second ratio is the ratio of the Jacobians due to reparametrisation.

1.7 Pseudo-likelihood approximation

Since the conditional distribution of $\mathbf{y}$, $f(\mathbf{y}|\mathbf{X}, \beta, k)$ requires the computation of $Z(\beta, k)$ which is intractable constant. Thus, we cannot use the Metropolis-Hasting algorithm.

The first approach proposed by the authors to get around this issue is to replace $f(\mathbf{y}|\mathbf{X}, \beta, k)$ by an approximation, the pseudo-likelihood defined as:

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

1.8 Path sampling

The second approach consists in using the true conditional distribution of $\mathbf{y}$, $f(\mathbf{y}|\mathbf{X}, \beta, k)$ but approximating $\log Z(\beta, k)$ using Monte Carlo techniques which results in:

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

To reduce the computational cost of this technique, the authors used the regularity of $Z(\beta, k)$ to only calculate it for some values of β and then numerically interpolate to extend the function to other values of β.

1.9 Perfect sampling implementation and Gibbs approximation

The third approach consists in introducing an auxiliary variable $\mathbf{z}$ on the same state space as $\mathbf{y}$ whose arbitrary conditional density $g(\mathbf{z}|\beta, k, \mathbf{y})$ is chosen as to cancel the normalization in the calculation of the MCMC acceptance probability. The authors of this method advocate the choice:

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

as reasonable, where $(\hat{\beta}, \hat{k})$ is a preliminary estimate, such as the maximum pseudo-likelihood estimate.

The major drawback of this technique is that the auxiliary variable z must be simulated from the distribution $f(z|\beta, k)$ itself. To circumvent this problem, in the case of binary classification task ($G = 2$) it is possible to simulate $\mathbf{z}$ using a monotone implementation of the Gibbs sampler.

To avoid explosion in the computational time, the authors introduced an additional accept-reject step based on smaller values of β. However, the obtained method is very sensitive on the value of $(\hat{\beta}, \hat{k})$.

2 Critic of the paper

We found that this paper was really interesting as it gives new insights to this old method by giving a coherent probabilistic framework and tools to actually implement "effectively" their method. It is also well written (well structured, good english, ...) so it make the paper quite easy to read. The authors also explained quite in details the theoretical aspects and even provide the assumptions/limitations of their paper.

However, it seems like the authors are quite stingy with implementation details (choice of $(\hat{\beta}, \hat{k})$ which the last method heavily depends on, details of the techniques, ...) so it is not really reproducible.

Moreover, the method seems a bit computationally heavy to be practical as it required $O(n^2 d)$ computation instead of $O(nd)$ for the vanilla KNN (and even if this method doesn't require the tuning of k as for the vanilla KNN, we still need to determine $\beta_{\max}$ and $(\hat{\beta}, \hat{k})$).

Perhaps those two reasons explain why this paper hasn't gain much popularity (according to the weak number of citations in Google scholar).

On a lighter note, we report 3 typos on the writing of the Metropolis-Hastings ratio in 3.4 (they forgot some ' for k and z).

3 Contribution

3.1 Python implementation

All the code can be found in the notebook. Unfortunately, we were only able to fully re-implement the first method. Although we produced the code for the second method, it simply took too much computation time (in particular to compute the grid of values of $\log Z$) and we were not able to try it in practice. We could not implement the third method because there were not enough implementation details in the article.

3.2 Time comparison with classical KNN

Although it is not mentionned in the article, it is obvious that the classical KNN method is much more efficient in practice. For instance, just computing the 50,000 (β, k) couples with the Metropolis-Hastings algorithm with the first method took us more than an hour, and then computing the test score on the 1000 test samples took another half hour. On the other hand, applying the KNN requires a single step which is almost instantaneous.

There is no point in getting more quantitative than this. In practice, the methods (this also applies to the second and third methods of the paper) described above are not viable.

3.3 Changing the prior on β

We decided to observe the effect of changing the prior distribution on (β, k). In particular, we changed the prior of β from a uniform distribution on $[0, \beta_{max}]$ to a parabolic distribution (i.e. a particular case of symmetric beta distribution that looks like an upside down parabola). In theory, changing this prior should have little to no influence on the resulting score. In practice, while with the uniform prior the score on the test set is 0.916 (as computed in the notebook), the score that results from the parabolic prior is 0.917. we therefore observer that, indeed, changing the prior has little effect on the score, which indicates the prior sensitivity is not too high and should therefore not be an issue (at least for this particular data set).

References

[1] Cucala, Lionel, et al. "A Bayesian reassessment of nearest-neighbor classification." Journal of the American Statistical Association 104.485 (2009): 263-273.