Revisiting of Bayesian reassessment of nearest-neighbor classification

Charbel-Raphaël Segerie
Master MVA
ENS Paris-Saclay
crsegerie@gmail.com

Etienne Peyrot
Master MVA
ENS Paris-Saclay
etienne.peyrot@gmail.com

Abstract

We studied the article "Bayesian reassessment of nearest-neighbor classification" by Lionel Cucala & al. (2009) [0]. After reproducing the results, we extended the applicability framework. Our first and second parts reproduce the numerical results of the article while explaining the main theoretical points. Our last part is a theoretical complement allowing to generalize the inference rules as well as to criticize the starting hypotheses in the choice of an energy in the form of a Potts model.

1 A probabilistic k-nearest-neighbour model

1.1 Goal and Plan

The k-nearest neighbor algorithm is very practical, but its theoretical justifications are not well understood.

We have searched by ourselves in the literature for some elements of k-nearest neighbor theory, but the statistical learning literature is more interested in asymptotic properties of consistencies of the classification rule, which can be seen as a Voronoi partitioning of the space and which has good properties such as Stone's lemma or consistency when k and n tend simultaneously to $+\infty$.[4]

This paper proposes to base the k-nearest neighbor model on a probabilistic model. Once well posed, the probabilistic model would allow to find confidence intervals and would allow to better understand the k-nearest neighbor model. The motivation is also to get rid of meta parameters like the number k of neighbors to consider.

1.2 Probabilistic model with underlying energy

The original paper uses an energy approach using a Boltzmann model with an underlying energy resulting from a Potts model with a potential energy in the form :

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

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. One could then naively take a probability model in the form :

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

Preprint. Under review.

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

In this model, β is a meta-parameter to be calibrated, and this meta-parameter will represent the degree of uncertainty of our model: a low β, will represent a high uncertainty (a maximum uncertainty for a $\beta = 0$ and thus a uniform distribution) and a high certainty for large β.

1.3 A symmetrised Boltzmann modelling

But the modeling exposed in the previous paragraph is incoherent, because the k-nearest neighbor relation $\ell \sim_k i$ is not a symmetric relation. We then need to consider not the marginal probability with respect to a point but the probability of the complete set of points:

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

where $Z(\beta, k)$ is the normalising constant of the distribution. The full conditional distributions corresponding to (2) can be written as

$$f(y_i|\mathbf{y}_{-i}, \mathbf{X}, \beta, k) \propto \exp\left\{\beta/k \left(\sum_{\ell \sim_k i} \delta_{y_i}(y_\ell) + \sum_{i \sim_k \ell} \delta_{y_\ell}(y_i)\right)\right\}\,, \tag{3}$$

where $i \sim_k \ell$ means that the summation is taken over the observations x_ℓ for which x_i is a k-nearest neighbour. But then we have two problems : the normalising constant $Z(\beta, k)$ is intractable, and depends on β and k.

1.4 Bayesian inference and the normalisation problem

When based on the conditional expression (3), the predictive distribution of a new unclassified observation y_{n+1} given its covariate x_{n+1} and the training sample $(\mathbf{y}, \mathbf{X})$ is, for $g = 1, \ldots, G$,

$$\mathbb{P}(y_{n+1} = g|x_{n+1}, \mathbf{y}, \mathbf{X}, \beta, k) \propto \exp\left\{\beta/k \left(\sum_{\ell \sim_k (n+1)} \delta_g(y_\ell) + \sum_{(n+1) \sim_k \ell} \delta_{y_\ell}(g)\right)\right\}\,, \tag{4}$$

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.

1.4.1 MCMC steps

The paper proposes to compute the posterior distribution $\pi(\beta, k|\mathbf{y}, \mathbf{X})$ by Metropolis-Hasting : both β and k are updated using random walk proposals.
The authors propose to upper bounds β by $\beta_{\max}$, which allows them to formulate a reparametrization of β by θ, this last variable being then defined on $\mathbb{R}$ as follows :

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

We can then simulate a Gaussian random walk on θ, $\theta' \sim \mathcal{N}(\theta^{(t)}, \tau^2)$. For k, we use instead a uniform proposal on the $2r$ neighbours of $k^{(t)}$, namely $\{k^{(t)} - r, \ldots, k^{(t)} - 1, k^{(t)} + 1, \ldots k^{(t)} + r\} \bigcap \{1, \ldots, K\}$. This proposal distribution with probabiltity 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 so as to aim at optimal acceptance rates, as does τ^2. The acceptance probability in the Metropolis–Hastings algorithm is thus

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

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

The upper bounding of β by $\beta_{\max}$ is legitimate because beyond a threshold, the model favors ditributions of y all either black or white. But we will have to find this threshold in custom models.

1.4.2 Pseudo-likelihood approximation

It is not possible to use the Metropolis-Hasting algorithm because the conditional density of $\mathbf{y}$, $f(\mathbf{y} \mid \mathbf{X}, \beta, k)$, requires the computation of the normalization constant $Z(\beta, k)$. The first approach proposed by the paper [1] gets around this problem by replacing $f(\mathbf{y} \mid \mathbf{X}, \beta, k)$ by an approximation, the pseudo-likelihood :

$$\hat{f}(\mathbf{y}|\mathbf{X}, \beta, k) = \prod_{i=1}^{n} \frac{\exp\left\{\beta/k \left(\sum_{\ell \sim_k i} \delta_{y_i}(y_\ell) + \sum_{i \sim_k \ell} \delta_{y_\ell}(y_i)\right)\right\}}{\sum_{g=1}^{2} \exp\left\{\beta/k \left(\sum_{\ell \sim_k i} \delta_{g}(y_\ell) + \sum_{i \sim_k \ell} \delta_{y_\ell}(g)\right)\right\}} \tag{5}$$

This approximation is exact in the case where the y_i are independent, this assumption is crude, but allows to obtain decent results with a low computational effort and little prior knowledge on the dataset.

1.4.3 Path sampling

The second approach presented in the paper comes from the article [2], this time we consider the true conditional density of $\mathbf{y}$, $f(\mathbf{y} \mid \mathbf{X}, \beta, k)$ and we approximate $\log\left(Z(\beta, k)\right)$ using a Monte Carlo sum thanks to the following equality:

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

To avoid having to recalculate each time this integral, the authors advise to use the regularity of $Z(\beta, k)$ to calculate only for some value of β and then use an interpolation. This method has a relatively low computational cost, provided that the interpolation of $Z(\beta, k)$ has been computed beforehand, and it provides good results, without making the independence hypothesis. We note however that this interpolation is delicate, because it is based on a Monte Carlo sum approximation whose number of elements must be chosen sufficiently large if we want to avoid any instability caused by the interpolation of the noise.

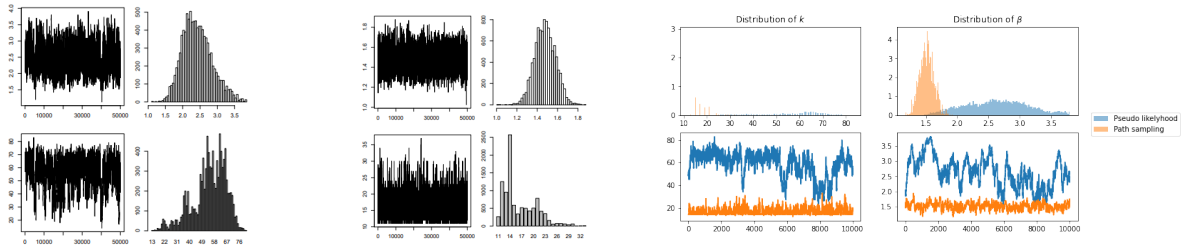

Figure 1: From left to right the distribution obtained by the authors for the pseudo-likelyhood approach, the path sampling approach and our result for both approach. We managed to get the same distribution.

1.4.4 Perfect sampling implementation and Gibbs approximation

This method gets around the difficulty of computing the normalization constant in a completely different way than the path sampling technique (2). To do so, the authors of the paper [3] use a trick which is to introduce a new variable $\mathbf{z}$ living in the same space as $\mathbf{y}$ whose conditional density $g(\mathbf{z} \mid \beta, k, \mathbf{y})$ is chosen so that the normalization constants cancel out in the calculation of the MCMC acceptance probability. The creators of this method recommend the following distribution:

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

where $\hat{\beta}, \hat{k}$ are calculated beforehand as the maximizers of the pseudo-likelyhood. This method, although ingenious, suffers from computational difficulty, as it is necessary to simulate $\mathbf{z}$ at each iteration of the MCMC algorithm according to its law $f(\mathbf{z} \mid \beta, k)$. Several approaches have been created to circumvent this problem and as the authors have chosen a binary classification task ($G = 2$) it is possible to simply simulate $\mathbf{z}$ using a Gibbs sampler.
The authors of the paper also noticed that for the task they wanted to perform, it was more appropriate to assign different values to $\hat{\beta}, \hat{k}$ however, they did not specify how they obtained these values. Unfortunately, we have not been able to replicate the results obtained for this technique, we suspect that the error comes from the way we simulate $\mathbf{z}$: indeed the authors have set up 3 strategies to circumvent the numerical instabilities, but they have not detailed enough the strategies for us to be able to reproduce their results. Furthermore, we have noticed that this approach is very sensitive to the choice of parameters $\hat{\beta}, \hat{k}$ and that one obtains catastrophic results if one has chosen bad values, moreover there does not exist to our knowledge an automatic way to determine good values for these parameters making the use of this method difficult on a new data set. Finally it is necessary to determine beforehand a sufficient burn in stage when simulating $\mathbf{z}$ with a Gibbs sampler.

2 Assessment of the three approaches

2.1 Evaluation of the pseudo-likelihood approximation

The approach using the pseudo likelyhood returns very different results from the two other methods, this indicates that the a posteriori law obtained is different from that of the other methods. This behavior is not desirable, but is directly due to the assumption of independence of the y_i which is wrong in general.

2.2 summary of the three methods

	Accuracy	Computation Time	Meta Parameters	Robustness
Pseudo Likelyhood	$\sim$	+	No	+
Path Sampling	+	$\sim$	Yes	$\sim$
Perfect Sampling	+	-	Yes	–

- The approach using pseudo-likelyhood gives fast and robust results but inaccurate.
- The path sampling approach is fast once $Z(\beta, k)$ is interpolated, this task is quite time consuming though.
- The path sampling approach is accurate in theory, but is numerically unstable and very slow because of the z Gibbs sampling.

After testing each method we recommend the pseudo likelyhood which is both simple to implement, fast and robust. Its results leave something to be desired but it is perfect for a novice user.
A more experienced user will be better off using Path sampling, provided that he is able to check that the meta parameters he has chosen are not the cause of instability in the method.

3 Theoretical complements

3.1 Gibbs prediction

We agree with the article that indeed, it is impossible to use the distribution of unclassified points to find the distribution of beta and k. But it is possible, once the distribution of k and beta is fixed, to simulate with a Gibbs sampler jointly Ytrain and Ytest by fixing Ytrain.

We created a dataset to showcase our idea, as shown in figure 2.

3.2 Errors in the article

We found the following errors in the article:

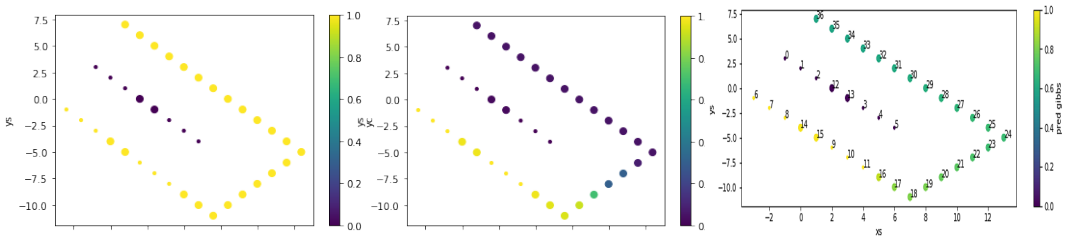

Figure 2: From left to right the ground-truth, sequential prediction and Gibbs prediction. The small and large dots are respectively in the train and test datasets. We can see that the sequential prediction fails to infer the yellow structure, while our Gibbs prediction succeeds in extending the yellow structure all the way to the top of the upper yellow arm.

- 3 typos in the rejection les mettre en gras
  formerly :

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

  correction :

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

- Ambiguity in the notations, the density of $\mathbf{z}$ conditional on β, k uses the notation $S(z)$ without clearly indicating the number of neighbors to consider. It would be clearer to indicate the dependence on $\hat{k}$.

3.3 Metaparameters and visualisations

Our code is available here. It contains animations, and many other visualizations allowing to better understand the Gibbs sampling and the choice of meta parameters.

References

[0] Lionel Cucala, Jean-Michel Marin, Christian Robert, Mike Titterington. A Bayesian reassessment of nearest-neighbour classification. Journal of the American Statistical Association, Taylor & Francis, 2009, 104 (485), pp.263-273. ff10.1198/jasa.2009.0125ff. ffinria-00143783v4f

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

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

[3] 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.

[4] L. Devroye, L. Györfi, and G. Lugosi. A Probabilistic Theory of Pattern Recognition. Springer, New York, 1996.