Robust risk estimation using exact resampling criteria for the kNN algorithm

Alain Célisse[1] and Tristan Mary-Huard[2,3]

[1]Laboratoire de Mathématique Paul Painlevé, UMR 8524 CNRS - Université Lille 1, France., Equipe-projet MODAL, INRIA Lille, France
[2]AgroParisTech/INRA, UMR 518 MIA, F-75005 Paris, France , UMR de Genetique Vegetale, INRA, Universite Paris-Sud, CNRS, Gif-sur-Yvette, France
[3]Corresponding author, maryhuar@agroparistech.fr

Abstract

In the binary classification framework, a closed form expression of the cross-validation Leave-p-Out (LpO) risk estimator for the k Nearest Neighbor algorithm (kNN) is derived. It is used to study the LpO risk minimization strategy for choosing k in the passive learning setting. The impact of p on the choice of k and the LpO estimation of the risk are inferred.
Keywords: Classification, Cross-validation, kNN.

1 Introduction

We consider the binary classification framework, where the goal is to predict the unknown label $Y \in \{0,1\}$ of an observation X. In the following, Z represents a random variable and z its realization. To this purpose, one aims at building from data $D = (X_1, Y_1), ..., (X_n, Y_n)$ a classifier $f : \mathcal{X} \to \{0,1\}$ whose classification error rate

$$L(f) = P(f(X) \neq Y|D)$$

is as low as possible, where $P(\cdot \mid D)$ denotes the probability with respect to (X, Y) given D. The risk of a classifier f is defined as $R(f) = \mathbb{E}_D\left[P(f(X) \neq Y \mid D)\right]$. The classification algorithm we consider here is the k Nearest Neighbor algorithm (kNN, [5, 6]), that has been successfully applied to many difficult classification tasks [7, 8]. The principle of the kNN classifier is simple: first, for a given observation x to classify, find $X_{(1)}, ...X_{(k)}$ the k closest points to x in the training set, then classify x according to a majority vote decision rule among these k neighbors. The performance of the kNN algorithm highly depends on the tuning of parameter k, that should be performed adaptively to the data at hand. To do so, resampling strategies such as Bootstrap or Leave-p-out (LpO) cross-validation can be used to estimate the prediction performance obtained with different values of k, and select the optimal value k^* that minimizes the prediction error rate. However, the computational cost of such strategies is prohibitive. In practice one often needs to limit the number of resamplings as the training sample size gets large, yielding poor approximation of the actual risk.

Recently, closed form expressions have been obtained for the LpO estimator when applied to the kNN algorithm [2]. This enables the practical use of LpO for kNN classifier at almost the same algorithmic cost as standard empirical risk minimization. The behavior of the minimizer k_p of the LpO estimator is investigated with respect to the sample size n and parameter p. In particular, it is shown that the choice of p is crucial for choosing k, unlike what happens for estimating the risk of a given kNN classifier.

2 Results

2.1 Closed form expression for LpO applied to kNN

Let $(x_1, y_1), ..., (x_n, y_n)$ denote the complete set of data. Each step of the LpO procedure splits this set into a training sample e of size $n-p$ and a validation sample $\bar{e}$ of size p. Let f^e denote the kNN classifier built from e and $\mathcal{E}$ the set of all possible training samples. Set $R_{LpO}(k)$ the estimation of the kNN performance based on LpO:

$$R_{LpO}(k) = \binom{n}{p}^{-1} \sum_{e \in \mathcal{E}} \left(\frac{1}{p} \sum_{i \notin e} \mathbb{I}_{\{f^e(x_i) \neq y_i\}} \right) \quad . \tag{1}$$

For a given point i in the validation set $\bar{e}$, let V_k^i denote the rank of its associated k^{th} neighbor in training set e. Let $(E, \bar{E})$ represent a random splitting of the complete set of data into 2 subsamples of size $n-p$ and p, respectively. Then,

$$\begin{aligned}
R_{LpO}(k) &= \binom{n}{p}^{-1} \sum_{e \in \mathcal{E}} \frac{1}{p} \sum_{i \in \bar{e}} \mathbb{I}_{\{f^e(x_i) \neq y_i\}} \\
&= \frac{1}{p} \sum_{i=1}^{n} \binom{n}{p}^{-1} \sum_{e \in \mathcal{E}} \mathbb{I}_{\{f^e(x_i) \neq y_i\}} \mathbb{I}_{\{i \in \bar{e}\}} \\
&= \frac{1}{p} \sum_{i=1}^{n} \sum_{e \in \mathcal{E}} \mathbb{I}_{\{f^e(x_i) \neq y_i\} \bigcap \{i \in \bar{e}\}} P(E = e) \\
&= \frac{1}{p} \sum_{i=1}^{n} P\left(\{f^E(x_i) \neq y_i\} \bigcap \{i \in \bar{E}\} \right) \\
&= \frac{1}{p} \sum_{i=1}^{n} P\left(f^E(x_i) \neq y_i | i \in \bar{E} \right) P\left(i \in \bar{E} \right) \\
&= \frac{1}{p} \sum_{i=1}^{n} \sum_{j=1}^{n} P\left(f^E(x_i) \neq y_i | i \in \bar{E},\ V_k^i = j \right) P\left(V_k^i = j | i \in \bar{E} \right) P\left(i \in \bar{E} \right) \\
&= \frac{1}{p} \sum_{i=1}^{n} P\left(i \in \bar{E} \right) \sum_{j=k}^{k+p-1} P\left(V_k^i = j | i \in \bar{E} \right) P\left(f^E(x_i) \neq y_i | i \in \bar{E},\ V_k^i = j \right) \quad .
\end{aligned}$$

Since E is uniformly distributed over $\mathcal{E}$, it comes

$$\forall i \in [1, n],\ P\left(i \in \bar{E} \right) = \frac{p}{n} \quad .$$

Similarly, one has

$$\begin{aligned}
\forall i \in [1, n],\ P\left(V_k^i = j | i \in \bar{E} \right) &= \frac{\binom{j-1}{j-k}\binom{n-j-1}{p-1-j+k}}{\binom{n-1}{p-1}} \\
&= \frac{k}{j} P\left(U = j - k \right) \quad ,
\end{aligned}$$

where $U \hookrightarrow \mathcal{H}(j, n-j-1, p-1)$ and $\mathcal{H}(a, b, c)$ denotes the hypergeometric distribution with a the number of white balls, k the number of black balls and c the number of balls to draw. Note that that none of these last two probabilities depend on i. To evaluate the last probability of the expression, let us consider the ordered sequence $X_{(1)}^i, ..., X_{(n-1)}^i$, where $X_{(k)}^i$ is the k^{th} neighbor of i in the complete sample. Since p observations (including i) are removed at a given step of the LpO procedure, the first k neighbors of i belong to $\{X_{(1)}^i, ..., X_{(k+p-1)}^i\}$. Once this list is obtained (by applying the $(k+p-1)$NN classifier to the complete data), one only needs to compute the number of times (over all splittings) the majority label is that of observation i, for each value of j. Since the computational cost to compute the last probability only involves the $k+p-1$ neighbors of i, it does not depend on n. As a consequence, the computation of R_{LpO} is linear in n. Let us now specify how to

compute this probability.

Let n_j^i be the number of 1s among the j neighbors of i in the complete sample. The quantity n_j^i can be obtained for all i and $j \in [1, k+p-1]$ by running the $(k+p-1)$NN classifier. We have:

$$P\left(f^E(x_i) \neq y_i | i \in \bar{E},\ V_k^i = j\right) = \mathbb{I}_{\{y_i=0\}} P\left(f^E(x_i) = 1 | i \in \bar{E},\ V_k^i = j\right) + \mathbb{I}_{\{y_i=1\}} P\left(f^E(x_i) = 0 | i \in \bar{E},\ V_k^i = j\right) \ .$$

Let N_i^E be the number of 1s among the k nearest neighbors of i in sub-sample E, and N_i^j the number of 1s among the j nearest neighbors of i in the complete training set. Assuming k is odd for sake of simplicity, one obtains:

$$\begin{aligned} P\left(f^E(x_i) = 1 | i \in \bar{E},\ V_k^i = j\right) &= P\left(N_i^E \geq k/2 | i \in \bar{E},\ V_k^i = j\right) \\ &= \mathbb{I}_{\{y_j=0\}} \left(1 - F_H\left(\frac{k+1}{2}\right)\right) + \mathbb{I}_{\{y_j=1\}} \left(1 - F_{H'}\left(\frac{k-1}{2}\right)\right) \ , \end{aligned} \tag{2}$$

where $H \hookrightarrow \mathcal{H}(N_i^j, j - N_i^j - 1, k-1)$, $H' \hookrightarrow \mathcal{H}(N_i^j - 1, j - N_i^j, k-1)$, and F_H stands for the cumulative distribution function of variable H. Similar formulas can be derived for $P\left(f^E(x_i) = 0 | i \in \bar{E},\ V_k^i = j\right)$.

2.2 Application to passive Learning

Using kNN classifiers in passive learning requires to choose k. This can be done using LpO. For every $1 \leq p \leq n$,

$$k_p = \arg \min_{1 \leq k \leq n} R_{LpO}(k) \ .$$

In the specific case $p = 1$, some theoretical results exist on the asymptotic behavior of k_1 with respect to n [4]. Having access to exact LpO enables to further infer the relationship between p and k_p, at least to a practical point of view. In the following, we investigate this relationship using 2-dimension simulated data. $X = (X^1, X^2)$ is generated using a mixture of 3 Gaussian distributions, with proportions $(0.2, 0.2, 0.6)$, means $(0.25, 0.25)$, $(0.5, 0.75)$, $(0.75, 0.75)$, and common covariance matrix I_2. The label Y is generated conditionally to X: if ($X^1 < 0.2$ and $X^2 < 0.2$) or ($X^1 > 0.8$ and $X^2 > 0.8$) then $P(Y = 1|X) = q$, otherwise $P(Y = 1|X) = 1 - q$. Several noise levels are considered: $q = 0, 0.1, 0.2, 0.3$, and 0.4. 100 repetitions of each condition have been performed.

Influence of n on k_p (p fixed) Calculations of Section 2.1 on the LpO estimator allow to study k_p with respect to n for various values of p. Figure 1 (left) displays k_p with respect to n for a level of noise $q = 0.2$, and gives a representative picture of the results. It shows that k_p is sub-linear with respect to n as long as p is kept independent of n. Since it is known that kNN estimators are consistent as long as $k = o(n)$ [4], it leads us to conjecture that the kNN classifier computed from k_p neighbors (with p fixed) is consistent.

Influence of p on k_p (n fixed) When several estimators are available, choosing the best one is a classical issue in statistics. Model selection is a typical strategy aiming at addressing this question. Choosing the number k of neighbors involved in the definition of the kNN estimator enters into this setting. First, considering Figure 1 (right) and Table 1, one observes that increasing p entails a smaller choice of k_p, which can be desirable as shown in the following. This phenomenon is observed with several noise levels from $q = 0.1$ up to $q = 0.4$ (not shown). Second, it is necessary to choose p larger than 1 as soon as the noise is not null. Indeed, LpO with small values of p leads to choose too large values of k when the noise is not null. This observation is supported by Figure 1 (right) and Table 1, where the minimum locations of red curves (small values of p) are larger than that of the black curves (which displays the true risk computed on a large validation set). This is also observed with a noise level $0.1 \leq q \leq 0.4$. A growing noise reduces the influence of the bias in the fitting of the kNN classifier, leading to a larger optimal k (compare black curves of Figure 1 between

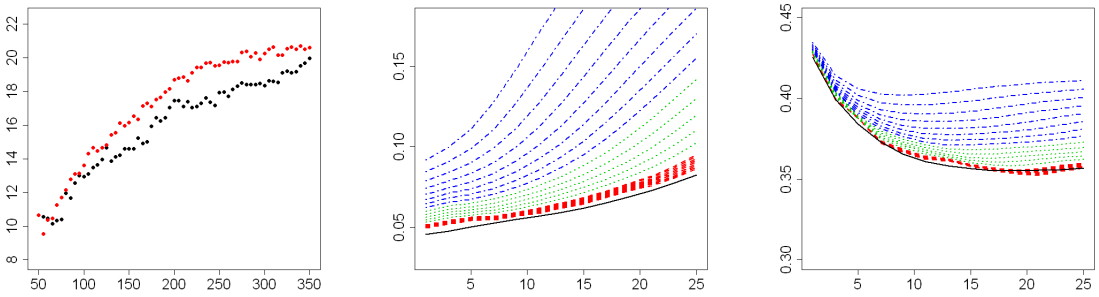

Figure 1: **Left:** Evolution of k (in ordinates) with respect to sample size n, $q = 0.2$. Black points correspond to k_p, red points correspond to the optimal choice of k (based on a large test sample). **Center:** Plot of the average classification error rate (in ordinates) evaluated by LpO with different p (colored curves) or on test samples (black curves), for different values of k (in abscisses) and for noise level $q = 0$. Red curves correspond to values of p lower than 20, green curves to values of p between 20 and 80, and blue curves to values higher than 80. **Right:** Same representation as previous, for noise level $q = 0.2$.

center and right panels). LpO with small values of p exhibits a higher sensitivity to this phenomenon than with larger values of p (Figure 1 right panel). Therefore, this trend can be balanced by using larger values of p (since higher p yield lower k_p). Indeed, we observe on Figure 1 that for some values of p larger than 1 (blue curves), the minimum location is close (or equal) to the best possible k. This suggests that (i) using L1O can be misleading, (ii) a convenient choice of $p > 1$ is required to provide a reliable k_p.

	$1 < p < 10$	$11 < p < 30$	$40 < p < 80$	$p > 80$	Test
k	21	19	17-15	13-9	17

Table 1: Choice of parameter k by LpO for different values of p, or by test sample, when $q = 0.3$.

Risk estimation In many applications, one is also interested in a sharp estimation of the performance of a given classifier. Due to the computational cost of LpO, this performance is often estimated with $p = 1$. One can wonder whether higher values of p should yield better results. First, Figure 1 shows that large values of p (blue curves) lead to biased estimations of the true risk (black curve). In other frameworks ([1, 3]), CV is known to be all the more biased as p is large. Second, these theoretical considerations entail that the least biased LpO estimator is obtained with $p = 1$. Figure 1 supports this conclusion since, for a fixed k, small values of p remain close to the black curve. Note that, depending on the noise level, larger values of p can also lead to reliable estimates of the true performance (not shown here). Third, an important conclusion arising from the case $q = 0$ (center of Figure 1) is that *model selection* and *risk estimation* can be *contradictory objectives*. All values of p lead to choose $k = 1$ from a model selection point of view. However, only $p = 1$ yields a (nearly) unbiased estimation of the risk.

3 Conclusions

In applications of kNN to real data, LpO is used either to assess the performance of a kNN classifier (risk estimation), or to choose k (model selection). In both cases, p is fixed at 1 in most cases for computational reasons. In the model selection setting, there is no guideline for practitioners about the relationship between p and k_p, or about the relevance of the selected value k_1. From a theoretical point of view, relating the optimal p to the signal-to-noise ratio and the size of the training set is of great interest. The closed-form expressions derived for the LpO estimator associated with kNN classifiers yield an efficient and practical tool to study the behavior of k_p, both for theoretical and practical purposes. Exact LpO should be preferred to its classical surrogate KCV, since LpO is less variable

(with $K = n/p$). The present simulation study is a preliminary work before the theoretical analysis of LpO in the passive learning setting. Some further work is required to get more insight toward a data-driven calibration of p.

References

[1] S. Arlot and A. Celisse. A survey of cross-validation procedures for model selection. *Statist. Surv.*, 4:40–79, 2010.

[2] A. Celisse and T. Mary-Huard. Exact cross-validation for knn and applications to passive and active learning in classification. *JSFdS*, 152(3), 2011.

[3] A. Celisse and S. Robin. Nonparametric density estimation by exact leave-p-out cross-validation. *Comput. Statist. Data Anal.*, 52(5):2350–2368, 2008.

[4] L. Devroye, L. Gyorfi, and G. Lugosi. *A probabilistic theory of pattern recognition.* Springer, 1996.

[5] E. Fix and J. Hodges. *Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques*, chapter Discriminatory analysis- nonparametric discrimination: Consistency principles. IEEE Computer Society Press, Los Alamitos, CA, 1991. Reprint of original work from 1952.

[6] E. Fix and J. Hodges. *Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques*, chapter Nonparametric Discrimination: small sample performance. IEEE Computer Society Press, Los Alamitos, CA, 1991. Reprint of original work from 1952.

[7] T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J.P. Mesirov, H. Coller, M.L. Loh, J.R. Downing, M.A. Caligiuri, C.D. Bloomfield, and E.S. Lander. Class prediction and discovery using gene expression data. *Science*, 286:531–537, 1999.

[8] P.Y. Simard, Y. LeCun, J.S. Denker, and B. Victorri. Transformation invariance in pattern recognition – tangent distance and tangent propagation. *International Journal of Imaging Systems and Technology*, 11(3), 2001.