Variable Selection and Weighting by Nearest Neighbor Ensembles

Jan Gertheiss

(joint work with Gerhard Tutz)

Department of Statistics
University of Munich

WNI 2008

Nearest Neighbor Methods

Introduction

- One of the simplest and most intuitive techniques for **statistical discrimination** (Fix & Hodges, 1951).
- **Nonparametric** and **memory based**.

Given data (y_i, x_i) with categorial response $y_i \in \{1, \ldots, G\}$ and (metric) p-dim. predictor x_i:

- Place a new observation with unknown class label into the class of the observation from the training set that is **closest to the new observation** - with respect to covariates.
- Closeness, resp. distance d of two observations is derived from a specific **metric in the predictor space**.
- Given the Euclidian metric $\Rightarrow d(x_i, x_r) = \sqrt{\sum_{j=1}^{p} |x_{ij} - x_{rj}|^2}$.

Nearest Neighbor Methods
Class probability estimates

- Use not only the first but the ***k* nearest neighbors**.
- The **relative frequency** of predictions in favor of category g among these neighbors can be seen as an **estimate of the probability** of category g.
- Estimates $\hat{\pi}_{ig}$ can take values h/k, $h \in \{0, ..., k\}$.
- If neighbors are weighted with respect to their distance to the observation of interest, $\hat{\pi}$ can in principle take all values in $[0, 1]$.

Nearest Neighbor Ensembles

Basic Concept

Final estimation by an **ensemble of single predictors**:

- Use the **ensemble formula** for computing the probability that observation i falls in category g:

$$\hat{\pi}_{ig} = \sum_{j=1}^{p} c_j \hat{\pi}_{ig(j)}, \text{ with } c_j \geq 0 \ \forall j \text{ and } \sum_j c_j = 1.$$

- With k nearest neighbor estimates $\hat{\pi}_{ig(j)}$ based on **predictor *j* only**.
- Weights - or coefficients - $c_1, \ldots, c_p$ need to be determined.

Nearest Neighbor Ensembles

More Flexibility?

- Why not

$$\hat{\pi}_{ig} = \sum_j c_{gj} \hat{\pi}_{ig(j)}, \text{ with } c_{gj} \geq 0 \; \forall g, j \text{ and } \sum_j c_{gj} = 1 \; \forall g \text{ ?}$$

- It can be shown: Restriction $\boldsymbol{c_{1j} = \ldots = c_{Gj} = c_j}$ is the only possibility to ensure that
  1. $\hat{\pi}_{ig} \geq 0 \; \forall g$ and
  2. $\sum_g \hat{\pi}_{ig} = 1$

  for all possible future estimations $\{\hat{\pi}_{ig(j)}\}$ with
  1. $\hat{\pi}_{ig(j)} \geq 0 \; \forall g, j$ and
  2. $\sum_g \hat{\pi}_{ig(j)} = 1 \; \forall j$.

Determination of Weights

Principles

- Given all $\{\hat{\pi}_{ig(j)}\}$, matrix $\hat{\Pi}$ with $(\hat{\Pi})_{ig} = \hat{\pi}_{ig}$ depends on $c = (c_1, \ldots, c_p)^T$.
- Given the training data with predictors $x_1, \ldots, x_n$ and true class labels $y = (y_1, \ldots, y_n)^T$, a previously chosen loss function - or score - $L(y, \hat{\Pi})$ is minimized over all possible c.

Note: The categorial response y_i is alternatively represented by a vector $z_i = (z_{i1}, \ldots, z_{iG})^T$ of dummy variables

$$z_{ig} = \begin{cases} 1, & \text{if } y_i = g \\ 0, & \text{otherwise} \end{cases}$$

Determination of Weights

Possible loss functions

Log Score

$$L(y, \hat{\Pi}) = \sum_i \sum_g z_{ig} \log(1/\hat{\pi}_{ig})$$

+ likelihood based
− Hypersensitive $\Rightarrow$ Inapplicable for nearest neighbor estimates.

Approximate Log Score

$$L(y, \hat{\Pi}) = \sum_i \sum_g z_{ig} \left((1 - \hat{\pi}_{ig}) + \frac{1}{2}(1 - \hat{\pi}_{ig})^2 \right)$$

+ Hypersensitivity removed
− Not "incentive compatible" (Selten, 1998), i.e. expected loss $E(L) = \sum_{y=1}^{G} \pi_y L(y, \hat{\pi}_y)$ not minimized by $\hat{\pi}_g = \pi_g$.

Quadratic Loss / Brier Score

$$L(y, \hat{\Pi}) = \sum_i \sum_g (z_{ig} - \hat{\pi}_{ig})^2$$

(introduced by Brier, 1950)

+ Not hypersensitive
+ Incentive compatible (see e.g. Selten, 1998)
+ Also takes into account how the estimated probabilities are distributed over the false classes.

Determination of Weights

Practical implementation

1. For each observation i create a matrix P_i of predictions:

$$(P_i)_{gj} = \hat{\pi}_{ig(j)}.$$

2. Create a vector $z = (z_1^T, \ldots, z_n^T)^T$ and a matrix $P = (P_1^T | \ldots | P_n^T)^T$.
3. Now the *Brier Score* as function of c can be written in matrix notation:

$$L(c) = (z - Pc)^T (z - Pc).$$

4. Given restrictions $c_j \geq 0 \ \forall j$ and $\sum_j c_j = 1$, weights c_j can be determined using **quadratic programming methods**; e.g. using the `R` add-on package `quadprog`.

Given the *approximate log score* the weights can be determined in a similar way.

Variable Selection

Variable Selection means setting $\boldsymbol{c_j = 0}$ for some j.

Thresholding:

- **Hard**: $c_j = 0$, if $c_j < t$; $c_j = c_j$, otherwise; e.g. $t = 0.25 \max_j \{c_j\}$.
- **Soft**: $c_j = (c_j - t)^+$.

(followed by rescaling)

Lasso based approximate solutions:

- If restrictions are replaced by $\sum_j |c_j| \leq s$, a **lasso type** problem (Tibshirani, 1996) arises.
- Lasso typical **selection characteristics** cause $c_j = 0$ for some j.

(followed by rescaling and $c_j = c_j^+$)

Including Interactions

Matrix P may be augmented by including **interactions of predictors**.

- Adding all predictions $\hat{\pi}_{ig(jl)}$, resp. $\hat{\pi}_{ig(jlm)}$ based on two or even three predictors.
- Feasible for *small scale problems* only; P has $p + \binom{p}{2} + \ldots$ columns.

Simulation Studies I

Two classification problems

There are 10 independent features x_j, each uniformly distributed on $[0, 1]$. The two class 0/1 coded response y is defined as follows (cf. Hastie et al., 2001):

- as an *"easy"* problem: $y = I(x_1 > 0.5)$, and
- as a *"difficult"* problem: $y = I(\text{sign}(\prod_{j=1}^{3}(x_j - 0.5)) > 0)$.

Simulation Studies I
Reference methods

Nearest neighbor methods:

- (3) Nearest neighbor based extended **forward / backward variable selection**.

  With tuning parameter S as the number of simple forward / backward selection steps that are checked in each iteration.

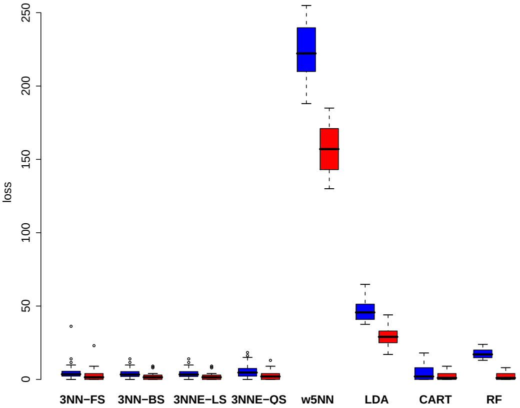
- **Weighted** (5) nearest neighbor prediction; `R` add-on package `kknn`.

Some alternative classification tools:

- Linear discriminant analysis (**LDA**); `R` add-on package `MASS`.
- **CART** (Breiman et al., 1984) and **Random Forests** (Breiman, 2001); `R` add-on packages `rpart`, `randomForest`.

Simulation Studies I
The easy problem

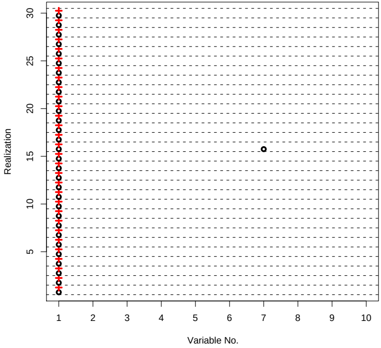
Prediction performance on the test set in terms of the Brier Score and No. of Missclassified Observations:

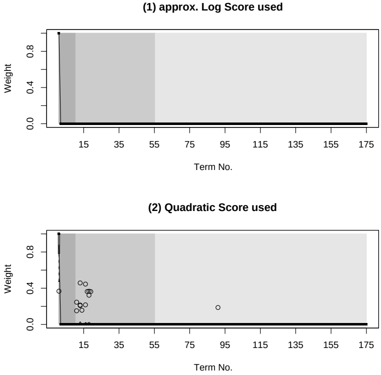


Simulation Studies I

The easy problem

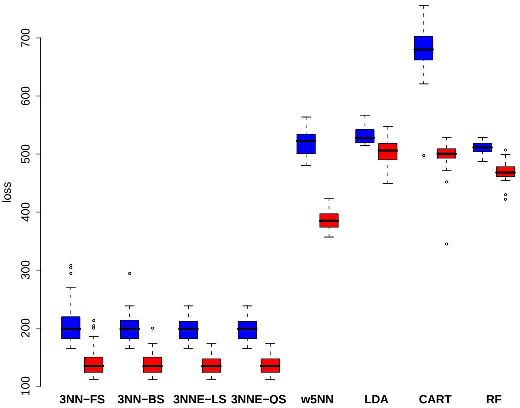
Variable selection/weighting by nearest neighbor based (extended) forward/backward selection (left) or nearest neighbor ensembles (right):





Simulation Studies I
The difficult problem

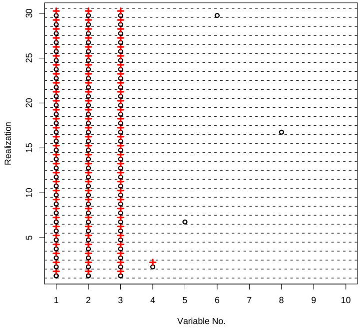
Prediction performance on the test set in terms of the Brier Score and No. of Missclassified Observations:

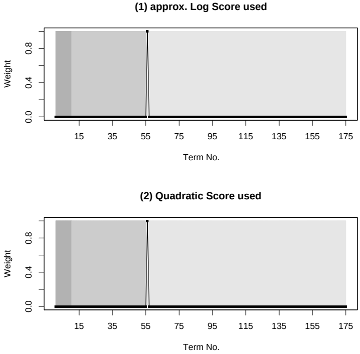


Simulation Studies I

The difficult problem

Variable selection/weighting by nearest neighbor based (extended) forward/backward selection (left) or nearest neighbor ensembles (right):



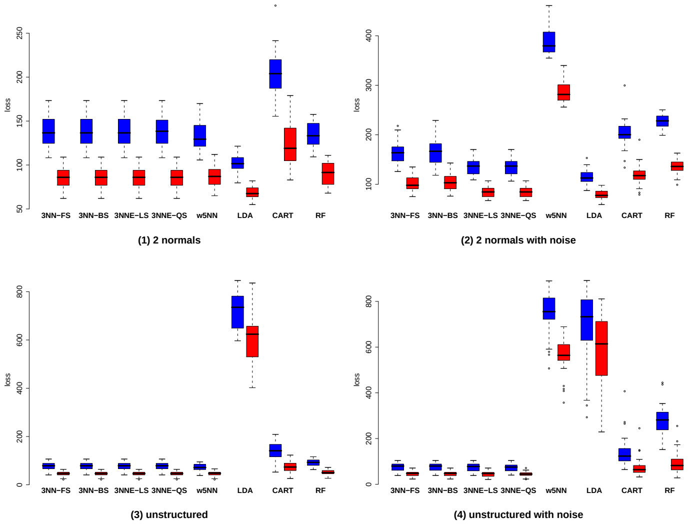


Simulation Studies II

cf. Hastie & Tibshirani (1996)

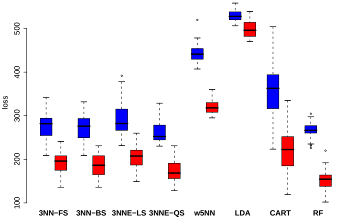
1. **2 Dimensional Gaussian**: Two Gaussian classes in two dimensions.
2. **2 Dimensional Gaussian with 14 Noise**: Additionally 14 independent standard normal noise variables.
3. **Unstructured**: 4 classes, each with 3 spherical bivariate normal subclasses; means are chosen at random.
4. **Unstructured with 8 Noise**: Augmented with 8 independent standard normal predictors.
5. **4 Dimensional Spheres with 6 Noise**: First 4 predictors in class 1 independent standard normal, conditioned on radius > 3; class 2 without restrictions.
6. **10 Dimensional Spheres**: All 10 predictors in class 1 conditioned on $22.4 < \text{radius}^2 < 40$.
7. **Constant Class Probabilities**: Class probabilities (0.1,0.2,0.2,0.5) are independent of the predictors.
8. **Friedman's example**: Predictors in class 1 independent standard normal, in class 2 independent normal with mean and variance proportional to $\sqrt{j}$ and $1/\sqrt{j}$ respectively, $j = 1, \ldots, 10$.

Simulation Studies II
Scenario 1 - 4

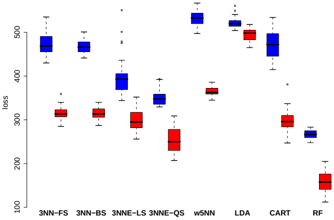


Simulation Studies II

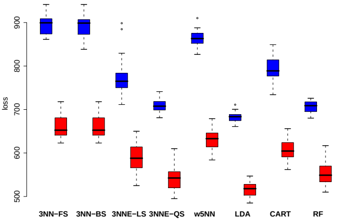
Scenario 5 - 8



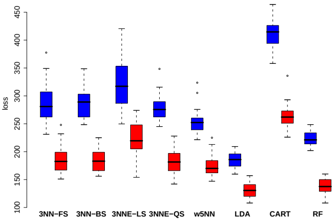
(5) 4D sphere in 10 dimensions



(6) 10D sphere in 10 dimensions



(7) constant class probabilities



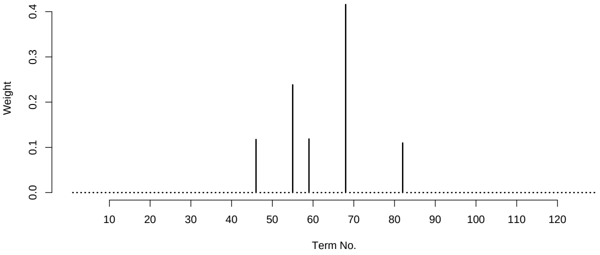
(8) Friedman's example

Real World Data

Glass data, R package `mlbench`

Forecast the type of glass (6 classes) on the basis of the chemical analysis given in form of 9 metric predictors.

- Result 3NNE-QS / all data

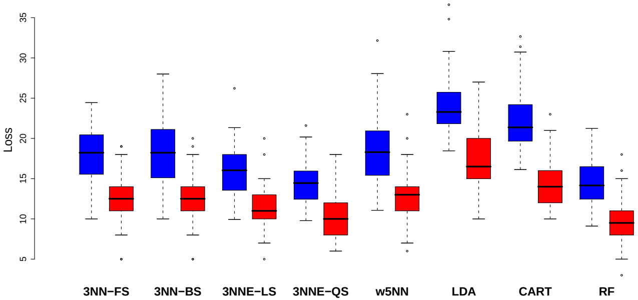


Real World Data

Glass data, `R` package `mlbench`

Forecast the type of glass (6 classes) on the basis of the chemical analysis given in form of 9 metric predictors.

- Performance / 50 random splits



Summary
Nearest neighbor ensembles

- Nonparametric probability estimation by an ensemble, i.e. **weighted average of nearest neighbor estimates**.
- Each estimate is based on a **single** or a **very small subset** of predictors.
- **No black box** (by contrast to many other ensemble methods).
- **Good performance** for **small scale problems**, particularly if pure noise variables can be separated from relevant covariates.
- Direct application to high dimensional problems with interactions is not recommended.
- Given microarrays possibly useful as **nonparametric gene preselection tool**.
- May be employed for **automatic choice** of the most **appropriate metrics** or the **right neighborhood**.
- Application to regression problems is possible as well.

References

BREIMAN, L. (2001): Random Forests, *Machine Learning 45, 5–32*

BREIMAN, L. et al. (1984): *Classification and Regression Trees*, Monterey, CA: Wadsworth.

BRIER, G. W. (1950): Verification of forcasts expressed in terms of probability, *Monthly Weather Review 78, 1–3*.

FIX, E. and J. L. HODGES (1951): Discriminatory analysis - nonparametric discrimination: consistency properties, US air force school of aviation medicine, Randolph Field Texas.

GERTHEISS, J. and TUTZ, G. (2008): Feature Selection and Weighting by Nearest Neighbor Ensembles, *University of Munich, Department of Statistics: Technical Report, No.33*.

HASTIE, T. and R. TIBSHIRANI (1996): Discriminant adaptive nearest neighbor classification, *IEEE Transactions on Pattern Analysis and Machine Intelligence 18, 607–616*.

HASTIE, T. et al. (2001): *The Elements of Statistical Learning*, New York: Springer.

R Development Core Team (2007): *R: A Language and Environment for Statistical Computing*, Vienna: R Foundation for Statistical Computing.

SELTEN, R. (1998): Axiomatic characterization of the quadratic scoring rule, *Experimental Economics 1, 43–62*.

TIBSHIRANI, R. (1996): Regression Shrinkage and Selection via the Lasso, *Journal of the Royal Statistical Society B, 58, 267–288*.