BIOINFORMATICS APPLICATIONS NOTE
Vol. 22 no. 22 2006, pages 2828–2829
doi:10.1093/bioinformatics/btl462

Genetics and population analysis

Model-based boosting in high dimensions

Torsten Hothorn[1,*] and Peter Bühlmann[2]

[1]Institut für Medizininformatik, Biometrie und Epidemiologie, Friedrich-Alexander-Universität Erlangen-Nürnberg, Waldstraße 6, D-91054 Erlangen, Germany and [2]Seminar für Statistik, ETH Zürich, CH-8092 Zürich, Switzerland

Received on July 9, 2006; revised on August 22, 2006; accepted on August 24, 2006
Advance Access publication August 29, 2006
Associate Editor: Keith A Crandall

ABSTRACT

Summary: The R add-on package mboost implements functional gradient descent algorithms (boosting) for optimizing general loss functions utilizing componentwise least squares, either of parametric linear form or smoothing splines, or regression trees as base learners for fitting generalized linear, additive and interaction models to potentially high-dimensional data.

Availability: Package mboost is available from the Comprehensive R Archive Network (http://CRAN.R-project.org) under the terms of the General Public Licence (GPL).

Contact: Torsten.Hothorn@R-project.org

Linking the clinical phenotype of a patient to his or her genotype, i.e. to measurements of potentially thousands of expression values, transcripts or proteins, is a challenging task for multivariate regression analysis. Roughly, two approaches can be distinguished. Either a univariate pre-selection of variables is performed before fitting a more classical regression or classification model or modern prediction methods, such as random forest, support vector machines and boosting, are used. The latter are more coherent, simultaneous approaches and seem to be more promising.

Recent developments in multivariate regression analysis utilize boosting methods as an optimization technique for the estimation of potentially high-dimensional linear or additive models (see Bühlmann and Yu, 2003; Bühlmann, 2006). The mboost add-on package to the R system for statistical computing (R Development Core Team, 2006 http://www.R-project.org) implements generalized linear and generalized additive models utilizing flexible boosting algorithms for (constrained) minimization of the corresponding empirical risk function. One single hyper parameter, the number of iterations in the boosting algorithm, must be chosen by the data analyst. Both classical and corrected Akaike information criteria (AIC) and cross-validation techniques (*k*-fold, bootstrap, etc.) are implemented for the selection of this hyper parameter.

As an illustration, we study a binary classification problem involving $p = 7129$ gene expression levels in $n = 49$ breast cancer tumor samples (date taken from West *et al*., 2001). For each sample, a binary response variable describing the lymph node status of the patient is to be explained by the gene expression profiles.

The data are stored in the form of an *exprSet* object (see Gentleman *et al*., 2004) and we first extract the matrix of expression levels and the response variable, and center the expression levels for each gene around zero:

```
> x <- exprs(westbc)
> x <- t(x - rowMeans(x))
> y <- pData(westbc)$nodal.y
```

We aim at fitting (including variable selection) a high-dimensional logistic linear regression model with $p = 7129$ covariates; such models are very competitive for tumor classification based on gene expression data (e.g. see Bühlmann, 2006). The function glmboost() fits the binary response variable y to the matrix of expression values x optimizing the negative binomial log-likelihood, which is specified via the family = Binomial() argument (similar to the classical glm() function). Initially, we use 500 boosting iterations in the algorithm:

```
> westglm <- glmboost(x, y, family = Binomial(),
      control = boost_control(mstop = 500))
```

The goodness of the model fit for varying numbers of boosting iterations can be studied using the AIC criterion (Fig. 1). Here, the AIC criterion suggests to stop after 264 boosting iterations; therefore, we use the final boosting fit by

```
> westglm <- westglm[mstop(westAIC)]
> plot(westAIC <- AIC(westglm, ''classical''))
```

Now, as for a classical linear model fit, we can extract the coefficients of the linear predictor via

```
> coef(westglm)
```

Only 30 out of 7129 coefficients are non-zero, i.e. 30 out of 7129 covariates are selected. Since boosting is fairly insensitive to the inclusion of additional noise covariates [i.e. covariates having no

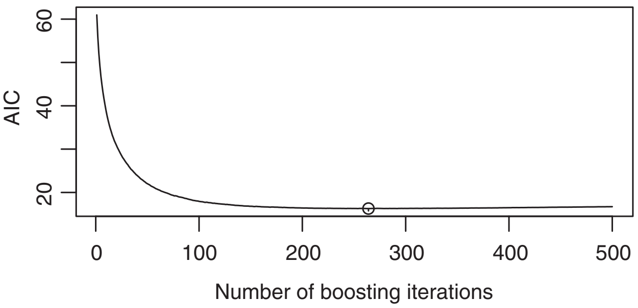

Fig. 1. Akaike information criterion based on the negative binomial log-likelihood for early stopping of the boosting algorithm.

[*]To whom correspondence should be addressed.

effect on the response variable, see Bühlmann (2006)], we do not advise to apply gene pre-selection using, for example, the two-sample *t* or Wilcoxon test here. Both approaches, owing to their univariate nature, could easily lead to many false selections and non-selections while boosting (without employing pre-selection) does multivariate selection of genes.

For the patients in the learning sample, the true and predicted lymph node status can be compared via

```
>table(y,predict(westglm,type=''response''))
y         negative positive
  negative     25        0
  positive      0       24
```

This approach to fit a generalized linear model to high-dimensional data, with implicit variable selection, is computationally attractive. Fitting the model as shown above takes $\sim 3\,\mathrm{s}$ on a simple desktop computer. The AIC criterion can be computed in roughly the same time.

The survival time of patients is an important endpoint, especially in clinical trials in oncology, and thus recent research focuses on regression models for high-dimensional survival problems (e.g. Hothorn *et al*., 2006). Sültmann *et al*. (2005) report results of a trial linking genomic measurements to the survival time of patients suffering from kidney cancer. Using their data, after centering the expression values and setting up an object describing the censored response variable via

```
> x <- exprs(eset)
> x <- t(x - rowMeans(x))
> y <- Surv(eset$survival.time, eset$died)
```

we can utilize the `glmboost()` function to fit a Cox proportional hazards model

```
> kidpackCox <- glmboost(x, y, family = CoxPH())
```

with initial 100 boosting iterations. Only

```
> sum(abs(coef(kidpackCox)) > 0)
[1] 14
```

gene expressions covariates have corresponding non-zero regression coefficients (using here the ad hoc number of 100 boosting iterations) and the linear predictor

```
> predict(kidpackCox, type = ''lp'')
```

could be used as a compound covariate, e.g. for a survival tree, in order to identify clinical risk groups.

Going beyond generalized linear models is possible using the `gamboost()` function for fitting generalized additive models. The general and flexible design of the `R` package `mboost`, implementing a well-defined framework rather than special purpose functionality, empowers the data analysts to set up their own families (and thus boosting algorithms) rather quickly. For example, the call

```
> gamboost(x, y, family = CoxPH())
```

essentially implements boosting of additive proportional hazards models as studied by Li and Luan (2005). Moreover, an implementation of tree-based boosting is available in function `blackboost()`. Cross-validation estimates of the empirical risk, based on techniques, such as the bootstrap, stratified bootstrap and *k*-fold cross-validation, can be computed by means of the `cvrisk()` function.

The analyses presented here are reproducible by running the commands in

```
> system.file(''mboost_Bioinf.R'', package =
    ''mboost'')
```

ACKNOWLEDGEMENT

The work of T. Hothorn was supported by Deutsche Forschungsgemeinschaft (DFG) under grant HO3242/1-1.

Conflict of Interest: none declared.

REFERENCES

Bühlmann,P. (2006) Boosting for high-dimensional linear models. *Ann. Stat.*, **34**, 559–583.

Bühlmann,P. and Yu,B. (2003) Boosting with the L_2 loss: regression and classification. *J. Am. Stat. Assoc.*, **98**, 324–339.

Gentleman,R.C. *et al*. (2004) Bioconductor: open software development for computational biology and bioinformatics. *Genome Biol.*, **5**, R80.

Hothorn,T. *et al*. (2006) Survival ensembles. *Biostatistics*, **7**, 355–373.

Li,H. and Luan,Y. (2006) Boosting proportional hazards models using smoothing splines, with applications to high-dimensional microarray data. *Bioinformatics*, **21**, 2403–2409.

R Development Core Team (2006) *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing Vienna, Austria. ISBN 3-900051-07-0.

Sültmann,H. *et al*. (2005) Gene expression in kidney cancer is associated with novel tumor subtypes, cytogenetic abnormalities and metastasis formation. *Clin. Cancer Res.*, **11**, 646–655.

West,M. (2001) Predicting the clinical status of human breast cancer by using gene expression profiles. *Proc. Natl Acad. Sci. USA*, **98**, 11462–11467.