STAT 770 Dec. 14 Lecture 29

Gradient Boosted Logistic Regression versus Decision Trees

Reading and Topics for this lecture: **gbm** software descriptions (posted to Decision Tree module in ELMS) and CRAN package descriptions, plus the R Script for this class: `Boosting.RLog`

(1) High-level discussion of gradient boosting.

(2) `R` Script case-studies, of `gbm`

Idea of Gradient Boosting (Friedman)

Read about this in online tutorials or blogs, e.g. the `gbm` Vignette by Ridgway or https://bradleyboehmke.github.io/HOML/gbm.html. For conceptual introduction, see Hastie, Tibshirani & Friedman, **Elements of Statistical Learning** book (2nd ed. free online).

Objective: to minimize loss functional $J(f) = \sum_{i=1}^{N} L(y_i, f(x_i))$ by choice of function f that is at least piecewise smooth.

Ingredients: gradient descent seeks to improve $J(\widehat{f})$ by moving $\widehat{f}$ at x_i in the direction $z_i = -\frac{\partial}{\partial t} L(y_i, t) \Big|_{t=\widehat{f}(x_i)}$.

When $L(y,t) = \frac{1}{2}(y-t)^2$, the z_i are residuals.

Idea of Gradient Boosting, continued

Sequential improvement is achieved at each of many stages by fitting a model (M): $z_i \sim g(x_i)$, moving $\widehat{f} \mapsto \widehat{f} + \rho\, g$, where step-size ρ is chosen to make $J(\widehat{f} + \rho g) < J(\widehat{f})$ small.

An earlier approach (Adaboost) was to move $\widehat{f}$ by incrementing it with a model for the residuals.

Model (M) for z_i can be based on weak learners such as decision-trees. Base learner model like logistic regression is built into L through squared estimating equation or negative log-likelihood.

The weak learner idea is that the individual models g at each stage should just *move in the right direction* and need not be very precise, e.g., may be a very shallow tree; a lack of fine discrimination actually helps maintain smoothness and stability.

Inside Gradient-boosting Software

gbm has options for shrinkage (multiplicative factor to make ρ smaller, when boosting will be done in many small stages), for cross-validation (to estimate error-rates at each stage), and for interaction depth (say 1 or 2, tree-depth for models g).

gbm also allows subsampling of data at each stage, but ‘extreme’ boosting implements in package xgboost includes the random forest idea of random data and split-variable subsampling at each stage.

Summary of **gbm** output tells which iteration-stage produces the best model and prediction: *this generally occurs at the last iteration*.

High-level ML Philosophy

How do these Machine Learning (**ML**) approaches differ from statistical approaches optimizing likelihoods or estimating equations ? The ML methods are not completely algorithmic but share a theoretical idea: they do not search only over models at the level of individual observations, but combine (often with *voting methods*) information about co-occurring clusters of observations and predictors. Resampling the universe of these (via bootstrap and variants) lies at the heart of ML successes.

Let us look at further comparisons of methods in `Boosting.RLog`

The success of these methods is hard to account for theoretically, so far. Useful references are given on the next slide.

Additional References

Breiman, L. (2001), Random Forests, Mach. Learning 45, 5-32.

gbmVignette pdf file on ‘Generalized Boosted Models’ (2019), by G. Ridgeway

J.Friedman, T.Hastie, R.Tibshirani (2000), Additive Logistic Regression: a Statistical View of Boosting, Ann. Stat. 28, 337-74.

J.H. Friedman (2001), Greedy Function Approximation:
A Gradient Boosting Machine, Ann. Stat. 29: 1189-1232.

J.H. Friedman (2002). Stochastic Gradient Boosting, Computational Statistics and Data Analysis 38(4):367-378.

http://statweb.stanford.edu/ jhf/R-MARTwebsite.