Revolutions

Learn more about using open source R for big data analysis, predictive modeling, data science and more from the staff of Revolution Analytics.

May 09, 2013

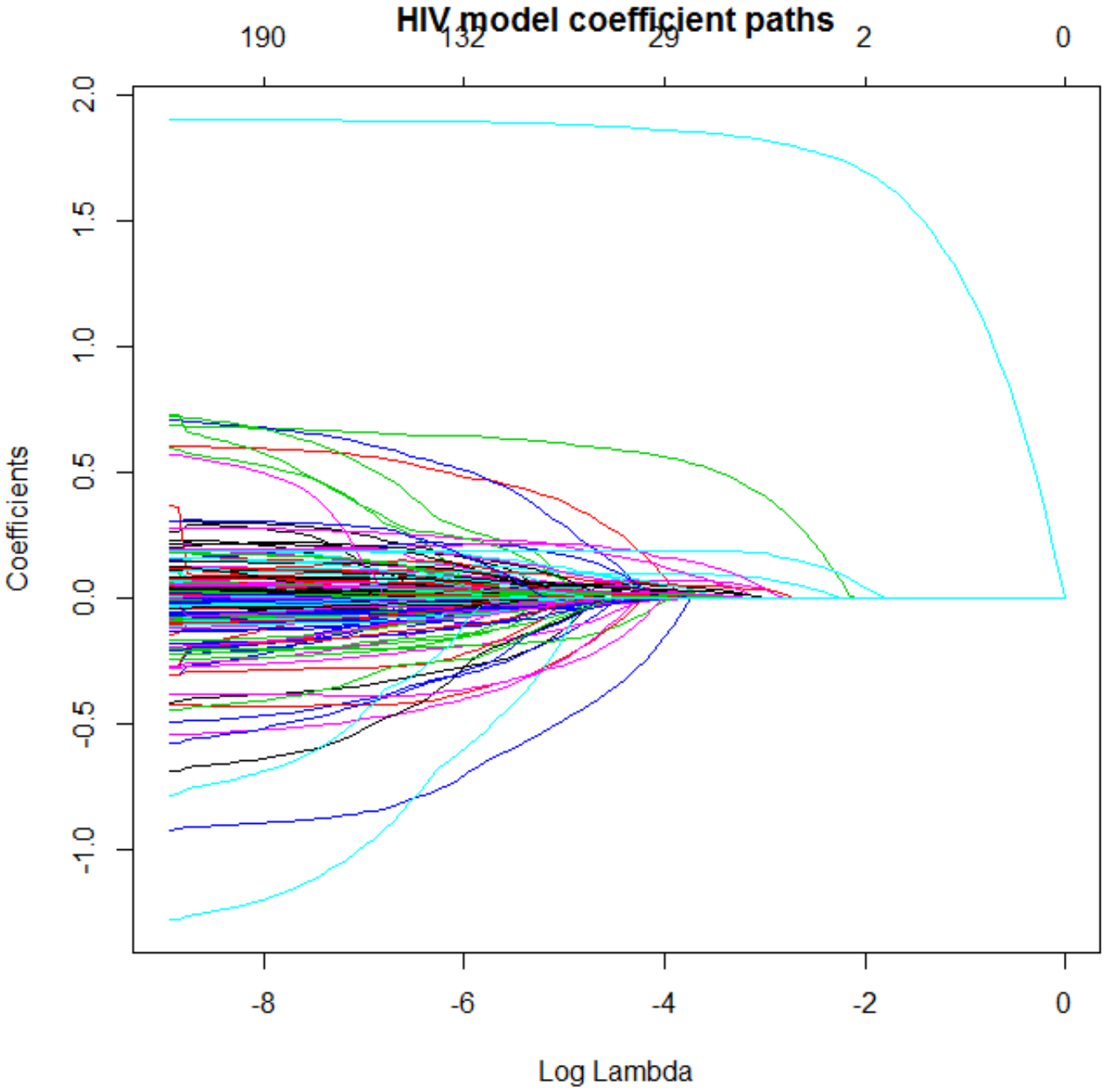
Trevor Hastie presents glmnet: lasso and elastic-net regularization in R

by Joseph Rickert

Even a casual glance at the R Community Calendar shows an impressive amount of R user group activity throughout the world: 45 events in April and 31 scheduled so far for May. New groups formed last month in Knoxville, Tennessee (The Knoxville R User Group: KRUG) and Sheffield in the UK (The Sheffield R Users). An this activity seems to be cumulative. This month, the Bay Area R User's Group (BARUG) expects to hold its 52nd and 53rd meet ups while the Sydney Users of R Forum (SURF) will hold its 50th. Everywhere R user groups are sponsoring high quality presentations and making them available online, but the Orange County R User Group is pushing the envelope with respect to sophistication and reach. Last Friday, I attended a webinar organized by this group where Professor Trevor Hastie of Stanford University presented Sparse Linear Models with demonstrations using GLMNET. This was a world-class presentation and quite a coup for Orange County to have Professor Hastie present.

The glmnet package written Jerome Friedman, Trevor Hastie and Rob Tibshirani contains very efficient procedures for fitting lasso or elastic-net regularization paths for generalized linear models. So far the glmnet function can fit gaussian and multiresponse gaussian models, logistic regression, poisson regression, multinomial and grouped multinomial models and the Cox model. The efficiency of the glmnet algorithm comes from using cyclical coordinate descent in the optimization process and from Jerome Friedman's underlying Fortran code.

Although Professor Hastie's presentation was primarily concerned with fitting models for the wide problem (the number of explanatory variables is much larger than the number of observations) the lasso and elastic-net algorithms are just as applicable to data sets with large numbers of observations. It is likely that in the future we will see glmnet implementations for variable selection on datasets with thousands of variables and hundreds of millions of observations. The following graph shows the regularization paths for the coefficients of a model fit the HIV data from one Professor Hastie's examples.



Each curve represents a coefficient in the model. The x axis is a function of lambda, the regularization penalty parameter. The y axis gives the value of the coefficient. The graph shows how the coefficients "enter the model" (become non-zero) as lambda changes. The following code, based on an example from the webinar, produces the plot and also shows how easy it is to perform cross-validation.

```
library(glmnet)     # load the package
load("hiv.rda")     # HIV data
class(hiv.train)    # The data are stored as a list
names(hiv.train)    # The names of the list elements are x and y
dim(hiv.train$x)    # The explanatory data consists of 704 observations of
                    # 208 binary mutation variables
head(hiv.train[[1]])    # Look at the explanatory data
head(hiv.train[[2]])    # Look at the response data: changes in susceptibility to
fit=glmnet(hiv.train$x,hiv.train$y)     # fit the model
plot(fit,xvar="lambda", main="HIV model coefficient paths") # Plot the paths for
fit                 # look at the fit for each coefficient
```

```
#
cv.fit=cv.glmnet(hiv.train$x,hiv.train$y)       # Perform cross validation on the
plot(cv.fit)    # Plot the mean sq error for the cross validated fit as a functio
                # of lambda the shrinkage parameter
                # First vertical line indicates minimal mse
                # Second vertical line is one sd from mse: indicates a smaller mo
                # is "almost as good" as the minimal mse model
tpred=predict(fit,hiv.test$x)   # Predictions on the test data
mte=apply((tpred-hiv.test$y)^2,2,mean)  # Compute mse for the predictions
points(log(fit$lambda),mte,col="blue",pch="*")  # overlay the mse predictions on
legend("topleft",legend=c("10 fold CV","Test"),pch="*",col=c("red","blue"))
```

Created by Pretty R at inside-R.org

Don't be content with this partial example. Professor Hastie and The Orange County R User Group have graciously made the slides available at this link; the code and data are available here. The webinar is well worth watching in its entirety.

As you might expect, Professor Hastie gives a masterful presentation: lucid, clear and succinct. This is inspite of the fact that Professor Hastie begins the presentation by commenting that it was his first webinar ever and that he was a little uncomfortable talking to his screen. (I think anyone who has ever given a webinar can relate to this: you talk to the screen and no energy from the audience comes back. Nothing is more disruptive to efforts to be enthusiastic than silence.) Nevertheless, Professor Hastie presents a difficult topic with a clarity that carries his audience along, and he is completely unphased by the inevitable glitch. Watch how he handles the upside down slide. You can download his slides, R scripts and data from the link below.

Trevor Hastie: Sparse Linear Models, with demonstrations using GLMNET

[**Updated** May 7 2014 with corrected link to code and data, with thanks to reader RL]

Posted by Joseph Rickert at 09:01 in packages, predictive analytics, R, statistics | Permalink

Comments

You can follow this conversation by subscribing to the comment feed for this post.

How do you obtain the beta coefficients? He only checked the model using cv. I am looking for the coefficients after LASSO shrinkage.

Posted by: Mike | May 23, 2013 at 13:26

Is there a way to fix the lag between audio and video? The sound is delayed by 5-10 sec so that the slide advances before he's finished discussing it.

Posted by: Tim | August 27, 2013 at 17:53

The comments to this entry are closed.