Comments on prediction competition

Nov 28, 2019

Stanislav Volgushev

The data set

Data set was combination of a real data set on pollution data in Beijing and artificial predictors. Real part from original data set:

Name in competition data	original name
y0	pm2.5
X1	year
X2	month
X3	day
X4	hour
X5	Dewpoint
X6	Temperature
X7	Precipitation
X8	wind direction
X9	Iws (cumulative wind speed)

Original data set:

`https://archive.ics.uci.edu/ml/datasets/Beijing+PM2.5+Data`

Comments on building a good model

Some useful steps to building a good predictive model (this is not a universal recipe and should not be applied blindly)!

1. Remove irrelevant predictors. We have discussed several ways to do this: stepwise and best subset methods, lasso, variable importance measures in bagging and random forests.

Note: direct application of lasso and stepwise methods with linear model only measures *linear* impact of predictors. This is not always appropriate (perhaps ok for this data set).

2. Look at relationship among predictors.

3. Find the right non-linear transformations of some predictors and build a gam.

As a first step, looking at plots of response vs each predictor individually can help to determine which transformations to try. Look at plots of residuals from your model against individual predictors to see if you need fine-tuning. Always compare models by cross-validation.

Data generation I

Generate additional predictors:

```
X = runif(100*nrow(res))
Xa = array(X,c(nrow(res),100))
Xa[,21:80] =  Xa[,21:80] + runif(nrow(res)) # makes predictors dependent

Xa[,c(5,11:15,80:90)] = round(5*Xa[,c(5,11:15,80:90)]) # make discrete
Xa[,c(3,16:20)] = round(25*Xa[,c(3,16:20)]) # make discrete
Xa[,92] = round(50*Xa[,92]) # make discrete
Xa = scale(Xa)

for(j in 4:10){
     Xa[,90+j] = sample(dat[,j]) # sample randomly from original data
}
```

Data generation II

New response generated according to

```
y = 2*y0
    + 5*dnorm(1/Xa[,93]) # non-linear influence
    + 2*sin(exp(1/Xa[,92])) # non-linear influence
    + 5*dnorm(-2+1/Xa[,100]) # non-linear influence
    + rowMeans(Xa[,21:80]) # slight linear influence
    + 5*rowMeans(Xa[,1:5]) # strong linear influence
    + 5*sin(5*rowMeans(Xa[,6:10])) # non-linear influence
```

where $y0$ is the response in the original data set.

- Predictors in competition data X101, X102, X109 correspond to 'special' influence.
- Predictors 20-29 and 90-100, 103-108 are noise predictors (completely irrelevant).

Things you can try: run ridge regression in a first step to capture all the linear influence and run boosting/random forest on residuals using only predictors with non-linear influence.

A simple solution to homework problems Q5-Q7

A simple solution to homework problems Q5-Q7

```
library(gbm)

d.train = read.csv('trainingdata2.csv')
d.test = read.csv('test_predictors2.csv')

boo = gbm(y ~ . ,
 data=d.train,
 distribution='gaussian',
 n.trees = 5000,
 interaction.depth = 1,
 shrinkage = 0.01,
 cv.folds = 5
 )

bi = gbm.perf(boo,method="cv")
yhat.boo = predict(boo,newdata=d.test,n.trees=bi)
```

gives test rMSE of 4.054 on public and 4.046 on private leaderboard