New models in chemmodlab

Jeremy Ash

March 9, 2017

- nnet for regression
  - Same package with linear output switched on (as opposed to the logistic output function default)
  - If you are using nnet for a regression (rather than a classification) problem you need to set linout=T to tell nnet to use a linear output (rather than a sigmoid output which is then thresholded for classification).
  - same tuning parameters
- knn for regression
  - using knnregTrain in caret, modified class::knn so that returns average of knn response for predictions
    - did this in case want to do tuning in caret, better to implement the same model in chemmodlab?
  - "FNN" (fast nearest neighbors) is faster. kd-tree algorithm seems preferrable
    - However, cannot tune in caret
    - http://www.cs.umd.edu/~mount/ANN/
    - In practice: – kd-trees work "well" in "low-medium" dimensions
    - Most commonly used nearest neighbor search algorithm
    - The default algorithm is the KD-Tree, which generally has both good computation time and accuracy. Linear search is guaranteed to find the true nearest neighbors, but has a very high computation cost O(n).
    - O(log(n)) complexity
  - Not sure how cover tree is different
    - The cover tree is O(n) space data structure which allows us to answer queries in the same O(log(n)) time as kd tree given a fixed intrinsic dimensionality
  - Review of different algorithms: https://arxiv.org/pdf/1509.06957.pdf
- Lasso and Lasso Glm
  - used Lars algorithm for lasso and glmnet for lasso glm
  - A binomial link can be used with glmnet, so this model can be used with a binary response
  - "LARS is faster for small problems, very sparse problems, or very 'wide' problems (much much more features than samples). Its computational cost is limited by the number of features selected, if you don't compute the full regularization path. On the other hand, for big problems, glmnet (coordinate descent optimization) is faster." - developer for scikit
  - default .1 value for lambda parameter as in chemmodlab ridge regression

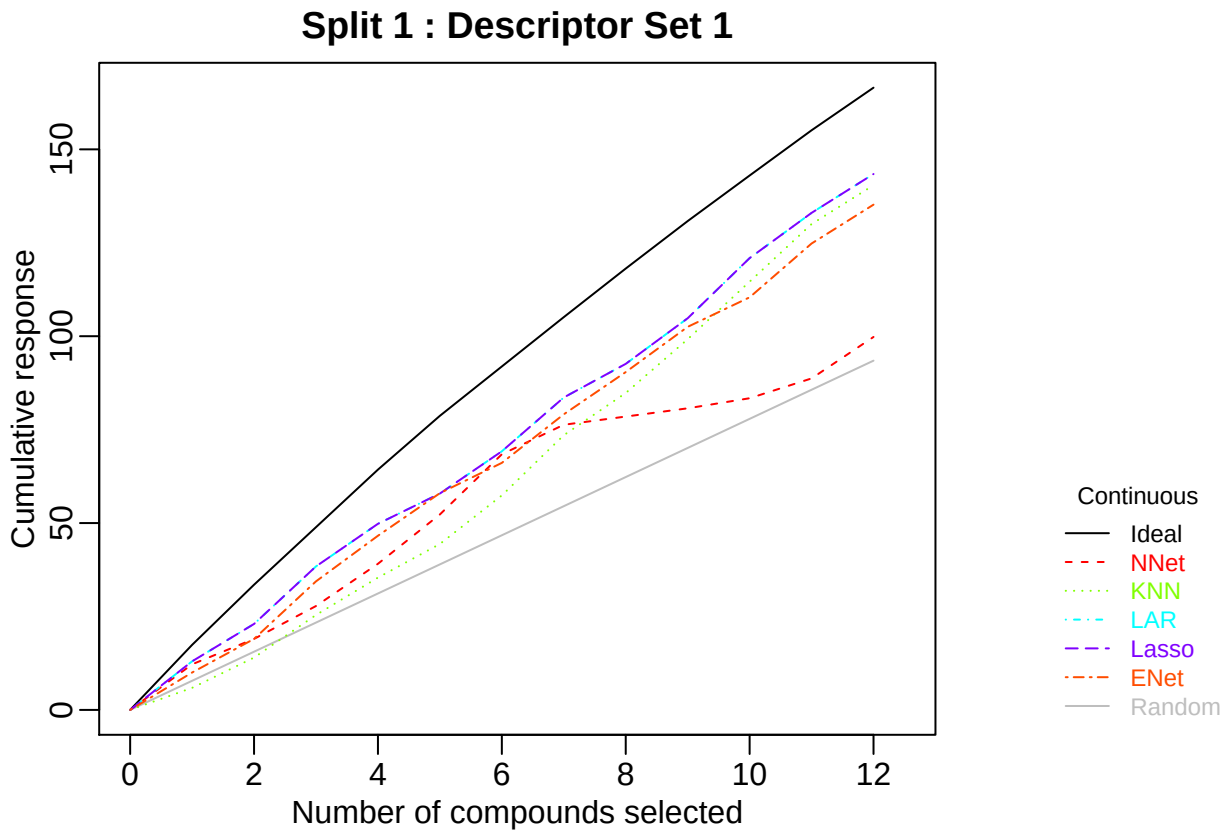
```
cml <- ModelTrain(USArrests, models = c("KNN", "LAR", "NNet", "ENet", "Lasso", "LassoGLM"))
```

```
## Warning in value[[3L]](cond): WARNING...LassoGLM not run: 'arg' should be
## one of "link", "response", "coefficients", "nonzero", "class"

## Warning in value[[3L]](cond): WARNING...LassoGLM not run: 'arg' should be
## one of "link", "response", "coefficients", "nonzero", "class"

## Warning in value[[3L]](cond): WARNING...LassoGLM not run: 'arg' should be
## one of "link", "response", "coefficients", "nonzero", "class"
```

```
plot(cml, splits = 1, meths = NULL)
```




```
CombineSplits(cml)
```

Multiple Comparisons Similarity (MCS) Plot