

Quick start guide for the `ncvreg` package

Patrick Breheny

December 13, 2016

This guide is intended to briefly demonstrate the basic usage of `ncvreg`. For more details, see the documentation for individual functions, as well as the references.

`ncvreg` comes with a few example data sets; we'll look at `prostate`, which has 8 features and one continuous response, `prostate$lpsa`, the PSA levels (on the log scale) from men about to undergo radical prostatectomy. The data is available as a data frame; we will turn it into a design matrix `X` and response vector `y` for the purpose of analysis

```
> ## Linear regression
> data(prostate)
> X <- as.matrix(prostate[,1:8])
> y <- prostate$lpsa
> head(X)

      lcavol  lweight age      lbph svi      lcp gleason  pgg45
1 -0.5798185 2.769459  50 -1.386294  0 -1.386294      6      0
2 -0.9942523 3.319626  58 -1.386294  0 -1.386294      6      0
3 -0.5108256 2.691243  74 -1.386294  0 -1.386294      7     20
4 -1.2039728 3.282789  58 -1.386294  0 -1.386294      6      0
5  0.7514161 3.432373  62 -1.386294  0 -1.386294      6      0
6 -1.0498221 3.228826  50 -1.386294  0 -1.386294      6      0

> head(y)

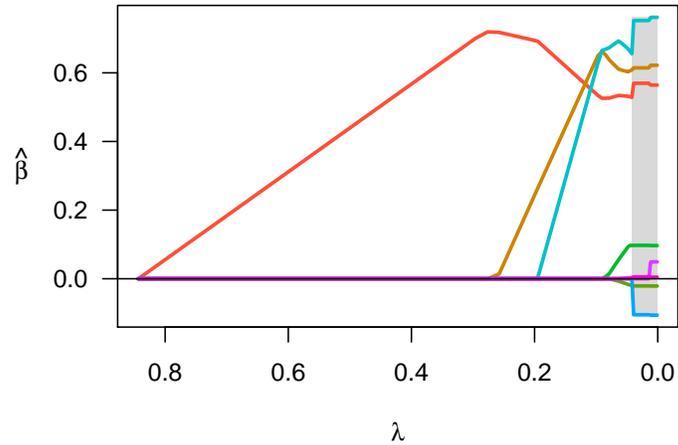
[1] -0.4307829 -0.1625189 -0.1625189 -0.1625189  0.3715636  0.7654678
```

To fit a penalized regression model to this data:

```
> fit <- ncvreg(X, y)
```

The default penalty here is the minimax concave penalty (MCP), but SCAD and lasso penalties are also available. This produces a path of coefficients, which we can plot with

```
> plot(fit)
```



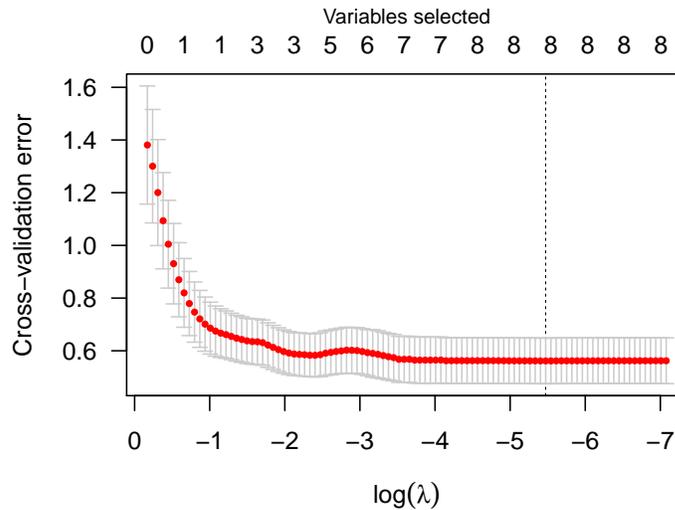
Notice that variables enter the model one at a time, and that at any given value of λ , several coefficients are zero. To see what the coefficients are, we could use the `coef` function:

```
> coef(fit, lambda=0.1)
```

(Intercept)	lcavol	lweight	age	lbph	svi
-0.6973059	0.5387509	0.6382717	0.0000000	0.0000000	0.6102800
lcp	gleason	pgg45			
0.0000000	0.0000000	0.0000000			

Typically, one would carry out cross-validation for the purposes of assessing the predictive accuracy of the model at various values of λ :

```
> cvfit <- cv.ncvreg(X, y)
> plot(cvfit)
```



The coefficients corresponding to the value of λ that minimizes the cross-validation error can be obtained via `coef`:

```
> coef(cvfit)

(Intercept)      lcavol      lweight      age      lbph
0.181560922  0.564341284  0.622019787 -0.021248184  0.096712519
          svi      lcp      gleason      pgg45
0.761673355 -0.106050929  0.049227917  0.004457512
```

Predicted values can be obtained via `predict`, which has a number of options:

```
> predict(cvfit, X=head(X))

      1      2      3      4      5      6
0.8229078 0.7612550 0.4416131 0.6199877 1.7315458 0.8434007

> predict(cvfit, type="nvars")

0.00420
      8
```

Note that the original fit (to the full data set) is returned as `cvfit$fit`; it is not necessary to call both `ncvreg` and `cv.ncvreg` to analyze a data set. Methods for logistic regression and Cox proportional hazards regression are also available.