Assignment 2 FAQ and Hints

Insurance prediction contest

How can I get some background on what is being sold?

Here is an NPR podcast (and transcript) about insurance sales to poor rural farmers, similar to what is being sold here to poor, rural Chinese farmers.

Some info about the variables:

- `**region**` code for region of household and farm
- `**village**` code for the village
- `**takeup**` whether they bought farmers insurance for this season -- our outcome
- `**age**` of head of household
- `**agpop**` is the number of people in the household
- `**rice_inc**` is a measure of income from selling rice last year
- `**ricearea_2010**` the size of the rice cultivation area
- `**educ**` shows education level (0 = illiteracy, 1 = primary, 2 = secondary, 3 = high school, 4 = college)
- `**educ_good**` is just an indicator for education being non-zero 1(educ > 0)
- `**disaster_loss**` is the loss in cultivation area from a disaster last year
- `**disaster_yes**` is just an indicator for whether they were affected by a disaster last year
- `**general_trust**` shows how much the farmer trusts government in general
- `**literacy**` is an indicator for literacy
- `***-missing**` is an indicator for the * variable being missing for the farmer (* could be age, agpop, etc.)

I want to better understand the penalized regression methods. What could help?

Chapter 6 (especially Section 6.2) of James et al.'s *Introduction to Statistical Learning* has additional details about these methods. Lab sections 6.6.1 and 6.6.2 illustrate this with `glmnet`, which was written by some of the authors.

You may also find this short video we produced helpful.

Another nice resource is this Coursera lecture.

How can I specify interactions and other basis expansions in my formula?

Here is a basic tutorial on model formulas in R.

In addition to the methods discussed there, some other useful functions you can use in formulas are:

- `poly`: add polynomial terms; for example, `poly(x, 3)` adds centered and scaled linear, squared, and cubic terms.
- `ns` (in the `splines` package): add a "natural spline", which can be a simple way to fit non-linear functions.
- `factor`: add a column for each unique value.
- Simple transformations such as `log` and `sqrt`.

How do you get the value of lambda that minimizes the mean cross-validated error from your penalized regression model?

Use the command `cv.1$lambda.min` where cv.1 is the name of your model created with the `cv.glmnet` function. Note that this gives the value of lambda, whereas the x-axis in `plot(cv.1)` graph shows the log(lambda).

As a side note, in some cases you may want to use `lambda.1se` instead of `lambda.min`. `lambda.1se` is the largest value of lambda such that the mean cross-validation error is still within 1 standard error of the minimum cross-validation error. Using `lambda.1se` makes your model slightly simpler, as it imposes a larger penalty on the regression model.

How do you look at the coefficients of the cross-validated penalized regression model?

Use the command `coef(cv.1)` where cv.1 is the name of your model created with the `cv.glmnet()` function.

You may want to specify a specific value of lambda to see the coefficients for with `coef(cv.1, s = 0.1)` or `coef(cv.1, s = cv.1$lambda.1se)` or `coef(cv.1, s = cv.1$lambda.min)`. Note that if you do not specify the value of lambda, the default is `cv.1$lambda.1se`.

My model matrix for the test data seems to have the wrong number of columns. There is an extra column for the additional region and I can't get predictions. What should I do?

You may have used `model.matrix`, rather than `sparse.model.matrix` to create the model matrices. They have somewhat different defaults about what to do with levels of a factor that

don't exist in the data. Try using `sparse.model.matrix`, which will include a column for region 3.

More generally, it is worth thinking about how you would want to get predictions about a new region. Should you treat it like region 1? Region 2? The average of the two regions?

How do you plot the takeup rate predictions against the actual takeup rates?

If you want to use R, you can plot using `ggplot2` function:

```
ggplot(
  aes(x = takeup.hat, y = takeup),
  data = data
) +
  coord_cartesian(ylim = c(0, 1)) +
  geom_point(alpha = .3, pch = "|") +
  geom_smooth()
```

You could also "bucket" `takeup.hat` into a series of groups and look at the rates for each, as we did in the Caravan example in the lab. You can also do either of these methods in Tableau.

This is a way of studying the *calibration* of the model. To quote Nate Silver:

> *"One of the most important tests of a forecast — I would argue that it is the single most important one — is called calibration. Out of all the times you said there was a 40 percent chance of rain, how often did rain actually occur? If over the long run, it really did rain about 40 percent of the time, that means your forecasts were well calibrated."*

Can I use cross-validation to choose α (alpha), which controls the mix of the L1 and L2 penalties?

The package we are using (`glmnet`) does not support this directly. However, you can use other packages, such as `caret`, that do support this (and also allow choosing other hyperparameters in other statistical machine learning methods).

Do I have to use penalized regression or can I use other methods?

We recommend most students start with what we've taught in class. But you are welcome to use more advanced methods. The data is almost entirely real data, so it isn't designed to favor any one method.