homework_03

Due Feb 27

Code To receive full credit all unit tests must pass and one copy of the exercises must be completed.

Exercises: 5.4.2, 5.4.3, 5.6.1, 5.7, 5.9.1, 5.12.1, 5.12.2, 5.14.1, 5.19.1

To start

Clone the repo into a local directory named `homework_03`. Do not use the original repo name. Replace `X` below with your team name.

```
git clone https://github.com/columbia-applied-data-science/homework_03_team_X.git \
  homework_03
```

See `demo.py` and the tests to get an idea of how things work.

Numerical techniques

See this section in the lecture notes information about pseudo inverses.

5-fold cross validation

You will make a 5-fold cross validation module. This is used as a way to pick out your regularization parameter delta. Our 5-fold cross validation is:

For every delta:

1. Divide the data up into 5 equal chunks
2. Pick out the first chunk as a cross-validation set, and group the other 4 together as training data.
3. Fit the model using the training data and use the cross validation set to measure both the training and cross-validation squared error |Xw - Y|^2

4. Repeat 5 times, each time using a different chunk as the cross validation set.

5. Average the training and cross-validation errors across the 5 folds.

Compare the average cross-validation errors and use this to choose delta. Note that the training error should not be used to choose delta. It is there to serve as a reality check and to diagnose the degree of over/under fitting.

Caution!

These routines are very picky about array shape. Some functions, e.g. np.dot, return arrays who have shape = (N,) (a tuple with only one element). In that case, you will often have to reshape this into a proper two dimensional array. The docstring for linear_reg.fit() tells you when to do this.

Two functions, linear_reg.fit() and cross_validator.cross_val() can handle pandas objects as their input. The others may or may not. However, these are the only public methods in their modules, so this is ok.