CSE 446: Homework 1

ayush29f@cs.washington.edu
Ayush Saraf

Problem 1

Let $T = Test\ is\ positive, D = Person\ has\ the\ disease$

$$P(D) = 10^{-4}$$

$$P(T|D) = 0.99$$

Now we have to calculate $P(D|T)$ using Bayes' Rule;

$$P(D|T) = \frac{P(T|D)P(D)}{P(T)}$$

$$P(D|T) = \frac{P(T|D)P(D)}{P(T|D)P(D) + P(T|D')P(D')}$$

$$P(D|T) = \frac{0.99 * 10^{-4}}{0.99 * 10^{-4} + (1 - 0.99) * (1 - 10^{-4})}$$

$$P(D|T) = 0.0098$$

Problem 2.1

In order to solve this question we will maximize the log likelihood (L) of the this distribution to be part of Poisson Distribution.

$$L(\lambda|\boldsymbol{x}) = \prod_{i=1}^{6} Poi(x_i|\lambda)$$

$$L(\lambda|\boldsymbol{x}) = \prod_{i=1}^{6} e^{-\lambda}\frac{\lambda^{x_i}}{x_i!}$$

Taking Log on both sides,

$$LL(\lambda|\boldsymbol{x}) = \sum_{i=1}^{6} -\lambda + x_i ln(\lambda) - ln(x_i!)$$

Setting the derivative to 0,

$$\frac{\partial LL(\lambda|\boldsymbol{x})}{\partial\lambda} = \sum_{i=1}^{6} \frac{x_i}{\lambda} - 1 = 0$$

$$\lambda = \sum_{i=1}^{6} \frac{x_i}{6}$$

$$\lambda = \frac{13}{6}$$

Checking 2nd derivative for maxima or minima,

$$\frac{\partial^2 LL(\lambda|\boldsymbol{x})}{\partial \lambda^2} = \sum_{i=1}^{6} \frac{-x_i}{\lambda^2} < 0$$

plugging $\lambda = 13/6$, and since $x_i > 0$ this is a maxima and therefore maximizing Likelihood at $\lambda = 13/6$

Problem 2.2

Given, $n_{clear} = 11$, $n_{cloudy} = 24$, $n_{rain} = 75$
Calculate p_{clear}, p_{cloudy}, p_{rain} We know,

$$p_{clear} + p_{cloudy} + p_{rain} = 1$$

$$L(p_{clear}, p_{cloudy}|n_{clear}, n_{cloudy}, n_{rain}) = \frac{110!}{11! * 24! * 75!} p_{clear}^{11} p_{cloudy}^{24} (1 - p_{clear} - p_{cloudy})^{75}$$

Taking Log on both sides,

$$LL(p_{clear}, p_{cloudy}|\boldsymbol{n}) = log\Big(\frac{110!}{11! * 24! * 75!}\Big) + 11log(p_{clear}) + 24log(p_{cloudy}) + 75log(1 - p_{clear} - p_{cloudy})$$

Take partial derivatives and set them to 0

$$\frac{\partial LL(p_{clear}, p_{cloudy}|\boldsymbol{n})}{\partial p_{clear}} = \frac{11}{p_{clear}} - \frac{75}{(1 - p_{clear} - p_{cloudy})} = 0$$

$$\frac{\partial LL(p_{clear}, p_{cloudy}|\boldsymbol{n})}{\partial p_{cloudy}} = \frac{24}{p_{cloudy}} - \frac{75}{(1 - p_{clear} - p_{cloudy})} = 0$$

Calculating for 2 variables and 2 equations,

$$p_{clear} = \frac{11}{11 + 24 + 75} = \frac{11}{110}$$

$$p_{cloudy} = \frac{24}{11 + 24 + 75} = \frac{24}{110}$$

$$p_{rain} = 1 - p_{clear} - p_{cloudy} = \frac{75}{110}$$

Checking 2nd derivative for confirming it is maxima for these values of p.

Problem 3.1

In order to estimate we only need 2 statistics from the ones shown above: $C_{xx}^{(n)}$, $C_{xy}^{(n)}$

Problem 3.2

In order to estimate we only need 4 statistics from the ones shown above: $\bar{x}^{(n)}$, $\bar{y}^{(n)}$, $C_{xx}^{(n)}$, $C_{xy}^{(n)}$

Problem 3.3

According to the definition of mean,

$$\bar{x}^{(n+1)} = \frac{1}{n+1}\sum_{i=1}^{n+1} x_i$$

$$(n+1)\bar{x}^{(n+1)} = \sum_{i=1}^{n+1} x_i$$

$$\frac{(n+1)}{n}\bar{x}^{(n+1)} = \frac{1}{n}\sum_{i=1}^{n+1} x_i$$

$$\frac{(n+1)}{n}\bar{x}^{(n+1)} = \frac{1}{n}\sum_{i=1}^{n} x_i + \frac{x_{n+1}}{n}$$

$$\frac{(n+1)}{n}\bar{x}^{(n+1)} = \bar{x}^{(n)} + \frac{x_{n+1}}{n}$$

$$\frac{(n+1)}{n}\bar{x}^{(n+1)} = \bar{x}^{(n)} + \frac{x_{n+1}}{n}$$

$$\bar{x}^{(n+1)} = \frac{n}{(n+1)}\Big(\bar{x}^{(n)} + \frac{x_{n+1}}{n}\Big)$$

$$\bar{x}^{(n+1)} = \frac{n\bar{x}^{(n)}}{(n+1)} + \frac{x_{n+1}}{n+1}$$

$$\bar{x}^{(n+1)} = \bar{x}^{(n)} - \bar{x}^{(n)} + \frac{n\bar{x}^{(n)}}{(n+1)} + \frac{x_{n+1}}{n+1}$$

$$\bar{x}^{(n+1)} = \bar{x}^{(n)} - \frac{\bar{x}^{(n)}}{(n+1)} + \frac{x_{n+1}}{n+1}$$

Hence Proved,

$$\bar{x}^{(n+1)} = \bar{x}^{(n)} + \frac{1}{n+1}\Big(x_{n+1} - \bar{x}^{(n)}\Big)$$

Similarly for $\bar{y}^{(n)}$

$$\bar{y}^{(n+1)} = \frac{1}{n+1}\sum_{i=1}^{n+1} y_i$$

$$(n+1)\bar{y}^{(n+1)} = \sum_{i=1}^{n+1} y_i$$

$$\frac{(n+1)}{n}\bar{y}^{(n+1)} = \frac{1}{n}\sum_{i=1}^{n+1} y_i$$

$$\frac{(n+1)}{n}\bar{y}^{(n+1)} = \frac{1}{n}\sum_{i=1}^{n} y_i + \frac{y_{n+1}}{n}$$

$$\frac{(n+1)}{n}\bar{y}^{(n+1)} = \bar{y}^{(n)} + \frac{y_{n+1}}{n}$$

$$\frac{(n+1)}{n}\bar{y}^{(n+1)} = \bar{y}^{(n)} + \frac{y_{n+1}}{n}$$

$$\bar{y}^{(n+1)} = \frac{n}{(n+1)}\Big(\bar{y}^{(n)} + \frac{y_{n+1}}{n}\Big)$$

$$\bar{y}^{(n+1)} = \frac{n\bar{y}^{(n)}}{(n+1)} + \frac{y_{n+1}}{n+1}$$

$$\bar{y}^{(n+1)} = \bar{y}^{(n)} - \bar{y}^{(n)} + \frac{n\bar{y}^{(n)}}{(n+1)} + \frac{y_{n+1}}{n+1}$$

$$\bar{y}^{(n+1)} = \bar{y}^{(n)} - \frac{\bar{y}^{(n)}}{(n+1)} + \frac{y_{n+1}}{n+1}$$

Hence Proved,

$$\bar{y}^{(n+1)} = \bar{y}^{(n)} + \frac{1}{n+1}\Big(y_{n+1} - \bar{y}^{(n)}\Big)$$

Problem 3.4

According to the definition of C_{xy},

$$C_{xy}^{(n+1)} = \frac{1}{n+1}\sum_{i=1}^{n+1}(x_i - \bar{x}^{(n+1)})(y_i - \bar{y}^{(n+1)})$$

$$C_{xy}^{(n+1)} = \frac{1}{n+1}\sum_{i=1}^{n+1}(x_i y_i - x_i\bar{y}^{(n+1)} - y_i\bar{x}^{(n+1)} + \bar{x}^{(n+1)}\bar{y}^{(n+1)})$$

$$C_{xy}^{(n+1)} = \frac{1}{n+1}\sum_{i=1}^{n+1}\Big(x_i y_i - x_i\bar{y}^{(n+1)} - y_i\bar{x}^{(n+1)}\Big) + \bar{x}^{(n+1)}\bar{y}^{(n+1)}$$

$$C_{xy}^{(n+1)} = \frac{1}{n+1}\sum_{i=1}^{n+1} x_i y_i - \frac{1}{n+1}\sum_{i=1}^{n+1} x_i \bar{y}^{(n+1)} - \frac{1}{n+1}\sum_{i=1}^{n+1} y_i \bar{x}^{(n+1)} + \bar{x}^{(n+1)}\bar{y}^{(n+1)}$$

$$C_{xy}^{(n+1)} = \frac{1}{n+1}(\sum_{i=1}^{n+1} x_i y_i) - 2\bar{x}^{(n+1)}\bar{y}^{(n+1)} + \bar{x}^{(n+1)}\bar{y}^{(n+1)}$$

$$C_{xy}^{(n+1)} = \frac{1}{n+1}(\sum_{i=1}^{n+1} x_i y_i) - \bar{x}^{(n+1)}\bar{y}^{(n+1)}$$

$$C_{xy}^{(n+1)} = \frac{x_{n+1}y_{n+1}}{n+1} + \frac{1}{n+1}(\sum_{i=1}^{n} x_i y_i) - \bar{x}^{(n+1)}\bar{y}^{(n+1)}$$

$$C_{xy}^{(n+1)} = \frac{x_{n+1}y_{n+1}}{n+1} + \frac{1}{n+1}(\sum_{i=1}^{n} x_i y_i - x_i\bar{y}^{(n)} - y_i\bar{x}^{(n)} + \bar{x}^{(n)}\bar{y}^{(n)} + x_i\bar{y}^{(n)} + y_i\bar{x}^{(n)} - \bar{x}^{(n)}\bar{y}^{(n)}) - \bar{x}^{(n+1)}\bar{y}^{(n+1)}$$

$$C_{xy}^{(n+1)} = \frac{x_{n+1}y_{n+1}}{n+1} + \frac{1}{n+1}(\sum_{i=1}^{n} x_i y_i - x_i\bar{y}^{(n)} - y_i\bar{x}^{(n)} + \bar{x}^{(n)}\bar{y}^{(n)}) + \frac{1}{n+1}\sum_{i=1}^{n} x_i\bar{y}^{(n)} +$$

$$\frac{1}{n+1}\sum_{i=1}^{n} y_i\bar{x}^{(n)} - \frac{1}{n+1}\sum_{i=1}^{n} \bar{x}^{(n)}\bar{y}^{(n)} - \bar{x}^{(n+1)}\bar{y}^{(n+1)}$$

$$C_{xy}^{(n+1)} = \frac{x_{n+1}y_{n+1}}{n+1} + \frac{nC_{xy}^{(n)}}{n+1} + \frac{1}{n+1}\sum_{i=1}^{n} x_i\bar{y}^{(n)} + \frac{1}{n+1}\sum_{i=1}^{n} y_i\bar{x}^{(n)} - \frac{1}{n+1}\sum_{i=1}^{n} \bar{x}^{(n)}\bar{y}^{(n)} - \bar{x}^{(n+1)}\bar{y}^{(n+1)}$$

$$C_{xy}^{(n+1)} = \frac{x_{n+1}y_{n+1}}{n+1} + \frac{nC_{xy}^{(n)}}{n+1} + \frac{n\bar{y}^{(n)}\bar{x}^{(n)}}{n+1} - \bar{x}^{(n+1)}\bar{y}^{(n+1)}$$

$$C_{xy}^{(n+1)} = \frac{1}{n+1}\Big(x_{n+1}y_{n+1} + nC_{xy}^{(n)} + n\bar{y}^{(n)}\bar{x}^{(n)} - (n+1)\bar{x}^{(n+1)}\bar{y}^{(n+1)}\Big)$$

According to the definition of C_{xx},

$$C_{xx}^{(n+1)} = \frac{1}{n+1}\sum_{i=1}^{n+1}(x_i - \bar{x}^{(n+1)})^2$$

$$C_{xx}^{(n+1)} = \frac{1}{n+1}\sum_{i=1}^{n+1}(x_i^2 + (\bar{x}^{(n+1)})^2 - 2x_i\bar{x}^{(n+1)})$$

$$C_{xx}^{(n+1)} = \frac{1}{n+1}(\sum_{i=1}^{n+1} x_i^2) - (\bar{x}^{(n+1)})^2$$

$$C_{xx}^{(n+1)} = \frac{n}{n(n+1)}(\sum_{i=1}^{n} x_i^2) + \frac{x_{n+1}^2}{n+1} - (\bar{x}^{(n+1)})^2$$

$$C_{xx}^{(n+1)} = \frac{n}{n(n+1)}(\sum_{i=1}^{n} x_i^2 + (\bar{x}^{(n)})^2 - (\bar{x}^{(n)})^2 - 2x_i\bar{x}^{(n)} + 2x_i\bar{x}^{(n)}) + \frac{x_{n+1}^2}{n+1} - (\bar{x}^{(n+1)})^2$$

$$C_{xx}^{(n+1)} = \frac{n}{n(n+1)}(\sum_{i=1}^{n}(x_i - \bar{x}^{n})^2 - (\bar{x}^{(n)})^2 + 2x_i\bar{x}^{(n)}) + \frac{x_{n+1}^2}{n+1} - (\bar{x}^{(n+1)})^2$$

$$C_{xx}^{(n+1)} = \frac{n}{n(n+1)}\sum_{i=1}^{n}(x_i - \bar{x}^{n})^2 - \frac{n}{n(n+1)}\sum_{i=1}^{n}(\bar{x}^{(n)})^2 + \frac{n}{n(n+1)}\sum_{i=1}^{n}2x_i\bar{x}^{(n)} + \frac{x_{n+1}^2}{n+1} - (\bar{x}^{(n+1)})^2$$

$$C_{xx}^{(n+1)} = \frac{n}{(n+1)}C_{xx}^{(n)} + \frac{n(\bar{x}^{(n)})^2}{(n+1)} + \frac{x_{n+1}^2}{n+1} - (\bar{x}^{(n+1)})^2$$

$$C_{xx}^{(n+1)} = \frac{1}{n+1}\Big(nC_{xx}^{(n)} + n(\bar{x}^{(n)})^2 + x_{n+1}^2 - (n+1)(\bar{x}^{(n+1)})^2\Big)$$

Problem 3.5

The two examples could be considered for online regression over batch regression:

1. Flight Price Prediction: Since there are so many flights with multiple incoming data points every minute across the world, it makes much more sense to have an online model that holds the statistics instead of recomputing the parameters

2. Twitter Retweet Prediction: Since there are so tweets and retweets with multiple incoming data points every millisecond across the world, it makes much more sense to have an online model that holds the statistics instead of recomputing the parameters

Problem 4.1.1

(a) The error on the training set will decrease compared to where $\lambda > 0$ because it will over fit to training set

(b) The error on the training set will increase compared to where $\lambda > 0$ because it will over fit to training set and not test set

(c) the magnitude of the values in $\hat{w}$ will be larger because we are not penalizing the $L1(\hat{w})$ in the loss function

(d) the number of non-zero elements in $\hat{w}$ will be more because there will be no feature selection

Problem 4.1.2

(a) The training error on the training set will increase because now a lot of features are eliminated and it will under fit to the training set

(b) The testing error will also increase for similar reasons given that we over estimated the λ because the model has eliminated too many features

(c) the magnitude of the values in $\hat{w}$ will be really small because we are penalizing the $L1(\hat{w})$ too much in the loss function

(d) the number of non-zero elements in $\hat{w}$ will be lower because there will be a lot of feature selection

Problem 4.2

1. Taking the derivative w.r.t. to $\hat{w}_i$

$$f(\hat{w}) = \lambda \|\hat{w}\|_1$$

$$f(\hat{w}) = \lambda \sum_i |\hat{w}_i|$$

$$\frac{\partial f(\hat{w})}{\partial \hat{w}_i} = \lambda \quad if \quad \hat{w}_i > 0$$

$$\frac{\partial f(\hat{w})}{\partial \hat{w}_i} = -\lambda \quad if \quad \hat{w}_i < 0$$

2. Taking the derivative w.r.t. to $\hat{w}_i$

$$f(\hat{w}) = \lambda \|\hat{w}\|_2$$

$$f(\hat{w}) = \lambda \sum_i \hat{w_i}^2$$

$$\frac{\partial f(\hat{w})}{\partial \hat{w}_i} = 2\lambda \hat{w}_i$$

3.
   - In the case of $\hat{w}_i < 1/2$, lasso will push more strongly again away from SSE solution because the differential has a magnitude of λ regardless the value of $\hat{w}_i$ unlike ridge where the magnitude of $2\lambda\hat{w}_i$ and if $\hat{w}_i < 1/2$ then the differential will be lower and will have lower impact.
   - Similarly, if $\hat{w}_i > 1/2$ ridge will have higher impact because now the $2\lambda\hat{w}_i > \lambda$.
   - If we have a really large λ then there will be a lot of regularization on the features in both cases. However, when we have a co related features ridge will end up pushing all the co-relating features to lower values and hence having a worse model. While in ridge regression since we use co-ordinate decent we will have one of the co-relating feature which is higher $\hat{w}_i$ value and others will be pushed to 0 and hence eliminating the relative features.

Lasso Regression

May 6, 2017

1 Lasso Regression

Using Lasso Regression to estimate `ViolentCrimesPerPop` based on the dataset provided

```
In [ ]: import pandas as pd
        import numpy as np
        import matplotlib.pyplot as plt
        from lasso import load_data
        %matplotlib inline

        # load all the data
        data = load_data('data', validation=True, split=0.9)
        X_train, y_train, X_val, y_val, X_test, y_test, df_train, df_test = data
        # X_train, y_train, X_test, y_test, df_train, df_test = load_data('data')
```

1.1 1. Building the Models

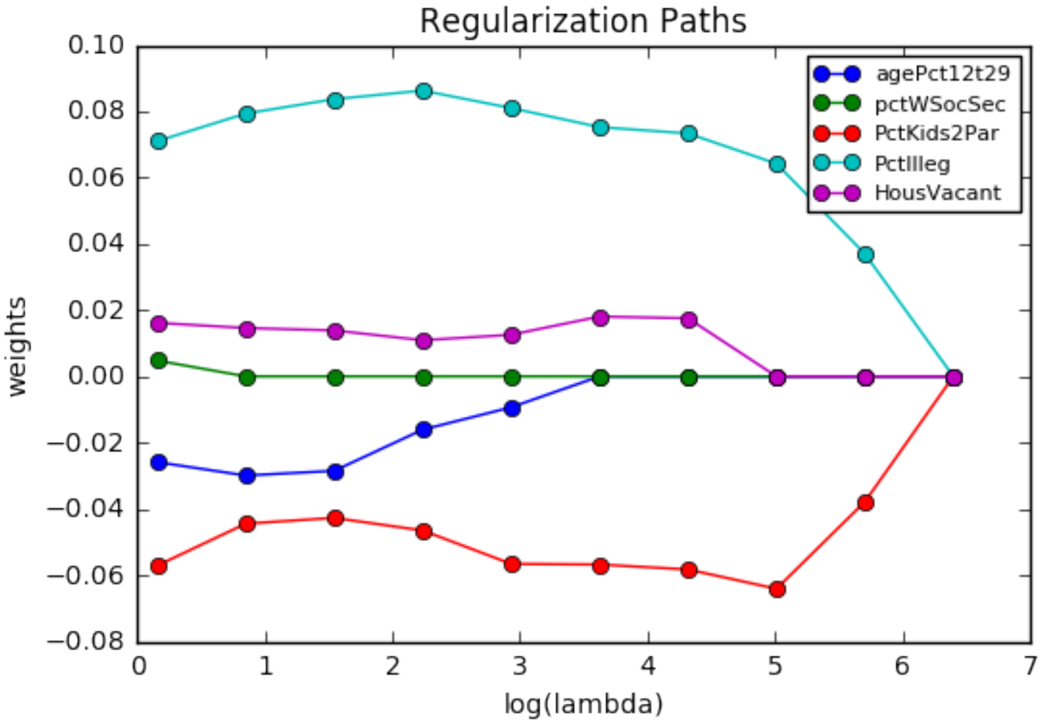
First we use the `lasso_models` method to get `W` that is a `10x95` matrix that has parameter for all 10 values of `lambda` or `reg`

```
In [2]: from lasso import lasso_models
        # build models for all 10 lamdas
        regs = np.array([600.0 / (2 ** i) for i in range(10)])
        W = lasso_models(X_train, y_train, regs)
```

1.2 2. Plot: Regularization Paths

The regularization paths (in one plot) for the coefficients for input variables agePct12t29, pctWSocSec, PctKids2Par, PctIlleg, and HousVacant — use log(λ) instead of λ.

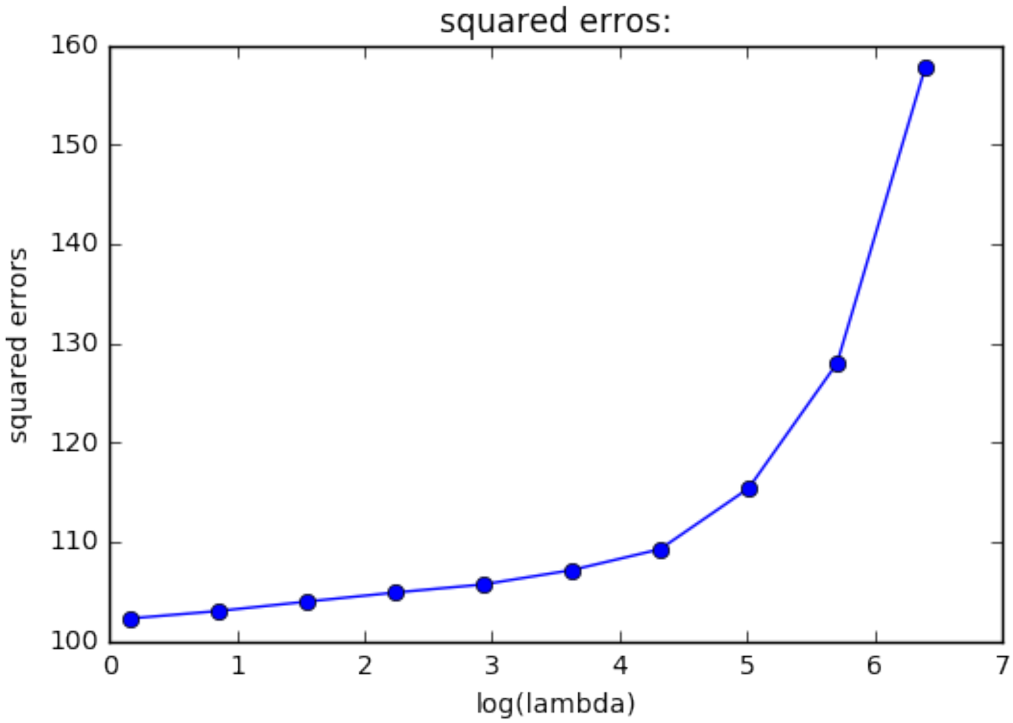
```
In [3]: from lasso import plot_regpath
        # plot regularization paths
        features = ['agePct12t29', 'pctWSocSec', 'PctKids2Par', \
                    'PctIlleg', 'HousVacant']
        ids = [df_train.columns.get_loc(feature) - 1 for feature in features]
        plot_regpath(W, regs, features, ids)
```



1.3 3. Plot: Squared Error In The Training Dataset

A plot of log(λ) against the squared error in the training data.

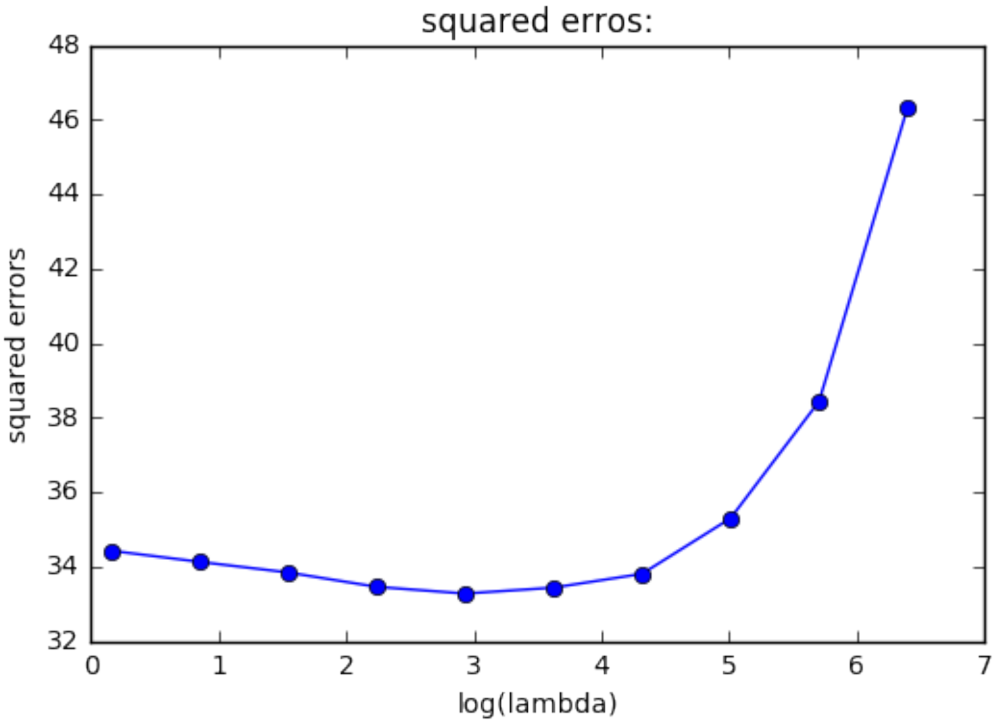
```
In [4]: from lasso import plot_sqerr
        # plot squared errors for training data
        plot_sqerr(X_train, y_train, W, regs)
```



1.4 4. Plot: Squared Error In Test Dataset

A plot of log(λ) against the squared error in the test data.

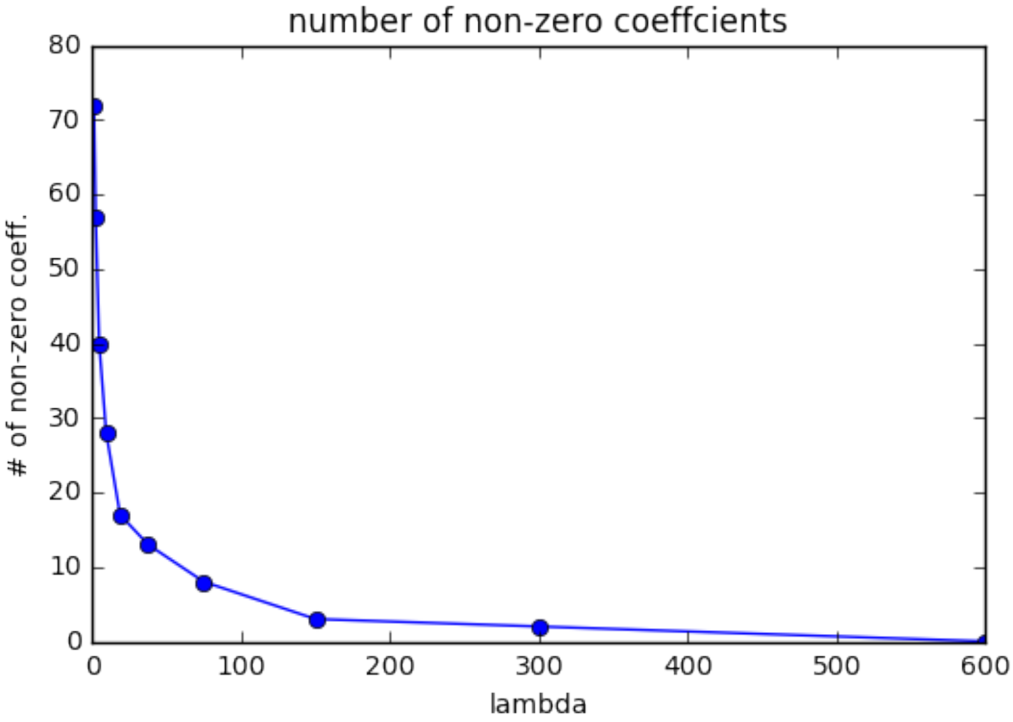
```
In [5]: # plot squared errors for testing data
        plot_sqerr(X_test, y_test, W, regs)
```



1.5 5. Plot: Number of Non-Zero Coefficients

A plot of λ against the number of nonzero coefficients

```
In [6]: from lasso import plot_nonzero
        # plot number of non-zero coeffcients
        plot_nonzero(W, regs)
```

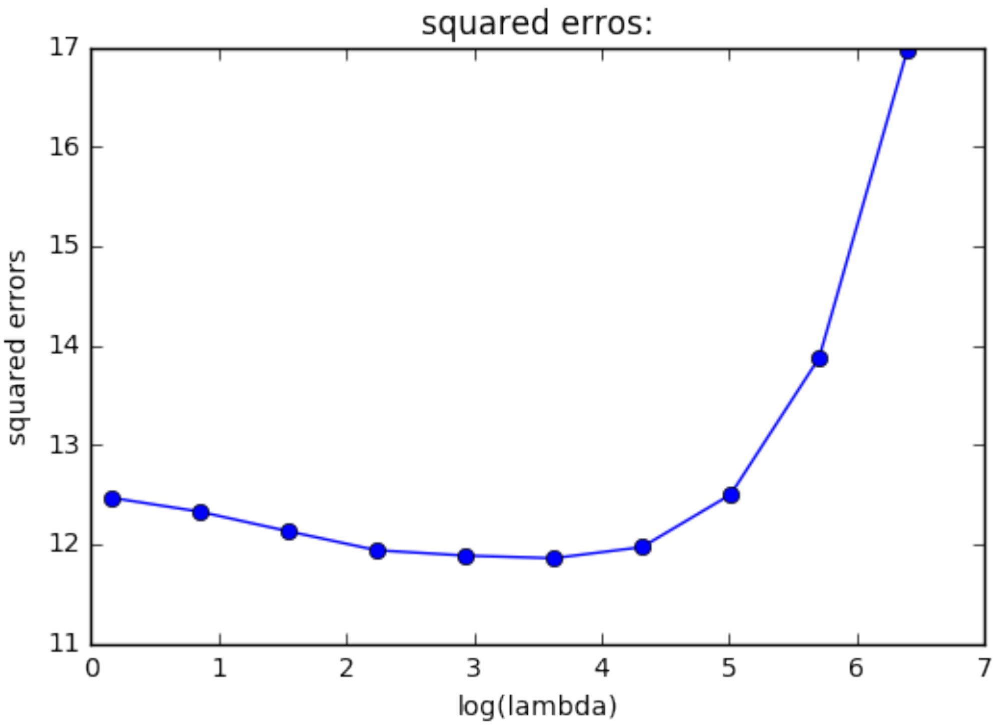


1.6 6. Hyper Parameter Tuning: λ

In this section, we will use the validation dataset to tune the hyper parameter λ.

We find the λ (reg) for which the validation error is the least. Since the model wasn't trained on the validation set and we have enough data we just use a 10% split on the original training set for the validation set. This gives us a good approximation for the error.

```
In [7]: from lasso import tune_reg
        plot_sqerr(X_val, y_val, W, regs)
        best_reg = tune_reg(X_val, y_val, W, regs)
        print 'best validation error at reg=' + str(best_reg)
```



```
best validation error at reg=37.5
```

1.7 7. Largest & Smallest Coefficient For Best λ

Maximum: PctIlleg: percentage of kids born to never married (numeric - decimal) Minimum: PctKids2Par: percentage of kids in family housing with two parents (numeric - decimal)

After looking at the largest positive weight it shows that houses wih higher 'percentage of kids born to never married' leads to a higher crime rate

After looking at the largest negative weight it shows that houses with higher 'percentage of kids in family housing with two parents' leads to lower crime rate

```
In [8]: from lasso import max_min_w
        max_w_i, min_w_i = max_min_w(df_train, W, regs, best_reg)
        print 'Maximum Parameter: ' + max_w_i + ' and Minimum Parameter: ' + min_
```

```
Maximum Parameter: PctIlleg and Minimum Parameter: PctKids2Par
```