1. Probabilistic Modeling

1. $P(1) = \mu_1,\ P(others) = 1 - \mu_1$

   $P(2) = \mu_2,\ P(others) = 1 - \mu_2$

   $P(3) = \mu_3,\ P(others) = 1 - \mu_3$

   $P(4) = \mu_4,\ P(others) = 1 - \mu_4$

   $P(5) = \mu_5,\ P(others) = 1 - \mu_5$

   $P(1) = \mu_6,\ P(others) = 1 - \mu_6$

2. It should be $\frac{1}{6}$, $\mu_1 = \mu_2 = \mu_3 = \mu_4 = \mu_5 = \mu_6 = \frac{1}{6}$

3. $P(1) = 0,\ P(2) = 1,\ P(3) = 0,\ P(4) = 0,\ P(5) = 0,\ P(1) = 0,$

4. The domain of the parameter should be [0,1](bigger or equal to 0 and smaller or equal to 1)

2. Weighted Squared Error

$$\frac{\partial E_{\hat{D}}(w)}{\partial w_i} = \sum_{n=1}^{N} \alpha_n \{t_n - w^T \phi(x_n)\} \phi(x_n)^T$$

Let $\frac{\nabla E_{\hat{D}}(w)}{\nabla w_i}$ be 0

$$\sum_{n=1}^{N} \alpha_n \{t_n - w^T \phi(x_n)\} \phi(x_n)^T = 0$$

$$0^T = \sum_{n=1}^{N} \alpha_n t_n \phi(x_n)^T - w^T \sum_{n=1}^{N} \alpha_n \phi(x_n)\, \phi(x_n)^T$$

let

$$\Phi_a = \begin{pmatrix} \alpha_1 \phi_0(x_1) & \cdots & \alpha_1 \phi_{M-1}(x_1) \\ \vdots & \ddots & \vdots \\ \alpha_N \phi_0(x_N) & \cdots & \alpha_N \phi_{M-1}(x_N) \end{pmatrix}$$

$$\Phi = \begin{pmatrix} \phi_0(x_1) & \cdots & \phi_{M-1}(x_1) \\ \vdots & \ddots & \vdots \\ \phi_0(x_N) & \cdots & \phi_{M-1}(x_N) \end{pmatrix}$$

$$\Phi_{\alpha}{}^{T}\vec{t} = w^{T}\Phi_{\alpha}{}^{T}\Phi$$

$$W_{ML\alpha} = (\Phi_{\alpha}{}^{T}\Phi)^{-1}\Phi_{\alpha}{}^{T}t$$

3. Training vs. Test Error

1. No, supposed the degree of the polynomial is very small, it can't regress the complex or fluctuated data points with high accuracy. So the training error will be large. If the validation data is more closed to the regression curve, the validation error will become smaller.
2. Yes, as degree 10 polynomial has more freedom. And actually if we set the w_{10} to 0, It is similar to the degree 9 polynomial.
3. No, if we regularized too much, for example, we set the λ positively infinite, the magnitude of w will be very small and performance of regression will be very bad. So the testing error will be larger.

4.1 Getting started

1. Niger has highest child mortality rate in 1990 with rate 313.7
2. Sierra Leone has highest child mortality rate in 2011 with rate 185.3
3. From the code below:

```
# Modify NaN values (missing values).
mean_vals = stats.nanmean(values, axis=0)
inds = np.where(np.isnan(values))
values[inds] = np.take(mean_vals, inds[1])
return (countries, features, values)
```

Fig.1 code for calculating the mean

We know it has taken an average of other values (axis=0 indicates the average of a column) and filled the missing data with this mean.

4.2 Polynomial Regression

1.

(1) Plot training error and test error (in RMS error) versus polynomial degree.

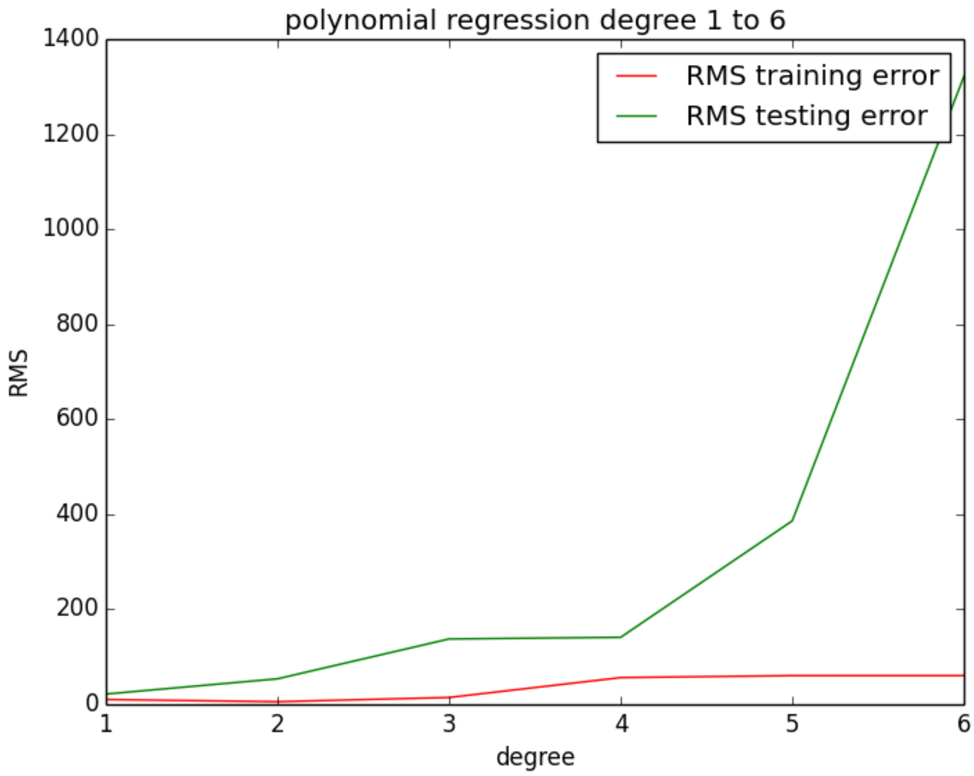

Fig.2 polynomial regression without normalization

The training error grows as the degree of the polynomial regression increase. It is incompatible to the fact that as the degree increases, the freedom of the ploy regression grows so the corresponding training error should decrease.

(2) Normalize the input features before using them.

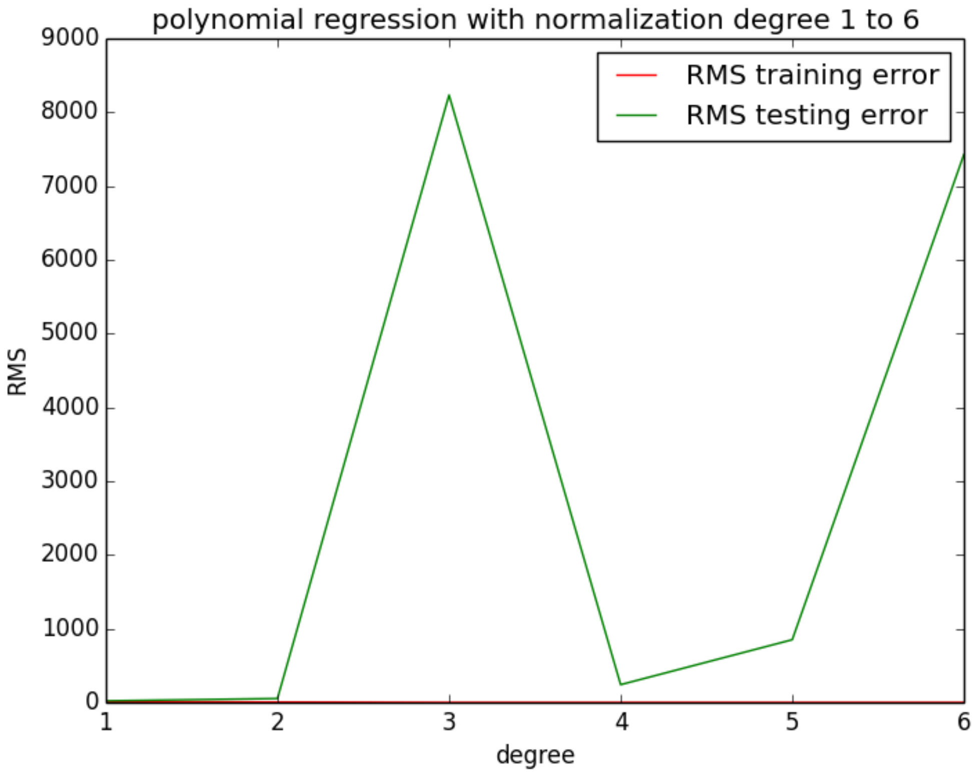

Fig.3 polynomial regression with normalization

2.

(1) Plot training error and test error (in RMS error) for each of the 8 features.

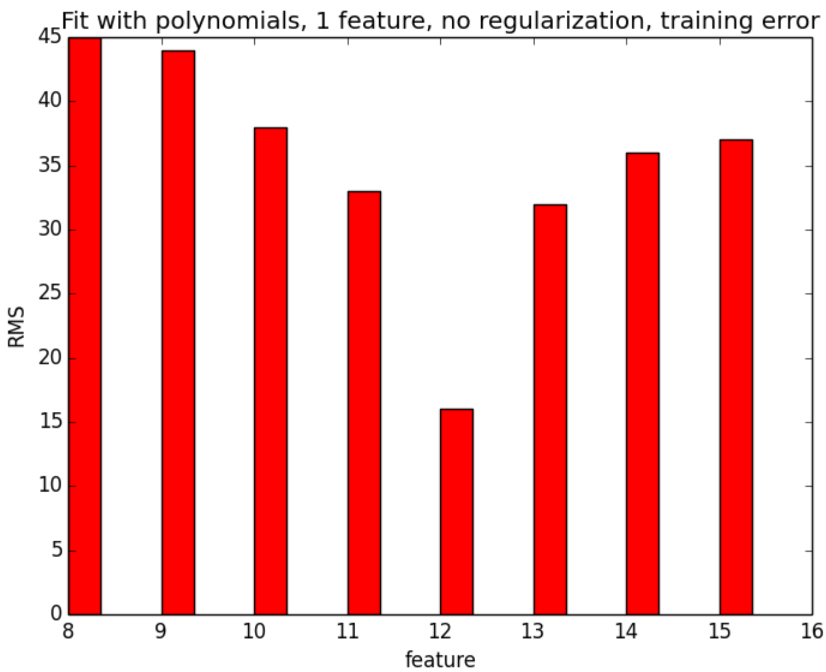

Fig.4 one dimension polynomial regression training error

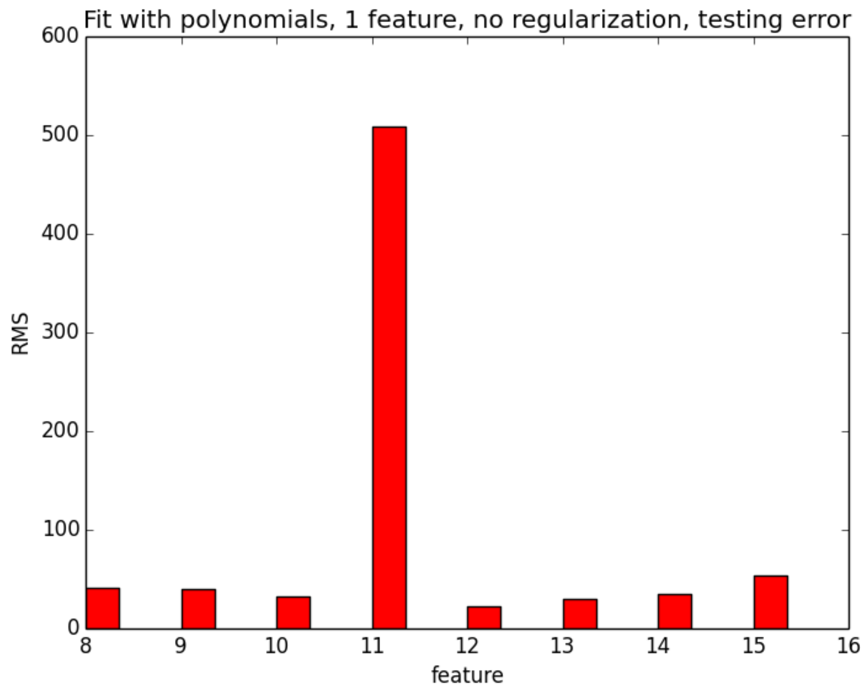

Fig.5 one dimension polynomial regression testing error

(2) Plots of the training data points, learned polynomial, and test data points.

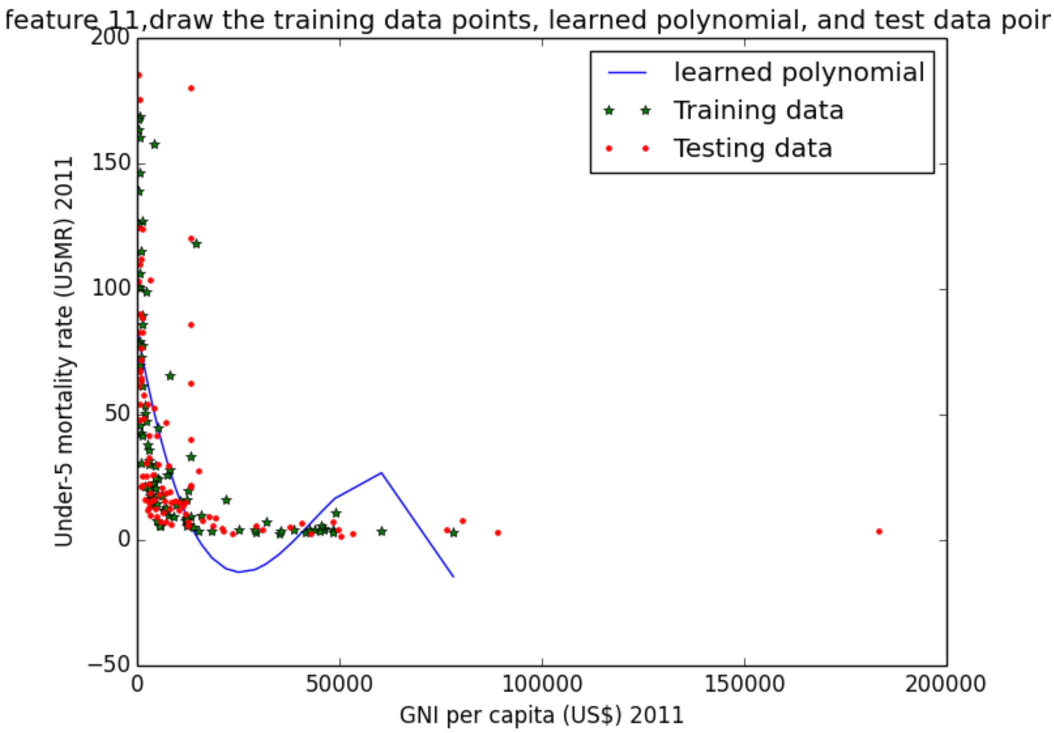

Fig.6 one dimension polynomial regression of feature 11, the training data points, learned polynomial, and test data points

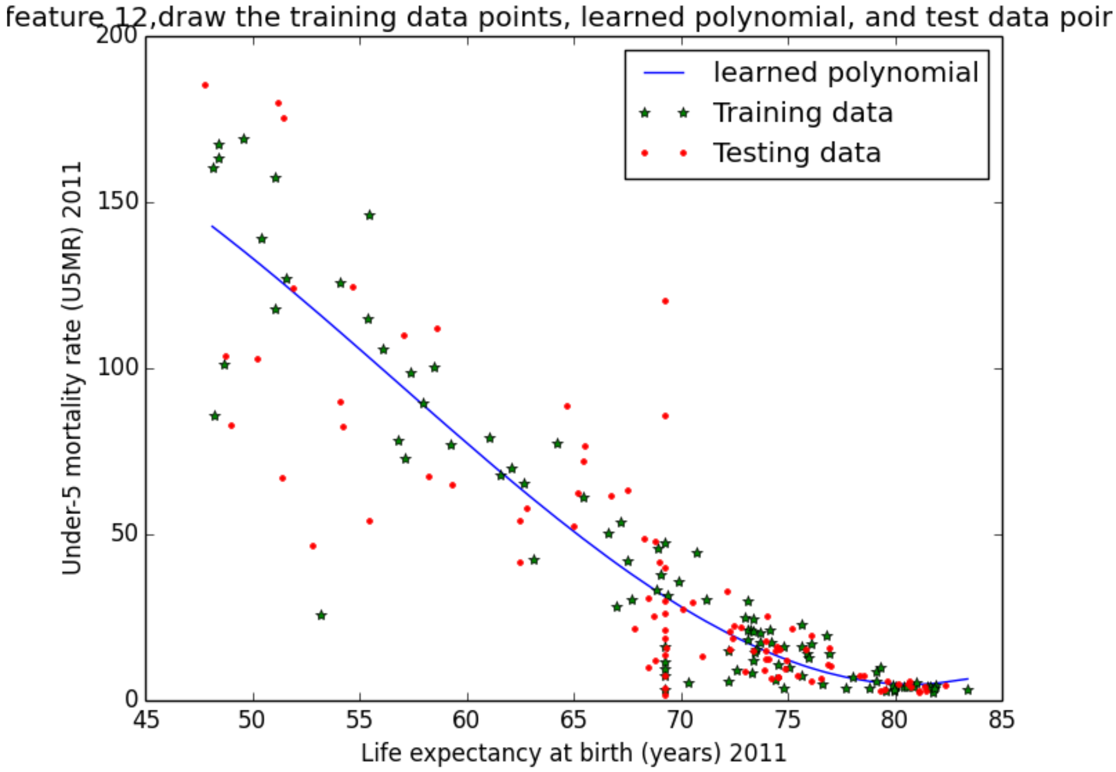

Fig.7 one dimension polynomial regression of feature 12, the training data points, learned polynomial, and test data points

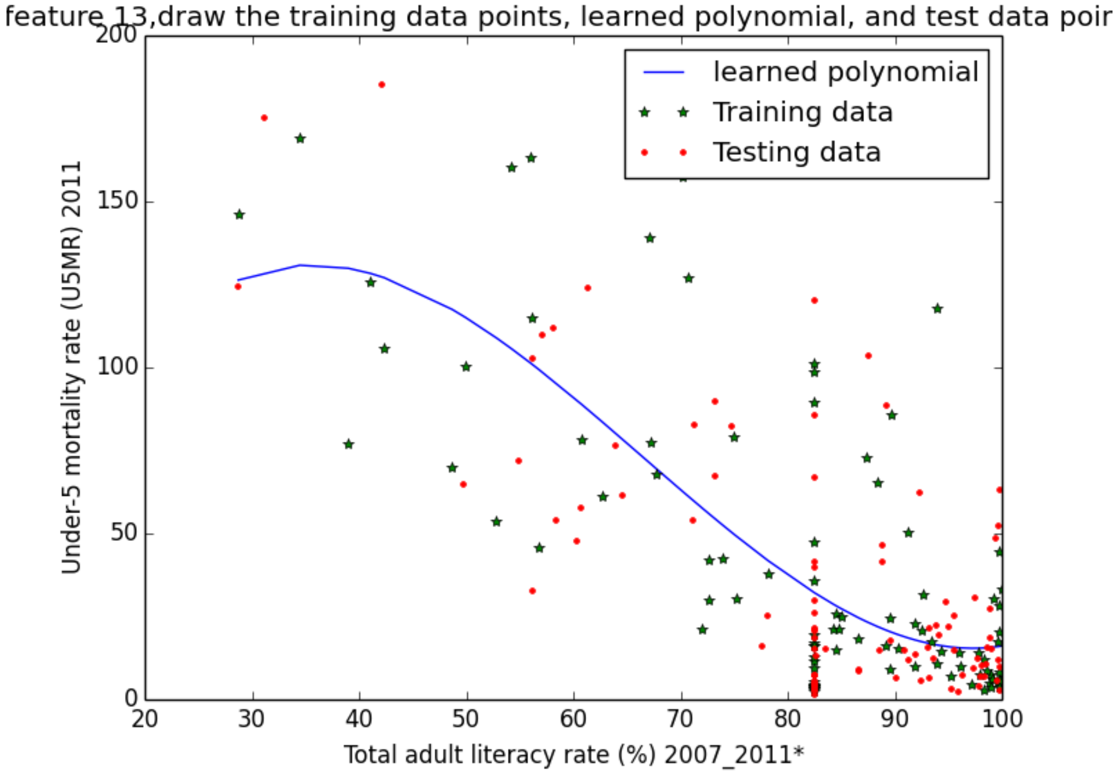

Fig.8 one dimension polynomial regression of feature 11, the training data points, learned polynomial, and test data points

4.3 Sigmoid Basis Functions

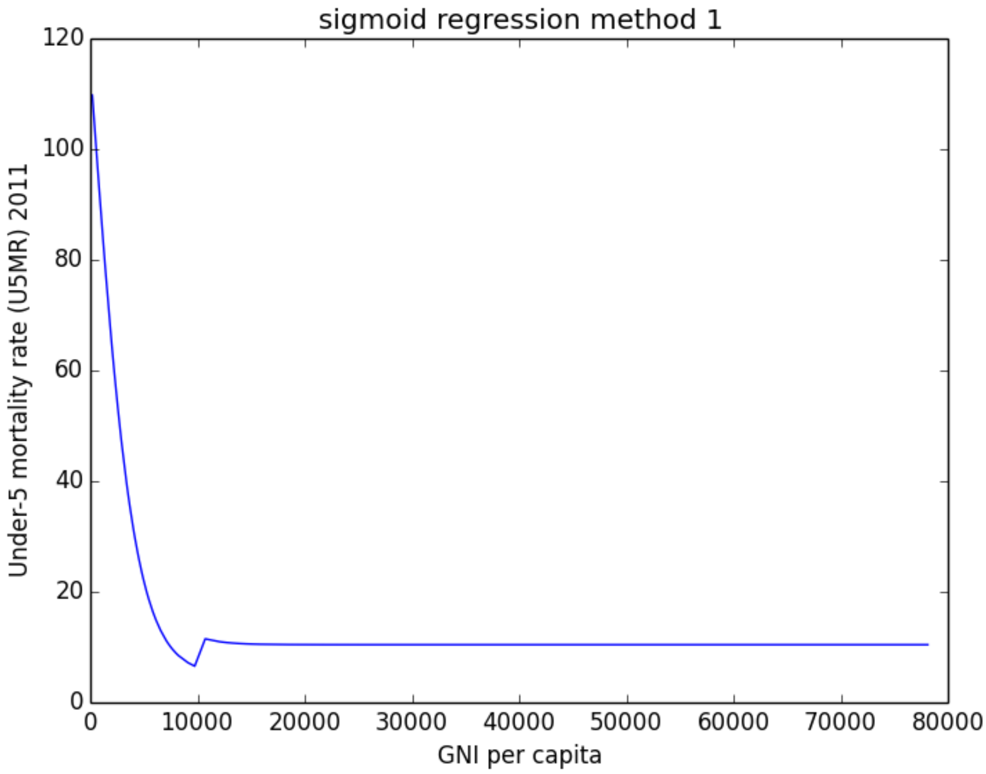

Fig.9 sigmoid regression using sigmoid basis functions

The training error of the regression is: 28.484635

The testing error of the regression is: 33.748573

4.4 Regularized Polynomial Regression

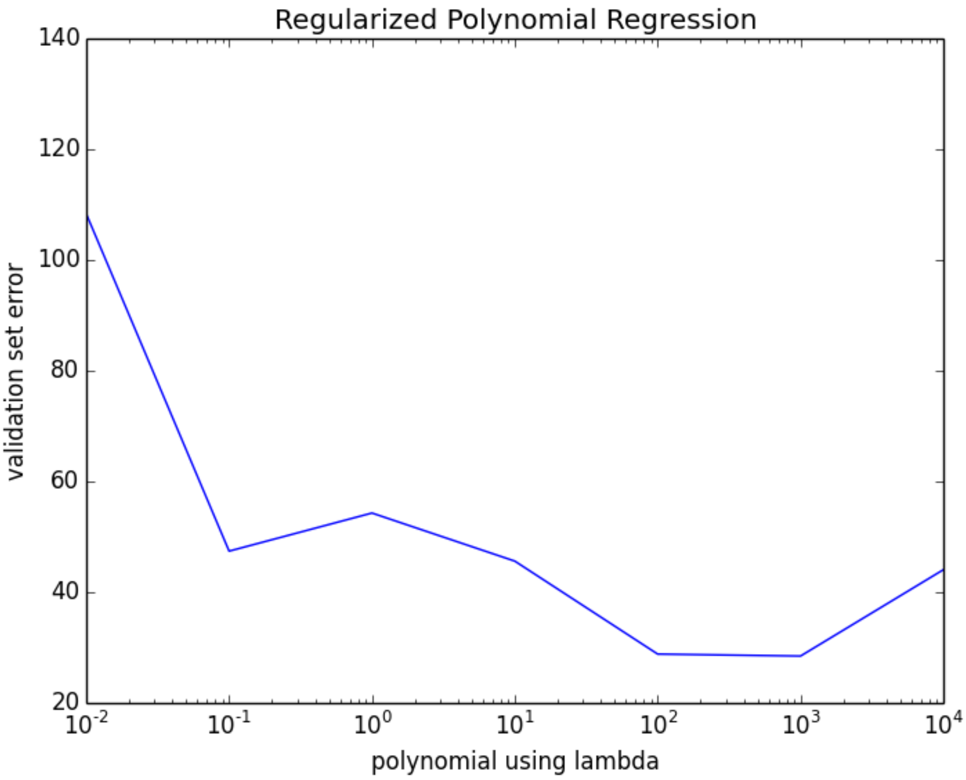

Fig.10 polynomial regression using lambda with cross validation

From the graph above, we could conclude that the best lambda is 10^3