Course Report: Regression

Nan He

1 Main objective of the analysis

In this report, I will be using one of my own data set. In chemistry, the interaction of gas molecules with a polarized surface is a interesting topic. It is especially suitable to be used as a model system to model weak interactions. I have a data set derived from numerically solving the electronic Schrodinger equations that listed the total energy of carbon monoxide (CO) on a salt (NaCl) surface. The data entries include 1) the C-O bond length, 2) the distance of CO center to the salt surface, 3) the angle of the CO molecule with the norm of the salt surface. And the label is the total energy at this position.

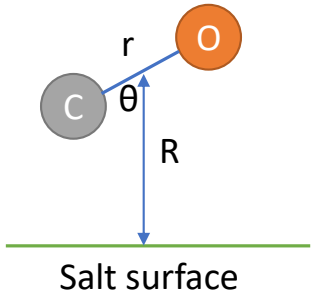

Figure 1: An geometrical illustration of the three properties for each data point.

Example data are:

	r	R	theta	Energy
0	0.876	2.236	0	-5706.803124
1	0.876	2.236	12	-5706.803137
2	0.876	2.236	24	-5706.815137
3	0.876	2.236	36	-5706.829497
4	0.876	2.236	48	-5706.842390

Figure 2: The raw data read directly from numerical experiments.

The data include the above three parameters, which uniquely defined a location of a CO molecule, and the energy measured at that location.

The objective of this report is to fit those data into a model, so we can predict the CO energy at any position, without conducting new numerical experiments since they are expensive.

2 Data exploration, cleaning, and feature engineering

The first step is to read raw data (in Excel spreadsheet) into a pandas data frame. This is done using xlrd package. The next step is to transform the angular parameter theta (θ) into a non-periodic variable. The simplest solution is to use $cos(\theta)$ instead of θ as the angular parameter, since $cos(\theta)$ defines a new unique parameter ranging from -1 to 1. Finally, we will drop all incomplete data (with any of the parameter empty). The final shape of the data frame is (4608,4).

	r	R	cos_theta	Energy
0	0.876	2.236	1.000000	-5706.803124
1	0.876	2.236	0.978148	-5706.803137
2	0.876	2.236	0.913545	-5706.815137
3	0.876	2.236	0.809017	-5706.829497
4	0.876	2.236	0.669131	-5706.842390

Figure 3: The data ready for analysis.

3 Testing different models

In this section, I will fit the energy using the three geometrical variables. I will test three regression models with the same train-test split. 80% of the data points will be used as training set, and the rest 20% as test set. The first model is a simple multi-variable linear regression. The model is fitted on the training set, the R^2 on the training set is 0.177. Then I use this on the test set, the predicted data (y_pred) are plotted against the real data (y_test). The R^2 is only 0.142, and the predicted and

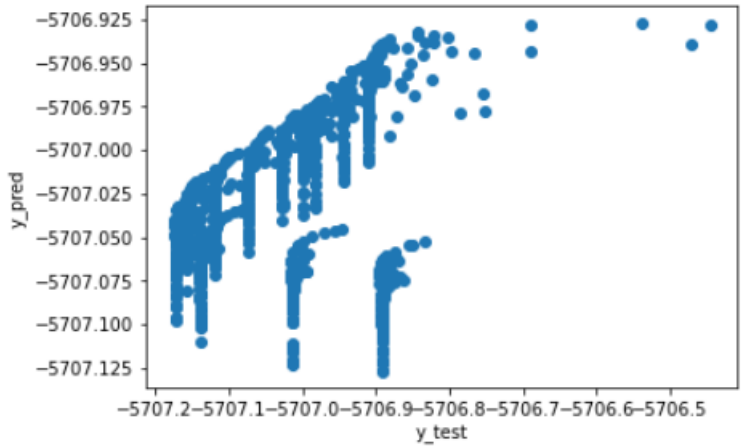

Figure 4: simple linear regression, $R^2 = 0.142$.

real data differs significantly, so the simple linear regression is not a suitable model to describe their relationship.

The second model is a linear regression with 4th-order polynomial features. I fit the model on the training set; the R^2 is 0.980. Similarly, I plotted the predicted data (y_pred) vesus the real data (y_test). The polynomial regression provides a much better results in term of the prediction accuracy

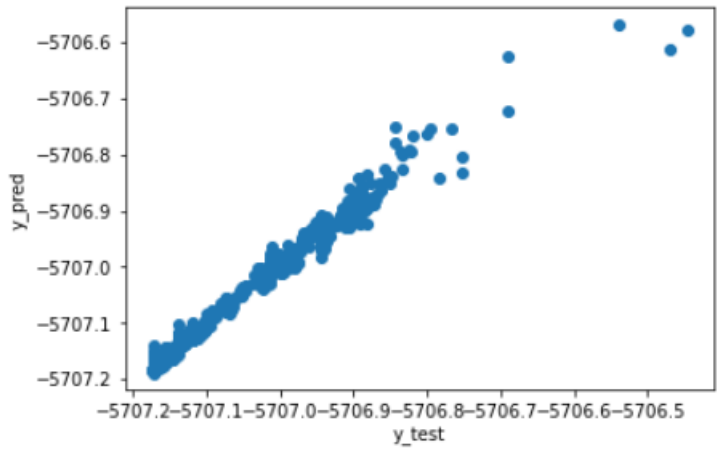

Figure 5: Polynomial regression (4th-order), $R^2 = 0.981$.

on the test set ($R^2 = 0.981$), and it has similar R^2 as in the training set. These results indicate that this 4th-order polynomial regression model captures the general trend of the original data well, without significant over-fitting.

The last model is a linear regression with 4th-order polynomial features, plus an L2 regularization. The α for the L2 regularization is 1.0. The R^2 on the training set is 0.843. The scatter plot for predicted data (y_pred) vesus the real data (y_test) is:

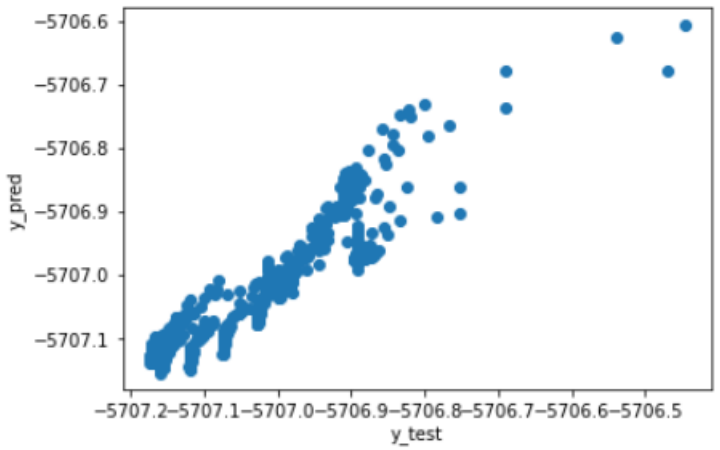

Figure 6: Polynomial regression (4th-order), $R^2 = 0.838$.

The R^2 on the test set is 0.838, similar with that on the training set. The regularization decreases

the prediction accuracy comparing to the second model.

4 Final model and prediction tests

For the three models I tested, the 4th-order polynomial regression provides best predictions on the test set. Since the difference of R^2 on the training set and the test set is similar, the model is not over-fitting. Therefore, it is not necessary to use the L2 regularization to prevent over-fitting. I will use the polynomial regression model to predict new CO energy data. A simple example for prediction: what is the energy of a CO at $r = 1.14$, $R = 3.0$, and $\theta = 15$? Using the second model, I can predict that its energy is -5707.1116.

5 Key Findings and Insights

Using the fitted model, I can predict CO energy without conducting actual any numerical experiment. A more significant usage of the fitted model is to optimize the geometry of the CO molecule. From the model, we can find at which geometry, the energy of CO on a salt surface is the lowest. Naturally, most CO molecules will tend to stay in that position. This problem can be investigated using scipy.optimize package. I first write the model as a function, then set an initial guess, optimize the parameter r, R, and, θ to minimize the energy.

```
# Optimize the energy function to find the minimal geometries using scipy
from scipy.optimize import minimize
x0 = [1.14, 3.0, 0]
res = minimize(CO_energy, x0, args=(lr_pf, pf, s), method='BFGS', jac='2-point', options={'maxiter': 100, 'disp': True})
print(res)
```

```
Optimization terminated successfully.
         Current function value: -5707.185014
         Iterations: 9
         Function evaluations: 48
         Gradient evaluations: 12
      fun: -5707.185013553603
 hess_inv: array([[ 0.17431727, -0.1046366 ,  0.        ],
       [-0.1046366 , 16.79693466,  0.        ],
       [ 0.        ,  0.        ,  1.        ]])
      jac: array([0., 0., 0.])
  message: 'Optimization terminated successfully.'
     nfev: 48
      nit: 9
     njev: 12
   status: 0
  success: True
        x: array([1.13904307, 3.50084348, 0.        ])
```

Figure 7: Minimize the CO energy to find the optimized geometry.

Here I use the polynomial regression model, optimize the parameters using BFGS algorithm and 2-point finite-difference gradient. The initial guess is $r = 1.14$, $R = 3.0$, and $\theta = 0$. I find one stable geometry at $r = 1.139$, $R = 3.501$, and $\theta = 0$, with energy -5707.185014. This indicates a "C-down" geometry of the CO molecule, which is consistent with my chemistry knowledge.

6 Summary and suggestions for next steps

In this report, I use the regression model to study the CO absorption on salt surface. The 4th-order polynomial regression model successfully capture the relationship between CO geometry and its energy. However, one fallacy I found is that the prediction will fail at geometries that is far away from the sampling area. For example, the current data set only have data between $R = 2.0$ and $R = 5.0$. If I try $r = 1.2$, $R = 20.0$, and $\theta = 30$, the energy I got is -4657.1123, which is impossible since the CO energy should always been around -5707.1. To extend the usability, one solution is to collect more data beyond $R = 2.0$ and $R = 5.0$; the other solution is to integrate my domain knowledge: for R smaller than 2.0, the CO will be too close to the surface, the energy will be very high, while for R larger than 5.0, the energy should be almost unchanged since the interactions are negligible. This mathematically implies an inverse exponential relationship between R and energy.

$$\delta E = Ce^{-\alpha R}$$

Therefore, one possible solution to try in the future is to conduct a logarithmic transformation for R before fitting.