# EE3-23 Introduction to Machine Learning Coursework

Zhongxuan Li
CID: 01050018

## Plagiarism Pledge

I, Zhongxuan Li, pledge that this assignment is completely my own work, and that I did not take, borrow or steal work from any other person, and that I did not allow any other person to use, have, borrow or steal portions of my work. I understand that if I violate this honesty pledge, I am subject to disciplinary action pursuant to the appropriate sections of Imperial College London.

## 1. Introduction

In this coursework, standard machine learning methodologies are applied to predict Beijing PM2.5 density. This report consists of dataset processing, discussion of loss function, comparison between distinct models, parameter tuning and conclusions. All the codes are written in Python, and results are obtained all in once.

## 2. Curating on the Dataset

The question requires prediction of any day's PM2.5 at 8 am based on data available until 8 am of the previous day. According to this specification, the following adjustment on the dataset is performed before the $csv.$ file is converted into numpy matrix.

- Remove all columns related to time, including 'index', 'year', 'month', 'day', and 'hours'. As they are not considered as features that influence PM2.5 on 8 am.
- On each row, append accumulated meteorology data from past 8 hours. In this way, we expand the feature matrix dimension by 8.
- Delete all rows that are not recorded at 8 am. Because we don not aim to predict PM2.5 at other time.
- Create a new column which includes 'PM2.5' of next day at 8 am. Thus today's features correspond to appropriate target in tomorrow.

Hence, we obtain a data matrix of dimension $1651 \times 73$. After extracting the last column out as target matrix, the remaining matrix forms feature matrix.

## 3. Set Splitting and Standardisation

We split either feature and target data matrix into two sets with ratio 0.15, obtaining four distinct sets called $\mathbf{X}_{train}, \mathbf{Y}_{train}, \mathbf{X}_{test}, \mathbf{Y}_{test}$. They are shuffled right after. Standardise every column (feature) in $\mathbf{X}_{train}$ with formula,

$$x'_{train} = \frac{x_{train} - E(x_{train})}{\sigma_{x_{train}}} \tag{1}$$

so as every column in $\mathbf{X}_{test}$ with formula,

$$x'_{test} = \frac{x_{test} - E(x_{train})}{\sigma_{x_{train}}} \tag{2}$$

where $E(x_{train})$ is mean of a single column vector in $\mathbf{X}_{train}$ and $\sigma_{x_{train}}$ is standard deviation of a single column vector in $\mathbf{X}_{train}$.

## 4. Loss Function

In linear regression model, we exam the deviation from correct targets. Therefore an error above the target causes the same loss as the same magnitude of error below the target. If the target is $y$, then a quadratic loss function is

$$L(y, f(x, w)) = (y - f(x, w))^2 \tag{3}$$

However in soft-margin SVM we used later on, the features are first mapped onto a m-dimensional feature space using kernel trick, and then a linear model is constructed in this feature space. we ignores errors that are within $\epsilon$ distance of the observed value by treating them as equal to zero and penalty outliers excluded from the $\epsilon$ tube[3]. The loss is measured based on the distance between observed value y and the boundary. This is formally described by the linear $\epsilon$-insensitive loss function.

$$L_\epsilon(y, f(x, w)) = \begin{cases} 0 & \text{if } |y - f(x, w)| \leq \epsilon \\ |y - f(x, w)| - \epsilon & otherwise \end{cases} \tag{4}$$

## 5. Base line: Linear Regression

Use $\mathbf{X}_{train}$ and $\mathbf{Y}_{\mathbf{train}}$ to train the linear regression model and test the model on the test set. The training error is 5453 and testing error is 4559. This is our base line

and no advanced algorithm should perform under this baseline. Linear regression has drawbacks as the model tries to fit every point in the feature space including noise therefore overfitting happens at unpredictable risk.

## 6. Ridge and Lasso Regression

Ridge regression prevent the model from overfitting by using L2 regularisation, introducing a model complexity penalty term, $\alpha||\omega||^2$. Use 5-fold cross-validation to select optimal penalty factor out of the range $\alpha \subset [10^{-4} - 10^4]$. Model performance is accessed by cross-validation error, which is defined as

$$E(L(y, f(x, w, \lambda))/N) \tag{5}$$

where L is quadratic loss function and $N$ is size of the validation set.

Plot the cross-validation(CV) error and training error over different $\alpha$ values on a log base.

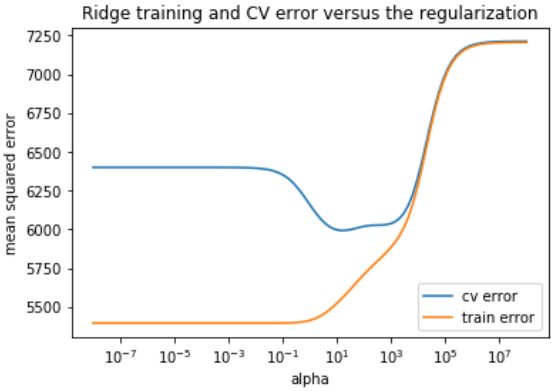


Figure 1. Ridge training and CV error as a function of $\alpha$

In figure 1, training error rises monotonically with increasing regularisation while CV error has a convex shape. This is because when $\alpha$ is small then $||w||^2$ values are large and we tend to overfit the data set. Hence the training error will be low because a complex model tries to fit every point in the training set perfectly. Nonetheless, the cross validation error will be high simply because overfitting. However when $\alpha$ is large then the values of $||\omega||^2$ become negligible almost leading to a polynomial degree of 1. These will underfit the data and result in a high training error and a cross validation error.

The chosen value of $\alpha$ should be such that the cross validation error is the lowest. A global minima occurs at $\alpha = 16, MSE = 6091$. We select this $\alpha$ as our optima parameter, train the model on the whole training set, getting overall training error 5584 and testing error 4541.

Lasso regression differs from ridge by using L1 norm, which allows sparse solution that can be used for feature selection. From figure 2, Lasso has similar error plot as Ridge regression. $\alpha$ value 1 has lowest error of 6017, final training error is 5885 and testing error is 4528.

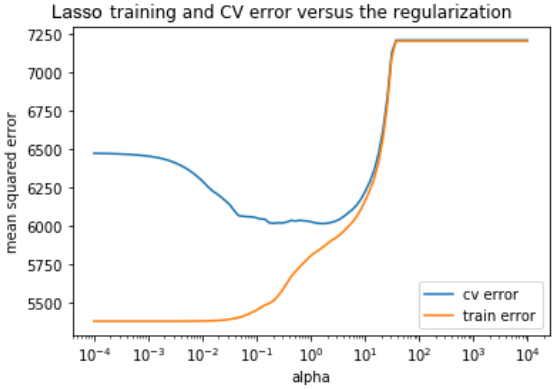


Figure 2. Lasso training and CV error as a function of $\alpha$

It is worthwhile to notice that Ridge and Lasso coefficients converges at different rate. Under Ridge regression, $\omega$ converges to zero as an exponential while under Lasso regression, due to sparse solutions, some $\omega$ drops to zero rapidly whereas some retain at a certain level. This fact proved that Lasso has done feature selection successfully.

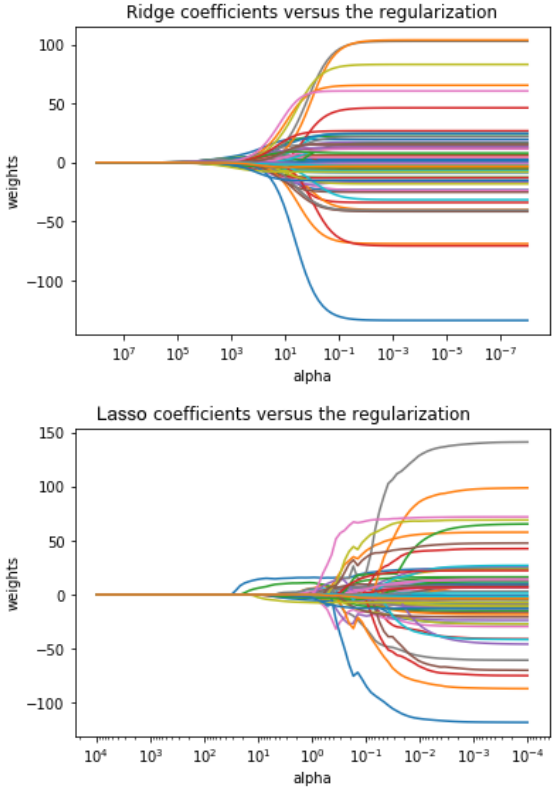


Figure 3. $\omega$ as a function of $\alpha$

An illustrative plot[2] of PM2.5 versus pressure in 2014 April shows that the dataset cannot fit a linear regressor(the

red line). Thus non-linear transformation will be our next choice.

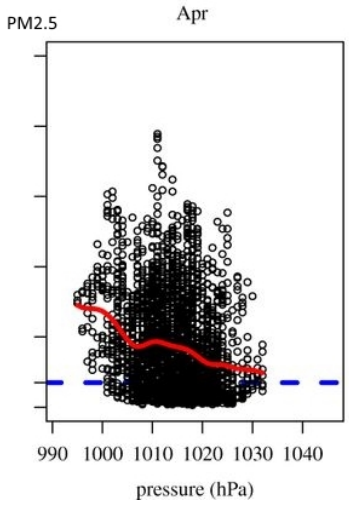


Figure 4. scatter plot PM2.5 versus air pressure

## 7. Advanced Method: Support Vector Machine

SVM regression performs linear regression in the high-dimension feature space using $\epsilon$ insensitive loss (equation4). Meanwhile soft-margin SVM allows some violations by introducing slack variables $\xi$. Hence we arrive at minimizing

$$||\omega||^2 + C\sum_{i=1}^{N}\xi_i \tag{6}$$

subject to

$$\begin{cases} |y_i - f(x_i, w)| - \epsilon \leq \xi_i \\ \xi_i \geq 0 \end{cases}$$

Parameter $C$ determines the trade off between the model complexity and the degree to which deviations larger than the $\epsilon$ tube are tolerated in equation 6. For example, if $C$ is too large (infinity), then the objective is to minimize training error only by regulating $\xi$ as small as possible, without regard to model complexity part $||w||^2$, and overfitting may occurs with high variance. If $C$ is small, then the model has a very relaxed slack variable leading to underfitting or high bias in terms of bias-variance analysis.

Use cross-validation to tune optimal $C$ and $\epsilon$ with RBF (gaussian) kernel. Then plot both errors as heatmap.

As shown in the figure 5, $C$ value 61, $\epsilon$ value 37 has lowest CV error of 6157, final training error is 5196 and testing error is 4560. It is obvious that we have obtained a wide $\epsilon$ tube implies that there is considerable high level of noise in the dataset. This model may not be able to find an appropriate regression polynomial, instead it tries to tolerance many points lies with in the $\epsilon$ tube and regulate the slack variable

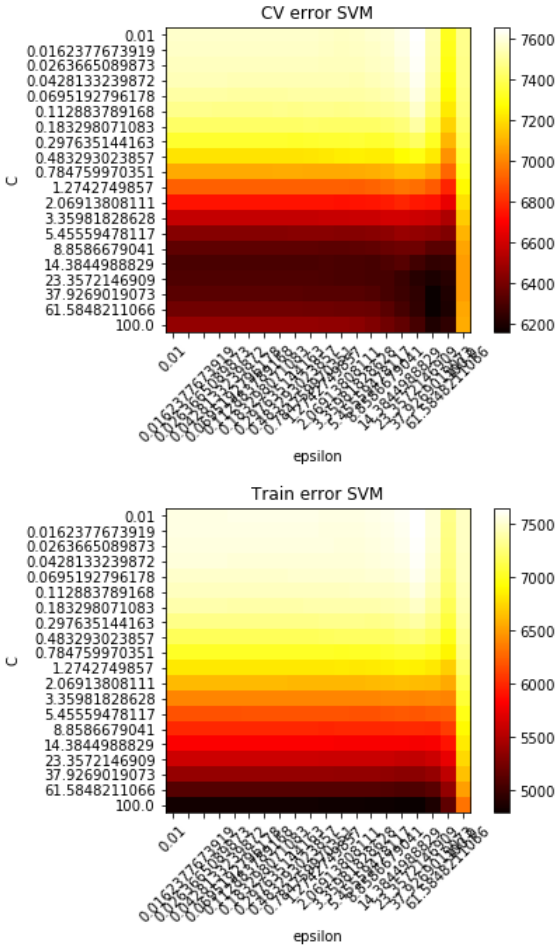


Figure 5. Lasso training and CV error as a function of $\alpha$

$\xi$ by having a moderately large $C$. In general, this SVM method does not excel the performance of linear methods.

Another solution has been proposed that instead of searching for optimal $\epsilon$, we find optimal $\gamma$ in the radial basis function.

$$K(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2\sigma^2}\right) = exp(-\gamma\|\mathbf{x} - \mathbf{x}'\|^2) \tag{7}$$

From equation 7, a low $\gamma$ causes high variance ($\sigma^2$), implies we have a wide gaussian distribution. If $\gamma$ is large, then variance is small implying the support vector does not have wide-spread influence, but high bias instead.

Again do cross validation of $C$ and $\gamma$ with $\epsilon$ fixed at 0.01. In figure 6, $C$ value 16, $\gamma$ value 0.01 has lowest error of 6323. Final training error is 5822 and testing error is 4581.

We are at extreme condition that having $\gamma$ goes to 0. Because we are getting an extreme wide gaussian distribution with variance goes to infinity. The region of influence of any selected support vector would include the whole training set even if the the Euclidean distance between them is

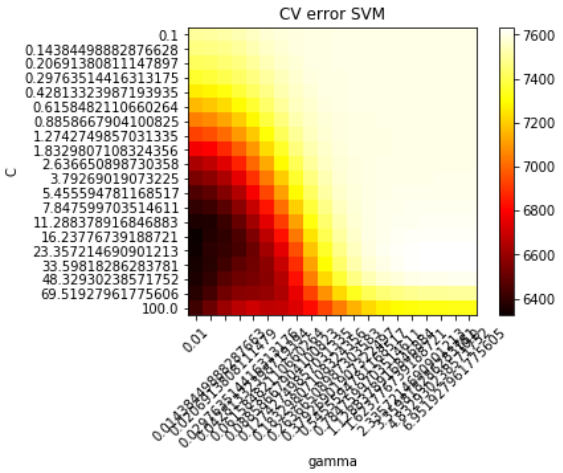


Figure 6. Lasso training and CV error as a function of $\alpha$

large. Under this scenario our result is more likely a global average, and we are losing the privilege of SVM. [1]

## 8. Proposed Solution

| Method | CV error | Training error | Test error |
|---|---|---|---|
| Linear Regression | NA | 5453 | 4559 |
| Ridge Regression | 6091 | 5584 | 4541 |
| Lasso Regression | 6017 | 5885 | 4528 |
| SVM | 6157 | 5196 | 4560 |

Table 1. Comparison between different methods.

In table 1 all four algorithms has close performance. Intuitively, as all linear methods try to fit a line to data by minimising the loss function. Despite that we can deploy a non-linear RBF kernel in SVM. In this case, we end up approximating the data points by an infinite order polynomial. Unfortunately, SVM does not give satisfactory result in the end. Albeit it has been proved in this report that SVM has performance at least no worse than linear method.

A proposed solution would be Lasso regression. On one hand, despite the fact that Lasso's CV and test error is gently lower, the feature dataset has relative large dimension, and indeed some of them are uncorrelated to the target(proved in figure 3). Therefore, feature selection becomes highly necessary. On another hand, model complexity is an important factor taken into account. In the future, we may accumulate a large amount of data and a faster model would be optimal provided all four algorithms have similar generalisation performance.

## 9. Conclusion

The meteorology features provided may not be strongly correlated with PM2.5 density. A scientific research paper about Beijing air pollution [2] reveals that the winter heating (from November to March) has contributed a more than 50 increase (on average) in PM2.5 severity whereas during the APEC meeting, the Chinese government had ordered a temporary closure of factories in Beijing and part of its neighbour provinces. These factors all have significant impact on PM2.5 but have not been added to features in the dataset. Ultimately, PM2.5 arises from human activities, not meteorology conditions. I suspect that adding new features such as "forum in progress" or "winter heating" will help improve the model performance.

Overall, our models give acceptable daily PM2.5 prediction with an average deviation of 67($\sqrt[2]{4530}$). The methodologies applied in this coursework are solely machine learning algorithms. In the future, statistical model such as autoregression and moving average could be introduced in order to enhance accuracy in time series forecasting.

## References

[1] O. Chapelle and V. Vapnik. Model selection for support vector machines. *In Advances in Neural Information Processing Systems*, 1999.

[2] H. H. Liang, X. and S. X. Chen. Assessing beijing's pm2.5 pollution: severity, weather impact, apec and winter heating. *the Royal Society A, 471, 20150257*, 2015.

[3] A. Smola and B. Schlkopf. A tutorial on support vector regression. *NeuroCOLT Technical Report NC-TR-98-030 Royal Holloway College, University of London*, 1998.