Gradient-descent linear regression: Implementation and experimentation with wine-quality data

Rabindra Nepal
Deparment of Physics, University of Nebraska-Lincoln
Lincoln, Nebraska 68588
Email: rnepal2@unl.edu

***Abstract*—We custom implement gradient descent linear regression model and use it to build a wine quality prediction model using winequality-red dataset. No machine learning library is used in this work. We reduce the mean-squared-eror to 0.38 using the simple linear regression model to predict the wine-quality.**

***Index Terms*—Gradient descent, linear regression, wine quality**

I. Introduction

Linear regression is a simple yet a powerful linear machine learning model. It is linear on the weight or coefficient, but contrary to its name it can also fit non-linear or polynomial functions using appropriate basis functions. In this project, we custom implement linear regression model using gradient descent algorithm with mean square error as the cost function,

$$J(\theta) = \frac{1}{2m}\sum_{i}^{m}(y_i - \theta^T X_i)^2 + \beta_i \tag{1}$$

where β_i is the penalty introduced to avoid the overfitting and m is the total number of sample points in the feature matrix X or correspondingly in target vector y. It is given by $\beta_1 = \frac{\lambda}{2m}||\theta||$ for l_1-regularization and $\beta_1 = \frac{\lambda}{2m}||\theta||^2$ in case of l_2-regularization. Then, weight θ is iteratively updated according to gradient descent algorithm as,

$$\theta := \theta - \frac{\eta}{m}X^T \cdot (X \cdot \theta - y) - \Lambda \tag{2}$$

where λ is given by $\frac{\eta\lambda}{m}\text{sgn}(\theta)$, $\frac{\eta\lambda}{m}\theta$ for l_1, l_2 regularizations and 0 in the absence of regularization. The parameters η and λ are called learning rate and constant of regularization respectively. The features matrix X can be augmented with polynomial function of degree n to fit the non-linear behavior of dataset, which we have implemented in our model.

Using our custom implemented version of linear regression model, we build a wine quality prediction model using winequality-red dataset. First, we carry out exploratory data analysis to get the insight of dataset for effective features selection. Using the simple linear regression model, we predict the wine quality of testing dataset and do the extensive analysis of the model performance.

II. Dataset

The dataset used is taken from UCI Machine Learning Repository for 1599 Portuguese red wine samples [1]. This dataset was originally used to study the human wine taste preferences using machine learning algorithms such as support vector machine, multiple regression and neural network [2]. The dataset has twelve different features including the overall quality measure in a scale of 3-8. The dataset is complete and there are not any missing values. All the feature values are floats except the quality feature which is an integer. We use 'quality' feature as the target in this study and the linear regression model is designed to predict the value of it. The basic statistical distribution of the features in the dataset is present in Table I.

III. Model

Here, we work with a simple iterative linear regression model using gradient descent algorithm. The extensive serach of the best hyperparameters is carried our for different values of learning rate, regularizations methods and its parameter λ. We discuss the best model parameters and performance followed by the features selection below.

A. Features selection and scaling

The dataset under our study originally has 12 features including the target variable in our study i.e. wine quality measure. We analyzed the collinearity of features as well as their individual correlation with the target variable. The experimentation suggests that there is not any significant difference in performance of the model in dropping few of the less important features to using all of them. However, based on the relativey smaller correlation of feature 'free sulfur dioxide' with the target variable as well as stronger collinearity with other feature 'total sulfur dioxide', we decided with dropping the feature 'free sulfur dioxide'.

If the features values range from a small number to very large, the importance of the features might not be correctly decided in the prediction model, see the basic statistical distribution of the data features in Table. [1]. Therefore, usually the data features are scaled using standard normal distribution, i.e. $\tilde{x} \rightarrow \frac{x-\mu}{\sigma}$, where μ and σ are the mean and standard deviation of the values x_i of a feature. We also observe the improvements in the performance of the model after features scaling. Moreover, the scaling is critical given that we are using iterative gradient descent algorithm with regularized mean-squared error as the cost function. Due to the presence of larger numbers in the some of the features in

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free SO_2	total SO_2	density	pH	sulphates	alcohol	quality
mean	8.31	0.52	0.27	2.53	0.087	15.87	46.46	0.99	3.31	0.65	10.42	5.64
std	1.74	0.18	0.19	1.41	0.05	10.46	32.89	0.002	0.15	0.17	1.06	0.81
min	4.60	0.12	0.0	0.90	0.012	1.0	6.0	0.99	2.74	0.33	8.40	3.0
max	15.9	1.58	1.0	15.5	0.61	72.0	289.0	1.0	4.01	2.0	14.9	8.0

TABLE I
STATISTICAL DISTRIBUTION OF DATASET

the dataset, the iterative weight update can quickly overshoot leading to the divergence very quickly even for a relatively smaller learning rate. Therefore, the features scaling is very important and we scale features in evey model in this study. In case of polynomial features augmentation, it becomes even more critical because of the possibility of appearing very large numbers in the dataset and we take this into account while building the prediction model with polynomial features augmentation.

B. Results and Discussions

We train a linear regression model with the training dataset prepared after features selection and scaling described above. The appropriate hyperparameters are selected after the extensive 5-fold cross-validation study for a broad spectrum of parameters such learning rate, regularization methods and parameter lambda. For the model with unagumented features of polynomial degree-1, we achieve the mean-squared error of 0.38 on the test data for the set of hyperparameters: $\eta = 0.01, \lambda = 10.0$ with l_1-regularization after 1000 iterations. The learning curve of the model is shown in Fig. (1). The figure suggests that the model is underfitting the data as both training and validation errors converge after sufficient training dataset yet yielding poor performance.

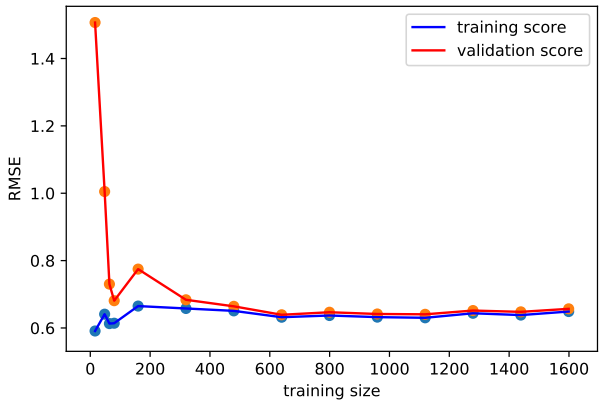

Fig. 1. (Color online) Learning curve with polynomial features of degree-1: the root-mean-squared error corresponding to training (blue line) and validation dataset (red line) is plotted against the size of the dataset used.

In addition, we further explore by augmenting the features into polynomial degree-3, see the learning curve in Fig. (2). We note that our gradient descent algorithm doesn't use the adaptive learning rate, therefore, we need to manually decrease the learning rate in the case we use features augmentation of higher polynomial degree in order to avoid overshooting. This requires the increased number of iterative steps in the weights updates to acquire a good performance. When the complexity of the model is enough, $d = 3$ in this case, the performance of the model is better while comparing with that for the validation data as the training dataset is increased (eventhough the effect is not enhanced enough in this case obtained though our custom implemented model). This signals that the model is overfitting after the sufficient training with enough dataset.

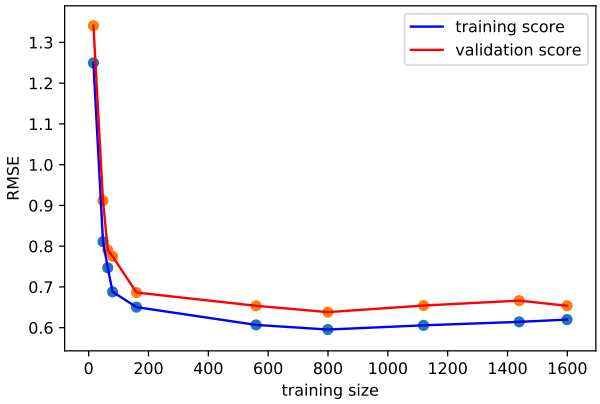

Fig. 2. (Color online) Learning curve with polynomial features of degree-3: the root-mean-squared error corresponding to training (blue line) and validation dataset (red line) is plotted against the size of the dataset used.

Since the model using features of polynomial degree-1 underfits the data, we tested the model by augmenting the features with higher degree polynomial, see Fig. (3). With the increase in the polynomial degree, the performance of the model initially increases upto degree 3 on both the training and validation dataset. However, the separation of performance scores in the training and validation dataset keeps on increasing with the polynomial degree. This indicates the model is overfitting the dataset with the increase in polynomial features degree. Clearly, as seen in the figure, the overfitting becomes very significant after the polynomial degree 4.

IV. CONCLUSION

We successfully implemented the linear regression model based on gradient descent algorithm. We then tested the model in order to build a prediction model using wine quality dataset to predict overall quality of the wine. Using the dataset, we

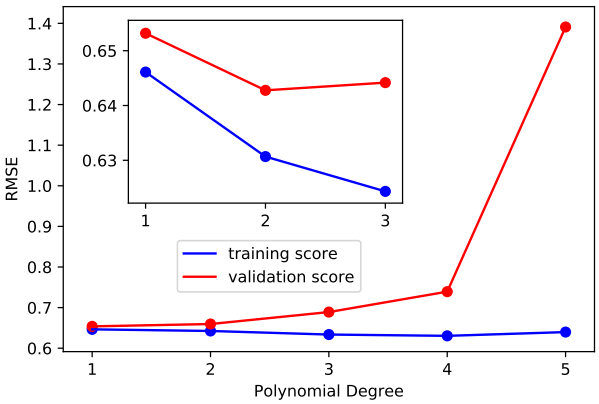

Fig. 3. (Color online) Polynomial model complexity: the root-mean-squared error of the model for training (blue line) and validation (red line) dataset for the features with different polynomial degrees. The inset is the same figure with smaller polynomial degree range presented for the sake of clear illustration.

studied the different scenarios of underfitting and overfitting of the model.

REFERENCES

[1] Cortez, P. (2009). Wine Quality Dataset, https://archive.ics.uci.edu/ml/datasets/wine+quality. UCI Machine Learning Repository.

[2] Paulo Cortez, A. C. (2009). Modeling wine preferences by data mining from physicochemical properties. Decision Support Systems, 47, 547–533.