Project-I by Group Barcelona

Eric Francis Bezzam
EPFL
eric.bezzam@epfl.ch

Manish Thani Awtaney
EPFL
manish.thaniawtaney@epfl.ch

Abstract

This report summarizes our observations and results for Project-I of the EPFL course entitled "Pattern Recognition and Machine Learning." The work was done over the course of one month from October 2, 2015 to November 2, 2015. We were given two datasets: one for regression and the other for classification. To the regression dataset, we applied several linear regression algorithms: least-squares, gradient descent, and ridge regression. Ultimately, we opted for a multiple model approach as it significantly improved the prediction results. A data example is first classified to one of three groups and depending on this decision, the corresponding ridge regression model is applied. To the classification dataset, we applied logistic regression and penalized logistic regression. The latter is chosen as it provides regularization. For both datasets, we tried incorporating a polynomial basis. It provided significant improvements for the regression dataset but not for the classification dataset.

1 Regression

1.1 Data Description

In the regression dataset, we are given $N_{tr} = 2800$ training examples. Each training example consists of an output variable y_n and a vector of input variables $\mathbf{x_n}$. Each input vector $\mathbf{x}_n$ is of dimensionality $D = 72$. Of these 72 variables: 61 are continuous, real-valued variables; 4 are binary variables; 2 have 3 categories; and 5 have 4 categories.

We also have $N_{te} = 1200$ test examples where we do not know the output value y_n.

1.2 Objective

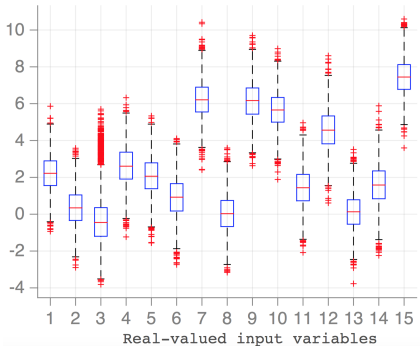
Our objective is to produce predictions for the test examples and to approximate the test-error using:

$$RMSE = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(y_n - \hat{y}_n)^2}, \tag{1}$$

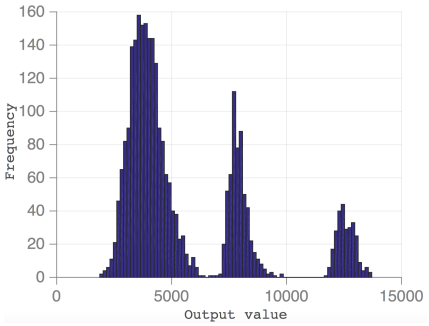
where y_n is the true output value and $\hat{y}_n$ is the corresponding prediction. We wish to find the model that will yield the smallest prediction error.

1.3 Exploratory Data Analysis

We performed basic exploratory data analysis in order to observe unique characteristics of the data. We used boxplots to display the distributions of the continuous input variables (Figure 1(a)) and a histogram to display the distribution of output values (Figure 1(b)).



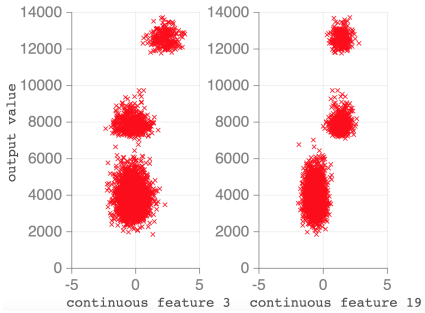
(a) Boxplots of first 15 continuous input variables of $\mathbf{X}$.



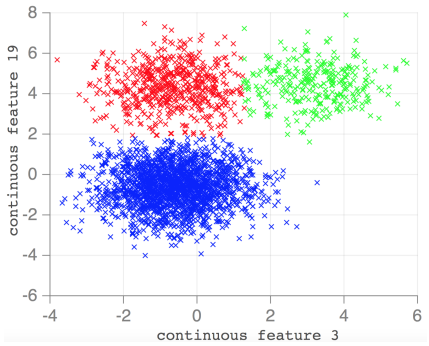
(b) Histogram of output values $\mathbf{y}$.

Figure 1: Visualizing input and output data

From Figure 1(b), we can observe three Gaussian-like shapes in the histogram of the output. However, when plotting each of the input variables against the output, none of them have three identifiable groups along the input dimension. The 3^{rd} and 19^{th} input variables, though, have two distinguishable groups (Figure 2(a)). By combining these two features into a single feature vector and using K-means (Figure 2(b)), we can divide the input into three groups corresponding to the three Gaussians in the output distribution.



(a) Plotting 3^{rd} and 19^{th} continuous input features against output.



(b) Combining 3^{rd} and 19^{th} continuous input features to discriminate three groups in output.

Figure 2: Exploiting 3^{rd} and 19^{th} continuous input features

Another important property of the input that we analyzed is its rank. The input matrix $\mathbf{X}$ is rank-deficient with a rank of 63.[1] Since the matrix is ill-conditioned, least-squares is not suitable for linear regression. Therefore, we will have to "lift" the eigenvalues with ridge regression or approach the optimal solution gradually using gradient descent.

1.4 Feature processing

From Figure 1(a), we can see that the continuous input variables are not centered. Therefore, we need to normalize them. For the categorical input variables, we must use dummy coding. We represent a categorical variable as multiple binary variables through dummy coding. Generally, we can represent a variable with K categories with $K-1$ binary variables. Therefore, the dimensionality of the input matrix $\mathbf{X}$ will increase to 84: 61 (continuous input variables) + 4 (binary variables) + 4

[1]The rank was found by using the MATLAB command `rref` in order to determine the number of linearly independent columns using the reduced row echelon form.

(dummy-coded variables resulting from 2 variables with 3 categories) + 15 (dummy-coded variables resulting from 5 variables with 4 categories).

1.5 Applied Methods and Analysis

One of our objectives is to approximate the test-error using Equation1. However, we cannot use the test samples to estimate this error as we do not have access to their true output values, y_n. Therefore, we *simulate* test data by dividing our training examples into a training set and a test set, using a $80-20$ split. The training set inputs and outputs are referred to as $\mathbf{X_{tr}}$ and $\mathbf{y_{tr}}$ respectively. The test set inputs and outputs are referred to as $\mathbf{X_{te}}$ and $\mathbf{y_{te}}$ respectively.

As a baseline for the test-error, we computed the error when using the mean of the training set outputs as the predicted value. This model yielded an $RMSE$ of 2785.2.

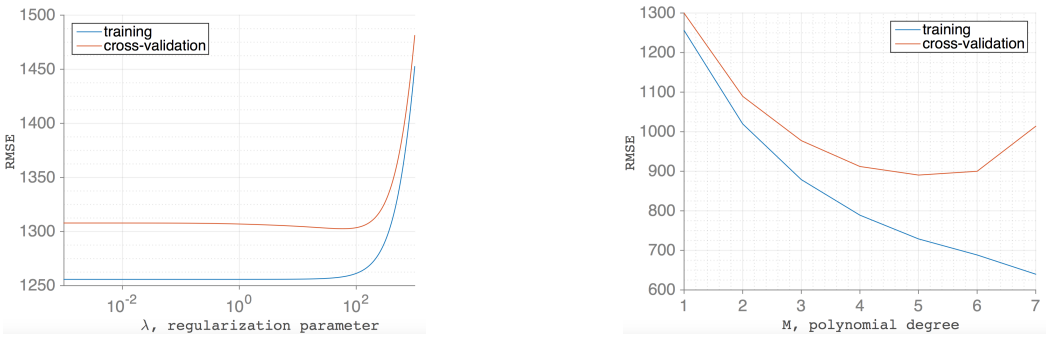
1.5.1 Least-Squares

When running least-squares on the training set to calculate the optimal coefficients for linear regression, we obtain a test $RMSE$ of 1307.2. However, we get the following warning in MATLAB: *"Matrix is close to singular or badly scaled. Results may be inaccurate."* This is due to the fact that the input matrix is ill-conditioned as previously stated in Section 1.3. This means that least-squares is not a suitable method.

1.5.2 Gradient Descent and Ridge Regression (RR)

Since the input matrix is ill-conditioned, we used gradient descent to calculate the coefficients for linear regression in a more stable manner. With a step size of 0.25, we approximated the test-error to be: $RMSE = 1307.2$.

An alternative approach to linear regression that we investigated is ridge regression. This method overcomes the problem of an ill-conditioned input matrix by "lifting" its eigenvalues. Furthermore, ridge regression has an advantage over gradient descent in that it provides *regularization*. We used 10-fold cross-validation on $\mathbf{X_{tr}}$ and $\mathbf{y_{tr}}$ in order to optimize the regularization parameter, λ. We calculated the training and cross-validation error for 500 points logarithmically spaced between 10^{-3} to 10^{3} (Figure 3(a)). The regularization parameter that yielded the minimum cross-validation error was 56.17. Using the coefficients calculated from ridge regression with λ set to 56.17, we approximated the test-error to be: $RMSE = 1303.1$.



(a) Varying regularization parameter for RR.

(b) Varying degree for polynomial RR.

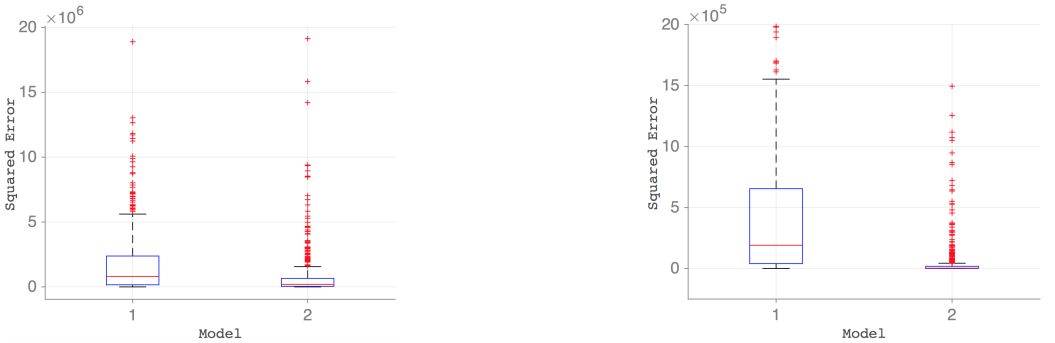
Figure 3: Training and cross-validation curves for RR (a) and polynomial RR (b)

1.5.3 Ridge Regression + Feature Transformation

Ridge regression implemented in the previous section still exhibits high bias. This can be seen from Figure 3(a). Even for a low value of λ (corresponding to a more complex model), both the training and test error are too high and there is a small gap between the two curves. According to Andrew

Ng (3), this is typical for high bias and we can fix this by trying a larger set of features. We can artificially create a larger set of features through *feature transformation*.

We create a polynomial basis out of each feature by raising each one to degrees of 1 to M. With 10-fold cross validation, we performed a grid-search for the degrees M from 1 to 7 and for 100 regularization values λ logarithmically-spaced from 10^{-3} to 10^{3}. The degree and regularization parameter that yielded the minimum cross-validation error were $M = 5$ and $\lambda = 7.56$. With these parameters, we approximated the test-error to be: $RMSE = 884.34$.



(a) Ridge regression (1) and polynomial ridge regression (2).

(b) Single model approach (1) and multiple model approach (2).

Figure 4: Comparing models using squared error distribution: $(y_{true} - y_{pred})^2$

Figure 3(b) shows how there is high bias for ridge regression with $M = 1$. For large degrees, the model starts over-fitting the training data; the training error continues to decrease while the cross-validation error increases. The degree that yields the minimum cross-validation error is $M = 5$. Figure 4(a) compares the distributions of the squared test-error of the linear ridge regression and polynomial ridge regression models. We can conclude that polynomial ridge regression performs better as the error variance is lower (narrower box) and the approximated test-error of polynomial ridge regression is significantly lower (32% decrease).

1.5.4 Multiple Polynomial Ridge Regression Models

Using the two continuous input features from Figure 2(b), we split the training data into three groups and train a polynomial ridge regression model for each group. The K-means algorithm (with $K = 3$) will be used to find the center of each cluster. The prediction for a given test data will be calculated by first finding which center the test data is closest to with respect to the 3^{rd} and 19^{th} continuous input feature. Then the corresponding polynomial ridge regression model will be applied. With this multiple model, we approximated the test-error to be: $RMSE = 529.63$.

Figure 4(b) compares the distributions of the squared test-error of the single model approach and the multiple model approach.[2] We can conclude that the multiple model approach performs better as the error variance is lower (much narrower box) and the approximated test-error is significantly lower (40% decrease from the single model approach).

2 Classification

2.1 Data Description

In the classification dataset, we are given $N_{tr} = 1500$ training examples. Each training example consists of a binary output variable y_n that can have a value of 1 or -1 and a vector of input variables $\mathbf{x_n}$. Each input vector is of dimensionality $D = 39$. Of these 39 variables: 34 are continuous, real-valued variables; 3 are binary variables; and 2 variables have 3 categories.

[2]The range of squared error is limited so that the boxplot for model 2 can be better visualized.

We also have $N_{te} = 1500$ test examples where we do not know the output value y_n.

2.2 Objective

Our objective is to produce predictions for the test examples and to approximate the test-error using RMSE (1 except that instead of $\hat{y}_n$ we use the prediction probability $\hat{p}_n$ that an output belongs to class $y_n = 1$), $0-1loss$, and the negative log-likelihood. The $0-1loss$ is defined as:

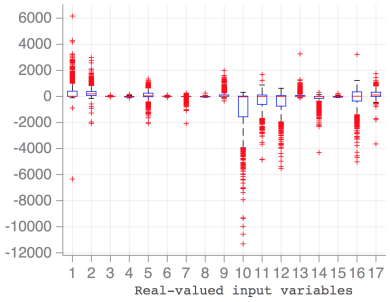
$$0-1loss = \frac{1}{N}\sum_{n=1}^{N}\mathbb{I}(y_n \neq \hat{y}_n), \tag{2}$$

where y_n is the true output value and $\hat{y}_n$ is the corresponding prediction. The negative log-likelihood is defined as:

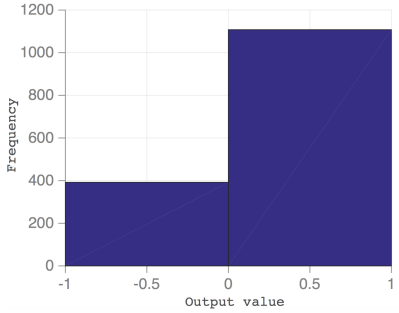
$$logLoss = -\frac{1}{N}\sum_{n=1}^{N}[y_n \log \hat{p}_n + (1-y_n)\log(1-\hat{p}_n)]. \tag{3}$$

2.3 Exploratory Data Analysis

Basic exploratory data analysis was done in order to better understand the classification dataset. We used boxplots to display the distributions of the continuous input variables (Figure 5(a)) and a histogram to display the distribution of output values (Figure 5(b)).



(a) Boxplots of first 17 continuous input variables of $\mathbf{X}$.



(b) Histogram of output values $\mathbf{y}$.

Figure 5: Visualizing input and output data

2.4 Feature Processing

From Figure 5(a), we can see that the continuous input variables have very different distributions. For example, the 10^{th} continuous feature has a very large variance while the 3^{rd} continuous feature has a very small variance. Therefore, we need to normalize the continuous input features. We did not use the binary and categorical variables as they yielded a singular matrix when applying the Newton method for logistic regression. Finally, the output values corresponding to -1 were changed to 0 so that our implementation of logistic regression could be applied.

2.5 Applied Methods and Analysis

Similar to our approach for the regression dataset, we use an $80-20$ split to divide the training examples into a training set ($\mathbf{X_{tr}}$ and $\mathbf{y_{tr}}$) of 1200 data samples and a test set ($\mathbf{X_{te}}$ and $\mathbf{y_{te}}$) of 300 data samples.

As a baseline model, we took the most common output in $\mathbf{y_{tr}}$ as the prediction for every $\mathbf{y_{te}}$. The results of this simple model can be seen in Table 1. The resulting error is seemingly low because the output values are not uniformly distributed. A significant majority of them belong to the class $y_n = 1$ as we can see from Figure 5(b).

2.5.1 Logistic Regression (LR) and Penalized Logistic Regression (PLR)

When running logistic regression with a step size of $\alpha = 0.4$, we obtained the errors as seen in Table 1. For penalized logistic regression, we used K-fold cross-validation with $K = 5$ to find the optimal value for the regularization parameter λ. The value $\lambda = 0.001$ yielded the minimum average cross-validation error so this value was chosen to train the model. The results for penalized logistic regression with $\alpha = 0.4$ and $\lambda = 0.001$ can be seen in Table 1. Figure 6(a) shows the training and cross-validation error curves when varying the regularization parameter.

(a) Varying regularization parameter for PLR.

Figure 6: Visualizing input and output data

We tried applying feature transformations for the input variables such as a polynomial basis. For $M = 2$, the error actually slightly increased; and for degrees above $M = 2$, the input matrix became singular so Newton's method could not be applied.

Model	RMSE	$0-1\text{loss}$	Negative Log-Likelihood
Baseline	0.4830	0.2333	–
LR	0.3747	0.1967	0.4463
PLR	0.3747	0.1967	0.4463
PLR (M=2)	0.3670	0.2000	0.4541

Table 1: Errors for classification

3 Summary

In conclusion, we analyzed the regression dataset and applied several models to it. We found that a multiple model approach provides a reasonable fit to the data. For this model, we estimate the $RMSE$ for the test data to be 529.63. To the classification dataset, we applied logistic regression. We opted for penalized logistic regression as it provides regularization. A polynomial basis was investigated for the classification dataset but it did not yield positive results. Therefore, we stuck with penalized regression using the normalized continuous input variables and omitting the categorical one. With this model, we estimate the $0-1loss$ to be 0.1967.

Acknowledgments

We would like to thank Emti and the teacher assistants for providing useful advice. The office hours and lab sessions were very helpful.

References

[1] Lecture notes by Emtiyaz Khan

[2] https://en.wikiversity.org/wiki/Dummy_variable_(statistics)

[3] Andrew Ng ML advice slides