Project-I by Group Rome

Viviana Petrescu
EPFL
vpetresc@epfl.ch

Gocken Cimen
EPFL
gcimen@epfl.ch

Abstract

We present our on two tasks, classification and regression on data that is not known. We process the data, investigate baseline methods and present our results here. For the regression task, out best model was ¡¿ and for classification was ¡¿.

1 Regression

1.1 Data Description

The training data $Xtrain$ contains 1400 observations, each 43 dimensional. One training sample has 36 real valued features and 7 categorical features. Our task is to predict the values for unseen test data, consisting in 600 samples. We measure the accuracy of our estimation using $RMSE$.

1.2 Data visualization and cleaning

Initially, our data was not centered, as seen in $Fig.$ 1a. We changed the categorical features into dummy variables, leading to a new vector of size 56. $Xtrain$ was then normalized to have 0 mean and standard deviation 1. We applied the same operations to the test data $Xtest$ on which we will report the results.

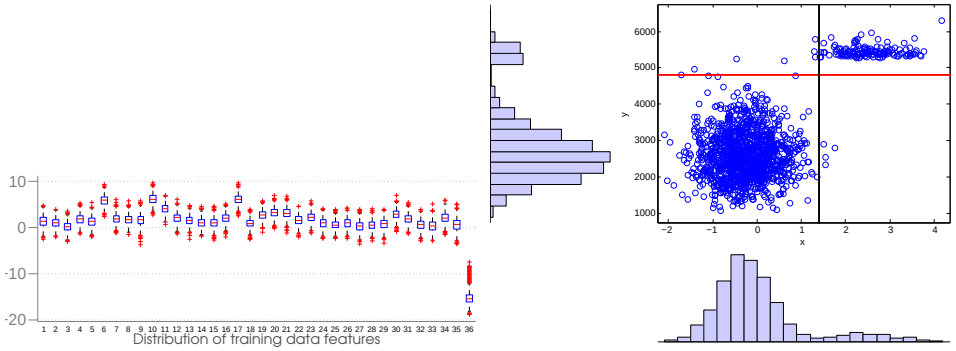
We plotted the correlation of every feature with respect to the output $ytrain$ and the scatter plots did not look random. We concluded the features explain the output, but we could not tell if one of them is insignificant.

The predicted values were real values in $[1000, 7000]$ and seemed to be grouped into two blobs. We believed initially the smaller blob represented outliers, but since it contained almost 10% of the data we decided not to ignore it.

After looking at every feature individually, we noticed that feature 36 offers a clear separation of the two blobs (see $Fig.$ 1b). We therefore chose to fit two models, one in which feature 36 has values > 1.4 (after normalization) and one in which it is smaller, corresponding to smaller predicted values.

We visually observed some linear correlations between certain features such as feature 2 and 24, 13 and 16, 17 and 20, but we decided to keep them since we did not have time to experiment with their removal or to test their signifcance. This is corroborated by the fact that $Xtrain$ is rank deficient, it has 57 columns after the use of dummy variables but rank 50.

If the input is normally distributed with mean 0 and std 1, then 99.99% of the samples appear between the values -3.891 and 3.891. We therefore remove any points that are outside this interval, considering them outliers.



(a) Mean and standard deviation for the first 36 real valued variables of Xtrain. The input is not normalised and feature 36 is the only negative one.

(b) Feature 36 versus output values. X = 1.4 (black line) and y = 4900 (red line) provide a good separation of the two blobs.

Figure 1: Data visualization.

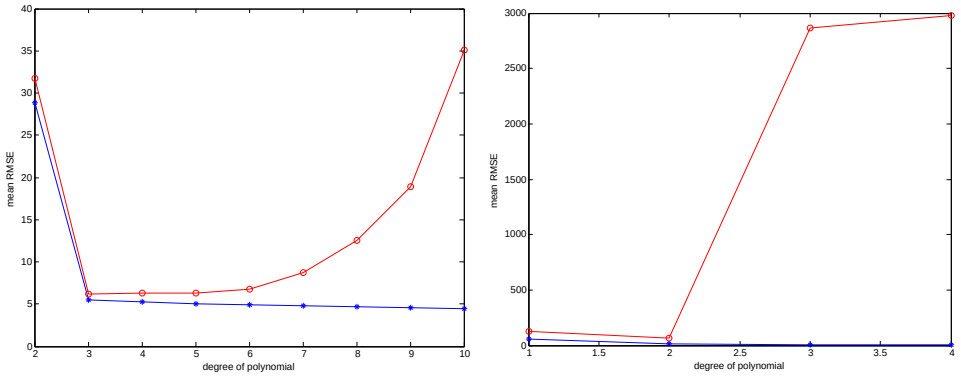
1.3 Ridge regression Baseline methods

1.4 Feature transformations

We tried polynomial and exponential transformations of the features. The exponential transformation did not prove to work well and we focused on polynomial regression. We only transformed the first 35 variables and kept the categorical variables as they are since polynomial transformation of categorical variables does not have an intuitive interpretation.

The degree of the polynomial was varied from 1 to 10 for both blob-models. In Fig2a,2b we plotted the mean training and validation $RMSE$ for the degree varying only between 2 - 10 and 2 - 6 respectively for readability (for the other values the RMSE was too big).

Using 5-fold cross validation with $\lambda = 1e-5, \alpha = 0.1$ we notice that a polynomial of degree 3 is a good fit for the first model. Increasing it further leads to overfitting, since the training errors becomes very small and the validation error increases. For the second model, using $\lambda = 0.001, \alpha = 0.1$ we obtain a best fit for a polynomial of degree 2.



(a) First Blob. Mean RMSE for training set (blue curve) and testing (red curve) versus polynomial degree. The errors were computed using 5-fold cross-validation.

(b) Second Blob. Mean RMSE for training set (blue curve) and testing (red curve) versus polynomial degree. The errors were computed using 5-fold cross-validation.

Figure 2: Selection of polynomial degree

We increased lambda to prevent overfitting, since we have 10 times less samples for the second model. However, we still noticed that our RMSE for the validation set was bigger for the second

blob. This might be due to the fact that we have a very small set for both training and testing of the second model.

Both polynomial regression models significantly outperform the normal ridge regression, which gave a RMSE ¿ 100.

Model1 mean 5.4649(std 0.0579) mean 6.2188(std 0.4801) Model2 mean 11.7039(std 2.0178) mean 53.1940(std 14.5599)

Note also that our final RMSE (reported on the whole training data) is smaller that the sum of the two separated RMSE. write here about best RMSE for both models separate, and for the final one.

Our wrong assumption was that in our training set we had a clear cut of the two models using the y value. The problem is that there are some points with feature 36 ¿ -13 which should actually be part of the other model. Because of this we expect our error on new data to be big. We tried splitting the data into 20 per cent for testing and 80 per cent for validation and as expected out test error got much worse.

Our model is similar with cite book here and suffers from the same problems of the multiple regression problem as stated there. Since we do not enforce continuiting at the X36 feature, our model will generate high errors for samples that have the value of feature36 near the threshold value. This happens because feature 36 does not offer a clear separation of the data. We could have tried to enforce smoothnes at the thresholded value. We believe we have two models that work well separately but perform bad at the border. We removed all samples between -12,-14 and indeed our RMSE got worse, showing that the worst error comes from that part of the model.

2 Classification

2.1 Data description

We have a two-class classification problem. The training data $Xtrain$ contains 1500 observations, each 32 dimensional. One training sample has 31 real valued features and 1 categorical feature, the 14th feature. Our task is to predict the category for unseen test data, consisting in 1500 samples. We measure the accuracy of our estimation using $RMSE$, $0-1loss$ and $log-loss$.

2.2 Data visualization and cleaning

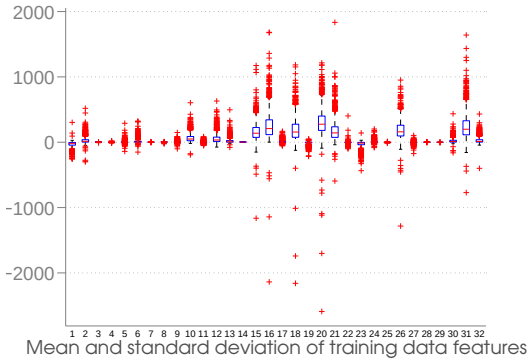
We noticed the samples were not equally distributed among classes, about 35% of the samples were coming from one class and 65% from the other. We did not have time to study the implications of this on our model estimation. We changed the labeling from -1/1 to 0/1 because our cost function was better expressed using 0 and 1.

The training data was again not centered as seen in $Fig.$3a After changing feature 14 to a dummy variable we normalized the data to have 0 mean and std 1.We noticed more outliers in the data of classification than in the one from regression. We removed outliers the same way as in the regression task which left us with 1431 samples (4% of the data).

2.3 Logistic Regression

We applied Logistic Regression (logRegression) and Penalized Logistic Regression (penLogRegression) using gradient descent with learning rate α. Best results are obtained on the train accuracy of 97.14% with $\alpha = 10$. We obtained the optimal α for logRegression and penalty factor λ for penLogisticRegression using the cross validation technique with a $50\% - 50\%$ split for training and validation data.

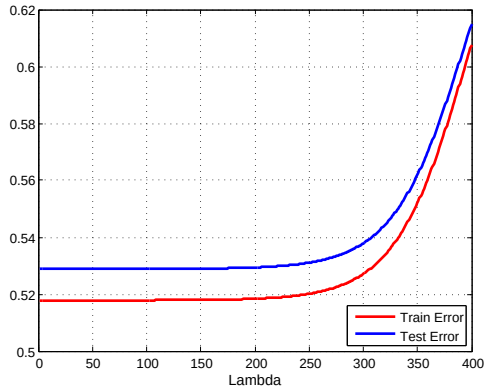
In penLogRegression, the choice of penalty factor (λ) is crucial. We sampled 400 values for λs in the interval 10^{-2} to 10^{3}. As can be seen in Figure 4a, for small λ values, training error is much lower than the test error while for large λ values the test error gets as high as the training error which is a sign of under-fitting.



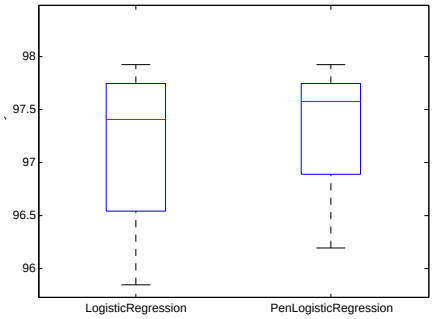
(a) Mean and standard deviation for the training data features. The input is not normalised and contains outliers.

Figure 3: Data visualization.

For small λ values, the training and test error estimated in penLogisticRegression show very similar behavior with the ones for logRegression. The same α value that is obtained for logRegression also gives the best accuracy for penLogisticRegression.



(a) Train and test error assessment with penalty factor (λ) for penalized logistic regression for 50-50 split.



(b) Comparison of logistic regression and penalized logistic regression based on validation accuracy

4b shows the classification accuracy percentage of logRegression and penLogRegression on the test data (4-fold cross validation). The improvements with penLogRegression is little but significant. Best results are obtained on the test accuracy of 97.318% with the values 0.1 for (λ) and 10.0 for (α).

Method	RMSE	0-1-Loss	Log-Loss
Logistic Regression	0.147896	0.020743	105.151844
Pen Logistic Regression	0.122288	0.014693	75.380959

Table 1: Some prediction error estimates for the test data

The table 1 shows error measurements of the test data with $\alpha = 10.0$ and $\lambda = 1.0$ using $RMSE$, $0-1loss$ and $log-loss$. Increasing the value of α decreases all error estimations until a maximum $\lambda = 10.0$ value and we obtained the best prediction accuracy with this value for the λ.

2.4 Feature transformation

We experimented with polynomial, exponential, sqrt(abs*)) of the values but did not manage to improve the previous results obtained using penalized logistic regression. Our best model setup (alpha = lambda = degree =) had a recognition rate of approx 96%, when we used a polynomial of degree 2. Since the model worked and but not better just as it is and since simpler models are preferred, we considered this not to be so relevant.

3 Summary

Acknowledgments

References