Project-I by Group SanFrancisco

Antoine Bastien
EPFL
antoine.bastien@epfl.ch

Quentin Praz
EPFL
quentin.praz@epfl.ch

Abstract

In this report, we expose what we found for the first PCML project. In the two parts of this project, our input matrices were well-conditioned, thus ridge regression and penalized logistic regression did not improve our results significantly. But we found that polynomial transformation was beneficial.

1 Regression

1.1 Data Description

Our dataset contains one train-data with output variables $\mathbf{y}$ and input variables $\mathbf{X\_train}$ and one test-data with input variables $\mathbf{X\_test}$. The train-data contains $N_{train} = 1400$ data examples and we have $N_{test} = 600$ data examples for the test-data. $\mathbf{X\_train}$ and $\mathbf{X\_test}$ are two matrices with dimensionality $D = 51$. Out of these 51 variables, 34 variables are real valued, 13 variables are categorical and 4 variables are binary.

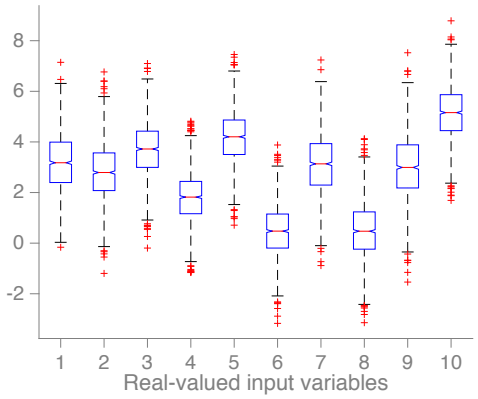
1.2 Data analysis

1.2.1 Visualization and cleaning

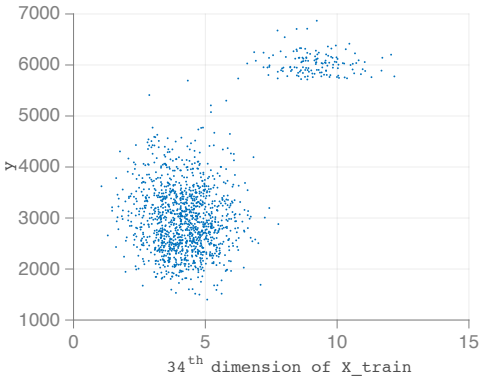
We first plotted the distribution of the 51 dimensions of the input. Figure 1(a) shows the first ten variables and their distribution. We clearly see that the data is not centered and we had to normalize it.
Then we plotted each columns of the train-data input against the output. Figure 1(b) shows an interesting thing that we found. We can see on this figure that there are two clusters, a small one in the upper right that we will call $cluster1$ and a bigger one in the lower left that we will call $cluster2$. We thus decided to apply a regression on the two clusters independently.

1.2.2 Find the best split

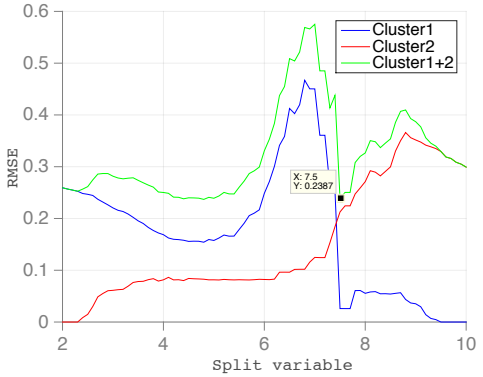
The next step of our data analysis was to find where was the best split decision between the two clusters. Naively, just by looking at figure 1(b), the best cut will be around 7. But this approach is not very academic. In order to be more precise, we first use a more mathematical approach. We did a function that split the dataset for a specific value and then compute the RMSE (using least squares). We then search the minimum of this function to find the best cut. Figure 1(c) shows the RMSE of the train-data and the optimal cut that we should use. But we didn't know if we could rely on this plot because it is not very logical. Indeed, when the cut is between 6 and 7 the RMSE is twice the RMSE when we split in the middle of cluster2 (split at 5). This is something we did not expect and we decided to use another method to confirm (or not) the previous result.
The second approach we use to find the best split is to use a classifier. We used the function $logisticRegression$ on the 34^{th} dimension of the train-data the find the best split. If the probability $p(y = 1|x) > 0.5$, we assign y to 1, otherwise to 0. Figure 1(d) shows the results of the classifier and we see that the best cut we can use it at $x = 7.18$.



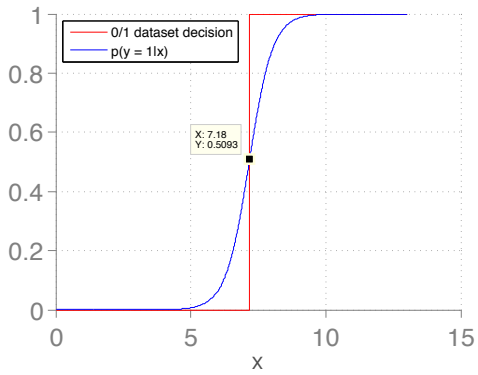
(a) Boxplot of first 10 real-valued columns of $\mathbf{X\_train}$.



(b) Plot of the 34^{th} dimension of $\mathbf{X\_train}$ against $\mathbf{y}$.



(c) Plot of RMSE using leastSquares depending on the split variable.



(d) Plot of the decision rule given by the classifier.

Figure 1: Data analysis.

Finally, if we compare the two results, we see that they are not so different, but since our goal was to minimize the RMSE, we chose to fix the split at 7.5.

After finding the best split, we had to apply some regression methods. The input matrices $\mathbf{X\_train}$ and $\mathbf{X\_test}$ are full-rank, so we did not expect that ridge regression improve our result a lot. But since this method is the most evolved that we have and it cannot increase our RMSE (comparing to least squares methods), we decided to focus on ridge regression.

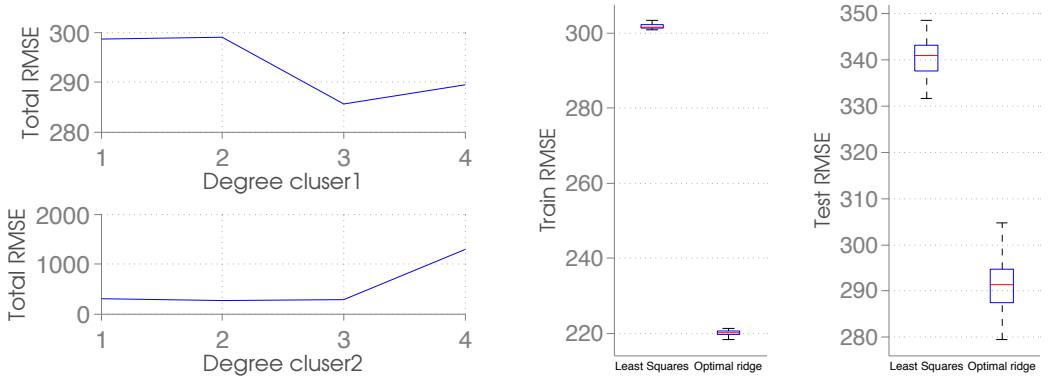
1.3 Ridge regression

We applied least squares using gradient descent, least squares using normal equations and ridge regression to this dataset. Since the $\mathbf{X\_train}$ is full-rank, least-squares methods and ridge regression gave us similar results. Indeed, when we performed ridge regression on our two clusters, the improvement is insignificant regardless of the value of λ.

	least squares (NE)	least squares (GD)	ridge regression
RMSE	337.31	334.19	337.12

Table 1: RMSE using different methods.

Table 1 shows the results obtained with the three methods when we use 25% of the data as test data and rest as training data. We can see that the three methods give similar results.



(a) Plot of the RMSE against the polynomial degree

(b) Comparison of least squares and ridge regression with feature transformation.

Figure 2: Feature transformation.

The next step of our regression process was to apply some feature transformation and find a better regression model.

1.4 Feature transformation

The main feature transformation that we used was a polynomial transformation. The main part of this transformation was to find the best degree for each cluster. For this, we compute the RMSE using ridge regression for different polynomial degree. Figure 2(a) shows the evolution of the RMSE and by looking at the those two plots, we can conclude that the optimal degrees are $d_1 = 3$ for $cluster1$ and $d_2 = 2$ for $cluster2$. We use the cross-validation technique to be sure about those degrees.
With this new transformation, we found that ridge regression could improve our result. We then use cross validation to find the best λ's (one for each cluster). We searched the lambdas that minimize RMSE. To find them, we varied the value of λ from 10^{-3} to 10^2, choosing total 50 points in between. We found that the best lambdas were $\lambda_1^* = 0.16$ and $\lambda_1^* = 9.42$. The RMSE with this transformations is shown in Fig. 2(b). We see that both test and train error decrease and the improvement is significant. We can also see that the variance of the train RMSE is very small compare to the variance of the test RMSE. This is what we expected because we use a K-fold with $K = 4$.

Table 2 shows the good improvement we achieve by using the optimal ridge regression. Because of the improvement we get using this feature transformation, we used this model to do our prediction on test-data.

	least squares	optimal ridge regression
RMSE	337.31	291.32

Table 2: RMSE of optimal ridge regression.

2 Classification

2.1 Data Description

Our dataset contains one train-data with output variables $\mathbf{y}$ and input variables $\mathbf{X\_train}$ and one test-data with input variables $\mathbf{X\_test}$. The train-data contains $N_{train} = 1500$ data examples and we have also $N_{test} = 1500$ data examples for the test-data. $\mathbf{X\_train}$ and $\mathbf{X\_test}$ are two matrices with dimensionality $D = 32$. Out of these 32 variables, 25 variables are real valued and 7 variables are categorical.

2.2 Data visualization and cleaning

We first look at the output $\mathbf{y}$. The values contained in $\mathbf{y}$ were either 1 or -1. The first thing we did was to change the value -1 to 0 to be consistent with the formulas and loss function we will use later.
Then we looked at the distributions of each dimension of $\mathbf{X\_train}$ and $\mathbf{X\_test}$, and we realized that there were outliers. We removed outliers by comparing $\mathbf{X\_train}$ and $\mathbf{X\_test}$ in order that $\mathbf{X\_train}$ looks like $\mathbf{X\_test}$ as much as possible. Note that we did this only one non-categorical variables. Finally, we normalize every non-categorical variables.

2.3 Logistic Regression

To achieve the best error rate, we first performed a backward and forward features selection algorithm [1]. After this selection, we removed 8 dimensions to the input variables. The dimensionality of our matrices $\mathbf{X\_train}$ and $\mathbf{X\_test}$ was reduced to $D = 24$. Like in the regression section, those matrices are full-rank. The penalized logistic regression is then not more efficient.

Table 3 shows the errors we obtained with three different error functions. We can see that those results are good, indeed we avec only 8.62% of 0-1-loss error. To find those results, we use a K-fold algorithm with $K = 4$.

	RMSE	0-1-loss	log-loss
logistic regression	0.3959	0.0862	0.5053
penalized logistic regression	0.3958	0.0869	0.5055

Table 3: Errors with logistic regression and penalized logistic regression.

2.4 Feature transformation

Since penalized logistic regression did not improve the results, we tried to apply some transformation on the features. The best transformation was to use polynomial transformation. Figure 3 shows the results we obtained after cross validation to find the optimal degree. We found that the best polynomial degree was $d = 2$.
With this new transformation, we found that penalized logistic regression could improve our result. We then use cross validation to find the best λ. We searched the lambda that minimize the 0-1-loss error since it was the purpose of this classification part. We found that the best lambda was $\lambda^* = 0.2610$. To find it, we varied the value of λ from 10^{-3} to 10^2, choosing total 50 points in between.
We can see in Tab. 4 that we improve our results using the feature transformation mentioned before. It is not very significant for the RMSE, but for 0-1-loss we decrease the error from 8.62% to 5.72%.

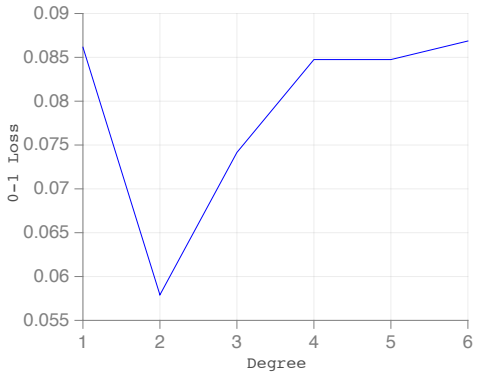


Figure 3: Plot of the 0-1-Loss error against the polynomial degree.

	RMSE	0-1-loss	log-loss
penalized logistic regression	0.3959	0.0862	0.5053
polynomial penalized logistic regression	0.3872	0.0572	0.4868

Table 4: Errors before and after feature transformation.

3 Summary

In the regression part, we found that the data could be divided in two. Then we apply ridge regression on the two clusters. Ridge regression did not improve our result a lot because our two clusters was well-conditioned. The main improvement was provided by a feature transformation. We applied a polynomial transformation on each cluster, the first one with $d_1 = 2$ and the other with $d_2 = 3$. With this transformation, we achieved a RMSE of 291.32. Note that this number is quit high because the input variable $\mathbf{y}$ were between 1400 and 7000.

Concerning the classification part, we removed the outliers to be consistent with the test-data and we selected a set of features using a backward and forward feature selection method. Like in the regression part, the matrix was well-conditioned. Penalized logistic regression did not improve our result, but it was expected du to this well-conditioned matrix. Then we searched some improvement, and like in the regression part, we apply a polynomial transformation. We found that the best degree was $d = 2$. With this transformation, we achieve a 0-1-loss of 5.72%.

Acknowledgments

We would like to thank Antoine for implementing the useful functions, Quentin for writing this report and all the PCML teaching team who made a great work to provide us the data and the help that we needed.

References

[1] Dr. Arun Ross. Feature selection. `http://www.cse.msu.edu/~cse802/Feature_selection.pdf`. [Online; accessed 27-October-2014].