Model Selection, Feature Selection, and Prediction: Drug Sensitivity with Binary Predictors

Guangju Wang
*Department of Industrial Engineering and Logistics Management
Hong Kong University of Science and Technology
Clear Water Bay, Hong Kong
gwangal@connect.ust.hk

Abstract

This report presents some the results of some statistical experiments on drug sensitivity dataset with binary features. Various statistical models are fitted and evaluated. With the help of cross-validation and visualization, this reports tries to explore the benefits and weakness of different models on this specific dataset. Non-linear models tend to perform better than linear models with feature set is large. However, after a smaller feature set is selected by cross-validation and manually tuned LASSO, linear models outperform non-linear ones significantly. Neural Networks, with similar performance, requires longer computation time and weakens the interpretability of the model.

1 Background and Overview

1.1 Data description

The Drug Sensitivity Dataset contains 642 cancer cell line samples in response to one drug, Afatinib, with 60 binary features as gene mutation status. Sample ID, cell line names, real valued response as drug sensitivity measured by logarithmic IC50, as well as 60 binary features are provided for each cell line sample. A lower IC50 value indicate that the cancer line is more sensitive to the drug.

The response of a random subset of 100 cell line samples are withdrawn as the test sample. For a better description of data, please refer to section 5.1 of:

http://math.stanford.edu/~yuany/course/2017.spring/project2.pdf

1.2 Overview of the Report

Section 2 sets some baselines to which we can compare other models. Insights about correlation between features are also given in Section 2. Section 3 focuses on feature selection with Cross-Validation LASSO and manually tuned LASSO. Section 4 compares performance of non-linear models to linear models. Section 5 provides a visualization of fitted result, and compares Neural Network to linear models in terms of prediction power, computation time, and interpretability. Section 6 concludes the report.

*The dataset is provided by Prof.Jiguang WANG and Dr. Biaobin JIANG and is used by Prof. Yuan YAO for education purpose. This report is only for grading purpose of class MATH 6380, held by Prof.Yuan Yao, at the Hong Kong University of Science and Technology.

1.3 Tools used

Most of the analysis are done in Python 2, with the help of 'pandas DataFrame' and the well-established Machine Learning package 'scikit-learn'(https://www.scikit-learn.org).

2 Some Baseline Models

Before any treatment, the set of samples to be used for training and validation (samples with known IC50 values) is separated from those with unknown IC50 values. From now on, unless specially noted, "samples" refer to those with known IC50 values.

2.1 Random Forest

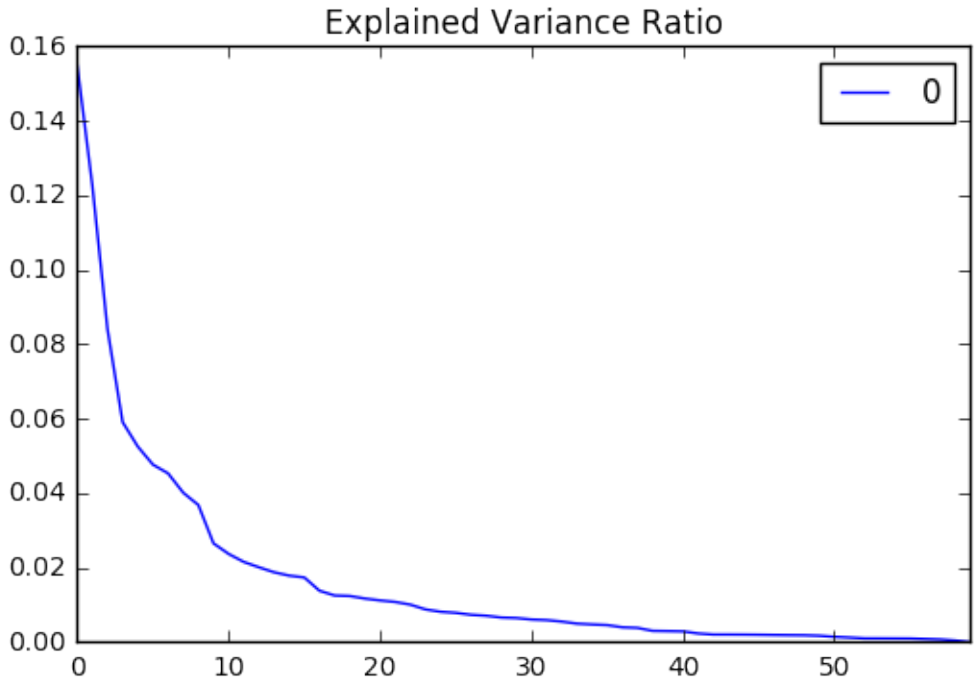
As all features are in binary form, the first baseline model is constructed by fitting a Random Forest regressor with all features. Intuitively, Random Forest method should be good at dealing with binary features. Without validation, all parameters are set as default in sklearn package. For details, please refer to:

`http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestregressor.html`

The performance all models below are measured in mean square error(MSE). Cross validation method is used for evaluation purpose. As usual, the training-validation ratio is 0.8/0.2 as usual. In such model without any tuning and validation, the average the MSE in 50 times of cross validation is around 3.7.

2.2 PCA and Linear Regression

To get a sense of the correlation structure of the feature set, Supervised Principle Component Analysis is performed on the feature set. The result indicates that the first 10 principle components account for most of variation of features.



With the above PCA result, it is natural to build a model with the top components. Firstly, each sample is transformed into combination of Principle Components. Then a linear model is fit on each sample's the coefficients of the top 10 Components. With the same evaluation method, the average MSE is reduced to around 3.2.

So far, we constructed two baseline models and concluded that the feature set is strongly correlated. However, just taking the top 10 Principle Components may not be enough. We shall try the more frequently used LASSO method in the next section.

3 Feature selection

3.1 Cross Validation LASSO

In order to apply LASSO appropriately, we need to determine the parameter λ (or alpha in sklearn syntax). 'Sklearn.linear_models' package provides us with a handy tool 'LassoCV' that determines the LASSO parameter for us through cross validation.

Fit the Cross Validation LASSO model with all samples, a λ value of 0.0153 is returned. And the selected feature set is:

BRAF, CDKN2A, EGFR, ERBB2, JAK2, MYC, NRAS, SMAD4, TP53, VHL, EWS_FLI1
(Feature Set 1, or FS1,l length 11).

With the cross validated LASSO model, the average MSE is further reduced to around 2.82.(The process was run for several times, this number may vary from about 2.79 to about 2.85)

Cross Validation LASSO kept 11 features in the selected feature set. However, the MSE was reduced significantly compared to the model of PCA and linear regression. This result may be because that Principle Components Analysis, although supervised, tend to keep features that explain more variance even if they do not help in prediction.

3.2 Random Forest

With a smaller set of features, Random Forest regressor model is fitted again. The average MSE is around 3.0. LASSO outperforms Random Forest on this smaller subset. One possible reason is that the Random Forest Model is lack of tunable parameters. With a smaller feature set, it is less powerful in revealing the numerical relationship between features and output.

3.3 Manually Tuned LASSO

With the concern that Cross Validation LASSO may not give the optimal features set, various lasso parameters (λ) are manually set. This process gives several features sets that potentially may give us a better model. After several trials, the feature set

BRAF, EGFR, ERBB2, MYC, NRAS, SMAD4, TP53
(Feature Set 2, or FS2, length 7)

seems to give least MSE of around 2.8. This result is close to the linear model with FS1. However, with a smaller feature set, the result of this model may be more stable, intuitively.

In fact, when submitted to Kaggle contest, the linear model with FS2 significantly outperformed the one with FS1. Which in return verified our conjecture that the former one may perform better on out of sample test.

4 Non-Linear models

To account for some possible non-linearity of the problem, Gradient Boosting Regressor and Nearest Neighbor regressor are considered. In case that the structure of the problem is non-linear in a strong way, these regressors may perform better.

4.1 Gradient Boosting Regressor

Gradient Boosting Regressor(GBR) is built on a number weak learners and is known to have relative strong predictive power. Another important feature of GBR is that it can be used for any differentiable loss function(because it uses gradient descent). For details, refer to

`http://scikit-learn.org/stable/modules/ensemble.html#gradient-boosting`.

GBR performs differently in two different scenarios:

1. On the whole feature set(length 60), GBR, with a average MSE of about 3, outperforms most of other models.
2. On Feature Set 2(length 7), GBR gives an average MSE of about 2.85, which is worse than simple linear regression.

It seems that GBR tends to perform better on a larger feature set. When applied on a small feature set, the large number of weak learners does not have the power to finely tune their parameters as the linear models do. Although the GBR model may account for some non-linearity of the problem, it is not as strong after we focus on the principle features.

4.2 Nearest Neighbors Algorithm

K-Nearest Neighbors regressor seem to perform badly no matter what K values we set and what distance metric we use. The average MSE was always above 3.1 or 3.2.

4.3 A brief recap

With all the analysis above, one conclusion may be that the performance of our regression model depends highly on the feature set we selected and the flexibility of the model in terms of its parameters. Table 1 is a summary of all models and their performance (Cross-Validation MSE).

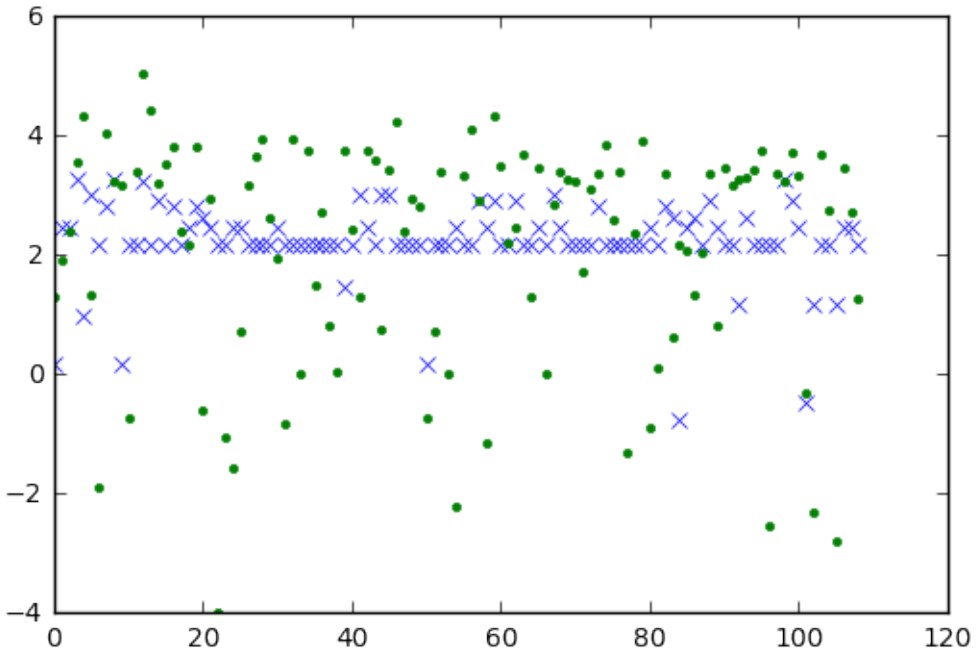
Table 1: Models

Models	Feature set	Average CV MSE
Random Forest	All Features	3.7
PCA&LR	Top 10 Comp	3.2
LASSO	FS1(11)	2.8(High Variance)
Random Forest	FS1(11)	3.0
LASSO	FS2(7)	2.8(More Stable)
GBR	All Features	3
GBR	FS2	2.85
KNN Regression	N/A	>3.1

5 Blackbox Model and Some Discussion

5.1 Visualization

To visually present the fitted result of the selected model, the predicted values and actual values of the a random validation data set are plotted. In the picture below, 'x' marks the predicted value, and '.' marks the actual value.



The result indicates that the linear model with FS2 is still lack of prediction power. Most of the predicted results lie around a line. Although the model provides some prediction for values below or above the line, most of the points that are far away from the line are not predicted well.

One possible reason for this phenomenon is that, the data may be clustered, and that, with limited features, linear models are lack of the ability to discover the difference between clusters.

However, without proper labels and other supplementary materials, the clusters are hard to determine. As mentioned above, supervised PCA did not improve our prediction much.

5.2 Blackbox

The former section focuses on selecting the optimal feature set with selected model. In all sections above, we tried to interpret why does one model works better than another. In this section, Neural Network Model is applied to the data without any validation and submitted to Kaggle. We hope that Neural Network model may discover more patterns than linear models in the dataset.

With this motivation, a Multi-layer Perceptron regressor is fitted to the data. Kaggle returned a score that was very close to the Linear model with FS2. That is to say, Neural Network did not provide more insights than what we had before. One may also argue that Neural Network makes the model less interpretable and may not provide as much insights as the linear models do.

6 Conclusion and Future Work

In this report, various statistical models are applied to a drug sensitivity data with binary input. To get started, some baseline models are constructed. To improve, we selected a small feature set by Cross Validation LASSO and manually tuned LASSO. Some non-linear models are also applied to cater for non-linearity of the model. Finally, a Multi-layer Neural Network is applied and turned out to have similar performance to linear model with selected parameters.

The findings of this report can be summarized as:

1. Feature selection is crucial. Cross-Validation LASSO may not give us the optimal feature set to use. Sometimes it is still necessary to manually tune the LASSO parameter.
2. Non-linear models tends to outperform linear models when the feature set is large. When focused on small subset of features, linear models has better ability to uncover the relation between numbers than tree-based methods.
3. Neural Network or blackbox model, may help us find out some mechanisms that are not easily found out if we use specific models. But in return we have less control on the model, and the interpretability of the model will largely decrease.

Acknowledgments

The author is thankful to Prof. Yuan Yao for the nicely prepared lectures and also for providing the datasets and hosting Kaggle competition. Without his help, all the experiments and discussions above would not be possible. This report should not be used for publication purpose or distributed outside of MATH 6380 class. Any enquiries, please get in touch with the author(gwangal@connect.ust.hk).