Drug Discovery: Predicting Molecular Activity with Machine Learning

Kostantinos Morfesis

May 19, 2019

Abstract

In the process of bringing a new drug to market, pharmaceutical companies can spend up to $2.7 billion in R&D and clinical trials. This hurts not only the pharmaceutical companies, but patients as well. The higher costs decrease incentive to develop new patient drugs which patients may benefit from. As a tool, machine learning can be used to expedite this process and lower some of the costs.

During R&D, it is necessary to determine how different molecules interact with a particular biological target. This can take time, but using machine learning one can predict the activity levels given molecular features. This paper determines the optimal models for this and uses them to obtain a prediction. Combining ridge and neural network regression to create an ensemble prediction, an average R-squared of 0.665 was found for the activity levels with an 80/20 train test split. The results indicate prediction with the given features can describe a significant portion of the variance in activity levels. The client, Merck pharmaceutical company, can use this in decreasing the number of molecules to test. In further analysis, additional features, hyperparameter optimization, and increased model usage may yield increased predictability.

Contents

1 Introduction

During the drug discovery process, there is a need to discern how a particular molecule will affect different biological targets. Where the biological targets refer to any entity which is being directed or targeted by some given molecules. The binding of each molecule to each target is measured by an activity level, this varies depending on the target. The testing of many different molecules on many different targets can be a costly process. As a result, being able to predict the activity level for a given molecule and target combination before measurement would reduce this cost. This paper aims to answer the question of: for a given molecule can one discern how well it will interact with intended and unintended targets?

The client for this project is Merck pharmaceutical company. With the successful information obtained through machine learning, the company may be able to increase efficiency in testing new drugs. In turn, this would reduce their R&D costs in drug production. It takes an average of 10 years [1] and an average cost of $2.7 billion to bring a new drug to market[2], and there is no guarantee of it being profitable. By reducing the R&D costs, the risk undertaken when developing new drugs can be lowered and the process can be expedited. Not only does this benefit the pharma company, but also the lives of patients who may benefit from novel drugs.

To investigate the molecule to target activity levels, some information will be needed about the molecules. Further, activity levels along with molecule names will be needed to train the models. This data comes from the Merck Molecular Activity Kaggle Competition[3], which contains the necessary information

The procedure by which the molecule to target activity levels will be predicted is outlined as follows:

- The data will be gathered, examined, and cleaned. This encompasses doing all which makes sure the data is ready to be modeled.
- Preliminary investigation will be done to check for abnormalities as well as to obtain a better picture/understanding of the data.
- Feature engineering and selection will be done to lower the number of irrelevant inputs.
- A choice of machine learning algorithms to use will be made. This will be made based on the information obtained in the previous steps.

2 Data Sources and Cleaning

The data source being used comes from Merck Molecular Kaggle Activity Challenge webpage. It consists of a training set with 15 csv files in the following format:

Column 1: Molecule ID - it should be noted that this is an anonymous identifier made for the competition. No knowledge of the actual molecules is given. IDs are in the format of M_1, M_2, etc.
Column 2: Activity - raw activity values which may be in different units depending on the measure.
Column 3: N: Molecular descriptors/features - This is also anonymous and so no true knowledge of the features is given. They are in the format of D_1, D_2, etc.

In total, the training set consists of 2G of data, thus making loading and computations a bit slow on the entirety of the dataset. Since this data is from a competition and is structured, there are relatively few data cleaning steps. However, there may be some molecules which have very few features which needs to be considered. These molecules would be unreliable in training unless those few descriptors accurately predict the activity level.

Upon investigating the data, it was clear many of the feature entries were zero. Looking at the sparsity of the data can be useful for multiple reasons. The average sparsity, or the ratio of zero values to non-zero values came out to an average of about 0.94. This implies about 94 percent of the data is zero. If the data is extremely sparse, one can use sparse matrix computations for lower run times. This was not implemented in the analysis, but should be noted for potential future analysis.

3 Exploratory Analysis and Feature Engineering

3.1 Exploratory Analysis

First step in exploring the data is looking at the distribution of prediction values. Histograms of the activity levels for each dataset (different targets) were plotted. This gave some information about the spread of activity levels. The spread differed noticeably from target to target. Some of the targets had a large spread, indicating a large feature dependence of the activity level (fig. 1). Where as some were much more localized, indicating the activity level acted as being more

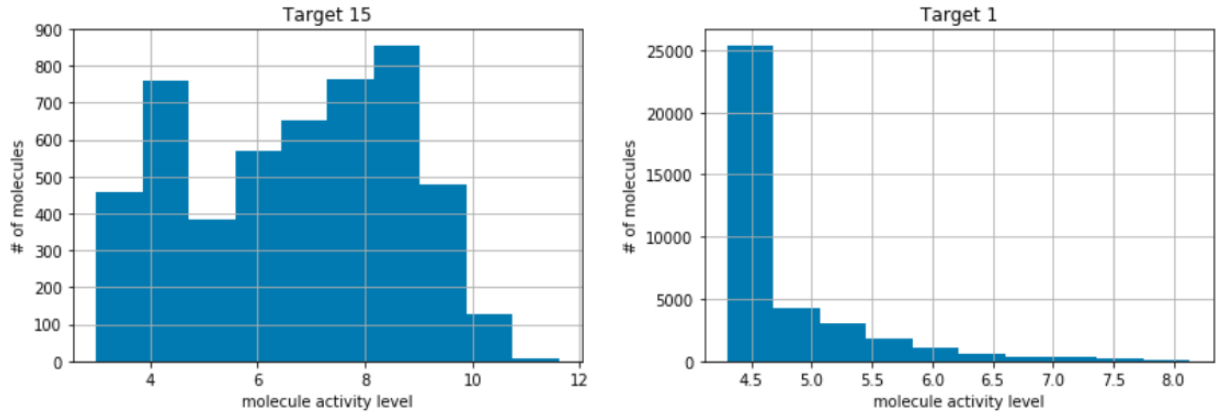

Figure 1: Activity Value Counts: Act 15 on the left and Act 1 on the right. One can see the huge difference in distribution of activity values.

independent of the features.

The next step in exploration was to determine which (if any) features had a column, but were not actually represented in the data. That is, which features were not represented in the data. This was done by simply summing the columns and seeing which were zero.

The last step involved was determining the correlation coefficient values for all of the features on the activity level. This yields the statistical significance of each individual feature with a 95 percent confidence level. The highest correlation found was in target 15 with a value of 0.61. Further, the top correlation values in target 15 were all large compared to other targets. This agrees with the observation above, where target 15 had the largest variation in values.

3.2 Feature Engineering

Given the results of the exploratory analysis, one can significantly reduce the number of features without reducing accuracy. This was done first by removing all features which had all zero values, they provide no useful information. Also, any feature which had a p-value ¿ .05 was discarded as well. This removes those features which have a non-statistically significant result on the final activity level. The final resulting size of data was reduced by about 55 percent. This significantly reduced the computation time, but did not cause any substantial loss in accuracy.

4 Machine Learning Prediction Methods

4.1 Classification Scheme and Accuracy Measures

The purpose of this prediction is to determine the activity level given some molecule features. The activity level is a continuous variable, so the type of modeling is regression.

The competition chose to measure the prediction accuracy with R-squared, which is a common metric for comparing the fit of a regression prediction. This is defined as follows:

$$R^2 = 1 - \frac{\sum_i (y_i - f_i)^2}{\sum_i (y_i - \bar{y})^2} \tag{1}$$

Where y_i are the actual dependent variable values, f_i are the predicted dependent variable values and $\bar{y}$ is the mean of y.

Following suite, R-squared will be used as the accuracy metric. Also, note that the training and testing here will be done entirely on the training set obtained through the Kaggle website. This is in contrast to how the testing is normally done, which should be on the provided test set. This comes from a lack of available test activity values, so accuracy cannot be measured. Given this, comparison of R-squared values with other competitor values will not yield a meaningful result. As the competition alluded to, the test set was performed on data which was taken after a certain date. Therefore, the features in each train test pair may be different, leading to difficulty in prediction. Since my prediction will be done on the training set, this is not an issue and a higher R-squared value should be expected.

4.2 Choosing Models and Preliminary Analysis

The simplest model to use in this case is ordinary least squares (OLS) regression. This will provide insight into what other models to try. As an initial analysis, OLS regression was trained on the Act 1 training set with an 80/20 train-test split. For the OLS, there was a clear sensitivity to outliers (fig.3). Even though the actual activity values ranged from about 4.5 to 8.0, the OLS regression could capture this trend for many points, but a few outliers were present. This lead the predicted activity values to range from -0.4 to $0.8\text{x}10^{14}$, which is tremendously inaccurate. The resulting R-squared value for the OLS model was -4.08.

As an attempt to remove the sensitivity to outliers and tendency to over fit, ridge regression can be used (fig. 4). Where an extra regularization term is added to the loss function of OLS which is scaled by the parameter α. The higher α, the greater the regularization and rigidity of the model. Using ridge regression and a grid-search with five fold cross validation, the best fit was obtained with $\alpha = 1000$. This is a comparatively large restriction on the flexibility of the model. The resulting ridge regression R-squared with $\alpha = 1000$ was 0.56. Since this is a tremendous improvement over OLS, ridge regression will be used in further analysis as well.

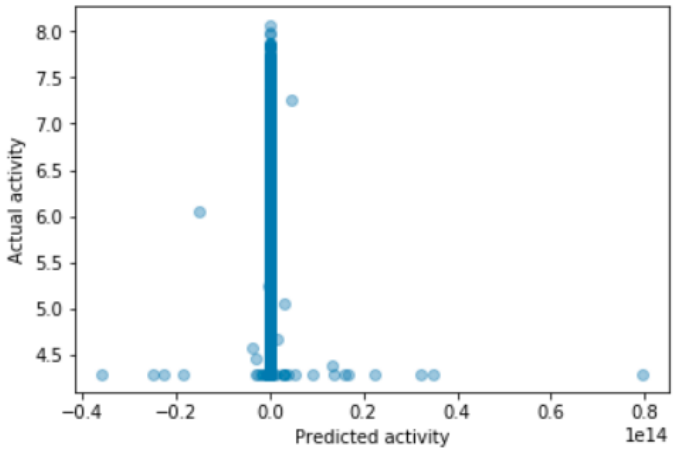

Figure 2: OLS Regression: this model had poor accuracy with an R-squared of -4.08 on Act 1. This was due to the large variance in predicted variables.

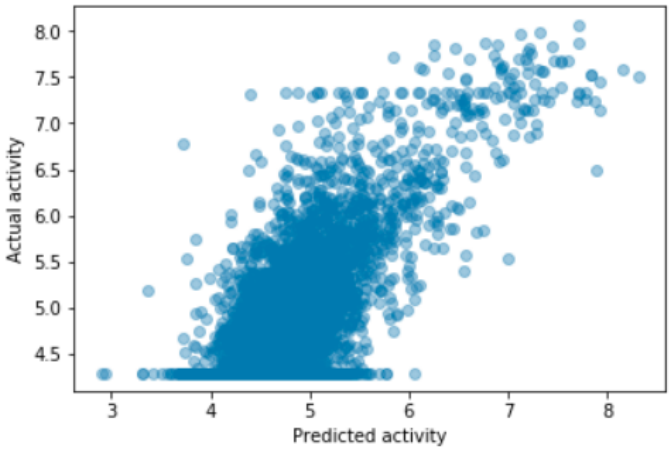

Figure 3: Ridge Regression: Act 1 ridge regression with $\alpha = 1000$. The addition of a regularization term greatly reduced the variance in predicted values and increased the R-squared.

Interestingly in fig. 4, the actual vs predicted values follow a slightly curved form. This could imply that the model will benefit from non-zero second order terms. There's many ways

to account for non-linear behavior. Here, simple polynomial regression and neural network regression will be used. The polynomial regression was done with a ridge loss function and the only difference was the addition of the square of the features to the feature space. This takes the following form:

$$y = \beta_0 + \sum_i (\beta_i x_i + \alpha_i x_i^2) + \epsilon \tag{2}$$

Where there are two terms in the sum, accounting for linear and quadratic relationships. Applying this on the Act 1 dataset did not result in a substantial improvement in the R-squared from the previously found ridge regression. Consequently, the polynomial regression isnt used in the remainder of the analysis.

In total, the final regression models which will be used are ridge regression and neural network regression. Further, once analysis is done individually, an ensemble prediction based on the three models will be done. In general, the creation of an ensemble prediction can work to increase the overall R-squared and effectiveness of the prediction.

5 Machine Learning Analysis

5.1 Ridge Regression Analysis

Since ridge regression is a fairly straightforward model to implement, not too much tuning needed to be done. Especially since from above, the optimal alpha parameter was found to be 1000 for the Act 1 data set. This value for alpha will be used in the remainder of the tests, but this does not necessarily imply it is the optimal parameter. During the training, the following pre-processing steps were performed:

- A grid search with 5-fold cross validation to determine the optimal alpha parameter.
- Scaling of the data such that each feature had mean 0 and variance 1.
- Train-test split was created with an 80/20 split.

The results over all of the acts can be seen in fig. 5, the mean is 0.62 and the standard deviation is 0.11. One can see that Acts 5 and 6 were the most difficult to predict with R-squared values

closer to 0.40. That being said, the ridge regression prediction was quite consistent over the datasets.

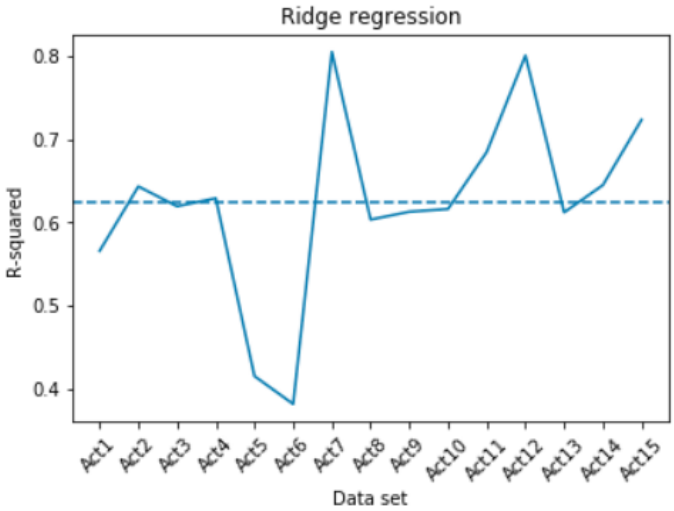

Figure 4: Ridge Regression R-squared values: Mean R-squared was 0.62 and Acts 5 and 6 were the most difficult to predict on.

5.2 Neural Network Regression Analysis

For the neural network analysis, care was needed to determine the optimal values of hidden layers, nodes, and other hyper parameters. From the initial analysis which looked at the addition of non-linear (squared) terms to the models, there was little change in accuracy. Consequently, one would expect there need not be a large number of hidden layers. In practice this turned out to be true, the optimal number of hidden layers was 2. While the total number of layers was 4 with the inclusion of the input and output layers. Further, in each of the hidden layers, rectified linear units (ReLU) for the addition of non-linearity. This is done such that after applying a weighted sum to each node, the total value is input into the ReLU function. Further, the approximate optimal number of nodes in the first hidden layer was found to be 100 whereas the optimal number was 50 in the second layer. It should also be noted that all algorithm optimizations were done on Act 1. It is possible the accuracy could increase if optimization was done on all datasets individually. To determine the optimal number of epochs, early callbacks were performed on a 80/20 train-test split. This works in such a way that once the mean-squared error of the validation set doesnt decrease for more than a patience of 2, the run is stopped. On average, the number of epochs stopped being advantageous after 8 (fig. 5).

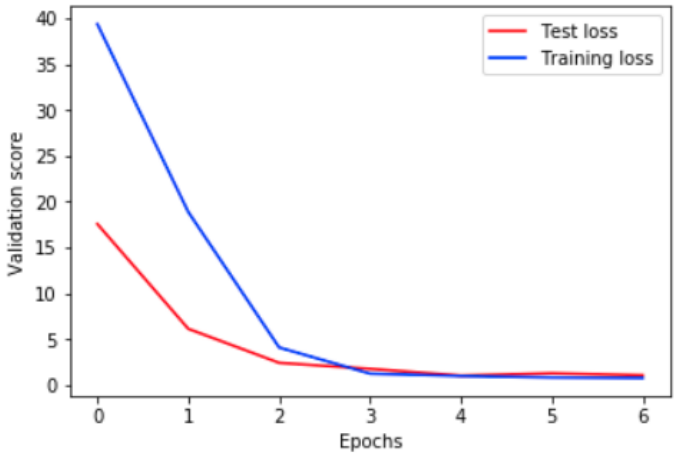

Figure 5: Train test Loss for NN Regression over Epochs: One can see the test set mean-squared error stops being beneficial after about 6 epochs. Note that the epochs were all trained with a batch size of the entire dataset.

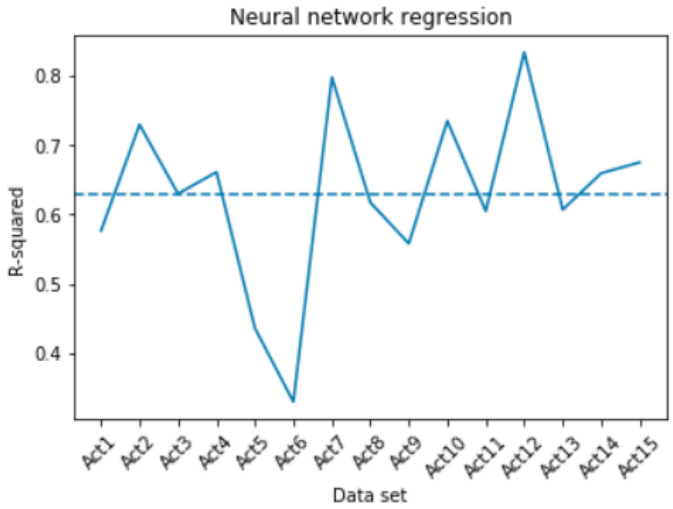

Figure 6: NN Regression R-squared Values: Note the difficulty in predicting Acts 5 and 6 exists in both the NN and ridge regressions. However, there is a noticeable increase in accuracy in Act 10.

One can also see the results of the NN analysis in fig. 6. With a mean of 0.630 and standard deviation of 0.128 the NN regression had the largest variation in values. Also, the NN R-squared was quite high given the small amount of optimization done. That is, if searching was done for optimal parameters across all datasets, one could expect a substantial improvement in accuracy.

5.3 Comparisons and Ensemble Averages

To increase the overall accuracy of the prediction and to see the differences and similarities between models, an ensemble can be created. This consists of averaging the predicted activity levels across all models (fig. 7). One can see that the average value has a noticeably improved R-squared value of 0.651. This indicates that the models are constructive and not destructive to one another.

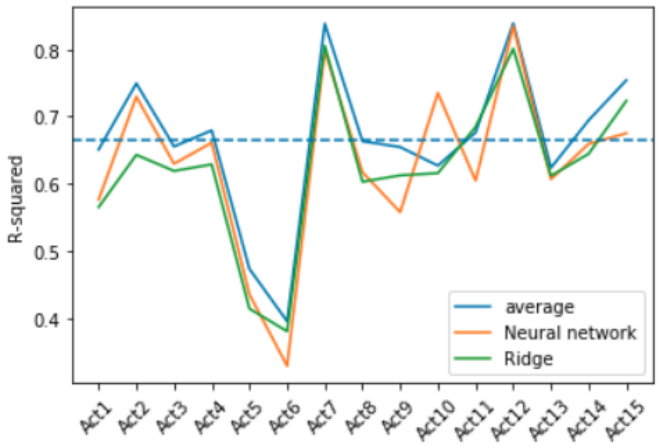

Figure 7: All Model R-squared Values: Note the average model which was obtained from a weighted average of predictions from other models.

6 Results and Future Analysis

In the drug discovery process, a researcher needs to know which molecules affect which biological targets and to what extent. In this analysis, thousands of molecules and their known activity levels on different targets were trained and then predicted on. For comparison, two different models were trained on: ridge regression and neural network regression. The final model with the highest accuracy consisted of an ensemble of the two regression models. The results indicate that for future data with the same number of features, one can expect to predict with accuracy of an R-squared = 0.651 on average. The model may not yield perfect activity levels, but could still yield actionable insights. For example, the data could be split into activity above or below

some given value. Molecules which are predicted to be below a certain value can be discarded during the discovery process, while the rest will remain. This can provide an easy way to sort through molecules without having to test all of them individually.

In future analysis, the R-squared may be improved by increasing the number of models in the ensemble average. Or through spending more time optimizing parameters in the neural network analysis.

References

[1] Joseph A. DiMasi; Henry G. Grabowski; Ronald W. Hansen. Innovation in the pharmaceutical industry: New estimate of r&d costs. *Journal of Health Economics*, 2016.

[2] PhRMA. Biopharmaceutical research and development. 2015.

[3] Merck. Merck molecular activity challenge. 2012.