Mechanisms of Action (MoA) Prediction

Chi Zhang

December 18, 2020

1 Introduce and Background

In the past, scientists derived drugs from natural products or were inspired by traditional remedies.However,today drug discovery rely on a more targeted model based on an understanding of the underlying biological mechanism of a disease.So understanding the underlying biological mechanism of a disease is the key factor to discover drug.Thus,researchers seek to identify a protein target associated with a disease and then develop a molecule which could modulate that protein target.And researchers call the biological activity of that given molecule "Mechanism-of-action(MoA)".[1]

In order to determine the MoAs of a new drug, researcher treat a sample of human cells with the drug and then analyze the cellular responses with algorithms that search for similarity to known patterns in large genomic databases, such as libraries of gene expression or cell viability patterns of drugs with known MoAs.[1]

Here in this project, with the dataset that includes gene expression and cell viability data in addition to the MoA annotations of more than 5,000 drugs,we need to predict multiple targets of the Mechanism of Action (MoA) response(s) of different samples.Since drugs can have multiple MoA annotations, this is a multilabel classification problem. Besides, we make use of the gene and cell data which contain a lot of features,so it is challenging to select proper features to train models. Thus, I'd like to consider to conduct feature engineering using pricipal component analysis first to reduce the dimension of the features, and then try to use linear and non-linear multi-label classification models to predict and at last compare these models.

2 Exploartory Data Analysis

2.1 Features

There are 23814 samples in the training dataset and 3982 samples in the testing dataset. For each sample there are 876 features, which include categorical and numerical features.

2.1.1 categorical features

First Let met introduce the categorical features. There are three categorical features: treatment type, treatment time and treatment dose. There are two types for treatment type. trt_cp means treatment group, while ctl_vehicle means the control group. For treatment time, there are three types,24h ,48h and 72h. Besides the treatment dose includes high dose

and low dose. Those categorical features describe how samples were divided and treated after collecting. Below are the distributions of these features

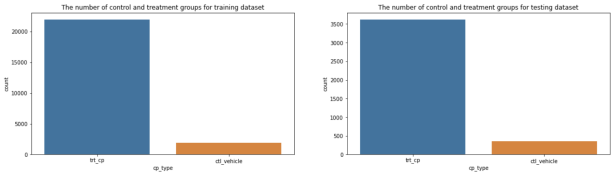

Figure 1: The distribution of treatment type.

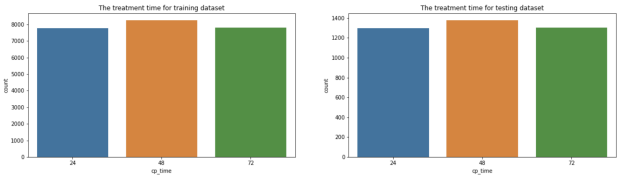

Figure 2: The distribution of treatment time.

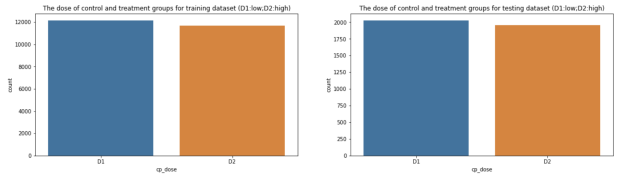

Figure 3: The distribution of treatment dose.

From these plots, we could observe that the most of treatment type are compound treatments ("trt_cp"), compared with only about 8% of control ("ctl_vehicle").The treatment dose and the treatment time are approximately evenly balanced.Since controls have no MoAs, we only consider treatment group to train the model.

2.1.2 numerical features

Numerical features include 772 features of gene expression data and 100 features of cell viability data. Since there are so many features, We choose to draw the distributions of some of them.

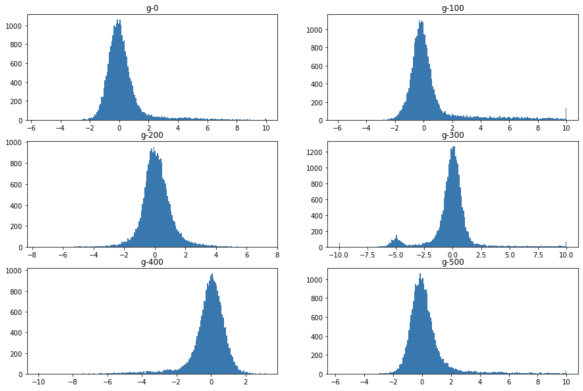

Figure 4: The distribution of gene expression(only select some of them).

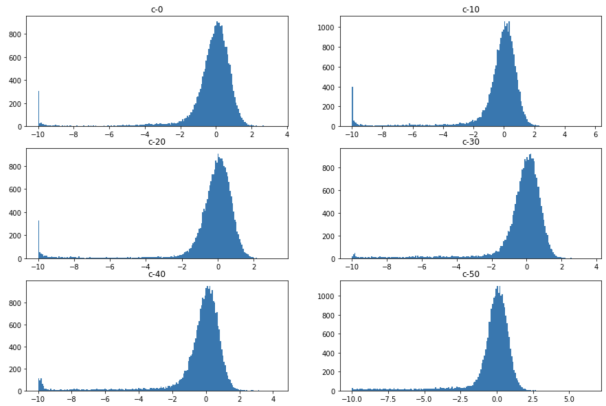

Figure 5: The distribution of cell viability data(only select some of them).

The distributions of gene expression are quiet normal.Though we still could observe there exits a bit of skew, I think there is no need to do transformation of them. Besides, the distributions of cell viability are also approximately normal, but with close observation, we could found that there are bumps around values of -10 for some of them.

Since there are much many features, we'd like to find whether there is correlation between them, which is also our motivation for feature engineering.

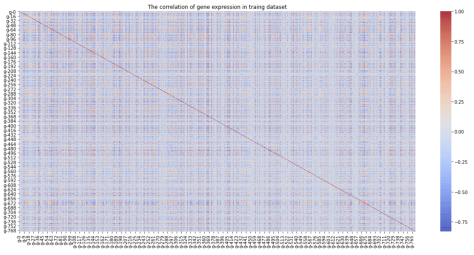

Figure 6: The correlation of gene expression.

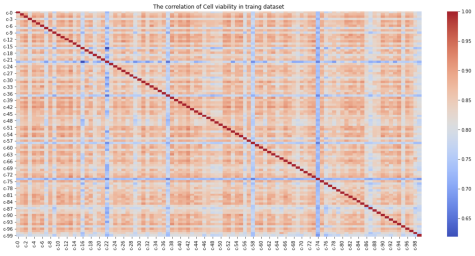

Figure 7: The correlation of cell viability data.

Through the whole pictures of gene expression correlation, there are certainly some pairs of features positive or negative correlated. But there are more positive correlated pairs than

negative correlated pairs. As for the cell viability, only exists positive correlated pairs and all pairs' correlation are larger than 0.6. The pattern of the correlation within cell viability data is also possibly influenced by extreme values.

2.2 targets

There are totally 206 targets, which are all binary variables. Our mission is to build a multi-label classification and for each sample, it could have multiple positive MOAs.

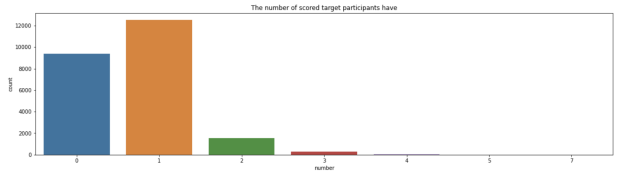

Figure 8: Number of MOA per sample.

The plot shows that at most 7 MoA are active for one sample in our dataset. About 39% of training samples have no positive MoA. 50% samples have one positive MoA which is the most.For 2 positive MoAs, there are about 6.5% samples and the case of 3 positives MoAs is slightly above 1%. It is quiet rare to have 5 or 7 positive MoAs.

3 feature engineering

3.1 PCA

Since there are so many features and many of them are correlated with each other, so we'd like to reduce the dimension of our features. So in this project, dimensionality reduction is achieved by using Principle Component Analysis (PCA). PCA first finds the eigenvectors and eigenvalues of the covariance matrix.Then with eigenvectors, original data can be projected into the reduced PCA space. It could convert our original feature matrix into a new matrix with same or less features And these features are independent with each other.[3] The new PCA features capture a significant portion of the information of the original data. The eigenvectors with larger eigenvalues contain more information. Here we choose the Principle components which explain about 95% variance of the original data.

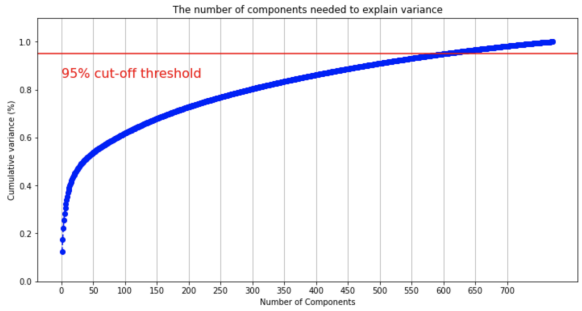

Figure 9: PCA variance plot for gene expression data.

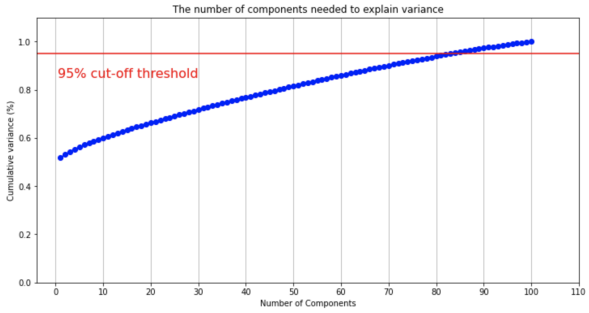

Figure 10: PCA variance plot for cell variability data.

These plots trace the variance Principle components explain. For gene expression data, we choose first 603 components and for cell validation data, we choose about first 84 components. Then after adding treatment type, treatment time and treatment dose, there are total 690 features we have selected.

4 Models

First, I'd like to clarify the evaluation we use is the logarithmic loss:

$$score = -\frac{1}{M}\sum_{m=1}^{M}\frac{1}{N}\sum_{i=1}^{N}\left[y_{i,m}\log(\hat{y}_{i,m}) + (1-y_{i,m})\log(1-\hat{y}_{i,m})\right]$$

Where N is the number of samples and M is the number of MoA scored targets.$\hat{y}_{i,m}$ is predicted probability of MoA m positive in sample i and $y_{i,m}$ is the ground true, 1 for a positive response, 0 otherwise.

We totally tried three models which include linear model and non-linear model. In this case, we could compare the performance of linear and non-linear model. For each model, we use 5-fold cross validation to get our training and testing average logarithmic loss. For K-fold cross validation, the data set is divided into k subsets, and the models is trained k times. Each time, one of the k subsets is used as the testing set and the other k-1 subsets are put together to form a training set. At last the average error across all k trials is computed.

4.1 multi-label Logistics Regression

First, I apply multi-label logistics regression on this problem and make use of the Multi-OutputClassifier wrapper in sklearn. When there is only one response, the log-likelihood of logistic regression is:

$$l(\beta) = \sum_{i=1}^{N}\{y_i log(p(x_i,\beta) + (1-y_i)log(1-p(x_i,\beta))\}$$

Here we have 206 targets which are all binary variable, so we apply logistics regression on each targets to get our prediction.

4.2 XGboost

Extreme Gradient Boosting(XGBoost) is a scalable end-to-end tree boosting system. It apply a novel sparsity-aware algorithm for sparse data and weighted quantile sketch for approximate tree learning[2]. XGBoost is in essence a Gradient Boosting Decision Tree. The objective function of XGBoost is:

$$L(\phi) = \sum_i l(\hat{y}_i, y_i) + \sum_k \Omega(f_k)$$

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda ||w||^2$$

The first part of the objective function of the loss function and the second part is the regularization term which controlls the complexity of the model and prevents overfitting. And then use forward stepwise algorithm to optimize the objective function

4.3 multi-label Neural Network

Here we use sequence model in Neural Network which groups a linear stack of layers.We use tensorflow.keras to run Neural network with three layers here. For the first two layer, we use the rectified linear activation function and use sigmoid activation function for the last layer. And we also set three Dropout layer which randomly set input units to 0 with a frequency of rate at each step during training time. This could help prevent overfitting. For the inputs do not set to 0, they are scaled up by $\frac{1}{1-rate}$ such that the sum over all inputs is unchanged. Besides, before running each layer, we also reparameterizes the layer by decoupling the weight's magnitude and direction, which could speed up convergence by improving the conditioning of the optimization problem.[4]

4.4 Model Results

Here are the average errors computed for the three models mentioned above.

	Model	log-loss
1	multi-label logistics regression	0.03146
2	Xgboost	0.01885
3	multi-label neural network	0.01147

Table 1: The results of each method.

From the results of the three models, multi-label neural network gives us the best performance, then is XGBoost and the last is multilabel logistic regression.

5 Summary and conclusion

In this project, we fist do the Exploartory Data Analysis to explore our features. Based on our observation, we could transform the features to the best condition. For example, we transform gene expression data and cell viability data into 0-1 range. After understanding the distribution and correlation of these features, we do the feature engineering to select the features which are more useful and reduce the dimension of our covariate matrix. Based on

our problem, we try a few models and we found in this problem that the neural network method is the best. However, there is still a lot improvement could be done, such as using more complex neural network to increase the prediction accuracy or ensemble models to reduce the generalization error of the prediction.

6 Reference

1.Description from kaggle website: https://www.kaggle.com/c/lish-moa/overview/description
2.Chen, Tianqi; Guestrin, Carlos;"XGBoost: A Scalable Tree Boosting System",2016
3. PCA :https://www.mikulskibartosz.name/pca-how-to-choose-the-number-of-components
4. tensoflow : https://www.tensorflow.org/addons/api_docs/python/