Final Report: Case Study in Statistical Learning

Student: WANG, Chuning **Supervisor**: YAO, Yuan

ABSTRACT

This report applies statistical learning methods to studying the impact of combinations of four kinds of anticancer drugs (Vincristine, Mitoxantrone, Etoposide and Daunorubicin) against leukemia. Each medicine has four dosage levels, therefore there are totally 256 sets of data. We apply feature scaling and studentized residual to normalize the data. And various kinds of models are adopted to do feature selection, including quadratic regression, Ridge regression, Lasso regression, Multi-Layer Perceptron neural network (MLP), as well as Random Forest. The performances and stability of models combined with different normalization would be evaluated and compared with each other. We would analyze the results and interpret the models we get. We would also discuss some interesting phenomena as well as questions that worth further study. Moreover, some insights into the problem would be given by discussing the effectiveness and drawbacks of the models, and some possible improvements would be proposed.

I. Introduction

In recent years, the rapid development of machine learning has engaged the interest of people from various fields since machine learning has a wide range of applications due to its strong ability of pattern learning. For example, we could find out how the combination of different medicines would influence the cancer cell viability by

statistical learning. That could be quite a difficult problem for classical methods since there are quite a lot of features and the relationship between features and outcome may be complicated. Therefore, classical models may result in high variance or bias. During this research, I plan to study some general methods to process data as well as models in statistical learning, and apply them to studying cases and solving problems.

II. Methodology

i) Data Processing

As Fig.1 shows, the original dataset consists of the dosage levels of Vincristine, Mitoxantrone, Etoposide and Daunorubicin, which are medicines for treating leukemia, as well as the corresponding cancer cell viability after applying the medicines. Since each medicine has four dosage levels, there are totally 256 groups of data. To take the effect of the combination of two drugs into consideration, we select two of the variables with replacement and multiply them together thus create 10 new variables. The new dataset applied for training model consists of 14 features, which are denoted as V, M, E, D, VV, VM, MM, VE, ME, EE, VD, MD, ED, DD, and one outcome, which is cell viability.

Statistical learning methods are applied to our data normalized in two ways. One is studentized residual, normalizing each column of our data to that with zero as mean and one as variance. The other is feature scaling, normalizing each column of our data to the [0,1]-interval, where the minimum in the column is mapped to zero, and the maximum to one. Models without normalization was also tested.

Moreover, to examine the effectiveness of the models, 32 groups of data are randomly selected to serve as test data, else being training data. And we then randomly reselect testing data and repeat the rest process for nine times to compare the results and thus evaluate the stability of the statistical learning methods on this problem.

test#	vincristine	mitoxantrone	etoposide	daunorubicin	cell viability
1	0	0	0	0	100.00%
2	0	0	0	9.375	90.19%
3	0	0	0	18.75	92.65%
4	0	0	0	75	89.67%
5	0	0	12.5	0	83.22%
6	0	0	12.5	9.375	84.28%
7	0	0	12.5	18.75	88.63%
8	0	0	12.5	75	85.55%
9	0	0	25	0	91.36%

Figure 1(a): The First Nine Groups of Data of the Dataset.

249	50	40	25	0	61.76%
250	50	40	25	9.375	55.19%
251	50	40	25	18.75	56.80%
252	50	40	25	75	47.42%
253	50	40	100	0	54.63%
254	50	40	100	9.375	52.95%
255	50	40	100	18.75	55.08%
256	50	40	100	75	46.66%

Figure 1(b): The Last Eight Groups of Data of the Dataset.

ii) Ridge Regression

Instead of least square, ridge regression estimates coefficients $\hat{\beta}^R$ minimizing

$$\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij})^2 + \lambda \sum_{j=1}^{p}\beta_j^2,$$

where $\lambda \geq 0$ is a tuning parameter. Therefore, ridge regression tries to make residue sum of squares small, while having the effect of shrinking β_j towards zero. Compared to least squares estimate, ridge regression trades off a small increase in bias for a large decrease in variance (James, 2013, p215). It is obvious that different λ

would give different estimations of β_j, so we apply 8-folds cross-validation to training data to determine the value of λ minimizing mean squared error (MSE), and then take the value of β_j at this λ to evaluate the effect of 14 features. Negative β_j shows that the feature has positive effect on killing cancer cells, and lower the value β_j takes, better the efficacy of the feature is. Especially for the cross terms, such as VM, VE, positive β_j means that the two kinds of drugs reject each other, and negative β_j means that they cooperate well with each other on treating the cancer.

iii) Lasso Regression

Lasso regression is similar to ridge regression while using ℓ_1 norm instead of ℓ_2 norm, estimating coefficients $\hat{\beta}^L$ minimizing

$$\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij})^2 + \lambda \sum_{j=1}^{p}|\beta_j|,$$

where $\lambda \geq 0$ is a tuning parameter. While ridge regression is only able to shrink coefficients towards zero, but not equal to zero, ℓ_1 penalty would force more coefficients to be zero as λ increases. Therefore, we study not only the λ minimizing mean squared error (MSE), but also that gives a sparse model including only a subset of all the features with a little sacrifice of MSE. So lasso regression could perform variable selection, and the model we get may contain only a subset of features (James, 2013, p219). The way we interpret the result of Lasso model is similar to the case of Ridge regression.

iv) Multi-Layer Perceptron

As shown in Fig.2, a basic unit of a neural net called a perceptron consists of n-vector input, weighted summation, activation function, as well as output. And as shown in Fig.3, MLP consists of an input layer, one or more hidden layers, as well as an output layer. In this paper, we only study MLP with one hidden layer and sigmoid function is set to be activation function in every situation. While training MLP, back propagation algorithm is used to adjust weights to minimize the error $E = \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{d}\|_2$, where $\boldsymbol{y}$ is the output vector we get by calculation, and $\boldsymbol{d}$ is the target output vector. As a model-free estimator, MLP may be able to perform well while the relationship of features and output are sophisticated.

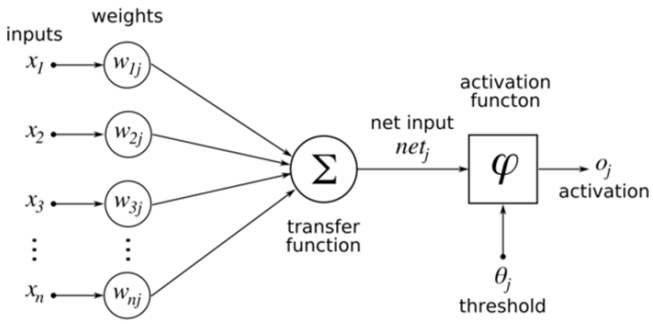

Figure 2: Structure of a Simple Perceptron (Artificial neural networks/activation functions - Wikibooks, open books for an open world, 2015).

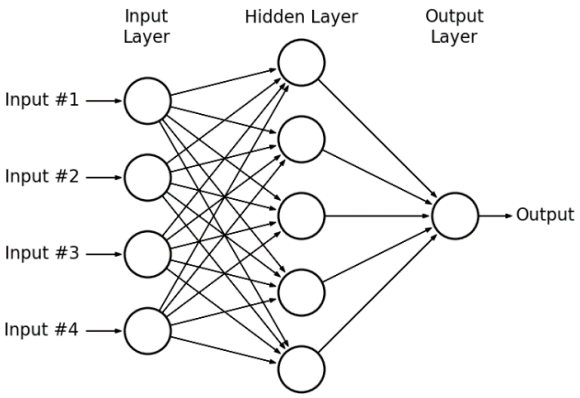

Figure 3: Structure of MLP with Single Hidden Layer (Minnaar, 2015).

v) Random Forest

Decision tree is the basic unit of random forest with its structure being like a flowchart. While constructing a decision tree, each split divides its input from a branch to two output branches according to an attribute. And each leaf would be considered as a class and combined with corresponding outcome.

While training our random forest model, many decision trees are constructed, each only taking into consideration a random subset of m features from the full set of features, or a random subset of all samples. In our model, 100 trees are constructed and the latter way of constructing subsets are adopted. Moreover, while making prediction, random forest takes the mean of predictions of each tree as output. This process may correct the possible overfitting of decision tree. Similar to MLP, if there is a highly non-linear and complex relationship between the features and the response, decision trees may outperform classical approaches. And random forests correct decision trees' habit of overfitting to their training set (Friedman, Hastie, and Tibshirani, 2009, p587-588).

III. Results

Different statistical learning models combining with different normalization methods give different results on evaluating the impact of features on decreasing cancer cell viability. For the methods developed from linear regression, since the dosage scale of different terms are different, and we interpret the coefficients of each term as their importance, normalization must be done. This statement can also be illustrated by our strange result of these models without normalization, which would not appear in our following discussion. Moreover, more focus would be on the results of Lasso regression and Random Forest, because Lasso regression may give a sparse model, and their accuracy and stability outperform other models as our analysis shows.

i) Quadratic Approximation

Quadratic approximation with feature scaling as normalization method suggests that V performs best, followed by M, E, VM, D. And that with studentized residual also suggests that V is more important than other terms, followed by E, D, M, VM.

ii) Ridge regression

Fig.4 suggests that while λ is chosen such that MSE is minimized, ridge regression with feature scaling suggests that V performs best, followed by VM, E, VV, EE. As for the ridge model with studentized residual, coefficients shown in Fig.5 shows that

V performs best, followed by E, D, M, and VM.

```
> ridge.coef
 (Intercept)            V            M            E            D           VV
 0.894796259 -0.118904130 -0.022453517 -0.063985423 -0.038184543 -0.047688197
          VM           MM           VE           ME           EE           VD
-0.093608825  0.028386012  0.001785041  0.009760485 -0.045140880 -0.033173053
          MD           ED           DD
-0.025496455 -0.013613778 -0.033360242
```

Figure 4: Coefficients of Ridge Model (Feature Scaling) for λ minimizing MSE.

```
> ridge.coef_norm
 (Intercept)            V            M            E            D           VV
 0.779363641 -0.075112064 -0.021882214 -0.034366658 -0.028071677 -0.005758138
          VM           MM           VE           ME           EE           VD
-0.012303492  0.011566293  0.004849164  0.003808414 -0.009140703 -0.001695916
          MD           ED           DD
-0.002785510  0.000247507 -0.009597225
```

Figure 5: Coefficients of Ridge Model (Studentized Residual) with MSE minimized.

iii) Lasso regression

In lasso model with feature scaling and λ minimizing MSE, as Fig.6 shows, V outperforms others, followed by M, E, VM, D. It is also worth noticing that both ridge and lasso regression suggest that in the domain of dosage, cell viability is even an increasing function of M when M is big enough. It is also an increasing function of VE and ME.

Moreover, as shown in Fig.8, the order that features enter model as λ decreases is V, VD, E, ED, VM, D, DD, EE, MD, MM, VE, ME, M, VV. It is worth noticing that as λ decreases, the coefficient of ED goes back to zero after the first time it becomes nonzero, and then enter the model again with small norm after every term were included in the model. Taking the largest value of λ such that MSE is within 1 standard error of the minimum, we get a sparse model (Fig.7) containing V, E, VM,

D, VD, DD, ED. This λ together with λ minimizing MSE and the coefficients at the two values of λ are also marked out by dashed lines in Fig.8.

```
> lasso.coef
  (Intercept)             V             M             E             D
 0.9403414994 -0.2972474493 -0.1524598017 -0.1248777109 -0.0594308222
           VV            VM            MM            VE            ME
 0.0919237851 -0.0874457688  0.1397286684  0.0330549898  0.0255761968
           EE            VD            MD            ED            DD
-0.0123819454 -0.0121481697 -0.0199385700  0.0005461514 -0.0292457390
```

Figure 6: Coefficients of Lasso Model (Feature Scaling) for λ Minimizing MSE.

```
> lasso.coef2
(Intercept)           V           M           E           D          VV          VM
 0.88342651 -0.16469825  0.00000000 -0.08572619 -0.04695900  0.00000000 -0.05952319
         MM          VE          ME          EE          VD          MD          ED
 0.00000000  0.00000000  0.00000000  0.00000000 -0.02416132  0.00000000 -0.01037125
         DD
-0.01336606
```

Figure 7: Coefficients of Lasso Model (Feature Scaling) for λ Giving Sparse Model.

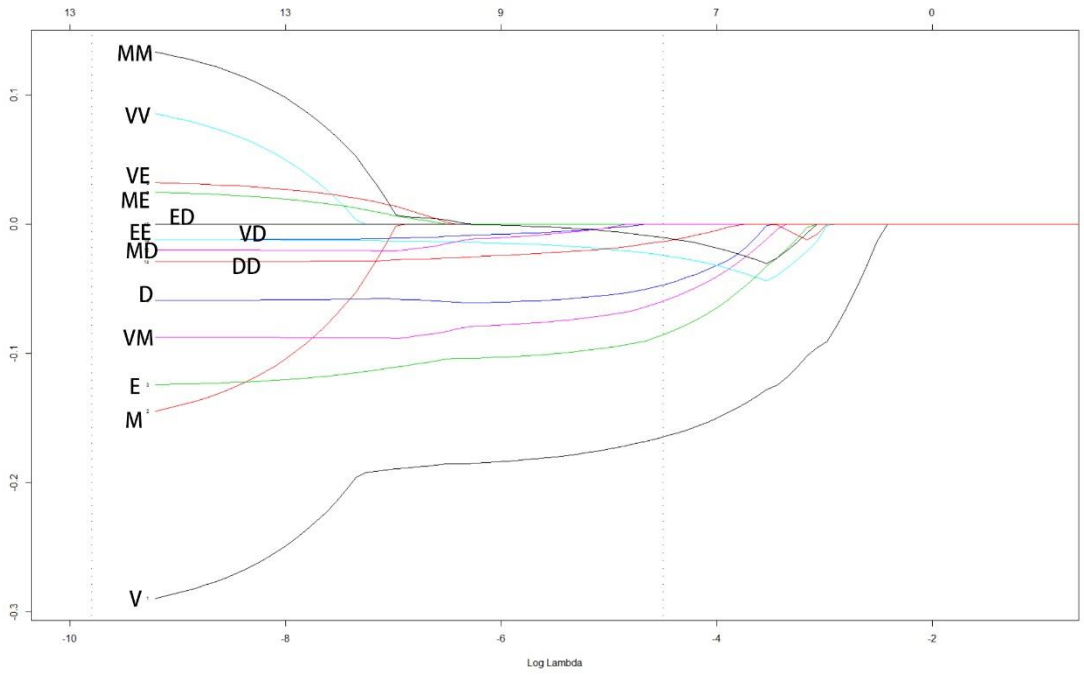

Figure 8: Coefficients with Respect to $\log\lambda$ (Lasso Model/ Feature Scaling)

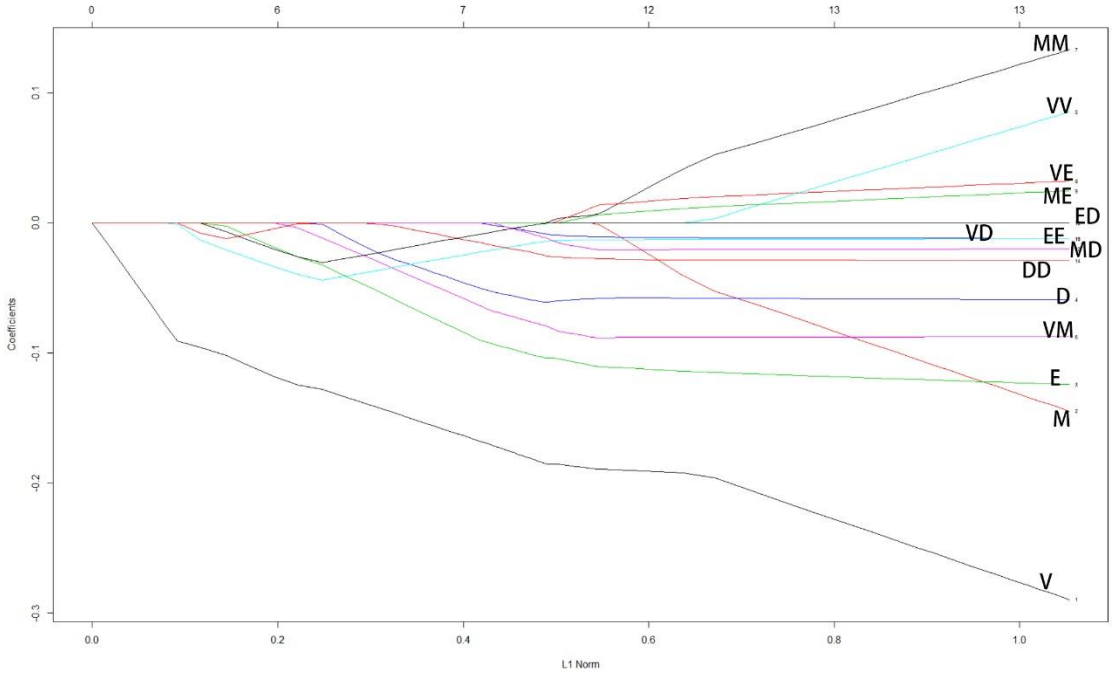

Figure 9: Coefficients with Respect to ℓ_1-norm (Lasso Model/ Feature Scaling)

While the inputs are normalized by studentized residual, the result with λ minimizing MSE is similar to that of the same situation for ridge model. Furthermore, the order that different terms enter the model, and their norms are shown in Fig.11. And as illustrated in Fig.10, the result with the largest value of λ such that MSE is within 1 standard error of the minimum gives a sparse model including V, E, D, M, VM, and DD, with importance decreasing. Similarly, in Fig.11, the two special λ are also marked out. Comparing Fig.8 and Fig.11, we can find that the results given by lasso model with two kinds of normalization are similar in some sense.

```
> lasso.coef2_norm
  (Intercept)             V             M             E             D            VV
 7.665468e-01 -7.730119e-02 -3.008762e-03 -3.669688e-02 -3.030528e-02  0.000000e+00
           VM            MM            VE            ME            EE            VD
-3.967139e-03  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00
           MD            ED            DD
 0.000000e+00  0.000000e+00 -6.269803e-05
```

Figure 10: Coefficients of Lasso Model (Studentized Residual) for the Sparse Model.

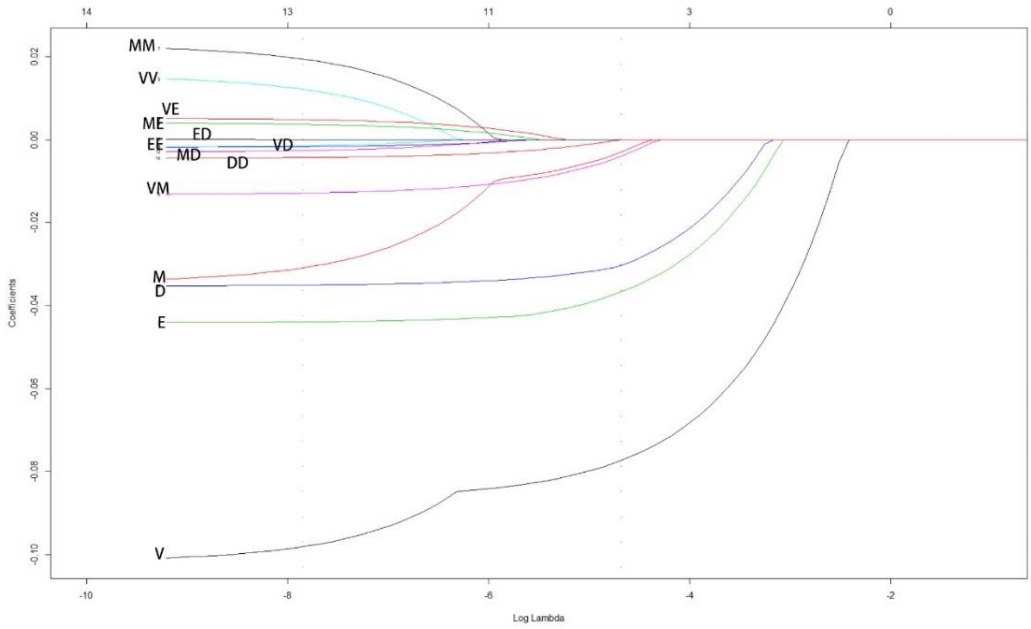

Figure 11: Coefficients with Respect to $\log \lambda$ (Lasso Model/ Studentized Residual)

iv) Multi-Layer Perceptron

Due to the restriction of the algorithm of MLP, normalization is also a must. Fig.12 suggests the importance of inputs of our trained MLP with feature scaling estimated by Garson's Algorithm. On the other hand, while training MLP with input normalized by studentized residual, we get that V performs far better than all the other terms.

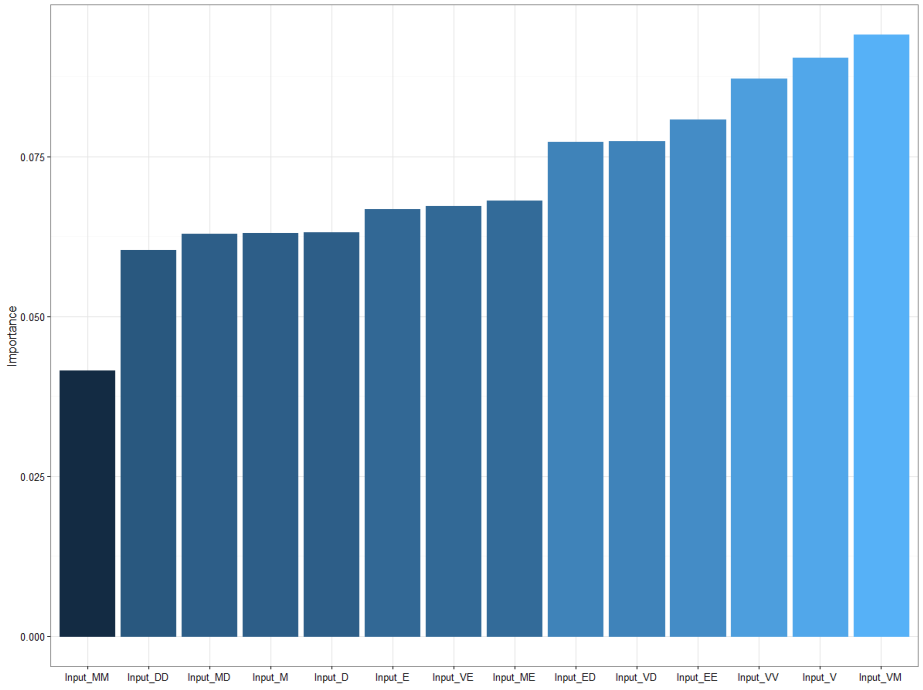

Figure 12: The Importance of Inputs in MLP with Feature Scaling.

v) Random Forest

While training random forest, model without normalization would give the same result as that with feature scaling. Fig.13 and Fig.14 respectively show the importance of features in random forest with feature scaling and studentized residual, more important the feature is, higher the value would be. V, VE and VV outperforms other terms in both situations. It is worth noticing that high importance does not mean that the efficacy is better. It is also possible that this term has significant negative effect.

```
> round(importance(RF.mod_0_1), 2)
   %IncMSE IncNodePurity
V     8.10          0.53
M     6.51          0.08
E     6.47          0.13
D     4.92          0.08
VV   10.11          0.62
VM    5.54          0.25
MM    5.37          0.08
VE   11.26          0.42
ME    7.57          0.11
EE    7.26          0.14
VD    8.34          0.37
MD    2.58          0.07
ED    4.73          0.18
DD    6.05          0.09
```

```
> round(importance(RF.mod_norm), 2)
   %IncMSE IncNodePurity
V     9.12          0.69
M     7.73          0.09
E     4.90          0.14
D     4.62          0.12
VV    7.39          0.54
VM    4.72          0.19
MM    3.74          0.05
VE    7.20          0.36
ME    3.22          0.12
EE    5.80          0.16
VD    5.84          0.36
MD    4.24          0.09
ED    6.56          0.20
DD    3.62          0.08
```

Figure 13: The Importance of Features in Random Forest with Feature Scaling.

Figure 14: The Importance of Features in Random Forest with Studentized Residual.

vi) Performance Analysis

We measure the performance of different models by calculating the MSE of test data. As Fig.15 and Fig.16 shows, quadratic regression performs worst, while lasso regression and random forest outperform other models.

```
> qr.MSE
[1] 0.02724675
> ridge.MSE
[1] 0.004061788
> lasso.MSE
[1] 0.003252355
> mlp.MSE
[1] 0.004595737
> randfore.MSE
[1] 0.003298203
```

Figure 15: MSE of Testing Data of Models with Feature Scaling.

```
> qr.MSE_norm
[1] 0.02724675
> ridge.MSE_norm
[1] 0.003640341
> lasso.MSE_norm
[1] 0.003260593
> mlp.MSE_norm
[1] 0.003995793
> randfore.MSE_norm
[1] 0.003471364
```

Figure 16: MSE of Testing Data of Models with Studentized Residual.

The stability of the models could be discussed based on the results given by running the similar process for ten times with the only difference being the random reselection of testing data. As illustrated before, more focus would be on the results of Lasso regression and Random Forest, therefore we only discuss the stability of these two models here. Retraining Random Forest with feature scaling for ten times, we get that considering the importance of the terms, VE, VV, VD, and V always outperforms other terms. And as Fig.17 shows, MSE are always small. As for Lasso models, the models at λ minimizing MSE always give that V, M, E, VM, and D performs better than others. The coefficients of VE and ME are always positive. And the coefficient of MM is always about 0.13 and suggests that if we give too much Mitoxantrone, cell viability may even be increasing. Furthermore, by Fig.18 we may know that the

features selected by the ten sparse models are also almost same. As for the stability of Lasso Model and Random Forest with studentized residual, the results give conclusion similar to our discussed condition, so the analysis is omitted here.

```
> randfore.MSE_0_1
     randfore.MSE
[1,]   0.00335316 0.001930076 0.002105391 0.002817019 0.002043166 0.004116592

[1,] 0.002160188 0.002329224 0.002987409 0.002805684
```

Figure 17: MSE of Testing Data of the 10 Random Forest models (Feature Scaling).

```
> gp.lasso.coef2
             gp.lasso.coef2   lasso.coef2 lasso.coef2   lasso.coef2  lasso.coef2
(Intercept)    0.888677076  8.876770e-01  0.88091407  0.8894077468  0.894300535
V             -0.168540161 -1.677863e-01 -0.16286854 -0.1690967072 -0.173033531
M              0.000000000  0.000000e+00  0.00000000  0.0000000000  0.000000000
E             -0.090925997 -9.020054e-02 -0.08307427 -0.0913897121 -0.094798284
D             -0.050802153 -5.006933e-02 -0.04512447 -0.0513376375 -0.054291848
VV             0.000000000  0.000000e+00  0.00000000  0.0000000000  0.000000000
VM            -0.064557505 -6.357106e-02 -0.05712957 -0.0652852997 -0.069399860
MM             0.000000000  0.000000e+00  0.00000000  0.0000000000  0.000000000
VE             0.000000000  0.000000e+00  0.00000000  0.0000000000  0.000000000
ME             0.000000000  0.000000e+00  0.00000000  0.0000000000  0.000000000
EE            -0.000372017 -2.544378e-05  0.00000000 -0.0006935732 -0.002539165
VD            -0.022089277 -2.249641e-02 -0.02515045 -0.0217885304 -0.019834384
MD             0.000000000  0.000000e+00  0.00000000  0.0000000000 -0.001766849
ED            -0.008284460 -8.693968e-03 -0.01135890 -0.0079829079 -0.005980649
DD            -0.015657695 -1.518948e-02 -0.01228269 -0.0160078743 -0.018335573
             lasso.coef2 lasso.coef2   lasso.coef2   lasso.coef2   lasso.coef2
(Intercept)  0.8892262694  0.88349339  8.871572e-01  0.8890457673  8.869615e-01
V           -0.1689584772 -0.16474693 -1.674092e-01 -0.1688209902 -1.672672e-01
M            0.0000000000  0.00000000  0.000000e+00  0.0000000000  0.000000e+00
E           -0.0912745386 -0.08579672 -8.965414e-02 -0.0911599841 -8.944844e-02
D           -0.0512046386 -0.04700789 -4.968866e-02 -0.0510723546 -4.954535e-02
VV           0.0000000000  0.00000000  0.000000e+00  0.0000000000  0.000000e+00
VM          -0.0651045364 -0.05958696 -6.307649e-02 -0.0649247447 -6.289030e-02
MM           0.0000000000  0.00000000  0.000000e+00  0.0000000000  0.000000e+00
VE           0.0000000000  0.00000000  0.000000e+00  0.0000000000  0.000000e+00
ME           0.0000000000  0.00000000  0.000000e+00  0.0000000000  0.000000e+00
EE          -0.0006137079  0.00000000 -1.989865e-05 -0.0005342719 -1.781102e-05
VD          -0.0218632273 -0.02413505 -2.269959e-02 -0.0219375227 -2.277608e-02
MD           0.0000000000  0.00000000  0.000000e+00  0.0000000000  0.000000e+00
ED          -0.0080578048 -0.01034514 -8.900760e-03 -0.0081322992 -8.978612e-03
DD          -0.0159208998 -0.01339481 -1.496681e-02 -0.0158343927 -1.488299e-02
```

Figure 18: The 10 Sparse Models Given by Lasso Regression (Feature Scaling).

IV. Conclusion

As our results show, random forest and lasso regression performs quite well at capturing the pattern of the impact of combination of medicines on cell viability. The

stability of models is also high. The combination of V with other three medicines, especially VM, may be candidates for more effective therapy against leukemia. Moreover, we could notice that the coefficient of VE and ME are slightly bigger than zero in both of the Ridge and Lasso models, which may suggest that the combination of Etoposide with Mitoxantrone or Vincristine may not work well. And the excessive use of Mitoxantrone may result in lower efficacy on treating leukemia. Our remaining problem is that the important features suggested by different model combining with different normalization are not always the same. As far as I am concerned, we may improve the data processing and learning methods for this problem in further study. For instance, we may construct models taking only cross terms as additional inputs, excluding VV, EE, MM, and DD, we may even do higher order regression. We could also combine lasso and ridge regression together, using both ℓ_1 and ℓ_2 penalty at the same time. There are also many interesting phenomena of our results worth exploring, such as the variation of the coefficient of ED as λ decreases in Lasso model. We may also apply the methods to new problems to study how they works in the future.

REFERENCES

Artificial neural networks/activation functions - Wikibooks, open books for an open world (2015) Available at: https://en.wikibooks.org/wiki/Artificial_Neural_Networks/Activation_Functions (Accessed: 23 October 2016).

Friedman, J., Hastie, T. and Tibshirani, R. (2009) The elements of statistical learning: Data mining, inference, and prediction, Second edition - 2nd edition. 2nd edn. New York, NY: Springer-Verlag New York.

James, G. (2013) An introduction to statistical learning with applications in R. New York, NY: Springer.

Minnaar, A. (2015) Implementing the DistBelief deep neural network training framework with Akka. Available at: http://alexminnaar.com/implementing-the-distbelief-deep-neural-network-training-framework-with-akka.html (Accessed: 23 October 2016).