Advantages and Disadvantages of SVM and NRWRH in Drug-gene Interaction Prediction

CHENG Chen
School of Life Sciences and Biotechnology, Shanghai Jiao Tong University, Shanghai, China

ABSTRACT: This report is about using the similarity of drugs or genes to predict possible interaction between drug and gene. I use SVM and NRWRH in different condition. And then analyze the advantage and disadvantage of them, respective. I also propose some factors which can impact those results.
KEYWORD: SVM; NRWRH; Drug-gene; Interaction Prediction

1 INTRODUCTION

In this lab, my job is to predict the possible interaction between the drugs and genes. The most important things in my work are SVM and NRWRH. The major tools which I used are R, matlab and python.

The first thing I done is divide the problem into two cases. The first case is that we have known some interactions between every test drug and some genes, and we want to find some new interaction of this drug. In this case, I use NRWRH (Network-based Random Walk with Restart on the Heterogeneous network). The second case is we don't know any interaction among this drug and genes, we only know the similarity between this drug and others, and the similarity among genes. In this case, I use SVM (Support Vector Machine)

The first time that SVM was proposed in 1995, and then it became very popular quickly. Specially in those years, most of paper which involve classification will mention it. It is good at deal with the small sample, non-linear and high dimensionality problem. It also can easy be popularize in other machine learning problems. SVM care about the VC dimensionality of question but not normal dimensionality. In SVM, the linear classifier is the most simple and useful.

I will use a 2 dimensionality question as a example. As figure 1, the cycle and square can be classify by a line. We can solve this question by find the function:

$$g(x) = wx + b$$

In other words, this problem is a linearly separable problem. Obviously, the H is not single, H can move between H1 and H2. H1 and H2 are parallel with H and throw the nearest point of H. The distance of H1 and H2 is bigger, the upper limit of error is smaller. And the biggest distance means the smallest ||w||:

$$\| w \|_p = \sqrt[p]{w_1^p + w_2^p + \ldots + w_n^p}$$

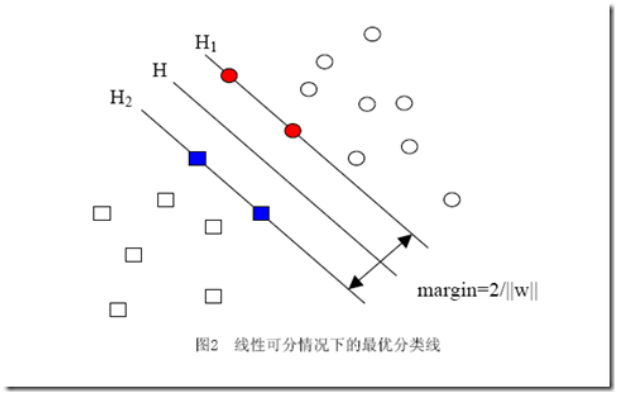

Fig.1 linear separable problem

If we get the w, calculate b is not a big problem.

If the problem is linear non-separable, we can change it to a linear separate question by using dimensionality raising. A simple example, like figure 2, in 1-D world, red line and blue line are two different class, we can't separate them by a point. If we transform it to a quadratic function, then we can separate them by a line. We transform a non-separable question to a separable question.

The core of NRWRH is RWR (random walk with restart). This algorithm include four steps: firstly, three networks (protein-protein similarity network, drug-drug similarity network, and known drug-target interaction network) are constructed and combined

into a heterogeneous network by known drug-target interactions; secondly, the initial probability of random walk is determined to make random walk start at the given drug nodes and seed target nodes simultaneously; then random walk on the heterogeneous network is implemented; finally the most probable targets are selected according to the stable probability of the walk.

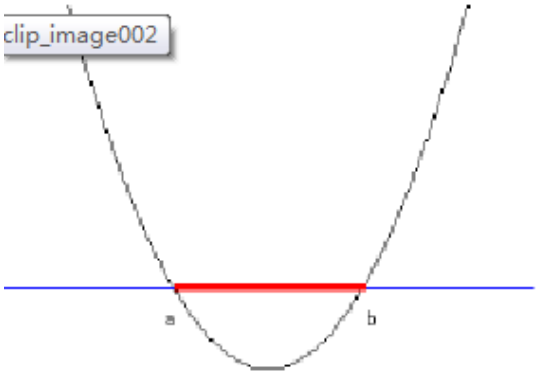

Fig.2 Transform a non-seperable question to a seperable question by dimension raising

2 METHODS

In SVM, I main used R and python. And in NRWRH, I main used matlab and python. Some codes of NRWRH come from internet, the details can be found in readme.tx

2.1 *SVM*

When we don't know any interaction about the drugs which we will predict, I choice SVM. In this SVM model, every node representative a relationship between a drug and a gene. When this node is belong to class 0, this means that this drug and gene haven't got a interaction, and if it belong to 1, it means that this drug and gene have a interaction. The characteristics of node is this drug's similarity with other drugs and this gene's similarity with other genes. For example, if the number of genes are m, the number of drug are n, the number of nodes are m*n, the number of characteristics of every node are m+n .

2.1.1 *Calculate basic information*

At the beginning, I combine three matrices (drug_similarity, gene_similarity, adjacency_matric) into one big matric which have m*n lines and m+n+1 columns. Every row represents a relationship between a drug and a gene. The last column is this relationship's class (0 or 1). And then take a 10-fold cross-validation. The source code is in folder svm.

And then using the characteristics and SVM model, a prediction is made. The details can be found in svm.r and readme.txt.

The output is not a series of integer because R not have a int. We will get a series of decimals such as 0.0125467(belong class 0) or 0.9993261(belong class 1). But I must mention that the bound of those two classes are clear. 99% data of 0-class are included in range (-0.1,0.1). 99% data of 1-class are include in (0.9,1). So although we need reprocess the output so that we can know which class the data is belong, this step wouldn't bring in error.

2.1.2 *Dependency of data scale*

I find that a matrix of $(m\times n) \times (m+n+1)$ is a really big matrix, and this algorithm is time-consuming. So I am interested in whether we really need so much data. Of cause, in most conditions, more data to build the SVM model can lead to more accurate results. But sometimes less data can save lots of time, and its accuracy is also can be accepted. I want to know how big the data is can lead to the critical accuracy.

So I change the scale ratio of training data and test data. The ratio are 8:2, 7:3, 6:4, and 5:5, respective.

2.1.3 *Stability*

Drugs and genes are all have lots of data. Our data only divide the interaction into two classes------have interaction or not, those data not distinguish the strength of interaction. It is normal and can't be avoided that some data are wrong. So it is necessary to study the model's stability. On the other words, if some data are wrong, whether we can still get the accuracy result.

I change the 1%, 3%, 5%, 10%, and 20% of the training data, respectively.

In the data, scale of 0-class and of 1-class are extremely imbalance. 0-class is overwhelmingly. I believe that this can decrease the result's accuracy and stability. So I do more experiments about them.

I choice the hole 1-class and random choice same scale 0-class data. Make them disorganize. And do the same things as 2.1.1, 2.1.2 and 2.1.3. Compare the results.

2.2 *NRWRH*

In this part, I use some codes which are from internet because I find the same work. The details can be found in references.

NRWRH is to predict the new drug–gene interaction by the interactions we have been know. So if we use the source data directly, we can't evaluate the results' accuracy. So I change 10% of

the drugs' data, every drug change one interaction or three interactions, just put 1 become 0. Then we predict the new interaction of this drug, and see if the predicted interaction is the one which we delete.

NRWRH's output is the rank of gene. If the serial number is 1 that means this gene is the most likely one which have interaction with this drug. If we only run the function 1 time (only predict one drug), the output not include the genes which we have already known that they have interactions with this gene, on the other word, the output only rank the predict genes. Here I have a hypothesis that the gene I delete is more likely have a interaction with this drug then the genes which we don't know whether they have interactions with this drug. That means I believe that the genes whose serial number is first or in the top three include the gene whose data have been changed.

NRWRH have eight parameters. One of them is drug's ID, another 4 parameters represent the weight of genes' similar and drugs' similar. I think the importance of genes' similar and drugs' similar are equal, so I setting all 4 parameters as 0.5. The last three parameters all are a number between 0 and 1, we need lots of experiments to make sure which one is better. But every group, I do no more then 100 times because of the limitation of personal computer and time. This can largely influence the accuracy of the results.

3 RESULT AND DISCUSSION

3.1 *The result of SVM*

The results' accuracy, stability and so on is not satisfy my expectation.

3.1.1 *Calculate basic information*

The 10-flod cross-validation's results are very good when I first saw them. The mean accuracy is very high, about 0.967. ROC(figure 3) is very close to the top left corner. AUC can achieve 0.964.(ROC is made by SPSS)

But when we analyze the data more, we will findthat the result has a problem. I use frequency instead of probability. If the true value is 0, the probability we predict it as 0 is 0.999274. If we get a 0, the probability that this is true value is 0.967, if we get a 1, the probability that this is true value is 0.83. But if the true value is 1, the probability we predict it as 1 is only 0.093.

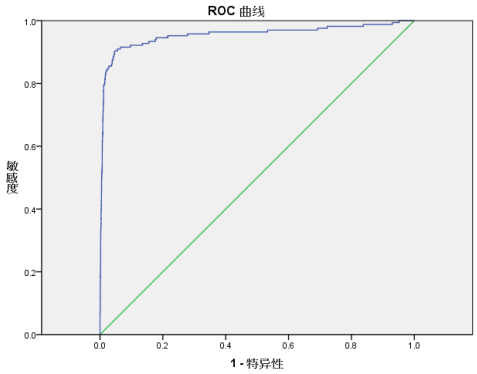

Fig.3 one of 10-flod cross-validation's ROC,the AUC is 0.955, the top and buttom limitaion, in this time, is 0.978and 0.932 respecially.

There major reason why we are difficult predict 1 as 1 is that the scale of 0-class and 1-class is differentiating. I will analyze this in the latter part.

To sum up, although when a drug and a gene have interaction, we are difficult correctly predict it, the prediction's result is very accuracy.

3.1.2 *Dependency of data scale*

Analyze the different group's data (table 1), we find the data fluctuate very little. We using ANOVA to analyze the difference between, the result shows that all this group's data are same. The little fluctuate is caused only by the random error.

There are two possible reasons. First, SVM not very rely on the scale of data, we don't need so many data to build the model. Second is the scale of 0 and 1 is so different that it influences result much more than the scale of training group. I will talk about this more in the latter part.

Table.1 0 pre is the accuracy of the data which true value is 0. Pre 0 is the accuracy of a predict value is 0.

	9:1	8:2	7:3	6:4	5:5
accuracy	0.967	0.967	0.968	0.968	0.967
0 pre	1	1	11	1	1
1 pre	0.092	0.083	0.077	0.070	0.054
Pre 0	0.967	0.967	0.968	0.967	0.967
Pre 1	0.833	0.823	0.867	0.949	0.949

3.1.3 *Stability*

After I change the data, the accuracy of all data and 0-class is not change. It is amazing that the accuracy of 1-class improve.

I think the major reason is that the changed data decrease the difference between the 0-class scale and 1-class scale. But the 0-class still predominate. So the data scale influences the result most, the influence of data error is covered up (table 2). We don't know which one is the major reason now, the study in next section can help us understand this more.

When I change the 20% of the data, R run more than 3 hours but still not have a result. This is likely because too many data changed lead to the problem non-separable.

Table.2 test SVM's stability

	0	0.01	0.03	0.05	0.1
accuracy	0.967	0.965	0.965	0.967	0.967
0 pre	0.999274	0.999271	0.999271	0.999029	0.999271491
1 pre	0.092002	0.108434	0.126506	0.168675	0.156626506
Pre 0	0.967331	0.96406	0.964789	0.966489	0.966002914
Pre 1	0.833034	0.857143	0.875	0.875	0.962962963

3.1.4 *Balance*

Just like table 3, when the scale of 0-class and 1-class is equal, the accuracy of 1 is increased greatly. There are two reasons. First, the scale of data is decreased, the number of characteristics of nodes is increased relative to the number of nodes. Second, the different classes with equal scale can help us get a better result. This tells us that don't add useless data blind although in most condition, more data lead to better result. If we want to predict more accurate, we need improve the ratio of 1-class.

In this section, when we have some wrong data, the accuracy of result decrease. Compare with the non-balance section, the amount of accuracy decrease is bigger, but the stability is still very high. We can find the details in the table 3 .SVM is a stability algorithm

Table.3 0 pre is the accuracy of the data which true value is 0. Pre 0 is the accuracy of a predict value is 0

balance		0.01	0.03	0.05
accuracy	0.88276	0.89310	0.8793103	0.86552
0 predict	0.910477	0.863014	0.856164384	0.849315
1predict	0.916644	0.923611	0.902777778	0.881944
Predict 0	0.846322	0.924658	0.904109589	0.883562
Predict 1	0.916644	0.869281	0.860927152	0.852349

Table.4 0 pre is the accuracy of the data which true value is 0. Pre 0 is the accuracy of a predict value is 0

	9,1	8,2	7,3
Accuracy	0.882759	0.874137931	0.86092
0 predict	0.910477	0.861872092	0.822525
1 predict	0.916644	0.886414267	0.896015
Predict 0	0.846322	0.885998091	0.8874
Predict 1	0.916644	0.866109189	0.845656

Just like table 4, I change the ratio of training group's scale and testing group's scale. Compare with the non-balance group, with the ratio which between training group and test group decrease, the accuracy decrease quicker and more regular. This can support the conclusion that non-balance group's data not change is a result of uneven data scale. SVM has a requirement of training data's scale.

3.2 *The result of NRWRH*

NRWRH's result is not ideal. The first group I only predict one interaction for every drug. When the first parameter is 0.7, we can get 2 correct answer for every 22 drugs. The second and third parameters change not lead to the result change. The details can find in table S1. When I predict 3 interactions for every drug, if the first parameter is 0.4, and the second is 0.1, every 22 drugs, I can get 4 drugs each of which has one correct prediction. When the first parameter is 0.7, every 22 drugs, I get 2 drugs each of which has one correct prediction, and 2 drugs each of which has 2 correct predictions.

In my view, it is two things lead to this result. First, when the first gene (or top 3) is not the gene which we delete before, we think it is wrong. But it is really possible that a gene, which we don't know having a stronger interaction with the drug, is more likely having interaction with the drug than the gene, which we know having interaction. Second, I don't test enough of the parameters because of the limitation of personal computer and time. Those parameters maybe not fit.

4 CONCLUSION

SVM in most condition has a high quality result but it is time-consuming. And when the scale of classes is not uniform, the effect of the algorithm will be affected. When we use SVM, it is better that the number of characteristics is more than the number of nodes. So it needs a large number of data.

When we use NRWRH to predict a drug, we'd better know some of this drug's interactions with other genes. If not, the results will be terrible.

REFERENCES

[1] Chen, X., Liu, M. X., & Yan, G. Y. 2012. *Drug–target interaction prediction by random walk on the heterogeneous network. Molecular BioSystems, 8(7), 1970-1978*