CS230

IDG-DREAM Drug-Kinase Binding Prediction

Mayukh Majumdar
Stanford University
maymaj@stanford.edu

Abstract

Protein kinase are kinase enzymes that modify proteins by chemically adding phosphate groups to them. 30 percent of all human proteins maybe modified by kinase activity. This provides a fertile ground for drug interventions to treat diseases. Discovering these interactions using clinical trials is very long and expensive. A search for using AI techniques to predict affinity of protein kinases with drug compounds based on data already available is of extensive interest. A method to use embeddings, a well-known AI technique, to predict the KD affinity between drug compounds and protein kinases is presented. The technique fits quite intuitively with the problem statement and ground affnity values are derived with a fairly basic neural network.

1 Introduction

Protein kinase are kinase enzymes that modify proteins by chemically adding phosphate groups to them. Human genome has 518 protein kinase genes. 30 percent of all human proteins maybe modified by kinase activity. So protein kinases provide a fertile ground for experiments with drug interactions.

Mapping the complete spectrum of potential interactions between compounds and their targets, both intended "primary" targets and "secondary" targets, is extremely critical part of drug discovery and development efforts. This will enable the exploration of the therapeutic potential of these chemical agents as well as a better understanding and management of any possible adverse reactions. More and more data indicates that most drugs bind to more than one target molecule within a biologically relevant affinity range.

Given the above, an open challenge has been presented to the AI researchers around the world focussing on protein kinase inhibitors - the "illuminating the Druggable Genome(IDG)-DREAM Drug Kinase Binding Prediction Challenge. This project is attempting to provide a prediction for the affinity between different drug compounds and protein kinases based on some of the affinities available in the DrugTargetCommon (DTC) database. This deep learning-based prediction model will, hopefully, be able to provide a systematic way to prioritize the most potent interaction for further physical trials – greatly reducing time for drug discovery.

2 Related work

The drug-kinase binding prediction has mostly been attempted from the structural aspects of the protein by performing clinical trials to collect data[1][2]. This has obviously been a long and ardous process. The attempt of this Challenge has been to find prediction models that can be developed using AI analysis of the clinical data observed so far. In this paper, we have tried to showcase the problem as an embedding problem such that the binding predictions are available in the embedding matrix.

CS230: Deep Learning, Winter 2018, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

In the context of embeddings, the most well known algorithms are the Word2Vec[3][9] and GLoVe[4] algorithms. These algorithms have become the basis of recommender systems in industry, the largest examples of which are the ones used at Netflix[5] and Amazon[6]. These recommendation systems are being used to "recommend" to viewers which movies they might want to see next based on their viewing history. No references were found on previous attempts to use the recommendation systems for drug-kinase binding prediction.

Though formulating the binding prediction problem akin to such a recommendation system seems new, it seemed very intuitive and provides the predictions we are looking for.

3 Dataset and Features

For the training data collection, the Challenge makes use of an open-data web-platform, DrugTarget-Commons(DTC), availble at *https://drugtargetcommons.fimm.fi*[8]. DrugTargetCommon(DTC) is a comprehensive resource to provide the training data for bioactivity annotated for various compound-target interactions and a wide range of end-point measurements. We are focussing on Kd measurements between drug compounds and protein kinases. The total database contains 1746997 compounds and 13,023 protein targets.

Since the initial database had an extremely large size of 1.6GB, we focussed on the KD measurements only as that was the main aim of the prediction challenge. After parsing the complete database for relevant measurements, the final dataset had KD affinity values for drug compound - protein kinase pairs with 13368 unique drug compounds and 1421 unique protein kinases.

A quickly hacked up R program by David provided some excellent insight into the data given. The most frequent activity types seemed to have a good correlation. We also compared the activity measures and found high correlation between activity types. Given this finding, he suggested that a multi-task approach should work as well. The code used to provide this insight is referred to in the results section of the paper.

4 Methods

The essential way to look at these affinities between drug-kinase pairs was to be analogous to the movie-user pairs – and so techniques used for recommender systems were a fairly natural fit here. The recommendation systems used in Netflix and Amazon for the movie shows we see was studied and found to be called collaborative filtering in deep learning circles.

Given the underlying alogorithms of the recommender systems were embeddings, it was decided to extract an embeddings from the data provided to have a better representation. In traditional statistics, these embeddings would be created using Princial Component Analysis (PCA) while in this case, the focus was on using the deep learning technique of embeddings to derive the same.

To extract the embeddings, the architecture selected was to have a simple 2-layer neural network with the following features :

1. *Input Layer* : The neural network could provide embeddings for the protein given the drugs or the embeddings for the drugs given the proteins. If drugs were chosen to be the input, there were 13368 nodes in the input layer and each training input vector was one-hot for the 13368 drug compounds. On the other hand, if the proteins were chosen to be the input, there were 1421 nodes in the input layer and each training input vector was one-hot for the 1421 nodes in the input layer.

2. *The hidden layer* : The next layer was the hidden layer which would be learning the embeddings for the neyral network. It was decided to make this a 100 node hidden layer, which seemed a good number to start with. This hidden layer was also going to be a fully connected (FC) layer.

3. *The output layer* : The output layer was to be decided based on whether this network was learning embeddings for proteins given drugs or for drugs given proteins. If the network was learning embeddings for proteins given drugs, there were going to be 1421 nodes in the output layer, each of which would be a sigmoid. This was chosen so that we could get continuous values of output since we needed continuous values to get the precise KD affinity value being predicted. If the network was

learning embeddings for drugs given proteins, there were going to be 13368 nodes in the output layer, each of which would also be sigmoid.

4. *Training vectors* : The final dataset was converted into data structures that provided drug2protein and protein2drug mappings. The drug2protein mapping provided KD affinity vectors which had non-zero affinity values for each protein that had a value provided in the database while all other values in the vector were zero. The vectors to train the network were derived in this fashion. The input vectors were one-hot for each of the unique drug compounds and the training output vector was the affinity vector described above which had non-zero values for affinities provided for the drug compound in question to various proteins while all the other values in the vector were zero. Since the data structures were able to provide the protein2drug mappings as well, similar training vectors were generated which carried the one-hot vectors for proteins as input and had the affinity vectors as output vectors where the non-zero values were the affinity values for known drug compounds for the protein in question and all the other values were non-zero.

5. *Loss function* : Since the predicted values needed to be continuous and not a result of classification, the loss function selected was for regressions such that the output values could take continuous values as well. In our case, we selected the L2 Loss function, which is also called the Mean Square Error (MSE), given by MSE = $1/n(Sumoveralli[(yi - yip)^2])$

6. *Final values* : Once the training of the embeddings was completed, getting the predictions for the KD affinity values would be fairly simple. In the case of the KD prediction for a protein given drug, a one-hot vector for the drug compound would be applied at the input layer. The output layer would provide a vector of 1421 elements. Some of these elements would match the affinity values that were in the training set, while all the other elements would be the corresponding predicted KD affinity value for each of the proteins given the input drug compound. This KD affinity value provided by the embeddings in the neural network would be the ground truth value for that drug-protein pair. A similar result would be obtained if we performed the experiment trying to find drugs given the proteins.

5 Experiments/Results/Discussion

The overall results were obtained for both the mappings - protein given the drug as well as drug given the protein - are given below. In both the cases, the 13368 input dug compounds were divided into a training set using 90 percent of the values available and the development set were the 10 percent remaining. The test set of 430 drug-protein pairs were already provided. The number of epochs for the training were chosen to be 10.

Model	Train RMSE	Train Samples	Test RMSE	Test Samples
Drug2protein	0.6823	12031	1.835	430
Protein2drug	0.8659	12031	1.746	430

There were several issues that need to be discussed regarding these results :

1. Input data set : As we mentioned, we created both the drug2protein and the protein2drug mappings from the input dataset. There were some peculiarities noticed in the data patterns arising from the fact that the number of drug compounds were about 10X the number of proteins. So in the drug2protein mappings, most of the drugs had 1:1 mappings with proteins while a few had upto 5 proteins mapped to each drug. Yet, there were some drugs that seemed very prevalent and had affinities with 40+ proteins. They definitely seemed powerful drugs. On the other hand, the distribution of the protein2drug was more even and all proteins had about 30 - 70 drugs they seemed to have affinity with. Some proteins remained with a 1:1 relationship with drugs as well.

2. One of the features of the way the algorithm was chosen was to make sure that we got ground truth values for the KD affinity predictions that were required. There was a discussion on

normalization that was raised, but it was not used in the final algorithm. The main reason was pointed out by several TAs was that once we normalized the input values, we would actually have a softmax output instead of the sigmoid outputs and we would not be able to use the output values directly. We would possibly have to use another neural network to take the softmax outputs generated from the output layer and then provide a way to generate the ground truth. That would definitely be a possible architecture, but given the time constraints, it was decided to do without normalization. There could be at least 2 kinds of normalization used : 1. normalize affinity of each drug compound or protein such that the sum of all would be 1. 2. normalize all the KD values to a standard normal distribution.

3. The L2 loss function was chosen as it was a continuous loss function for regression. Yet, it is widely known that the L2 loss function suffers from outlier values. The range of KD affinity values seen was large from 0.008 to 70000. So the results are probably affected by outliers. We need to check results using the HUber Loss or the Log-Cosh Loss functions to overcome the weaknesses of the L2 loss function.

The code for this experiment is being shared at : "https://github.com/maymaj/CS230_Fall_2018_project2"

The code that David wrote to find correlations in the data is at : "https://github.com/davidaknowles/idg_dream"

6 Conclusion/Future Work

Overall, the embeddings idea worked almost intuitively with the problem statement here. The best part of the exercise was that the ground truth KD affinity values seemed to just fall out of the basic embeddings generated. That seemed to be the greatest success of this exercise.

In terms of future work, we could :

1. Try the normalization options that have already been discussed and try to add the neural network to generate the ground truth outputs if required from the softmax outputs.

2. Try to apply other algorithms of the recommendation system like Collaborative Filtering to see if we can achieve better results.

3. Last, but not the least, the main issue noticed with the dataset seemed to be the fact that a lot of the data that could not be used were because of affinity readings on measure other than KD. There seem to be at least 10 other measures possible between drugs and proteins. We have just shown that embeddings are fairly easy to apply to any one of these measure. Maybe if we derive embeddings for all of these measures and then create a network that can use the embeddings from all these measures, the prediction of drug-protein association would be really very strong and may provide really meaningful results. This would be an excellent followup work to this.

7 Contributions

Mayukh has been doing all the research into the methods used. He is also working on implementing the neural network itself on AWS. He also wrote some of the scripts looking into the dataset.

David Knowles is a Post Doctoral Scholar in the Pritchard and Plevritis Lab who introduced me to the IDG-DREAM Challenge problem. He has been looking into the initial data analysis and provided the code for the analysis of the dataset in R – indicating the tight correlation between the values given.

References

[1] Martin E. M. Noble, Jane A. Endicott, Louise N. Johnson, "Protein Kinase Inhibitors : Insights into Drug Design from Structure", SCIENCE, 19 March 2004, Vol 303.

[2] Philip Cohen,"Protein Kinases - the major drug targets of the twenty-first century ?", Nature Reviews Drug Discovery 1, pp 309-315 (2002)

[3] Tomas Mikolov, Ilya Sutskevar, Kai Chen, Greg Corrado, Jeffery Dean, "Distributed Representations of Words and Phrases and their Compositionality", arXiv:1310.4546v1 [cs.CL] 16 Oct 2013.

[4] Jeffery Pennigton, Richard Socher, Christoper D. Manning, "Glove : Global Vectors for Word Represenation", Processings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Jan 2014.

[5] Netflix Recommender System, https://www.woired.co.uk/article/how-do-netflixs-algorithms-work-machine-learning-helps-to-predict-what-viewers-will-like

[6] Amazon Recommender System, http://rejoiner.com/resources/amazon-recommendations-secret-selling-online

[7]Machine Learning for Recommender Systems, https://medium.com/recombee-blog/machine-learning-for-recommender-systems-part-1-algorithms-evaluation-and-cold-start-6f696683d0ed

[8] IDG-DREAM Drug-Kinase Binding Prediuction Challenge Wiki

[9] Google's Developer Program's Page on Embeddings in machine learning. https://developers.google.com/machine-learning/crash-course/embeddings/obtaining-embeddings