Local Low-Rank Matrix Approximation for Drug-Target Interaction Prediction

AuthorList

July 22, 2018

Abstract

This is abstract to be filled.

1 Introduction

In pharmaceutical science, it is important to identify drug-target interactions in drug discovery process. However, with some many drugs and targets, it is costly to test all of the drug-target interactions. In fact, only a small portion of them has been experimentally validated. To improve the efficiency of drug discovery, we need to develop some accurate computational approaches that can predict the drug-target interactions. In this paper, we proposed a new method based on local low-rank matrix approximation for predicting drug-target interactions when only a small portion of drug-target interactions are validated.

...

In essence, the DTI(Drug-Target Interactions) prediction problem is a recommendation task which aims at suggesting potential DTIs. ... In this paper, we used an new approach based on matrix approximation called LLORMA(Local Low-Rank Matrix Approximation).

Matrix approximation is frequently used in machine learning. Given a few observed entries of a matrix $\{M_{a_1,b_1}, ..., M_{a_m,b_m}\}$, the task of matrix approximation is to use an algorithm to obtain an approximated matrix $\hat{M}$. In fact, the problem of approximating a matrix when a few entries are available is ill-posed, since there are infinite number of possibilities of approximating. When no other contexts are given, it is impossible to prefer some result to others. In practice, a common assumption is that M is a low-rank matrix. To formulate, we approximate matrix $M \in \mathbb{R}^{n_1 \times n_2}$ by a rank r matrix $\hat{M} = UV^T$, where $U \in \mathbb{R}^{n_1 \times r}$, $V \in \mathbb{R}^{n_2 \times r}$, and $r \ll min(n_1, n_2)$. This assumption is realistic in some real datasets. Also, low-rank matrix approximation often performs well when generalizing to the unobserved entries.

In this paper, we used a new approach based on matrix approximation into drug-target interaction. We relax the low-rank assumption of matrix approximation. We assume M can be constructed by several low-rank matrices $T_1, T_2, ..., T_n$.

We assume they are low-rank in their vicinity and their combinations works as a new approximation of M.

2 Low-rank matrix approximation

In this section, we formulate the problem and establish the notation in this paper for local low-rank matrix approximation. Also, an approach of low-rank approximation will be described. We denote matrix using upper case letters. The partially given matrix is denoted by $M \in \mathbb{R}^{n_1 \times n_2}$. A low-rank approximation of M is denoted by $\hat{M} = UT^V$, where $U \in \mathbb{R}^{n_1 \times r}$, $V \in \mathbb{R}^{n_2 \times r}$, and $r \ll min(n_1, n_2)$. The set of integers $\{1, 2, ...n\}$ is abbreviated as $[n]$. And the set of given entries are $\{(a_1, b_1), (a_2, b_2), ..., (a_m, b_m)\} \in [n_1] \times [n_2]$. We use $M_{i,j}(a, b)$ to denote the entry (i, j) of $M(a, b)$. We define a projection $\prod_A$ with respect to a set of indices A by the following function:

$$[\textstyle\prod_A(M)]_{a,b} = \begin{cases} M_{a,b} & (a,b) \in A \\ 0 & otherwise. \end{cases}$$

We denote by $\odot$ the component-wise product of two matrices. There are three types of matrix norms: the Frobenius norm $||X||_F = \sqrt{\sum_i \sum_j X_{i,j}^2}$, the sup-norm $||X||_\infty = sup_{i,j}|X_{i,j}|$, and the nuclear norm $||X||_* = \sum_{i=1}^r \sigma_i(X)$,where $\sigma_i(X)$ is the i'th singular value of X.

In this section, we mainly discuss two popular approaches for constructing a low-rank approximation $\hat{M}$ of M. The first is based on minimizing the Frobenius norm of $\Pi_A(M - \hat{M}$ and the second is based on minimizing the nuclear norm of a matrix according to the constraints.

$algorithm : IncompleteSVD$ This method constructs two matrix U and V by solving

$$(U, V) = \arg\min_{U,V} \sum_{(a,b)\in A} ([UV^T]_{a,b} - M_{a,b})^2 \tag{1}$$

or

$$\hat{M} = \arg\min_X ||\Pi_A(X - M)||_F \quad s.t.\ rank(X) = r. \tag{2}$$

The two equations are equivalent and can be solved by using gradient-based methods. But the rank r should be known in advance so as to determine the size of matrix U and V.

3 LLORMA: Local low-rank matrix approximation

In this section, we extend the method in the previous section to a local low-rank matrix approximation.

The idea of local low-rank matrix approximation is to estimate the matrix M by combination of several local low-rank matrices of the same size. We assume the a mapping of every entry of matrix M:

$$f : [n_1] \times [n_2] \to \mathbb{R}^{n_1 \times n_2} \quad \textit{where} \;\; T_{a,b}(a,b) = M_{a,b}$$

Instead of assuming the matrix M is low-rank, we here assume the matrices that make of it are low-rank. In order to make the idea of local works, we also assume that, as is often down in non-parametric statistics, the mapping f is slowly varying. Since the domain of f is discrete, we here assume it is Holder continuous instead of the classical definitions of continuity or differentiability.

A *local* method takes advantage of its vicinity and the concept of distance must be defined first. The distance $d((a_1, b_1), (a_2, b_2))$ of two points evaluates the similarities between rows a_1 and a_2 and columns b_1 and b_2. In practice, the distance can be determined in many ways. For example, in our DTIs, the similarity between drugs and targets are known from the datasets. But if no prior information is available, we may use a global low-rank matrix approximation method to determine the similarities between rows and columns.

With the definition of distance, we can articulate the idea of local with a smoothing kernel function $K_h(s_1, s_2)$ which is symmetric unimodal, where s_1,$s_2 \in [n_1] \times [n_2]$. the entries are non-negative since it means the distance of two entries. the value of h here means the spread of the kernel. The value of h means the spread of the kernel. If h is large, the kernel is wide.
There are three commonly used smoothing kernel:

Uniform kernel:

$$K_h(s_1, s_2) \propto \mathbf{1}[d(s_1, s_2) < h]$$

Triangular kernel:

$$K_h(s_1, s_2) \propto (1 - h^{-1} d(s_1, s_2))\mathbf{1}[d(s_1, s_2) < h]$$

Epanechnikov kernel:

$$K_h(s_1, s_2) \propto (1 - d(s_1, s_2)^2)\mathbf{1}[d(s_1, s_2) < h]$$

Based on the above equation, we here denote $K_h^{a,b}$ the matrix whose (i, j)-entry is $K_h((a, b), (i, j))$. Therefore, the local method of incomplete SVD can be extended to a local method as follows:

$$\hat{T}(a, b) = \arg\min_X ||K_h^{a,b} \odot \Pi_A(X - M)||_F \quad s.t. \; rank(X) = r. \tag{3}$$

With the algorithm above, we may estimate matrix $\hat{T}(a, b)$ for every $(a, b) \in [n_1] \times [n_2]$ and get the matrix M:

$$\hat{M}(a, b) = \hat{T}_{a,b}(a, b), \qquad (a, b) \in [n_1] \times [n_2]. \tag{4}$$

However, in practice, the matrix M is often two large, therefore calculating the $\hat{T}(a, b)$ for every $(a, b) \in [n_1] \times [n_2]$ results high computational complexity which is not practical to compute in practice. Instead, with the local method, we put forward a method for approximating the entries of a matrix in the next section.

4 Global approximation

Since we can not compute a matrix for every entry of M, we may compute some matrices from a certain points and use these matrices to approximate the whole matrix. The formulation we use estimate each entry is as follows:

$$\hat{\hat{M}}(s) = \Sigma_{i=1}^{q} \frac{K_h(s_i, s)}{\Sigma_{j=1}^{q} K_h(s_j, s)} \hat{T}(s_i). \quad where \;\; s \in [n_1] \times [n_2] \tag{5}$$

Using the above equation, we don't have to compute matrix for every entry of M and therefore is feasible in practice when the value of q is reasonable. The denominator is used to normalize the result. The performance of estimation may vary when we use different values of q. When q is larger, the error$\hat{\hat{M}}(a, b) - M(a, b)$ tends to be smaller which means the estimation is more accurate, but one the other hand, the cost of time tends to be higher.

The points $s_i, i = 1, 2, ..., q$ are called anchor points and the ways of selecting them will influence the result of estimation as well.

5 DTI and LLORMA

In this section, we will show the algorithm of LLORMA can be applied in the context of DTI. The performances of DTI prediction algorithms are evaluated on four benchmarks, including Nuclear Receptors, G-Protein Coupled Receptors, Ion Channels and Enzymes. These datasets were provided by A and are available at `http://web.kuicr.kyoto-u.ac.jp/supp/yoshi/drugtarget/` There are 3 types of matrices provided in each dataset. The observed DTIs; the drug similarities; the target similarities. Which can be used as the observed entries, distance between rows and distance between columns, respectively. The statistics of the datasets are shown in Table 1.

Table 1: The statics of the DTI dataset.

	Nuclear Receptor	GPCR	Ion Channel	Enzyme
Number of drugs	54	223	210	445
Number of targets	26	95	204	664
Number of interaction pairs	90	635	1476	2926
Average number of drugs per target	3.46	6.68	7.24	4.41
Average number of targets per drug	1.67	2.85	7.03	6.58
Sparsity of the interaction matrix	0.9359	0.97	0.9655	0.9901
Percentage of drugs that have only one interaction target	72.22%	47.53%	38.57%	39.78%
Percentage of targets that have only one interaction drug	30.77%	35.79%	11.27%	43.37%

Table 1 shows that the matrix of observed drug-target interactions is sparse and it is suitable to apply our LLORMA algorithm.

6 Experiments

7 Conclusion

8 Related work

9 Summary

References