Predicting Results of a Biological Experiment Using Matrix Completion Algorithms

by

Trevor Sabourin

A research paper
presented to the University of Waterloo
in partial fulfillment of the
requirement for the degree of
Master of Mathematics
in
Computational Mathematics

Supervisor: Prof. Ali Ghodsi

Waterloo, Ontario, Canada, 2011

I hereby declare that I am the sole author of this report. This is a true copy of the report, including any required final revisions, as accepted by my examiners.

I understand that my report may be made electronically available to the public.

Abstract

Biologists from McMaster's Brown Lab are interested in the mechanism of action of uncharacterized inhibitors of the growth of bacteria. To achieve their goal, they systematically combined 186 small molecules of unknown action with 14 known antibiotics of diverse mechanisms. The result of all these experiments is a 186 × 14 matrix of "synergy values" between 186 small molecules and 14 known antibiotics. The entries of the data matrix represent normalized bacteria growth rates in the presence of an antibiotic and a small molecule. The goal of this project is to create an algorithm that can predict all results of this biological experiment given only a subset of the results in question. This project is motivated by: the time and money it could save for experimental biologists, the opportunities for collaboration with people in other disciplines and the fascinating application of algorithms to a problem they were not designed to solve. We quickly realized that predicting all results of a biological experiment given a subset of the results boils down to completing a matrix given a subset of its entries. The methods that we tried fall into the broad categories of Collaborative Filtering (CF) and Nyström methods. In total we tried 4 methods: a Memory-Based CF method, a Model-Based CF method, the original Nyström method and Landmark Multidimensional Scaling. All of these methods present different ways to complete a matrix given only a subset of the entries. Both the Nyström and Landmark Multidimensional Scaling methods were unable to predict the results of our biological experiment with any accuracy. Possible reasons why these two methods failed are discussed. The Memory-Based CF method performed well enough to seem a viable choice for our application. Finally, of all the methods, the Model-Based CF method based on matrix factorization was the most successful.

Acknowledgements

I would like to thank Prof. Ghodsi for all the help with this project. I would also like to thank: Prof. Eric Brown for the data, Yuhan Ma for her loving support, my friends for the good times and my classmates for all their help.

To my family

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Biologists from McMaster's Brown Lab are interested in the mechanism of action of uncharacterized inhibitors of the growth of bacteria. To achieve their goal, they systematically combined 186 small molecules of unknown action with 14 known antibiotics of diverse mechanisms. They were interested in the small molecules' effects on the growth of bacteria in the presence of the known antibiotics. If the combination, of the small molecule and the antibiotic together, suppressed the growth of bacteria significantly more than the antibiotic alone − at least 3 times more − the combination was considered to be synergistic.

n the Dr. Eric Brown paper [3], they were able to draw meaningful conclusions about biological molecules by using their expert knowledge to study a heat-map (an image) of their experimental results

The result of all these experiments is a 186 × 14 matrix of "synergy values" between 186 small molecules and 14 known antibiotics. The data can be found in: [3]. The entries of the data matrix represent normalized bacteria growth rates in the presence of an antibiotic **and** a small molecule. This matrix of synergy values provides a fingerprint of the biological activity of these unknown small molecules. Biologists can use these fingerprints to hypothesize the mechanisms of action of the small molecules [3]. Visualizing the matrix as an image (called a heat-map) is useful when analyzing these fingerprints. Knowing the chemical activity of these small molecules could lead to new drugs to help combat drug resistant bacteria. The synergy values should be bounded by zero and one, where one represents a growth rate equal to that of the control and zero represents no bacteria growth at all. The bacteria growth in the presence of **only** the antibiotic was used as the control. It is important to note that synergy values equal to zero are not possible, since there will always be some bacteria growth. The Eric Brown lab further binned the data into two categories: synergy observed and no synergy observed. Synergy values less than or equal to 0.25 correspond to cases where synergy was observed while synergy values greater than 0.25 correspond to cases where no synergy was observed.

The goal of this project is to create an algorithm that can predict all the results of this biological experiment given only a subset of the results in question. The motivation for this project comes from three sources. First, this algorithm would be a powerful tool that could save time and money for experimental biologists; the biologists could run only a **subset** of the experiments and still obtain **all** the results. Second, biologists present a great opportunity for collaboration because they are not likely to be familiar with the power of computational mathematics. Finally, this project presents an interesting application for algorithms that were not designed for this purpose.

Predicting all the results of this biological experiment given only a subset of the results is equivalent to completing a matrix given a subset of its entries. There are many methods for completing a matrix given a subset of its entries. These methods are based on the assumption that the complete matrix is low rank. Therefore, each entry of the full matrix can be approximated as a weighted linear sum of a small number of basis vectors. In the case of our synergy values, we must assume that there exists only a small number of chemical fingerprints that are unique and that all other chemical fingerprints are just a weighted sum of these unique fingerprints. This is a reasonable assumption for most real-life data matrices that must be made for our problem to be solvable.

There are two types of incomplete matrices: one type has all the known values in one block and the other has the known values scattered randomly. Each of these types of incomplete matrices lead to different matrix completion algorithms. For incomplete matrices with a known block, we tried both the Nyström algorithm and the Landmark Multi-Dimensional Scaling (LMDS) algorithm. For incomplete matrices with scattered known entries, we tried Collaborative Filtering methods. Under Collaborative Filtering, we tried both a Memory-Based and a Model-Based algorithm.

Chapter 2

Methodology and Method Evaluation

To achieve our goal, the algorithms must only look at a subset of the data. Therefore, we start all of our simulations by creating 2 matrices: the original (full) data matrix and a second matrix that is mostly empty. We choose the locations of missing values differently for different methods. For the Nyström and LMDS methods we kept the first k columns and made empty the last k columns, while for the Collaborative Filtering methods in each row an equal number of randomly located empty entries were chosen. The algorithms only used the incomplete second matrix as input data. Finally, to determine the accuracy of the methods, we compared the guessed values for the entries that were missing with the true values from the original matrix. We used Root Mean Squared Error (RMSE) as our measure of accuracy because it is quite sensitive to the occasional large error and is generally accepted as a good measure of precision. RMSE measures the square root of the average of the squares of the errors. Where the error is the amount by which the predicted values differs from the true value.

In our case RMSE was calculated as follows:

$$RMSE(D,\hat{D}) = \sqrt{\frac{\sum_{i,j\in R}(d_{i,j} - \hat{d}_{i,j})^2}{n}} \tag{2.1}$$

where D is the true complete data matrix, $\hat{D}$ is the predicted data matrix and R is the set of all data points (a,b) excluded from our input data (i.e. the set of all points that need to be predicted). [1]

[1] Note: For the LMDS method the true complete data matrix was put into the kernel space so that the

The Eric Brown lab focused on classifying the data into two groups. To follow suit, we introduced another error measure that we called Binning Error. The Binning Error represents the number of predicted values that were miss-classified with respect to the 0.25 cut-off. Binning Error comes in two forms: false positives and false negatives. A false positive is recorded when the true value is non-synergistic (greater than 0.25) but the value was predicted to be synergistic (less than 0.25). A false negative is the opposite of a false positive and is recorded when the true value is synergistic (less than 0.25) but the value was predicted to be non-synergistic (greater than 0.25). We kept all three errors separate because a biologist, doing real experiments, may be interested in one error more than the others. In that case, the biologist may want to bias the algorithm used to minimize the error that interests him, while keeping track of the other errors to ensure his results remain meaningful.

Binning With a Probability Function

Recall that we observed two different types of error for our simulations. We observed both numerical error (RMSE) and Binning Error. Numerical error is the most intuitive and was easy to handle. Binning Error on the other hand was difficult to deal with because the cut-off line (0.25) is artificially chosen and the amount of numerical error allowed before binning error occurs is dependent on the true values. For predicted values far from the cut-off line, it is easy to be confident that the predicted values fall into the correct bin; whereas for predicted values close to the cut-off line, it is unclear if these values are in the correct bin or if the error from our method has pushed them over the cut-off line and into the wrong bin. Therefore, we suggest a probability function to give a probability of being in each bin. We used a sigmoid as our probability function.

$$P(x) = \frac{1}{1 + e^{(s \times (x-p))}} \tag{2.2}$$

where x is the predicted synergy value, P is the probability of x belonging to the non-synergistic bin, s is a parameter that affects the steepness of the sigmoid and p is a parameter that affects the centre of the sigmoid (the point where the probability of being in each bin is equal).

We centered our sigmoid on the cut-off line between synergistic and non-synergistic behavior and chose the steepness see in the image above. In other words, parameter p was set to

error on the predicted values could be calculated.

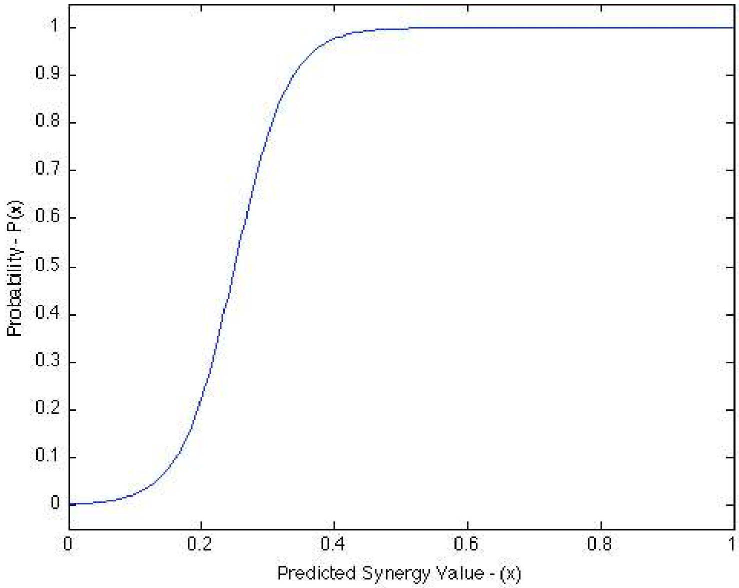

Figure 2.1: Sigmoid With $p = 0.25$ and $s = 25$

0.25 and parameter s was set to 25. We chose this steepness because the results matched well with our intuitive confidence levels.

We ran our most successful method, with and without the sigmoid binning to see if this probability function improved the results. Details are discussed in section 4.4.3.

Chapter 3

Nyström and LMDS Methods

3.1 A Brief Introduction to the Nyström and LMDS Methods

Landmark Multi-Dimensional Scaling [10] and the Nyström method [13] are two other techniques for completing matrices given a limited number of entries that we tried. It has been shown LMDS is equivalent to a modification on the Nyström method [8]. These methods require the user to start with the known values organized in a block [2][13]:

- Given a symmetric positive semi-definite (PSD) $n \times n$ kernel matrix $K = \begin{pmatrix} A & B \\ B^T & C \end{pmatrix}$
- Where $k \times k$ sub-matrix A and $k \times (n-k)$ sub-matrix B are **known** but the $(n-k) \times (n-k)$ sub-matrix C is **unknown**.
- The Nyström and LMDS methods approximate the unknown sub-matrix C.

C is approximated exactly if the rank of K is at most k. This fact is based on the following property of PSD matrices:

For every PSD matrix K, there exists points in a Euclidean space such that their corresponding kernel matrix is K [12]

Suppose, the K matrix can be fully represented on an m-dimensional manifold (Where $m \leq k$). Knowing the relative distances between all k points in A and the distances from these k points to every other $k+1$ to n points in B, eliminates all degrees of freedom from every point in A and B. Therefore, the exact locations of all points on the manifold can be determined. If the locations of all points were known, solving for C is a trivial matter of measuring the distances between points and transforming these distances back to the kernel space.

If the rank of K is greater than k the entries of the K matrix can only be partially represented on a k-dimensional manifold and the Nyström method yields an approximation to C.

3.1.1 The Landmark Multi-Dimensional Scaling Modification

LMDS is equivalent to the Nyström method but we are assumed to start with a symmetric distance matrix $D = \begin{pmatrix} E & F \\ F^T & G \end{pmatrix}$ instead of a PSD kernel matrix. Therefore, entry $D_{i,j}$ is assumed to be the distance between points i and j. The method works by first mapping the distance matrix to a PSD kernel matrix, then using the Nyström method on this kernel matrix. The mapping is done using the Landmark MDS algorithm, which is a modification of the original MDS algorithm. These algorithms work by double centering our distance matrix D to convert it to the needed kernel matrix K [8].

3.1.2 Why Try the Nyström and LMDS Methods

We decided to try these methods to predict the results of our biological experiment for two reasons. First, although these methods assume certain properties about our data, which may or may not be consistent, these methods can be viewed simply as algorithms that guess the entries of incomplete matrices. The second reason is that, from a practical point of view, data collection that results in an incomplete matrix where the known data is in a block is simpler than data collection resulting in a matrix with scattered known values. Columns of data need to be collected for the block matrix − where one column represents experiments done all with one of the two chemicals remaining constant. The column based data collection method is simpler and surely more desirable than the scattered approach. We decided to try both the original Nyström method as well as LMDS because both methods assume different properties regarding the initial data; assumptions that may or may not be true.

3.2 Special Methodology

Both the Nyström and LMDS methods require special methodology on top of the basic methodology explained in section 2.

3.2.1 Satisfying the Conditions

As mentioned previously, both the Nyström and LMDS methods require the full data matrix to be symmetric. The LMDS method also requires the entries to represent distances between points, while the Nyström method requires the matrix to be a kernel matrix. These conditions presented small challenges that were resolved using data preprocessing and some intuition.

The Symmetry Condition

First, to satisfy the symmetry condition, prior to both methods we added a 14×14 symmetric matrix containing all combinations of synergy values between the 14 known antibiotics to our data. Adding that 14×14 matrix made our data a 200×14 rectangular piece of a 200×200 symmetric matrix (this will be sufficient to satisfy the symmetry condition as we will see in the next section).

It is important to note that an antibiotic cannot synergize with itself. In fact, adding an antibiotic to itself is equivalent to the control used in these biological experiments. Therefore, the diagonal of the 14×14 matrix added to our data should be all ones.

The Kernel Condition

A kernel matrix is a matrix where the entries represent similarities between objects. The domain, in which these similarities exist, does not need to be known. If we make the reasonable assumption that small molecules that have similar synergy values are similar to each other in some domain $-$ where this domain can be anything from molecular structure to biological action $-$ then we can consider the matrix of synergy values to be a kernel matrix.

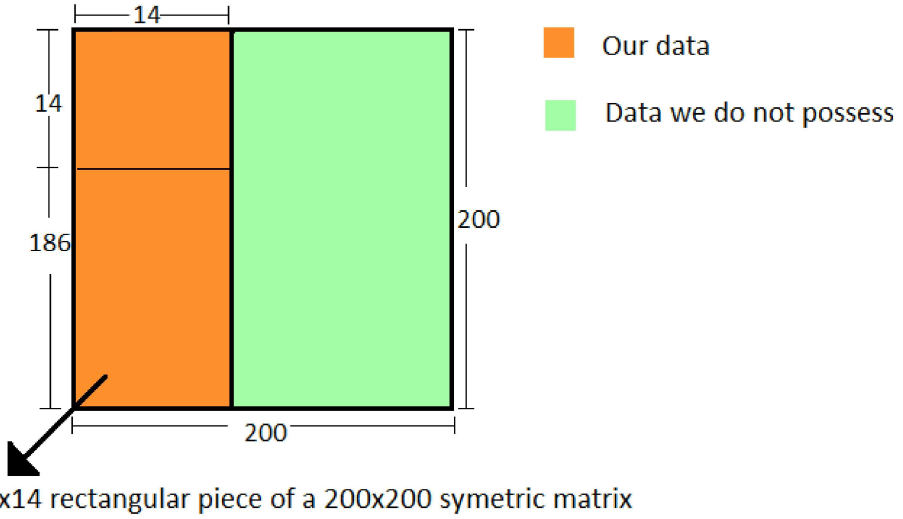

Figure 3.1: Our Data as a 200 × 14 Rectangular Piece of a 200 × 200 Symmetric Matrix. *Note: The image is not to scale to highlight the symmetric top left hand corner.*

The Distance Condition

Finally, given that we has already interpreted the synergy values as a measure of similarity between the molecules, we can further interpret these entries as distances between the molecules. This extension amounts to forcing the domain in which the similarities exist to be spacial locations. Some data processing is required before this extension is reasonable.

Prior to the LMDS method we took a 200 × 14 matrix full of ones and subtracted our data from it. Subtracting these two matrices still left us with a data matrix full of values between zero and one but it reversed the meaning of the values. Values greater than 0.75 now corresponded to synergistic combinations, values less than 0.75 corresponded to non-synergistic combinations and a value of 0 corresponded to the control. It was important to reverse the meaning of the values because a distance matrix, by definition, must have zeros all along the diagonal; the distance between a point and itself must be zero under any valid metric. Note that: reversing the meaning of the values does not effect our kernel assumption. Therefore we can still consider this modified matrix of synergy values as a distance matrix.

3.2.2 Using the Nyström and LMDS Methods on a Rectangular Matrix

The Nyström and LMDS methods only work on symmetric matrices and only square matrices can be symmetric.This fact implies that we need our complete data matrix to be a square matrix to use a Nyström method. Fortunately, a Nyström method can also be used on a rectangular matrix, as long as this matrix represents a rectangular portion of a larger symmetric matrix.

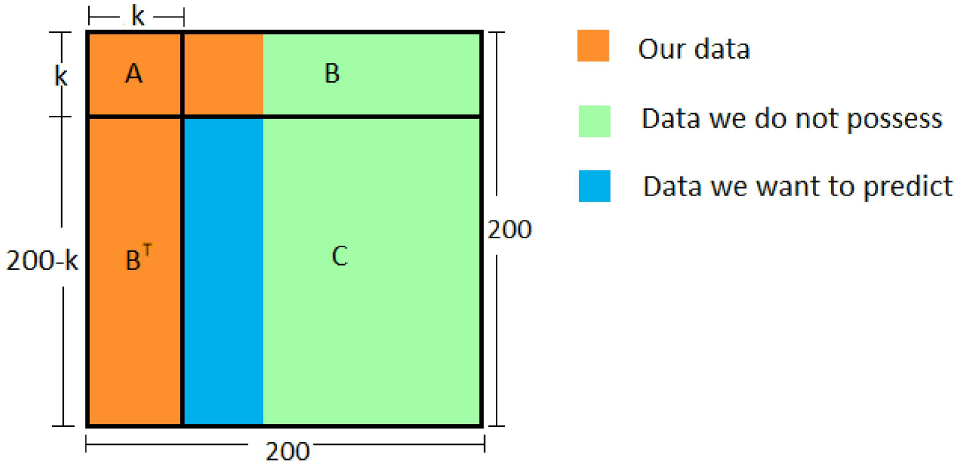

Figure 3.2: How We Used the Nyström and LMDS Methods on Our Rectangular Matrix. *Note: k must be* < 14*.*

Using a Nyström method, given the A and B^T sub-matrices from the above image, we should be able to solve for the C sub-matrix. Note that, due to the preprocessing discussed above, the information for both the A and B^T sub-matrices are contained in the section corresponding to our data. Therefore, we have all the information needed to use a Nyström method on our rectangular data matrix. We also point out that most of the C matrix corresponds to data results we do not possess. The entries of the C matrix corresponding to this data were dropped because there was no way to verify the correctness of these results.

3.3 Algorithms

In this section we explore the exact algorithms that we employed. The algorithms are first presented in their original form. Following each presentation, a modified variation of the algorithm is proposed in order to handle the specific application.

3.3.1 Nyström

For the Nyström method we used the following algorithm (from [2][13]):

- Given positive semi-definite $n \times n$ kernel matrix $K = \begin{pmatrix} A & B \\ B^T & C \end{pmatrix}$
- Where $k \times k$ sub-matrix A and $k \times (n-k)$ sub-matrix B are **known** and $(n-k) \times (n-k)$ sub-matrix C is **unknown**.
- Approximate C as:
$$\hat{C} = B^T A^{-1} B \tag{3.1}$$
- Approximate K as:
$$\hat{K} = \begin{pmatrix} A & B \\ B^T & \hat{C} \end{pmatrix}$$

The algorithm was used exactly as described above. As mentioned in section 3.2.2, the predicted synergy values in the final $\hat{K}$ matrix that correspond to experimental results that we do not possess were ignored.

3.3.2 Landmark Multi-Dimensional Scaling

For the LMDS method we used the following algorithm (from [8][10]):

- Given symmetric $n \times n$ distance matrix $D = \begin{pmatrix} E & F \\ F^T & G \end{pmatrix}$
- Where $k \times k$ sub-matrix E and $k \times (n-k)$ sub-matrix F are **known** and $(n-k) \times (n-k)$ sub-matrix G is **unknown**.
- Find A and B of PSD kernel matrix $K = \begin{pmatrix} A & B \\ B^T & C \end{pmatrix}$ using:

$$A_{i,j} = D_{i,j}^2 - \frac{1}{m}\sum_i D_{i,j}^2 - \frac{1}{m}\sum_j D_{i,j}^2 + \frac{1}{m^2}\sum_{i,j} D_{i,j}^2 \tag{3.2}$$

$$B_{i,j} = F_{i,j}^2 - \frac{1}{m}\sum_i F_{i,j}^2 - \frac{1}{m}\sum_j F_{i,j}^2 \tag{3.3}$$

- Use A and B to approximate C using equation 3.1
- Approximate K as:

$$\hat{K} = \begin{pmatrix} A & B \\ B^T & \hat{C} \end{pmatrix}$$

The algorithm was used exactly as described above. Once again, as mentioned in section 3.2.2, the predicted synergy values in the final $\hat{K}$ matrix that correspond to experimental results that we do not possess were ignored.

3.4 Results

3.4.1 Nyström

# of Known Columns	RMSE
2	0.2923
3	0.3651
4	1.8844
5	1.9260
6	1.9698
7	1.9983
8	7.3072
9	16.0220
10	11.1367
11	1.2676
12	1.6868
13	0.9215

Table 3.1: Nyström Method: RMSE

To put the scale of the errors in context: With 7 unknown columns: the true values we are trying to predict have: min = -0.0031, max = 0.9969 and mean = 0.6018. A RMSE greater than or approximately equal to the mean value we are trying to predict is unacceptable.

3.4.2 Landmark Multi-Dimensional Scaling

# of Known Columns	RMSE
2	3.1851
3	302.3723
4	499.8042
5	556.3088
6	937.8035
7	455.6081
8	355.8806
9	274.1246
10	345.2073
11	642.4776
12	292.3814
13	413.1089

Table 3.2: LMDS Method: RMSE

To put the scale of the errors in context: With 7 unknown columns: the true kernel values we are trying to predict have: min = -0.5776, max = 10.0000 and mean = 3.9755. A RMSE greater than the mean value we are trying to predict is unacceptable.

3.5 Discussion

From our results (Tables 3.1 and 3.2), we can see that both the Nyström and LMDS methods were unable to predict synergy values with enough accuracy. Although the errors with 2 and 3 known columns in Table 3.1 are relatively low, the values predicted by the method with so few columns were not desirable. With so few known columns, the predicted values in each row barely varied from the mean of the known values for that row. Predicting the mean of the know values does not provide the user with any new information and is therefore not desirable. Due to the failure of the methods, there must be a fundamental problem that arises when these methods are used in our particular application. Three issues with these methods, with respect to our data, need to be addressed.

The first issue relates to the Nyström method. This method requires the complete data matrix to be a positive semi-definite matrix. In our case, the complete data matrix is a

200×200 matrix of which a 200×186 subset corresponds to data that have not been observed (as seen in Figure 3.1). Though we are missing most of the data, we can be certain that the matrix is, unfortunately, not a PSD matrix. To verify this claim we use a property of PSD matrices that states:

Any block from the diagonal of a PSD matrix must itself be a PSD matrix. [6]

Unfortunately, the 14×14 block that we added to our data to make it symmetric is not PSD. Despite many efforts to modify our data to be PSD we were unable to do so.

The second issue relates to LMDS. This method requires the complete data matrix to be a distance matrix. Unfortunately, the triangle inequality does not hold for our data. Given that all metrics (even non-linear ones) must satisfy the triangle inequality, our data cannot follow any underlying metric. Therefore, our data cannot be any sort of distance matrix.

The final issue is that the 14×14 symmetric matrix (containing all combinations of synergy values between the 14 known antibiotics) added to our data may not be consistent with the rest of the data (a 186×14 matrix containing the synergy values between 186 small molecules and the 14 known antibiotics). These results may not be consistent because the antibiotics used are all much larger than the small molecules and one of the factors that influences biological results is the relative size of the molecules involved. If two big antibiotics can not be seen as using the same distance measure as one big antibiotic and one small molecule, then this “apples and oranges comparison” would explain the failure of the methods. This issue alone suggests that the Nyström and LMDS methods may not be appropriate for use on our particular set of data.

Chapter 4

Collaborative Filtering

4.1 A Brief Introduction to Collaborative Filtering

Collaborative Filtering (CF) methods were invented to complete very sparse matrices filled with randomly located known entries. CF methods are typically used to predict a user's preferences for unseen items – typically movies or books – based on a database of many users' ratings or preferences for the items they have seen [11]. These methods became a popular research topic towards the end of the last decade due to the Netflix Prize competition [5]. A competition, put on by Netflix in 2007, offering a USD 1M award to the first research group that could beat the Netflix prediction algorithm (Cinematch) by at least 10 percent. The competition had 51051 contestants on 41305 teams from 186 different countries [5]. The interest this competition generated sparked many advances in the field of Collaborative Filtering. Two categories of CF techniques are: Memory-Based and Model-Based [11].

4.1.1 Memory-Based Collaborative Filtering

The first CF algorithms are Memory-Based algorithms because they are more intuitive. Memory-Based CF algorithms are based on the idea that users who agree on content, that they have both already seen, are likely to agree on content that only one of them has seen. In simpler terms, Memory-Based CF is a large-scale version of asking your friend, who has similar interests as you, to recommend you a book or a movie. The algorithms work by prescribing weights to every other user that depend on the similarity between the preferences of the items you have both already seen. Then these weights are used to perform a weighted average to predict one's preference for unseen items based on the preferences of one's peers for those items.

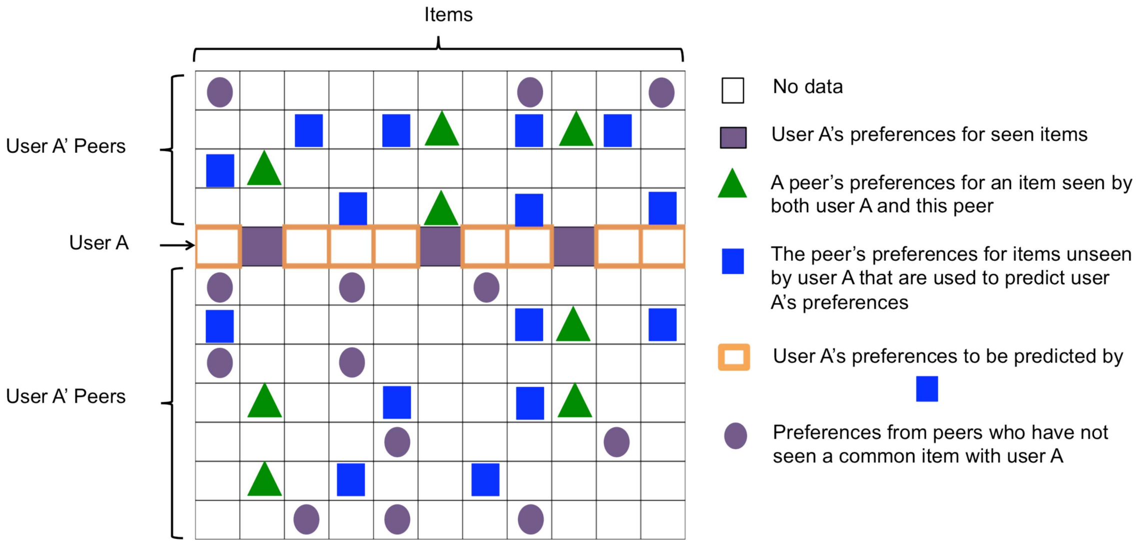
Items
User A' Peers
User A
User A' Peers
No data
User A's preferences for seen items
A peer's preferences for an item seen by both user A and this peer
The peer's preferences for items unseen by user A that are used to predict user A's preferences
User A's preferences to be predicted by
Preferences from peers who have not seen a common item with user A

Figure 4.1: Visualizing Memory-Based CF

4.1.2 Model-Based Collaborative Filtering

More recently, thanks to the Netflix prize, Model-Based algorithms have become popular. Model-Based techniques are usually based on machine learning algorithms and are often not intuitive [11]. The goal for Model-Based methods is to first create a model then, using the known preferences of users, solve for the unknowns parameters in this model. Finally the model is then used to predict users' preferences for unseen items. In particular, the Model-Based algorithm we use in this paper is a Matrix Factorization method. These aptly named methods are based on creating two or more matrices whose product gives an approximation to the full matrix of users' preferences. For example:

- If the users' preference matrix U is $m \times n$, *where m is the number of items and n is the number of users.*

- Then the factors V and W, solved for such that $V \times W \simeq U$, are chosen to have dimensions $m \times k$ and $k \times n$ respectively, *where* $k \ll 0.5m$ *and* $k \ll 0.5n$.
- Therefore, instead of using a limited amount of data to try and solve for $m \times n$ entries, we use a limited amount to try and solve for $k \times m + k \times n$ entries.

$$\text{Where: } k \times m + k \times n \ll 0.5n \times m + 0.5m \times n = m \times n.$$

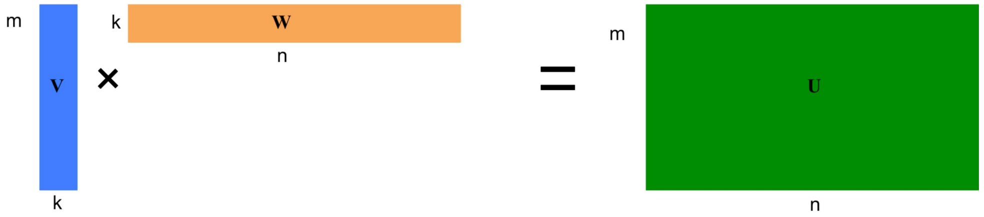

Figure 4.2: Visualizing Matrix Factorization

The entries of the smaller matrices, V and W, no longer represent users' preferences, they represent features related to the users' preferences. The $(m \times k)$ V matrix represents how much each item has of each feature and the $(k \times n)$ W matrix represents how much each user likes each feature. Choosing these features is easily done using a gradient descent method [7].

4.1.3 Relevant Notes on Collaborative Filtering

There are two main differences between most Memory-Based and Model-Based methods that are worth highlighting. First, Memory-Based methods predict the unknown values directly, while Model-Based methods solve for the parameters of the model and then use these parameters to predict the values. The second difference is that Memory-Based methods only make predictions for the unknown values, while Model-Based methods use the model to predict all values and compare the known values with their predictions to train the model.

Factors that affect the results of CF methods are: the number of items one has seen, the number of items one's peers have seen and the number of common seen items between one and one's peers. For all CF methods, the more items one has seen and the more items one's peers have seen, the better the method works. Less intuitively, for Memory-Based CF methods, the results are also dependent on the number of common seen items between one and one's peers [11].

4.1.4 Why Try Collaborative Filtering

We thought that CF techniques could predict the results of our biological experiment, given a subset of the results, because although biological experiments are not movie ratings, they are not random numbers either; there are underlying reasons behind the results of an experiment just as there are behind the ratings of a movie. Whether for a movie or a biological experiment, these underlying reasons form the basis of our results. CF methods assume that these basis vectors are few in number and try to predict the results as a weighted sum of a set of basis vectors. We were confident that the chosen CF methods would be able to access these basis vectors and accurately predict the results of our experiment − as they have shown to do for movie ratings.

4.2 Algorithms

In this section we explore the exact algorithms that we employed. The algorithms are first presented in their original form. Following each presentation, a modified variation of the algorithm is proposed in order to handle the specific application.

4.2.1 Memory-Based Collaborative Filtering

For Memory-Based CF we used the following method (from [1]):

Suppose an incomplete matrix U is given, where each row represents a user and each column represents an item. The goal is to predict all unknown values of U using a weighted sum of the known values of U.

- $U_{i,j}$ = preference of user i on item j.

- I_i = the set of all items which user i has seen.
- Mean preference for user i is
$$\bar{U}_i = \frac{1}{\mid I_i \mid} \sum_{j \in I_i} U_{i,j} \tag{4.1}$$
- The predicted preference for user a on item j is a weighted sum
$$P_{a,j} = \bar{U}_a + \kappa \sum_{i=1}^{n} w(a,i)(U_{i,j} - \bar{U}_i) \tag{4.2}$$
Where κ is a normalizing factor.
- The weights $w(a,i)$ are determined from
$$w(a,i) = \sum_{j} \frac{U_{a,j}}{\sqrt{\sum_{k \in I_a} U_{a,k}^2}} \frac{U_{i,j}}{\sqrt{\sum_{k \in I_i} U_{i,k}^2}} \tag{4.3}$$
- All predicted preferences were solved for using equation 4.2 for each user on each unseen item.

The method was used as described above, except for the meaning of the variables which were changed to give predictions of biological results:

- $U_{i,j}$ = synergy value of small molecule i with antibiotic j.
- I_i = known results involving small molecule i.
- $\bar{U}_i$ = mean synergy value for small molecule i.
- $P_{a,j}$ = predicted synergy result for small molecule a with antibiotic j.

4.2.2 Model-Based Collaborative Filtering

For Model-Based CF we used the following algorithm (presented by Funk in [4]):

Suppose an incomplete matrix U is given, where each row represents a user and each column represents an item. The goal is to find matrices V and W such that $V \times W = \hat{U}$ where $\hat{U} \simeq$ the complete U matrix.

- $V =$ a $(m \times k)$ matrix relating each item to each feature.
- $W =$ a $(n \times k)$ matrix relating each user to each feature.
- γ is the learning rate and λ is a penalizing factor to avoid over-fitting.
- The V and W matrices are initially full of zeros.
- Pseudo Code of the Algorithm:

```
 1: for each feature $f = 1 \rightarrow k$ do
 2:    we set the columns corresponding to that feature in the $V$ and $W$ matrices to
       some initial value $\vec{a}$
 3:    while the following procedure has looped fewer than $n$ times do
 4:       Find the current predicted values $P = V \times W^T$
 5:       Find the current errors $E = U - P$ for all the known user preferences.
 6:       for all known entries $(u, i)$ in the $U$ matrix do
 7:          Solve $V_{i,f} = V_{i,f} + \gamma \times (E_{u,i} \times W_{u,f} - \lambda \times V_{i,f})$
             and $W_{u,f} = W_{u,f} + \gamma \times (E_{u,i} \times V_{i,f} - \lambda \times W_{u,f})$
             *The above equations come from the gradient descent technique.*
 8:       end for
 9:    end while
10: end for
11: The final predictions are found by repeating step 4 with the final $V$ and $W$
    matrices.
```

The algorithm was used as described above with the following details:

- U was the matrix of known synergy values.

- $V =$ a $(m \times k)$ matrix relating each antibiotic to each feature.
- $W =$ a $(n \times k)$ matrix relating each small molecule to each feature.
- $\gamma = 0.001$ and $\lambda = 0.02$[1]
- We chose to use $f = 4$ features.
- The initial value $\vec{a}$ used was a vector with every entry equal to 0.1
- We iterated the main part of the procedure $n = 800$ times for each feature f.

The results from this method improve as the number of features increase. This relationship follows a curve similar to a logarithmic function of the features. We chose $f = 4$ features because at this point there was a major drop-off in the improvement of the method for each subsequent feature added and the run-time of the method depends on the number of features. *For our application, it is useful to think of these features as the vectors of our basis.*

[1] We chose these γ, λ, $\vec{a}$ and n values because they were recommended by Funk [4] and they worked well on our data.

4.3 Results

The results from Tables 4.1- 4.3 are the average of 50 runs where the known entries in each row were randomly chosen each time:

	RMSE	
Unknown Entries / Row	Memory-Based	Model-Based
1	0.1524	0.1440
2	0.1499	0.1440
3	0.1536	0.1466
4	0.1586	0.1463
5	0.1603	0.1478
6	0.1625	0.1493
7	0.1662	0.1494
8	0.1732	0.1512
9	0.1830	0.1532
10	0.1939	0.1579
11	0.2148	0.1665
12	N/A	0.1941
13	N/A	0.3141

Table 4.1: Collaborative Filtering Methods: RMSE

To put the scale of the errors in context: The true values we are trying to predict have: min = 0.0017, max = 1.1406 and mean = 0.6659. These RMSEs are much better than those associated with the Nyström and LMDS methods.

	False Negatives		Prob. a Prediction is a False Neg.	
Unknown Entries / Row	Memory-Based	Model-Based	Memory-Based	Model-Based
1	12.3000	8.9200	0.0661	0.0480
2	24.0667	19.0400	0.0647	0.0512
3	38.3000	27.4000	0.0686	0.0491
4	51.3000	36.1800	0.0690	0.0486
5	63.1333	46.3600	0.0679	0.0498
6	77.8333	55.6200	0.0697	0.0498
7	89.9333	65.9400	0.0691	0.0506
8	108.4000	78.4200	0.0728	0.0527
9	124.7000	89.1400	0.0745	0.0532
10	139.6667	105.0000	0.0751	0.0565
11	160.3667	123.8800	0.0784	0.0605
12	N/A	155.5200	N/A	0.0697
13	N/A	111.5400	N/A	0.0461

Table 4.2: Collaborative Filtering Methods: False Negatives

	False Positives		Prob. a Prediction is a False Pos.	
Unknown Entries / Row	Memory-Based	Model-Based	Memory-Based	Model-Based
1	4.2333	4.8800	0.0228	0.0262
2	7.0000	9.0000	0.0188	0.0256
3	10.1333	14.2800	0.0182	0.0256
4	15.0667	17.8800	0.0203	0.0240
5	19.1667	24.1800	0.0206	0.0260
6	23.0333	28.8400	0.0206	0.0258
7	27.8667	32.1000	0.0214	0.0247
8	31.9000	37.9600	0.0214	0.0255
9	35.6667	40.3600	0.0213	0.0241
10	44.9333	43.7000	0.0242	0.0235
11	55.0667	50.1600	0.0269	0.0245
12	N/A	61.8600	N/A	0.0277
13	N/A	261.5000	N/A	0.1081

Table 4.3: Collaborative Filtering Methods: False Positives

For Tables 4.2 & 4.3: Note that on average the number of missing values that should be labeled synergistic is 20.7143 × # of missing values per row.

Unknown Entries / Row	Standard Deviation
1	0.1432
2	0.1433
3	0.1437
4	0.1442
5	0.1446
6	0.1459
7	0.1468
8	0.1472
9	0.1494
10	0.1536
11	0.1588
12	0.1715
13	0.1480

Table 4.4: Model-Based CF: Standard Deviation of Predicted Values

Note: The true standard deviation of the complete data matrix is 0.2092.

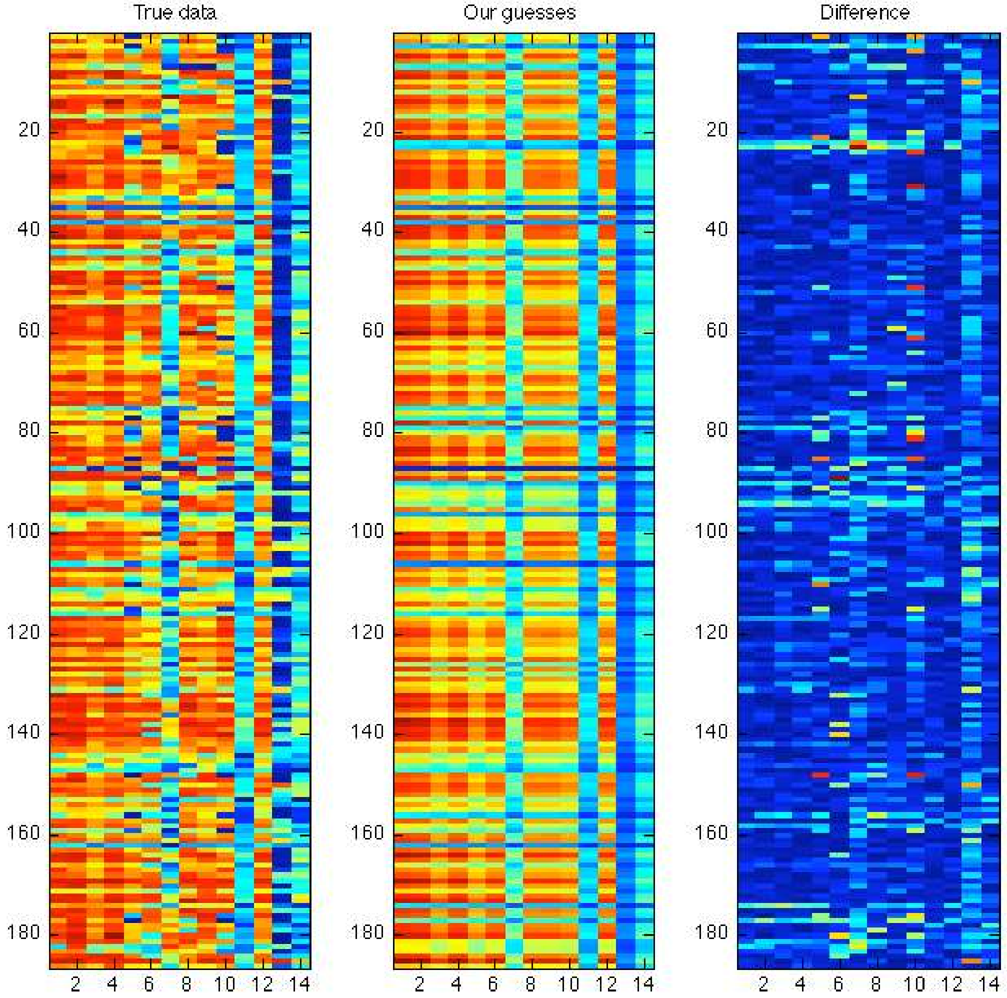

Figure 4.3: Model-Based CF: A Visual Representation of our Results Using 3 Known Values in Each Row. These results have a RMSE = 0.1577 and there are 183 falsely binned values. *The colours range from blue to red where: blue rectangles represent small entries and red rectangles represent large entries.*

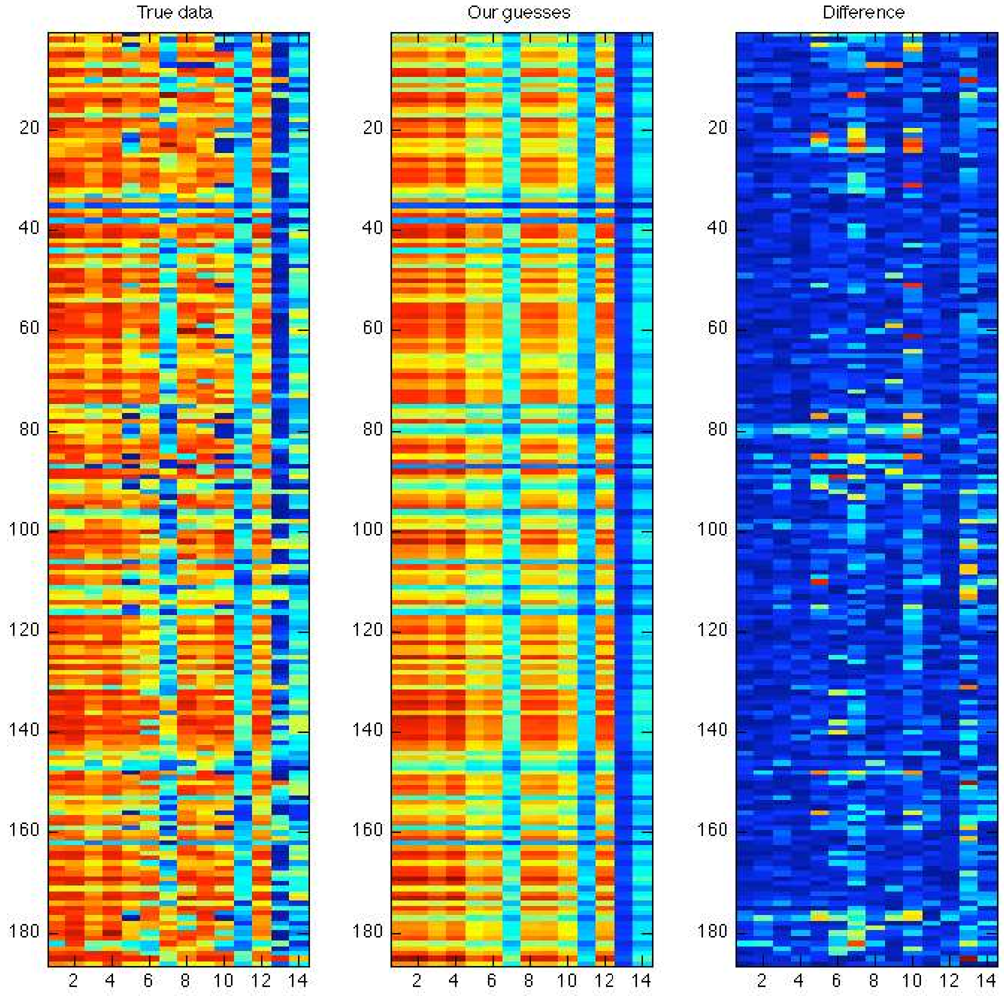

Figure 4.4: Model-Based CF: A Visual Representation of our Results Using 7 Known Values in Each Row. These results have a RMSE = 0.1496 and there are 100 falsely binned values.*The colours range from blue to red where: blue rectangles represent small entries and red rectangles represent large entries.*

	Without Sigmoid	With Sigmoid
False Pos.	31.6000	19.4000
False Neg.	69.5500	53.1500
# Unsure Values	N/A	56.6500
RMSE	0.1492[1]	0.1492

Table 4.5: Model-Based CF: Sigmoid Binning Results – 3 Known Values per Row

Note: "Unsure Values" are predicted values that were deemed synergistic with a probability between 0.3 and 0.7, as determined by our sigmoidal probability function .

4.4 Discussion

Both our Collaborative Filtering methods performed much better than the Nyström and LMDS methods. This is a reasonable result because, beyond the low rank assumption made by all matrix completion methods, our CF methods make no assumptions about the data – assumptions that are likely to be incorrect given our unique application of these methods.

4.4.1 Memory-Based Collaborative Filtering

From our results (Table 4.1), we can see that the Memory-Based CF method predicted synergy values with a RMSE of 0.1939 when there were 10 missing values in each row. Although this method performed very well when there were many known entries in each row, the errors were still quite high when there were few entries in each row. These large errors are likely due to over-fitting. Although there are many way to regularize Memory-Based CF methods to avoid over-fitting, none of the modifications we tried improved our errors. This over-fitting problem is mostly due to our small data sample. The larger the data sample, the easier it is to regularize the methods and avoid over-fitting.

4.4.2 Model-Based Collaborative Filtering

Our Matrix Factorization CF method performed the best out of all the methods (as seen in Table 4.1). With only 3 known entries in each row we found a RMSE of 0.1665. Our

[1] The RMSE is not influenced by the probability function

Model-Based method likely outperformed the Memory-Based method because we were able to limit the complexity of our model by using only four latent features. Limiting the model to a low number of features regularizes the model to help avoid over-fitting. Our results suggest that this method was able to access underlying features that are representative of the biological action that takes place in the experiments. If these features are biologically relevant, they imply that there are only four main "basis reactions" and all reactions are a linear combination of these four. It would be a very interesting biological result if sets of reactions could be broken down into their "basis reactions". Unfortunately, there is no known way to find out what are the features that the Matrix Factorization method has found.

4.4.3 Binning Results and Alternatives

Both of our CF methods had difficulty binning the data (as seen in Tables 4.2 and 4.3). With only 3 known entries in each row, despite the excellent RMSE, our Model-Based CF method miss-classified 174.0400 values (8.5% of the predicted values). One likely reason why our methods had difficulty with binning is that the results from CF method are known to often be slightly compressed compared to the true data. This phenomenon can be observed from Table 4.4, where the standard deviation of our predicted results were always less than the true standard deviation (0.2092) of the complete data matrix. Compressing the data is not desirable when we have a cut-off near one of the extremes of the data. As mentioned earlier, we tried using a sigmoid probability function as the cut-off to solve our binning issues . The probability function reflected our uncertainty towards, which bin values near the cut-off should fall into. Using the probability function, over the hard cut-off, gave us many fewer miss-bins among the values we could confidently place but, in exchange, it gave us many values that we could not confidently place (as seen in Table 4.5). This binning problem is hard to deal with because it is too reliant on absolute values and arbitrary cut-offs.

On the other hand, the images generated from our results are a visualization of the relative differences between the predicted values. These differences do not depend on hard cutoffs and are not prone to error from slightly compressed results. In the Dr. Eric Brown paper [3], they were able to draw meaningful conclusions about biological molecules by using their expert knowledge to study a heat-map (an image) of their experimental results. These factors imply that one way to use of our methods, from a biologist's standpoint, is to study an image of the results for desirable patterns. In this application the relative, not the absolute, values are of interest and therefore binning is not necessary.

Chapter 5

Conclusions

The goal of this project was to create an algorithm that could predict all results of a biological experiment given only a subset of the results in question. In an effort to achieve our goal, we tried using algorithms made to complete matrices given only a subset of the entries. We tried 4 different methods: two Collaborative Filtering methods, the Nyström method and LMDS. Both the Nyström and LMDS methods failed to give meaningful results. These methods failed most likely because the data used for this project violated assumptions critical to the functioning of the methods. Fortunately, the CF methods fared better. The Memory-Based CF method performed well enough to be viable choice for our application but it seemed to struggle with over-fitting our data. Of all the methods, the Matrix Factorization CF method was the most successful. Despite the success of the CF methods, they still struggled to make sure the results fell on the proper side of the 0.25 cut-off. To mitigate this problem a sigmoid probability function was used as the cut-off, instead of the original binary one. Future work for this project includes: trying our methods on other sets of biological results, and exploring the idea of experiment design (intelligently choosing which experiments to perform and which to predict so that we may improve the results from our prediction method).

References

[1] John S. Breese, David Heckerman, and Carl Kadie. Empirical analysis of predictive algorithms for collaborative filtering. pages 43–52. Morgan Kaufmann, 1998.

[2] A. Farahat, A. Ghodsi, and M. Kamel. Greedy nystrom approximation. Technical report, University of Waterloo, 2010.

[3] M.A. Farha and E.D. Brown. Chemical probes of escherichia coli uncovered through chemical-chemical interaction profiling with compounds of known biological activity. *Chemistry & Biology*, 17(8):852 – 862, 2010.

[4] S. Funk. Netix update: Try this at home, aug 2011. http://sifter.org/ simon/journal/20061211.html.

[5] Netflix Inc. Netflix prize homepage, aug 2011. http://www.netflixprize.com/.

[6] C.R. Johnson. Positive definite matrices. *The American Mathematical Monthly*, 77(3):259–264, 1970.

[7] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Advances in Artificial Intelligence*, 42(8):30–37, 2009.

[8] John C. Platt. Fastmap, metricmap, and landmark mds are all nystrom algorithms. page 15. Microsoft Research, 2005.

[9] D. Reid, B.S. Sadjad, Z. Zsoldos, and A. Simon. Lasso - ligand activity by surface similarity order: A new tool for ligand based virtual screening. *Journal of ComputerAided Molecular Design*, 22(6-7):479–487, 2008.

[10] Vin De Silva and Joshua B. Tenenbaum. Global versus local methods in nonlinear dimensionality reduction. In *Advances in Neural Information Processing Systems 15*, pages 705–712. MIT Press, 2003.

[11] X.Y. Su and T.M. Khoshgoftaar. A survey of collaborative filtering techniques. *Advances in Artificial Intelligence*, 2009(421425):19, 2009.

[12] P. Tarazaga, B. Sterba-Boatwright, and K. Wijewardena. Euclidean distance matrices: special subsets, systems of coordinates and multibalanced matrices. *Computational and Applied Mathematics*, 26(3):415–438, 2007.

[13] Christopher Williams and Matthias Seeger. Using the nystrom method to speed up kernel machines. In *Advances in Neural Information Processing Systems 13*, pages 682–688. MIT Press, 2001.