Predicting Protein-Protein Interactions with Collaborative Filtering Algorithms

István Albert

Collaborative Filtering (CF) has proven to be a valuable tool for predicting relevant items in many different domains from e-commerce to information retrieval. In this paper we explore the applicability of CF to predicting protein-protein interactions by using the interaction patterns between first and second order neighbors in the interaction network. We investigate three different prediction strategies, describe the algorithms that support the prediction process and evaluate the performance of the methods on the *Saccharomyces cerevisiae* interaction network containing 3360 proteins and 13669 interactions. We find that for a large number of proteins the methods perform surprisingly well, on average being able to correctly predict missing interactions in 20% of cases! The most significant factor affecting the quality of predictions appears to be the number of interactions already known about a given protein. It is equally intriguing that this prediction performance is achieved *without* using any kind of prior biological knowledge during the prediction process. We believe that with a judicious use of such knowledge the quality of the predictions could be greatly improved.

The explosive growth of the World Wide Web and the emergence of E-Commerce have led to the development of *recommender systems* [1-4]. These systems provide people with personalized knowledge discovery mechanisms that learn the users' preferences from past choices and/or explicitly expressed opinions. Collaborative Filtering (CF) is one such widely used recommender system that works by finding the preference neighborhoods that best describe a given person. CF may also be considered an *unsupervised, instance-based learner* that first needs to be presented with a large number of training cases and then, in the prediction phase, matches the input to one or more instances in the training set.

Since CF methods are widely used in e-commerce (Amazon, Launch) there is a significant body of work addressing the fundamental challenges of their *accuracy* and *scalability*. Great progress has been made in the past few years and the algorithms and methodologies available today can generate high quality predictions based on millions of entries in less than a second. Still these prediction methods have not yet been used in bioinformatics, mostly because the *Shopper-Item-Rating* mapping that all CF methods rely on does not seem to lend itself to problems faced in biology. Yet on closer inspection we can easily see that the *Shopper* relation describes the framework in which the *Items* are being evaluated, while the *Rating* field above can be thought of as an expression of the relevance (appropriateness) of the *Item* for the *Shopper*. This discovery, coupled with the understanding of how the prediction process takes place, allows us to apply the metaphor of "people shopping around for items" to the biological domain where "proteins choose which other proteins they prefer to interact with". The algorithms that we chose to evaluate for this paper are *model-based top-N* methods that can build an internal prediction model with the so-called *user-item*, *item-item* or *probabilistic* approaches. We define the term *prediction* to mean the result of a process that produces a certain number of candidates. We will consider a prediction to be correct if one or more of the candidates match a known, relevant item. To evaluate our methods we follow the

standard methodologies used in evaluating machine learning algorithms. We will create training and test sets for each of our three methods and we will attempt to predict the items in the test set based on the items in the training set. As we will show later, the amount of available data greatly affects the performance of the algorithms. Therefore we aimed to reduce the size of the test set and raise the number of cross validations. Due to the highly heterogeneous nature of the underlying network it would be very difficult to estimate the statistical errors caused by the limited number of cross validations. We therefore chose the method that is often referred to as "*all-but-one*" where we placed a single interaction in the test set and then used the all the remaining data as the training set. For each method we repeated the process 13699 times to cover every possible interaction. We will note here that the interactions are symmetrical with respect to the proteins. Thus each removal may be independently predicted from both proteins thus for each method we were able to generate 27398 predictions. We define the word *basket* as the collection of items that are used as the seed for the prediction. In our mapping for example, if we were to remove the interaction between protein **P** and protein **Q** we would have two possible baskets, all the remaining interactions of **Q** and all the remaining interactions of **P** . The training data would consist of all the interactions with the exception of the link between **P** and **Q**. During the prediction process we would attempt to predict the link between **P** and **Q** based on the training data and by using both baskets in turn. In our terminology both **P** and **Q** are *users* and predicting the interaction between them means determining whether **Q** is an *item* of **P** or **P** is an *item* of **Q**.

We'll now describe the algorithms used during the prediction process. Let's consider a space with N users and M items. In the *user-item* scenario an M dimensional vector characterizes every user, where each index corresponds to a single item. The value of the vector at an index is the rating (relevance) expressed by the user on the item in question. In most cases the distribution of items across users is highly non-linear and the users tend to rate only a very small subset of the available items, thus making these rating vectors very sparse. The *user-item* prediction process takes a basket and searches the training set for a neighborhood. If found, this neighborhood will contain those rating vectors that are most similar to the basket. Then as the final step the neighborhoods are aggregated to generate the most likely candidates. Both the neighborhood formation process and the aggregation step have been extensively studied and a wide variety of similarity distances and weighting strategies have been employed. For this study we are using the cosine metric, where the similarity between two vectors is expressed as the cosine of their angle in the M dimensional space. Items are weighted with this same metric, by summing these weights across all neighbors.

The *item-item* method operates on the same rating matrix as the previous method. In this method, however, it is the items that are grouped into neighborhoods based on the N dimensional vector representing the users that considered the item relevant. Aggregating the most similar items corresponding to the items in the basket generates the predictions. We used the same cosine similarity as before as our distance metric. Finally the *probabilistic* approach uses a similarity measure proportional to the conditional probability of an item being present. In particular the conditional probability that protein that interacts with **P** also interacts with **Q** is the ratio of the number of proteins that

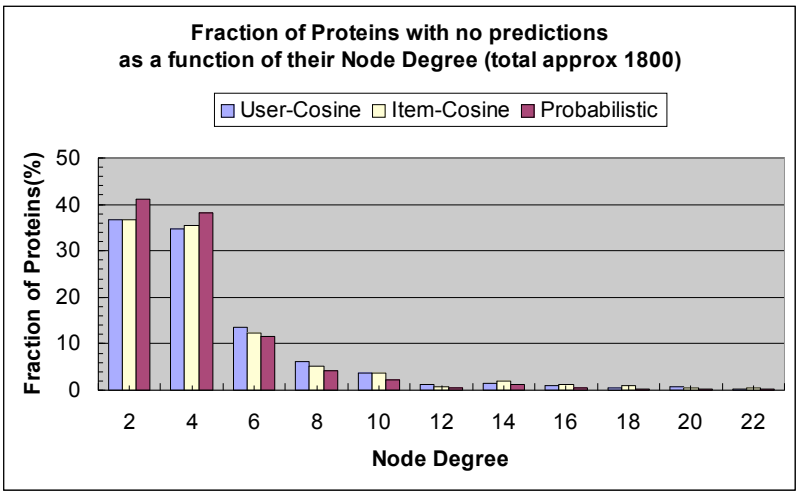
interact with both **P** and **Q** and the number of proteins that interact with **P**. The formula while simple is subject to various artifacts and limitations. Numerous corrections and normalizations need to be employed to address these. The exact details regarding the implementation of these in our methods can be found in [3].

Studies show that depending on the values of *N* and *M* as well as the distribution of ratings across them there may be significant performance gains between methods, but the absolute accuracy of them remains approximately the same. Most intriguingly, it has been also shown that whenever multiple correct (relevant) answers exist, these different methods will return notably different subsets of the relevant candidates. Each method is more sensitive to a different kind of symmetry present in the relations that are being studied. Our goal is to gain some insight into how well these methods work and also to infer rules and patterns within the subsets that they return.

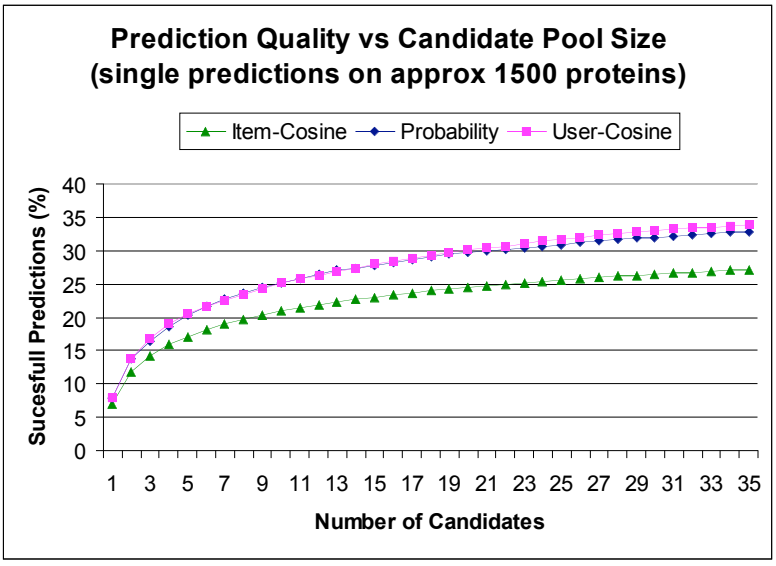
We obtained the protein-protein interaction data from DIP (Database of Interacting Proteins) containing a total of 4716 proteins and 15116 interactions among them. If we were to represent these interactions as a network where the nodes are the proteins and the edges represent an interaction between two proteins then the resulting network will exhibit a scale free property. A consequence of this property is that there are a large number of proteins that have only one interaction (low degree). Since our study focuses on discovering interaction pattern conservation between proteins these singly connected items cannot possibly be predicted with the methods that we have set out to evaluate. We therefore chose to remove these from our network. As the removal process may affect the connectivity of other proteins it needs to be performed in an iterative manner until no more changes are needed. By the end of this process we had 3360 proteins and 13669 interactions left, with every protein participating in at least two interactions. Since the protein interaction symmetrical, in the actual representation the total number of interactions was 2*13669=27388. The next step was to map the CF framework onto this interaction network.

Every protein in the network interacts with two or more proteins that in turn interact with at least one more protein. This allows us to consider every protein as a “user” that has selected a number of “items” that happen to be other proteins. In our case we can assume the mere presence of an interaction means that it has biological relevance, therefore we consider it as a positive rating. Thus this problem has been mapped from a biological problem of protein interactions to the E-commerce problem where each protein is treated as a shopper that has already “bought” a number of other proteins and is looking “to buy” another one. The most obvious question one faces is why would an e-commerce method be applicable to a biological problem. The immediate answer to it is that none of these methods have any built-in knowledge of the domain that they are being used for. If the pattern of buying a DVD player and DVDs is observed, that is not because the system was coded to identify the connection between the two. What the system recognizes is a non-explicit manifestation of this connection over multiple transactions. Similarly, the affinity of a number of proteins towards each other may have various explanations, yet the CF algorithms will be able to identify missing elements without knowing what was the reason that these proteins interact with each other.

To quantify the quality of the CF algorithms we have first set out to verify the ability of the algorithms to predict a missing interaction from a training set that is missing only the interaction that will be predicted.

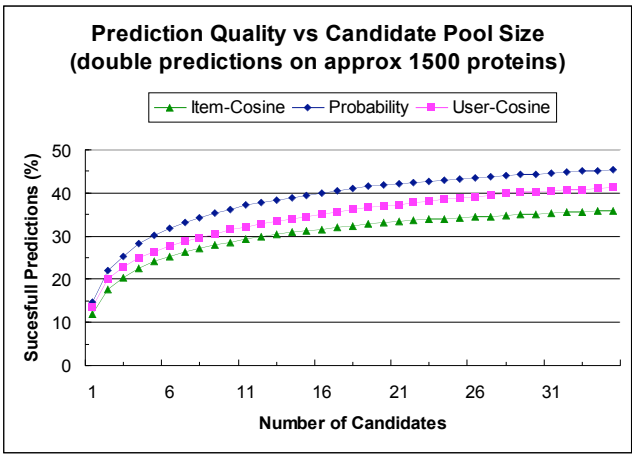


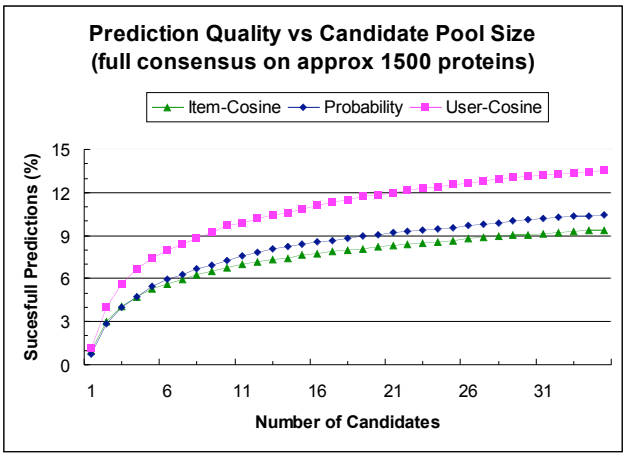
After running our programs we found that we were able to generate at least one correct prediction for only 44% of the proteins. The methods that we employ critically depend on the connections within the interaction network. The fewer connections the less likely is to be able to infer further information. In the figure at the left we show that the ability to generate at least one correct prediction strongly correlates with the node degree (the number of interactions a protein participates in) with the vast majority of the "unpredictable" proteins having less than 5 interactions. We note that there is a good number of low degree proteins that can be predicted for, so a low connectivity does not necessarily preclude a protein from being correctly identified. The percentage across methods does not vary significantly therefore we report an average number of 1800 to convey the order of magnitude of the numbers involved.



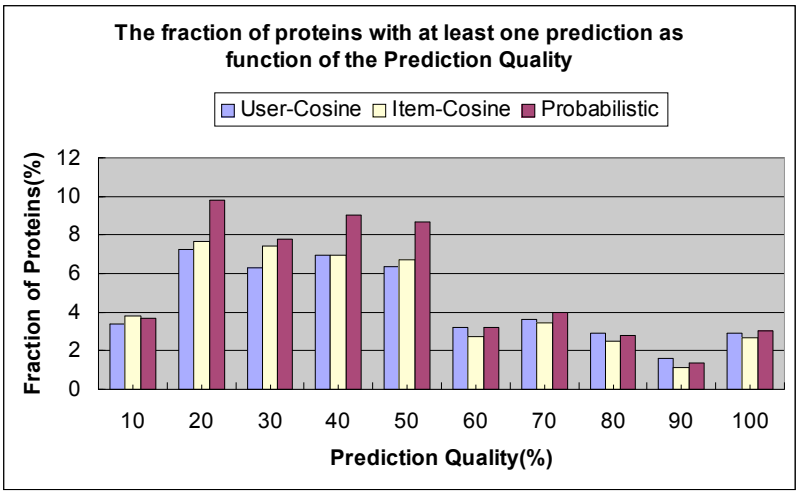
In this next chart we show a comparison of the performance of the three methods on the proteins that we could correctly predict at least once (approximately 1500) as a function of candidate number. We defined a prediction to be successful if the removed protein was among the candidates generated as a prediction. For every method the candidates are ranked by their estimated relevance, thus higher order candidates are expected to be less accurate. As one might expect an increasing number of candidates increases the rate of success at the expense of generating false positives. But we note that the rate of increase is highly nonlinear, tapering off at

high number of candidates. What this means is that the methods are accurate enough to produce their best candidates in the first few returns. The plots show that even with a single candidate the prediction process is able to identify a missing interaction 8% of the time. The frequency of the correct guesses allowing only the first five candidates is more than 20%!



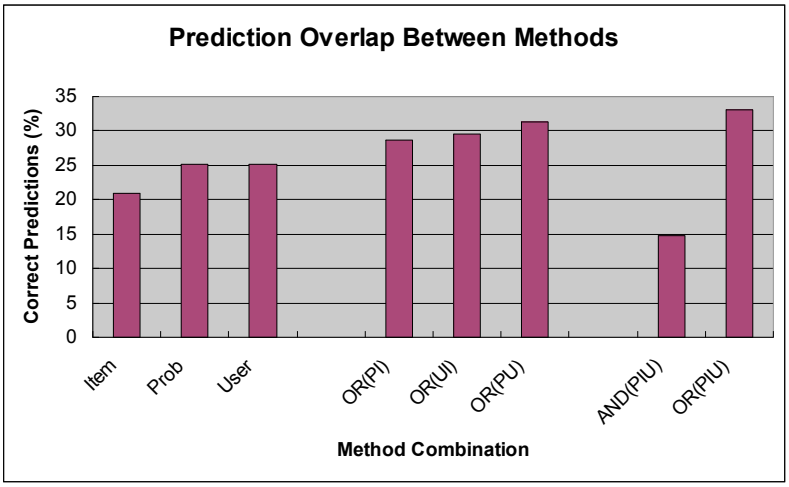


Since predictions can be generated from the baskets corresponding to both proteins we chose to also display the ability of the methods to correctly identify a missing interaction if any one of the baskets predicted it (double predictions) or whenever both baskets predicted it (full consensus). We note that the *user-item* method appears to significantly outperform the other two in the full consensus mode, while in other instances the *item-item* based method appears to be the least effective.



We now turn our attention to analyzing the distribution of correct predictions across the entire set of proteins. Specifically, we are interested inferring what makes certain proteins more suited to be predicted for and whether the distribution of the predictions can help us identify some groups of proteins. Each of the three methods appears to produce similarly shaped distributions, with the probabilistic approach performing much better at lower qualities. The graph above shows the number (as percentage) of proteins that could be predicted with a given quality. We define the quality to be the percentage of successful predictions for a given protein based on the removal and prediction of every interaction in turn. For a protein with 10 interactions there will be 10 predictions based on baskets that are each missing a different interaction.

If five of these predictions are correct the prediction quality for this protein will be 5/10, that is 50%..



A way to quantify the differences between methods is to analyze the overlaps between the candidates generated by the different methods. The chart on the left displays the percentage of the correct predictions for a 10-candidate prediction strategy for various combination methods. For example the **OR(PI)** label specifies the case where a prediction were considered to be correct if either of the two methods *probabilistic* and *item-item* generated a correct prediction for any given interaction. The increase in the number of correct prediction will be directly proportional with the correct prediction produced by one method but missed by the other. As we can observe the *probabilistic* approach is best augmented by the *user-item* yet even a full consensus among all methods can find the missing interaction in 15% of the cases. A more precise comparison of the differences between these methods would involve an in depth analysis of the *sensitivity* and *selectivity* of each method.

In conclusion, we have applied a group of recommender systems used in e-commerce to a novel domain. Early results suggest that the method has the potential to be a valuable addition to other bioinformatics methods. We were able to correctly predict a high percentage of the interactions in a protein interaction network. Further studies are needed to identify the underlying mechanisms that promote or hinder the applicability of the methods. We plan to make use of results published in recent Bayesian interaction prediction studies [6] to incorporate more prior knowledge at the prediction phase.

References:

1. B. M. Sarwar, G. Karypis, J. A. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In Proceedings of the 10th International World Wide Web Conference (WWW10), Hong Kong, May 2001

2. Sarwar, B. M., Karypis, G., Konstan, J. A., and Riedl, J. "Analysis of Recommender Algorithms for E-Commerce". In proceedings of the ACM E-Commerce 2000 Conference. Oct. 17-20, 2000, pp. 158-167

3. Suggest Recommendation Engine: http://www-users.cs.umn.edu/~karypis/suggest/index.html

4. McNee, S., Albert, I., Cosley, D., Gopalkrishnan, P., Lam, S.K., Rashid, A.M., Konstan, J.A., & Riedl, J. (2002). On the Recommending of Citations for Research Papers. In Proceedings of ACM 2002 Conference on Computer Supported Cooperative Work (CSCW2002), New Orleans, LA, pp. 116-125

5. A gene recommender algorithm to identify coexpressed genes in C. elegans. OwenAB, Stuart J, Mach K, Villeneuve AM, Kim S., Genome Res. 2003 Aug;13(8):1828-37.

6. A Bayesian networks approach for predicting protein-protein interactions from genomic data., Jansen R, Yu H, Greenbaum D, Kluger Y, Krogan NJ, Chung S, Emili A, Snyder M, Greenblatt JF, Gerstein M., Science. 2003 Oct 17;302(5644):449-53