CS224W Project Report

Neighborhood-based Rating Prediction and Inter-network Robustness

1. Introduction

Product review and rating by customers are common features in many e-commerce systems, including Amazon and eBay. Customers use ratings to express their opinion about the products they purchased. Thus, e-commerce platforms can use product ratings to rank the product search results and to improve their recommendation system. As a result, the products with good ratings tend to rank higher in the search result as well as being more likely to be recommended to customers. Despite the effectiveness of this practice, new or less popular products are unfavored due to a lack sufficient number of reviews, which then results in a rich-get-richer phenomenon. This phenomenon would be offset if the ratings of the under-reviewed products can be predicted. In this work, we attempt to predict the ratings of these under-reviewed products using community clustering[1], RolX[2], and node2vec[3], and evaluate their efficacy using multi-label classification error as detailed in Section 4. Furthermore, we investigate whether motif similarity of two co-purchased product networks implies rating similarity of products with similar community or role on both networks.

In this study, we use the Amazon product rating data[4], the characteristics of which are described in Section 3. As illustrated in Fig.1, we model each product as a node, and products purchased by the same customers are deemed to be connected. We define these products to have been "co-purchased", disregarding potential time gap between the purchases. While customers typically leave reviews only for a fraction of their purchases, we believe that a co-reviewed product network can be used to approximate a co-purchased product network. This belief holds under the following assumptions: the ratio between the number of ratings and the number of purchases for every product is the same, and there are insignificant number of fake reviews by non-purchasers.

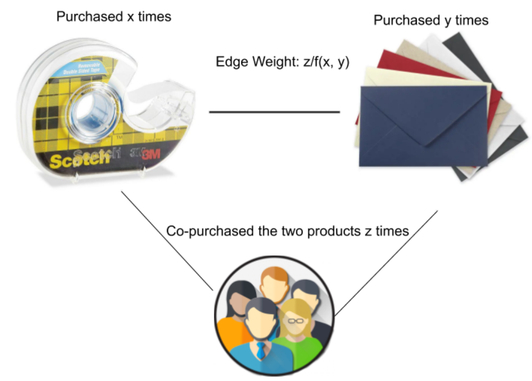

Fig. 1 Co-purchase example

We hypothesize that a product and its neighbors in a co-purchased network receive similar ratings. For example, a fan of an author may buy all of the author's books and rate most of them highly. We also hypothesize that products with similar roles in the co-purchased network receive similar ratings. There are products that often serve as complementary, but secondary items to different purchases. For example, wine and cheese are often bought as complementary items when planning meals regardless of recipes. We test our hypothesis in two main ways in this study. First, we will predict the ratings of under-reviewed products based on other reviewed products that are co-purchased by creating neighborhoods with community clustering[1], RolX[2], and node2vec[3]. Second, we will expand on the idea and predict the ratings of products in an unreviewed product network based on another fully-reviewed product network.

2. Related Works

The traditional ways of predicting online product ratings typically include incorporating text messages of the reviews[5][6]. However, when processing new products or inactive products with no associated reviews, these methods would be mostly limited. Additionally, the information about users is also commonly used in predicting the rating of products[5][6], but there are also concerns with user privacy. To retain privacy, some reviews are provided anonymously and thus the signals from users will be missing. There are also attempts at predicting the ratings based on computer vision, matching different product categories. However this approach is limited to products with rich visual elements[7][8].

The relationship of products that are co-purchased has not been fully explored. When user information and text reviews are missing, we propose to build a network of products based onco-purchasing information. Our project framework and intuitions are based on 4 highly relevant papers: “Fast unfolding of communities in large networks”, “RolX: Structural Role Extraction & Mining in Large Graphs”, “node2vec: Scalable Feature Learning for Networks”, and “Network Motifs: Simple Building Blocks of Complex Networks” [1][2][3][9]. For brevity in the rest of this proposal, these papers are referred by the following names - Louvain paper, RolX paper, node2vec paper, and motifs paper, respectively.

The Louvain [1] and RolX [2] papers present two distinct methods of defining neighborhoods - by a node’s membership within a community based on edge connections and by its structural role within the network. node2vec [3] paper presents a method of considering both of these through a balance of exploration and exploitation. Furthermore, it provides a framework upon which machine learning algorithms can be applied, and has the added benefit of providing robustness against incorrect edge and label data. Finally, motifs paper[9] gives insight on finding similar networks based on their basic building blocks.

There is difficulty in determining whether a prediction is accurate because ratings are continuous, rarely uniform in distribution, and have relative ordering. As further discussed in Section 4, work by Gupta et al.[10] and Chen et al.[11] provide insight on how to define prediction accuracy.
Since co-purchased networks may have weighted or multiple parallel edges, the creation of a null model could not be done naively. Carstens and Horadam[12] provide a mathematical derivation of edge-switching methodology to create unbiased null models and Fosdick et al.[13] provide pseudocode of the switching model implementation.

3. Datasets and Preprocessing

We worked with Amazon product review datasets[4], which consists of dataset for many different categories of products. We used the “Office Products” category as the base dataset, on which we tuned our clustering algorithms to maximize prediction accuracy. As part of data preprocessing, we sampled the network, calculated fractional edge weights, and binned ratings and co-purchasing frequencies.

3.1. Co-Purchased Network Creation

The dataset contains <reviewer, product> pairs along with the rating given by the reviewer. This represents a bipartite graph, which we projected to <product, product> undirected network. We created both unigraph and multigraph to calculate the weight of each edge. Additionally, we collected aggregated rating per product.

The original dataset contains ~**1.2M reviews** with ~**130K unique products** and ~**910K unique reviewers**. The <reviewer, product> bipartite graph contains ~45K disjoint components with largest connected components containing **88% of all nodes**. We focus our experiments on the largest connected component given the majority of the nodes are contained in this component. Projection of the largest connected <reviewer, product> component to a <product, product> graph resulted in **~80K nodes** and **~1M edges** with average edge weight of **1.23** for the multigraph. Fig.2 illustrates histogram of number of product reviews with 20 bins. The rightmost bin count products with greater than 20 reviews for the largest connected components. Mean number of review is **6.579** while median is **3.0**. There is a high discrepancy in mean and median because number of reviews have long-tail values with the largest number of reviews as high as 4398. Fig.3 illustrates that the per product mean rating is skewed to higher range with an average of **4.034**, indicating the necessity for better rating discrimination as discussed in Sections 3.3.

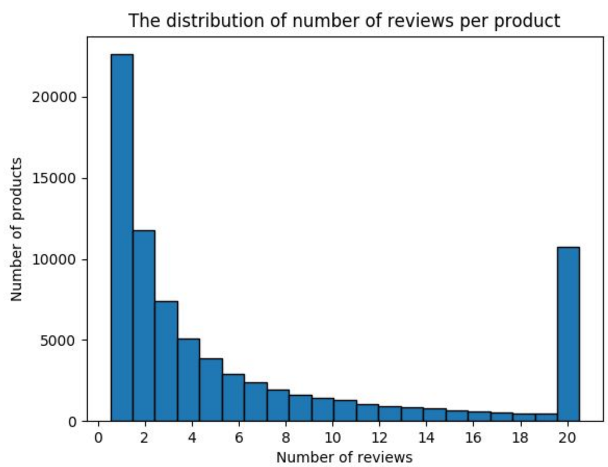

Fig.2 Per product review counts histogram

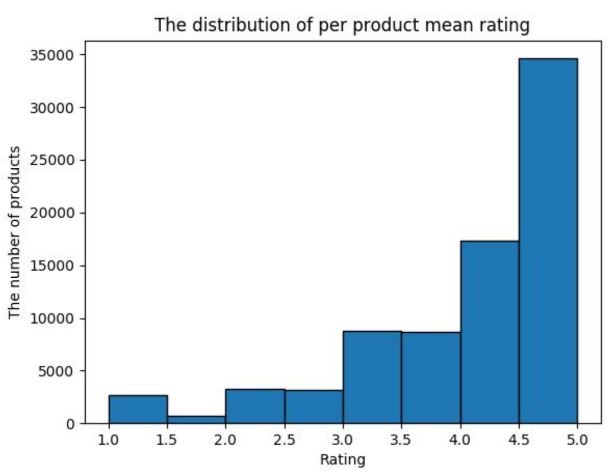

Fig.3 Mean rating per product histogram

3.2. Fractional Weight Network

Using the frequency of two products being co-purchased does not appropriately represent the proportion of times the products are co-purchased. For example, it cannot distinguish the following two scenarios: each product is purchased 1000 times but co-purchased 10 times vs each product is purchased just 10 times but always purchased together. We created fractional edge weight from the co-purchasing frequency and the number of product reviews of both products to deal with this issue.

We define the fractional weight of edges with the following equation:

$$\frac{cpCnt(P1, P2)}{f(pCnt(P1), pCnt(P2))}$$

where cpCnt(P1,P2) = co-purchase count of P1 and P2; pCnt(P) = purchase count of P. We explored different f(x, y) to calculate the most effective fraction weight such as sum, min, and square root of product.

3.3. Randomly Sampled Network

3.3.1. True rating

Since our work focuses on predicting ratings for under-reviewed products, it is important to define a sensible ‘true’ rating for such products to validate our predictors. However, an under-reviewed product, by definition, would have few reviews and these reviews could exhibit high variance. Even worse, if there is just a single review, the product’s ‘true’ rating would be at the mercy of that review. To circumvent this issue, we take a random subset of the data as described in Section 3.3.2 during the creation of the co-purchased network. We define the ‘true rating’ of a product to be the rating of the product based on **the original dataset**. By getting the rating of each product from the original data but creating the network and its edge weights only on the subset of data, we obtain ‘under-reviewed’ products with low-variance ratings. The rating aggregation method is discussed in Section 4.2.

3.3.2. Random sampling

First, **we dropped products with less than 3 reviews** to filter noisy data. Fig.4 on the next page shows the histogram of product with 3+ reviews with less skewed rating distribution compared to the original dataset (i.e. Fig. 3). Next, we **randomly sampled 10% of reviewers**, created the bipartite graph from the reviews of sampled reviewers, and projected the largest connected component of this induced subgraph.

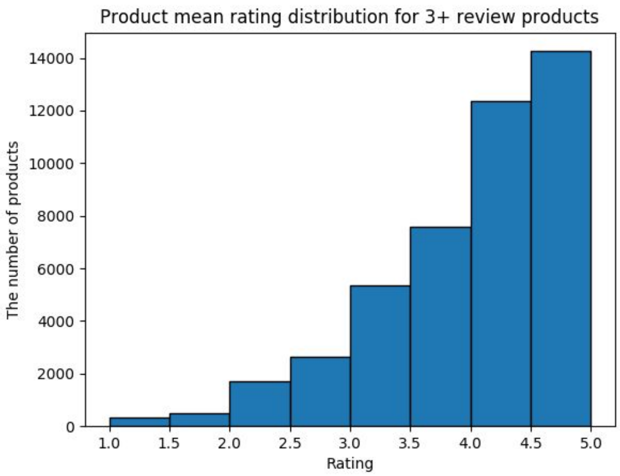

Fig.4 Mean rating histogram for 3+ review products

We make the assumption that the sampled graph is representative of the original graph in subgraph structures such as neighborhoods. Various simple analysis of the original and sampled graphs suggested the correctness of our assumption. For example, graph diameter, 90% effective diameter, and clustering coefficient remained similar between the original graph and sampled graphs of different sampling rates. Furthermore, as illustrated in Fig.5, we observed a linear relationship between the sampling rate and the sampled graph's number of nodes/edges. Finally, we know that motif occurrences will remain approximately constant relative to network size as noted in the motif paper[9].

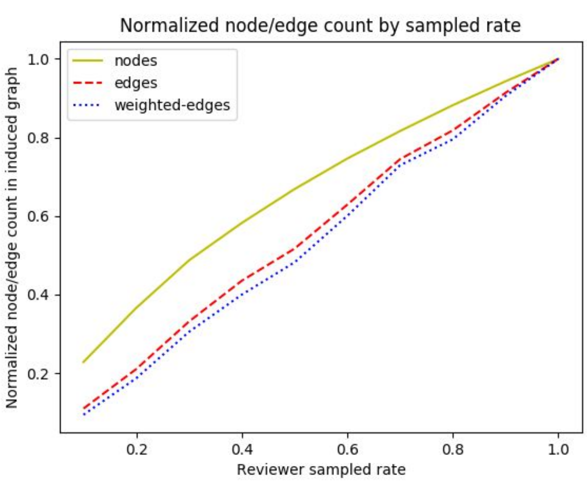

Fig.5 Reduced graph node/edge rate by sample rate

4. Evaluation and Methods

4.1. Prediction Error

Since the ratings are discrete, ordered, integer values between 1 and 5 inclusive, the most obvious definition of prediction error is the sum of squared errors between the predicted ratings and the true ratings, i.e. $\sum_{i=1}^{n}(R_{t_i} - R_{p_i})^2$. However, this definition is problematic in potentially causing the predictor to avoid values at the extremes of the range. Furthermore, since ratings of products are clumped above 4.5 as seen in Fig.4, always returning a prediction around 4.5 may result in minimum error.

We define the prediction error as a multi-label classification error. This eliminates the concept of extremes in the rating range, but at the cost of losing the relative ordering of ratings. However, the work by Gupta et al.[10] also on rating prediction showed that the classification approach out-performed both linear regression and multi-threshold ordinal regression. A potentially better error definition we considered is the listwise ranking loss defined in equation 3 of Chen et al.[11]. We decided that this is much more advanced on machine learning side and out of scope of this course. Multi-label classification is also suitable for handling the skewed distribution of ratings. Instead of discretizing ratings by rounding to the nearest integer,we put products into bins of the same size according to their rating. Intuitively, this is fine since ratings are ordinal and the relative rating matters more than absolute rating. We found that 3 bins resulted in an even distribution of sizes; larger number of bins did not work due to tens or hundreds of products having the same average rating.

In determining bin ranges, we initially considered binning all ratings rather than average product ratings (i.e. a product with 10 ratings contributes more to bin size than a product with 3 ratings). However, we realized this will not work as there are only 5 valid rating values, each rating appearing tens of thousands of times. Thus, we tried a 2-pass system: create initial bins based on average product ratings, then create final bins based on aggregate product bin values. Here, an aggregate product bin value was found by binning each rating of a product into the initial bins, then averaging the selected bin numbers. We ended up simply using the initial bins on average product ratings because two methods were different less than 1% of the time.

4.2. Intra-network Analysis

We test our hypothesis of the predictive power of neighbors by predicting the ratings of products within a partially-labeled product network. Specifically, we define products with just 1 rating in the sampled co-purchased network as the **under-reviewed products**. Note that by construction as per Section 3.3.1, these products have 'true ratings' which are based on 3 or more ratings.Then, depending on the neighborhood definition we use, we mask out some or all of the ratings of the under-reviewed products. This allows us to evaluate the performance of our models in predicting the ratings of under-reviewed products. For each of the methods, the **aggregate rating** of neighbors is defined as the mode rating bin of the neighbors. For example, if a neighborhood has 10 nodes in rating bin 0 (representing poor reviews), 20 nodes in rating bin 1, and 30 nodes in rating bin 2 (good reviews), then the mode bin would be bin 2. In the case that multiple bins had the same number of products, we selected one of the bins at random to keep the selection unbiased. Note that the rating bin of under-reviewed products in the neighborhood are not included when finding the mode.

4.2.1. Homophily

We apply Louvain algorithm [1] to our co-purchased product network, forming some k communities once the algorithm converges. For each community C_k, we assign a rating R_k which represents that community's aggregate rating. We mask out all under-reviewed products' ratings in this method. For each under-reviewed product P_i belonging to some community C_j, the predicted rating for that product R_i is simply R_j. We applied the Louvain algorithm on both the weighted and unweighted Office Product network.

4.2.2. Role

We apply RolX algorithm [2] to our product network. We initially used simple egonet features as the basic feature set: degree of node, number of edges within the egonet, and the number of edges leaving the egonet. We normalized the base features by dividing each element by the max absolute value seen in the entire graph - this ensures that no single feature dominates the cosine similarity. We then explored other features to decrease prediction error: product purchase count, number of neighbors (different from degree since multigraph), average number of edges per neighbor, and the aggregate of node's edges' fractional weights. We also explored recursion depth in creating the feature vector.

To evaluate the prediction error, we mask out 20% of the under-reviewed products' ratings. To predict the rating of an under-reviewed product, we calculate the product's role's cosine similarity against that of other labelled products. We find 20 products with the highest similarity and a similarity value of at least 0.975, and use the aggregate of their ratings in predicting the under-reviewed product's rating.

4.2.3. node2vec

We apply node2vec algorithm[3] on the product network to get node embeddings, which does not rely on product ratings. Similar to the role method, in predicting a product's rating, we find 100 products with the most similar embeddings and used the mode rating bin as the prediction. We used 20% of under-reviewed products to find the return parameter p and in-out parameter q which minimize the multi-label classification error, and used a different 20% of under-reviewed products to determine the performance of this node2vec approach.

4.3. Inter-network Analysis

We extend the work in Section 4.2. to the prediction of ratings for one product category based on a different product category. For example, suppose that for three product networks with categories of "automotive", "outdoors", and "cooking", "automotive" products are fully rated while the other two product categories are not. We might find that "automotive" and "outdoors" product networks have similar motif significance profile while "automotive" and "cooking" networks do not. Thus, we will use the "automotive" network ratings to predict the ratings of "outdoors" products. We refer to the labeled network as the 'training' network and the unlabeled network as the 'test' network. Note that we create a test network by simply masking out all rating information from a labeled network.

4.3.1. Motif analysis

We deem two co-purchased networks to be similar if their motif significance profile vectors have high cosine similarity. To make the detection of motifs tractable, we bin the fractional edge weights of the product networks into 3 groups - "usually purchased together", "often purchased together" (less frequently than 'usually'), and "sometimes purchased together". This gives 16 possible 3-node motifs as shown in Fig.6, where edge **u** denotes "usually purchased together", edge **o** denotes "often purchased together", and edge **s** denotes "sometimes purchased together".

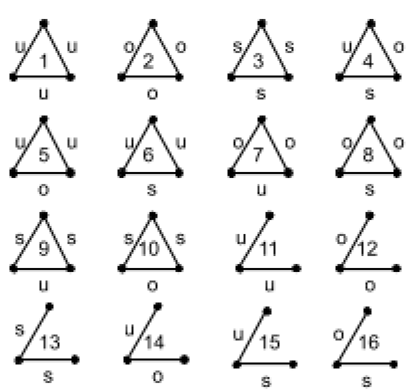

Fig.6 Co-purchasing frequency motifs

To determine the expected mean and standard deviation in the number of each motif, we create 10 null models per product category. The null model should reflect the likelihood of a pair of products being co-purchased if co-purchasing behavior is random. Thus, we keep constant the number of products and the number of times each product is purchased (i.e., constant degree) while we randomize the pairs of co-purchased products. This means the null model should be created from the unbinned multigraph where the edge weight represents the number of times products are co-purchased. Then, the null model can be preprocessed like the original co-purchased network to create frequency-binned networks.

We cannot use configuration model nor simple edge switching to create our null models, as these methods do not allow for multiple edges. Even if we do not reject multiple edges using these methods, uniformity across possible null models is not guaranteed[12]. Based on the work by Carstens and Horadam[12] as well as the pseudocode by Fosdick et al.[13], we implemented modified edge-switching to create unbiased null modes. In particular, for a randomly chosen pair of edges, a switch is accepted with an acceptance probability inversely proportional to the product of the edges' weights.

4.3.2. Homophily

As in Section 4.2.1., we apply Louvain algorithm to the test network. For each product P_i belonging to some community C_j in the test network, we try to find a similar community C_k from the training network. We will naively define two communities to be similar if they have similar clustering coefficient. Then, the predicted rating for P_i will be the aggregate rating of community C_k.

4.3.3. Role

As in Section 4.2.2., we extract features for the test network using the same set of base features. For a product in the test network, we find 20 products in the training network with the highest cosine similarity. We use the aggregate rating of these 20 training network products to predict the rating of the test network product.

4.3.4. node2vec

node2vec algorithm was run on the test network to find the product embeddings. In this case, the validation set for use of hyperparameter tuning was selected from the test network rather than the training network (i.e. Office Products). Similar to Section 4.2.3., a test network product's rating were predicted based on similar products from the training network.

5. Results and Discussions

We discuss our results as prediction rate or prediction accuracy, which is simply (1 - prediction error).

5.1 Motif analysis

The cosine similarity of motif significance profiles between Office products and other products are shown in Table.1.

Product	Auto	Baby	Beauty	Food	Pet	Toys
Cosine sim.	0.893	0.985	0.48	0.54	0.96	0.847

Table.1 Cosine similarity of products to office products

Baby products had the highest cosine similarity, so the Baby product co-purchased network is chosen as the test network for inter-network analysis.

As seen in Fig.7, motifs 9, 13 and 15 are overrepresented in both Office and Baby products. These three motifs are (s,s,u), (s,s) or (s,u) - both co-purchase networks have an overrepresentation of products being sometimes purchased together or usually purchased together, but not often (between sometimes and usually).

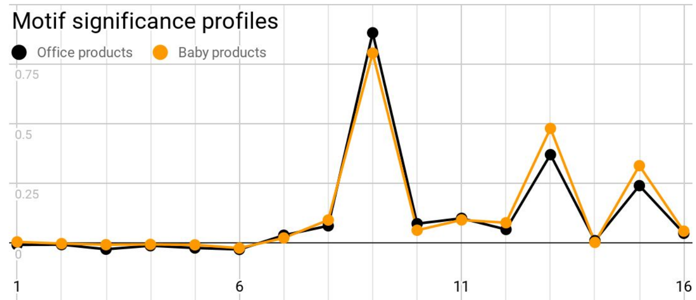

Fig.7 Motif significance profiles

Note that motif analysis was also run on 3-node motifs with 2 frequency bins (7 distinct motifs) and 4-node motifs with 2 frequency bins (62 distinct motifs). The 3-node, 2 frequency motifs did not provide enough discrimination between networks. 4-node motifs were very computationally expensive and had very few occurrences for majority of the motif types in both our dataset and null model, thus this was not pursued.

5.2. Homophily

Fig.8 shows the distribution of the size of the communities detected by the Louvain algorithm on Office products. Table.2 illustrates the edge weighting methods we used and the accuracy of the corresponding prediction. The prediction accuracy of the unweighted network was **0.413**, which is better than random, which would be 0.33 with 3 rating bins. The prediction accuracy indicates the communities that products belong to have an impact on the ratings of products. There was no significant difference in prediction accuracy between weighted and unweighted networks, nor between different weighting mechanisms. Both weighted and unweighted office product networks have similar distributions of predicted ratings.

Weight method	unweighted	min	sum
Prediction rate	0.413	0.416	0.420

Table.2 Louvain algorithm prediction accuracy

We used the same intra-network prediction method on Baby product network. The prediction accuracy for baby products on the unweighted network was **0.443**, which is more accurate than the office product network. Fig.9 shows the distribution of the size of the communities detected in the baby product network. The higher prediction accuracy for baby product network is because, dissimilar to the community distribution in the Office Product network, which has many small communities composing only 2 or 3 nodes, the distribution of baby products is close to normal distribution.

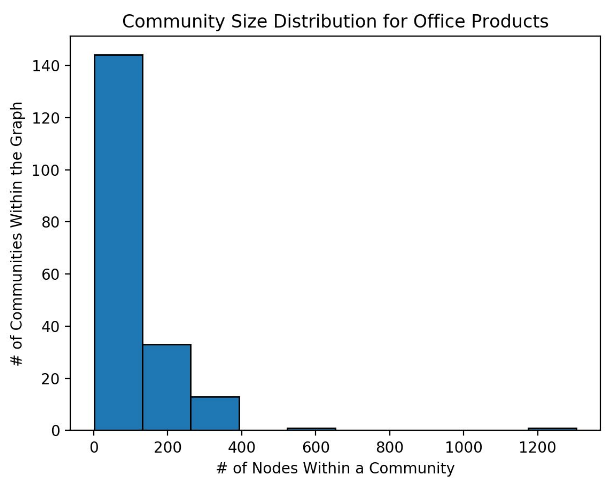

Fig.8 Office product community size distribution

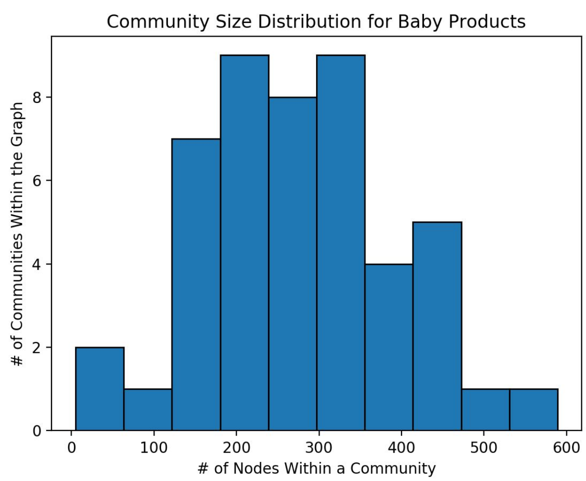

Fig.9 Baby product community size distribution

For Inter-network analysis, we calculated the clustering coefficient for each community in the office product network and the baby product network. We found ten pairs of communities (one from each network) with the most similar clustering coefficient. We used the most frequent rating bin in every ten communities in the office product network to predict the most frequent rating bin in corresponding communities in the baby product network. Out of **10 pairs** of communities, **three pairs** were a correct prediction, which is close to random prediction. Though office product network and baby product network have high motif similarity, that does not mean they have similar communities.

5.3. Role

Table.3 summarizes the prediction rate of role-based clustering for different base features. Base set refers to degree of node, number of edges within the egonet, and the number of edges leaving the egonet as defined in Section 4.2.2.

Feature set	**Base set**	**Set 1**	**Set 2**		**Set 3**	
Recursion depth	3	3	2	3	2	3
Prediction rate	0.356	0.358	0.383	0.371	0.367	0.404

Table.3 Prediction rate on Office products

In feature set 1, features related to absolute purchase count were added: the number of purchases of the product, the number of neighbors, and the average number of co-purchases with a neighbor. This did not improve the prediction rate compared to the base set; both base set and set 1 have just slightly higher than random prediction rates.

In set 2, the average number of co-purchases with a neighbor **minus 1** and the number of purchases of the product were added to the base set. The reason for not adding some features from set 1 is that many of them were highly correlated when we looked at the recursive features. The value 1 was subtracted from the average because we recognized that in a co-purchased network, every pair of products is co-purchased at least once. The subtraction increased the discrimination ability significantly and resulted in a prediction rate increase.

Finally, set 3 adds in the frequency of co-purchase on top of set 2. Specifically, we take the logarithm of the product of the node's frequency edge weights to maximize discrimination between different values. We also tried the average frequency edge weight, but that gave much poorer results. With set 3, recursive feature extraction of depth 3 gave better results than using depth 2 - this suggests that the role of a node in a larger subset of the graph is better at finding nodes with similar true ratings. Given that the diameter of our network is around 6 and that higher recursion depth incurs a great computational cost, we did not try out higher recursion depths.

RolX was ineffective at prediction ratings across networks - finding Office products with high role similarity to Baby products resulted in **0.335** prediction rate. From looking at the predicted rating bin versus the actual rating bin shown in Table.4, it is clear that a large portion of Baby products had high role similarity to a small portion of office products. This means that role similarity does not work across networks.

Rating bin	**[0, 3.92)**	**[3.92, 4.41)**	**[4.41, 5.0]**
Predicted count	132	188	454
Real count	253	235	256

Table.4 Predicted rating bin for Baby products

5.4. node2vec

In node2vec, our best sampling strategy was with low *q* value (=**0.3**) and *p* value (=**0.9**). Min fraction weight edge performed prediction better than unweighted by 3% (~0.415) and sum fraction weight edge by 4.5% (~0.405) with train set accuracy of **0.452**. Compared to high (i.e. >1) *p* and *q* values, prediction performed better by approximately 5%. Tuning other node2vec parameters (e.g. dimensions, number of walk) from default value provided by Grover et al.[3] did not significantly influence the prediction accuracy. Intra-network test set accuracy was **0.450** while inter-network test set prediction accuracy with the Baby product network had result of **0.306**.

The node2vec prediction, which indirectly combiness DFS (i.e structural roles of nodes) and BFS (i.e., network communities), had the best accuracy out of the three methods we experimented. The best sampling strategy with low *q* value indicates that structural equivalence had played a more significant role than the homophily despite the fact that our results in Sections 5.2. and 5.3. did not indicate significant difference. Given that we observed role mattered more than community in node2vec, we may be able to achieve better prediction with RolX with different feature selection or increased recursion depth. We may have ended with low prediction rate for inter-network analysis because we identified the similarity based on local structure (i.e. motify) of the network.

6. Conclusion

Table.5 summarizes the tuned prediction rates of different predictors using intra-network and intra-network analysis.

	Louvain	**RolX**	**node2vec**
Intra-Network	0.413	0.404	0.450
Inter-Network	0.3	0.335	0.306

Table.5 Summary of prediction rates

For all prediction methods for intra-network analysis, we see prediction rates that are meaningfully higher than random prediction rate of 0.333. This indicates that the ratings of products in the neighborhood of an under-reviewed product can give a hint on its own rating.

All prediction methods indicated that neighborhood clustering to predict ratings does not work across networks - at least not when networks are chosen based on motif similarity. Even though Office and Baby product co-purchase networks had similar motif significance profiles, other network characteristics seemed very different. With homophily, we observed that the community size distributions are very different, as shown in Fig.8 and Fig.9. With roles, a large portion of Office products had a similar role as a small portion of Baby products, as shown in Table.4.

7. Contributions

annezhen - homophily
mattskl - roles, motifs
yamamura - dataset preprocessing, node2vec

8. References

[1]V. Blondel, J. Guillaume, R. Lambiotte and E. Lefebvre, "Fast unfolding of communities in large networks", *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2008, no. 10, p. P10008, 2008. Available: 10.1088/1742-5468/2008/10/p10008.

[2]K. Henderson, B. Gallagher, T. Eliassi-Rad, H. Tong, L. Akoglu, D. Koutra, L. Li, S. Basu, and C. Faloutsos, “RolX: Role Extraction and Mining in Large Networks,” Sep. 2011.

[3]A. Grover and J. Leskovec, “node2vec: Scalable Feature Learning for Networks,” *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD 16,* 2016.

[4]J. McAuley, “Amazon product data,” Amazon review data. [Online]. Available: http://jmcauley.ucsd.edu/data/amazon/. [Accessed: 17-Oct-2019].

[5]B. Wang, B. Chen, L. Ma and G. Zhou, "User-Personalized Review Rating Prediction Method Based on Review Text Content and User-Item Rating Matrix", *Information*, vol. 10, no. 1, p. 1, 2018. Available: 10.3390/info10010001.

[6]F. Li et al., "Incorporating reviewer and product information for review rating prediction", *Proc. 22nd Int. Joint Conf. Artif. Intell.*, pp. 1820-1825, 2011.

[7]W. Ye, Y. Zhang, W. Zhao, X. Chen and Z. Qin, "A Collaborative Neural Model for Rating Prediction by Leveraging User Reviews and Product Images", *Information Retrieval Technology*, pp. 99-111, 2017. Available: 10.1007/978-3-319-70145-5_8 [Accessed 12 November 2019].

[8]J. Qiu, C. Liu, Y. Li and Z. Lin, "Leveraging sentiment analysis at the aspects level to predict ratings of reviews", *Information Sciences*, vol. 451-452, pp. 295-309, 2018. Available: 10.1016/j.ins.2018.04.009 [Accessed 12 November 2019].

[9]R. Milo, “Network Motifs: Simple Building Blocks of Complex Networks,” *Science*, vol. 298, no. 5594, pp. 824–827, 2002.

[10]N. Gupta, G. Di Fabbrizio, P. Haffner, "Capturing the stars: Predicting ratings for service and product reviews", *Proceedings of the NAACL HLT 2010 Workshop on Semantic Search. Association for Computational Linguistics*, pp. 36-43, 2010.

[11]W. Chen, T. Liu, Y. Lan, Z. Ma and H. Li, "Ranking Measures and Loss Functions in Learning to Rank", *Advances in Neural Information Processing Systems*, vol. 22, pp. 315-323, 2009. [Accessed 12 November 2019].

[12]C. Carstens and K. Horadam, "Switching edges to randomize networks: what goes wrong and how to fix it", *Journal of Complex Networks*, p. cnw027, 2016. Available: 10.1093/comnet/cnw027 [Accessed 2 December 2019].

[13]B. Fosdick, D. Larremore, J. Nishimura and J. Ugander, "Configuring Random Graph Models with Fixed Degree Sequences", *SIAM Review*, vol. 60, no. 2, pp. 315-355, 2018. Available: 10.1137/16m1087175 [Accessed 4 December 2019].