A LATENT FACTOR MODEL FOR BOARD RECOMMENDATIONS IN PINTEREST

A Thesis

by

VARDHARAJ LAKSHMINARASIMHAN

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Chair of Committee,	James Caverlee
Committee Members,	Yoonsuck Choe
	Ann McNamara
Head of Department,	Dilma Da Silva

December 2014

Major Subject: Computer Science and Engineering

ABSTRACT

The past two years have seen the rise of a new online social network – Pinterest – which has grown more rapidly than any other social network (now reaching 70 million users). Pinterest is primarily organized around photos (or "pins"), where users reveal their interests via organizing pins into self-assigned categorical boards. However, one of the key challenges for new and existing users of Pinterest is to find boards of interest from the overall collection of 750 million boards. Hence, this thesis focuses on the problem of *board recommendation* in Pinterest towards identifying personalized, high-quality boards without requiring exhaustive search or browsing by the user. Board recommendation in Pinterest is challenging for a number of critical reasons: (i) Unlike community-oriented recommenders for movies, books, and other media, boards are highly personalized and not viewed or rated by many others. (ii) Many pins and boards lack descriptive text and other features that are typically used to power modern recommenders. (iii) Finally, evaluating the quality of a Pinterest board recommender is difficult, since there are no baseline nor ground truth recommendations of Pinterest to compare against

With these challenges in mind, this thesis proposes a new latent factor model for generating Pinterest board recommendations. To tackle the feature sparsity and personal boards challenges, the overall approach generates ratings for every user-board pair which is then fed to a latent factor model which factorizes the sparse matrix to give ratings for unrated user-board pairs and the top rated boards form the recommendation list. Two of the key components of the proposed latent factor model are the (i) definition of the universe of users around each target user for identifying candidate boards to recommend; and (ii) the approach for assigning implicit ratings

to each user-board pair for this universe of users (as the basis of the latent factor model). For the first component, we investigate three universe types: a collection of randomly selected users, a collection of users in the target user's personal Pinterest network, and a collection of users who are "similar" to the target user. For the second component, we construct ratings via three approaches: a board-count method, a category-based method, and and LDA-based method. We investigate these design choices through a comprehensive set of experiments over a dataset of around 50,000 Pinterest users, 100 million pins, and around 570,000 boards.

ACKNOWLEDGEMENTS

I would like to thank my advisor, James Caverlee, for his continued guidance for the work I have done so far. He has helped me proceed in the right direction and helped me structure my work. I would like to express my gratitude to my thesis committee members, Yoonsuck Choe and Ann McNamara for their advice and support. I would also like to thank the members of infolab for helping and advising me throughout the course of my work.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

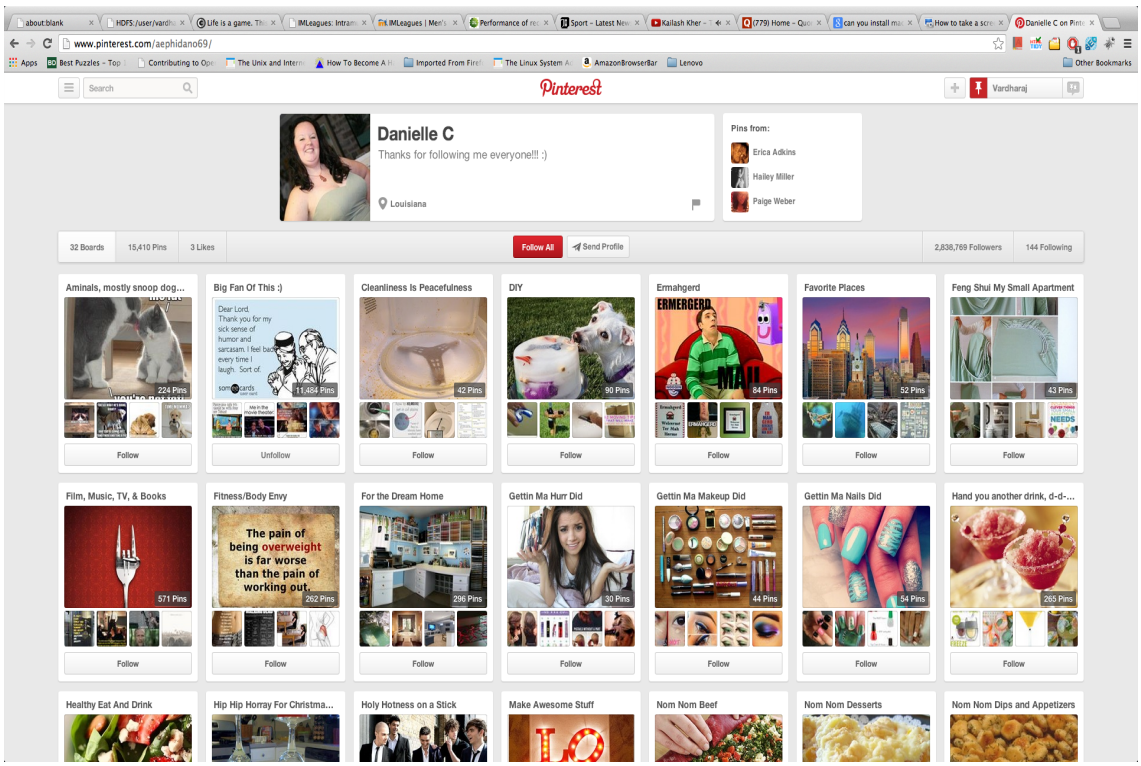
TABLE Page

1. INTRODUCTION

The past decade has witnessed the rapid adoption and increasing impact of large-scale online social networks. Facebook, Twitter, Google+, and LinkedIn have become central venues for connecting with friends, sharing opinions, discovering new media, finding jobs, and so forth. And in particular, the past two years have seen the rise of a new online social network – Pinterest – which has grown more rapidly than any other social network (now reaching 70 million users). Pinterest is primarily organized around photos (or "pins"). While many social networks have incorporated photo sharing, Pinterest is distinguished by a number of unique properties: (i) These pins are often centered around items with high potential commercial value (e.g., expressing purchase desires); (ii) Pinterest users reveal their interests via organizing pins into self-assigned categorial boards; and (iii) Pinterest is noted for its self-expression and personalized style, as expressed through these boards. To illustrate, Figure 1.1 shows a sample Pinterest user page consisting of a personal collection of pins and boards. This example user has over 30 self-organized boards including "DIY (do-it-yourself)", "Favorite Places", and "Make Awesome Stuff".

One of the key challenges for new and existing users of Pinterest is to find boards of interest. There are around 750 million boards on Pinterest, with around 100,000 being created every day. Currently, Pinterest supports board discovery via searching or by browsing. These methods are inherently time-consuming and ill-suited to the task of identifying interesting boards for individuals. For example, a search-based method requires a user to formalize an underlying information need into a query that is understood by the system, but these queries can be difficult to construct since the aspects of an interesting, personal board are often not easily understood even for the

Figure 1.1: Example Pinterest user page



individual. Alternatively, a browsing-based method requires a user to sift through thousands of boards, at great time and expense.

Hence, this thesis focuses on the problem of *board recommendation* in Pinterest towards identifying personalized, high-quality boards without requiring exhaustive search or browsing by the user. The goal is to recommend a set of new boards $\{b_1, b_2, b_3, b_4, ...\}$ to the user given the user u_1 and her set of existing boards $\{b_{u11}$, b_{u12}, b_{u13}, $b_{u14}, ...\}$.

However, board recommendation in Pinterest is challenging for a number of critical reasons:

- **Personal Boards.** First, boards are fundamentally personal, often with only a single user who has interacted with the board. In contrast, many recommender

approaches that have proved popular for movies, books, and other media [8] assume that the domain of the recommender is inherently *community-oriented*; that is, movies on Netflix may be viewed and rated by a community of 1,000s of users, revealing strong patterns that may link similar movies and similar users. Since boards are highly personalized and not viewed or rated by many others, how can we build and evaluate a good recommender?

- **Feature Sparsity.** Second, many pins and boards lack descriptive text and other features that are typically used to power modern recommenders. And for pins that do contain some descriptive text, in many cases the text consists of smileys, slang, or other highly-personal content that may not be suggestive of the pin or helpful for linking pins across users. In addition, [3] observed that 40% of boards lack a user-assigned category label, further exacerbating the problem of feature sparsity.

- **Lack of Ground Truth.** Finally, evaluating the quality of a Pinterest board recommender is difficult, since there are no baseline nor ground truth recommendations of Pinterest to compare against. As a result, any method toward recommending boards will need to additionally tackle the evaluation challenge.

With these challenges in mind, this thesis proposes a new latent factor model for generating Pinterest board recommendations. To tackle the feature sparsity and personal boards challenges, the overall approach generates ratings for every user-board pair which is then fed to a latent factor model which factorizes the sparse matrix to give ratings for unrated user-board pairs and the top rated boards form the recommendation list. Two of the key components of the proposed latent factor model are the (i) definition of the universe of users around each target user for identifying candidate boards to recommend; and (ii) the approach for assigning implicit ratings

to each user-board pair for this universe of users (as the basis of the latent factor model). For the first component, we investigate three universe types: a collection of randomly selected users, a collection of users in the target user's personal Pinterest network, and a collection of users who are "similar" to the target user. For the second component, we construct ratings via three approaches: a board-count method, a category-based method, and and LDA-based method. We investigate these design choices through a comprehensive set of experiments over a dataset of around 50,000 Pinterest users, 100 million pins, and around 570,000 boards.

The rest of the thesis is organized as follows. Chapter 2 presents a literature review. Chapter 3 explains the problem statement in detail. Chapter 4 elaborates on each technique with appropriate motivation, including the dataset and results discussion. Chapter 5 concludes the work and describes future directions on Pinterest board recommendation.

2. LITERATURE REVIEW

Recommender systems are typically classified as either content-based, collaborative filtering based, or hybrid recommenders.

Content based recommenders try to recommend items similar to the items the user has liked in the past [8]. Systems employing a content-based recommendation approach analyze a set of documents or items previously rated by the user and build a user model based on the features of the item. This user model is now a user interest profile which can be adopted to recommend new items. The recommendation process consists of matching of item features against the user features. Thus a good user model generates good recommendations. This is employed not just in text based systems but also in multimedia based systems [17]. [16] enhances the basic vector space model for content based recommenders. Traditional vector space models with TF-IDF weighting is one of the most common ways to model the documents (users or items). The advantages of content-based systems are that these recommenders exploit solely the ratings of the user to build the user profile, and they don't need others to rate the item. Also, these systems are capable of recommending new items which aren't rated by anybody.

Collaborative Filtering (CF) is considered to be the most important and widely used algorithm in recommender systems [8]. Collaborative filtering is most famously used by Amazon.com for its item recommendations where they recommend the next item to buy for a user. CF systems need to relate two fundamentally different entities: items and users [8]. There are two primary methods incorporating CF, namely Neighborhood models and Latent factor models. Neighborhood models focus on relationships between items or users. Latent factor models transform users and

items to a latent feature space which tries to explain ratings.

Hybrid recommenders [18] employ both of the above techniques to improve performance. Many methods are used to combine the two techniques like weighted average, feature combination, switching according to items, etc. Such combinations help overcome the challenges faced in both types of systems.

In the context of this thesis, we are motivated by the recent development of *social network* focused recommenders. In the context of social networks, recommender systems typically recommend users or content. [13] recommends users to follow in Twitter based on graph algorithms and a hubs and authorities based novel algorithm. [2] is another method that experiments with content-based and CF based methods for recommending Twitter users. They demonstrate how profiling of users generates high-quality recommendations. [11] works on recommending tweets (content) to the users. This method makes use of the content of the tweets as well as Twitter features like favorites and retweets for their recommendations. Some recommender systems like [9] and [10] use tweets to model the user using entity-based, hashtag-based, or topic-based strategies to generate personal news recommendations and [6] does a study on how to get your interests follow you on Twitter. [13] exaplains how Twitter serves recommendations at this scale, uses complex recommendation models and a good service architecture to deliver them. A recent work [14] employs recommendations to a completely different domain and displays which smartphone application to use next. These examples help understand the wide range and complexity of a recommender system.

Of particular interest to this thesis is research on the still emerging Pinterest community. [1] gives a statistical overview of Pinterest and describes what drives repins, likes, and so forth, [4] analyzes the role of gender in Pinterest. [3] tries to analyze how coherent user defined categories are to the actual content in a Pinterest board. And

yet, there is little if any work on recommendations in Pinterest. Recommendations for Pinterest can either be user recommendations or content recommendations (like boards or pins). [5] describes preliminary recommendation techniques using content-based and supervised recommendations. Recommendations systems at this scale are difficult to evaluate as well. [7] neatly sums up the various evaluation techniques used these days.

This motivates the need to attempt this problem using other approaches and experiment with the open data available. All the previous studies and the methods used motivate us to try it on a new network. The Pinterest recommendation problem is unique in itself and poses a lot of challenges. This thesis concentrates on combining content based approaches and collaborative filtering to generate high-quality board recommendations for a user.

3. PROBLEM STATEMENT

Consider a social network consisting of users $U = \{u_1, u_2, u_3 ...u_n\}$ where every user has boards $b_{ui} = \{b_{u1}, b_{u2}, b_{u3} \ ... \ b_{un} \}$. The problem of recommending boards B for a particular user u_k in the network is defined as: given user u_k and her set of boards $b_{uk} = \{b_{u1}, b_{u2}, b_{u3} \ ... \ b_{un} \}$, recommend a set of boards B that the user u_k is likely to *follow* and *add* to his collection of boards.

In the following section, we address this board recommendation problem and seek to address the following research questions:

- Pinterest offers a unique problem as the boards are highly personalized and not viewed or rated by many others. How can we build and evaluate a good recommender in such a constrained environment?
- How does the online neighborhood of the user influence her interests?
- Can we build a good user model using textual features alone?

4. PROPOSED SOLUTION

In order to generate board recommendations for Pinterest users, and because of the array of challenges this problem poses, I have formulated a latent factor model for generating Pinterest board recommendations. To tackle the feature sparsity and personal boards challenges, the overall approach generates ratings for every user-board pair which is then fed to a latent factor model which factorizes the sparse matrix to give ratings for unrated user-board pairs and the top rated boards form the recommendation list.

This chapter begins by explaining the latent factor model and then moves on to discuss two key components of the proposed latent factor model: the (i) definition of the universe of users around each target user for identifying candidate boards to recommend; and (ii) the approach for assigning implicit ratings to each user-board pair for this universe of users (as the basis of the latent factor model).

4.1 Latent Factor Model

In a typical recommender setup, user-item ratings are depicted in a matrix notation. This matrix contains ratings of each user and her boards. Note that the matrix will only contain values for a user and her **own boards**. This means that the matrix we generate is going to be a sparse matrix with many missing values. We wish to find given these ratings, how the user rates the other boards. The Latent Factor Model fits this problem description perfectly as it works even with the missing values. The Latent Factor Model fits the ratings in order to minimize the Root Mean Square Error (RMSE), a standard recommender system metric of choice. The top-n boards derived from the Latent Factor Model then form the recommendation list $\{ B_1, B_2, \ldots , B_n \}$. The top-n boards retrieved are not re-ranked for the final results. We are

trying to evaluate new techniques and the concept of universe, hence we leave rank based results for future work.

LFMs are well versed in the recommender systems literature. LFMs model a collaborative filtering problem by uncovering latent features from the matrix. These latent features are generated such that the observed ratings can be explained.

In information retrieval, SVD is the default method for uncovering the latent semantic factors. But, applying SVD to explicit ratings in the CF domain is not possible because of the matrix being very sparse. When the matrix has missing values, SVD is undefined. Trying to model the very few ratings leads to overfitting. So some earlier works used a technique called imputation, which makes the rating matrix dense by inputting the missing values. The drawback of imputation is the significant increase in data and being computation intensive. Also, inaccurate imputation might lead to faulty data. Hence, recent works focus on modeling only the obsered ratings avoiding the cons of previous techniques. [8].

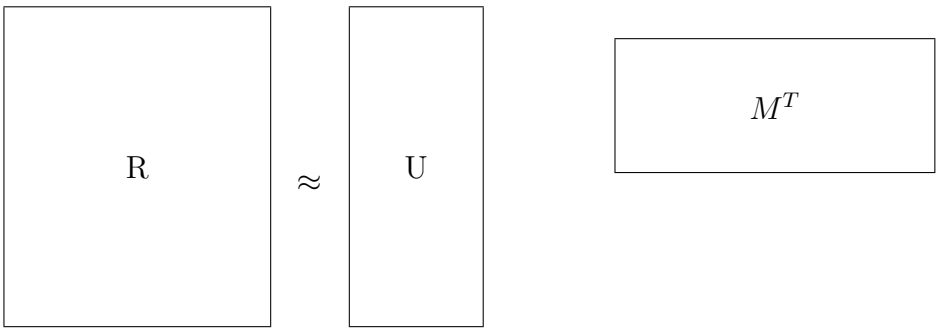
Matrix factorization models work by mapping both users and items to a joint latent factor space of dimensionality f. The user-item interactions are modeled in the latent space. The latent space infers user feedback and considers these latent factors while trying to explain the ratings. For example, when the products are movies, factors might measure dimensions such as comedy versus action, amount of drama or action, child friendly and other such dimensions.

Let num = # of pins in category of board b_j and $denom$ = # of pins of u_i For each user u_i and his boards b_j, we calculate the rating $r_{u_i b_j} = \frac{num}{denom}$. Now that we have the ratings, we can feed them to the Latent Factor Model in the form of a sparse matrix shown by Figure 4.1:

Figure 4.1: Sparse matrix of ratings

$$R = \begin{bmatrix} r_{u_i b_j} & r_{u_i b_{j+1}} & . & . & . & . & . \\ . & . & r_{u_{i+1} b_j} & r_{u_{i+1} b_{j+1}} & . & . & . \\ . & & & & & & \\ . & & & & & & \\ . & & & & & & \end{bmatrix}$$

Figure 4.2: Matrix factorization



Each item i is associated with a vector $q_i \in R_f$, and each user u is associated with a vector $p_u \in R_f$. For a given item i, the elements of q_i measure the distribution of interest in these factors. For a given user u, the elements of p_u measures the extent of interest the user has in items that possess the factors.

The dot product, $q_i^T p_u$, maps the interaction between user u and item i i.e., the total interest of the user in various characteristics of the item. The final rating is created by adding in the baseline predictors that take only the user or item into consideration. Thus, a rating is predicted by the rule:

$r_{ui} = \mu + b_i + b_u + q_i^T p_u$.

In order to learn the model parameters (b_u, b_i, p_u and q_i) we minimize the regularized squared error:

$$min_{b,q,p} \Sigma (r_{ui} - \mu - b_i - b_u - q_i^T p_u)^2 + \lambda (b_i^2 + b_u^2 + q_i^2 + p_u^2).$$

Thus after factorizing the matrix as shown by Figure 4.2, we can retrieve any user-item rating using the above equation. Given this basic setup, there are two key questions we must explore. First, how do we fill in the known user-board pair ratings in the first place? Since Pinterest does not support explicit ratings of boards, we need techniques to assign implicit scores toward driving the overall recommender. Second, what is the appropriate universe of candidate users around the target user to use for generating recommendations?

4.2 Assigning Implicit Ratings to User-Board Pairs

We consider three methods for assigning implicit ratings to user-board pairs:

4.2.1 Board Level Counts

In this method, we assess a user's interest in a board by the ratio of the number of pins he has in that particular board over her total pins.

Consider user u_1 and board b_1, and u_1 has P_{u1} pins overall of which p_{b1} are from

b_1. In this case user-board rating is generated as follows:

$r_{u_1 b_1} = p_{b1}$ / P_{u1}

The number of pins in a board gives an idea of which board he prefers the most. The ratings are fed accordingly to the matrix and recommendations are generated. This serves as a brute-force approach which serves as a baseline to compare the other approaches to.

4.2.2 Category based scoring

Pinterest allows users to assign categories to the boards they create. In this method, we try to assess the categorical interest of the user by self assigned categories to their boards. We generate ratings for the a user-board pair by number of pins in category of board/total pins. Thereby capturing the user's categorical interest by the categories provided by Pinterest. We feed the ratings to the matrix and apply latent factor model to generate the recommendations. The intuition behind this approach is the ratings reflect a categorical interest of the user therefore inspiring recommendations from similar categories. There are two main challenges with this approach, boards with same category will get the same score and there are numerous boards which don't have user-assigned categories or are assigned to "other" and "none" categories which don't provide any insight on the interests of the user.

Suppose a user u_1 has 5 boards $\{b_1, b_2, b_3, b_4, b_5\}$ which have respective categories $\{c_1, c_2, c_1, c_2, c_3\}$ and pin counts $\{p_1, p_2, p_3, p_4, p_5\}$.
Now the number of pins in category c_1 = number of pins in b_1 + number of pins in $b_3 = p_1 + p_3$

Thus, rating $r_{u_1 b_1}$ = number of pins in category c_1/ total pins = $p_1 + p_3$/ P

where P = $p_1 + p_2 + p_3 + p_4 + p_5$ To tackle these problems we come up with LDA-based topic discovery.

4.2.3 LDA-based category method

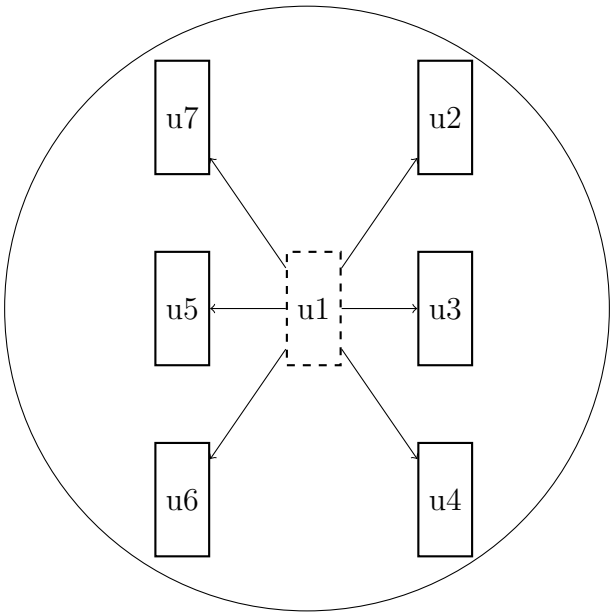
Instead of user assigned categories, we train an LDA model [15] with the training set of users and assume their labeling is appropriate. Using this trained LDA model, we label all the user's boards by modeling a board as a bag of words containing text from the pins of the board. We model the board as a TF-IDF vector resulting from the above vector. In LDA, a document (a single board) is viewed as a mixture of various topics. Individual words contribute to the probability of each topic, and labeling resulting by the most probable topic. As LDA is a supervised model, the initial training set is the user assigned category label to the boards. After the categories are assigned, we follow the same method as above to assign ratings to user-board pair and thus generate recommendations. This in addition to recommendations, also shows how accurately the users self assign categories to the boards. This will be helpful for further research relying on the user-assigned categories. We are not exploring image based features in this thesis mainly due to limited and incomplete data, but also because we are exploring rating based methods and trying to evaluate the role of the universe.

4.3 The Universe of Users

We now introduce the concept of *universe* around the user as the set of users $U = \{u_1, u_2, u_3, ...u_n\}$, used for the board recommendation problem for user u_1 depicted by Figure 4.3. For example, consider user u_1 who has followers $u_{followers} = \{ u_2, u_3 ... u_k\}$ and follows users $u_{following} = \{ u_{k+1}, u_{k+2}, ... u_n\}$. This set of users form the universe around user u_1, denoted $U = \{u_1, u_2, u_3 ...u_n\}$. Each user has boards $b_{ui} = \{b_{u1}, b_{u2}, b_{u3} ... b_{un} \}$. These boards form the universe of boards around the users which we will collectively call B. We take a strong motivation from the fact that the

universe moulds your interests. We experiment with three different types of universe:

Figure 4.3: Universe of users



- Random users: This serves as a benchmark for the techniques to improve upon.
- Network users: The actual user's network and its influence in helping the recommendation algorithm work its way.
- Similar users: Similar users possess similar interests. Thus putting them together yields a better universe thereby driving better recommendations.

All these techniques are applied on the 3 types of data I have collected and their performance is evaluated using techniques described below.

4.4 Dataset

The dataset that I am using for my research is crawler cached data from December 2012 to present. The crawler collects all the data available by scraping the HTML of the webpages of Pinterest in a breadth-first-fashion. The crawler was developed using Python and MongoDB. The data currently comprises of 47,998 users, 99,286,242 pins and 575,976 boards as seen in Table 4.1. As there are a lot of null users (that is, users with no boards or pins), for this thesis I am only considering users who have more than 10 boards and whose network has been crawled. This is because we need sufficient textual features for LDA model to work. Also, the reason we have different number of users for similar universe is that we cluster the users to form the similar user set hence needing a higher number of users.

Table 4.1: Dataset

	Users	Boards	Pins
Random	1000	27201	534,874
Similar	2000	81812	1, 527, 986
Network	2300	64941	1, 400, 670

4.5 Evaluation and Results

In this section we describe the metrics we use to evaluate our results, then share the experimental results of the methods described in the above section and assess their performance.

4.5.1 Metrics

Recall that the only ground truth available with this data are the user's boards themselves. As Pinterest develops its own recommendations, we will be able to evaluate against those results. We need to solidify our claim that our results are good, hence multiple methods are used. We adopt three evaluation metrics:

- **Root mean square error (RMSE).** RMSE is a standard evaluation technique that tells us how well the ratings are fit. For an input in the matrix form, RMSE = $\Sigma(r_{pred,i,j} - r_{i,j})^2$. Lower RMSE numbers indicate better fitting of data. Note, however, that RMSE alone is not an indicator of how good recommendations are. RMSE indicates how well the matrix was factorized and how well the ratings were fit but that alone doesn't evaluate the recommendation results.

- **Cosine similarity.** The recommendations generated are converted to TF-IDF vectors and compared with the user to ascertain how similar they are to the user. Cosine similarity is a standard technique for the same. Higher similarity numbers indicate better recommendations. Cosine similarity analyses the recommendation results instead of just evaluating the matrix factorization that RMSE provides. Also, as this system doesn't deal with end users, cosine similarity is a good way to judge whether the user will like the recommendations or not.

- **Precision and Recall over Held-Out Boards.** This idea again is a standard evaluation approach. The user under consideration has some rated boards already in the matrix. We add few of his boards to the matrix and make others rate it, but not the user himself. Then, when the recommendations

are generated, we calculate recall as the number of boards retrieved from the boards that were held back initially. We calculate precision as the number of such boards retrieved over the total number of recommendations generated.

Let's say the user u_1 has 10 boards $\{b_1, b_2, b_3, b_4, b_5, ..., b_{10}\}$ and we generate user-board ratings for only 5 boards $\{b_1, b_2, b_3, b_4, b_5\}$, the rest of the boards are rated by other random users. When these ratings are fed to the LFM, out of the boards $\{b_6, b_7, b_8, b_9, b_{10}\}$, the number of boards retrieved corresponds to the recall of the recommendation algorithm. Similarly, the ratio of number of boards retrieved to the number of such extra boards corresponds to the precision of the algorithm. Finally, the hiding boards evaluation technique is the robust technique which encapsulates most of the challenges mentioned above, but for a real users test which this system doesn't deal with. This system deals with new users by taking average of all ratings of the users present in the universe and assigns ratings to boards to generate initial recommendations.

4.5.2 Experiments

There are three parameters for the experiments described below namely *the implicit rating technique, the universe of users, and the number of recommendations.* All experimental results are preceded by the parameter description and the intuition behind the experiments. In all the experiments, the *boards* column represents the number of boards per user. These boards are rated by the users to which they belong. *Extra boards* represent the hidden boards which are *not* rated by the users they belong to, instead are rated by a random number of users. The extra boards are iterated over 1 to 10 and the metrics are averaged over these iterations.

1. **Method:** Board Level counts.

Dataset: Random users.

Number of recommendations: 20

This is the baseline method that I describe above. This method does not give any precision-recall measures for 10 recommendations, hence only the results for 20 recommendations have been depicted by the Table 4.2 below. Board Level counts is a naive rating technique which captures interest just by the number of pins in a particular board. Running this method on a Random universe is expected to perform the least when compared to other methods.

Table 4.2: Board Level counts - Random - 20

Users	Boards	Extra-Boards	Recall	Precision	RMSE	Similarity
100	2	1-10	0.315	0.17	8.27	0.081
	3		0.135	0.07	8.19	0.076
	4		0.135	0.07	8.34	0.078
200	3	1-10	0.002	0.002	7.36	0.076
500	3	1-10	-	-	7.49	0.067
1000	3	1-10	-	-	8.12	0.061

2. **Method:** Category based scoring

   Dataset: Random users.

   Number of recommendations: 10

   The Category based scoring makes a remarkable improvement over the baseline method as seen in Table 4.3, but as this is a random user dataset the scores

depreciate with increasing number of users. So for 1000 users, I cluster them and apply this method again which yields in good results. We start with 10 recommendations and then move on to 20 to see how the recall and precision metrics improve. Also, this helps for further work which would want to rerank a fixed number of recommendations.

Table 4.3: Category based scoring - Random - 10

Users	Boards	Extra-Boards	Recall	Precision	RMSE	Similarity
100	2	1-10	0.33	0.164	3.71	0.044
		3	0.411	0.206	5.03	0.047
		4	0.397	0.199	4.94	0.049
200	3	1-10	0.432	0.286	2.50	0.051
500	3	1-10	0.268	0.214	1.36	0.046
1000	3	1-10	0.0	0.0	1.5	0.050
1000-cluster	3	1-10	0.68	0.48	1.95	0.058

3. **Method:** Category based scoring

   Dataset: Random users.

   Number of recommendations: 20

   This method performs really well in comparison to the baseline method for the same number of 20 recommendations. Increasing the number of recommendations helps us in understanding the performance of the method better. We see here in Table 4.4 that even though the metrics have improved, the method

doesn't yield any precision-recall for 1000 users. We attribute this to two factors namely larger matrix to factorize and a Random universe. We see in later experiments that changing the universe leads to better results.

Table 4.4: Category based scoring - Random - 20

Users	Boards	Extra-Boards	Recall	Precision	RMSE	Similarity
100	2	1-10	0.67	0.395	2.14	0.082
	3		0.68	0.396	3.10	0.078
	4		0.66	0.383	2.62	0.079
200	3	1-10	0.491	0.279	2.10	0.071
500	3	1-10	0.378	0.281	1.17	0.080
1000	3	1-10	0.0	0.0	1.91	0.075
1000-cluster	3	1-10	0.67	0.47	1.95	0.072

4. **Method:** Board Level counts

   Dataset: Similar users

   Number of recommendations: 10

   In Table 4.5, we try the naive Board Level counts on similar user universe to see whether there's any improvement from the Random universe. Firstly, note that we are getting precision-recall measures even for 10 recommendations. Secondly, we can see good recall metric for 100 and 500 users, this means that these universe sets are more cohesive than the other sets.

Table 4.5: Board Level counts - Similar - 10

Users	Boards	Extra-Boards	Recall	Precision	RMSE	Similarity
100	3	1-10	0.354	0.247	7.80	0.041
270	3	1-10	0.044	0.026	7.759	0.056
350	3	1-10	0.035	0.026	8.13	0.045
500	3	1-10	0.308	0.208	8.12	0.056
700	3	1-10	0.0	0.0	8.36	0.024

5. **Method:** Board Level counts

   Dataset: Similar users

   Number of recommendations: 20

   This is an extension of the previous experiment with increase in the number of recommendations. We run this experiment to see how the improvement in precision-recall metrics with increase in number of recommendations. Another point to note is that even this experiment shows that the universe containing 100 and 500 users is more cohesive than the rest depicted by Table 4.6.

Table 4.6: Board Level counts - Similar - 20

Users	Boards	Extra-Boards	Recall	Precision	RMSE	Similarity
100	3	1-10	0.441	0.308	7.78	0.080
270	3	1-10	0.061	0.037	7.79	0.104
350	3	1-10	0.058	0.044	8.15	0.084
500	3	1-10	0.392	0.270	8.12	0.111
700	3	1-10	0.002	0.002	8.38	0.047

6. **Method:** Category based scoring

   Dataset: Similar users.

   Number of recommendations: 10

   In this experiment, we use Category based scoring technique on a similar user universe. We try to see how change of the universe affects the metrics. We see in Table 4.7 that this technique performs really well on the similar user universe. This supports the intuition that surrounding the user by similar users leads to better recommendations.

Table 4.7: Category based scoring - Similar - 10

Users	Boards	Extra-Boards	Recall	Precision	RMSE	Similarity
100	3	1-10	0.709	0.453	3.15	0.040
270	3	1-10	0.678	0.44	1.67	0.065
350	3	1-10	0.525	0.355	1.63	0.048
500	3	1-10	0.652	0.411	2.83	0.059
700	3	1-10	0.66	0.446	1.88	0.024

7. **Method:** Category based scoring

   Dataset: Similar users.

   Number of recommendations: 20

   This experiment deals with 20 recommendations with the same setup as above. In Table 4.8, we see an improvement in the metrics over the setup with 10 recommendations. As expected, more relevant boards are retrieved for 20 recommendations.

Table 4.8: Category based scoring - Similar - 20

Users	Boards	Extra-Boards	Recall	Precision	RMSE	Similarity
100	3	1-10	0.703	0.503	3.70	0.119
270	3	1-10	0.701	0.499	1.64	0.117
350	3	1-10	0.688	0.559	1.57	0.087
500	3	1-10	0.663	0.47	2.88	0.115
700	3	1-10	0.683	0.524	1.68	0.054

8. **Method:** Board Level counts

   Dataset: Network users

   Number of recommendations: 10

   The following experiments use network users as the universe. Network users contain a user's follower-following network. As the users themselves choose who they follow, and their followers tend to have similar interests, the network universe is naturally expected to be a cohesive universe. We start with the naive rating technique, Board Level counts on this universe. In Table 4.9, firstly we see precision-recall metrics recorded for 10 recommendations setup which could not be observed on a Random universe. Finally considering the performance of Board Level counts over the three types of universe, it performs best in the network universe scenario.

Table 4.9: Board Level counts - Network -10

Users	Boards	Extra-Boards	Recall	Precision	RMSE	Similarity
100	3	1-10	0.539	0.246	7.14	0.052
200	3	1-10	0.46	0.21	7.85	0.048
500	3	1-10	0.003	0.001	7.75	0.061
1000	3	1-10	0.0	0.0	8.24	0.055

9. **Method:** Board Level counts

   Dataset: Network users

   Number of recommendations: 20

This experiment as depicted by Table 4.10 increases the number of recommendations to see improvements in the metrics for 100, 200, 500 users. Still, this method is unable to generate precision-recall metrics for 1000 users.

Table 4.10: Board Level counts - Network - 20

Users	Boards	Extra-Boards	Recall	Precision	RMSE	Similarity
100	3	1-10	0.543	0.247	7.17	0.071
200	3	1-10	0.48	0.229	7.85	0.080
500	3	1-10	0.007	0.004	7.70	0.111
1000	3	1-10	0.0	0.0	8.30	0.780

10. **Method:** Category based scoring

    Dataset: Network users

    Number of recommendations: 10

    Here we experiment with the Category based rating technique on a network universe. Following this method's good performance for other universe types, it performs the best on select number of users over all types of universe as seen in Table 4.11. This makes it clear that the network of the user is the ultimate guide to a user model and generating good recommendations.

Table 4.11: Category based scoring - Network - 10

Users	Boards	Extra-Boards	Recall	Precision	RMSE	Similarity
100	3	1-10	0.75	0.348	4.26	0.056
200	3	1-10	0.636	0.299	4.32	0.061
500	3	1-10	0.465	0.199	4.68	0.054
1000	3	1-10	0.0	0.0	4.30	0.054

11. **Method:** Category based scoring

    Dataset: Network users

    Number of recommendations: 20

    We apply the same method for 20 recommendations and see improvement in the metrics in Table 4.12. Also, this method wasn't giving precision-recall metrics for 1000 users and 10 recommendations, but we can see in the following table that we get those metrics showing the effectiveness of the method.

Table 4.12: Category based scoring - Network - 20

Users	Boards	Extra-Boards	Recall	Precision	RMSE	Similarity
100	3	1-10	0.758	0.348	4.03	0.112
200	3	1-10	0.698	0.324	4.23	0.091
500	3	1-10	0.586	0.274	4.006	0.114
1000	3	1-10	0.126	0.075	4.878	0.085

12. **Method:** LDA topic based

    Dataset: Random users

    Number of recommendations: 10

    Here we explore the final rating technique which is the LDA category based scoring. We are trying to evaluate an external topic modeling algorithm on the pinterest data and compare it with the user defined categories in various scenarios. This experiment as seen in Table 4.13 runs this method on a Random dataset with 10 recommendations.

Table 4.13: LDA topic based scoring - Random - 10

Users	Boards	Extra-Boards	Recall	Precision	RMSE	Similarity
100	3	1-10	0.409	0.24	3.85	0.0492
200	3	1-10	0.37	0.203	2.98	0.052
500	3	1-10	0.31	0.188	3.41	0.044
1000	3	1-10	0	0	3.51	0.042

13. **Method:** LDA topic based

    Dataset: Random users

    Number of recommendations: 20

    This is an extension of the previous experiment described in Table 4.14 that increases the number of recommendations to 20 to see how the metrics improve.

Table 4.14: LDA topic based scoring - Random - 20

Users	Boards	Extra-Boards	Recall	Precision	RMSE	Similarity
100	3	1-10	0.645	0.388	2.87	0.116
200	3	1-10	0.503	0.297	3.15	0.113
500	3	1-10	0.367	0.289	2.40	0.086
1000	3	1-10	0	0	3.65	0.089

14. **Method:** LDA topic based

    Dataset: Similar users

    Number of recommendations: 10

    We experiment using LDA based scoring on the similar user dataset for 10 recommendations. Improved metrics are observed on this dataset in Table 4.15 as this is a more coherent dataset. We also observed improved metrics for larger number of users and are able to observe the effectiveness of the universe as well as the method.

Table 4.15: LDA topic based scoring - Similar - 10

Users	Boards	Extra-Boards	Recall	Precision	RMSE	Similarity
100	3	1-10	0.644	0.399	5.739	0.038
270	3	1-10	0.634	0.35	1.97	0.066
350	3	1-10	0.568	0.30	2.28	0.047
500	3	1-10	0.605	0.303	1.87	0.058
700	3	1-10	0.609	0.259	1.32	0.042

15. **Method:** LDA topic based

    Dataset: Similar users

    Number of recommendations: 20

    This is the same experimental setup as above but with 20 recommendations described in Table 4.16.

Table 4.16: LDA topic based scoring - Similar - 20

Users	Boards	Extra-Boards	Recall	Precision	RMSE	Similarity
100	3	1-10	0.704	0.407	1.06	0.081
270	3	1-10	0.67	0.44	1.947	0.117
350	3	1-10	0.665	0.392	2.035	0.087
500	3	1-10	0.635	0.399	1.740	0.117
700	3	1-10	0.615	0.370	1.209	0.087

16. **Method:** LDA topic based

    Dataset: Network users

    Number of recommendations: 10

    Now we employ this method on the network user universe and compare the performance to other algorithms on the same dataset in Table 4.17.

Table 4.17: LDA topic based scoring - Network - 10

Users	Boards	Extra-Boards	Recall	Precision	RMSE	Similarity
100	3	1-10	0.58	0.325	6.01	0.042
200	3	1-10	0.53	0.234	4.47	0.027
500	3	1-10	0.469	0.216	2.36	0.037
1000	3	1-10	0.01	0.01	3.45	0.046

17. **Method:** LDA topic based

    Dataset: Network users

    Number of recommendations: 20

    The Table 4.18 is the final experiment where we try to evaluate the effectiveness of this method by increasing the number of recommendations to 20.

Table 4.18: LDA topic based scoring - Network - 20

Users	Boards	Extra-Boards	Recall	Precision	RMSE	Similarity
100	3	1-10	0.625	0.388	3.87	0.117
200	3	1-10	0.603	0.297	2.15	0.112
500	3	1-10	0.567	0.289	2.47	0.089
1000	3	1-10	0.21	0.199	2.65	0.087

Figure 4.4: Recall versus extra-boards for boards = 2

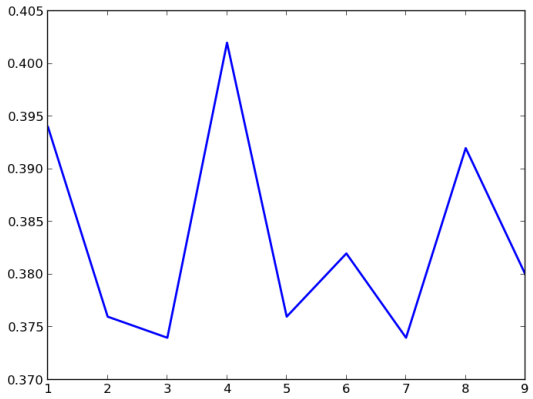


Figure 4.5: Precision versus extra-boards for boards = 2

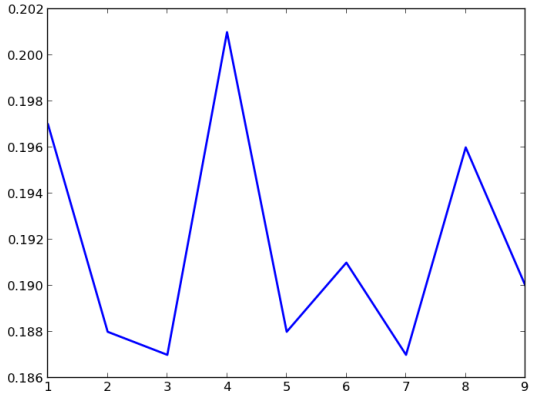


Figure 4.6: Recall versus extra-boards for boards = 3

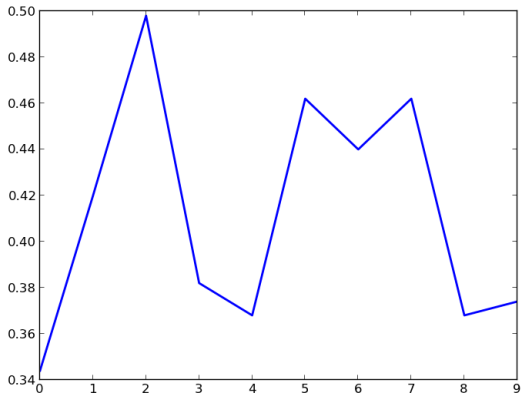


Figure 4.7: Precision versus extra-boards for boards = 3

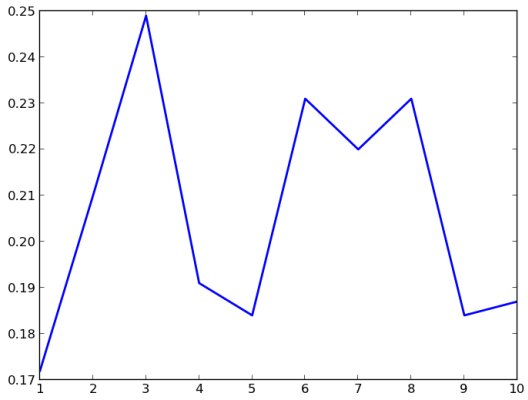
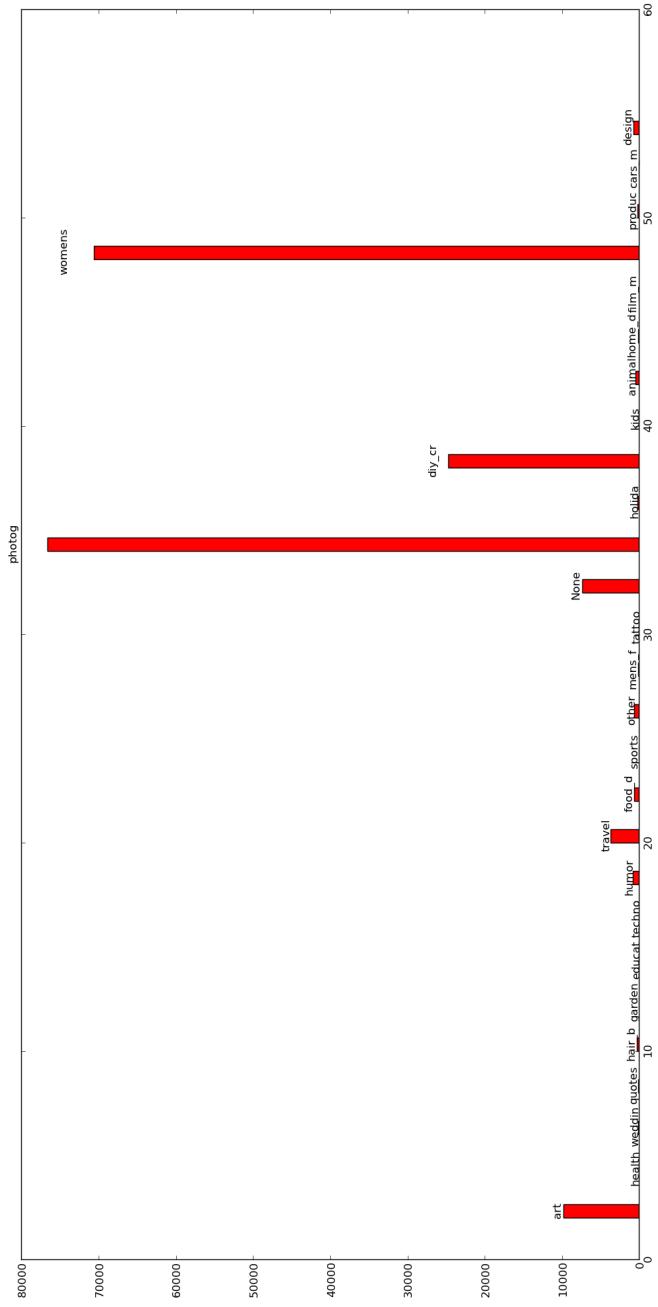


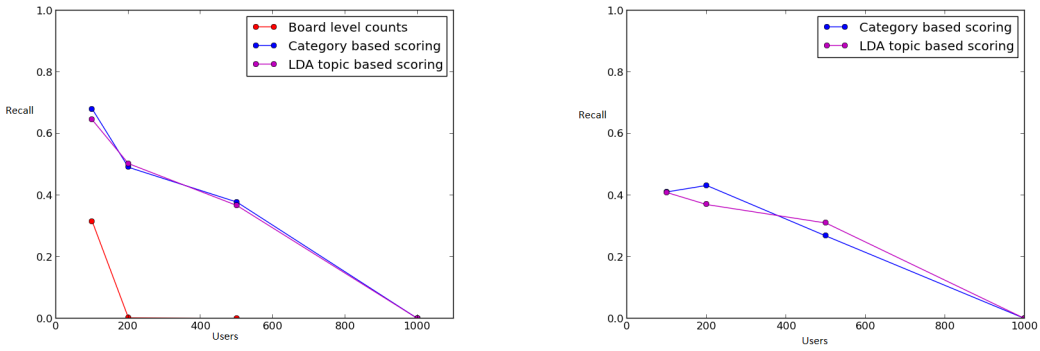
Figure 4.8: Sample category distribution of pins



- Figure 4.4 and Figure 4.6 show how recall varies with varying extra boards for number of boards equalling two and three. Figure 4.5 and Figure 4.7 show the variation of precision versus extra boards. These experiments help us decide the number of boards to start with. Figure 4.8 shows a sample distribution of pins over all categories.

Figure 4.9: Performance of methods over a Random universe

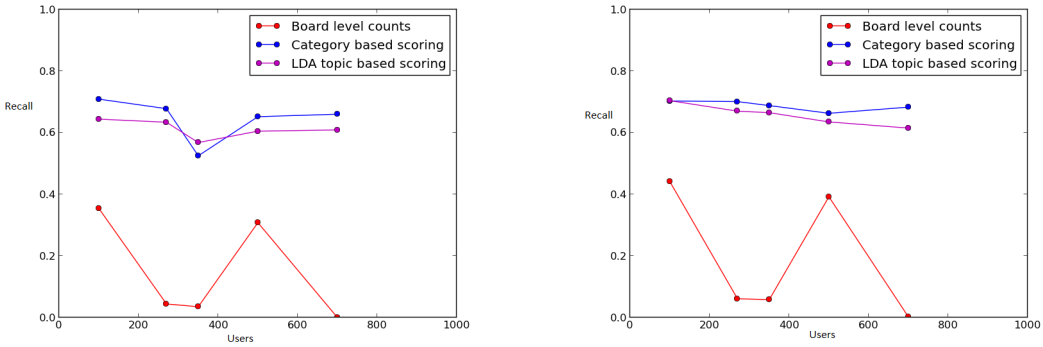
(a) N = 20 (b) N = 10



- There are two points to observe in Figure 4.9 . Firstly, Board Level counts doesn't give any results for N = 10. Secondly, with increasing users, recall decreases. We attribute this to the matrix getting larger in size and thus making the factorization difficult.

Figure 4.10: Performance of methods over a Similar universe

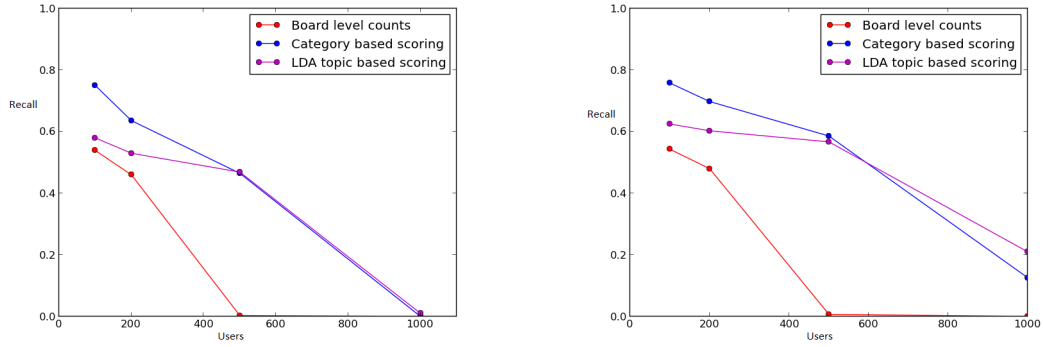
(a) N=10 (b) N=20



- In Figure 4.10, we observe that a similar universe yields very consistent and high performance of two methods with increasing users. We infer that this behavior is due to the coherence of the similar universe which generates high recall.

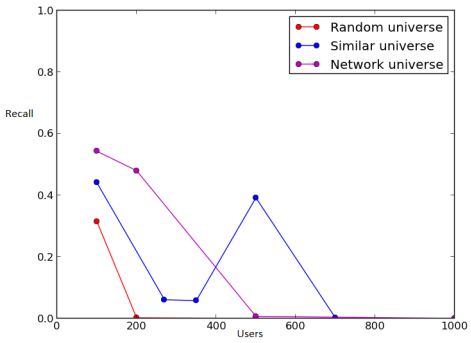
Figure 4.11: Performance of methods over a Network universe

(a) N=10 (b) N=20



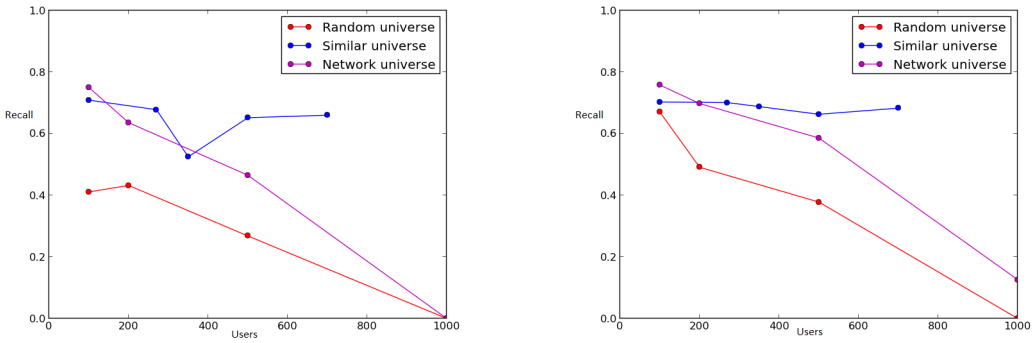
- The network universe gives the best recall performance with a low user base as seen in Figure 4.11. This indicates that a small network is a coherent network. But as we increase the number of users, recall decreases. Here, we infer that this is due to development of varied interests as the network grows.

Figure 4.12: Performance of Board Level counts over all universe, N = 20



- The graph in Figure 4.12 compares the performance of the universe for a rating method. For a naive Board Level counts method, we see that network universe starts out the best but a more coherent similar universe gives a more consistent performance. The Random universe as we observe is unable to scale up with the number of users.

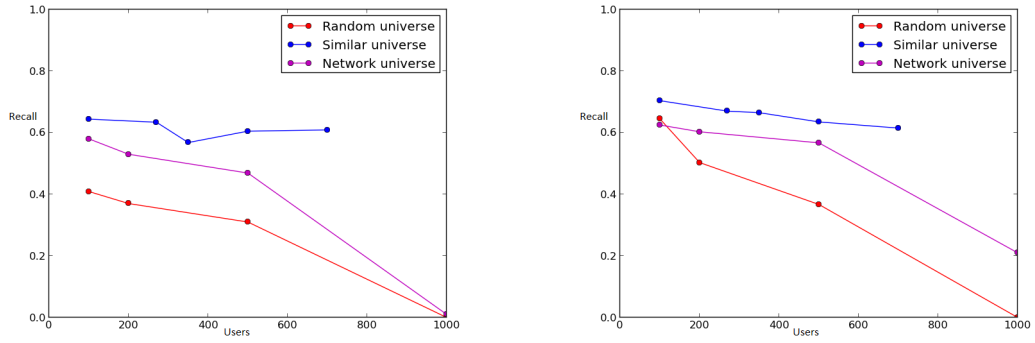
Figure 4.13: Performance of Category based scoring on all universe



(a) N = 10 (b) N = 20

- Figure 4.13 shows that not only have the similar and network universe performances improved, even the Random universe shows a significant improvement. Again the similar universe proves to be more consistent.

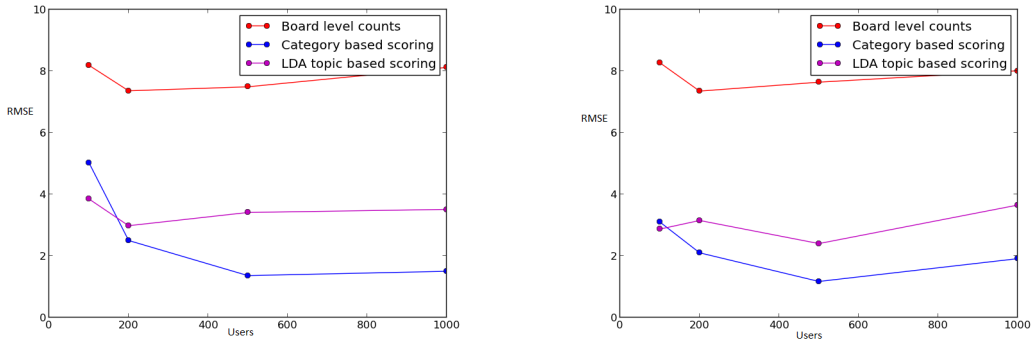
Figure 4.14: Performance of LDA topic based scoring on all universe



(a) N = 10 (b) N = 20

- LDA topic based scoring edges out Board Level counts but the graph in Figure 4.14 proves that it's not better than category scoring. We attribute this to lack of sufficient data for the LDA model to perform well.

Figure 4.15: RMSE metric of all methods on Random universe

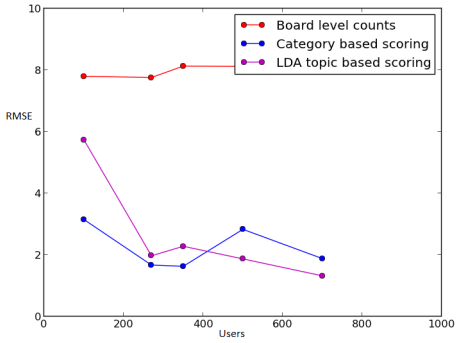


(a) N = 10

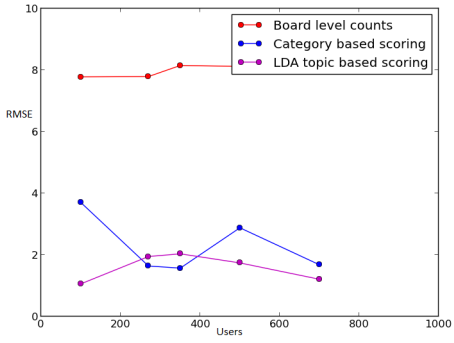
(b) N = 20

- In Figure 4.15, we observe the metric RMSE for all the methods. This graph shows that Category based scoring records lowest average RMSE followed by LDA topic scoring. Board Level counts records high RMSE throughout.

Figure 4.16: RMSE metric of all methods on Similar universe



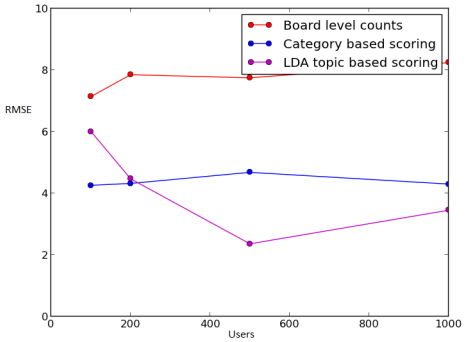
(a) N = 10



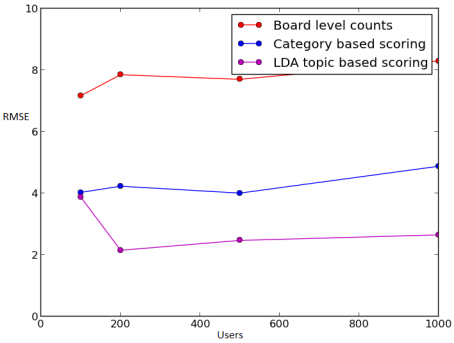
(b) N = 20

- Over a similar network, we see that LDA topic scoring and Category scoring alternatively outperforming each other. Another observation in Figure 4.16 is the RMSE of these methods is lower than the same methods on a Random universe. This again reiterates that the similar universe is coherent.

Figure 4.17: RMSE metric of all methods on Network universe



(a) N = 10



(b) N = 20

- For a network universe we observe high RMSE values for all methods when compared to similar universe in Figure 4.17. RMSE proves that the network universe is not as coherent as similar universe.

4.5.3 Summary of experiments

With these experiments and plots we are able to compare all the rating methods against each other as well as the different universes. Board Level counts acts as the baseline method and Category based scoring records consistent and high recall performance. LDA topic scoring falls short of category scoring due to insufficient textual data. Both recall and RMSE prove that similar universe is the best universe possible for a recommendation scenario in Pinterest. These observations help us identify and explain the causes of lower performance of some methods and also help us in identifying the right set of users to put together for a good recommender.

5. CONCLUSION

This thesis has proposed a new latent factor model for generating Pinterest board recommendations. Some of the key challenges this method faces include: (i) that boards are inherently personal, meaning that traditional community-oriented recommendation methods may be inappropriate; (ii) that many pins and boards lack descriptive text and other features that are typically used to power modern recommenders; and (iii) the challenge of evaluating such a recommender without access to a ground truth dataset. To tackle the feature sparsity and personal boards challenges, the overall approach generates ratings for every user-board pair which is then fed to a latent factor model which factorizes the sparse matrix to give ratings for unrated user-board pairs and the top rated boards form the recommendation list. Two of the key components of the proposed latent factor model are the (i) definition of the universe of users around each target user for identifying candidate boards to recommend; and (ii) the approach for assigning implicit ratings to each user-board pair for this universe of users (as the basis of the latent factor model).

Through investigation over a dataset of around 50,000 Pinterest users, 100 million pins, and around 570,000 boards, we have seen that careful tuning of these key design choices can greatly impact the quality of recommendations. We see that all three implicit rating approaches perform best under the network universe achieving a maximum of 75% recall. A random universe yielded the lowest performance as expected with a maximum recall of 67% and the similar user universe yields a maximum of 70.3% recall. Also, amongst the methods the category-based method performs best in all universe types yielding a maximum of 75% recall.

5.1 Next Steps

While encouraging, these results do highlight the challenge of generating high-quality personalized board recommendations. In our future work, we are interested in investigating the robustness of the proposed approach over a large-scale evaluation harness of millions of users. In addition, there are many enhancements that can be done to the current method, like incorporating other features like board coherence [3], entropy-based features, etc. Other recommendation algorithms like neighborhood models can additionally be integrated into the proposed method. Accurate categorization of boards will also be crucial for methods making use of that feature of Pinterest.

I would also like to research more on user modeling as I feel a good user model will lead to a good recommendation model. In the future when Pinterest opens its API and provides time-sensitive data, a timeline of user interests can be created and we can visualize how a user's interest in a topic grows or degrades. Time sensitive data can also help us visualize the lifecycle of a pin and help us predict the popularity of a pin. Pinterest has recently launched geo-tagged pins. I have already collected some geo-tagged pins and plan to continue working on this aspect of Pinterest. Pinterest also has the feature of adding place boards where a board will be about a location. This is very valuable information as this draws pretty images from various websites along with geolocations. Most importantly, as this is a website of pictures, this website has a lot of scope for image processing applications. Image similarity and other image based features can play a crucial role in recommendation. This can help overcome the challenges of no text or very little text present in descriptions of many boards. I have urls for the images contained in the pins already collected with the data I have. This is something I would like to venture upon later as I feel this is the

future of Pinterest applications.

REFERENCES

[1] Gilbert, E., Bakhshi, S., Chang, S., Terveen, L. *I Need to try this: A Statistical Overview of Pinterest.* In Proceedings of CHI-2013

[2] Hannon, J., Bennet, M., Smyth, B. *Recommending Twitter Users to Follow using Content and Collaborative Filltering Approaches.* In Proceedings of RecSys10. Random House, N.Y.

[3] Kamath, K., Popescu, A., Caverlee, J. *Board Coherence: Non-visual Aspects of a Visual Site.* In Proceedings of WWW 2013 (poster).

[4] Ottoni, R., Pesce, J.P., Las Casas, D., Franciscani, G., Kumaruguru, P., Almeida V. *Ladies First: Analyzing Gender Roles and Behaviors in Pinterest.* In Proceedings of ICWSM 2013.

[5] Kamath, K., Popescu, A., Caverlee, J. *Board Recommendation in Pinterest.* In Proceedings of UMAP Workshops, 2013.

[6] Pennacchiotti, M., Silvestri, F., Vahabi, H., Venturini, R. *Making your Interests Follow you on Twitter.* In Proceedings of CIKM 2012.

[7] Herlocker, J. Konstan, L. Terveen, and J. Riedl. *Evaluating Collaborative Filtering Recommender Systems.* In Proceedings of CIKM 2012.

[8] F. Ricci, L. Rokach, B. Shapira, and P. Kantor. *Recommender Systems Handbook* ISBN 978-0-387-85819-7. Springer Science+ Business Media, LLC, 2011, 1, 2011.

[9] F. Abel, Q. Gao, G. Houben, K. Tao. *Analyzing User Modeling on Twitter for Personalized News Recommendations.* In Proceedings of UMAP 2011.

[10] Fabian Abel, Qi Gao, Geert-Jan Houben, and Ke Tao. 2011. *Analyzing Temporal Dynamics in Twitter profiles for Personalized Recommendations in the Social*

Web. In Proceedings of the 3rd International Web Science Conference (WebSci '11).

[11] Kailong Chen, Tianqi Chen, Guoqing Zheng, Ou Jin, Enpeng Yao, and Yong Yu. 2012. *Collaborative Personalized Tweet Recommendation.* In Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '12)

[12] Kyew, S.M., Lim, E., Zhu, F. *A Survey of Recommender Systems in Twitter.* In Proceedings of SocInfo 2012.

[13] Pankaj Gupta, Ashish Goel, Jimmy Lin, Aneesh Sharma, Dong Wang, and Reza Zadeh. 2013. *WTF: The Who to Follow Service at Twitter.* In Proceedings of the 22nd International Conference on World Wide Web (WWW '13).

[14] Nagarajan Natarajan, Donghyuk Shin, and Inderjit S. Dhillon. 2013. *Which App will you use next?: Collaborative Filtering with Interactional Context.* In Proceedings of the 7th ACM Conference on Recommender Systems (RecSys '13).

[15] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. *Latent Dirichlet Allocation.* Journal of Machine Learning Research, 2003.

[16] Cataldo Musto. 2010. *Enhanced Vector Space Models for Content-based Recommender Systems.* In Proceedings of the Fourth ACM Conference on Recommender Systems (RecSys '10). ACM, New York, NY, USA, 361-364.

[17] Tkalcic, M, Odic, A. ; Kosir, A. ; Tasic, J. *Affective Labeling in a Content-Based Recommender System for Images.* In Proceedings of IEEE Multimedia Transactions, 2013.

[18] Robin Burke. 2002. *Hybrid Recommender Systems: Survey and Experiments.* User Modeling and User-Adapted Interaction 12, 4 (November 2002)