12-2017

Popularity Prediction

Andrew Furgiuele
University at Albany, State University of New York

UNIVERSITY AT ALBANY, SUNY

Honors Thesis

Popularity Prediction

An honors thesis presented to

the Department of Computer Science,

University at Albany, State University of New York

in partial fulfillment of requirements for graduation

with Honors in Computer Science and Applied

Mathematics, and graduation from The Honors College

Andrew Furgiuele

Reviewed by:

Charalampos Chelmis, Ph.D.

Assistant Professor, Department of Computer Science

Honors Research Advisor

December 2017

Abstract

With the rise in popularity of the Internet, data describing unique types of items has been collected into easy to access sources. Using this newly acquired data, is it possible to predict if an item will become a bestseller while another fade away with time? Popularity prediction is a problem that has attracted a great deal of research recently, and for good reason. The ability to predict an items future rise to popularity or fall to obscurity is a possibly priceless skill and sought out in many different industries such as sales, investments, and marketing.

This report enumerates and analyzes a number of factors assumed to be an indicator of popularity. Additionally, we propose a number of popularity prediction methods, and evaluate using a cohost of evaluation metrics, and state of the art baselines. Our findings show promising potential for popularity prediction, based on a combination of structural and temporal properties indicative of popularity. The key proposed metrics include a measure of similarity between two items, and various definitions of popularity evolving with time. Experiments on a large scale real dataset from *Yelp* allow us to demonstrate the performance of our methods on predicting the popularity of businesses. We believe the methods described below can be extended to be used for diverse types of data.

Acknowledgements

Above all, I would like to thank Dr. Charalampos Chelmis for all the knowledge and support he gave while acting as my Honors Research Adviser. I have learned a tremendous amount of real-world applicable information under his guidance this past year. He has continuously gone out of his way to help me in every step of the process of this thesis.

I would also like to thank the University at Albany's Honors college for its support and belief in me. Since my admission freshman year, it has continuously pushed me into becoming a better student. Lastly, I would like to thank the Computer Science department and College of Engineering and Applied Sciences for the recourses it provides the Computer Science students. It provides a good atmosphere for learning in both classrooms and outside of school.

Table of Contents

List of Figures

Chapter 1
Introduction

It is not difficult to imagine the benefits of predicting the future popularity of an item. Imagine a company with two products, but only one opening in the next catalog. With the ability to know which product will sell better, the company can guarantee the greatest possible revenue. However, popularity prediction comes with difficult obstacles that are hard to quantify. A correct forecast of popularity is valuable to a variety of different industries and products, and may very well be possible.

First and foremost, what defines popularity? Take *YouTube* videos for example, is number of views, number of likes, or number of times shared indicative of popularity? On the other hand, the popularity of a product on *Amazon.com* may be measured as the number of units sold. Unclear definitions of popularity by itself makes the problem of prediction challenging [1]. To add another layer of complexity, knowing that a product will become popular may not be enough; predicting the time that it will become popular may be just as important. Some items are 'fads' and become popular very quickly, then just as quickly fade away. Some may start slow and progressively become more popular throughout time.

In this thesis, we tackle the problem of popularity prediction by studying both, descriptive, and temporal data. Specifically, we consider the network of interactions between users and businesses in *Yelp* and use this to predict both if, and when an item will become popular. In Chapter 2, we discuss research related to the concepts used in this thesis, such as measurements of similarity and attributes indicative of popularity. In Chapter 3, we explain the problem we wish to accomplish, and the difficulties related to popularity prediction. Chapter 4 is composed of a formal outline of the experiments that we perform, the results and analysis of which are in Chapter 5. Finally, Chapter 6 is a conclusion and a plan for extending this work. We believe that our proposed method described in this thesis is generalizable, and as such it can be applied to a diverse set of items.

Chapter 2

Related Work

The idea of popularity prediction for various items has been researched thoroughly. The following are ideas from papers that has significantly inspired our work.

One of the integral metrics used in the methods proposed in this report is a measurement of similarity between two items. This is a challenging problem to generalize as items of different types may not have the same metrics of similarity. To combat this, many methods generalize data into graphs and use common attributes that relates nodes in the graph. As the size of the graph rises (as large datasets are common in problems like this) there is a significant increase in computational complexity for calculating graphical attributes. One method used to circumvent this problem is using random walks to compute approximations close enough to the true values. Yin, Gupta, and Han [3] investigated the use of random walks to measure the relevance of attributes such as centrality, preferential attachment, and influence in social graphs. There use of random walks over weighted social graphs composed of users lead to success in link recommendation. Benchettara, Kanawati, and Rouveirol [4] were able to use a supervised machine learning approach instead of random walks to measure the same neighborhood based metrics along with other topological attributes to group similar nodes and predict links in co-authorship graphs.

Similarity is the building block of our method of popularity prediction, but other papers have also investigated prediction of popularity using different methods. [1] and [2] both deal with popularity prediction related to various online items. He, Gao, Kan, and Liu [1] propose a method that predicts popularity of Web 2.0 items based on user comments of YouTube videos. They create a user-item ranking system based on bipartite graph structure to capture the temporal, social, and current factors related to popularity. Westwood, Johnson, and Bunge [2] focus their workon popularity of clusters in the social network as their work deals with popularity of communities rather than singular items. Lastly, [3] is a

survey of 19 recent papers that deal with popularity prediction of various web 2.0 items using 5 distinct prediction models, which speaks to the importance and amount of research put into the problem at hand.

The research described above allowed us to draw inspiration for the proposed method and allowed for a solid framework to build off. The method described in this report not only expands the ideas conveyed here but also adds the use of temporal data to create an entirely new method. There is a distinct lack of papers that combine similarity metrics with temporal attributes to achieve popularity prediction. We propose a method in chapter 4 that combines concepts from these papers along with unique ideas to create a new model for prediction of future popularity.

Chapter 3

Problem Formulation

We formulate the *popularity prediction* task as a categorical-classification problem. Specifically, given an item A at time *t*, our goal is to predict which category item A will most likely fall into. We consider three categories: increase in popularity, decrease in popularity, and stagnate. To address the classification problem posed above, we begin by calculating the similarity between items by using a random walk framework over a bipartite graph constructed of items and their corresponding descriptive attributes (section 3.1). Next, we track changes in popularity over logical time intervals to get a sense of item's past popularity (section 3.2). As popularity is not an instantaneous property that fluctuates in short periods, the intervals must be long enough to be dense with interactions, but fine enough to not lose the overall pattern. Finally, we classify an item's future popularity based on its past popularity, and the past popularity of "similar" items (section 3.3).

Chapter 4

Proposed Approach

We propose an approach based on two concepts: first, items that have been previously popular will be popular in the future. Second, if an item is related to a recently popular item, that item will soon become popular. The steps needed are therefore: group related items, measure their past popularity, and predict the popularity of the items in the future.

4.1 Static Network

We consider the bipartite graph G(*I, A*), of a set of items *I,* and a set of descriptive attributes *A*, of items in *I*. Each item is associated with a list of attributes from *A.* For example, a Chinese Food restaurant may possess the list of items: take-out, delivery, restaurant, etc. Figure 3.1 demonstrates a toy bipartite graph where nodes A-D represent Items, whereas nodes marked as 1-6 represent attributes.

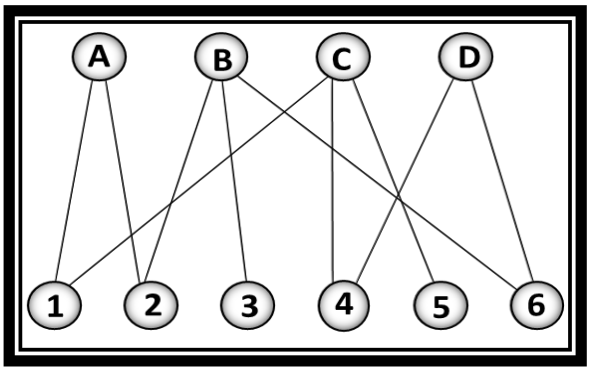

Fig 3.1: Toy Bipartite Graph

The bipartite graph G allows us to measure "similarity" between items based on network structural properties [3]. Next, we discuss the attributes extracted from the bipartite graph of businesses from Yelp that we consider in our approach:

1) **Homophily**: Tendency of items to associate with "similar" others. Items that share a common attribute are more likely to be similar compared to items that share no common attributes. The more attributes shared further strengthens the possibility of similarity.

2) **Rarity**: Attributes with a high degree (i.e. more common attributes such as "restaurant") are connected with more items in *I*, which may dilute the importance of two nodes sharing the same attribute. Thus we consider common attributes to be less indicative of "similarity" as opposed to "rare" attributes.

3) **Common Neighbors**: We call two items that share the same attribute "neighbors". Two items that do not directly share an attribute are more likely to be "similar" if they share many common neighbors as opposed to none.

We propose a variable-step random walk based algorithm across G(*I, A*) to compute the similarity between items in *I* as a function of homophily, rarity, and common neighbors. Specifically, for each item in *I*, we perform a random walk of the form i_k -> a_j -> i_l -> … -> i_m, where $a_j \in A$, and i_k, i_l, and $i_m \in I$. The number of steps is randomized to take a value in the set [1,5]. This range allows for a good tradeoff between connecting items that are multiple steps away, but also prioritizing closer items. We denote the algorithm for a random walk on item $i \in I$ by S_i, a vector containing the approximate similarity between item i and all other items. Formally:

$$S_i = [P_1, P_2, \ldots, P_n] \text{ where } P_k = \frac{\text{frequency of i} -> \text{k}}{\text{total trials}} \text{ and i} = i_i, \text{k} = i_k \, \varepsilon \, I \quad (1)$$

All values in S_i are in the range of [0,1], where 0 denotes least similarly, and 1 represents highest similarity. When sorted, S_i represents a list of items in descending order of similarity rating compared to item *i*. Because of the stochasticity of random walks, we run the random walk until S_i is stable between iterations, i.e. $S^k_i \approx S^{k+1}_i$, where S^k_i denotes S_i at iteration *k*. Our intuition is that 1) Items that are directly connected via an attribute are 'close'. The closer an item is the better chance the random walk will end there. 2) Common attributes have many outlinks, therefore items connected by rare attributes are more likely to be connected on the walk. 3) Since we use a variable step random walk, some pairs of items will have a non-zero similarity with no shared attributes. In this case, the items that are more likely to be connected are items that share common neighbors.

4.2 Dynamic Network

In order to track popularity as it evolves over time, we consider the dynamic network of user interactions at it follows, let *U* be the set of users, *I* be the set of items as before, and *T* be as a set of intervals, i.e., $T = \{t_1, t_2, \ldots, t_n\}$, where n is the number of intervals. For any given interval $t \, \varepsilon \, T$, we have a list of users U_j^t that interacted (e.g. reviewed business *j*) during the interval *t*. We consider users in U_j to be connected with item *j* until either of the following two conditions are met: 1) they interact with a different item *k*, in which case they "move" from *j* to item *k*, or 2) γ time-intervals is observed with no new interaction. We introduce variable γ to reduce the effects of users with low number of overall interactions 'sticking' to users. Without this variable, users with low amount of interactions will stay attached to one item, potentially resulting in erroneous popularity estimates over time. As time progresses

users interact with, and therefore "move" to different items, as demonstrated in Figure 3.2. Intuitively, items that are consistently associated with many users are considered to remain popular; such items are gaining new interactions in each time interval.

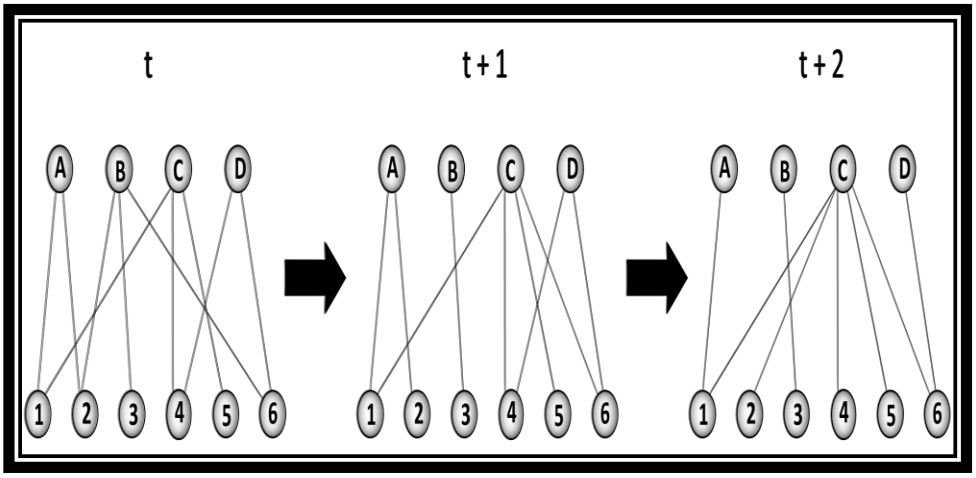

Fig 3.2: Example dynamic bipartite graph, evolving over two time intervals.

As mentioned in chapter 1, different metrics of popularity can be used given the problem on hand. Two commonly used metrics are the amount of interactions (views, purchases, etc.) and ratings (Good/Bad, 1-5-star rating, etc.) [1]. In our approach, we track both metrics to study popularity from multiple lenses.

The cumulative number of interactions for i ε *I* during interval t is represented by $U_i(t)$. We capture the history, $Hu_i = \{U_i(t-1), U_i(t-2), \ldots, U_i(t-\beta)\}$, of the number of interactions for item *i* in the past β time intervals and the change in between consecutive intervals, $Hu^*_i = \{U_i(t) - U_i(t-1), U_i(t-1) - U_i(t-2), \ldots, U_i(t-\beta+1) - U_i(t-\beta)\}$, to identify items increasing in popularity, i.e. being marked mostly with positive entries in Hu^*_i. Similarly, unpopular items can be identified by mostly zeroes or mixture of positive and negative entries in Hu^*_i. We use the history if interactions over time to estimate popularity as a single quantity, denoted as Ω. This method is easily adjusted from popularity defined by number of interactions to an items rating, simply let Hu_i be the star rating of *i* at each interval and

change Hu_i* accordingly. To ensure that more recent time-intervals are given a higher priority in calculating **Ω**, we experiment two different methods of aggregation. Specifically:

1. **Summation of Hu***: Naively sum the past β intervals as (2). A positive value indicates an increase in popularity.

$$\Omega = \sum^{\beta}_{i=0} Hu^*(i) \quad (2)$$

2. **Exponential Decay (ED):** To signify the importance of recent intervals, we will use an exponential decay function (3) to weaken the bias of older intervals. The variable α signifies the decay parameter, a value between 0 and 1.

$$\Omega = \sum^{\beta}_{i=0} (\alpha^i * Hu^*(i)) \quad (3)$$

4.3 Combination and the Classification Tree

Given the information derived from sections 4.1 and 4.2, we formulate a method of classification for the future popularity of an item. Our method of prediction considers a few assumptions. First, item's past interactions are an important attribute when considering future popularity. Different research [6] in items such as YouTube videos and Digg stories have found strong correlation between past and current popularity. Second, items that have a high similarity metric will correlate in popularity. This derives from the definition of homophily [3], which says related items in a network will act accordingly. Two very similar items will behave in a more analogous manner compared to two items that have little or no similarity. Using these assumptions, we use a classification tree to classify an item into one of three groups:

1. **Increase**: an item is predicted to show significant increase in ratings or number of interactions in the following time intervals

2. **Decrease:** an item is predicated to show significant decrease in ratings or number of interactions in the following time intervals
3. **Stagnate:** an item is predicated to show negligible increase or decrease in the number of interactions in the following time intervals.

. The input variable, ϕ, used for the classification tree is formulated of S_i and Ω from sections 4.1 and 4.2 respectively. It measures the summation of the past popularity of the *n* most similar items. We combine them as:

$$\phi(i) = x\left(\frac{\left(\sum j=0(\Omega(Sj))\right)}{n}\right) * y(\Omega(i)) \quad \text{where } x+ y = 1, \text{ and n = length of } S_j \qquad (4)$$

This metric is a linear combination of an items past popularity along with the average past popularity of the most popular businesses. The variables *x* and *y* are used to control the trade-off of both factors. The metric (4) is inputted to the classification tree modeled in figure 4.1. The classification tree categorizes each item according to their value compared to two parameters A and B such that A< B and A,B are real numbers. The value of the two parameters are dependent on the type of item being studied, definition of popularity, and variation of the values in HU*.

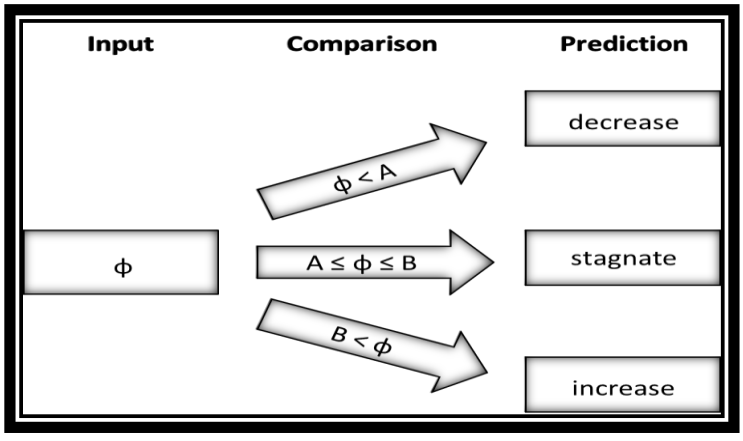

Fig 4.1: Classification Tree

Chapter 5

Experimental Evaluation

In this section, we test our methods on a large-scale real dataset and measure the effectiveness of our approaches using common metrics for classification problems.

5.1 Dataset Description

To test our proposed method of prediction, we use data provided by the *2017 Yelp Dataset Challenge* [7]. Businesses from the dataset make a viable candidate for prediction because: they have a list of descriptive attributes, they have time-stamped user interactions (in the form of reviews/check-ins), and they go through periods of popularity. Interactions through the form of reviews are especially valuable to this research because they provide not only a time-stamped interaction between a business and user, but also a rating between 1 and 5. The dataset comprises business ratings spanning 12 locations across 4 different countries. For our evaluations, we used data from the state of Ohio. We chose this state for its large representation in the Yelp dataset; over 100,000 businesses are included with information for businesses in many cities spread across the state.

The data from Yelp spans 13 years, from early 2004 to the middle of 2017. Since *Yelp*'s start in 2003, the amount of traffic has increased exponentially causing an increase in the amount of reviews received. This caused an imbalance in the frequency of reviews throughout the 13-year span, with the amount of reviews per interval increasing throughout time, as shown in Figure 5.1.

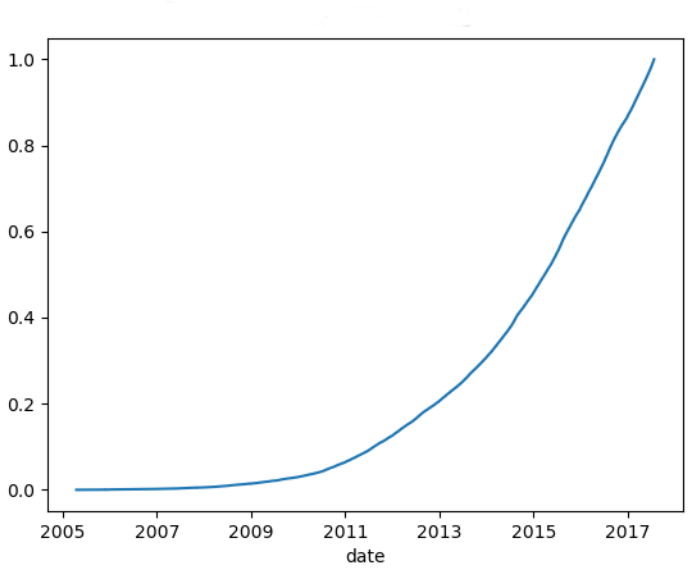

Figure 5.1: Reviews of *Yelp* businesses over time

5.2 Accuracy Metrics

We perform a 75/25 split of the time-stamped interactions, the first group for training (T_{train}), the second for testing (T_{test}). We use data in the training interval to gain a sense of past popularity, and we use the testing interval to make predictions and evaluate our method. Our model (Section 4.3) is evaluated using metrics commonly used in classification tasks, i.e., Precision, Recall and F1-Score:

- **Precision**: represents the percent of correctly classified instances out of the guesses

$$\text{Precision} = \frac{Number\ of\ true\ positive\ examples\ Identified}{Predictied\ number\ of\ positive\ examples},$$

- **Recall:** represents the percent of correct instances guessed

$$\text{Recall} = \frac{Number\ of\ true\ positive\ examples\ identified}{Total\ amount\ of\ true\ positives},$$

- **F1-Score:** represents the average of Recall and Precision

$$\text{F1-Score} = \frac{2*precision*recall}{precision+recall}.$$

5.3 Analysis and Results

To evaluate the proposed method, we perform the classification task for two definitions of popularity, ratings and number of reviews, for the businesses during the time interval T_{test}. We aggregate the Test intervals, into two sets: $T_{test\ 1}$ and $T_{test\ 2}$. T_{test1} is the first half of T_{test} or the interval right after T_{train}, and T_{test2} is the interval after the first. This method allows us to interpret how accurate a prediction is with a large gap in time.

Throughout chapter 4, we have described multiple variables that alter how we predict popularity, below are values tested for each variable:

- Variables (*X*,*Y*), which control the tradeoff of importance between a business's past popularity and the past popularity of similar businesses (section 4.3), were set to be **(0.75, 0.25)**, **(0.5, 0.5)**, **(0.25, 0.75)**. These values weight the two variables equally, give more importance to an item's own past over the past of similar businesses, and weight **t**he past of similar businesses greater than the past of the business in question accordingly.
- The decay parameter α used when valuing the weight of a business's past was set to **0.5**, **0.75**, and **0.9**. The lowest value will decay faster, which gives diminishing importance to businesses less similar. In the highest value of α, the importance decays slower, thus giving less similar businesses a larger input.
- The variables *t* and γ, which represent the length in days of the time intervals, and the number of intervals until an interaction 'expires' accordingly, used are *t* =**30 days** and γ = **6.** The importance of γ was described in chapter 4, and only applies to the experiments associated with

popularity defined by the number of interactions. We choose these values as it provides an ideal balance of aggregation without losing finer details.

- The variables A and B represent the bounds of the classification tree illustrated in figure 4.1. We choose values of (-2, 2) for the bounds. These bounds are large enough that items that show little movement will be classified correctly as stagnating, rather increasing or decreasing.

Figures 5.2 and 5.3 show results for the prediction of number of reviews, while Figures 5.4 and 5.5 show results for the prediction of ratings.

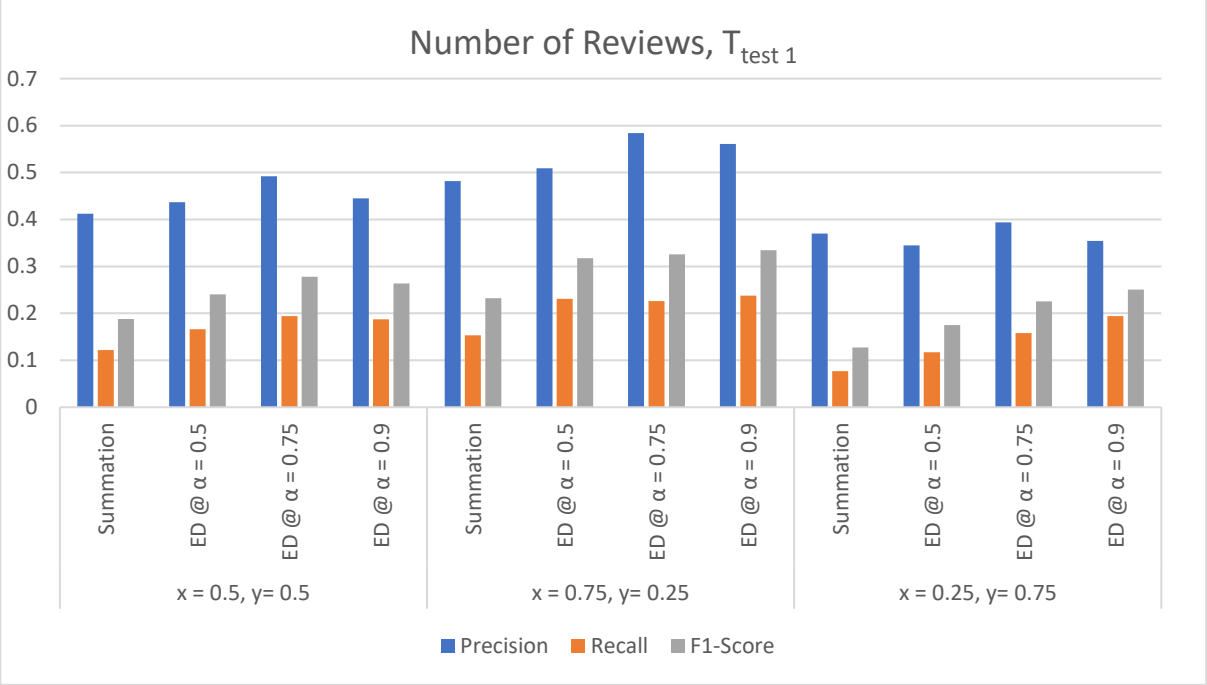

Figure 5.2: Performance of classification for number of reviews in $T_{test\ 1}$

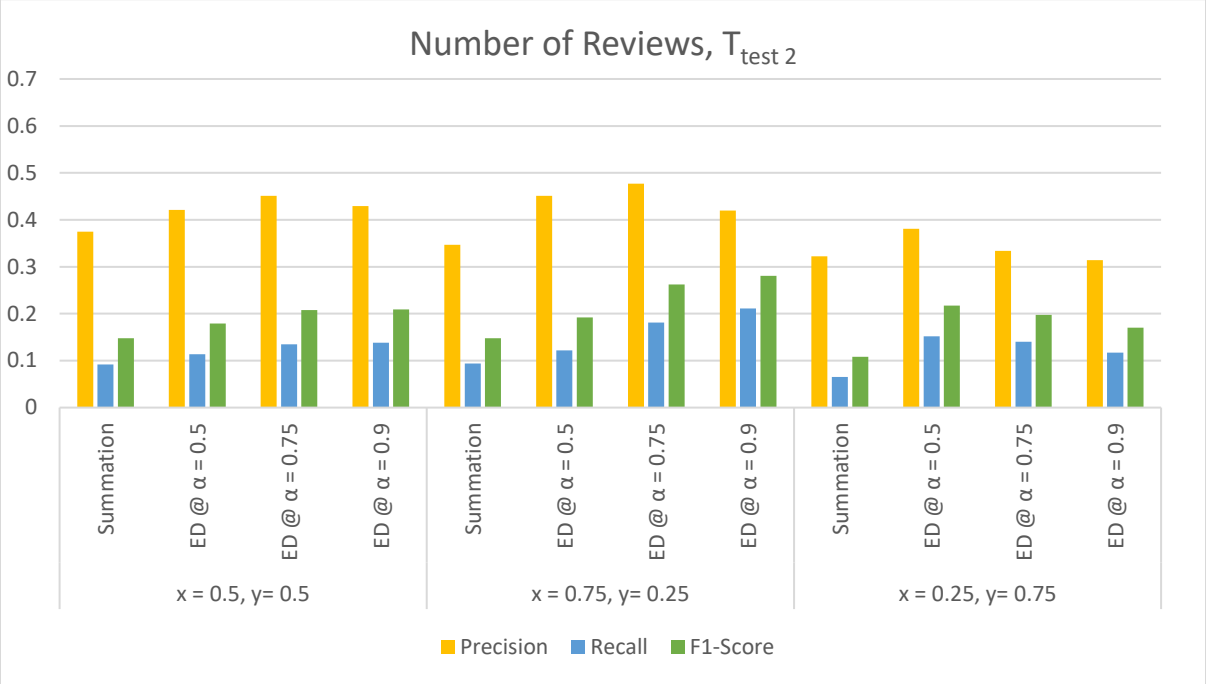

Figure 5.3: Performance of classification for number of reviews in $T_{test\ 2}$

Observation 1: Figures 5.2 and 5.3 indicate the most effective values of (X,Y), in decreasing order are: (0.75, 0.25), (0.5, 0.5), (0.25, 0.75). This suggests that an item's own past is more important than the past of similar items when considering popularity in the form of number of interactions. It is worth noting that Figure 5.3 shows the results obtained for (X= 0.5, Y= 0.5) are comparable to (X= 0.75, Y= 0.25) in the second interval. This suggests that similar items' popularity may be more of a factor in continuous popularity, rather than instantaneous. In both cases setting X and Y to 0.75 and 0.25 respectively resulted in promising results, which further strengthens the hypothesis that an item's own past outweighs the past of similar items.

Observation 2: Figures 5.2 and 5.3 suggest that our model is less accurate at predicting popularity farther in the future. This is expected, as our predictions are based on past measurements of each item in the training set only. Thus, when trying to predict the number of reviews for the second interval, the additional information about "movement" that occurred in the first testing interval is ignored.

Observation 3: Figure 5.3 shows that the difference in accuracy between values of α decreases with time, which correlates with our model's accuracy decreasing over longer periods of time. Figure 5.2 has differences in precision as large as 0.24, while Figure 5.3 only has businesses as big as 0.15.

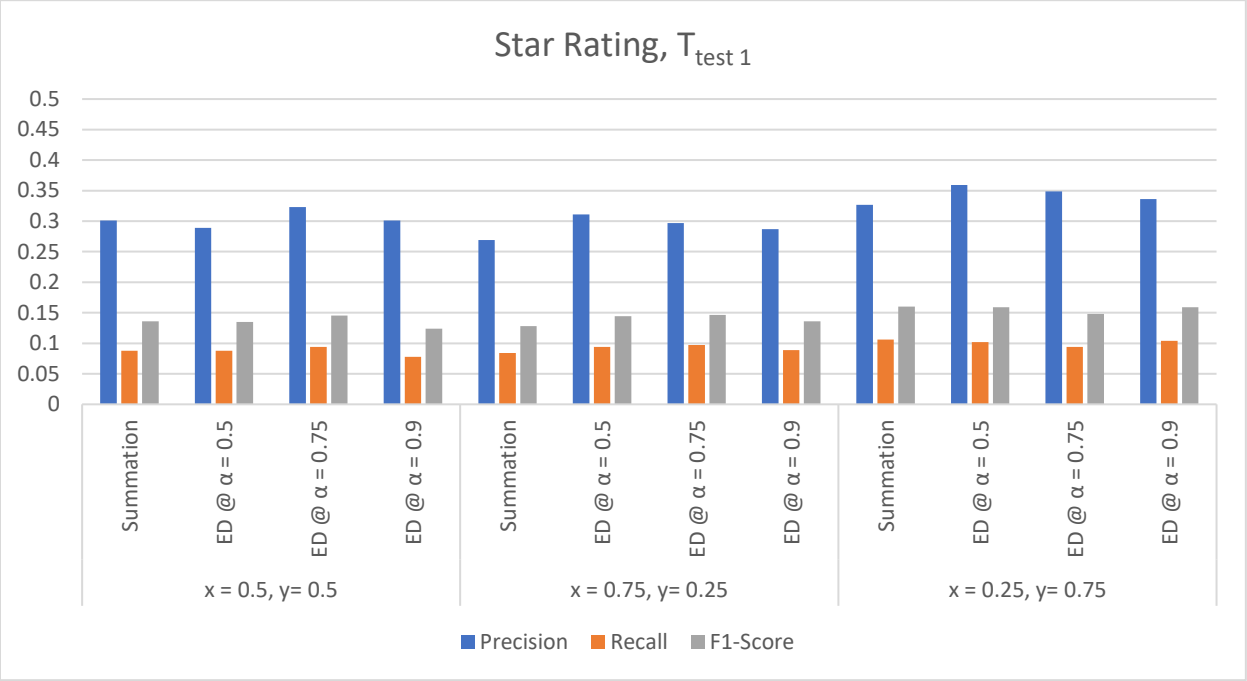

Figure 5.4: Performance of classification for star ratings in $T_{test\ 1}$

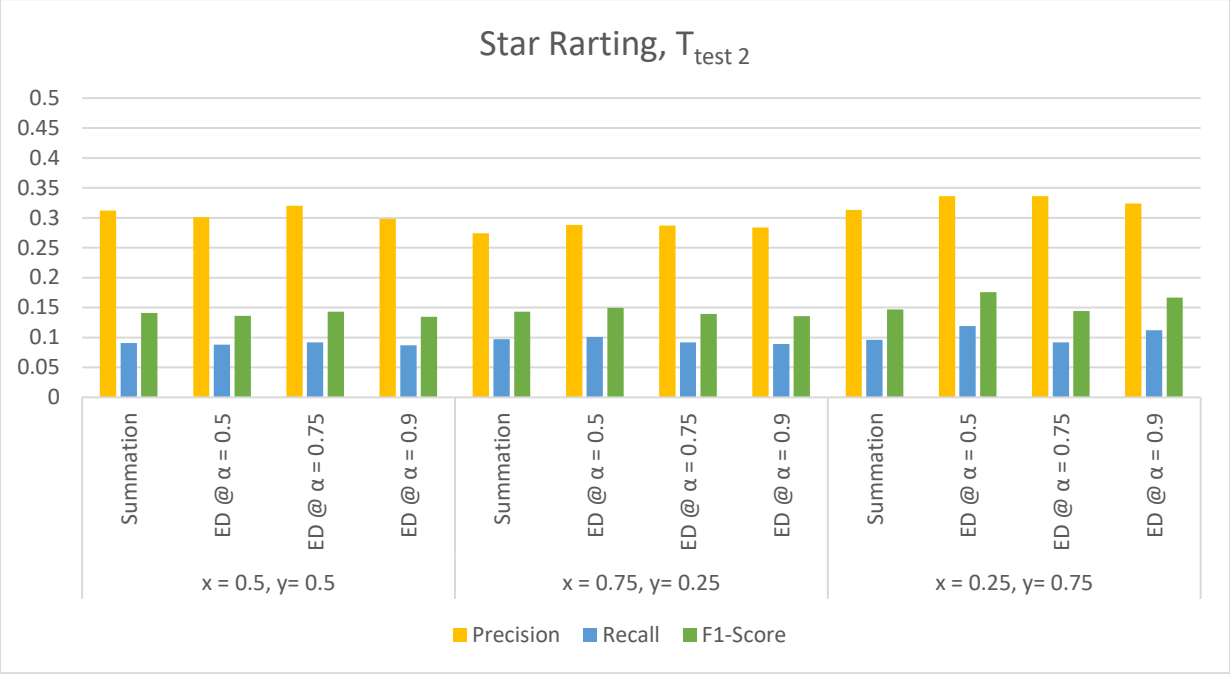

Figure 5.5: Performance of classification for star ratings in $T_{test\ 2}$

Observation 4: The prediction of future star rating is less accurate than the prediction of number of reviews. In the first time interval of the first experiment, we received an average F1-Score of 24.6%, while the current experiment produced only an F1-score of 19.33%. In addition, the highest levels of precision were found in the first experiment. In contrast, the prediction of star rating 'aged' better, meaning that the results from the first and second time interval are more similar. This implies that star rating is a more stable score.

Observation 5: Figures 5.4 and 5.5 indicate the most effective values of (X,Y), in decreasing order are: (0.25, 0.75), (0.5, 0.5), (0.75, 0.25). This is the inverse order that was observed in the first experiment. This observation implies that a star rating is closely correlated with the ratings of similar items. This differs from population defined through number of reviews, which is strongly correlated with its own past. The differences in accuracy of each value of (X, Y) are not as pronounced as seen in the first experiment.

Observation 6: The values of $\alpha = 0.5$, and $\alpha = 0.75$ perform the best in the first interval. In the second interval, the values of $\alpha = 0.5$, and $\alpha = 0.75$ show better performance than the others, but the difference in accuracy is small (± 0.05). Additionally, we see that the method of purely summation performs consistently the worst, which reaffirms our belief that there is diminishing value to the similarity metric.

Chapter 6

Conclusion

In this thesis, we have studied the predictability of an item's popularity, and demonstrated using business from *Yelp*. We investigated multiple definitions of popularity, and shown their differences. The method proposed combined historical observations of similar items, along with the history of a given item, to classify an item as increasing, decreasing or stagnating. We evaluated our methods using a subset of the Yelp Challenge Dataset and showed that accurate popularity prediction can be significantly improved.

Future work in this subject would first include working on improving the accuracy of our proposed method. To achieve this a wide range of features could be considered in addition to the set we explored in this thesis. Furthermore, the number of categories used in the classification tree could be separated for a finer granularity. We hope that by expanding the number of classifications, more accurate predictions can be achieved. Finally, the next step is to test the methods used with other forms of e-commerce items. Since the methods used in this paper are not specified to one data type, we believe we can expand our model to predict popularity for a wide array of items.

References

[1] Xiangnan He, Ming Gao, Min-Yen Kan, Yiqun Liu, Kazunari Sugiyama. Predicting the Popularity of Web 2.0 Items Based on User Comments, 2014

[2] Sophia Westwood, Melissa Johnson, Brie Bunge. Predicting Programming Community Popularity on StackOverFlow from Initial Affiliation Networks, 2013

[3] Zhijun Yin, Manish Gupta, Tim Weninger, Jiawai Han. A Unified Framework for Link Recommendation Using Random Walks, 2010

[4] Nesserine Benchettara, Rushed Kanawati, Celine Rouveirol. Supervised Machine Learning applied to Link Prediction in Bipartite Social Networks, 2010

[5] Alexandru Tatar, Marcelo Dias de Amorim, Sege Fdida, Panayotis Antoniadis. A survey on predicting the popularity of web content, 2014

[6] Gabor Szado, Bernardo Huberman. Predicating the popularity of online content, 2010