**Cross-Domain User Modeling: Applying Graph Techniques for Reasoning on Personal Data from Social Networks**

**Research proposal for M.sc degree in the I.S department**

**Oz Shapira**

Contents

[1. Introduction 3](#_Toc348252994)

[2. Background and Related works 4](#_Toc348252995)

[2.1. Background 4](#_Toc348252996)

[2.1.1. Recommender systems 4](#_Toc348252997)

[2.1.2. Graphs as data structures 6](#_Toc348252998)

[2.1.2.1. Graph traversal 6](#_Toc348252999)

[2.1.3. Social networks (SN) 7](#_Toc348253000)

[2.2. Related work 8](#_Toc348253001)

[2.2.1. Generic Semantic-based Framework 8](#_Toc348253002)

[2.2.2. Taste Fabric of Social Networks 8](#_Toc348253003)

[2.2.3. Network Profiles as Taste Performances 8](#_Toc348253004)

[2.2.4. On the Social Web 8](#_Toc348253005)

[2.3. Summery 8](#_Toc348253006)

[3. Research Goals and Questions 9](#_Toc348253007)

[4. Tools and Methods 9](#_Toc348253008)

[4.1. Methods 9](#_Toc348253009)

[4.1.1. Data Source 10](#_Toc348253010)

[4.1.2. Crawling Pinterest 10](#_Toc348253011)

[4.1.3. Data collection and graph representation 11](#_Toc348253012)

[4.2. Tools 13](#_Toc348253013)

[4.3. Evaluation 14](#_Toc348253014)

[5. Timetable 14](#_Toc348253015)

[6. Initial Results 15](#_Toc348253016)

[7. Research Contributions 15](#_Toc348253017)

[8. References 16](#_Toc348253018)

# Introduction

What if we know that someone loves pizza, James bond movies and he also loves jogging on at the beach and will asks for a recommended recipe of a cake she may love? How can we make use of the information we already know about the personal preferences of this user regarding fast food, entertainment and sports in order to suggest a preferred recipe for a cake? This research will try to address this challenge and suggest a recommender system able to answer this kind of questions, of how to use information from one domain, for reasoning about preferences in another domain, using general graph based techniques.

“Recommender systems represent user preferences for the purpose of suggesting items to purchase or examine. They have become fundamental applications in electronic commerce and information access, providing suggestions that effectively prune large information spaces so that users are directed toward those items that best meet their needs and preferences“ [Burke 2002]. Recommender systems became an important research area since the appearance of the first papers on collaborative filtering in the mid-1990s [Adomavicius and Tuzhilin 2005]. There has been much work done both in the industry and academia on developing new approaches to recommender systems over the last decade. Examples of such applications include recommending books, CDs and other products at Amazon.com, movies by IMDB, and news at VERSIFI Technologies (formerly AdaptiveInfo.com) [Adomavicius and Tuzhilin 2005]. Most recommender systems nowadays are focused on providing a personalized service in a specific domain, as does for instance Pandora – a music recommender system or IMDB movies recommender system (see figure 1 as an example).



Figure 1: from IMDB website as movie recommender system

In order to provide a personalized service to their users, recommender systems need to have relevant personal information about their users, or a “model” of them - a “User Model”. When this information is available, then the task of recommendation may be straight forward – provide a service based on the relevant information. In our example, if the system knows the user’s preferences about cakes, then finding a similar recipe becomes and easy task. However, in many cases, like in our example, the system does not have this information. This may be the case of a first time visitor to a cultural heritage site/a new city/ restaurant etc. The lack of sufficient user modeling data at the onset of a service is among the classical and well known problems of user modeling and recommender systems – the “cold start” problem [Guo 1997].

Nowadays, as we surf the web and visit websites, we leave identifiable digital “fingerprints”, not to mention explicit definition of interests and preferences that are used (or may be used) for personalization purposes. As a result, many Web sites contain partial and specific user models which reflect user characters that are relevant for personalizing the services they provide. In recent years, social networks that contain freely available and diverse information about users became a major source for personal information [Boyd 2007]. The freely available personal information, scattered over various online sources (including social networks) may be a valuable source of information for building an initial user model for recommendation. However, even though these social networks may be rich in personal information, they may lack specific personal information that is required for a specific personal service requested.

In order to address this issue, of using information available about a user in one domain for recommendation in another domain, “cross domain” recommendation/personalization was defined – how can we use personal information available about the user in one domain for providing service in another domain [Berkovsky et al. 2007]. Still, user models “Mediation”, the solution suggested by [Berkovsky et al. 2007], requires some semantic knowledge and specific mediation mechanism. Other interoperability approaches surveyed by Carmagnola et al. [2007] for exchanging users data through cross applications.

The proposed research is intended to address the issue of cross domain recommendation by integrating personal information that is freely available in social networks with a simple yet powerful graph based representation. User characteristics will be represented by nodes and relations between them will be represented by edges. Traversing the graph will enable to find out relations and links between characteristics that were not explicitly defined in the original information sources. We plan to explore, demonstrate and evaluate the ability to use graph based representation of user modeling for representing and reasoning on data elicited from social network in order to help solving the cross-domain user modeling challenge.

# Background and Related works

## Background

### Recommender systems

Recommender systems are now an integral part of some e-commerce sites such as Amazon.com and CDNow [Schaferet al. 1999]. They apply knowledge discovery techniques to the problem of making product recommendations during a live customer interaction. These systems are achieving widespread success in E-commerce nowadays, especially with the advent of the Internet [Sarwar et al. 2000]. There are many types of recommendation systems and each one of them may have different approach for recommendation (see as examples Pandora[[1]](#footnote-1) used content-based and collaborative filtering (hybrid), Google[[2]](#footnote-2) search update search after crawling the web it's basically a content base technique, YouTube[[3]](#footnote-3)and Amazon[[4]](#footnote-4) uses collaborative filtering etc.).

There are several types of recommendation techniques when the major fundamental techniques are:

#### Content-Based,

Recommendations are based on matching semantic properties (preferences) of items similar to those that user liked in the past. A content-based recommender learns a profile of the user’s interests based on the features present in objects the user has rated. The user model depends on the learning method employed. There are many methods for establishing content based recommendations, including decision trees, neural nets, vector-based approaches and more Burke [2002]. Adomavicius et el. [2005] conclude the description about content based recommendation technique, as: when we Recommend a new item assigned to “similar” families of items when the estimation will base on recommendation utility. for example let’s assume we have music recommender system: In order to recommend song to user, the system will try to understand the commonalities among the music the user has rated highly in the past and then only songs that have the highest similarity degree to user’s learned preferences will recommended.

**Content-based disadvantage** when recommendations are based on past user preferences and will recommend only “more of the same” – items that are similar to those the user liked. Another problem is the start-up problem in that they must accumulate enough ratings to build a Reliable classifier [burke 2002].

#### Collaborative-Based

This technique implemented in variety of commercial systems, where the recommendations are based on “mutual taste” as represented by previous ratings of users to items, with the assumption that users who agreed in the past on item ratings are likely to agree again in the future, in some case the ratings is binary (like/dislike) like Pandora or real-valued indicating degree of preference like movie rating in IMDB. Some of the most important systems using this technique are GroupLens/NetPerceptions [Resnick et al. 1994], Ringo/Firefly [Shardanand & Maes 1995], Tapestry [Goldberg et al. 1992] and Recommender [Hill et al. 1995]. These systems can be memory-based[[5]](#footnote-5) or model base[[6]](#footnote-6) to make predictions [Breese et al. 1998]. Model-based recommenders have used a variety of learning techniques including neural networks [Jennings & Higuchi, 1993], latent semantic indexing [Foltz, 1990], and Bayesian networks [Condliff, et al. 1999].

The collaborative technique is completely independent of any machine-readable representation of the objects being recommended, and works well for complex objects such as music and movies where variations in taste are responsible for much of the variation in preferences. Schafer et al. [1999] call this “people-to-people correlation.

**Collaborative-based disadvantages** – This approach suffers from cold start problem- when recommendation is needed to new user with too few ratings or when there is a new item to recommend since these recommender systems depend on overlap in ratings across users and have difficulty when the space of ratings is sparse: few users have rated the same items. [burke 2002],

#### Hybrid systems

”Hybrid recommender systems combine two or more recommendation techniques to gain better performance with fewer of the drawbacks of any individual one” [burke 2002]. Since hybrid systems are combination of several techniques they have the abilities to overcome on each techniques weakness.

Burke [2002] surveys additional common technique like Demographic, Utility-based and Knowledge-based.

Generally speaking, recommender systems are commonly based on estimating rating technique which saved on internal data information while recommendation algorithm is gain throw users rating [Adomavicius et al. 2002]. So far, most recommender systems are domain-specific. As a result of that users need to maintain different profiles on different systems causing to interspersion of user model data through separated systems when on each system need to initialize, maintain and collect the same user data. most recommendation systems in particular systems how used collaborative filtering are suffering from the cold start problem can't fill the knowledge gaps they need even when the data is exist on the web in different systems or different domains Since is not consider as solid stat of user preference .

### Graphs as data structures

A graph is a representation of a set of objects where pairs of objects are connected to each other by links. The interconnected objects are represented by mathematical abstractions called vertices, and the links that connect some pairs of vertices are called edges (corman et el 1990[[7]](#footnote-7) ). Graphs are widely used for modeling complicated data, including chemical compounds, protein interactions, XML documents, and multimedia [jiang 2007]. The main advantage of using graphs to model data is the set of theory, methods and abilities to traverse and reason on them.

### Graph traversal

Graph traversal (the search problem) is the problem of visiting all the nodes in a graph in a particular manner, updating and/or checking their values along the way. Various algorithms exist for traversing graph for like BFS[[8]](#footnote-8),DFS[[9]](#footnote-9) ,Dijkstra[[10]](#footnote-10)[Cormen et al 1990] , a different way is to change graph structure for reduce searching time ,other common approach is to represent both graphs and queries on graphs by sequences, thus converting graph search to subsequence matching[jiang 2007].

### Social networks (SN)

Social networking service or at the they short name Social networks are an online services, platform, or sites that focuses on facilitating the building of social networks or social relations among people who, for example, share interests, activities, backgrounds, or real-life connections. A social network service consists of a representation of each user (often a profile), his/her social links, and a variety of additional service.[Wikipedia[[11]](#footnote-11) 2012] They have been with us since 1997 (the first one was sixDegrees.com), social networks site (SNS) have successfully changed worldwide communication. They gave personal users the ability to reach any user in the world. They attracted millions of users, many of whom have integrated these sites into their daily practices. As of this writing, there are hundreds of SNSs, with various technological affordances, supporting a wide range of interests and practices (for example Facebook[[12]](#footnote-12) ,Google+[[13]](#footnote-13) ,twitter[[14]](#footnote-14) ,Linkedin[[15]](#footnote-15) etc) those abilities allow SNS the to connect between separate type of population using SNS users, boyd [2007] rise the fact the SNS can provide rich sources of personalize data. Profile and linkage data from SNSs can be gathered either through the use of automated collection techniques or through datasets provided directly by the company, enabling network analysis researchers to explore large-scale patterns of friending, usage and other visible indicators [Hogan, B 2007], continuing an analysis trend that started with examinations of blogs and other websites.

SNS basically contain social circles when each one of those circles can relate to different aspect. For example a regular user in LinkedIn (LinkedIn a professional SNS specialize on work relation between work colleagues) will have a work circle but s/he also can be at a different circle like friend from school or military service. A similar concept can append in Facebook: A user may have friends from different circles: school, university, work place, neighborhood, preferred music, food etc. in Google+ they even coded this feature as you can create or join to “circle”. Important circle is shared interests and preference circle, in this circle users like to connect to each other through shared subjects. For example fans group of rock band, movie fans, members at sushi restaurant etc. These values can establish large data collection of user’s preference and interests. This effort of collecting data have been mention before by Rhodes, Bowie and Hergenrather[2003] that concluded that using the web as empiric tool for behavioral science research will increase the tested population from local to global distribution.

#### Social Network (sn) as a source

Social networks (SN) can be effective sources for establish database, the main key in social networks is individual sharing to the common population.

Abdesslem, Parris, and Henderson [2011] concluded the use of SN for collecting data, they speared the collecting to two sections – collect user social behavior and collect user characteristics. From their aspect used SN as our source not only collect user preference and characteristics, we also can create social profile from the his data.

The second issue rise is how to collect random user data but still earn user relations? Fehmi [2012] used Facebook, for creating random sampling, in is work he created recursive process which extract new users from each user friends this approach gain random sampling.

## Related work

### Generic Semantic-based Framework

Fernández-Tobías et al. [2011] try to create an automated recommender system in two different domains. In their approach they used graphs for mapping connections between music and locations domains . this approach adopt the *Content-based recommendations* mention on paper by Adomavicius1 and Tuzhilin [2005] ,

In their system they develop knowledge-based description frameworks built upon semantic networks. They used DBpedia as theirs database source .( DBpedia is a graph based database that contains values from Wikipedia) for establishing a connection graph between music and architecture . Using graph analysis technique they successfully establish recommender algorithm for those two different domains.

### Taste Fabric of Social Networks

Liu, et. al. [2006] mined 100,000 social network profiles, using machine learning technique they segmented them into interest categories such as music, books, films, food, etc. . Liu examined ways for understanding user tastes. In his technique he establish semantically flexible user representations data structure with he called it "taste fabric " with help him to constitute an alternate network structure that help to provide the recommendation algorithm .his recommendation was cross-domain and based on users tastes and help to understand the computation of taste-similarity between people.

### On the Social Web

Abel and Herder [2011] developed and evaluated the performance of several cross-system user modeling strategies in the context of recommender systems. They analyzed large dataset of more 25,000 user profiles from Facebook, LinkedIn, Twitter, Flickr and Delicious. The aggregated data was then integrated into a single source and used for improving recommendation results. The evaluation results showed that the proposed method solve the cold-start problem and improved recommendation quality significantly, even beyond the cold-start.

## Summery

Recommendation systems use people’s feedback about items in in order to help people choose other items, usually in a specific domain. The Major barriers for interoperability of recommender systems are that they use different recommending techniques, in specific and different domains with different user contexts [kuflik 2012]. The result is that on the World Wide Web users have incomplete as well as duplicated personal information, data that was available in a system in one domain is not available systems in other domains. Berkovsky [2008] understood this problem and proposed a general framework for enhancing the accuracy of user modeling in recommender systems. He he suggested user models mediation process that is cross-user, cross-item, cross-context and cross-representation, in his research he developed a generic mediation mechanism for integrating user modeling data in a distributed environment. However, his approach requires the creation of specific “mediating mechanisms” between techniques and domains.

We want to take this solution a step forward and used Berkovsky’s conclusion was to create cross-domain recommendation system for representing more accurate user model which covers many different domains. with the same technique like Liu we can collect user interest preference through social networks and used machine learning techniques for analyzing interest connections. To overcome the cold start problem of recommender systems we can construct contend base recommender system which used on graphs for mapping user interest relations. Fernández-Tobías et al 2011 mapped using graph relations between music and location interest we want to create generic process with abilities to map any type of interest over large scale graph. Recommendation process can be applied through graph traversal, for finding the optimal recommendation.

# Research Goals and Questions

The “Cold Start” problem is a well-known problem in user modeling and recommender systems – how to bootstrap a user model in order to provide the user with a specific personalized service. Given the fact that a lot of personal information may be available in various sources, and the fact that this information may not exactly represent the user interests/needs/preferences in the target domain, a question is how can we use information available about a user in one domain for modeling the user in another domain, or “cross domain” modeling or recommendation.

As social networks are known to be rich source for freely available diverse personal information, we plan to explore the use of such source for cross domain recommendation. The goal of the proposed research is to explore the possibility to use freely available information on a social network for cross-domain recommendation using a graph representation of a user model.

It is assumed that the wealth and variety of information that is available in social networks can be used for cross-domain user modeling when represented in a simple graph data structure and by applying generic graph search techniques.

The research will answer the following question:

***How we can use the social network of curators for mining the links between topics of interest for cross domain recommendation using graph techniques?***

# Tools and Methods

## Methods

The research is a design research [Hevner et al. 2004]. As such, an experimental tool will be built. It will be used for representing user models over a graph and graph based techniques will be used for cross domain recommendation generation. Personal information will be collected from web Social network called Pinterset by our research tool Called **TraitsFinder.** The information will then be uploaded to the graph using also **TraitsFinder.**

### Data Source

Our data source is users’ personal preferences that exit at a SN called Pinterest. From which we can extract users albums that represent user interests like hobbies food, music etc. Since we want to rely on social networks (SN) as source, the logical conclusion was to use Facebook, since (at least for now) Facebook is the biggest SN that exists, it has more them one billion members, it is frequently updated – today most of their users update data using mobile devices and due to its popularity it covers almost any type of population at any age. However, after investigation with Facebook API we discovered that Facebook does not allow developers to directly collect data from Facebook users. Instead, a regular user needs to access to some kind of application (it’s can be game, puzzle, quiz interview or any application we want) and once the user accessed the application then through the application researchers/developers can access the user’s profile using Facebook query language (FQL). Hence we decided to abandon Facebook for the following problems:

* **Sampling problem** - in Facebook we can’t sample random users’ instants and we need to create some kind of “bait” for calling our users, with this action our effort will be leaning for particular population (population that was interest in our application).
* **Circle problem -** since we are not sampling random users we can’t get access to users’ friends. We can only publish in our participant users’ wall – this action will not help to expand our sampling, intend it’s will create a circle of users that will use the application.
* **Semantic problem –** in Facebook users upload pictures, update status, join groups, check in places etc. from all these actions it is hard to understand user characteristics and there is a need for a sematic parser for analysis of user preferences and traits. For example if user uploads a picture of a birthday cake with no explanation, what can we understand from this picture? That he has birthday party? That he loves to bake cakes? Or he just loves cakes?
* **Legal issues** – if we use Facebook we will need to ask or mention to the user this is academic experiment - this can also harm our user sampling.

Unlike Facebook we can use Pinterset SN for collecting relevant users’ information. **Pinterest** is a [pinboard](http://en.wikipedia.org/wiki/Pinboard)-style [photo sharing](http://en.wikipedia.org/wiki/Photo_sharing) website that allows users to create and manage theme-based image collections such as events, interests, hobbies, and more [wikipadia[[16]](#footnote-16)]. Pinterest is not only simple and have specific attribution we need Pinterest users can create albums, usually those albums represent their interest we can consider those albums as cataloged subjects. , we also get the connection between users – when user upload photo and catalogues it, any other user that will pin this picture is now connected to this picture. Therefore we can seek from each picture new users with correlation to the selected user. We have also very big advantage in Pinteres the subjects are basically our interest that we seek. In addition we don’t have to become entangled with random sampling issue – when can just sample all the users through recursive crawling process base on pictures pin.

### Crawling Pinterest

Unfortunately Pinterest does not have an API, thus in order to obtain users’ data we are required to download and parse Pinterest HTML web pages and extract the data from each page manually (aka Scraping). Pinterest website structure is based on folders hierarchy, where each folder has subfolders (see Figure 3 for the structure).

Figure 3 : Pinterest hierarchy

Since the hierarchy is relatively simple, we can explore the website and construct a tool for collecting relevant data, the crawling algorithm is describe as follows:

*Go to main site Pictures page G*

*Crawl(****G****)*

*{*

*If* ***G*** *is empty exit*

*Else*

*{*

*Find pictures* ***P*** *from* ***G***

*Save Comment* ***C*** *from* ***P*** *under* ***P***

*Foreach user* ***X*** *in* ***C***

*{*

*Add user* ***X*** *to group* ***U***

*Foreach subject* ***S*** *in* ***X***

***{***

*Save subject* ***S****i under user* ***X***

*Save all items (it) under* ***S****i*

***}***

*}*

*Crawl(X)*

*Crawl(U)*

*}*

*}*

### Data collection and graph representation

The data we are going to extract will be saved in files for each: user, subject, picture and comment. The crawling process will convert the HTML pages to standard XML files while maintaining the folder hierarchy of Pintrest (see figure 3). The same approach will be used to save the subjects and pictures page files.

The crawling process will create hierarchies of folders and xml files. From those files we can parse the content and use it to build graph based model.

#### Graph Based Model of Pinterest’s Data

In order to be able to analyze the user data available at Pinterest using graph based methods, we are required to map the extracted entities and their relations to a graph based model. Following is a description of the website’s entities and their equivalents in the graph based model.

The graph model representing Pinterest’'ss data will marked as G. Each vertical (V) in the graph will represent an object in Pinterest hierarchy, e.g., a user, subject , or item (aka “Pin”). The edges (E) in the graph will connect two vertices, in case they are linked to each other in Pinterest’s hierarchical module, hence E ⊆ {u,v∈V }. For example, presume user ‘X’ that has two albums/subjects ‘Pizza’ and ‘Animals’ listed under her profile in Pinterest’s data model would be represented in the graph model by: user\_x ∈ V, Animals ∈ V, Pizza ∈ V and e1(user\_x, Animals) ∈ E, e2(user\_x, Pizza) ∈ E. If there is another user, Y, who also love animals and pizza, and interested in cars on top, he will have three edges: e3(user\_y, Animals) ∈ E, e4(user\_y, Pizza) ∈ E, and e5(user\_y, Cars) ∈ E, this is illustrated in the left side of Figure 4.

The pizza subject may contain sub items such as tuna\_pizza ,olives\_pizza and mushrooms\_pizza (tuna\_pizza ∈ V ,olives\_pizza ∈ V and mushrooms\_pizza ∈ V). Such items will lead to the creation of edges between them and the item they are related to: (tuna\_pizza ,pizza) ∈ E , (olives\_pizza, pizza) ∈ E and (mushrooms\_pizza ,pizza ) ∈ E (illustrated in the right side of Figure 4).

With this graph we can analysis the connection between each pair of objects in Pinterest’s data model, and try to infer hidden links between distant entities (e.g., being interested in cars and liking olive pizzas).



Figure 4: graphical view main graph

#### Interests connection graph

Preliminary data collection experiments on 100 user only show that the resulting graph model is quite large. For example 100crawled users there were approximately ~600000 (subjects and items) related entities which lead to the same amount of vertices being created, and about ~600resulting edges. Search and traversal times for such a graph would be very slow, especially if we plan to run an exhaustive search for hidden links between any combinations of entities. In order to improve run time performance we have decided to minimize the graph by abstracting some of the relations. The abstracted graph, G2, will contain only Subjects as vertices and thus relations between them. A edge between two subjects is created if there is at least one user who is connected to both. So each set: {v1 ∈ V(users); s1,s2 ∈ V(subjects); e1,e2 ∈ E(users🡪subjects)} would be abstracted into {s1,s2 ∈ V(subjects); e1(s1, s2) ∈ E(subjects🡪subjects)}. The edges between subjects would also contain weight labels denoting the amount of users that had those two subjects co-listed in their profiles. In our example the interest graph will have only the interest nodes: {animals, pizza, cars} = V the edges are represent the native of the interests connection when (animals, pizza)∈E & (pizza ,cars)∈E when ι(animals,pizza)=2 and ι (Cars,Pizza)=1 the weight animals-pizza edge is 2 since user\_x and user\_y are both like animals and pizza.



Figure 5: graphical view Interests connection graph

Eventually we will have weighted undirected graph that will represent the our graph database when the number of nodes (vertex) is equal to the number of users interests , |V|=numof(interests) . by this attitude we can reduce graph nodes and edged – in our example the Pinterest's data graph contain 8 nodes with 9 edges , the same repression in interests connection graph will lead to 3 nodes with 2 edges only . this way we can analyze the power of each interest connections (edgeds weight) which lead to rating recommendation algorithm. .

## Tools

We will construct a research tool called **TraitsFinder** that will allow us to collect user’s information and extract the data [[17]](#footnote-17)from a specific social network – Pinterest and build a graph that will represent the connection between traits, the research will work by two steps:

* **TraitsFinder** will crawl pinterest social networks and collect user’s information in our servers, the outcome of this step is users folder with user information save in XML files.
* **TraitsFinder** will create graph base users crawled data – this step can run in offline mode or online (user information are update immediately after saved) the outcome of this step update neo4j graph.

For the purpose of the planned research to use the following tools:

* **TraitsFinder** - we will construct a multithread application cross OS research tool called TraitsFinder, implemented in java that will collect data from Pinterest website. TraitsFinder will Crawl Pinterest and will save data as local xml files. It will have also the ability to upload the information to graph database.
* **Neo4J[[18]](#footnote-18)** – is a high-performance, NOSQL graph database with all the features of a mature and robust database. **TraitFinder** will upload user’s interests to Neo4j graph.
* **Gephi[[19]](#footnote-19)** – Gephi is an interactive visualization and exploration platform for all kinds of networks and complex systems, dynamic and hierarchical graphs. Gephi will be used to survey and analysis neo4j graph .
* **Gremlin** – one of TinkerPop[[20]](#footnote-20) tools , Gremlin is a domain specific language for traversing property graphs – with gremlin we can mediate between Neo4j graph to different graph libraries.

## Evaluation

The proposed research will be evaluated by collecting publicly available data from social networks regarding users’ preferable items and using it to train a graph based recommendation engine. Approximately 100,000 profiles are to be crawled; initial test with 1000 crawled users show an average of 30 subjects of interests per user, with 20 items in average listed under each interest album.

The cross domain recommendations will be evaluated by using a 10 fold validation as described in Kohavi [1995]. Of the collected data, 90% will be used to train the recommendation engine and 10% for its evaluations. The 10% selected for evaluation will be changed during the evaluation process to include different features of the data. Once the evaluation iterations will result with a minimal error (matching recommendations considered as ‘hits’) single test iteration will provide a final result.

As part of the evaluation it is also intended to analyze how the size of the dataset/size affects recommendations results. This will be done by taking different subsets of the available data and measuring the changes in recommendations quality.

Since our system is cross-domain recommendation system the variety of interest is basically infinite Kohavi [1995] have been investigating using cross-validation and bootstrap for analyzing bias learning. He concluded the more the K-fold is bigger it reduces the variance while increase the bias, since our graph will be high interests variance we will need to equalize graph to the K value in the K-fold cross validation – for better evaluation we find the K value by measuring the all graph interest nodes.

For evaluate this graph we be cross validation technique, we will run cross validation runs as described in the following table (in this example k-fold test when k=1000):

|  |  |
| --- | --- |
| Train size (creating graph based on X users) | Number of folds tested user check |
| 1000 | 1 |
| 10000 | 10 |
| 100000 | 100 |

The tested fold user will checked by checking recommitting hits, for each tested user we scan is interests and valid our recommitting algorithm on is only 30-40 % of is interests , the recommitting that TraitFiner will return will cross examined with is actually traits. Using cross validation is common technique especially for learning system, since our recommendation system is learning with graph base learning algorithm , we will applied this technique on our system.

# Timetable

Each of the phases discusses the primary focus of each time period:

Phase I - Literature Survey and Focus.

Phase II - establish first stage of TraitsFinder – the web crawler

Phase III - collecting users traits (exit criteria: at least 100,000 users)

Phase IV - broadening TraitsFinder - add graphanalyzer.

Phase V - improve TraitFinder: add automatic graph analyzer, create cluster graph

Phase VI - Graph analysis, evaluation and algorithm establish

Phase VII - Writing Thesis.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Phase | Summer2012 | Fall2012 | Spring2013 | Summer2013 | Fall2014 |
| I |  |  |  |  |  |
| II |  |  |  |  |  |
| III |  |  |  |  |  |
| IV |  |  |  |  |  |
| V |  |  |  |  |  |
| VI |  |  |  |  |  |
| VII |  |  |  |  |  |

# Initial Results

According to now we successfully establish the first stage TraitsFinderframeworks and PinterestTraitsFinder tools.

TraitsFinder Frameworks – is a generic cross platform frameworks library implemented on java, build for multithreaded crawling at any website and serialize the data at any data base , for now the TraitsFinder frameworks had the ability to save data only at neo4j graph data base. PinterestTraitsFinder used TraitsFinder Frameworks and contain implementation for crawling Pinterest's website and creating the interests connection graph.

So far PinterestTraitsFinder have been successfully crawled 35,000 users. Form these user we created 3 testes graph :

1. Interest connection graph with 5300 users – this graph it the base for the interest larger graph at the next research stage we add all the data from our crawled users.
2. interest connection graph with 500 users –with this graph we evaluated PinterestTraitsFinder , using Gephi we examine graph creation.

The collecting process is and graph creation is based on two machines, when one machine is collected user data when the second machine add user's data to interest graph.

# Research Contributions

The proposed research main contribution to the field of user modeling will be a cross domain recommender algorithm. The algorithm will apply graph analysis methods for the purpose of recommendation generation and will be based on data (interests) extracted publicly available data. It will suggest and demonstrate a generic, graph-based approach for cross-domain recommendation using social networks data that can be applied in variety of contexts

# References

1. Abdesslem, F. B., Parris, I., & Henderson, T. (2012). Reliable online social network data collection. *Computational Social Networks*, 183-210.
2. Abel, F., Herder, E., Houben, G. J., Henze, N., & Krause, D. (2011). Cross-system user modeling and personalization on the social web. User Modeling and User-Adapted Interaction (UMUAI), Special Issue on Personalization in Social Web Systems, 22(3), 1-42.
3. Adomavicius, G., & Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. Knowledge and Data Engineering, IEEE Transactions on, 17(6), 734-749.
4. Al-Khalifa, H. S., & Davis, H. C. (2006, November). Folksannotation: A semantic metadata tool for annotating learning resources using folksonomies and domain ontologies. In Innovations in Information Technology, 2006 (pp. 1-5). IEEE.
5. Berkovsky, S., Kuflik, T., & Ricci, F. (2008). Mediation of user models for enhanced personalization in recommender systems. User Modeling and User-Adapted Interaction, 18(3), 245-286.
6. Boim, R., & Milo, T. (2011, April). Methods for boosting recommender systems. In Data Engineering Workshops (ICDEW), 2011 IEEE 27th International Conference on (pp. 288-291). IEEE.
7. Breese, J. S., Heckerman, D., & Kadie, C. (1998, July). Empirical analysis of predictive algorithms for collaborative filtering. In Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence (pp. 43-52). Morgan Kaufmann Publishers Inc.
8. Burke, R. (2002). Hybrid recommender systems: Survey and experiments. User modeling and user-adapted interaction, 12(4), 331-370.
9. Carmagnola, F., Cena, F., & Gena, C. (2011). User model interoperability: a survey. *User Modeling and User-Adapted Interaction*, *21*(3), 285-331.
10. Carmagnola, F., Cena, F., Cortassa, O., Gena, C., & Torre, I. (2007). Towards a tag-based user model: how can user model benefit from tags?. User Modeling 2007, 445-449.
11. Condliff, M. K., Lewis, D. D., Madigan, D., & Posse, C. (1999, August). Bayesian mixed-effects models for recommender systems. In Proc. ACM SIGIR (Vol. 99).
12. Ellison, N. B. (2007). Social network sites: Definition, history, and scholarship. Journal of Computer‐Mediated Communication, 13(1), 210-230.
13. Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P.,and Uthurusamy, R., Eds. (1996) - Advances in Knowledge Discovery and Data Mining
14. Fernández-Tobías, I., Cantador, I., Kaminskas, M., & Ricci, F. (2011, October). A generic semantic-based framework for cross-domain recommendation. In Proceedings of the 2nd International Workshop on Information Heterogeneity and Fusion in Recommender Systems (pp. 25-32). ACM.
15. Foltz, P. W. (1990, March). Using latent semantic indexing for information filtering. In ACM SIGOIS Bulletin (Vol. 11, No. 2-3, pp. 40-47). ACM.
16. Guo, H. (1997). Soap: Live recommendations through social agents. In Fifth DELOS Workshop on Filtering and Collaborative Filtering, Budapest.
17. Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. MIS quarterly, 28(1), 75-105.
18. Hill, W., Stead, L., Rosenstein, M., & Furnas, G. (1995, May). Recommending and evaluating choices in a virtual community of use. In Proceedings of the SIGCHI conference on Human factors in computing systems (pp. 194-201). ACM Press/Addison-Wesley Publishing Co.
19. Hogan, B. (2008). Analyzing social networks via the Internet. Sage Handbook of Online Research Methods. Thousand Oaks, CA: Sage.
20. Hogan, B. (2008). Analyzing social networks via the Internet. Sage Handbook of Online Research Methods. Thousand Oaks, CA: Sage.
21. Jiang, H., Wang, H., Yu, P. S., & Zhou, S. (2007, April). “Gstring: A novel approach for efficient search in graph databases”. In Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference on (pp. 566-575). IEEE.
22. Dean, J., and Ghemawat, S. “MapReduce: Supplied Data Processing on Large Clusters” *Google, Inc (2004)*.
23. Jennings, A., & Higuchi, H. (1993). A user model neural network for a personal news service. User Modeling and User-Adapted Interaction, 3(1), 1-25.
24. Liu, H. (2007). Social network profiles as taste performances. Journal of Computer‐Mediated Communication, 13(1), 252-275.
25. Liu, H., Maes, P., & Davenport, G. (2006). Unraveling the taste fabric of social networks. International Journal on Semantic Web and Information Systems (IJSWIS), 2(1), 42-71.
26. Resnick, P., N. Iakovou, M. Sushak, P. Bergstrom, and J. Riedl. GroupLens 1994: “An open architecture for collaborative filtering of netnews”. In Proceedings of the 1994 Computer Supported Cooperative Work Conference.
27. Rhodes, S. D., Bowie, D. A., & Hergenrather, K. C. (2003). Collecting behavioural data using the world wide web: considerations for researchers. Journal of Epidemiology and Community Health, 57(1), 68-73.
28. Kohavi, R. (1995) A study of Cross-validation and bootstrap” for accuracy Estimation and model selection” ,Stanford university.
29. Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2000). Application of dimensionality reduction in recommender system-a case study (No. TR-00-043).Minnesota university ,dept of computer since. MINNESOTA UNIV MINNEAPOLIS DEPT OF COMPUTER SCIENCE.
30. Schafer, J. B., Konstan, J., & Riedi, J. (1999, November). Recommender systems in e-commerce. In Proceedings of the 1st ACM conference on Electronic commerce (pp. 158-166). ACM.
31. Shardanand, U., & Maes, P. (1995, May). Social information filtering: algorithms for automating “word of mouth”. In Proceedings of the SIGCHI conference on Human factors in computing systems (pp. 210-217). ACM Press/Addison-Wesley Publishing Co.”.
32. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rives Introduction to Algorithms chapter VI Graph Algorithms (1990)
33. Kuflik, T., Kay, J., and Kummerfeld, B., (2006) “Challenges and Solutions of Ubiquitous User Modeling”
34. Jennings, A., & Higuchi, H. (1993). A user model neural network for a personal news service. User Modeling and User-Adapted Interaction, 3(1), 1-25.

1. <http://www.pandora.com> [↑](#footnote-ref-1)
2. <http://www.google.com> [↑](#footnote-ref-2)
3. <http://www.youtube.com> [↑](#footnote-ref-3)
4. <http://www.amazon.com> [↑](#footnote-ref-4)
5. **Memory-based** Collaborative Filtering Algorithms –" Memory-based algorithms utilize the entire user-item data-base to generate a prediction. These systems employ statistical techniques to find a set of users , known as neighbors, that have a history of agreeing with the target user. Once a neighborhood of users is formed, these systems used different algorithms to combine the preferences of neighbors to produce a prediction or top-N recommendation for the active user." [Sarwar at el 2000]. [↑](#footnote-ref-5)
6. **Model-based** Collaborative Filtering Algorithms provide item recommendation by first developing a model of user ratings base on probabilistic algorithm as first step of prediction , using machine learning [↑](#footnote-ref-6)
7. http://en.wikipedia.org/wiki/Graph\_%28mathematics%29 [↑](#footnote-ref-7)
8. breadth-first search (**BFS**) is a [strategy for searching in a graph](http://en.wikipedia.org/wiki/Graph_search_algorithm) when search is limited to essentially two operations: (a) visit and inspect a node of a graph; (b) gain access to visit the nodes that neighbor the currently visited node. The BFS begins at a root node and inspects all the neighboring nodes [↑](#footnote-ref-8)
9. Depth-first search (**DFS**) is an algorithm for traversing or searching a tree, tree structure, or graph. One starts at the root (selecting some node as the root in the graph case) and explores as far as ossible along each branch before backtracking [↑](#footnote-ref-9)
10. **Dijkstra's algorithm**, conceived by Dutch computer scientist Edsger Dijkstra in 1956 and published in 1959,[1][2] is a graph search algorithm that solves the single-source shortest path problem for a graph with nonnegative edge path costs, producing a shortest path tree. This algorithm is often used in routing and as a subroutine in other graph algorithms. [↑](#footnote-ref-10)
11. <http://en.wikipedia.org/wiki/Social_networking_service> [↑](#footnote-ref-11)
12. <https://www.facebook.com/> [↑](#footnote-ref-12)
13. https://plus.google.com/ [↑](#footnote-ref-13)
14. https://**twitter**.com/ [↑](#footnote-ref-14)
15. www.linkedin.com [↑](#footnote-ref-15)
16. http://en.wikipedia.org/wiki/Pinterest [↑](#footnote-ref-16)
17. www.Pinterest.com [↑](#footnote-ref-17)
18. http://neo4j.org/ [↑](#footnote-ref-18)
19. https://gephi.org/ [↑](#footnote-ref-19)
20. http://www.tinkerpop.com/ [↑](#footnote-ref-20)