Multi-Agent Domain Model of Risk Aware and Privacy Preserving Web Recommender System

By

Vishnu Srivastava

A thesis
Presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Computer Science

Waterloo, Ontario, Canada, 2017

AUTHOR'S DECLARATION

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners. I understand that my thesis may be made electronically available to the public.

Abstract

Recent progress in the field of recommender systems has led to the increase in the accuracy and better personalization of the recommendations. These results are being achieved by gathering more user data and generating insights from it. However the privacy concerns of the user are often under estimated and ignored. In fact, many users are not sufficiently aware of the data that is collected or if such data is sold to third party. Moreover, the generated recommendations do not take into account the factors such as user intention and the current situation of the user situation.

Research in the area of web recommender system should strive towards not only achieving high accuracy of the generated recommendations but also maintain user privacy and making recommender systems aware of the user context in terms of intentions of the user and the current situation of the user. Through research it has been established that a tradeoff is required between accuracy, privacy and risk in a recommender system and that it is highly unlikely to have recommender system having a high value of all those attributes. Nonetheless, a significant attempt can be made to describe a system based on the level of inclusion of those attributes within the recommender system.

This thesis focuses on the multi agent domain model of a web recommender system in which a system is described as a point in the privacy space and the risk space. The web recommender system is viewed as consisting of multiple subsystem, operated by an agent having a specific role in order to achieve a prescribed goal by performing activity to sustain that subsystem. Such a description of a system will be able to represent a small subset of web recommender systems which can be classified as risk aware and privacy preserving web recommender system. A case study of stock recommendation system will reveal that such system can be enabled to have risk aware and privacy preserving features to be enlisted as a subset of recommender systems that can be described by the multi agent domain model described in the paper.

Acknowledgements

I would like to thank Professor Paulo Alencar, for his patience, understanding, kindness and guidance. I learned a lot while working with him and I am proud to be his student. I am thankful to Professor Daniel Berry for serving as my co-supervisor.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

In recent times, Recommender systems can take advantage of semantic reasoning capabilities to overcome common limitations and improve the recommendations quality. These systems uses domain ontologies to enhance the personalization. Current research in the area of recommender system has been focussed on the context aware recommender system. A context-independent representation may lose predictive power because potentially useful information from multiple contexts is aggregated. The ideal context-aware recommendation system would, therefore, be able reliably to label each user action with an appropriate context and effectively tailor the system output to the user in that given context. The majority of existing approaches to recommender systems focus on recommending the most relevant content to users using contextual information and do not take into account the risk of disturbing the user in specific situation. However, in many applications, such as recommending personalized content, it is also important to consider the risk of upsetting the user by not being aware of the user's situation and intentions. Therefore, the performance of the recommender system depends in part on the degree to which it has incorporated the risk into the recommendation process. The risk in recommender systems is the possibility to disturb or to upset the user which leads to a bad feedback from the user. With the advent of enormous amounts of personal data collection for the sake of personalization, the focus of the current research on the recommender system has been shifting to Privacy Protection. Personalization provides users with conveniences, and it can have a direct impact on marketing, sales, and profit. On the other hand, privacy, which is a serious concern for many users, is the price users have to pay for the convenience of recommender systems in a world with booming information. Users normally have no choice but to trust the service provider to keep their sensitive personal profile and information safe.

1.1 Research Issue

Since the majority of the focus in the area of recommender systems has been on the improvement of accuracy of the recommendations generated by the recommender system, it makes it really hard to have a domain model of the recommender system which takes care of the aspect of upsetting the user by not having sufficient knowledge of the user’s context and not taking enough measures to take into account the privacy of the users. A domain model of such a system would make things much easier for domain experts to study and extend the research in the area of risk aware and privacy preserving recommender systems.

The past ten years has seen both an increase in the size of development teams and a proliferation of software research in the form of empirical studies providing insight and tools providing aid for the commensurate collaboration that is required in such projects. Despite these successes, there is still much to be done, both because there still exist chasms between what is needed and what has been provided today and because the software development landscape is changing rapidly, particularly in the web development spaces. Multi-Agent software development is being utilized as a way to develop software by working on different aspect of a software system as separate agents, working in coherence to achieve the objective of the system. But a multi-agent approach to risk-aware and privacy protection is much scarce.

Most of the current research is focused on evaluating recommender system in terms of the accuracy of the results produced by such systems. There needs to be a way to evaluate a domain model of a recommender system, not just on the accuracy of the underlying algorithm but also on the features it contains to enhance the user experience such as use’s intention, context and also the privacy of the data that is used to produce recommendations.

1.2 Thesis Statement

The aim of this research is to provide a system model of a recommender system keeping in mind the privacy aspect and the risk aspect of the user data and information in order to describe and evaluate a large subset of recommender systems using this approach in terms of level of privacy and utilization of the user information by the recommender system.

1.3 Major Contributions

Nowadays, most of the research on evaluation of recommender systems is based on the increase in accuracy of the recommender systems. This research focusses on the privacy and the risk aspect of the recommender systems i.e. how much a recommender system safeguards users' privacy and also how a recommender system utilizes the contextual data of a user in order to benefit the owner of the information as well as other users of the recommender system.

The approach utilizes multi-agent system description in a sense that the designers of the recommender systems can focus on individual units. This breakdown of the recommender system into small individual units enables fast and fault tolerant development of the system. It enables the designers of the recommender system to be aware of the each of the small objectives that must be accomplished by the each individual units in order to fulfil the objective of the entire system.

A five dimensional approach, discussed in this paper can be used to gain information about the recommender system functioning across different application areas and a common metrics can be established in the future to better evaluate different recommender systems using a common evaluation tool.

This High level approach to describe the system is helpful for domain experts to gain valuable knowledge of a recommender system, operational in a particular application area in a short period of time.

1.4 Thesis Organization

The thesis is divided into three parts. The first part introduction of the problem addressed in this paper, along with a survey of the Recommender Systems field and brings into light the risk and privacy issues, where this thesis is framed. The second part describes the related work in the recommender systems literature and provides an analysis of the design alternatives and statistical biases that may arise. It also provides a detailed discussion of the proposed approach to solve the issues with the existing domain models of the web recommender system. Towards the end of this part a brief case study of this domain model is show in context of job recommender systems. The last part describes the work to be done in the future to extend this system model.

In more detail, the contents of this thesis are distributed as follows:

Part I. Introduction

Chapter 1 In this chapter, a brief description of the recommender system has been provided followed by the problems encountered by the researcher and the domain experts regarding the recommender systems. The thesis statement is then provided to give an idea of what this paper is trying to achieve. This is followed by listing out some of the major contributions of this paper.

Chapter 2 provides an overview of the state of the art in recommender systems, considering a classification of the main types of recommendation approaches. We also describe the weaknesses of the different recommendation techniques and present a broader class of hybrid recommenders that aim to overcome these limitations. We also discuss the risk and privacy issues in the recommender systems. And how these issues arises in the first place in such systems. The discussion is carried forward with the description of the some of the mitigating techniques that can be used to address this problem.

Part II. The System Model

Chapter 3 mentions some of the works in the field of recommender system that has contributed toward the conceptualization of the approach that is discussed in the paper.

Chapter 4 presents the proposed approach. In this section, a detailed description of the system is provided along with the explanation of different aspects of this model.

Chapter 5 presents the case study of the proposed approach in terms of the job recommender system. In this chapter a discussion of the previous approach to a job recommender system has been provided, followed by providing substantial evidence that such systems are a subset of the recommender approach which is discussed in this paper. I also discusses and enhancement over the previous approach in order to make such recommender system privacy preserving and risk aware.

Part III. Future work

Chapter 6 Discusses some of the enhancements that can be carried out in the future to improve upon this approach of evaluating recommender systems across different application areas.

Chapter 2

Recommender Systems

Recommender systems are said to typically produce a list of recommendations in one of two ways, through collaborative and content-based filtering or the personality-based approach.[20] Collaborative filtering approaches building a model from a user's past behavior (items previously purchased or selected and/or numerical ratings given to those items) as well as similar decisions made by other users. This model is then used to predict items (or ratings for items) that the user may have an interest in.[22] Content-based filtering approaches utilize a series of discrete characteristics of an item in order to recommend additional items with similar properties.[23] These approaches are often combined into Hybrid Recommender Systems.

2.1 Context Aware Recommender Systems

When recommending a personalized content, it is not sufficient to consider only user's profiles and documents. It is also important to recommend documents adequate to the user's situation. Therefore, a good recommendation depends on how well the recommender system (RS) has incorporated the relevant contextual information into the recommendation process. Recently, some RS have taken the context into account, being called Context-Aware Recommendation System (CARS). However, for a long time many works deal with the context in other areas, like IR, mobile-learning and advertising since context become inescapable. The notion of context appeared in several disciplines, like computer science, linguistics, philosophy, psychology, etc., and every discipline gives its own definition, often different from the others, which is more specific than the generic definition i.e. "conditions or circumstances that have an effect on something". Therefore, there are several definitions of context across varied disciplines. In context-aware computing, the authors in [21] have considered the context as a key component to increase human-machine interactions, and they have given the subsequent definition of context that is now ordinarily accepted: “Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves.”. According to Grudin [24], the context acquisition is the process through which contextual information is captured. The context can be obtained by different methods, depending on the contextual information that the system needs. Context models formalize the representation of the context as a structure (ontology, class of vectors of terms, set of concepts, etc.) or a set of specific and different information structures. We present now the most interesting models found in the literature. The most simple context models are based on attribute-value pairs to represent context, where attributes capture various characteristics of contextual elements.

2.2 Privacy in Recommender Systems

In [19] the privacy in recommender systems is described as the user’s need to reveal information in order to make use of the desired functionality of a recommender system, a trade-off exists between utility and user privacy. Obtaining accurate recommendations is one thing, but sharing personal information may also lead to privacy breaches. In this section, we will look into privacy in recommender systems and potential privacy concerns with a focus on user privacy.

2.2.1 Information in Recommender System

We will summarize the types of information typically used in a recommender system and the information flow in the recommender system. This is a brief discussion to explore the diversity of information used in recommender systems

Behavioral information is the implicit information that the recommender system can gather while the user interacts with the broader system. For example, product views in a webshop or not fully watching a movie on a video on demand site.

Contextual information describes to the context in which a recommendation query is made. Common examples of contextual information are location, social group, time, date, and purpose.

Domain knowledge specifies the relationship between a user stereotype and content items. Domain knowledge is usually static but can change over time. *Item metadata* is descriptive information about content items. Examples of metadata are kitchen for restaurants, genre for movies, and artist for music.

Purchase or consumption history is the list of content that has previously been purchased or consumed by the user.

Recommendations are the output of a recommender system, typically a ranked list of items. In some systems, the relevance score for each content item is also given to the user.

Recommendation feedback is information about the recommendation provided by the user. Feedback can be expressed as positive, negative, or something more nuanced (stating a reason as well).

Social information describes the relationship between different users. Many sites allow users to specify a friendship relation (or similar) to other users, community membership, or both.

User attributes describe the user. Examples of user attributes are demographic information, income, and marital status.

User preferences are explicitly stated opinions about items or groups of items. Preferences are expressed by either a scalar measure (rating items on a scale of 1–5 stars), a binary indicator (keeping a list of favorites), or text (tags and comments).

2.2.2 Privacy and Confidentiality

The word privacy has many subtly different meanings, each with their own definition. Privacy on the Internet revolves mainly around *information privacy*. Kang [20] used the wording of the Information Infrastructure Task Force (IITF), as cited below:

Information privacy is "an individual's claim to control the terms under which personal information – information identifiable to the individual – is acquired, disclosed or used."

This concept of information privacy is strongly related to the notion of *confidentiality*, from the field of information security, but not to be used interchangeably. Confidentiality is concerned with the secrecy of individual pieces of information. Information privacy focusses on the individual who is the subject of said information, the effects that disclosure have on this person, and his or her control and consent. In our overview of privacy-protection technologies, the focus will lie on preventing unwanted disclosure and usage of information, but not on the effects on the person. When using online applications, users generally share a lot of (personal) information. Whether it is uploading ratings or comments, posting personal information on a profile, or making purchases, information is always shared within a particular *scope* [28]. Privacy involves keeping a piece of information in its intended scope. This scope is defined by the size of the audience (breadth), by extent of usage allowed (depth), and duration (lifetime). When a piece of information is moved beyond its intended scope in any of these dimensions (be it accidentally or maliciously), a privacy breach occurs. So, a breach may occur when information is disclosed to a party for whom it was not intended, when information is abused for a different purpose than was intended, or when information is stored beyond its intended lifetime. Weiss [39] stated that on the traditional Web, privacy is maintained by limiting data collection, hiding users' identities, and restricting access to authorized parties only. Often, in practice, information and identity become closely linked and visible to large groups of people. Profiles may be publicly visible, comments can be seen by all viewers of a content item, and some sites list the last users to visit a particular page. It becomes harder for a user to monitor and control his personal information, as more of it becomes available online. This problem mainly applies to systems where the user logs in to an account and where tools are available to express a user's preferences. Often, users are not very aware of their (lack of) privacy. In a study on social network users in particular, Gross and Acquisti [18] showed that most users do not change the default privacy settings, while sharing a large amount of information on their profile. Tufekci [38] concluded in his case study that privacy-aware users are actually more reluctant to join social networks, but once they do join, they still disclose a lot of information. As opposed to social networks, in most recommender systems, privacy toward other users is probably not the largest issue. Users place a lot of implicit trust in service providers, expecting them to handle user information in a fair and conscientious way and continue to do so in the future. By using the system, users enter into a relationship with the service provider, who can generally view *all* information in the system, including private uploads, browsing and purchase behavior, and IP addresses. It is also the service provider who decides which information is stored, how long it is kept, and how it is used or distributed. Usually, privacy statements are offered to display the position the service provider takes and to acquire the user's consent. However, this leaves users little choice: they can either agree to the terms or will not benefit from using the system. The power balance is clearly in favor of the service provider.

Privacy breaches can involve a variety of parties (fellow users, the service provider, or outsiders) and may be a deliberate act (snooping, hacking), or accidental (mismanagement, lingering data). Depending on the sensitivity of information involved, such incidents may have serious consequences. Lam et al. [23] already identified some threats to privacy in recommender systems. Their concern is the amount of (personal) information that is collected by the service provider and the potential leakage of this information. Independent of their work, we explicitly identify the privacy concerns in recommender systems and classify them as follows:

Data Collection. Many users are not aware of the amount and extent of information that a service provider is able to collect and what can be derived from this information. This may be due to the fact that privacy statements are seldom read, and people have become used to pursuing online activities. Usually there is no way to opt-out of such data gathering, other than not using the system at all. As collection practices do not match with the users' expectations, this concern relates to the extent of information usage.

Data Retention. Online information is often difficult to remove, the service provider may even intentionally prevent or hinder removal of data. This is because there is commercial value in user information, for both competitive advantage through analysis and/or data sales. Furthermore, information that is apparently erased from one place may still reside somewhere else in the system, for example, in backups, to be found by others. The data retention concern relates to the intended lifetime, as information can be available longer than intended.

Data Sales. The wealth of information that is stored in online systems is likely to be of value to third parties and may be sold in some cases. Users' ratings, preferences, and purchase histories are all potentially interesting for marketing purposes. Data sales usually conflicts with the privacy expectations of users. Even though data is often anonymized before being sold to protect user privacy, re-identification is a threat that is often overlooked or ignored. For example, the information published by Netflix as part of their recommender systems prize, though anonymized, allowed for re-identification [27]. Narayanan and Shmatikov linked the anonymized records to publicly available records (such as IMDb) based on rating similarity and time of rating. If two records give a similar rating to a movie around the same time, they are likely to be from the same person. A higher number of similar movie ratings (in rating and in time) increases the confidence of the link between the records. This concern relates mainly to the extent of information usage.

Employee Browsing Private Information. The service provider as an entity has full access to the information, and its employees might take advantage of this. This is in conflict with the intended breadth of the audience, and the privacy that the service provider has promised its users.

Recommendations Revealing Information. Recommendations inherently are based on the information contained in the recommender system. For example, in collaborative filtering that information is the ratings of all the users, or in knowledge-based recommender systems it is the expert knowledge. Each recommendation reveals a tiny piece of information about the private information. It is unclear how a large number of recommendations impact the disclosure of information. This could be used to reveal information about other users (compromising their privacy) or information about the recommender system itself (potentially leading to reverse engineering of the system). Here, we focus on the privacy of the user, not the security of the system. Ramakrishnan et al. [31] looked at the privacy of eccentric users (users with unusual ratings) from a graph perspective. When looking at recommendation results, these users are at a higher risk than average users. As eccentric users cannot hide in crowds of other users, when their data is used for making recommendation, other data is often not. The recommendations output by the system are then based on only a few users, with a strong correlation between the input of the eccentric users and the recommendation output. This is in conflict with the intended breadth of the audience.

Shared Device or Service. Privacy at home can be just as important as privacy online. When sharing a device like a set-top box or computer, or a login to an online service, controlling privacy toward family and friends may be difficult. For example, a wife who wants to hide from her husband the fact that she purchased a gift for him. Unless she has a private account, her husband might inadvertently see her purchase or receive recommendations based on it. Many would want to keep some purchases private from their kids or their viewing behavior from their housemates. While some services allow for separate accounts, this is not always possible. For example, targeted advertising works with cookies that are stored in the browser, which is implicitly shared on a computer. This is related to the intended breadth of the audience.

Stranger Views Private Information. Users can falsely assume some information to be kept restricted to the service provider or a limited audience, when in reality it is not. This can be due to design flaws on the part of the service provider or a lack of the user's own understanding or attention to his privacy. When a stranger views such private information, there is a conflict with regard to the intended breadth of the audience. Rosenblum [33] showed, for example, that information in social networks is far more accessible to a widespread audience than perceived by its owners.

2.2.3 Privacy Protection

We have seen a wide variety of privacy issues associated with recommender systems. Research from many areas could be applied to alleviate some of the aforementioned concerns. We will provide an overview of research areas and briefly discuss their mechanisms, advantages, and limitations.

2.2.3.1 Anonymization, Randomization and Differential Privacy

Anonymization involves removing any identifying (or identifiable) information from the data, while preserving other structures of interest in the data. This mainly stems from the fact that information can only be *partially* removed or obfuscated, while other parts *must be kept intact* for the dataset to remain useful. In the real world, it is difficult to predict which external sources of information may become available, allowing pieces of data to be combined into identifiable information. When looking at anonymization during recommendation, Ciss´ee and Albayrak [11] utilized trusted agents (essentially moving the trust around) to act as a relay and filter the information that is sent. This way, the user can interact (through the agent) with the recommender system in an anonymous way. The user hides his personal information from the service provider and is safe from the service provider linking his rating information to a person. However, the user still needs to trust that the agents (either hardware or software based) and the service provider do not collude.

Similar to anonymization is randomization. In randomization (sometimes referred to as perturbation), the information fed into the system is altered to add a degree of uncertainty. Polat and Du [29] proposed a singular value decomposition predictor based on random perturbation of data. The user's data is perturbed by adding a random value (from a fixed distribution) to each of the ratings; unknown ratings are filled in with the mean rating. They go on to show the impact on privacy and accuracy and their inherent trade-off due to perturbation. In later work [30], their setting is different. A user wants two companies to collaboratively compute recommendations for him. This user acts as a relay for the two companies. The user's privacy is based on randomizing values. Berkovsky et al. [6] proposed to combine random perturbation with a peer-to-peer structure to create a form of dynamic random perturbation. For each request, the user can decide what data to reveal and how much protection is put on the data. Different perturbation strategies are compared based on accuracy and perceived privacy. Shokri et al. [35] added privacy by aggregating user information instead of perturbing. Aggregation occurs between users, without interaction with the recommender system. Thus, the recommender system cannot identify which information is part of the original user information and what is added by aggregation. A degree of uncertainty is added to the user's information similar to randomization. Recently the field of randomization is shifting toward differential privacy [13], which aims to obscure the link between single users' information in the input (the user's information) and output (the recommendation). This is accomplished by making users in released data computationally indistinguishable from most of the other users in that dataset. This is typically accomplished by adding noise to the inputs or output, to hide small changes that arise from a single user's contribution. The required level of noise depends on how and how often the data will be used and typically involves a balancing act between accuracy of the output and privacy of the input. Such indistinguishability also applies strongly to collaborative recommender systems, where a user should be unable to identify individual peers' ratings in the output he receives. As each recommendation leaks a little bit of information about the input (even with noise), with a larger number of recommendations, the added noise should be greater to provide the same level of privacy. McSherry and Mironov [26] proposed collaborative filtering algorithms in the differential privacy framework. Noise is added to the item covariance matrix (for item similarity). Since the item covariance matrix is smaller than the user covariance matrix, less noise needs to be added and more accuracy is preserved. The drawback of these techniques is that the security of these methods is hard to be *formally proven*, as is done in classical cryptography. The noise levels in differential privacy techniques must not overwhelm the initial output data and thus remove utility of the results completely. At the same time, enough noise must be added in order to hide the contribution of a user. When combined with multiple computational results and external information, even more noise is needed to protect the privacy of a user.

2.2.4 User control

In addressing privacy concern issues in recommender systems, much attention has been put on creating solutions, such as granting users control over information release [31] or providing disclosure justifications [27]. In principle, control enables users to better manage their information flow and make decisions on information sharing, so as to reduce concerns about privacy.

2.3 Risk Aware Recommender Systems

This requirement expresses the ability to be aware of the risk level of the user's situation during the recommendation process in order to not recommend documents in risky situations for example. Three techniques are pro- posed to compute the risk. The "variance of the cost" approaches ([80, 1]) have the advantage to be not user dependent but they still have the deal with the cold start problem. The expected environment cost" approaches [60, 122, 55] have the disadvantage to depend on manually tuning techniques. The hybrid approaches ([41]) have been developed by combining the two latest techniques so that, the inability of the "expected environment cost approach" to detect new dangerous states (cold start) is reduced by "the variance of the cost approach" [41], but they suffer from the lack of semantic.

Chapter 3

Related Work

3.1 Recommender Systems

In this section we will discuss the related work in the field of recommender system.

3.2 Modelling Recommender system

In [1] a description of an ontology-driven domain model for usage mining in the context of agent-based Web recommender systems is provided. It first starts with a description of MADEM (Multi-Agent Domain Engineering Methodology) as a software development methodology for multi-agent domain engineering followed by the description of the modeling concepts, tasks and products for the development of a family of multi-agent systems in a problem domain.

3.3 Risk Aware Recommender Systems

After gaining understanding the concept of multi-agent system in context of recommender systems we now discuss the Dynamic risk aware recommender system, as described in [7].Dynamic Risk Aware Recommender System (DRARS) is utilized in order to model the user and the situation by modelling the user with two facets: the profile and the context and then proceeds to define and model the situation as an instantiation of the user context. The user's profile is structured as multidimensional features, and the user's context is modelled with ontologies. Each situation is linked to a user's interest and stored in a case base. Following the dynamicity of the user's content the paper then proceeds with an algorithm that is called R-UCB, which is a combination of CBF and the Upper Confidence Bound (UCB) algorithm to follow the dynamicity of the user's content. The UCB algorithm constructs a reward estimate for each document previously seen. The reward is computed as the mean of the observed number of clicks added to an additional term that is inversely related to the number of times the document has been recommended. The document with the highest reward estimate is selected for recommendation. The paper suggests that the reward estimates in this way encourages exploration of documents that have been infrequently selected followed by the algorithm’s use of CBF to identify the similar resources to those selected by the UCB algorithm. This approach allows to follow the user’s content dynamicity by proposing documents which are sometimes the most probable to be clicked and, other times, documents randomly chosen to improve the knowledge of the system. Considering the situation risk level and intrusiveness of information delivery, the paper then considers the situation risk level when managing the exploration-exploitation trade-off in the recommender system. This strategy has been shown achieves high exploration when the current user’s situation is not risky and achieves high exploitation in the inverse case. The paper then conclude with the discussion of aggregated three approaches for computing the risk. The first discussed approach computes the risk using concepts from the application domain, permitting to get the risk directly from the risk of each of those concepts. The second approach is shown to compute the risk using the similarity between the current situation and situations stored in the system, assuming that similar situations have the same risk level. The third approach discusses the computation of the risk using the variance of the reward, assuming that risky situations get very few user’s clicks. Finally, an evaluation of the proposed approaches is provided by using different models through off-line experiments, recording the user’s navigation activities in a first step, and test them in a second step using an iterative process.

3.4 Privacy Preserving Recommender Systems

Paper [5] presents a collaborative privacy framework for preserving users’ profile privacy in a social recommender service. It also provides an overview of EMCP components and the interaction sequence for a recommendations process in an IPTV content distribution scenario followed by the description of a novel two stage concealment process that offers to the users a complete privacy control over their ratings profiles. The concealment process utilizes hierarchical topology, where users will be organized in peer-groups, from which super-peers are elected based on their reputation. Super-peers aggregate the preferences obtained from underlying users and then encapsulate them in a group profile and then send them to PRS. This paper also provides a test of performance of the proposed framework on a real dataset and the evaluation of how the overall accuracy of the recommendations depends on a number of users and requests. The experimental and analysis results showed that privacy increases under proposed middleware without hampering the accuracy of recommendations. Moreover the approach used in the paper has been shown to reduces privacy breaches on the concealed data without severely affecting the accuracy of recommendations based on collaborative filtering techniques by realizing that there are many challenges in building a collaborative privacy framework for preserving privacy in social recommender service. Paper [6] provides an evidence that the disclosure of user preferences in a recommender system seriously threatens users' personal privacy especially when service providers move their user data to an untrusted cloud. In this paper, a novel solution, called APPLET is presented, to address the significant challenges in privacy-preserving locationaware recommender systems. For APPLET, an introduction of multiple cryptography methodologies for protecting the privacy of the recommender system users without affecting the recommendation quality. Moreover, an evaluation has been provided that the effectiveness and performance of APPLET turns out to be well suited. In [12], the author(s) proposed a novel method for privacy preservation in collaborative filtering recommendation systems. The author(s) addressed the problem of protecting the users' privacy in the presence of an untrusted central server, where the server has access to users' profiles. To avoid privacy violation, a mechanism is proposed where users store locally an offline profile on their own side, hidden from the server, and an online profile on the server from which the server generates the recommendations. The online profiles of different users are frequently synchronized with their offline versions in an independent and distributed way. Using a graph theoretic approach, the author(s) developed a model where each user arbitrarily contacts other users over time, and modifies his own offline profile through a process known as aggregation. To evaluate the privacy of the system, this approach is then applied to the Netflix prize data set to investigate the privacy-accuracy tradeoff for different aggregation types. Through experiments, it is concluded in the paper that such a mechanism can lead to a high level of privacy through a proper choice of aggregation functions, while having a marginal negative effect on the accuracy of the recommendation system. The results illustrated that similarity-based aggregation functions, where users receive items from other users proportional to the similarity between them, yield a considerable privacy level at a very low accuracy loss. The findings in [14] suggest that users' online information is multi-dimensional regarding privacy concerns, especially in a recommender context. Although this seems self-explanatory, it is often neglected in privacy research and recommender system design. Specifically, demographic information that is frequently required for online service registration can be divided into two categories i.e. unidentifiable information and identifiable information. Unidentifiable information consists of items describing one's personal attributes that cannot be used to uniquely pinpoint the individual, whereas identifiable information is more accurate in pointing to the individual's identity exclusively. It has been argued that the people are significantly more concerned about the recommender system accessing their identifiable information than their unidentifiable information. In a similar manner, product items can be broadly grouped into non-sensitive types and sensitive types. It is also shown that the users are significantly more worried about their previous purchases of sensitive products being accessed for personalized recommendations than they are about their previous purchases of non-sensitive products (e.g., jewelry and shoes). These item-based analyses and categorizations provide a relative information-ranking system in terms of privacy concern in recommender systems, thus refining existing research on general privacy concern about user information. The categorization provided in the paper is extended prior research by extracting new factors, which can be used as a reference in future studies and designs. These new factors are shown to suggest that recommender system designers should treat users' information discriminatively and strategically based on their levels of sensitivity for pattern prediction and personalized recommendations. It is also concluded that the algorithm developers should be well aware of what information users are more hesitant to disclose, so as to adjust the degree of information tracking and use, as well as to provide appropriate coping strategies. In line with the "privacy-personalization trade-off," unsolicited access to users' sensitive information may trigger severe privacy concerns that could affect users' overall experiences therefore, identifiable and sensitive data should be more cautiously handled in exchange for prediction accuracy. As a design suggestion, recommender systems should introduce user control or privacy assurance mechanisms to help alleviate users' privacy concerns. Also, user data with different sensitivity levels (e.g., identifiable vs. unidentifiable information) can be potentially protected with different levels of privacy remedies.

3.5 Privacy Preserving Methodologies for Recommender Systems

Chapter 4

Proposed Approach

In this chapter we will discuss a multi-agent system model of a web recommender system. Before going further in the description of the system model, it is indispensable to describe the notations used in this approach. Various types of UML diagrams are used to provide the system description. These diagrams are explained in the following section.

4.1 UML Diagrams

UML stands for Unified Modeling Language and is used in object oriented software engineering. Although typically used in software engineering, it is a rich language that can be used to model an application structures, behavior and even business processes. There are 14 UML diagram types but for the purpose of this thesis, we will be focusing only on the Activity Diagram and the Sequence Diagram.

4.1.1 Activity Diagram

The basic purposes of activity diagrams are similar to captures the dynamic behavior of the system by showing the message flow from one activity to another. Activity is a particular operation of the system. Activity diagrams are not only used for visualizing dynamic nature of a system but they are also used to construct the executable system by using forward and reverse engineering techniques. The only missing thing in activity diagram is the message part. It does not show any message flow from one activity to another. Activity diagram is some time considered as the flow chart. Although the diagrams looks like a flow chart but it is not. It shows different flow like parallel, branched, concurrent and single.

4.1.2 Sequence Diagrams

UML sequence diagrams are used to represent or model the flow of messages, events and actions between the objects or components of a system. Time is represented in the vertical direction showing the sequence of interactions of the header elements, which are displayed horizontally at the top of the diagram. Sequence Diagrams are used primarily to design, document and validate the architecture, interfaces and logic of the system by describing the sequence of actions that need to be performed to complete a task or scenario. UML sequence diagrams are useful design tools because they provide a dynamic view of the system behavior which can be difficult to extract from static diagrams or specifications. Although UML sequence diagrams are typically used to describe object-oriented software systems, they are also extremely useful as system engineering tools to design system architectures, in business process engineering as process flow diagrams, as message sequence charts and for protocol stack design and analysis.

4.2 The Approach

In this approach we will start with breaking-down the system into subsystems. Each subsystem will be responsible for accomplishing a pre-defined task. Each sub system will also consist of an agent. For the specification of the problem domain to be solved, we will focuses on modeling goals of the subsystem, roles of the agents, activities performed by the agents and finally the interactions of the agents. Agents possesses knowledge and that knowledge is used by it to exhibit autonomous behavior. A subsystem is composed of agents having specific goals that establish what the subsystem intends to accomplish. The achievement of specific goals by the agents within a subsystem, allows reaching the general goal of the entire system when put together.

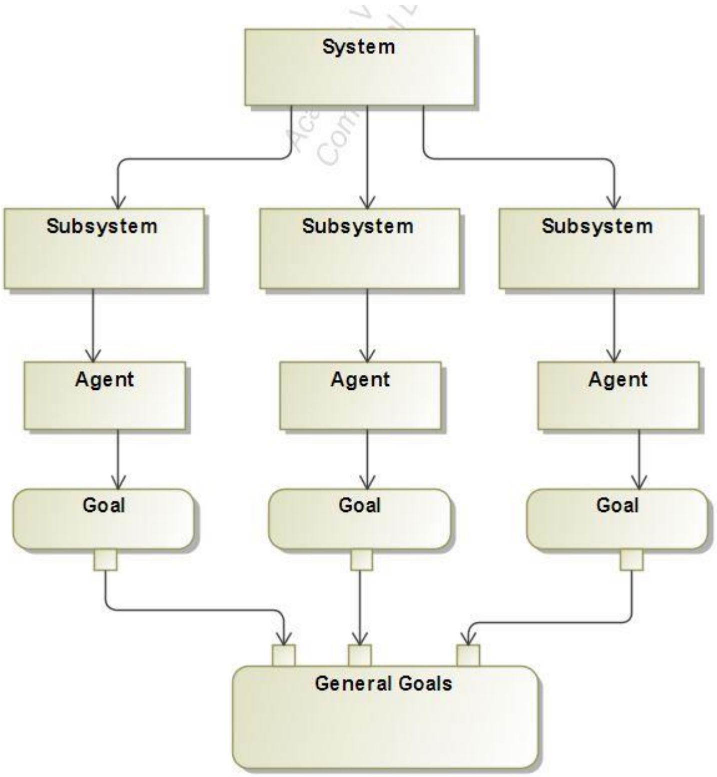


Figure 1 Description of system constituents

Specific goals by the agent within a subsystem are reached through the performance of responsibilities that agents have, by playing roles with a certain degree of autonomy. Responsibilities are exercised through the execution of activities by each individual agent within the subsystem. The set of activities associated with a responsibility are a functional decomposition of it. Roles have skills on one or a set of techniques that support the execution of responsibilities and activities in an effective way within the subsystem. Preconditions and post-conditions may need to be satisfied for/after the execution of an activity by each agent within the subsystem. Knowledge can be consumed and produced through the execution of an activity. Skills can be, for instance, the rules of the subsystem that agents know to access and structure its information sources. Sometimes, agents have to communicate with other agents to cooperate in the execution of an activity. This approach allows for such communication to take place between the agents within the subsystems.

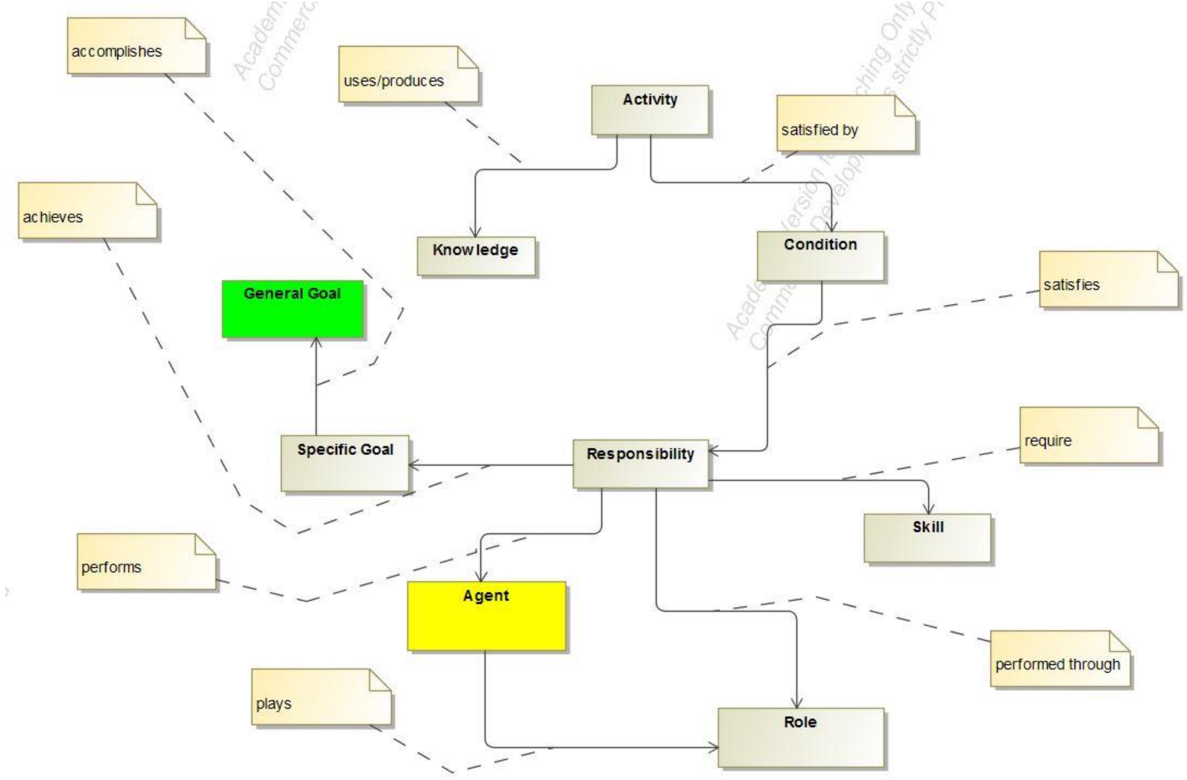


Figure 2 Relationship model for subsystems

4.3 Subsystems

We will now list out the subsystems which makes up a risk aware and privacy preserving web recommender system and also explain the contribution of each sub systems and the agents involved in the respective subsystems.

4.3.1 Data Subsystem

Let us first start with the data subsystem manages. This subsystem is responsible for managing the data inflow and outflow from the recommender system. This subsystem absorbs data in form of User Preferences and User feedback. It has multiple elements which performs the task that brings out the functioning of the data subsystems. The below diagram is the representation of the data subsystem. This subsystem consists of two agent which are the Data Manager Agent and the Aggregator Agent.

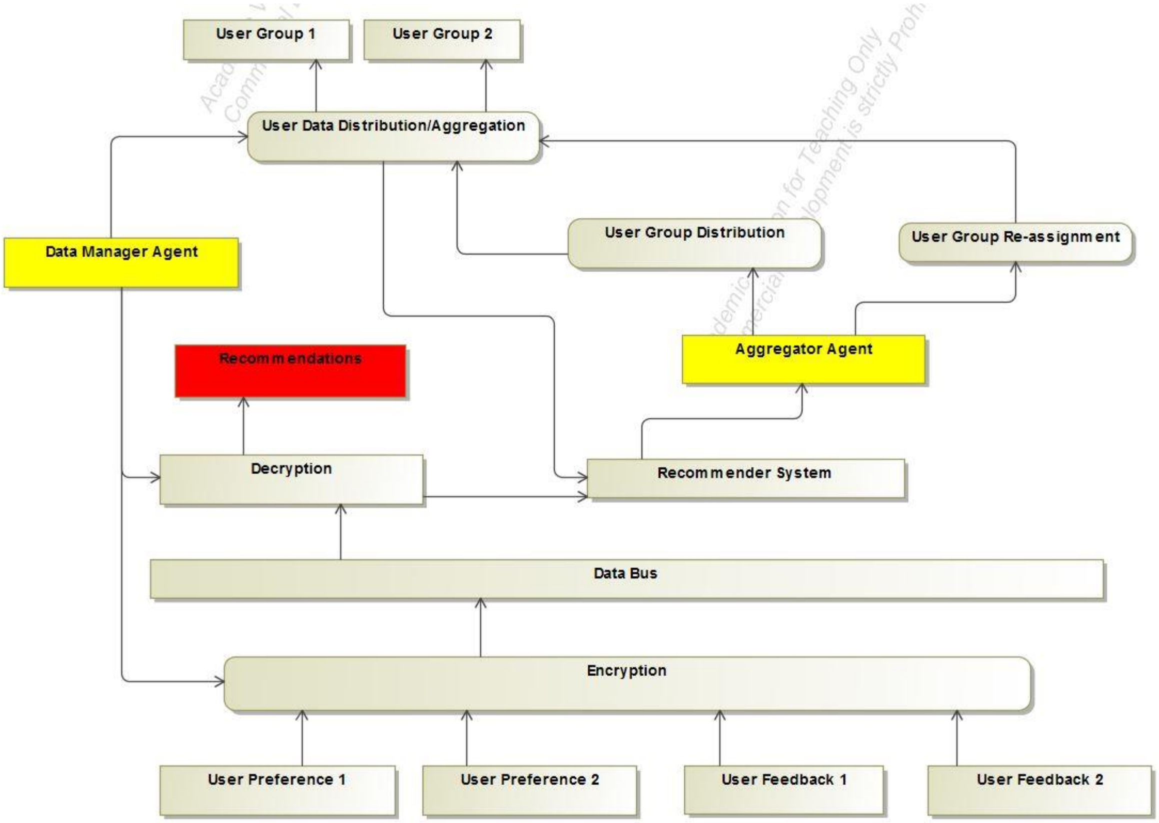


Figure 3 Activity Diagram of Data Subsystem

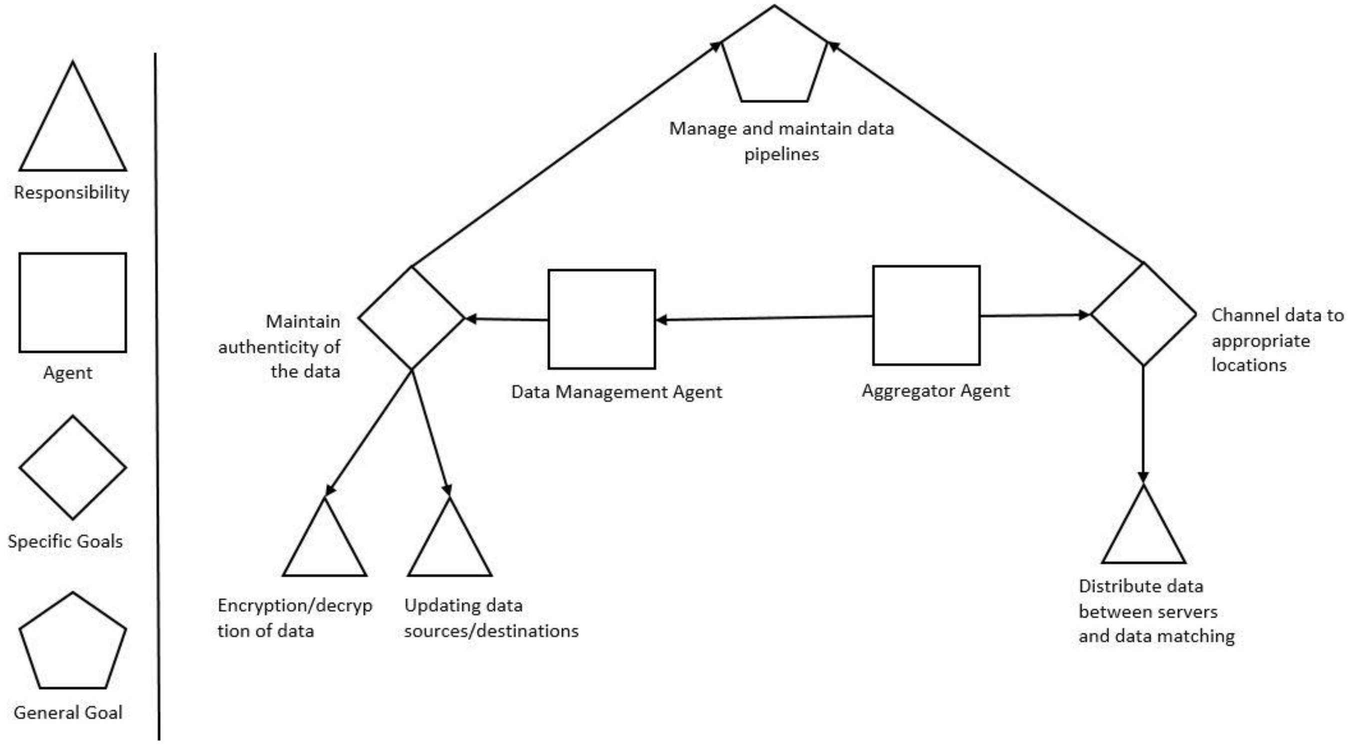

Responsibility
Agent
Specific Goals
General Goal
Manage and maintain data pipelines
Maintain authenticity of the data
Data Management Agent
Aggregator Agent
Channel data to appropriate locations
Encryption/decryp tion of data
Updating data sources/destinations
Distribute data between servers and data matching

Figure 4 Goal model for the data agent and the aggregator agent

The goal of the data manager agent is to maintain the authenticity of the data by preventing it from getting corrupted and also to manage the piping of data from source to the desired destination. This goal for the data agent is achieved by fulfilling two responsibilities i.e. the responsibility of properly encrypting and decrypting the data from the source and the destination respectively and by updating the proper locations of source and destination for the data to be used by the system.

The main task of the Aggregator agent is to channel between the user interface and the various servers for computation, storage and generating recommendations. This specific goal is achieved by the proper distribution and redistribution of data within the system. The best example of an aggregator agent is a messaging brokers used in modern applications. Apache Kafa and RabbitMQ are two such message brokers. Together these two agents actives the objective of the data subsystem i.e. management and maintenance of the data pipelines within the system to enable the system.

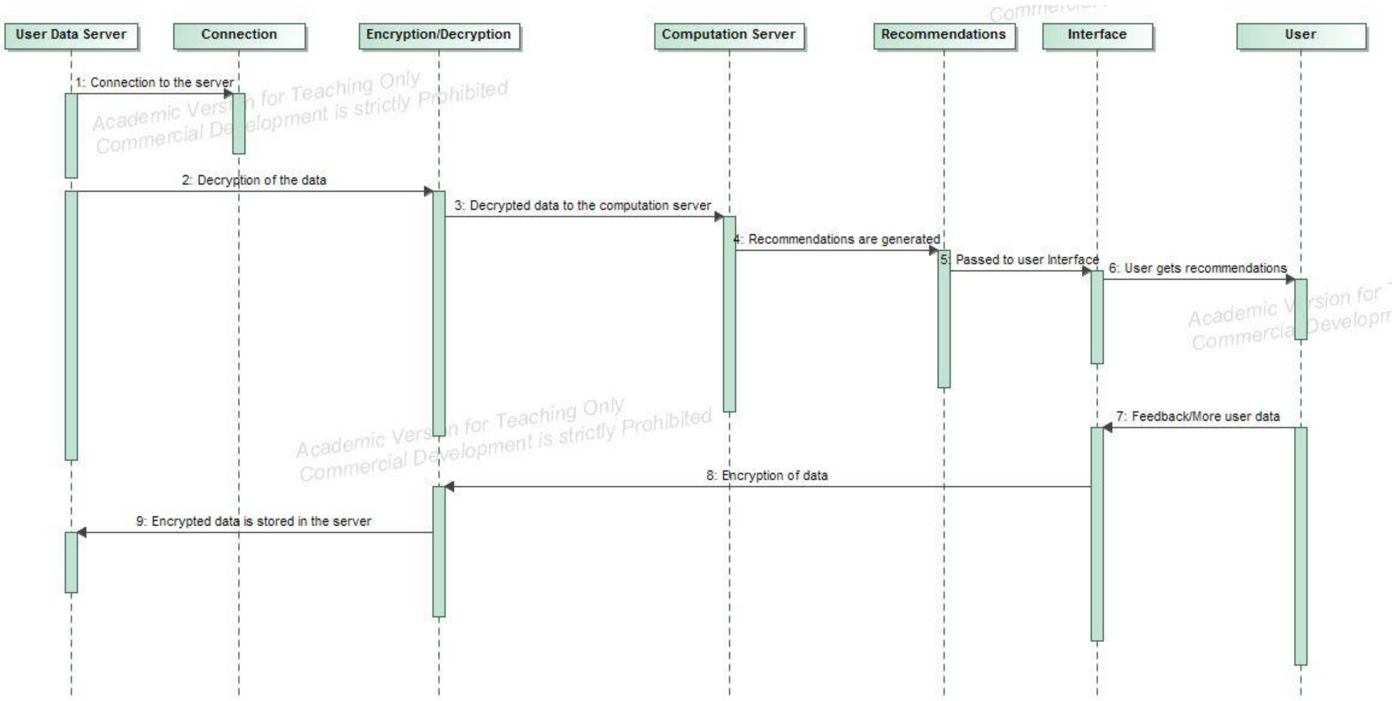

User Data Server
Connection
Encryption/Decryption
Computation Server
Recommendations
Interface
User
1: Connection to the server
2: Decryption of the data
3: Decrypted data to the computation server
4: Recommendations are generated
5: Passed to user Interface
6: User gets recommendations
7: Feedback/More user data
8: Encryption of data
9: Encrypted data is stored in the server

Figure 5 Data Subsystem sequence diagram

The sequence diagram of the data subsystem has been provided. In this diagram, a recommendation generation process starts when a connection is established between the user-data database and the computation server where the data to be used is decrypted. This data is then piped to the computation server. After the processing at the communication server, the recommendations are generated and are then forwarded to the user through an interface. Based on the quality of recommendation, the user provides a feedback which is stored in the user-data database. The transfer of data between the servers including the encryption and the decryption process is carried out within the data subsystem. These tasks can be assumed to be carried out by the data agent and the aggregator agent within the data subsystem, the outline of which has been provided in the previous section.

4.3.2 Privacy Subsystems

The Privacy subsystem manages the privacy aspect of the web recommender system. Within this subsystem the contextual and personal information is extracted from the user and fed into the recommender system. An addition differential privacy server is used to handle the differential privacy aspect of the subsystem. This sub system consist of the User Privacy Agent to carry out its operations.

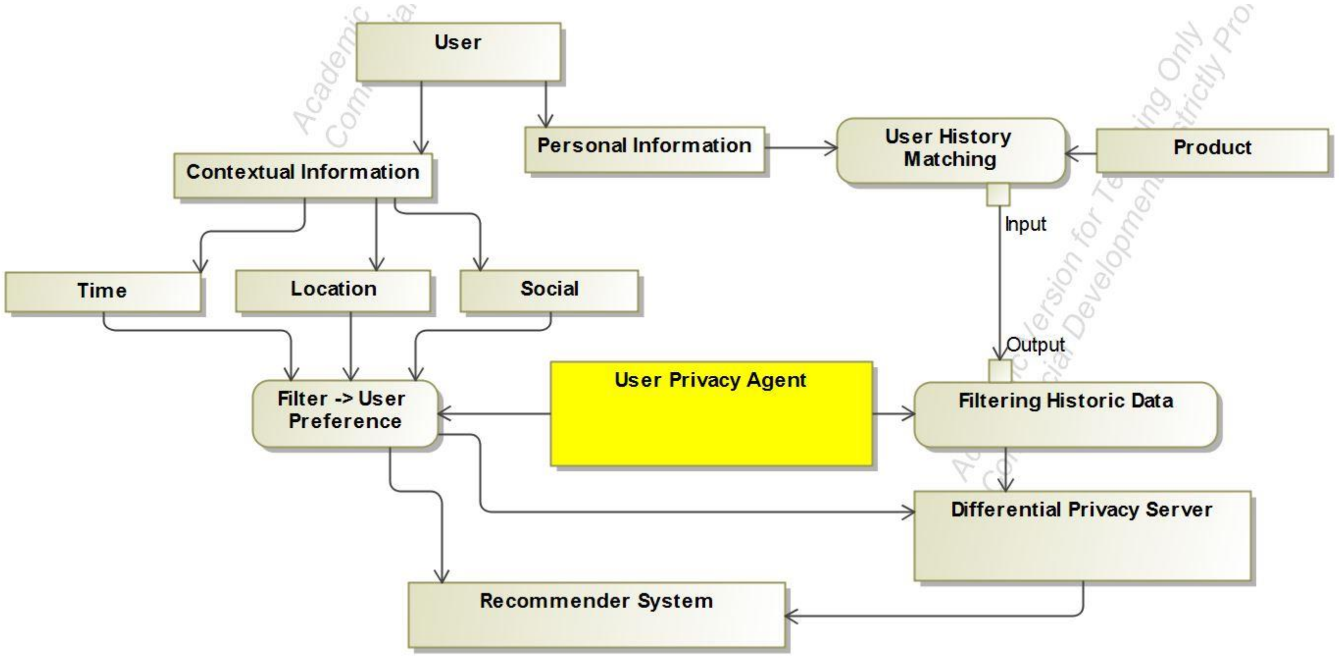


Figure 6 Activity diagram for the Privacy Agent

The main role of this subsystem is to provide the user’s contextual data and the historic data of the user to the computation server in order to generate recommendations for the users. The contextual information from the user can be in from of location, social information of the user, combined with the timing of the information. The user history data refers to the user’s behavior while using the system that is being recorded for analysis. The contextual data from the user along with the historic data of the user provides valuable insights in order to provide quality recommendations to the user.

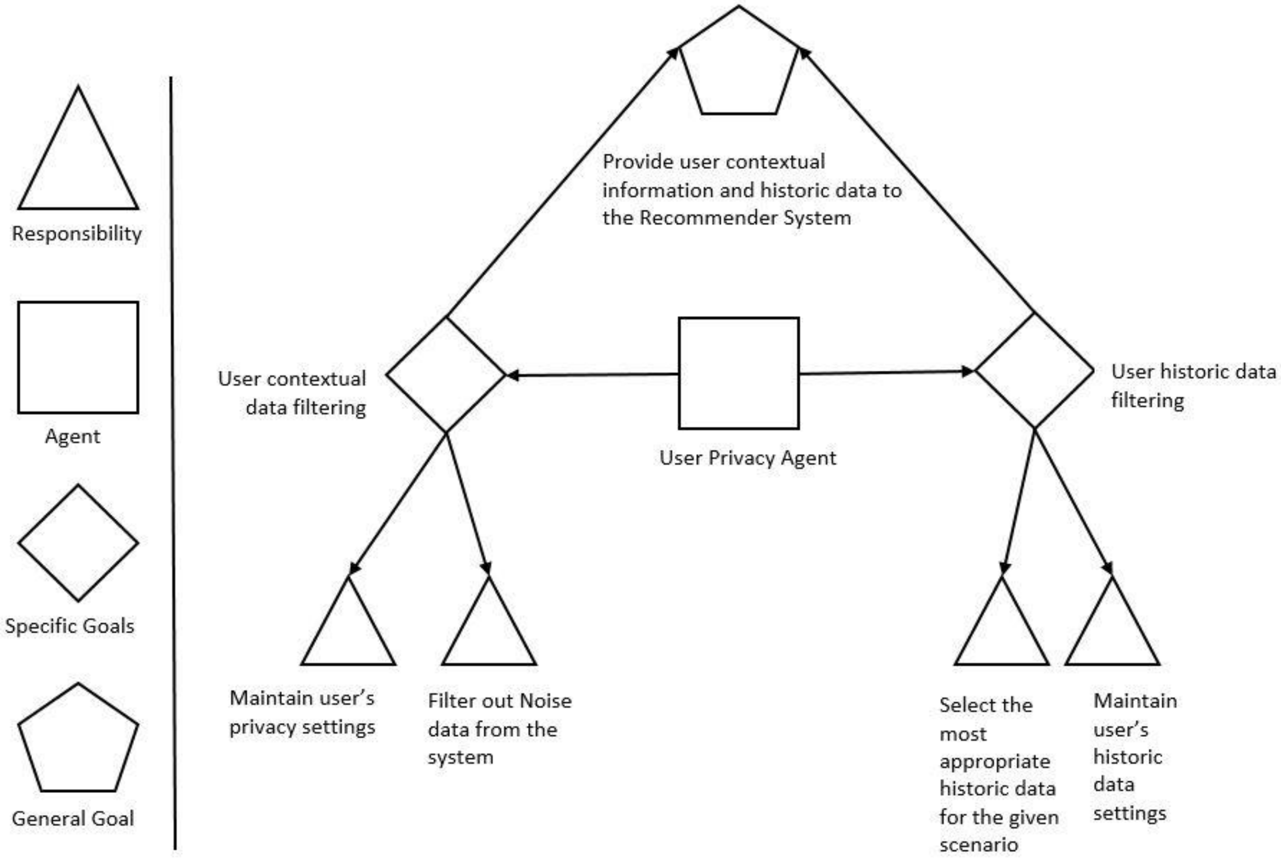

Responsibility
Agent
Specific Goals
General Goal
Provide user contextual information and historic data to the Recommender System
User contextual data filtering
User Privacy Agent
User historic data filtering
Maintain user's privacy settings
Filter out Noise data from the system
Select the most appropriate historic data for the given scenario
Maintain user's historic data settings

Figure 7 Goal model for the user privacy agent

To understand the role of the privacy subsystem within the recommender system mode, we need to look at the goals of the user privacy agent. The user privacy agent performs the task of maintaining user's privacy settings for the contextual data and also the responsibility of filtering out the noise from the contextual data being obtained from the user. These two responsibilities form the specific goal of filtering and maintaining user's contextual privacy information. On the other hand, the user privacy agent also fulfills the responsibility of maintaining the access to user's historic data based on the settings provided by the user and selecting the most appropriate data for generating the recommendations by filtering out the noise from the historic data.

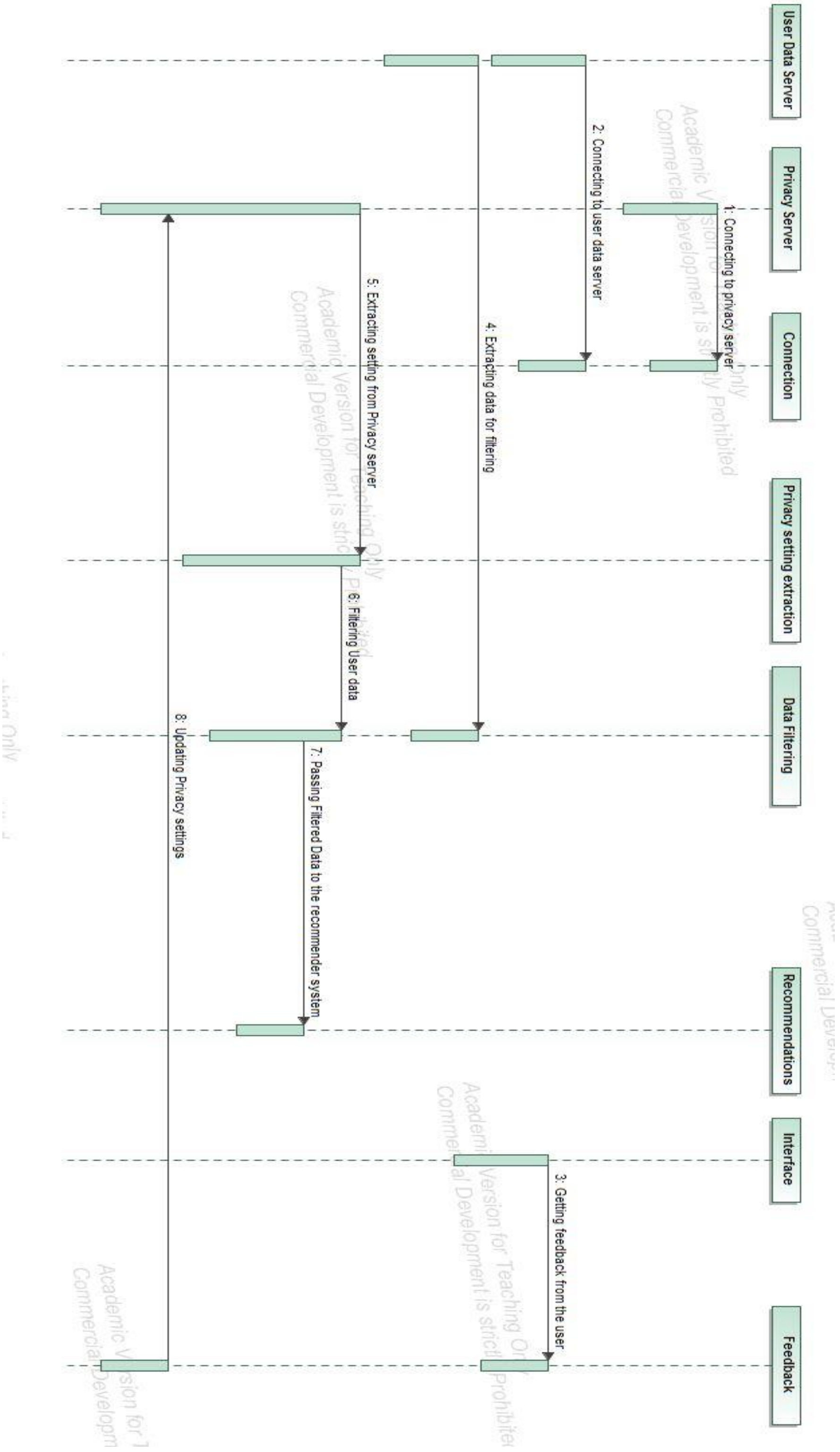

User Data Server
Privacy Server
Connection
Privacy setting extraction
Data Filtering
Recommendations
Interface
Feedback
1: Connecting to privacy server
2: Connecting to user data server
3: Getting feedback from the user
4: Extracting data for filtering
5: Extracting setting from Privacy server
6: Filtering User data
7: Passing Filtered Data to the recommender system
8: Updating Privacy settings

Figure 8 Privacy subsystem sequence diagram

In order to gain the understanding of the privacy subsystem is to get the knowledge of flow of control that happens within the subsystem. The first step involves establishing a connection with the user-data server and with the privacy server. This is followed extracting the user data and user privacy settings from the system. Once the data has been extracted from the server, it is filtered against the user settings. The user data includes the contextual data i.e. location, time and social data as well as the user's previous behavior pattern obtained while the user interacted in the system. The user is made aware of the data through the user controls and asking permission from the user to utilize the data for generating the recommendations.

Once the data has been filtered of the noise and against the user settings, it is piped through the computation server to generate the recommendations to the user. Once the recommendations has been generated, they are forwarded to the user via interface.

Based on the quality of the recommendations, the user provides a feedback or exhibits certain behavior pattern (clicks, navigation, dismiss) which indicates the user's perspective on the quality of the generated recommendations. This feedback data is then encrypted and stored in the user-data database to serve as an input future for the future computations for generating recommendations.

4.3.3 Contextual Risk Subsystem

This sub system handles the contextual risk by getting the contextual information i.e. time, location and social information from the user and then feeding this information to the recommender system. It consists of two agents the Context Analyzer Agent and the User Risk Agent.

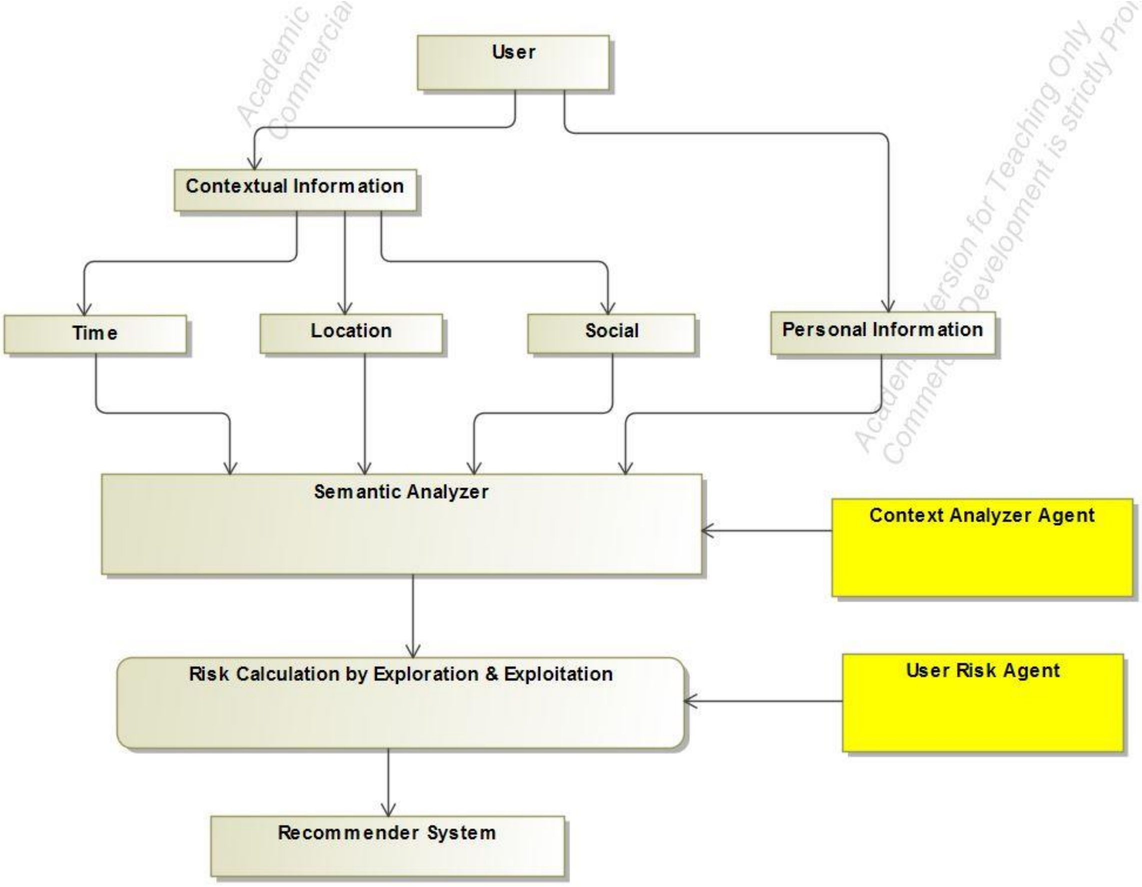


Figure 9 Activity Diagram for the Risk Subsystem

The information processed in this step is utilized by the recommender system to generate more contextually aware system by not only providing more relevant information to its users but also keeping itself aware of the risk associated with disturbing or negatively affecting the user with the bad recommendation. This tradeoff of providing relevant recommendations and the associated risk is the part of risk calculation through the exploration and exploitation problem.

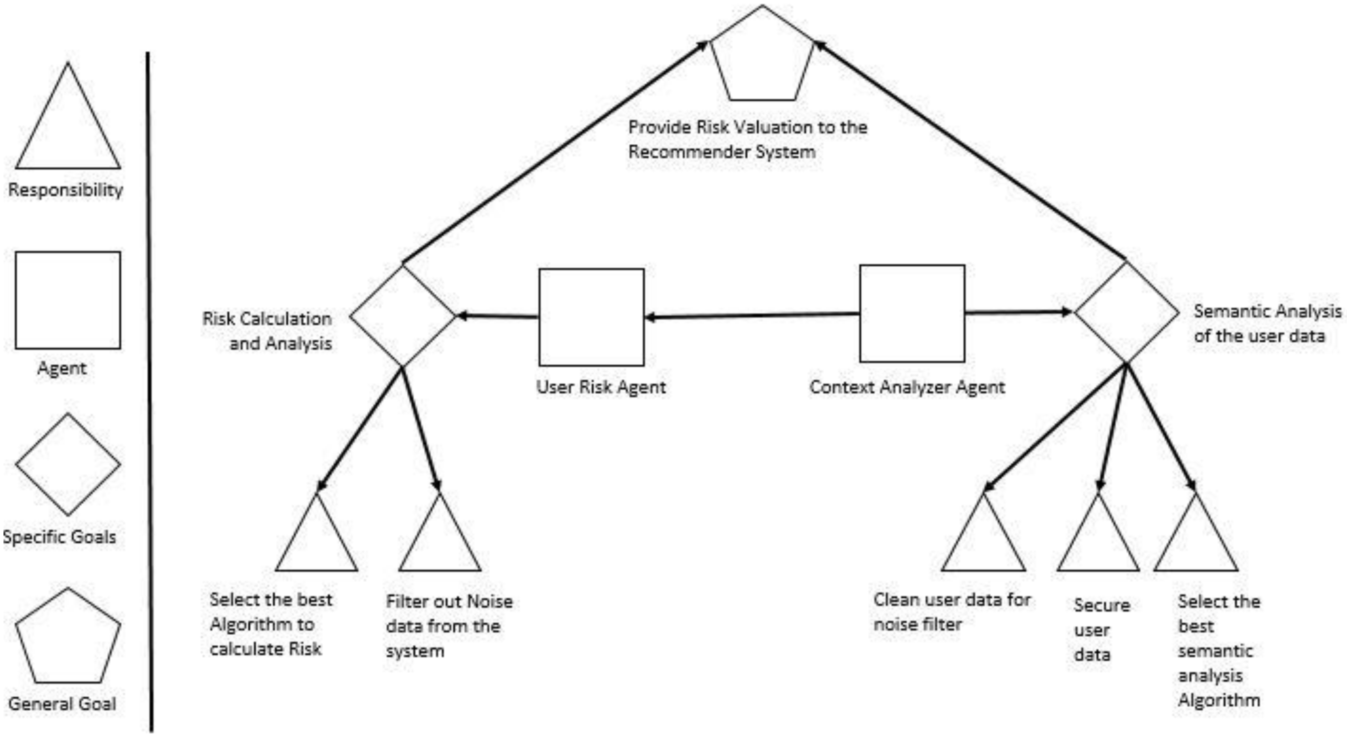

Responsibility
Agent
Specific Goals
General Goal
Provide Risk Valuation to the Recommender System
Risk Calculation and Analysis
User Risk Agent
Context Analyzer Agent
Semantic Analysis of the user data
Select the best Algorithm to calculate Risk
Filter out Noise data from the system
Clean user data for noise filter
Secure user data
Select the best semantic analysis Algorithm

Figure 10 Goal model for the privacy agent

The two agents have some specific goals and responsibilities. The responsibility of the user risk agent is to ensure that no noise remains in the data and to calculate the risk tradeoff for generating the recommendations and relevance of those recommendations to the user from user feedback for the previously generated recommendations. These two responsibilities helps in achieving the goal of carrying out risk calculation and analysis of the user data. The context analyzer agent is responsible for cleaning the data obtained from the risk calculation stage, selecting the best possible algorithm for the analysis and then by securing the generated data to be forwarded as recommendations to the user. This helps in achieving the task of semantic analysis of the user data and finally providing the analysis results as recommendations to the user of the system.

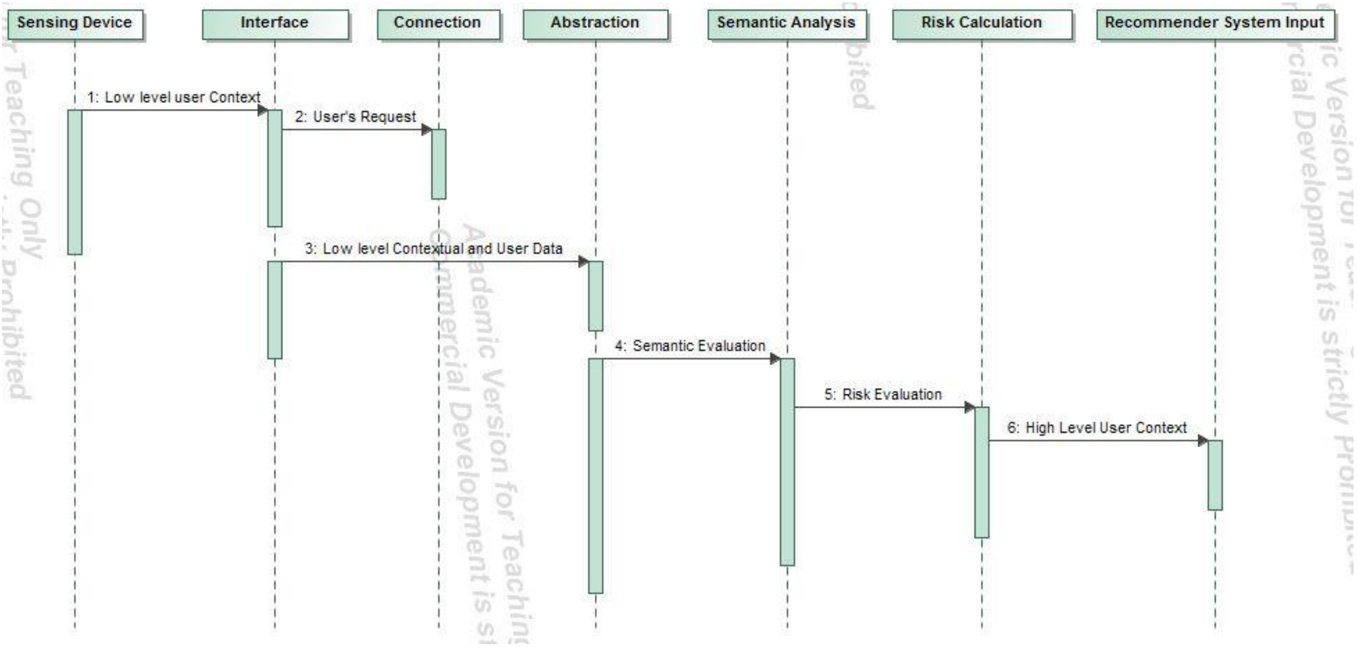


Figure 11 Contextual Risk Subsystem Sequence Diagram

The sequence diagram helps in understanding the steps that take place within the contextual risk subsystem. First, a connection is established with a sensing device at the user's end, through an interface. This step is followed by the low level abstraction of the user's data and feeding it to the servers running the semantic analysis. As a result of this, the risk is calculated and based on the value of this parameter the recommendations are forwarded to the user.

4.3.4 Activity model of the system

The domain model of the web recommender system consists of the aggregation of the individual subsystems and the coherence of the agents working within each working subsystem to achieve the goals of the entire system. The advantage of breaking down the web recommender system is to provide error detection and fault tolerance within the system. It also facilitates the understanding of the system in a clear sense. This model could be a better was of estimation of the value provided by the recommender system than the traditional way in the sense that it provides the domain experts with a better evaluation criteria. In the next section we will be discussing the evaluation criteria to be used for such recommender systems.

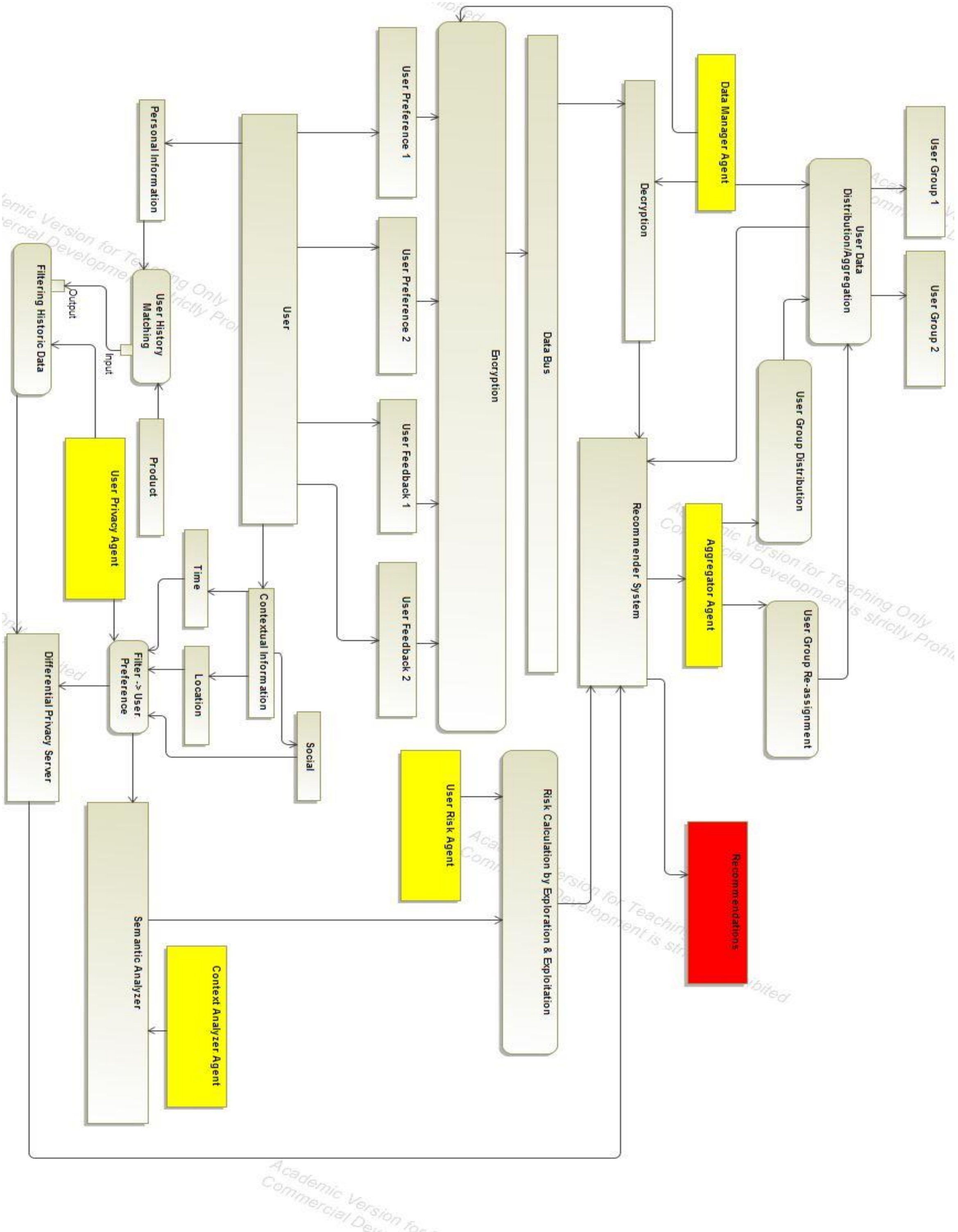


Figure 12 Complete Activity model of the system

4.4 System State Evaluation

We will now discuss the system evaluation method to determine the state of a web recommender system in terms of the degree of privacy protection and risk awareness it exhibits.

4.4.1 Privacy scope of a system

We introduce a coordinate system to describe the state of a web recommender system in terms of the privacy it offers to the user. It is a 3 dimensional representation with each of the mutually independent axis representing the state of the recommender system. On one of the axis we have a feature which states the size of the audience to which recommendations will be disclosed using data of a participant in the system i.e. if a user allows the system to use his/her data, then how many people other than the user, will be able to receive the recommendation based on that user's data in a collaborative environment. The extent of usage axis refers to the amount of information that is extracted from each participant in the system. The third and the final axis represents the duration for which the data remains in the system.

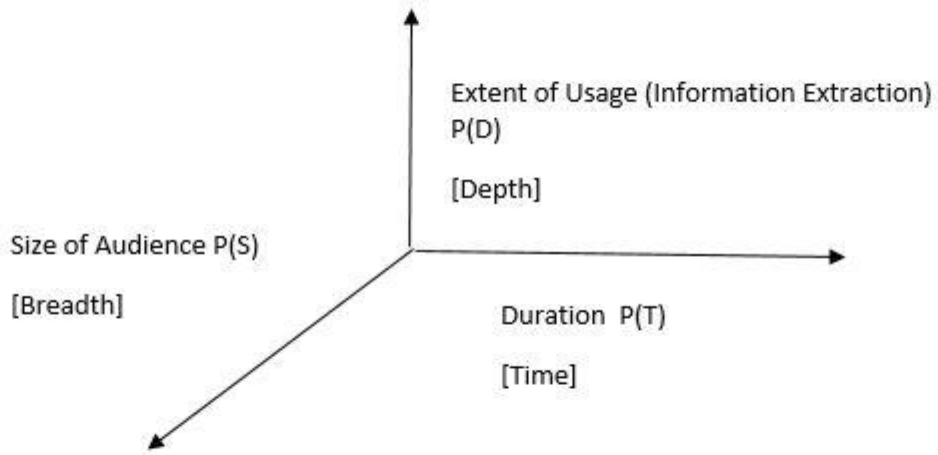


Figure 13 Privacy Scope

4.4.2 Contextual risk scope

This section describe the contextual risk scope of a recommender system. Similar to the description of the privacy scope, the contextual scope is also a three dimensional representation for the purpose of characterizing recommender systems. The three axis of the contextual risk scope are mutually independent. The first axis is the similarity axis denoted of the R(s) notation. It is defined as the extent of similarity between the user and the user group into which the user is placed. The second axis denoted by R(C) is the axis of intention and is described as the extent of awareness of the user's intention by the system. This axis is conceptual i.e. the valuation provided by the recommender system based on this metric is highly based on experimentation results. The third and the last axis is the axis of duration and is the measure of how long the contextual data will be stored by the system.

This axis is represented by the notation R(T) and may also represent the period of data used by the recommender system for the purpose of generating recommendations.

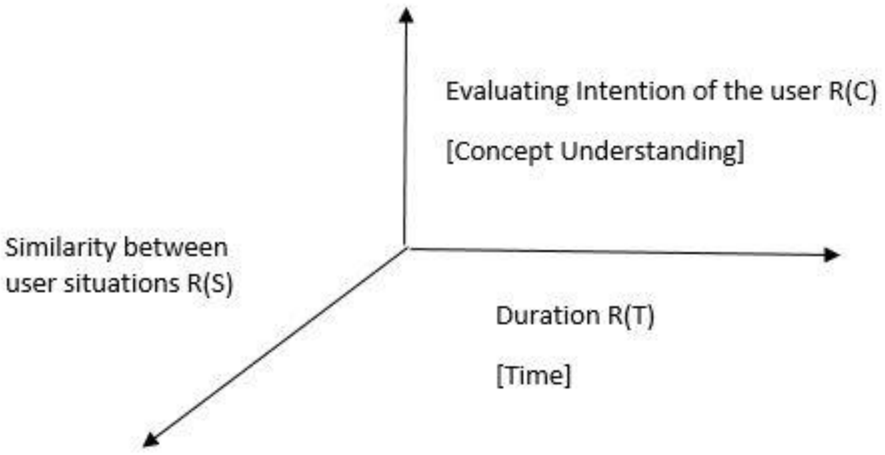


Figure 14 Contextual Risk Scope

4.4.3 Explanation of a multidimensional system diagram

We are now in position to describe a web recommender system in a five dimensional representation. Parallel coordinates is a visualization technique used to plot individual data elements across many dimensions. Each of the dimensions corresponds to a vertical axis and each data element is displayed as a series of connected points along the dimensions/axes. Thus, a recommender system can be described as a series of connected points along the diagram, intersecting at these axis/dimensions.

Since it is a start of a research for visualizing a recommender system by these axis, there is the dearth of data for exactly calculating the exact value of a particular recommendation. Hence, through this thesis, an approximate representation is used for visualizing a recommender system by using this method.

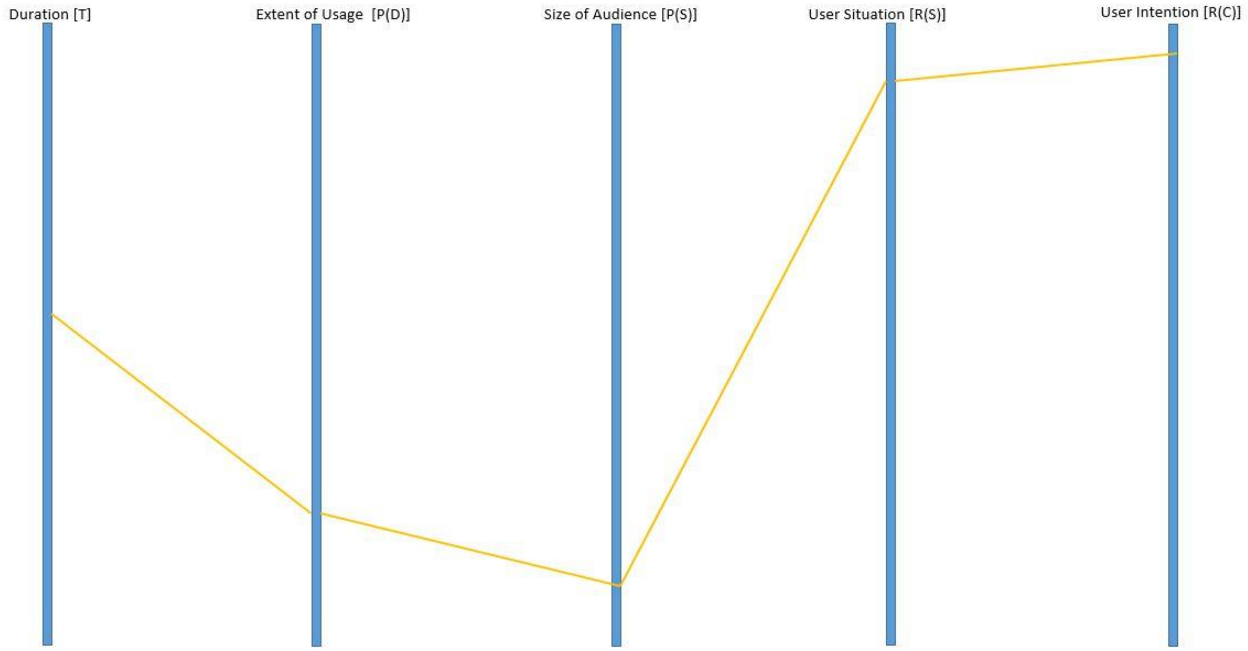
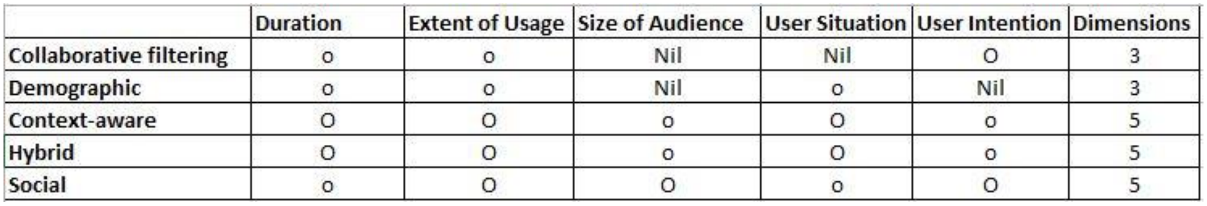


Figure 15 Dimensional Plot of a recommender System

Table 1 General Dimensional Analysis of various Algorithms



	Duration	Extent of Usage	Size of Audience	User Situation	User Intention	Dimensions
Collaborative filtering	o	o	Nil	Nil	O	3
Demographic	o	o	Nil	o	Nil	3
Context-aware	O	O	o	O	o	5
Hybrid	O	O	o	O	o	5
Social	o	O	O	o	O	5

In the above table an approximate idea has been provided about the possible dimensions that can be used by the recommender system using different methodologies, architecture and algorithms. The conclusion listed above is a result of review of current text in the area of web recommender systems. The extent of utilization/valuation of different metrics on the five dimensional recommendations is being distinguished in the table by using two different notations. The ‘O’ symbol is used where the utilization/valuation on a particular dimension is conceptually high and the ‘o’ notation is used for those system whose utilization/valuation on a particular dimension is relatively low.

Chapter 5

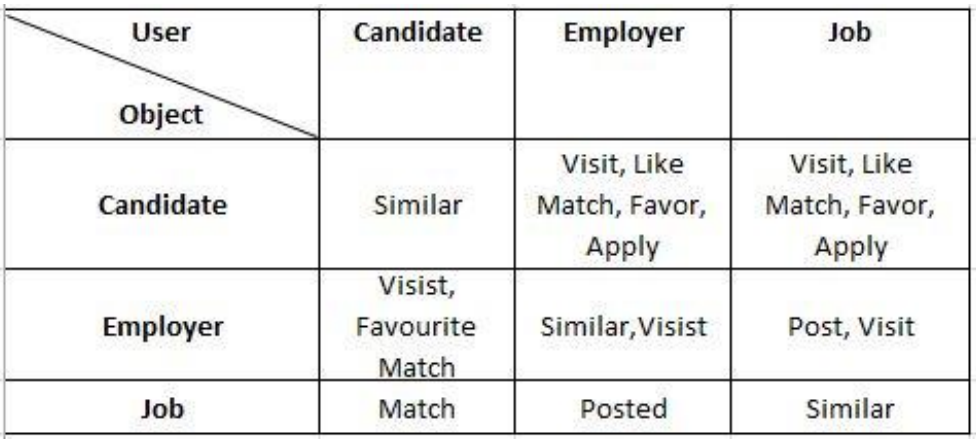
Case Study: Job Recommender System

Generally a recommendation system suggests personalized choices from a large set of possible options with the objective of reducing complex decision making. The last decade has witnessed the emergence of lots of job portals offering such services to their users. Generally a recommendation system works on information filtering technique and provides information which is of the interest of the concerned user. Typically, a recommendation engine, which employs a set of algorithms, compares the user's profile to some reference characteristics collected from the job description across multiple jobs posted on the job portal or the user's social environment, and seeks to predict a set of suitable jobs for the user.

5.1 Problem Description

We will now describe a recommender system proposed by [9] in 2013 and [10] in 2016. Paper [9] describes a hybrid recommender system for job seeking and recruiting websites. The described hybrid recommender system exploits the job and user profiles and the actions undertaken by users in order to generate personalized recommendations of candidates and jobs. The data collected from the website is modeled using a directed, weighted, and multi-relational graph, and the 3A ranking algorithm [16] is exploited to rank items according to their relevance to the target user. This paper also provides a preliminary evaluation based on simulated data and production data from a job hunting website in Switzerland. The approach in the paper consisting of modelling the entity-interaction based relations in the followed by the formation of a graph consisting of these entities and computation of ranking from this graph.

Table 2 Interaction Entities proposed in [9]



User Object	Candidate	Employer	Job
Candidate	Similar	Visit, Like Match, Favor, Apply	Visit, Like Match, Favor, Apply
Employer	Visist, Favourite Match	Similar,Visist	Post, Visit
Job	Match	Posted	Similar

The technique used in the paper involves interaction based relations. The first of these relations is the ‘POST’ relation which is described as a bidirectional relation between the employer and its jobs which comes into play while comparing two similar jobs posted by different employers. The next relationship that is described in the paper is the ‘APPLY’ which signals that a candidate is interested in the job. This signal leads the candidate to other jobs similar to the ones he/she applied for. The next relationship that is described in the paper is ‘FAVOURITE’, using which, a user can add an entity into his/her ‘favorite list’. This is also a strong and explicit signal of interest. Similar to the previous relationship, the ‘LIKE’ relationship with a difference that a user may not revisit the items they liked. In the paper, the ‘LIKE’ relationship is considered as an explicit feedback but weaker than ‘APPLY’ or ‘favorite’. The final relationship signal that is discussed in the paper is ‘VISIT’ which is an implicit feedback for user’s interest.

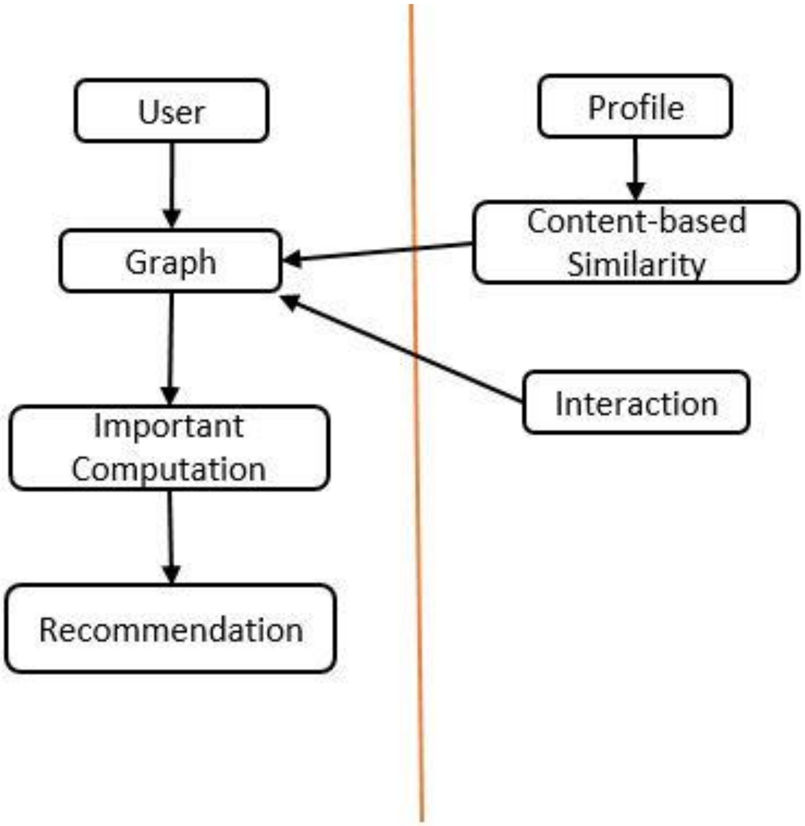
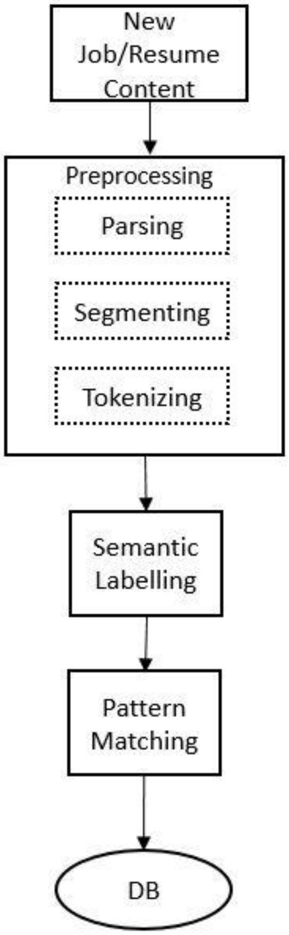


Figure 16 Graph Framework described in [9]

A pipelined hybrid recommendation approach is described and implemented in [9] along with providing the result of content-based similarity which is fed into a relation-based algorithm as an additional relation after normalization. The above figure shows the recommendation framework described in the paper for generating personalized job recommendations to the users.

On the other hand, paper [10] describes a resume matching system, “RésuMatcher”, which intelligently extracts the qualifications and experience of a job seeker directly from his/her résumé, and relevant information about the qualifications and experience requirements of job postings. Using a novel statistical similarity index, RésuMatcher returns results that are more relevant to the job seekers experience, academic, and technical qualifications, with minimal active user input. The two figures describe the RésuMatcher system in detail.

Figure 17 Resume matcher System as described in [10]



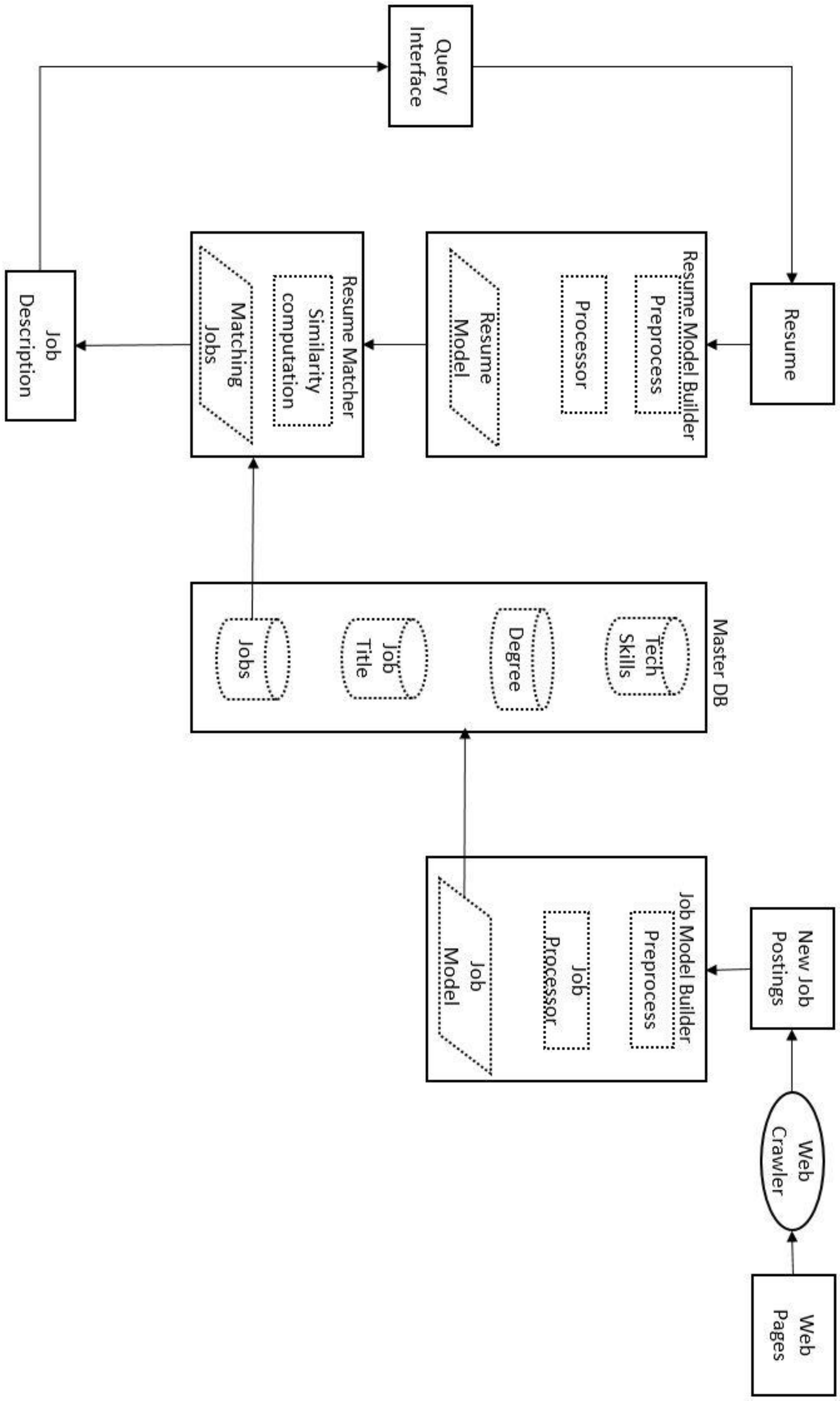


Figure 18 Information Processing Pipeline described in [10]

5.2 Approach

Representation of the two job recommender systems that are described in the previous section and combining them in a model of a job recommender system which can be efficiently described by the approach used in this thesis will be discussed in this section. The first step towards this description is to determine the features of the described recommender systems and then laying out those features in terms of the discussed approach. This involves breaking down the recommender system and focusing on the multi-agent aspect of the system and separating different components of the system into different subsystems and finding out a way to integrate the subsystems into one compact unit.

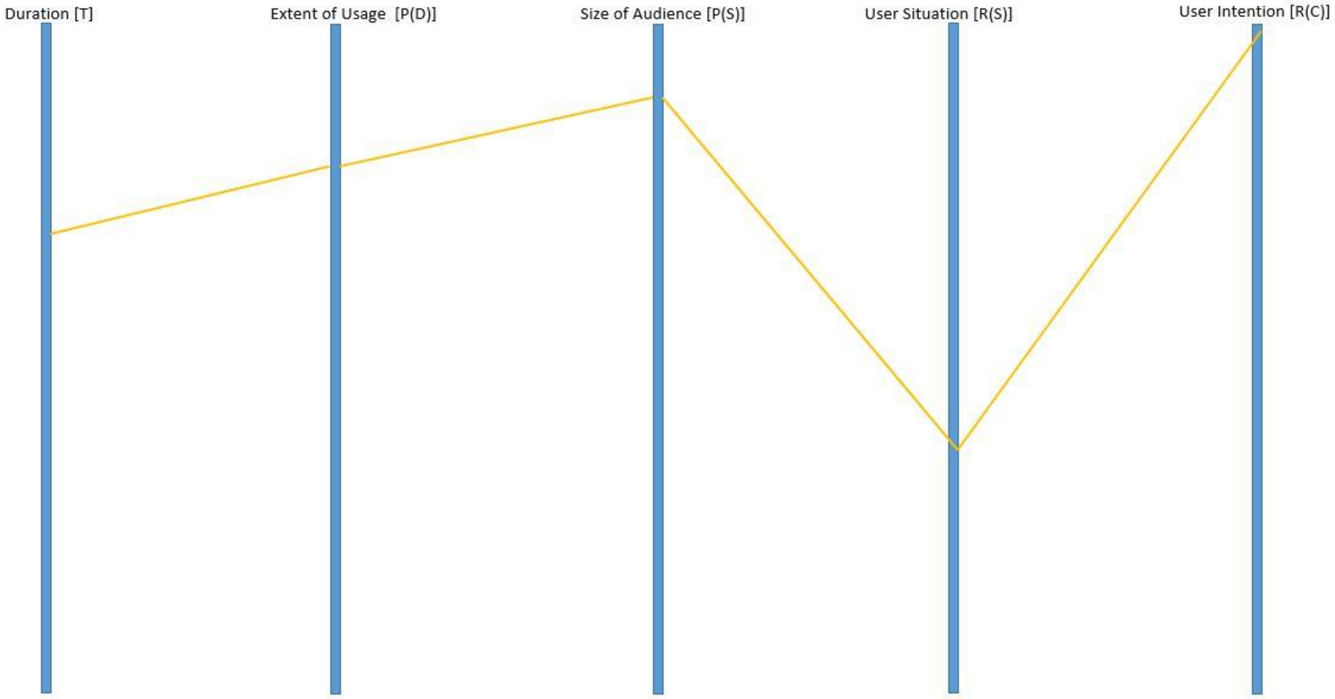


Figure 19 Multidimensional description of the Job Recommender system

The five dimensional representation of the recommender system, described in the previous section is now provided. The duration dimension is described as the period of time for which the job data and the resume were kept in the system and duration of chunk of historic data being used for generating recommendation. It is evident from the papers that this factor is on the higher side.

The next factor to consider is the extent of usage of user data by the recommender system. Since, large extent of user's personal data is available to the system in form of resume and user's actions (like, favorite, apply etc.) were being recorded by the system, the extent of data usage is supposed to be at high levels.

The size of audience in this scenario is also a big factor on the higher side. It can be considered to be higher than the valuation/utilization of the two previously discussed dimension because the data is available to many organizations and users that are accessing the system for their job search and getting recommendations from the system.

Since, most of the user data that is obtained, stored and utilized by the system is in static form involving personal information of both the job applicant and the employers, the value of user situation awareness by the recommender system is on the lower side.

Finally, the user intention factor of the system is at a high level in the graph because of the fact that the main objective of the system is to obtain meaningful job recommendation to the user and being aware of the user's intention to display better results.

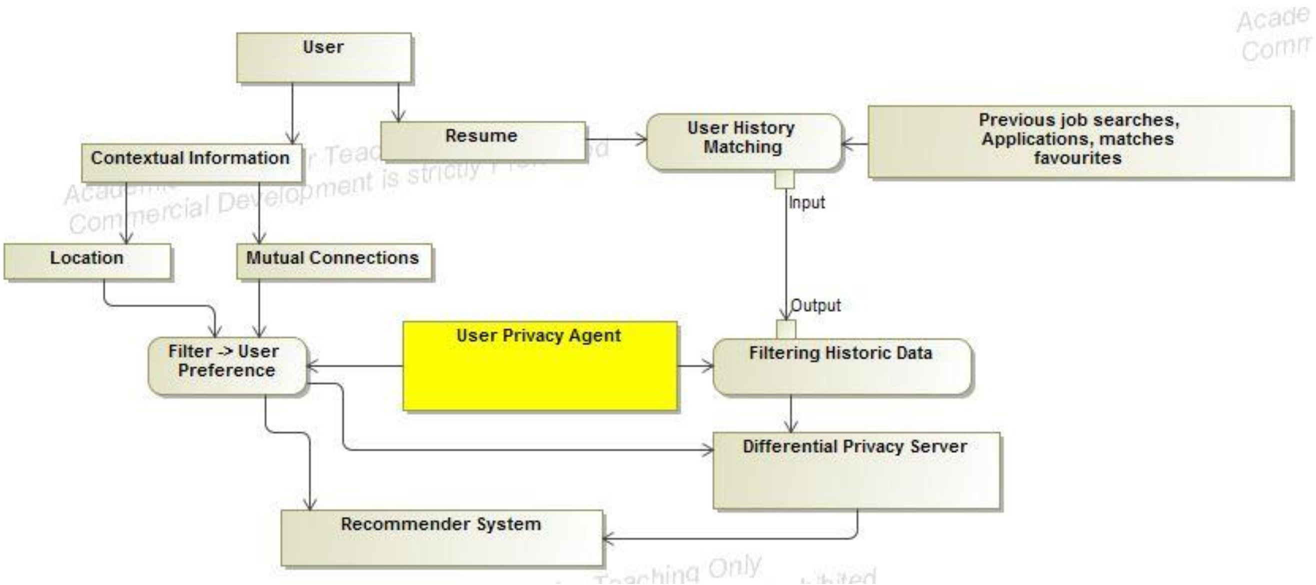


Figure 20 Privacy Agent for job recommender

The above diagram is the activity model of the privacy subsystem of the web recommender system. Within this subsystem the contextual i.e. time and location and the personal information in form of resume information is extracted from the user and fed into the recommender system. A differential privacy server manages the anonymity of data within this subsystems by implementing privacy differential algorithms. The main role of this subsystem is to provide the user's contextual data, personal information and the historic data i.e. favorites, visits and applications, of the user to the computation server in order to generate recommendations for the users. The user history data refers to the user's behavior while using the system that is being recorded for analysis. The contextual data from the user along with the historic data of the user presents valuable insights in order to provide quality recommendations to the user.

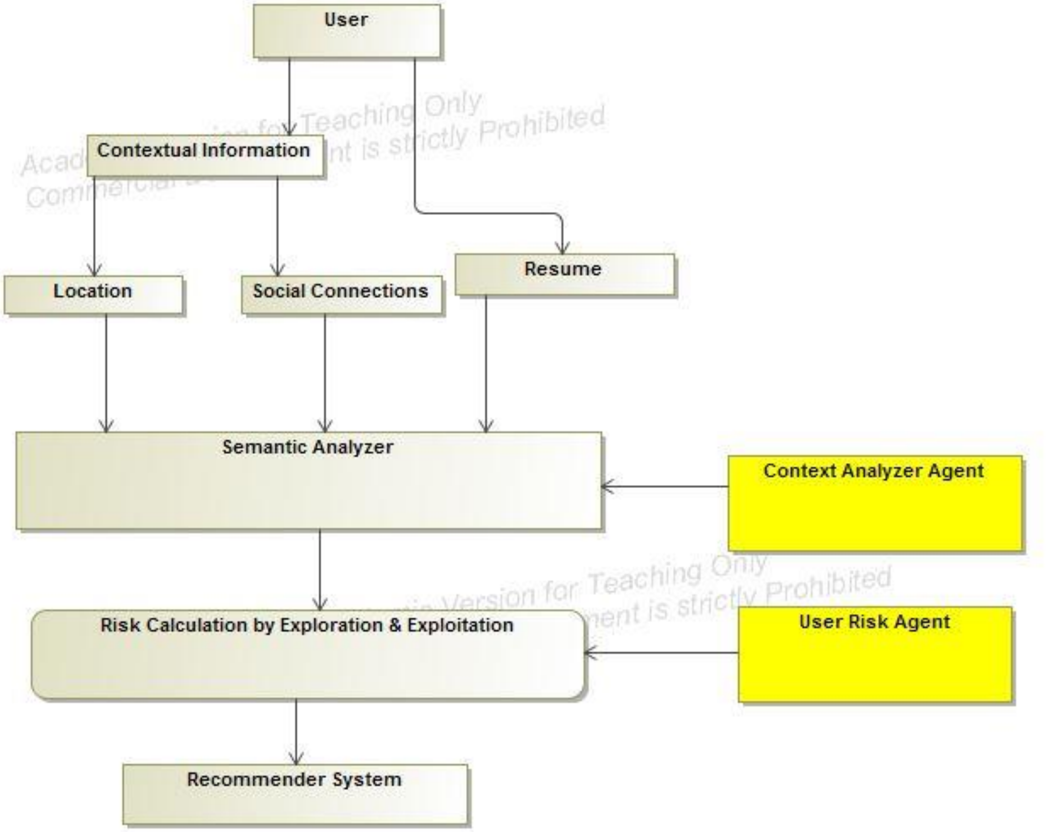


Figure 21 Risk Agent for job recommender

The contextual risk subsystem, as described earlier provides the risk calculation in order to generate suitable recommendations by the recommender system. The contextual information in this case can be location, time, social activities (i.e. likes, page visits).As described in the earlier sessions, this system consists of two agents the Context Analyzer Agent and the User Risk Agent. The information processed in this step is utilized by the recommender system to generate more contextually aware system by not only providing more relevant information to its users but also keeping itself aware of the risk associated with disturbing or negatively affecting the user with the bad recommendation. This tradeoff of providing relevant recommendations and the associated risk is the part of risk calculation through the exploration and exploitation problem.

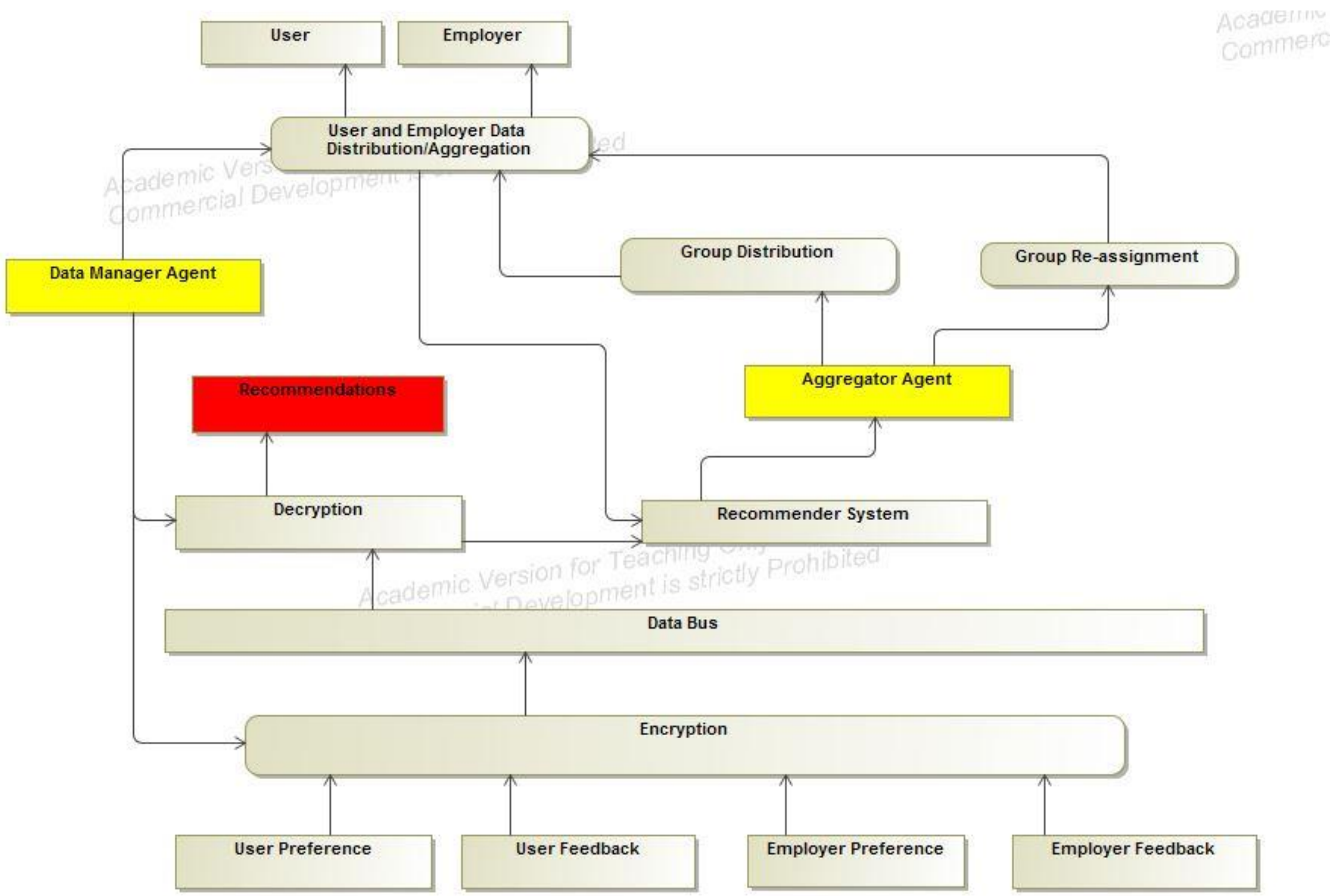


Figure 22 Job Data Agent

The Data subsystem manages the data flow within the recommender system. It manages the data from the user and the employer and the subsequent distribution of that data between different channels and filters such as noise filters which is usually followed by encryption/decryption. This is one of the most important subsystems and probably serves as the backbone of the entire system.

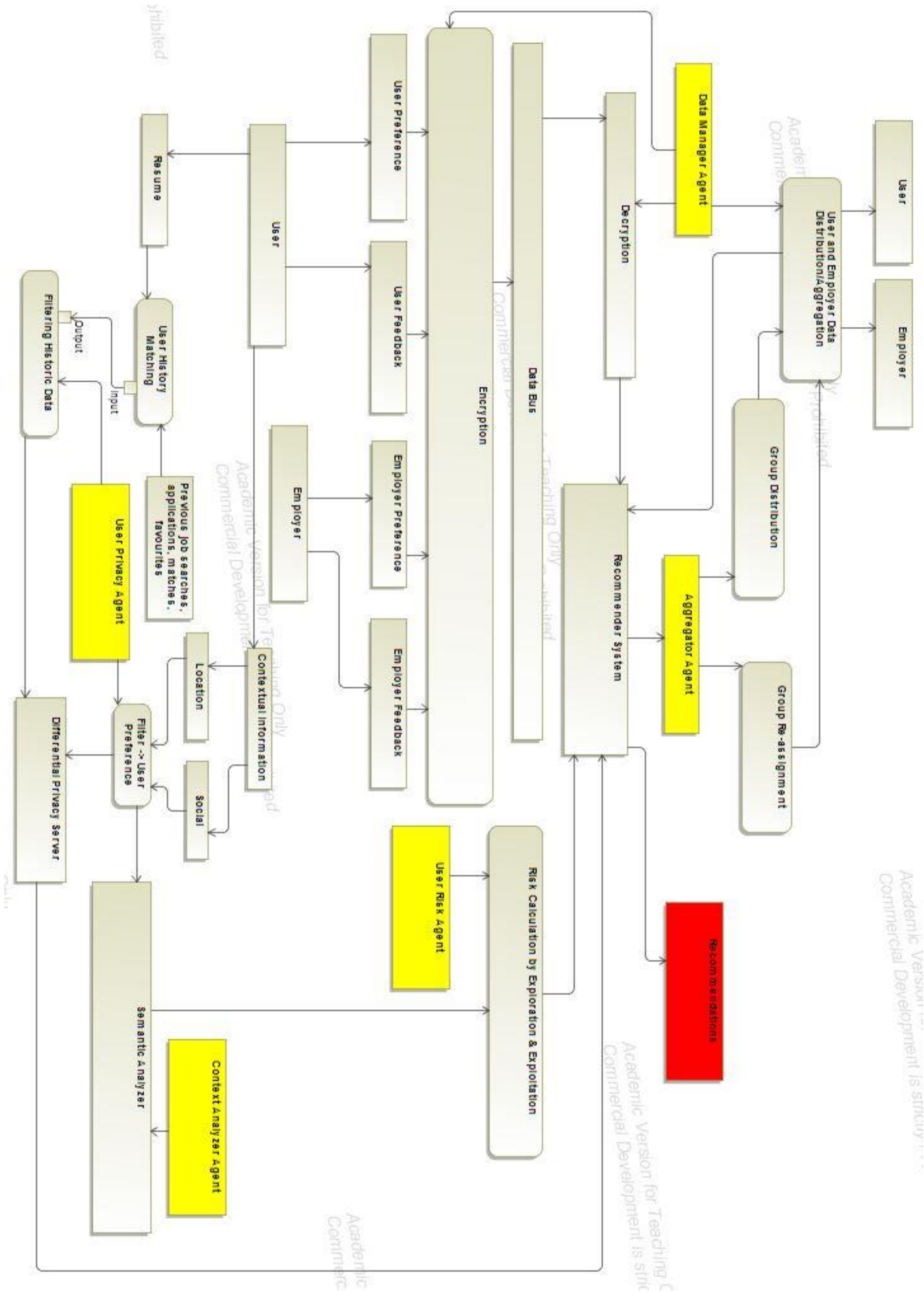


Figure 23 Job recommender system model

Chapter 6

Conclusion & Future Work

6.1 Conclusion

We have now show how the proposed approach provides a broader aspect to the system description and evaluation of a recommender system across multiple application areas. We have also shown that this approach utilizes multi-agent system description in a sense that the designers of the recommender systems can focus on individual units by breaking the recommender system into small individual units enables fast and fault tolerant development of the system. It also enables the designers of the recommender system to be aware of the each of the small objectives that must be accomplished by the each individual units in order to fulfil the objective of the entire system. Also the five dimensional approach, discussed in this paper can be used to gain information about the recommender system functioning across different application areas and a common metrics can be established in the future to better evaluate different recommender systems using a common evaluation tool. This high level approach to describe the system is helpful for domain experts to gain valuable knowledge of a recommender system, operational in a particular application area in a short period of time.

6.2 Future Work

The future work can be focused on either reducing the number of dimensions from five, in order to better represent the system by finding relationship or equations between the existing dimensions. More dimensions can be added into the system by figuring out more parameter for the evaluation of the recommender system across multiple platforms.

Quantitative analysis can be performed over the recommender systems across multiple dimension in order to find the optimal values for each of the existing dimension that the recommender system must satisfy. These optimal value can be served as the threshold values for these dimensions and the recommender systems can be characterized based on these threshold values. The characterization of the recommender system could lead to a standard for evaluation for these systems contrary to the existent metric i.e. the accuracy and predictability.

This metric of recommender system evaluation will be more efficient and fair because sample used for predicting the accuracy can be colluded or the testing and validation set for determine the accuracy of the recommender system might not be applicable in the real world applications.

Bibliography

[1] Girardi, Rosario, and Leandro Balby Marinho. "A domain model of Web recommender systems based on usage mining and collaborative filtering." *Requirements Engineering* 12.1 (2007): 23-40.

[2] Rasmussen, Curtis, and Rozita Dara. "Recommender Systems for Privacy Management: A Framework." *High-Assurance Systems Engineering (HASE), 2014 IEEE 15th International Symposium on*. IEEE, 2014.

[3] Sankar, C. Prem, R. Vidyaraj, and K. Satheesh Kumar. "Trust Based Stock Recommendation System–A Social Network Analysis Approach." *Procedia Computer Science* 46 (2015): 299-305.

[4] Zhao, Vicky Na, Melody Moh, and Teng-Sheng Moh. "Contextual-Aware Hybrid Recommender System for Mixed Cold-Start Problems in Privacy Protection." *Big Data Security on Cloud (BigDataSecurity), IEEE International Conference on High Performance and Smart Computing (HPSC), and IEEE International Conference on Intelligent Data and Security (IDS), 2016 IEEE 2nd International Conference on*. IEEE, 2016.

[5] Elmisery, Ahmed M., Seungmin Rho, and Dmitri Botvich. "Collaborative privacy framework for minimizing privacy risks in an IPTV social recommender service." *Multimedia Tools and Applications* 75.22 (2016): 14927-14957.

[6] Ma, Xindi, et al. "APPLET: a privacy-preserving framework for location-aware recommender system." *Science China Information Sciences* 60.9 (2017): 092101.

[7] Bouneffouf, Djallel. *DRARS, A Dynamic Risk-Aware Recommender System*. Diss. Institut National des Télécommunications, 2013.

[8] Wang, Zhibo, et al. "Friendbook: a semantic-based friend recommendation system for social networks." *IEEE Transactions on Mobile Computing* 14.3 (2015): 538-551.

[9] Guo, Shiqiang, Folami Alamudun, and Tracy Hammond. "RésuMatcher: A personalized résumé-job matching system." *Expert Systems with Applications* 60 (2016): 169-182.

[10] Lu, Yao, Sandy El Helou, and Denis Gillet. "A recommender system for job seeking and recruiting website." *Proceedings of the 22nd International Conference on World Wide Web*. ACM, 2013.

[11] McSherry, Frank, and Ilya Mironov. "Differentially private recommender systems: building privacy into the net." *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2009.

[12] Shokri, Reza, et al. "Preserving privacy in collaborative filtering through distributed aggregation of offline profiles." *Proceedings of the third ACM conference on Recommender systems*. ACM, 2009.

[13] Shang, Shang, et al. "Beyond personalization and anonymity: Towards a group-based recommender system." *Proceedings of the 29th Annual ACM Symposium on Applied Computing*. ACM, 2014.

[14] Zhang, Bo, Na Wang, and Hongxia Jin. "Privacy concerns in online recommender systems: influences of control and user data input." *Symposium on Usable Privacy and Security (SOUPS)*. 2014.

[15] Zimmermann, Thomas, and Christian Bird. "Collaborative software development in ten years: Diversity, tools, and remix culture." *Proceedings of the Workshop on The Future of Collaborative Software Development*. 2012.

[16] El Helou, Sandy, et al. "The 3A contextual ranking system: simultaneously recommending actors, assets, and group activities." *Proceedings of the third ACM conference on Recommender systems*. ACM, 2009.

[17] Adomavicius, Gediminas, and Alexander Tuzhilin. "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions." *IEEE transactions on knowledge and data engineering* 17.6 (2005): 734-749.

[18] Beel, Joeran, et al. "Research-paper recommender systems: a literature survey." *International Journal on Digital Libraries* 17.4 (2016): 305-338.

[19] Jeckmans, Arjan JP, et al. "Privacy in recommender systems." *Social media retrieval*. Springer London, 2013. 263-281.

[20] Jafarkarimi, Hosein, Alex Tze Hiang Sim, and Robab Saadatdoost. "A naive recommendation model for large databases." *International Journal of Information and Education Technology* 2.3 (2012): 216.

[21] Dey, Anind K. "Understanding and using context." *Personal and ubiquitous computing* 5.1 (2001): 4-7.

[22] Melville, Prem, and Vikas Sindhwani. "Recommender systems." *Encyclopedia of machine learning*. Springer US, 2011. 829-838.

[23] Mooney, Raymond J., and Loriene Roy. "Content-based book recommending using learning for text categorization." *Proceedings of the fifth ACM conference on Digital libraries*. ACM, 2000.

[24] Grudin, Jonathan. "Partitioning digital worlds: focal and peripheral awareness in multiple monitor use." *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 2001.