Recommender systems and market approaches for industrial data management

PhD thesis
From

Torben Jeß
Robinson College

Distributed Information and Automation Lab
Institute for Manufacturing
Department of Engineering

University of Cambridge

Date:
6. December 2017

This dissertation is submitted for the degree of Doctor of Philosophy

Recommender systems and market approaches for industrial data management

Author: Torben Jeß

Abstract

Industrial companies are dealing with an increasing data overload problem in all aspects of their business: vast amounts of data are generated in and outside each company. Determining which data is relevant and how to get it to the right users is becoming increasingly difficult. There are a large number of datasets to be considered, and an even higher number of combinations of datasets that each user could be using.

Current techniques to address this data overload problem necessitate detailed analysis. These techniques have limited scalability due to their manual effort and their complexity, which makes them unpractical for a large number of datasets. Search, the alternative used by many users, is limited by the user's knowledge about the available data and does not consider the relevance or costs of providing these datasets.

Recommender systems and so-called market approaches have previously been used to solve this type of resource allocation problem, as shown for example in allocation of equipment for production processes in manufacturing or for spare part supplier selection. They can therefore also be seen as a potential application for the problem of data overload.

This thesis introduces the so-called RecorDa approach: an architecture using market approaches and recommender systems on their own or by combining them into one system. Its purpose is to identify which data is more relevant for a user's decision and improve allocation of relevant data to users.

Using a combination of case studies and experiments, this thesis develops and tests the approach. It further compares RecorDa to search and other mechanisms. The results indicate that RecorDa can provide significant benefit to users with easier and more flexible access to relevant datasets compared to other techniques, such as search in these databases. It is able to provide a fast increase in precision and recall of relevant datasets while still keeping high novelty and coverage of a large variety of datasets.

Declaration

I hereby declare that this dissertation titled *“Recommender systems and market approaches for industrial data management”* is the result of my own work and includes nothing, which is the outcome of work done in collaboration except as declared in the Preface and specified in the text. Additionally, this dissertation is not substantially the same as any that I have submitted, or, is being concurrently submitted for a degree or diploma or other qualification at the University of Cambridge or any other University or similar institution except as declared in the Preface and specified in the text. I further state that no substantial part of my dissertation has already been submitted, or, is being concurrently submitted for any such degree, diploma or other qualification at the University of Cambridge or any other University of similar institution except as declared in the Preface and specified in the text.
The length of this dissertation is less than 65,000 words, including appendices, bibliography, footnotes, tables and equations not to contain more than 150 figures.

Torben Jeß

Cambridge
6. December 2017

Table of Contents

List of figures

List of Tables

1. Introduction

1.1 The data overload problem

Industrial companies, such as suppliers, manufacturers, and distributors, maintain databases with large amounts of data [1], [6], [3] and this data is constantly increasing [1], [2], [4]–[8]. The data is allocated to users in the form of datasets to help them make better decisions, for example, decisions about supplier selection, manufacturing operations scheduling, and inventory management, among many other areas. However, due to the increase in the amount of data, finding the relevant data for a user can be difficult. Companies are often overloaded with datasets [4], [9]–[15] and they often cannot decide which datasets to present to users. This is due to the following two data challenges faced by many industrial companies.

- *Large amounts of data*: Driven by the increasing availability of new technology, such as increased storage capacity and better sensor technologies, the amount of data increases by approximately 40-100% every year [1], [2], [4], [5].
- *High variability of user data requirements*: Task difficulty increases are driven by a) the increasing diversity of tasks that a user is required to perform, and b) the automation of simpler tasks. Some studies suggest the task complexity per user increases by 6.7% every year [16], [58].

Due to these problems, companies miss opportunities in their data [4], [11], [12], [18]–[22]. For example, if procurement has data on all orders placed to a certain supplier, it can ask for discounts [23]. However, this is often not the case for various reasons, such as old legacy systems. In other situations, users are overloaded with data and cannot decide what is relevant to the decision at hand. This problem is called data overload [11].

Current approaches to these problems are requirement analysis and decision theory, intended to identify and provide the right data. However, these approaches often rely on static allocation of data to users using fixed queries or manual user searches of various databases based on fixed requirements.
Search as another alternative approach often shows the data the user enters for the search. All these approaches often require heavy implementation in complex information systems, difficult analysis of users' needs, or searching through company databases. Therefore, current solutions i) enable less discovery of datasets previously unknown to the user or the organisation, and ii) do not prevent irrelevant data from appearing.

Recommender systems and market approaches have shown good results for similar types of problems. However, there is a lack of methods, applications, and proven benefits from these techniques regarding industrial data allocation. This thesis analyses architectural approaches using recommender systems and market approaches. These approaches are used in this thesis individually and in combination. Based on this analysis, the thesis identifies an approach called RecorDa (Recommender systems and market approach based data allocation) with two variations. One variation uses a standalone recommender system and another variation combines the recommender system with a market approach. RecorDa provides the user with additional relevant data in a flexible manner and improves the user's decision-making.

The recommender system finds interesting data and recommends it to the user. For the second variation, the market approach uses the rankings and details on data usage from the recommender systems to further analyse data relevance and to influence the recommendations based on this analysis.

1.2 Introducing recommender systems and market approaches

1.2.1 Recommender systems and data overload

Users often have various choices (i.e. datasets) and not enough time to review them. To address this, recommender systems are used successfully to solve various data overload problems, such as online shopping (where stores typically show 'additional items') and online movie selection [24], [25], [26]. However, recommender systems must be adjusted for industrial data for two main reasons.

First, industrial data has specific characteristics, such as many recommendable items in the form of many data fields. Many of these data fields are similar in their content descriptions. For example, every row in a table has a similar structure and may have similar values either syntactically (e.g., all values in a column are yes or no, all values in a column are dates) or semantically (e.g., all values in a column are surnames). This makes it more difficult for a recommender system to distinguish between items of data, and to recommend the relevant data to a user.

The second reason is that many techniques used by recommender systems require a content description of the data to make recommendations (these systems are so called content-based recommender systems). Therefore, a technique for describing the content of data for recommender systems is required.

1.2.2 Market approaches and the resource allocation problem

In industrial companies, there are

a) *datasets*, which have costs for providing them, and
b) *users*, who have only a limited ability to be presented with data and receive varying benefit from different datasets.

This is similar to a food market, where there are

a) *products*, which cost money to provide, and
b) *customers*, who have limited money.

Ensuring that only the relevant data is kept for potential presentation is a resource allocation problem[1] known from market approaches [27]. Market approaches work by assigning value to resources given the user's needs [28]–[31] and then letting the user bid in auctions for the use of the resource. This thesis aims to ensure that the relevant data reaches the right user. The problem of data allocation is therefore similar. Various researchers have used market approaches in similar domains [32]–[35] or suggested their application to data management problems [22], [28]. However, to date there are no specific applications or suggestions for implementing market approaches in the problem of data overload to thus obtain better data allocation. The main difficulty is finding mechanisms to show different combinations of data to the user and to identify their relevance, which can then be used with a market mechanism. This thesis therefore tests market approaches to improve the overall relevance of data shown to the user.

1.3 Research questions and methodology

In the previous sections, this thesis introduced one of the main problems in data allocation, data overload, and the potential of recommender systems and market approaches to overcome this limitation. Therefore, this thesis sets out to answer two main research questions:

1. What is the best way of using recommender systems and / or market approaches in industrial data allocation to improve performance in terms of precision, recall, novelty, coverage and computation time?
2. Can recommender system and market approach individually or in combination identify relevant data better than potential alternative techniques?

These questions are addressed by assessing different architectural approaches and through a series of case studies and experiments comparing recommender systems and market approaches against alternatives.

[1] Resource allocation is the allocation of a limited resource, such as food to users, who are interested in this resource.

1.4 Using recommender systems and market approaches for data allocation

Given that recommender systems and market approaches have worked successfully in other applications, the main challenge of this thesis is to show that they can overcome the problem of data overload either individually or when they are combined. There are various ways to use recommender systems and market approaches. This thesis assesses them against different alternatives by analysing their potential for solving the data allocation problem.

Based on this assessment this thesis focusses on a series of similar specific architectural approaches called RecorDa, which use the following components:

- a recommender systems component, to suggest relevant datasets to the user and identify which data the user would like to be presented with regularly, and
- a market approach component upon the recommender system, to use the datasets most often presented to the user, evaluate their relevance, and improve the overall relevance of datasets that are presented to the user.

The recommender system focusses on initial data presentation and is similar to recommender systems used at Amazon and similar online retailers [36]. A series of adaptations makes it applicable to data allocation challenges. Using the recommender system on its own is one variation tested in this thesis.

As an additional variation, the market approach component uses the usage and ranking information from the recommender systems and evaluates its overall relevance to all users using a utility function. Thus, it decides which datasets should be eliminated and no longer shown to any user because the relevance of the datasets is too low.

This thesis tests various variations of this approach by examining different variables and components.

1.5 Definitions

This thesis uses a series of terms that require further definition to ensure clarity in their application.

Term	Definition
Data	‘Data is the representation of facts as text, numbers, graphics, images, sound or video’ [37].
Decision	The selection of an option from a series of available options based on the available information.

Decision-making	The process that a user follows to reach a decision. It often involves looking for additional data.
Information	Information is defined as ‘data placed in context’ [20], [37].
Information system	“Information systems use data stored in computer databases to provide needed information” [40].
Knowledge	Knowledge is defined ‘as anything that is known by somebody’ [38], [39] in the organisation.

Table 1: Key definitions for this thesis

The terms *information* and *data* are closely linked. This thesis assumes that a user is presented with relevant information can place it in context and transform it into information and knowledge. This thesis therefore primarily uses the term data for the concept of showing additional data to the user, who can then transform it into information and knowledge, if the user understands it. If the user does not understand it, the data is considered irrelevant and hence less likely to be shown. This thesis only refers to information or knowledge where it is used as a standard term in the industry (e.g., Value of information, Knowledge management). Besides these core terms, there are various types of data that require definition before method proposed in this thesis can be described.

- *Data table*: A set of data values using a model of vertical columns (identifiable by name) and horizontal rows [37], [41].
- *Database*: A collection of data tables.
- *Dataset*: A subset of rows and columns from a data table.
- *Relevant data*: Relevant data is data that would improve a user’s decision. Relevant data can be either known or unknown.
- *Known data*: Data that is relevant to the user and the existence of which the user is aware of.
- *Presented known data*: Data that is presented to the user within the existing graphical user interfaces of the user’s information systems.
- *Non-presented known data:* Data that is not regularly presented to the user, requiring the user to search for it in a system that the user would not normally use for decision-making.
- *Unknown data*: Data that is relevant to a user, but the existence of which the user is not aware of within the company or from external sources. This could for example be a dataset about the likelihood of supplier bankruptcy that has not been given to a user in procurement who must make a supplier selection. Unknown data can be either organisationally known or organisationally unknown.
- *Organisationally aware data*: The user does not know this data for various reasons. For example, the user could be a new employee, or may not be aware of a new dataset that is available.

- *Organisationally unaware data*: Pieces of data (or datasets) that were only found to be relevant when they were presented to the user, with no prior knowledge from anyone in the organisation.

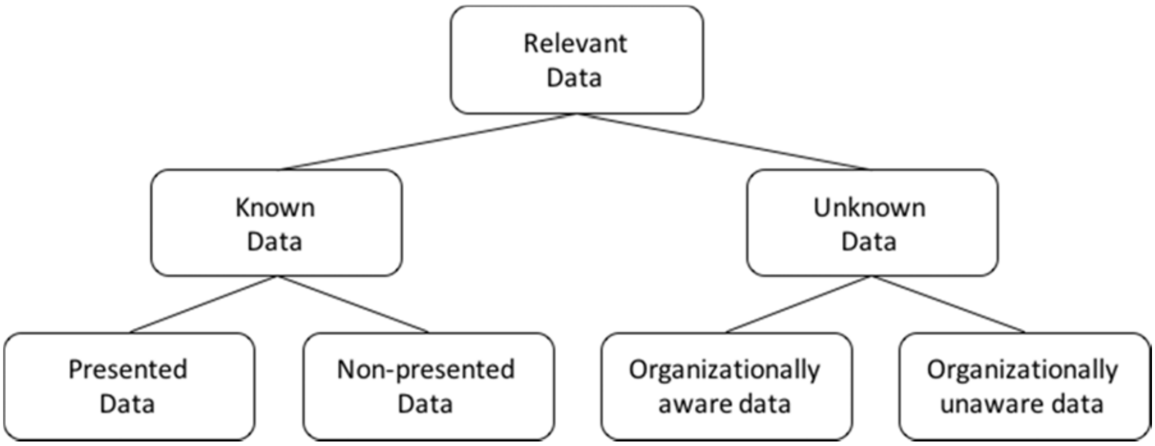


Figure 1: Types of relevant data and their relations to each other

1.6 Evaluation of the RecorDa approach

The evaluation of the RecorDa approach attempts to verify that the treatment – the RecorDa approach – can identify relevant data for a given user and present this data to the user in a better way than can existing techniques. For comparison, this thesis mainly uses search because other techniques are not able to cover a large number of datasets. Regarding search, this thesis assumes different types of search behaviours and compares them to the RecorDa approach.

The aim of all techniques is to improve the relevance of the data for a user in order to improve data allocation and reduce data overload. To evaluate data relevance, this thesis classifies data into eight categories. For each category, different measurements can be used to evaluate the benefits of the different treatments.

	Known data		**Unknown data**	
	Presented data	**Non-presented data**	**Organisationally aware data**	**Organisationally unaware data**
Relevant data	Measured with precision and recall metrics			Measured with metrics such as novelty or coverage
Non-relevant data				

Table 2: Matching types of data with their relevant evaluation metrics

A good treatment should achieve a high relevance of data. It can do this by correctly allocating as much data as possible to the following categories:

- Relevant and known presented data
- Non-relevant and known non-presented data
- Relevant and organisationally aware data

These can be easily measured with precision and recall metrics, evaluating how accurate a specific technique is in providing the relevant data to the user. This is the typical metric used in information retrieval.

However, there is an additional group of data, organisationally unaware data, which is found to be relevant or irrelevant only once it has been shown to the user. Ideally, presenting many datasets to the user reduces the amount of data in this group, because the user can then form an opinion about it. These types of data therefore cannot be evaluated with precision or recall metrics, but instead require metrics such as novelty (how new is the dataset presented to the user) and coverage (how many of the possible available datasets have ever been shown to the user) [42].

1.7 Thesis novelty, results, and contributions

Existing applications that address the data allocation problem are often limited, and there is the potential to use recommender systems and market approaches to close this gap. The main limitations in the existing research are the following:

- Recommender systems and market approaches have been suggested as solutions to data-management problems, but these suggestions are often highly unspecific and not adjusted for the data allocation problem.
- There are no applications of data allocation that use recommender systems or market approaches.
- The benefits of market and recommender system approaches for the data-allocation problem have never been tested.

This thesis aims to address this research gap by finding the most promising approach (for improving precision, recall, novelty, and coverage) that uses recommender systems and market approaches for data allocation. Based on this evaluation, this thesis identifies a suitable approach (called RecorDa) and demonstrates how this approach would work. The results show that some existing techniques (i.e., requirement analysis techniques) are more precise than RecorDa in providing the user with relevant datasets, but more inflexible in finding additional datasets and in showing datasets of which the user is not aware. The RecorDa approach further outperforms similar techniques such as search in its ability to provide relevant datasets under changing conditions.

The findings of this research could help industrial companies develop and use better systems of data allocation. By incorporating RecorDa into their software, these companies can better leverage their data. These companies would gain a tool for use in situations that require additional flexibility when reacting to new user interests in data. The tool can also help them reduce the impact of data overload.

1.8 Applicability of this research

This thesis is focussed on data management for industrial companies. In addition, it is most suitable to the following situations:

- Types of data: This thesis focusses on structured data. It mainly addresses data at the data-table level. The approaches presented in this thesis show structured datasets to the user. However, these approaches could potentially be expanded to use unstructured data.
- Types of users: This thesis is focussed on user decision-making based on data presented by an information system (e.g., ERP systems)
- Types of decisions: The approaches presented in this thesis help users obtain more relevant data to improve their decision-making. These are found to be most useful in repeated decisions instead of just one-off decisions (made only one time in an organisation). The approaches require decisions to be made repeatedly to improve the data shown to users and to generate benefits. However, these decisions do not need to be made by one user. The presented approaches perform particularly well when the same type of decision is made by several people independently.
- Types of information systems: The approaches presented in this thesis require interaction with the information system to identify the current data a user is looking at and show additional relevant datasets. The approaches are therefore limited to information systems that enable this type of functionality.

While elements of this research might be suitable to other applications, these are the situations this thesis is analysing and testing.

1.9 Key assumptions

This thesis is based on a series of assumptions. These assumptions guided this research and helped clarify its direction. The main assumptions are the following:

- Users can rate the relevancy of data presented to them: This thesis assumes that when a user sees an additional dataset, the user is able to determine its relevance with a certain degree of accuracy and will provide ratings (on a scale from 1 to 5) to the approach presented in this thesis.

This assumption requires several abilities from the user and is hence relaxed in the comparison in Chapter 7 of different accuracies in user selection.

- Users improve their data selection abilities: Users are capable of improving their data selection abilities and improve in selecting the type of data they are interested in.
- Additional data can be presented to the user: The approach presented in this research aims to present additional data to the user. It therefore must assume that the existing systems enable it to capture data that is currently presented to the user and enable additional data to be presented to the user (e.g., on the left or right side of the existing system).

1.10 Thesis outline

This thesis is organised into the following chapters:

- Chapter 2 – Research background: This chapter presents the initial relevant research background by providing an overview of the current industry and research practices and their limitations, and by discussing the possibility of using recommender systems and market approaches in other areas with similar characteristics.
- Chapter 3 – Research methodology: This chapter presents the research questions, as well as the guiding hypothesis and the approach used to answer them.
- Chapter 4 - An approach to using recommender systems and markets: This chapter compares potential methods of using recommender systems with or without market approaches and selects the variation of an approach called RecorDa most likely to improve precision, recall, novelty, and coverage.
- Chapter 5 – Introducing the functionality of the RecorDa recommender system component: To answer the research questions, this chapter introduces the first part of the RecorDa approach. It describes how the recommender systems and their potential variations operate.
- Chapter 6 – Introducing the functionality of RecorDa with the market approach component: Building on the recommender system component, this chapter describes how the market approach component of the RecorDa approach and its potential variations work.
- Chapter 7 – Evaluation: This chapter analyses the benefits of the proposed approach (i.e., the RecorDa approach) and its variations by comparing it to existing solutions.
- Chapter 8 – Conclusion: Based on the evaluation, this chapter provides an overview of and an outlook on the main findings of this thesis.

2. Research background

2.1 Introduction

This chapter describes the academic background of this research as well as the existing literature on the topic. It does this by examining:

1. current data management practises; and
2. applications of market approaches and recommender systems.

For the first part, the chapter aims to provide an overview of industrial data management challenges (specifically in data allocation due to the data overload problem) and the current data management practises, in order to describe the environment in which a market and recommender system approach will have to operate. It further explains why solutions such as market approaches and recommender systems are used.

For the second part, the chapter provides details about applications of market approaches and recommender systems in other domains, and the challenges they help to address there. The aim is to show the similarities between these domains and current data allocation problems.

The chapter is structured as follows:

1. Chapter 2.2 focuses on the industry background and describes the underlying issues driving the necessity of a market and recommender system approach. It describes the increasing amount of data used by industrial companies and the problem of increasingly complex tasks for decisions makers. It demonstrates the issues arising from these problems for industrial data management.
2. Chapter 2.3 describes the current data allocation challenges and approaches that are based on the underlying industry complexity. It also discusses the existing techniques to overcome these problems.
3. Chapter 2.4 and 2.5 examine other applications of market approaches and recommender systems in domains with similar characteristics.
4. Chapter 2.6 then describes the research gap using market approaches and recommender systems to overcome the problems in data allocation.

2.2 Industry background

Industrial companies' decision-making is based on data [39]. These decisions may for example be related to investments or supplier selection. In order to facilitate this decision-making process, data is directly allocated to the user by the information system. An information system can be defined as 'the entire collection of data sources and related service capabilities, both internal and external to the organization, from which the users of the system may obtain messages' [39].

The problem is that the allocation of specific data to specific users for decision-making is becoming too complex, causing companies to miss various opportunities for better decision-making. The reasons for this are the increase in the amount of data and the increased complexity of user tasks. These factors lead to the data overload faced by many industrial companies. Reducing the problem of data overload caused by these two industrial developments is the main aim of this research.

2.2.1 Increasing number of data and data users

The amount of data is increasing rapidly every year. Precise numbers vary depending on the source and analysis. According to Manyika et al. [1], there is an increase of 40% every year, while this is 54% according to BAE Systems Detica [2] and some estimate that there is up to a 100% increase for the top companies [6]. Moreover, according to Feldman [4] and Bughin et al. [5], the amount of data doubles every 18 months, Industrial companies store a large proportion of this data. Rolls Royce is collecting huge amounts of data from its turbines [7], and a new Boeing 787 generates over half a Terabytes of data during every flight [8].

In 2012, 18% of companies reported data silos and data volume among their top concerns [6]. The average 1,000-employee UK company already stores 870 terabytes of data [2], which is more data than the Library of Congress [1], [3]. There are various drivers for this increasing amount of industrial data, including:

1. Increasing storage capacity
2. Improved sensing technologies
3. More publicly available datasets
4. More metadata generation

Increased storage capacity

The first driver for increasing the amount data is the increasing storage capacity [43]. This makes storage cheaper and easier; hence companies store more data. Given More's law, which has continued to be observed since 1965 [44]–[47], it can be assumed that this trend of increasing storage capacity will continue into the near or medium-term future. Although storage is increasing, however, the underlying amount of data produced by various systems is predicted to grow at an even faster rate [43], making relevance decisions about which data to store even more difficult in the future, especially considering costs associated with storing data, such as running the storage equipment [47].

In some data-rich industries, such as credit card lending, retail, and health care, the data collected is outgrowing the reduction in storage costs, resulting in a net increase in storage spending [48].

Improved sensing technologies

Over the last few years, sensing technology has improved drastically, making sensors cheaper and more accurate. These improved sensing techniques include better cameras on mobile phones, for example. With these techniques, industrial companies can collect large amounts of data from their operations, supply chains, products, etc. Recent trends such as the Internet of Things clearly reflect this tendency. In 2011, manufacturers embedded over 30 million sensors in their products, and this number increases by 30% every year [49]. While this offers great opportunities for businesses, such as better asset management [50] and improved supply chain management [51], it also creates an increasing amount of data.

More publicly available datasets

Beside the data already created individually by single users or organisations on their own, there is also an increase in public data. This data comes in the form of datasets, such as Amazon's large dataset repository [52] and the various open data initiatives by governments[2], for example. The increase in public data has been facilitated by technologies such as the semantic web over the last few years [55]. Other drivers include the increase in users' video and picture sharing on the Internet [56], and in semi-public datasets that can be acquired from companies.

More metadata generation

Metadata is data about data [37]. Metadata has increased massively in recent years, growing twice as fast other digital data [56].

The underlying trends of all of these drivers are likely to continue. This presents companies with the challenge of leveraging this data to obtain the most benefit from it[3].

2.2.2 Increasing task and organisational complexity

Today's business environment is becoming more dynamic and complex, and it is continuously changing [17]. Indices using the number of procedures, vertical layers, and other organisational complexity metrics to measure company complexity show this complexity increasing by 6.7% every year [16]. Various business trends are causing this increased complexity, including the following examples:

[2] Examples include data.gov in the US [53], or data.gov.uk in the United Kingdom [54]

[3] Some estimates say that only 0.01 per cent of companies' data is valuable/relevant [3].

1. More intense interaction and integration between suppliers and customers due to a reduction in production depth [57].
2. Increased task automation within companies, leaving more users to the more complicated tasks that cannot be automated.
3. Specialisation within the workforce makes the requirements of the whole workforce more diverse. Especially in large industrial companies, employees more often perform a specific task.
4. Companies are focusing more on making decisions based on data, including recent industry trends such as data analytics or data-driven organisation [1]. This requires more specific analysis and more tailored responses to the analytical results.
5. The decision-making is becoming more complicated. Allocating the relevant data to an increasingly complicated decision-making process is hence becoming an increasingly difficult challenge.

In order to overcome some of these challenges and continuously increasing amount of data, software tools have evolved. Among the recent trends are tools and techniques such as data analytics [58] and master data management [59]. Some enterprises have up to 5,000 applications [60] and an increasing number of these are data management tools. While these tools make several aspects of data allocation easier and more efficient, they also make the decision and selection process for many companies even more complicated.

2.2.3 Data overload

The previous two sections showed the problems of increasing amounts of data and increased task complexity. Combined, they lead to the following data problems for industrial companies:

- Data overload of individual users due to this increasing diversity in the user's tasks and environment [11], [12]; and
- Lack of data for specific tasks, and specifically a lack of sharing data along organisational boundaries, such as different departments within a company or different companies [19], [21].

Users are asking for more data sharing among departments and industries[4], while they are suffering from an overload of data [11] (see Table 3 for an illustration).

[4] With lack of data sharing being one of the main failures of the various US secret services that could have prevented the 9/11 terror attacks for example [19].

User problems	Drivers for these problems	
	Increasing amount of data	***Increasing diversity of user tasks***
Data overload	Due to the increasing amounts of data, every user is presented with more data for each data source	Due to the increasing amount of tasks, every user is presented with more varying data sources for each task
Lack of data	Due to the increasing amounts of data, ensuring that the relevant data reaches the right user is becoming more difficult	Due to variations of tasks, the user does not receive all the data that the user would require for each of these tasks

Table 3: Illustration of the user problems caused by data overload and user task diversity

The underlying problem, the so-called data overload or file allocation problem, is shown to be NP-complete [61], [62]. Solving this internal resource allocation problem in the best possible way can be a great source of operational advantage [63]. According to Eppler and Mengis [11], data overload can be categorised into the following groups:

- Cognitive overload
- Sensory overload
- Communication overload
- Knowledge overload
- Information fatigue syndrome

The cause for these issues can in turn be categorised into five groups [9]:

- Data itself (too much, too frequent, too intense, or data quality)
- The person receiving the data
- Processing and / or communicating this data
- Task or process that the user needs to complete
- Organisation and the design of the organisation

Data overload usually happens as a combination of these issues. It often leads to disregarding low-priority inputs, paying less attention to each input presented to the user, shifting some of the data overload problem to other users, filtering specific data, refusing to receive communication, and creating institutions that receive the data [9].

Providing all data to users is not an option due to the large amounts available and the need to drastically limit it. Simply providing the most current data is also not a suitable approach [9], [48]. Therefore, approaches to provide the user with the relevant data for the right task at the right time are needed. Data is typically relevant when allocated in the right amount at the right time. Too much or too little data is less relevant (see Figure 2).

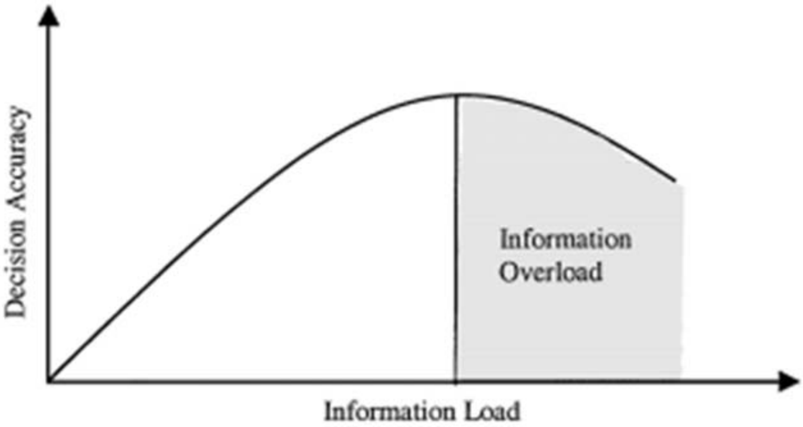


Figure 2: Concept of information overload (referred to as data overload in this thesis) from Eppler et al. [11]

The relevance of data initially increases when more data is shown, but then decreases once a certain amount of data is reached. The ideal amount of data varies depending on the user and the context of that user's decision.

2.3 Data management

The goal of information systems is to 'improve the solutions to decision problems whose outcomes are consequential to the organization' [39]. Information systems are expensive and up to 40% of all company information technology (IT) spending is for maintenance [64]. They therefore need to find ways to deal with the data overload presented in the previous section. Data sources are a key component of information systems and one of the main sources of the data overload problem. Data needs management in large organisation. 'Data management is a corporate service which helps with the allocation of data services by controlling or co-ordinating the definitions and usage of reliable and relevant data' [20]. It 'consists of: The planning and execution of policies, practices, and projects that acquire, control, protect, deliver, and enhance the value of data and information assets' [37]. Good data management can help companies to obtain an competitive advantage [65], [66], and various existing techniques within data management can already be considered to address the problem of data overload.

In order to achieve this overall objective, data management can be divided into five strategic and four non-strategic goals [37]. The strategic goals are:

1. 'To understand the information needs of the enterprise and all its stakeholders.' [37]
2. 'To capture, store, protect, and ensure the integrity of data assets.' [37]

3. ‘To continually improve the quality of data and information […].’ [37]
4. ‘To ensure privacy and confidentiality, and to prevent unauthorized or inappropriate use of data and information.’ [37]
5. ‘To maximize the effective use and value of data and information assets.’ [37]

The non-strategic goals are:

6. ‘To control the cost of data management.’ [37]
7. ‘To promote a wider and deeper understanding of the value of data assets.’ [37]
8. ‘To manage information consistently across the enterprise.’ [37]
9. ‘To align data management efforts and technology with business needs.’ [37]

All of these goals are covered by a broad range of existing research [37].

The aim of a market and recommender system approach is to overcome the data overload by giving the relevant data to the right users. It therefore focuses on the strategic goals 1 and 5 of data management. It also supports the non-strategic goals 6 and 7 by providing cost and revenue estimates and a prioritisation of importance for different datasets. However, there are a series of current techniques that already address parts of these challenges:

1. Value of Information (VoI) techniques
2. Search
3. Data analytics and business intelligence
4. Data development
5. Data architecture management
6. Metadata management
7. User interface design

The following sections will provide a detailed overview of the existing work in these fields regarding data management.

2.3.1 Value of Information techniques

The main fields that use VoI are computer science, economics, and business management [67]–[71]. In these fields, VoI is used to analyse data quality questions about specific issues in some data sets, as well as in more strategic problems about the sharing of data with partners of an organisation. In order to make decisions related to these issues, a value of different pieces of data needs to be calculated or estimated.

Calculating the VoI is difficult because information is an experience good [72]. In order to calculate information, most researchers have drawn from decision theory, the influence that a piece of information has on a decision, and the assessment of the economic value of this information [43],[73]. Different decisions have different outcomes based on the action implied by the decision. For decision theory, information influences the way the actions are selected and therefore the outcome. When all other variables stay the same, it is then possible to analyse how different information impacts the outcome of a decision, which can be used to calculate the value of this information [39], [73]. Howard's first paper on the VoI [74], [75] already defined VoI in this way. VoI often relies on analytical techniques, such as Bayesian networks or other approaches [74], [76]–[79]. A different, less analytically focused research field relies mainly on surveys [80]–[84]. This implies asking users for their estimates or opinions regarding which data they would find relevant for their task. These techniques can be applied to specific information or to whole information systems [83].

In a sensor network, the value estimates are used to decide which sensors to keep and which not to keep [85]–[87]. Yemini et al. [88] provide an example of how value estimates are used to dynamically adjust the allocation of services in an information system environment. Other research has shown that value-based file storage is a promising approach [89].

Other approaches use the Value at Risk of information [48] or policies [90] to identify which data is more relevant for a company. However, they also rely on estimates from administrators and experts to calculate a solution to determine which data should be kept in which manner. In addition, they do not address the process of delivering the data to the user.

Overall, there are various issues with VoI techniques.

- It has been shown that more data does not always lead to better decision-making, such as in Huber et al. [91].
- Information is an experience good, which often requires the user to use the data in order to decide on its value [72].
- These techniques often assume a certain user reaction when presented with this data. However, the user's reactions might be case-dependent, and may vary over time, making the analysis or survey more complicated.
- The value of a piece of data can depend on the type of access, the time of acquiring it, or the specific content. A user can subscribe to a data source or pay for it on a per-use basis. For instance, a user can access limited data of the Financial Times website for free or pay for a subscription, while a user of Apple's iTunes pays for every separate song. Balazinska et al. [92] describe various forms of subscriptions and related issues in market approaches.

- The assessment of data can be a complex process. It requires identifying all potential inputs (or messages) from the data, the statistical output of all of these messages, and the relationship between different messages [39], [73]. It is therefore difficult to conduct this kind of analysis for a large number of different types of data and different combinations of users with different tasks. Performing this analysis for every piece of data is too complex, particularly because each piece of data could have various impacts within a company, which are not always predictable, or could be included in analytical models that tend to focus on a limited number of impacts on decisions. Moshowitz [93] mentions that this analytical or mathematical approach ‘is not primarily economic value’, meaning that an analytical approach does not primarily cover the true value of a piece of data because it does not consider the various economic influences of a piece of data.
- VoI techniques are used in high-impact and specific cases, such as oil or gas, healthcare, plant, and manufacturing design [78]. This is because the effort of conducting these analyses is often only justifiable when the impact and the stakes are large enough.

VoI approaches are therefore less scalable and not suitable for the large amount of lower-impact VoI analyses that take place in many industrial companies.

2.3.2 Search

Search describes the process in which the user types in keywords and then uses these keywords to scan through a set of databases. It has the ability to find certain pieces of data within a larger mass of data. The most famous applications for search are on the Internet and include websites such as Google and Yahoo. Tools such as data retrieval and indexing can also be seen as search mechanisms [94].

Search existed even before computers did, in the form of registers in libraries, for example. Search has long been used in computer science and especially in personal computers, for example with the Unix ‘find’ function. Within the current information systems environment, search finds either identical, or semantically or syntactically similar terms to the term written in the search query within a database. In order to execute the search process, search engines typically rely on the following three kinds of techniques.

1. Syntax and Semantics: Searching for similarity of the keyword typed into the search engine to the word in the database. The first search engines in computers relied intensively on measures for semantic similarity in order to find the right website to match a user’s request.
2. Structure: Using connections between different data items in order to identify the most relevant one for the user. Google, for example, uses PageRank [95], and other search engines use other kinds of network

measures, such as those presented in Kleinberg [96] and Lempel et al. [97]. These measures use hyperlinks, for instance, to identify the most central elements within the network. This gives them an indication of the quality of a website. If a page has various links, it should be better than other sources. These structural measures are often combined with syntax, semantic, and categorisation.

3. Categorisation: This is the structuring of content into specific categories to make it easier to find that content.

Search has already been applied in companies' databases, with techniques such as Google enterprise solutions, for example [98]. By producing search results, search engines implicitly make assumptions about which data is the most relevant to the user, and provide data in a more flexible manner. Search is one of the solutions with the most similarity to the approach that will be described in this thesis. However, it is limited because it requires the user to know what to search for and where to search for it. Neither is a given in large, complex organisations. Moreover, search also often involves additional process steps that many users are not always willing to take to complete a task.

There are three types of search that are relevant to the present research. The first is when the user knows where to find data and which system or database provides this data (called directed search within this thesis). This would be the case, for instance, if the user goes directly to Amazon to find a specific product to buy. The second type of search is when the user does not already know where to find this data (called undirected search within this thesis). This would be the case if the user uses Google to find certain data. The third is a combination in which the user initially does not know where to find the data but then over time improves in finding it by gaining additional experience (called learning search within this thesis).

2.3.3 Data analytics and business intelligence

Big data uses machine-learning techniques to draw insights from datasets. It automatically analyses datasets to improve the decision-making of industrial companies [58]. Similar approaches have been used for several years and are known as business intelligence [99], [100], [104]. Data analytics can present a great advantage for industrial companies [59]. However, data analytics does not always help the user's decision-making, as the latter is often influenced by various factors and values. Data analytics focuses on the data of which the user is aware, and helps with the decision-making after the data is available for the user. Identifying the relevant datasets for analysis and decision-making remains one of the main challenges of industrial companies.

2.3.4 Data development

Data development aims to identify the requirements for data that exist in a company, creating solutions to the problems and then implementing them [37]. It uses specific modelling of user requirements of the data to inform the detailed design of the data using specific databases or data tools. It then develops the most suitable design for the data within the company, and provides the data to the user using various techniques, such as data quality management and data integration. Finally it implements these techniques by converting data to different databases, for example [37].

However, data development mainly addresses the overall design of the system, and not the specific allocation of additional datasets to users and the overcoming of data overload. These are implicit issues. Data development faces the same issues of increase in data volume and increase in user complexity that have already been mentioned for data management in general. It does not overcome these problems, but instead applies existing solutions, such as search, as part of its toolbox.

2.3.5 Data architecture management

Data architecture management involves '[d]efining the data needs of the enterprise, and designing the master blueprints to meet those needs.' [37] Data architecture management aims to design standards and architectures for data management based on the company's higher-level goals [37]. Similarly, to data development, it does not help to truly overcome the problem of data overload. It provides more of a framework in which a market and recommender system approach operates rather than a true alternative to overcome the problem of data overload.

2.3.6 Metadata management

Metadata management ensures that the right metadata is collected. Metadata can include data such as time of data usage, amount of data usage, and time of data creation or change, for example. This metadata can help to identify the relevance of data to users. However, metadata does not provide a specific allocation of data to users or overcomes the problem of data overload. Nevertheless, it can inform further analysis or additional techniques [37].

2.3.7 User interface design

There are various techniques to improve the capability of users to comprehend data by presenting the data in the right way. These approaches can help the user better understand the presented data [102], [103]. However, it does not solve the issue of data overload or identify the relevant data. Instead, it improves its presentation[5].

2.3.8 Overview

The literature regarding the allocation of data to users can be separated into three types of analysis or approaches:

- Analytical analysis: Using techniques from VoI and decision theory to analyse the impact of specific decisions; and
- Interview/survey-based analysis: Using surveys to identify which users require which data.
- Search based approaches: Using search to find the data know to the user

All of these three types have the following limitations in common:

1. They require users to know about the data in all its potential contexts and applications, which is becoming increasingly difficult given the increase in the volume of data.
2. It is difficult to maintain fine differences within different user groups who might slightly vary in the data in which they are interested. Users are often grouped together without any further differentiation between their tasks. With more task complexity and data volume, this user-group-based allocation can become difficult.
3. Users have to actively look or ask for additional data, and someone has to make an effort to obtain this data

A market and recommender system approach presented in this thesis aims to overcome these limitations to better deal with data overload.

2.4 Market approaches in data management

2.4.1 Background

Markets are used for various applications in the current economy, such as supermarkets. They concern buyers interested in certain products and sellers who

[5] The approach developed in this thesis can integrate with these methods. However, this thesis focusses on the actual comprehension of data into information by the user. This research will therefore not address the issue of data presentation. Although research has shown that it can have significant impact [86], [87]. It could be envisioned that these techniques about user interfaces are incorporated in the presentation of the RecorDa approach.

sell this product to them. Adam Smith identified the use of markets in resource allocation and value estimation in his book, *The Wealth of Nations*, in 1776 [104].

Computer science has adopted the concept of markets to solve certain problems using market-based algorithms. A market-based algorithm 'is the overall algorithmic structure within which a market mechanism or principle is embedded' [105]. It uses concepts from markets such as auctions and negotiations to solve algorithmic challenges (and is called market approach throughout this thesis). Its features include 'decentralization, interacting agents, and some notion of resource that needs to be allocated' [106]. These features are effective in allocating resources and estimating value [105], [107], [108]. The reasons for this attribute [6] of market approaches are still disputed, and various factors need to be considered when discussing them. Potential reasons are the distribution of the allocation problem to various participants, the individual incentive to improve the allocation and valuation, robustness towards a changing environment, and the increased flexibility of individual users.

Tucker and Berman [105] define a market method as 'the overall algorithmic structure within which a market mechanism or principle is embedded'. They distinguish between strong market methods and quasi-market methods. The strong market mechanism is 'close in structure and behaviour to a human market' [105] in which the agents have a 'high degree of independence in their demand and utility functions and their endowments' [105]. Quasi-market methods use fewer degrees of freedom in the sense of flexibility available to agents in the market, [105] and therefore make less use of the market mechanism while following the same principles. The quasi-market approach offers better control for system-wide optimisation and can typically compute better results [105]. Strong market mechanisms are used more often in open systems with access by different parties [105]. This thesis will therefore follow a quasi-market approach, similarly to most research that uses market methods [105].

The strength of market approaches is that it computes complex problems with relatively simple properties [106]. Conversely, related disadvantages are that it can be difficult to design the right properties for these approaches, and that their behaviour is difficult to predict [106]. The present research can build upon a large amount of existing research, as market approaches have been studied intensively.

Criticism regarding applying market approaches usually concerns the following problems that could occur in their applications: the risk of only finding a local optimum, and their reliance on simple game theory rules. Both can cause market approaches not to find the optimal solution [105].

[6] Which Adam Smith called the "invisible hand" [94]

One of the main components of market approaches is auctions [109]: 'An auction is a market institution with an explicit set of rules determining resource allocation and prices on the basis of bids from the market participant' [110]. It is the mechanism combining the buyers' utility or interest to pay for a certain item with the sellers' costs and willingness to sell for a certain price. There are different types of auctions, and they are usually used to allocate a resource from the seller or various sellers to a buyer or a selection of potential buyers. Auctions manage this process with a series of different items, buyers, and sellers, and have been shown to manage the process efficiently. Auctions have been studied intensively [109], [111]. The most common auction forms are English and Dutch auctions [109].

English auctions: In the English auction, the price is set low and then continuously raised until only one bidder remains, who wins and buys the item. The English Auction is equivalent to a second-price bid, in which the person with the highest bid wins the auction but pays the price of the second highest bid [109].

Dutch auctions: In the Dutch the auction, the price is set high and then continuously reduced until the first buyer agrees to it. This type of auction is equivalent to a first-price bid, in which the highest bidder wins and pays the offered price [109].

There are a series of other pricing mechanisms for auction theory. However these two or a slight adjustment of them are the main ones used for most auctions [109].

Many differences exist among different types of auctions. An overview can be found in Krishna [109] or Klemperer [112]. However, there are some specialities regarding the auctions and market approaches used for this thesis that should be explained in further detail:

1. A user interacts with various combinations of datasets instead of just one dataset: This thesis deals with a specific type of auction – the combinatorial auction – for market approaches [109]. In this type of auction, the user does not just bid for one item, but for a combination of items of interest. Various researchers have addressed these auctions in further detail [113]–[115], [125]. Combinatorial auctions lead to a better economical allocation but are also more computationally complex [117]. However, there are approaches for computing computational auctions efficiently [118].
2. User utility: It is difficult to extract the utility from the user [119]. Goldberg et al. [120] describe various steps around these auctions. However, they require a specific value for one item for each individual user, which is the main challenge that the approach needs to overcome.
3. Procurement auctions: Procurement auctions are auctions where the sellers sell items with the goal of maximizing their earnings [109].
4. Low variable costs: Data can easily be duplicated and shared. The incremental costs of reproduction are relatively low in comparison to other

goods, such as manufacturing products for example. This provides specific challenges with regard to pricing and valuing this data [72].

Due to the intensive research about market approaches, various special and established algorithms have evolved and are for example used to find a price equilibrium using a Walrasian approach [62], [121].

2.4.2 Applications of market approaches in industrial companies

Market approaches have been successfully used in various resource allocation tasks, often performing better than alternative resource allocation systems [27]. They have been successfully applied in different industrial scenarios, such as supply chain management, radio spectrum sharing [122], workforce allocation [123], truck allocation [124], airport traffic control [125], project management [126], robot coordination [127], [128], and task scheduling [112], especially in dynamic and complex situations [129].

Applications of market approaches in information systems include the pricing of computation resource use [29], [31], such as memory space or available EC2 instances [32], [33]. Other applications are the protection of information systems with MarketNet [88], [107], [130]–[132], database management using market approaches for query management among various databases [34], bandwidth allocation [73] [31], allocation of CPU and IO capacity [35], and supply chain management systems [134]. The principle of applying market approaches to data management was suggested for a long time [71]. Market approaches are also used to facilitate interactions among different companies, such as supply chain interactions [134], and even to facilitate intercompany data exchange about products [135]. Market approaches to data are particularly difficult because of the low duplication costs [120].

Brydon [136] and others [137] identify market approaches to be useful for resource allocation as a solution to intra-company allocation problems. Brydon [136] mentions ‘self-interest’ and ‘gains from trade’ as the main source of benefits because they allow the decomposition of the problem into various smaller problems. The author acknowledges that market approaches might solve the NP-complete resource allocation problem but at the same time create the winner determination problem in this market, which is also NP-complete [138]. Brydon [63] further presents various issues around developing these market approaches, such as 1) decomposition of the problem in a way that it can later achieve global-level optimisation; 2) identification of value for the various agents and entities within the market; and 3) the decomposition of the problem using market approaches. Overall, market approaches seem to have a good ‘time-quality trade-off’, to be more flexible and robust, and to be used for various objectives [63].

2.4.3 Applications of market approaches in data management

Beside this general work on market approaches in information systems, various authors have realised the potential use of market approaches with regard to internal data and data resource allocation and valuation (see Table 4).

Authors	Description of other market approaches
Yemini et al. [88]	The authors introduce market approaches as a concept of application and service resource management for large-scale information systems, which also provides benefits of relevance estimation. The authors identify various elements that have influence on the market, such as user utility, user budgets, and optimisation targets, as well as the potential to apply more advanced market techniques, such as futures and options. However, this work does not show a concrete application of this market approach type and does not specifically apply it to data, but rather focuses on access to resources and services.
A et al. [87]	The authors describe a market-based approach to a sensor network to help to identify which sensors provide benefits. Their idea is also tested in other research [85], [86]. They address the issue of sensor networks struggling with data overload due to the large number of sensors. Their work examines the data allocated based on user interests. It combines different tasks with the sensors used to execute these tasks. However, the authors mainly focus on sensor use and not on data allocation, and they rely on user input to provide the relevance of data/sensors for a specific task.
Koifman et al. [139]	Koifman et al. describe a system where various webpages deal with data tuples among each other in a network. They use techniques to estimate the quality of a piece of data and develop the negotiation mechanism between the different pages. However, their model mainly aims for the trade between different websites and does not try to identify questions regarding companies' data allocation.
Christoffel [140]	The author describes a market approach used for integrating various data sources. The work shows that markets have abilities, which makes them more flexible and introduces various agent types required to build a market-based approach. However, the work does not cover industrial data allocation.
Wang et al. [62]	The authors introduce market approaches to better manage the allocation of data from data sources to users. The idea they describe is to have agents compete to deliver data to users. However, they only address the relocation of data resources in order to be more attractive to the users, but do not address industrial data allocation to specific users.
Koroni et al. [28]	Koroni et al. provide an overview of so-called 'internal data markets', and their approach and is similar to the approach that this research aims to develop. They introduce the idea that market approaches can be used for data evaluation, evaluation of data's quality, costs

	of data, and benefits that data can create. They also identify the main challenges in developing an ‘internal data market’: 1) organisational buy-in in the form of data transaction evaluation; 2) data quality problems; 3) standardisation, meaning issues around development of a consistent data product that can be sold repeatedly; and 4) product packaging, since the data often needs to be pre-processed before it is shown to the user. Overall, they indicate some potential benefits and challenges but they do not show ways to overcome these issues or concrete implementations of ‘internal data markets’.
Wijnhoven et al. [141]	These authors present an approach to align internal database ontologies for a data market and the importance of ontologies for internal data markets. They provide insight into the standardisation of internal data products with regard to quality and ontology, but do not address data allocation.
Dignum and Dignum [22]	The authors describe the application of market approaches for knowledge management. In their work, market approaches serve to incentivise participation in knowledge management.
Koutris et al. [105K]	Koutris et al. describe an approach of trading with online datasets and user queries accessing these datasets, called query market. This approach mainly addresses combining different datasets for user queries while still enabling payment to a combination of data providers. The authors generally describe how the pricing for a combination of datasets is a computationally difficult problem [142], [143], [144]. However, they do not address the issue of evaluating data relevance or of using fixed prices set by the data provider. They also do not influence the user’s selection of the data.

Table 4: Overview of data-related applications of market approaches

Besides the explicit uses of market approaches in data management, various implicit uses also exist. These various applications use different auction mechanism, market protocols, and other kinds of variations in market techniques [105]. Overall, however, the existing work on market approaches within companies has several limitations:

1. It does not address the issue of data allocation to users.
2. It only outlines the concept and leaves various open questions for practical application.
3. It does not show how data can be evaluated with limited user input (which can be expected in industrial companies)

This research aims to analyse the potential benefits of market approaches by focusing the applications of markets on these three limitations and potentially using them with recommender systems to overcome data overload.

2.5 Recommender systems in data management

2.5.1 Background

Recommender systems recommend items that they identify as relevant to a particular user. They are intensively used in online stores [24], such as Amazon [145]. Research on recommender systems started with Goldberg et al.'s work [146]. A review of the existing work on recommender systems can be found in Park et al. [147]. Recommender systems use items (the entity that is recommended) and users (the entity to which the item is recommended). To make their recommendation, they try to estimate the ranking that a user is expected to have for a previously unseen item. This ranking can either be made directly by the user, with the latter specifically stating the ranking, or indirectly by the user's actions, such as clicking on a link, selecting an item to buy, and spending time on a website or product description.

The estimation techniques for these rankings can be clustered into three categories of content-, user- or item-based recommendation [25], [148]:

- Content-based filtering: These techniques suggest items to the user that are similar to the item that the user is looking at [25].
- Item-based filtering: These techniques suggest items to the user that are similar to items that other similar users rate highly [146].
- User-based filtering: These techniques suggest items to the user that are rated highly by similar users [146].

All three techniques compute a similarity score of the unseen item based on the existing rankings and other similarity functions. They then use this similarity score to calculate the expected missing rating. This rating is subsequently used to generate suggestions for the user. In addition, hybrid approaches also exist that combine these two techniques [25]. They often outperform algorithms that belong strictly to one class in some practical applications [149].

Recommender systems have been shown to reduce search effort [150] and to address the data overload problem [151],[152]. Some researchers have claimed that recommender systems might make search redundant in the future [153]. However, these two techniques are often combined, such as in Google's auto-complete functionality.

2.5.2 Applications of recommender systems in industrial companies

Recommender systems in industrial companies have been used in various applications, the best-known being the presentation of items in ecommerce [24], [154] such as Amazon, and the search for content [150], [153] such as in Google

and Netflix. However, these applications are usually outward-facing towards the customer, suppliers, or other external entities. Besides these external-facing applications, there are also adoptions of recommender systems towards internal usage. They have been applied to knowledge management [26], internal documents [151], corporate services [155], recommending datasets to a user in the field of economics [156], and SQL query recommendation for users [157].
Although there are various similarities between existing approaches, recommender systems have not been applied to data allocation and data overload.

2.5.3 Applications of recommender systems in data management

Recommender systems are an intensively researched field, and various techniques and approaches have been tried with various adjustments, such as linked data [158],[159] and recommender systems for apps [160], for example.

The five most important existing approaches with regard to the present research are the following.

- Market-based recommender systems: Market approaches have been applied in recommender systems and researched in various domains by Wei et al. [161]–[165], Bohte et al. [166], [167], Melamed et al. [168], and Bothos et al. [25]. The authors use a variety of different recommender systems that compete for the user's attention and have to make bids in order to obtain that attention [169],[170].
- User-focused recommender systems: Many existing recommender systems mainly work on the side of the selling company, such as Amazon. These systems' main goal is to ensure that they increase the revenue of the selling company, and they only partially account for the interest of users. To overcome this limitation, recommender systems that increase the user's utility have been developed [171],[172].
- Recommender systems within data allocation: Recommender systems have also been applied to companies' internal data, such as in knowledge management [26]. Glance et al. [94] introduce an approach to using recommender systems within organisations called the Knowledge Pump. Users can bookmark data, receive recommendations, and make recommendations to other users. The authors describe various issues regarding employing recommender systems within companies, such as the smaller number of users, the need to be used intensively by people, and issues around incentivising users to use the recommender systems and make recommendations[7].

[7] In order to incentivize users they developed a virtual currency as a reward for good recommendations and they specifically mention the potential of their system to help in the calculation of Value of Information [94].

- Distributed recommender systems: These comprise different recommender systems that exchange data with each other to improve their recommendations [173].
- Profitability-based recommender systems: These systems aim to improve the profit of the selling company [174] instead of simply finding what the user might like.

The overview of existing approaches shows that recommender systems are a good tool for allocating data items to users in a flexible way, and that various ways of doing this have already been analysed. They have been shown to deal well with data overload in online news [175], for example, but have not been used in any application regarding data allocation in companies.

2.6 Summary

This literature review has demonstrated that the following five types of approaches can potentially be used to address the data overload problem:

- Search
- Analytical approaches or decision theory
- Survey/interview-based approaches or requirement analysis
- Market approaches
- Recommender systems

These approaches have been applied to the data allocation problem in varying degrees. When examining the different types of implementation, this review found the following degrees to which they have been applied and implemented.

a. Industrial application: Checks whether an approach has been used for other industrial applications.
b. Data allocation in non-industrial applications: Checks whether an approach has been used for data allocation in a different domain.
c. Suggested for industrial data allocation: Checks whether an approach has been suggested as a solution for data allocation.
d. Methods for industrial data allocation: Checks whether an approach has been adjusted to work as a method for industrial data allocation (e.g. dedicated architectures).
e. Applied to industrial data allocation: Checks whether there are implementations of this approach for industrial data allocation.
f. Tested benefits for industrial data allocation: Checks whether this approach has shown proven benefits compared to existing techniques and whether the nature of these benefits is clear.

An overview of the degree of application of each of the approaches discussed in this chapter can be found in Table 5. The table shows the current lack of implemented and tested market approaches and recommender system approaches. While both recommender systems and market approaches have been suggested for industrial data allocation, to date no research has applied them to industrial data allocation. The suggested methods are vague and lack detailed description of how this application in industrial data management or data allocation might work. Given the limitations of the existing techniques, such as the scalability of surveys and analytical approaches, and the need to know what data to look for in search, recommender systems and market approaches can create benefits in industrial data allocation.

This thesis aims to address this research gap by proposing an architectural approach based on recommender systems and / or market approaches. The following chapter will describe how this thesis will do this, and which questions need to be answered.

Degree of application and implement.	**Search**	**Analytical approaches**	**Survey/ interview approaches**	**Market approaches**	**Recommender systems**
Industrial application	✓	✓	✓	✓	✓
Data allocation in non-industrial applications	✓	✓	✓	✓	✓
Suggested	✓	✓	✓	(✓)	(✓)
Methods	✓	✓	✓	(✓)	(✓)
Applied	✓	✓	✓		

Tested benefits	✓	✓	✓		

Table 5: Overview of different approaches to the data allocation problem and their degree of application and implementation

3. Research methodology

3.1 Research questions

The previous section illustrated the research gap consisting of a lack of scalable and flexible data allocation techniques used to identify relevant data for the user, as well as the identified potential of recommender systems and market approaches in addressing this gap. This thesis aims to develop and test an approach based on recommender systems and/or market approaches for industrial data allocation and to close the research gap identified in chapter 2. This thesis therefore adopts the following hypothesis:

Based on the characteristics identified in the previous chapter, recommender systems and market approaches can be used to identify the relevant data for users in a company and increase the amount of relevant data allocated to the user while reducing the problem of data overload.

To test this hypothesis, this thesis must first answer the question:

1. What is the best way of using recommender systems and / or market approaches in industrial data allocation to improve performance in terms of precision, recall, novelty, coverage, and computation time?

The literature review discussed the potential of these techniques, but the lack of existing methods and applications of data allocation (see chapter 2) has demonstrated the need to initially address this question. The first research question is answered by comparing different ways of using recommender systems and market approaches for data allocation (chapter 4) and then further describing the detailed development of these approaches (chapters 5 and 6).

The approach must then be tested by answering the following question:

2. Can the recommender system and market approach individually or in combination identify relevant data better than potential alternative techniques?

The second question is answered by comparing the accuracy of different techniques in providing different types of relevant data to users by reducing data overload and improving data allocation.

3.2 Research approach

This section describes how this thesis aims to answer the questions identified above. It first describes the research philosophy underlying the epistemological approach of this thesis, which forms the basis for identifying possible existing methodologies. This leads to the methodology selected for this thesis.

3.2.1 Epistemological approach

This thesis adopts a realistic approach to its ontology, assuming that the world is independent of the researcher's perspective and that science must observe nature in order to progress [176], [177]. Vessey et al. argued that information systems research is either descriptive or evaluative [178]. Since this study aims to test the benefits of recommender systems and market approaches in data allocation, it is evaluative by nature. With an evaluative approach, according to Vessey et al. the research can be either positivist or interpretivist [182]. A positivist approach is based on hypotheses, deductions, and causalities. Its results must be replicable and generalisable. Its results must also be quantitative and measurable [177]. The interpretivist view assumes that the world is affected by the subjective judgement of people. Its results must be interpreted and generalised into context [176], [177]. The present research adopts mainly a positivist approach, which fits the underlying realistic ontology, and the hypothesis-driven nature of this research [179], [180]. It attempts mainly to quantitatively measure the positive benefits of the approaches developed and presented in it. However, for the design of the experiments and the case studies, it adopts an interpretivist view to gather qualitative input through expert opinions. The specific application of these two views will become clearer in the following subsection.

3.2.2 Selected research approach

In answer to the first research question, this thesis discusses the limitations of the existing techniques identified in the previous chapter. This thesis then identifies potential ways of using recommender systems and market approaches, either alone or in combination, and selects the approach most likely to improve precision, recall, coverage, and novelty for detailed analysis.

Leveraging the approach identified by the first question, the second research question uses a framework from Yin [179] identifying the appropriate method to use in research (see Table 6). This thesis adapts this framework in accordance with Kitchenham and Pickard [180], including the question of 'Which is better?' for experiments and case studies.

Given the research questions and the focus on contemporary events, experiments and case studies were identified as possible approaches. These are the methods typically used in information systems research [180].

Strategy	Form of research question	Requires control of behavioural events?	Focus on contemporary events?
Experiment	How, why, which is better?	Yes	Yes
Survey	Who, what, where, how many, how much?	No	Yes
Archival analysis	Who, what, where, how many, how much?	No	Yes/No
History	How, why?	No	No
Case study	How, why, which is better?	No	Yes

Table 6: Framework for research method adaptation based on Yin [179] and Kitchenham and Pickard [180]

Experiments and case studies offer different benefits for answering the research questions. In addition, to the difference in level of control identified by Yin [179], further differences are highlighted by Pfleeger [181] (see Table 7).

Factor	Experiments	Case Studies
Level of control	High	Low
Difficulty of control	Low	High
Level of replication	High	Low
Cost of replication	Low	High

Table 7: Factors relating to choice of research technique identified by Pfleeger [181]

Important to answering the second research question, the characteristics of experiments and case studies have desirable attributes for different research stages. Due to the novelty of using recommender systems and market approaches for data allocation, various parameters within these approaches needed to be tested, requiring more replications and a larger level of control within the testing environment. Compared to informal experiments, formal experiments are often smaller in scale, more scientifically rigorous, and better when comparing different approaches [180], and they have a higher level of control (meaning the ability to adjust an experiment more directly, precisely, and systematically) [181], all of which were desirable when conducting the initial testing of the recommender systems, market approaches, and potential alternatives. Therefore, experiments were adapted in the early stages of this research to identify key factors influencing the performance of the different approaches to data allocation.

However, the use of case studies is favourable due to the following: the control of behavioural events mentioned by Yin [179]; the desirable aspect of less control in a realistic environment to identify behavioural variables (of companies and / or employees) not considered in the experiments; and the limited generalisability of experiments to a range of industrial problems[8] [180]. Case studies provide a deeper, more valid, and more testable understanding of the true industrial environment [182], they can help judge whether a technology can be used in a company [180], and they help identify potentially previously unidentified variables [183]. This thesis uses different case studies to identify a broader range of variables [184] in different data allocation scenarios. Hence, a set of case studies follows the discussion of the initial experiments. This ensures that the most suitable configurations identified are tested in a more industrially relevant case study environment.

In the experiments and case studies, this research generally followed a hybrid approach based on qualitative research and quantitative approaches, as suggested by various researchers [176], [185], following the positivist and interpretivist views outlined in the epistemological approach. Qualitative approaches were used in the development and identification of the experiments and initial assessments of the different architectural approaches. Quantitative measures were used to evaluate these experiments and case studies. The qualitative research in the development of the experiments and identification of case studies ensured industrial relevance. It was mainly based on a literature review (research as well as industrial white papers) and focussed on unstructured interviews with industrial experts. The aim was to identify typical characteristics of data allocation scenarios to select representative experiments and case studies. Due to the relatively small sample size of available experts with specific domain expertise on the various datasets, and due to the detail of information required, interviews offered the best option for obtaining the required information. Surveys do not offer the required level of detail of information. In addition, quantitative approaches provide the measurable facts and evaluation to answer the research questions in a logical and structured manner, following the positivist approach.

3.3 Research methodology

The qualitative analysis identified the research gap, the current industrial problems, and the current industry standard. To close the research gap, this thesis developed an approach using recommender systems and/or market approaches. It initially compared various architectures and identified the most promising approach to improving precision, recall, novelty, and coverage using qualitative criteria. An architecture could be either only a market approach or only a recommender system or a combination.

[8] While Pfleeger [181] mentioned that experiments are more generaliseable the author also illustrated their limitation to the specific experimental setup. In an industrial context with a variety of variables this specific setup can therefore not be generalised.

Based on this analysis, the approach most likely to improve precision, recall, novelty, and coverage was then developed. It was further adjusted and evaluated with quantitative analysis using experiments to develop a set of suitable setup variables. Further case studies and experiments were used to compare this approach against alternatives (see Figure 3).

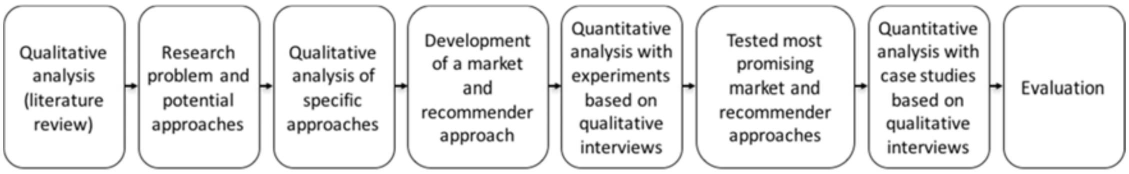

Qualitative analysis (literature review)
Research problem and potential approaches
Qualitative analysis of specific approaches
Development of a market and recommender approach
Quantitative analysis with experiments based on qualitative interviews
Tested most promising market and recommender approaches
Quantitative analysis with case studies based on qualitative interviews
Evaluation

Figure 3: Research process

The experiments and case studies were based on the literature review and interviews with experts in different companies. The experiments followed the approaches outlined by Pfleeger [181] and Kitchenham et al. [180]. These approaches are similar to the steps suggested by Basili et al. [186], who categorised preparation and execution into ‘experiment operation’, and analysis, dissemination, and decision-making into ‘experiment interpretation’.

3.4 Summary

This thesis used experiments and case studies to verify its hypothesis that recommender systems and market approaches can help companies identify relevant data and solve many of its data overload and data allocation problems. The combination of experiments and case studies ensured the large number of repetitions needed to test this approach and the industrial relevance. The experiment design and case study selection were informed by interviews and a literature review. The research was based on a positivist view and a realistic ontology.

The following three sections first determine the most suitable approach (to improve precision, recall, novelty, and coverage) to using recommender systems and/or market approaches to overcome the data allocation problem. Chapter 7 then compares this approach’s performance to alternative approaches to solving the data allocation problem.

4. An approach to using recommender systems and markets

4.1 Introduction

Chapter 2 discussed the research gap regarding methods with tested benefits using recommender systems and market approaches. To address this gap, Chapter 3 identified the following initial research question: *'1. What is the best way of using recommender systems and / or market approaches in industrial data allocation to improve performance in terms of precision, recall, novelty, coverage and computation time?'.* This chapter discusses this research question and defines the recommender systems and market approaches.

Each approach can be divided into two elements:

- High-level architecture: Ways of using recommender systems and market mechanisms to address the industrial data-allocation problem
- Functionality: Functional elements required for recommender systems and market mechanisms to successfully allocate data to users

This chapter first identifies the high-level architecture enabling the main functionalities and then reviews the key specific functional elements for recommender systems and market mechanisms used individually or in combination.

4.2 Selection of high-level architecture

4.2.1 Criteria for selection of high-level architecture

For any approach to presenting additional data to the user, its architecture must have a specific set of high-level functionalities. Functionality based evaluation is a main part of most architecture evaluations [187]–[190].

The approaches developed in this thesis aim to support information systems by providing better data to users[9]. To achieve this, each approach requires specific functionalities. The approach must identify the user and what the user is working on (the task; see criterion A), identify the datasets that the user needs for the current task (see criterion B), and then present these to the user (see criterion D).

However, as shown in subsection 2.2.3, the approaches tested in this thesis aim to reduce data overload with recommender systems and market approaches by finding the most relevant data to the user. To achieve this, these approaches

[9] For this thesis information, systems are defined as systems that "use data stored in computer databases to provide needed information" [40].

require the functionality of ordering data by relevancy (see criterion C) and, ideally, input from the user to improve the ordering of data (see criterion E).

In terms of this research, there are five main functional requirements for the architecture:

A. Identify current task: Relevant data must be applicable to the task of a user.
B. Identify datasets relevant to the current task: The approach requires a mechanism to identify which additional datasets may be relevant out of all the datasets existing in a company.
C. Order datasets by relevance: To present these datasets to a user, the approach must rank them. This way, the most relevant datasets are allocated to the user.
D. Present the most relevant datasets: After ranking these datasets, the approach needs a mechanism to display the relevant datasets.
E. (Optional) Improve on the selected datasets: This criterion is not always required for relevant data allocation. However, given the increasing numbers of datasets and increasing complexity of data (see Chapter 2), few approaches are likely to provide the correct information immediately, and therefore, the approach requires some ability to adjust its relevance evaluation.

Without these steps, no approach can process the large numbers of available datasets and select the ones most relevant to the existing user.

4.2.2 Potential high-level architectures

There are various potential architectures for using markets and recommender systems for industrial data allocation. This subsection tests various combinations. It considers naïve solutions and existing applications for arranging recommender systems and market approaches.

The naïve architectures can generally be broken down into the following four archetypes:

- Recommender only (standalone recommender system): This architecture uses only a recommender system without a market approach.
- Market approach only: This architecture uses only a market approach without a recommender system.
- Recommender first: This architecture uses a recommender system first and then builds the market approach on the recommender system using the recommender system results for its market analysis.
- Market approach first: This architecture uses a market first and then builds the recommender systems on the market using the market results for its recommendation.

In addition to these naïve ways of using recommender systems and market approaches, the literature review (see chapter 2) also revealed the following types of combinations of market approaches and recommender systems:

- Market based recommender systems: An approach in which a market is used for a competition between different recommender systems. Its architecture is similar to the *recommender only* archetype with the difference that, within the recommender systems, various types of sub-recommender systems compete for being allowed to present a recommendation to the user (see Figure 4).
- Recommender systems in online market places: These are recommender systems like those used in online marketplaces (e.g., Amazon). The recommender system presents items. However, if an item receives bad reviews or does not sell enough to make up its purchasing costs, it will no longer be offered to the user. The market therefore helps regulate the offers made by the recommender system. This recommender-system architecture is identical to the *recommender first* archetype.

An overview of each of these systems can be found in Figure 4. The following subsection compares these five approaches.

4.2.3 Comparison

The initial stage of architecture selection compares each of the four high-level architectures from subsection 4.2.2 against the criteria identified in subsection 4.2.1. An overview comparing each of these approaches against the five main criteria can be found in Table 8.

Overall, *market approach only* and *market approach first* have the disadvantage of having no existing technique for initially presenting relevant datasets to the user. These approaches must rely on initial random input or another ordering system, which can then be used to further evaluate the dataset. The recommender system has the existing capability of quickly improving and evaluating initially presented items because it is used for this in other applications, for example, ecommerce.

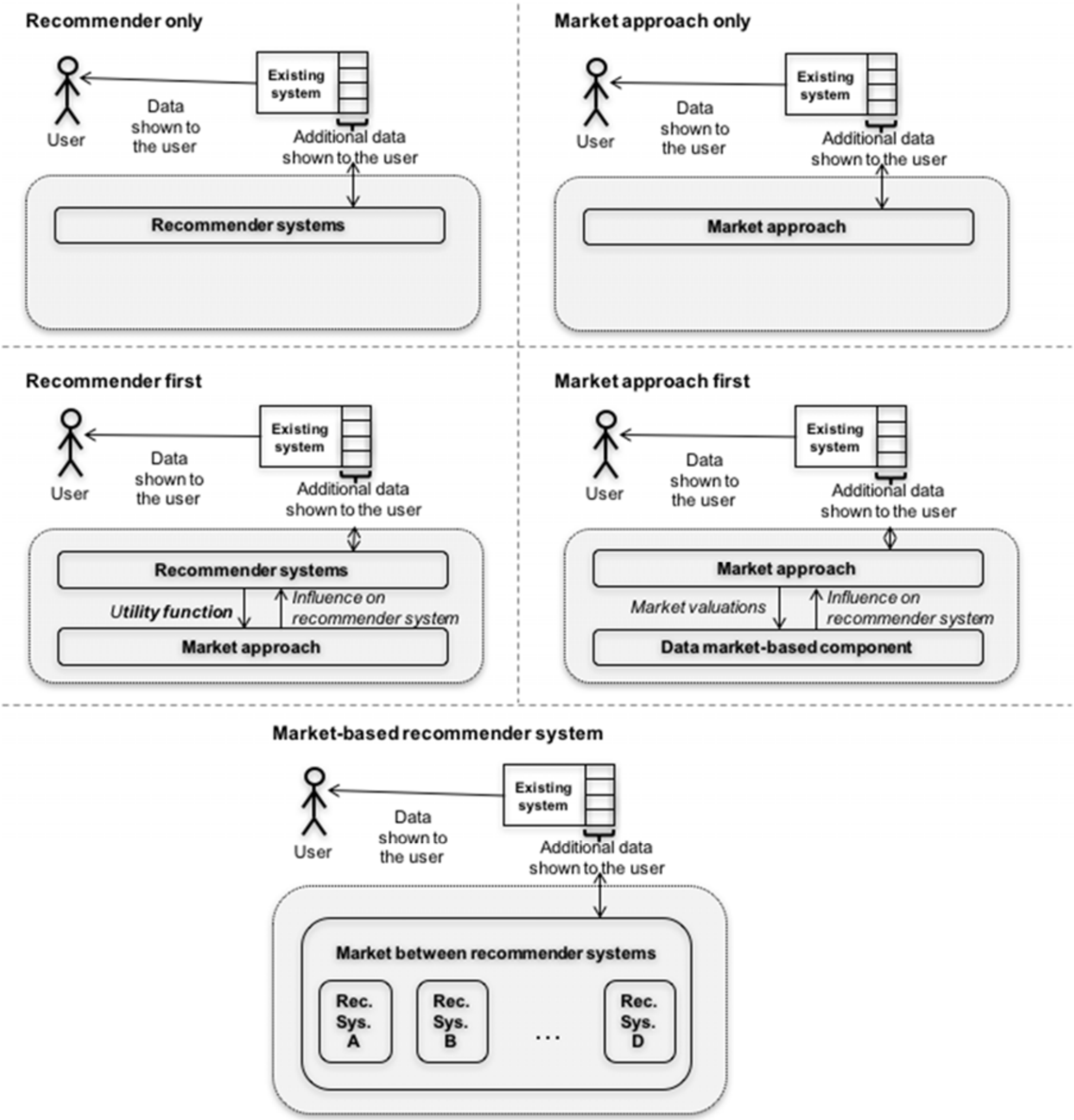


Figure 4: Overview of architectures using market approaches and / or recommender systems

	Recommender only	**Recommender first**	**Market approach only**	**Market approach first**	**Market-based recommender systems**
Identify current task	All approaches can identify the current task and the current dataset.				
Identify datasets relevant to the current task					
Rank datasets by relevance	All approaches can evaluate and therefore rank datasets.				
	Content-based recommendations have been shown to work well without prior information. Recommender systems can therefore deal with the cold-start problem.		Market approaches do not have a method to deal with the cold-start approach of initially presenting datasets.		Setup of these systems is often more complicated due to the large number of recommender systems. They also require the existence of well-functioning *recommender only* archetypes to compete within the market, which is not the case in data allocation.
Present the most relevant datasets	All approaches can present the most relevant datasets to the user.				
Improve on the selected datasets	Recommender systems can receive ranked and therefore have a direct feedback mechanism.		Markets must transform a rating into a utility and they therefore rely on indirect input.		Although these are similar to *recommender only* and *recommender first*, their feedback mechanism might be more complicated because rankings must be attributed to various recommender systems.

Table 8: Comparison between different recommender system and market approach archetypes

Market-based recommender systems are an option for combining market approaches and recommender systems. However, they are more difficult to design and set up. They are also normally developed for domains that are already using existing recommender systems. Several types of *recommender first* approaches should be successfully tested in a domain before market-based recommender systems are applied, which is not the case in data allocation.

Therefore, the *recommender only* (referred to as *standalone recommender system* for the remainder of this thesis) and the *recommender first* approach (referred to as *recommender system with market approach component* for the remainder of this thesis) are the high-level architectures most suited to improving precision, recall, novelty, and coverage. These are also the two most prominent ways of using recommender systems and/or market approaches for websites and other applications. Standalone recommender systems are used on various websites (e.g., for presenting movies to sell to the customer), and recommender first systems with a market approach component are used on websites such as Amazon and eBay, where a recommender system shows recommended items but the market determines which items are profitable enough to be on the website.

4.3 Main functionality

The following subsection describes the main elements of the recommender systems and the market approaches, and then determines the main functionality decisions of each of these approaches separately and in combination.

4.3.1 Recommender system functionality setup

As indicated in the literature review (see chapter 2), hybrid recommender systems often produce the best results. The recommender systems component used in this thesis therefore adopts a hybrid approach combining all three types of recommender systems (i.e., content-, user-, and item-based recommender systems). In a hybrid approach, each system relies on a series of separate functions for computing its results based on the user input, and these results are then aggregated.

The user- and item-based functions are usually based on standard similarity measures (i.e., cosine similarity, Euclidean distance). This thesis therefore compares multiple functions in the experiments of chapter 6. Content-based systems rely on comparison of the content, which for these recommender systems is data. Therefore, for the content-based system, a new approach for data comparison needed to be developed, which is outlined in chapter 5.

For the aggregation, there exists a potentially infinite number of functions combining these different recommender systems. They range from a simple average to more complex techniques such as neural networks. This thesis uses potentially simpler functions, such as average, max, or min. This offers three specific benefits:

- Improvements over other recommender systems: These relatively simple functions have worked successfully in various recommender systems [36], [191], [192].
- Initial nature of this research: This is the first application of recommender systems towards industrial data allocation. Using established and simple algorithms that have been used successfully and repeatedly reduces the risk of problems due to too many complex techniques and enables establishing a performance baseline which can be improved through additional research.
- Attribution of benefits to specific sub-recommender systems: Simpler functions can be understood more easily, which makes it easier to attribute successful recommendations to one of the sub-recommender systems.

Further details on these two critical functional setups (recommender system functions and aggregation) can be found in chapter 5.

4.3.2 Market approach functionality setup

Chapter 2 showed that market approaches rely on two types of input to use their resource allocation capabilities: utility and costs [87], [131], [193]. In addition, the literature review showed that markets need an auction mechanism to combine these two inputs. The market approach therefore needs to decide on the following three main functional variations:

- **Utility function:** There are various potential utility functions which can be adopted using inputs such as data quality and usage. A detailed assessment of the existing literature on potential criteria can be found in section 6.3. It shows one of the most common indicators is usage. Therefore, a usage based utility function was used in this thesis.
- **Cost description:** This thesis attempts to capture all costs for maintaining and providing datasets to users in the future. A detailed breakdown of the costs can be found in section 6.5. Estimating these is difficult. There are various complex methods for estimating software development costs and costs of datasets. Finding a suitable method for a large number of datasets is complex, however. This thesis found that experts can provide helpful estimates for these costs. It hence applied an interview based method for cost evaluation.
- **Auction mechanism:** An auction mechanism covers the type of auction and the mechanism controlling the participants.

- Type of auction: English auctions and Dutch auctions are used for similar problems [87] and are the two most commonly used auction types [112]. They also provide similar outcomes to various alternative auction types [112]. This thesis therefore applies these two auction types.
- Auction organisation: There are centralised approaches in which a market maker takes all the price offers and a decentralised approach in which each market participant trades with each other [112]. This thesis uses a centralised approach because it is easier to compute[10] and is a commonly used approach for similar problems [87].

For this initial research, this thesis used a market maker approach and tested the most common auction types, English auctions and Dutch auctions.

These three functional decisions were key to setting up the market approach. They determined the initial direction of development which is detailed in chapter 6.

4.3.3 Setting up the Interface between the Recommender system and Market approach

Furthermore, the market approach required an output in the form of impact on the recommender system. The main questions were a) How are the recommendations influenced, and b) By which measure are they influenced?

- Type of influence: The recommender system could be influenced by either the rank of the datasets in the market approach or their specific evaluations. Both techniques are developed and tested in chapter 7.
- Type of measure: The mechanism could potentially use revenue, costs, or profit as variables computed using the auctions and utility function. Of these, revenue and profit are mainly influenced by the dataset's relevancy to the user. They are therefore the two approaches tested in chapter 7.

These different combinations were tested to find the best way to influence the recommender system based on the market approach analysis.

4.4 The RecorDa approach

The analysis of architectural approaches has shown that, overall, either of the following two high-level architectural approaches seems most suitable (in terms of improving precision, recall, novelty, and coverage) to allocating data to users:

[10] Decentralised solutions are often more complex to develop because various auction participants must be coordinated. It is also more difficult to ensure a good result. As long as the number of datasets does not reach several millions of tables, a centralised approach should remain computable.

- Standalone recommender system, which provides data recommendation without a market approach
- Market approach based on a recommender system, which leverages the recommender system to determine the inputs to the market evaluation

Within each of these, the analysis in this chapter found the following key functional decisions to be the most suitable for future development.

- The following decisions relate to the recommender system:
    - Use of all types of sub-recommender systems (content-, user-, and item-based) adopted towards the problem of data overload
    - Use of a series of simpler functions for aggregating the sub-recommender systems due to the benefits these functions showed in other recommender systems and the initial nature of this research, and to more clearly attribute benefits to specific sub-recommender systems
- The following decisions related to the market approach:
    - Use of utility function based on usage, because this is the most established criteria for evaluating data relevancy
    - Use of cost assessment based on expert interviews because this is only a single effort per dataset and is hence potentially scalable and easiest to implement
    - The market approach attempts to use the two most established auction mechanisms, that is, English auctions and Dutch auctions. They represent the most typically and commonly used types of auctions and compute results equivalent to a series of other auction mechanisms. The approach will combine these with a market-maker mechanism.
- For the impact of the market approach on the recommendations, the selected approach will test four potential types to influence the recommendations.

These methods of using a recommender system, either in combination with a market approach or as a standalone recommender system, with these functionality elements, is called RecorDa (shown in Figure 5).

The first variation of the RecorDa approach consists of a market approach component built on the recommender system component. The standalone recommender system variation relies on only the recommender approach component. The recommender system and market approach architecture rely on both components.

The recommender system component initially identifies which additional datasets are relevant to the user (using the user details allocated in recommender system step 1) by providing the user with likely relevant datasets (step 3) using a combination of different recommender system techniques, such as content-based, user-based, and item-based recommendations (generated in step 2). It improves the recommendations using ratings from the user on these additional datasets (step 4).

The ratings and the logs serve as input to the RecorDa market approach component. A utility function that uses the number of times an additional dataset has been presented to the user evaluates the relevance of different combinations of datasets (step 5).

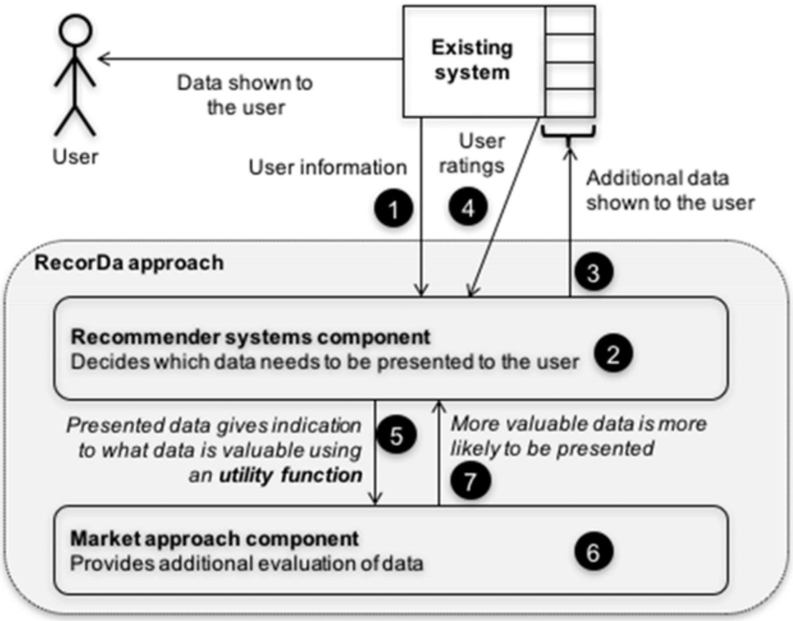

Existing system
Data shown to the user
User
User information
User ratings
Additional data shown to the user
1
4
3
RecorDa approach
Recommender systems component
Decides which data needs to be presented to the user
2
Presented data gives indication to what data is valuable using an utility function
5
More valuable data is more likely to be presented
7
Market approach component
Provides additional evaluation of data
6

Figure 5: High-level architecture of data relevance evaluation and data allocation in RecorDa

Within the market component, the valuation of a combination of datasets is then allocated to individual datasets (step 6). Once the relevance of a dataset has been identified, it must influence which data is presented to a user. A function influencing the recommendations to the user manages this interaction (step 7). This process is completed iteratively each time new datasets are shown to the user to continuously improve the presented data.

This architecture enables the RecorDa approach to follow the steps required for successful data allocation. Details on the two main components can be found in chapters 5 and 6. Chapter 7 then initially analyses various key configurations outlined in chapters 4–6 and compares the best performing configuration against alternatives like search. An initial comparison of market approaches and recommender systems against potential alternatives can be found in chapter 7.

5. Recommender system component

5.1 Introduction

The previous chapter identified the RecorDa approach as the most suitable for using recommender systems and market approaches. The following chapter will describe the functionality for the recommender system component of the RecorDa approach. The functionality follows the key functionalities identified for the RecorDa approach in chapter 4.

5.2 Data allocation with recommender systems

As seen in chapter 2 a recommender system is typically based on three types of recommender systems (user, item, and content based) which are combined to recommend items to the user.
Before using these approaches, the recommender system component of the RecorDa approach first needs to know who the user is and on which datasets the user is using. This allows the approach to approximate the user's task. The user is known based on the login details. The data with which the user is working can be identified and captured from the user interface. The recommender system component therefore takes the data presented to the user within the current information system (called operational record) (see Step 1 of Figure 6 and algorithm 5.1).

Next, the recommender system component identifies the source (the data tables) of the currently presented data (called working tables in Algorithm 5.1). Only then it asks the recommendation engine for additional data tables of relevance to the user (Step 2). All further sub-recommender systems work on a data table and task per user basis[11]. In order to identify data tables relevant for the current task, the recommender system component uses a recommendation engine. As outlined in chapter 4 the recommendation engine is based on the following three separate sub-recommender systems.

<u>User recommender system:</u> Identifies additional data tables by looking for users with similar rankings for other data tables[12]. The RecorDa user recommender system uses existing similarity measures for recommender systems (e.g. Cosine similarity) from the standard Mahout library [194]. Different combinations of these similarity functions are tested in the experiments of chapter 7.4.3.

[11] This ensures a high specificity to provide relevant datasets, but it is also generic enough for the recommender systems to be able to collect various user interactions. Changing the granularity by being more specific e.g. on the data record would create too many combinations (i.e. between all records in a database) or being more generic would lose a lot of granularity.

[12] This thesis uses the Mahout recommender system library and Pearson correlation to find similar users as one implementation.

Item recommender system: Recommends data tables to the user by looking for data tables that are similar to the currently presented data table in a way that they have received similar ratings[13]. It is also using standard similarity measures from the Mahout library [194] which are tested in the experiments to identify the best performing similarity function.

Content-based approach: Uses data characterisation [195] to identify similar data tables. It takes all columns from a data table and generates metadata about the data in the column (e.g. mean word length, fraction of NULL values). A neural network is used to find matches between columns. Data tables with columns that have a high likelihood of matching are recommended as similar content. The benefit of the content-based approach is that it does not require any input from the user. Data characterisation or automatic schema matching is a standard method and various papers have been written about it [195]. It is used to pre-compute the various similarity measures between data tables. However, adopting it for data recommendations by using its results for a content-based approach is one of the novelties of this thesis.

Both the user and the item recommender system are using previous rankings provided by the user to influence their recommendations.

Each of these provides a list of potentially relevant data tables and a relevance estimate for each data table (called table similarity in Algorithm 5.1). Each sub-recommender system is based on the standard techniques typically used in order to ensure relevance of the provided recommendations.

As shown by the second functionality from chapter 4 to rank the datasets, the results of these three separate sub-recommender systems are aggregated into a single list of recommended data tables (Step 3) using the average, maximum or minimum, of the individual calculated recommendation scores (all variations are tested in chapter 7 and typically use in recommender systems). However, there are potentially other implementations which could be analysed in the future.

In order to ensure that the most relevant datasets are presented to the user, it is critical that the recommender system component does not just present the full tables, because the user will not be able to find the relevant records within such large tables. The RecorDa therefore uses the operational record to identify similar records in the recommended data tables by accessing the relevant database (Step

[13] This thesis uses the Mahout recommender system library and Pearson correlation to find similar users as one implementation.

4). Records with an identical join[14] to the operational record are extracted from the system (Step 5) and presented to the user in descending order of the rating (Step 6). The user is presented with data sets that are relevant to the data sets on which the user is working. The user is only shown the matching records (data sets) from these data tables, significantly reducing the search effort and improving the relevance of the presented data.

In the current setting of RecorDa, the user is initially presented with the first five recommended tables (called Top tables in algorithm 5.1) on the side of the existing information system, and has the option to click through to additional recommendations. The user also has the opportunity to rate the data with the getUserRating function in Algorithm 5.1); these ratings are then used to further improve the data presented by the recommender systems by using the item and user based recommendations.

An overview of the recommender system approach can be found in Figure 6. The figure and algorithm 5.1 show all of the computational steps of the RecorDa recommender system component required to ensure that the recommender system can allocate data to the user.

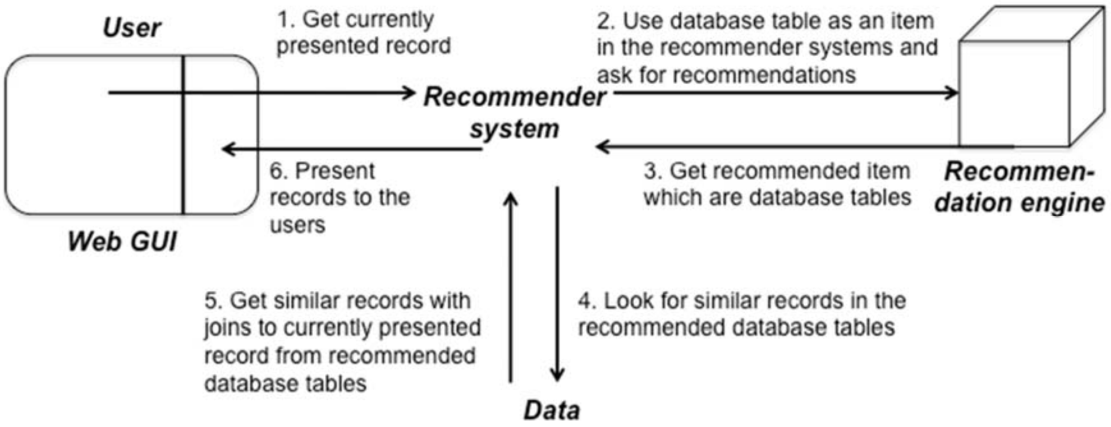

User
1. Get currently presented record
2. Use database table as an item in the recommender systems and ask for recommendations
Recommender system
6. Present records to the users
3. Get recommended item which are database tables
Recommen-dation engine
Web GUI
5. Get similar records with joins to currently presented record from recommended database tables
4. Look for similar records in the recommended database tables
Data

Figure 6: Description of the different process steps of the recommender system

Algorithm 5.1: Recommender system algorithm

```
Variables:
Task = Defines the task that a user is working on
User = Defines the specific user (e.g. via ID, or name)
Record = Defines the specific datasets that a user is working on
WorkingTables = Defines the tables that contain the data from the
record variable
```

[14] The current system works with identical joins. However, further approaches could include not identical joins and also show similar items using techniques like fuzzy matching [196] for example.

TableSim = Defines a matrix of tables and similarity scores for the existing workingTables

SimScores = Matrix of all tables against all other tables containing similarity scores based on its data characteristics. It is pre/computed with data characterisation algorithms.

TopTable = List of highest rated tables that are recommended to the user

MatchingRecords = Records from the TopTable that have a direct syntactic match to the Record variable

Ratings = Contains a list of ratings of user, data table, and rating score for each element in the list

Functions:

GetCurrentUserAndTask = Identifies the current task, user, and record that a user is working with in the Information system

GetWorkingTables = Identifies the tables that a user is currently working with

UserRecommenderSystem = Applies the existing similarity measures for user recommendations from the mahout library using the completed ratings from the user

ItemRecommenderSystem = Applies the existing similarity measures for item recommendations from the mahout library using the completed ratings from the user

contentRecommenderSystem = Identifies tables that have likely similar content from the precomputed SimScores variable by selecting the tables with the highest ranking for the current workingTable

Aggregate = Combines the different TableSims by taking the min, max or average of the values from the sub-recommender system

Match = Gets a list of tables and the current record. It identifies the records from the tables in the list that have a direct syntactic match to the current record

Prsent = Shows the recommended datasets to the user

getUserRatings = Gets the rating from the user when a dataset is presented

Algorithm:

```
// Step 1:
Task, user, Record ← GetCurrentUserAndTask()
// Step 2:
Working_Tables ← getWorkingTables(user, task, record)
// Step 3 (recommendation engine):
TableSimA ← userRecommenderSystem(user, task, ratings)
TableSimB ← itemRecommenderSystem(user, task, ratings)
TableSimC ← contentRecommenderSystem(user, task, simScores)
TableSimAggregate ← Aggregate(TableSimA, TableSimB, TableSimC)
// Step 4:
TopTable ← SelectTopTables(TableSimAggregate)
// Step 5:
MatchingRecords ← Match(Record, TopTable)
// Step 6:
Present(MatchingRecords)
// Capture ratings:
Ratings ← getUserRatings(ratings, user, task, TopTable)
```

There are different functions and approaches for recommender systems. They can be used for all six steps. An overview of these steps can be found in Table 9. Chapter 7 provides further details regarding the evaluation of these configurations for each step.

Step	Configuration
1	This step captures the currently presented tables and rows. There are no different configurations for this capturing process. It relies on existing logs or administrator input.[15]
2	Each of the sub-recommender systems uses a series of different configurations as mentioned in chapter 4.
	User sub-recommender system; there are various methods for identifying similar users given their recommendations: Log Likelihood Similarity [197], City Block Similarity, Euclidean Distance Similarity [197], Pearson Correlation Similarity [198], Spearman Correlation Similarity [197], Tanimoto Coefficient Similarity [197], and Uncentered Cosine Similarity [198]. These are implemented based on the Mahout library [194].
	Item sub-recommender system; there are various methods for identifying similar items given their recommendations: Log Likelihood Similarity [197], City Block Similarity [199], Euclidean Distance Similarity [197], Pearson Correlation Similarity [198], Tanimoto Coefficient Similarity [197], and Uncentered Cosine Similarity [198]. These are implemented based on the Mahout library [194].
	Content-based sub-recommender system: the content-based sub-recommender system relies on external pre-computed input from data characterisers [195]. The specific setup of the data characteriser is not the focus of this research.
	In addition, recommender systems typically rely on thresholds to decide which data to present. Different thresholds are tested in the experiments.
	Cuff-off low-rated data: If a data table has a low ranking, the RecorDa approach will not present this data to a user. This threshold ensures that the user will not see data that the user has ranked lowly. It can be set between 1 and 5.
	Item sub-recommender threshold: threshold for the recommendations from the item sub-recommender system to be considered for the following step.
	User sub-recommender threshold: threshold for the recommendations from the user sub-recommender system to be considered for the following step.
3	Each of these recommender systems provides a ranking from 0 to 1 for the different potentially recommended tables. This step combines these three ratings into one rating. The following are approaches for this aggregation process (as described in chapter 4).
	Max: Takes the maximum from the three recommendation systems.
	Min: Takes the minimum from the three recommendation systems.
	Average: Takes the arithmetic mean from the three recommendation systems.

[15] In the approach presented in this thesis works with tables of rows and columns using relational databases. However, the concepts could be extended to hierarchical data structures such XML or unstructured data such as text files or pdf documents assuming an alternative approach for the content-based recommendations.

	Total threshold: Threshold that the aggregated score needs to achieve to be considered as a recommendation.
4, 5, 6	These steps take the operational records and find rows where there is an identical data cell value in the recommended table for the columns that have sufficiently close metadata. RecorDa then does a join of the tables, providing additional columns from the recommended tables, joined where the data cell values match. This thesis does not use different configurations.

Table 9: Overview of the detailed configurations for the recommender system component

5.3 Summary

The RecorDa approach recommender system component shows that recommender systems can be adjusted in order to provide relevant data to the user. The key adjustments compared to existing recommender systems techniques are the following.

- Defining users: Normal recommender systems work on a per-user basis. However, for RecorDa this needs to be adjusted to the user and task levels to ensure that specific dataset is allocated to each user, without confusing different actions that a user might take in a given system.
- Item definition: Existing recommender systems often work with very granular items (e.g. products on Amazon). However, when addressing datasets, this granularity is often difficult to handle because there are several layers of these datasets. The RecorDa approach addresses this issue by aggregating data records on a table level to provide the relevant records from a table to the user.
- Finding an approach for content-based sub-recommender systems: There is currently no approach that deals with content-based matching for recommender systems on datasets. However, content-based matching is often critical for recommender systems to overcome the ‘cold-start problem’ [156], [191] (as seen in chapter 4). By adopting techniques from other domains (e.g. data characterisation) and applying their rating scheme to recommender systems, this thesis closes this gap.
- Fine tuning the systems: There are various variables involved in setting up the recommender system. Chapter 7 provides some insights into considerations and initial results.

In the following section this thesis will address the market approach component.

6. Market approach component

6.1 Introduction

Chapter 4 showed the key architectural decisions for the market approach. It is based on the recommender system (outlined in chapter 5) and uses the following four main elements:

- A utility function based on usage
- Cost assessment based on expert interviews
- Two of the most established auction mechanisms, i.e. English Auction and Dutch Auction.
- Four potential types to influence the recommendations by influencing their ranking

This chapter will describe how the market approach works in more detail based on the architectural decisions and the recommender system component introduced in the previous sections. It will also provide a more detailed reasoning for specific design selections in addition to chapter 4 and give a detailed rational for using market approaches.

6.2 Overall market architecture

As seen in chapter 4 market approaches rely on two types of input to use their resource allocation capabilities: utility and costs [87], [131], [193]. These are then combined via auctions.

For utility the challenge is to identify the relevance of a dataset (see section 6.3). The utility functions are based on the datasets a user finds relevant initially identified with the recommender system component. They identify the utility of combinations of datasets. The results from the utility functions are combined in the Value Map. It is needed to represent the results of the utility functions which are specific valuations for combinations of datasets for specific users (see section 6.4).

For costs it is important to measure the cost of providing the data (see section 6.5).

The RecorDa market approach component (see Figure 7) uses these input of utility and cost. It has the data combinations and their utility on the one side, and the different datasets with their costs on the other. As shown in chapter 2 identifying which datasets provide a high enough benefit in these different data combinations is a difficult NP-hard problem (see section 6.6). Market approaches overcome this problem with auctions (see section 6.7). Based on the profits and losses for datasets, market approaches can then be used to influence the order in which they are presented. To then further improve the datasets presented to a user the RecorDa market approach component influences the data shown to the user (see section 6.9). It is thus also able to continuously improve its allocation. A high-level architecture of the market approach within data allocation can be found in algorithm 6.1.

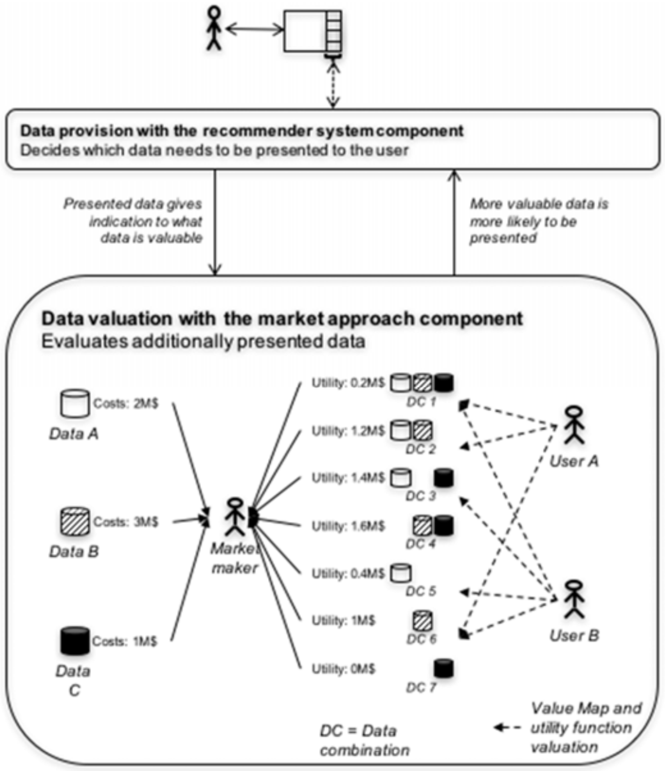

Data provision with the recommender system component
Decides which data needs to be presented to the user
Presented data gives indication to what data is valuable
More valuable data is more likely to be presented
Data valuation with the market approach component
Evaluates additionally presented data
Costs: 2M$
Data A
Costs: 3M$
Data B
Costs: 1M$
Data C
Market maker
Utility: 0.2M$
DC 1
Utility: 1.2M$
DC 2
Utility: 1.4M$
DC 3
Utility: 1.6M$
DC 4
Utility: 0.4M$
DC 5
Utility: 1M$
DC 6
Utility: 0M$
DC 7
User A
User B
DC = Data combination
Value Map and utility function valuation

Figure 7: Description of the overall market architecture

Algorithm 6.1: High-level market architecture

Variables:

NumberOfViewsPerTableComb: Counts the number of times a combination of tables has been viewed by each user

TableCombsUtil: Contains the calculated utility for each combination of tables

TableCosts: Contains the costs for providing each individual table

ProfitsPerTable: Contains revenues and profits generated by each individual table

Functions:

getViewedTablesCombs(): Provides the number of times a combination of tables has been viewed per user

getUtilities(): Calculates the utility per combination of data tables as shown in subsection 6.3

getCosts(): Calculates the costs per table as described in subsection 6.5

AuctionMechanism(): Executes the auction algorithm as shown in subsection 6.7

InfluencePresentedTables(): Influences which tables are presented based on the profits or revenues as shown in subsection 6.9

```
Algorithm:
NumberOfViewsPerTableComb = getViewedTablesCombs()
TableCombsUtil = getUtilities(NumberOfViewsPerTableComb)
TableCosts =  getCosts()
ProfitsPerTable = AuctionMechanism(TableCombsUtil, TableCosts)
InfluencePresentedTables(ProfitsPerTable)
```

6.3 Utility function

The previous section showed the overall functionality of RecorDa market approach component. This section shows the first step: the utility functions.

The RecorDa market approach component requires an evaluation of the datasets in order to determine their relevancy. It has to transform the data presented from the recommender system component into a relevancy created by this data. Utility functions are used in market approaches to identify the relevance of a specific product for a user.

In this thesis, a ‘pay-per-use’ utility function is used, which is based on the number of views that a dataset receives. Research has found a series of measures linking characteristics of a dataset and its usage to the relevance of the data [200]. Usage has a strong connection to file relevance; in that Wijnhoven et al. [89] state: ‘We found that the perceived frequency of use and user grade determine file use value.’ [89]. Wijnhoven et al. [89] include a table (see Table 10) describing different methods for deciding which files to keep (hence files that are relevant to the company), which demonstrate the significance of usage for data valuation. Number of use is the most prominent surrogate for values in a series of studies [89]. This makes it a good initial selection for a utility function.
While views do not necessary equal use it is assumed to be the case within this thesis. The reason is that the RecorDa approach learns which dataset the user likes to see based on the ratings provided from the users. The user is therefore presented with the most relevant datasets. Given that the most relevant datasets will most likely impact the user’s decisions they are therefore also most likely to be used by the user. Number of views is therefore very similar to usage within the RecorDa approach due to its capability to learn user interests.
This thesis therefore focuses on number of times a dataset is viewed as the only element of metadata to evaluate combinations of datasets due to the initial nature of the present research.

This evaluation approach is based on the fact that the number of data items that a user utilises is limited. Each user can only comprehend so much data. Various studies indicate that humans can only remember around seven things at a time [201]. Future work could extend the list of valuation criteria including some of the methods (i.e. time) mentioned in Table 10.

Method	Goal of data retention policy	Important file attributes
[Chen 2005]	Capture the changing nature of file value throughout the lifecycle and present the differences in values among different files	Frequency of use; recency of use
[Turcczyk et al. 2007]	Determine the probability of the future use of files to store them in the most cost-effective location	Time since last access; age of file; number of access; file type
[Bhagwan et al. 2005]	Lay out storage system mechanisms that can ensure high performance and availability	Frequency of use
[Verma et al. 2005]	Optimise storage allocation based on policies	Frequency of use; file type
[Mesnier et al. 2004]	Automatically classify and predict the properties of files as they are created	Frequency of use; file type; access mode
[Zadok et al. 2004]	Select files that can be compressed to reduce storage consumption	Directory; File name; user; application
[Strange 1992]	Optimise storage in a hierarchical storage management (HSM) solution	Least recently used
[Gibson and Miller 1999]	Reduce storage consumption on primary storage location	Time since last access
[Shah et al. 2006]	Design a data placement plan that provides cost benefits while allowing efficient access to all important data	Metadata; user input; policies

Table 10: ‘File Retention Policy Determination Methods’ by Wijnhoven et al. [89]

Because it is using the number of views as a proxy for utility this thesis adopts a cardinal utility function [202]–[205] and assigns its utility purely on its only criteria usage.

The pay-per-use utility function works by giving each user a fixed utility budget which is allocated towards dataset combinations based on the number of times a dataset is viewed. Whenever the user is presented with a combination of datasets, the presented data combination receives part of the budget (see Figure 8). This allocation mechanism is in line with typical cardinal utility functions [202]–[205]. A cardinal utility function was selected due to its underlying measure, the number of views, being on a cardinal scale and hence making a cardinal utility measure feasible based on a direct linear relationship between utility and number of view. In addition, a cardinal utility function has desirable properties such as adaptability among various users [202]–[205]

.

Overall, the utility function works as shown in Algorithm 6.2.

Algorithm 6.2 for getUtilities(): Utility function

```
Pre-set values from Administrator:
B(u): Budget for user u
U: List of users

Important functions:
```

```
views(u,c): Outputs the number of times user u has looked at data
combination c. (This data could be collected by logging the use of
data from the user.)
datacombos(u): Outputs the set of data combinations user u viewed.

Relevancy calculation:
For each user u in U:
     For each c in datacombos(u):
          M(u,c) = 0
For each user u in U:
     view_count = 0
     For each c in datacombos(u):
          view_count = view_count + views(u,c)
     For each c in datacombos(u):
          M(u,c) = (B(U) / view_count) * views(u,c)

Output:
M(u,c) = Map of values of user u for a combination of datasets c
```

The budget represents an overall amount of relevance perception that a user assigns to combinations of datasets based on the number of times they are viewed. It represents the amount of money a person can spend within a market. These relevance values are then fed into the value map (see next subsection). This function also allows RecorDa market approach component to be continuously updated. The values assigned with the budgets will change as the user's usage of data changes, hence allowing a continuing improvement process.

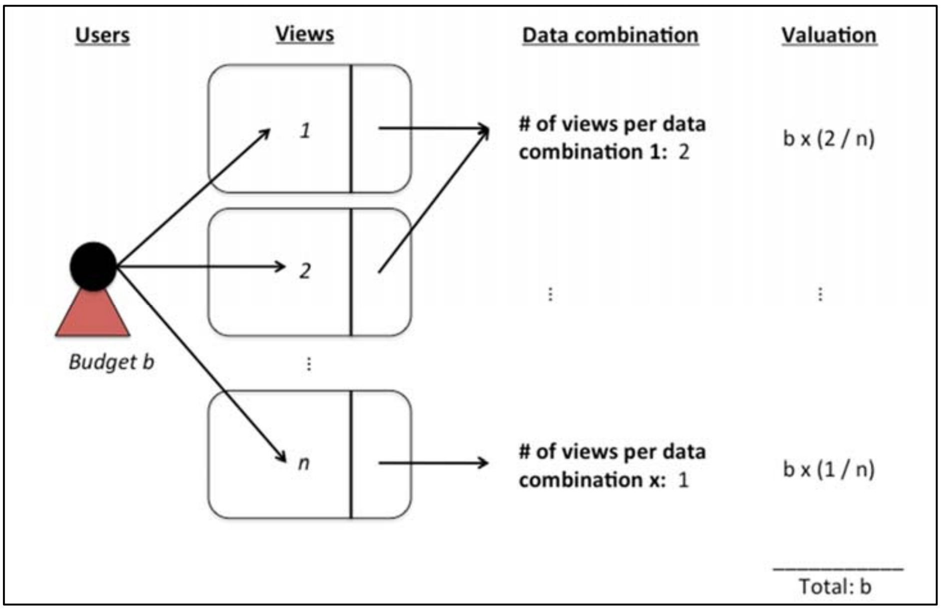
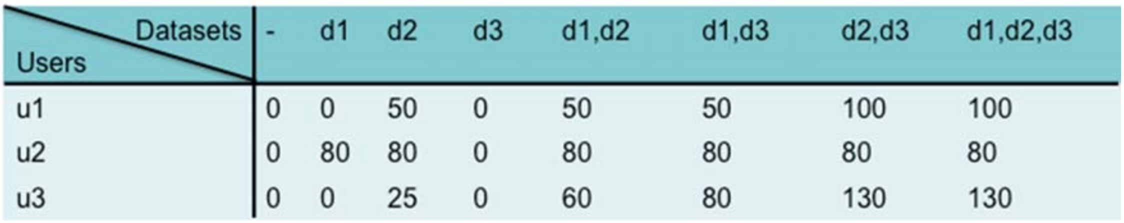


Figure 8: Illustration of the data combination evaluation process for one user

6.4 The Value Map

After identifying the potential of utility functions this subsection addresses the use of the results from the utility function.

The output of the functions is a relevance value for a combination of datasets for a specific user. The utility function evaluates dataset combinations, when they are presented to the user. The distribution of the budget is relative to the user's number of views of different dataset combinations. The output of all of the utility functions can be combined in a table format consisting of the different combinations of datasets in one axis and the different users along another axis (see Figure 9 for an example).



Datasets / Users	-	d1	d2	d3	d1,d2	d1,d3	d2,d3	d1,d2,d3
u1	0	0	50	0	50	50	100	100
u2	0	80	80	0	80	80	80	80
u3	0	0	25	0	60	80	130	130

Figure 9: An example of relevance allocations for all users regarding different combinations of datasets

This Value Map contains all possible combinations of the different datasets that a user can potentially use for a decision-making problem, along with the valuation that the utility function has found for each data combination. The benefit of the Value Map is that it can be continuously updated based on new incoming evaluations. However, it runs the risk of fast growth.

Growth with more users

The Value Map grows with the number of users. The number of rows represents the number of users, meaning that it increases linearly with the number of users. However, this thesis assumes that if an employee makes decisions in a group with other employees, this is a different user than when the user makes a decision independently. This means that the number of users could potentially increase exponentially with the number of employees in a company. Given that the number of personal combined decision-making opportunities for a human is limited, however, and because this thesis focuses on datasets and their evaluation instead of employee interactions and group decision-making, this issue will not be further addressed.

Growth with more datasets

The number of dataset combinations in this Value Map has the size

$$number\ of\ dataset\ combinations = 2^{\#\ of\ datasets}$$

including the case in which none of these dataset combinations is used. Precisely as Avasarala et al. [87] observe different combinations of sensors, this can lead to a dramatically growing number of dataset combinations for each user.

It is important to note that the utility functions do not necessarily provide relevance values for a single data source. This thesis is therefore not specifically able to take the valuation for data combinations and allocate them to a separate dataset. Some of these datasets might be supplementary to each other, meaning that one dataset can be used instead of another one without causing a significant increase in user utility. A typical example of supplementary datasets would be to rely on a Bloomberg financial dataset [206] instead of one internally generated by one's own finance division. Moreover datasets may also be complementary, which is the case if two datasets obtain a higher relevance value when they are combined. This could be the case when the list of suppliers is combined with an external dataset containing details about this supplier's solvency. When combined, the user is able to make better decisions about the suppliers, whereas each one on its own does not provide any additional relevance.

These examples of complementary and supplementary dataset values are just two examples of various combinations of valuations of a combination of datasets. It should be noted that these types of combinations of valuations for a combination of products are often seen in market situations. Identifying which products truly create value for a company means combining these values with the costs of obtaining these datasets and identifying the ideal combination. This type of calculation can be complex, and will be introduced in detail in subsection 6.6, 6.7, 5.8, and 6.9.

6.5 The costs of data

Beside the utility function and their use in the value map covered in the previous subsection, section 6.2 also introduced the importance of costs for market approaches.

As shown as a second key functionality this thesis uses interviews to identify the costs (see subsection 4.3.2). It will examine the following costs for dataset allocation:

- Maintenance costs: Internal costs for ensuring that the data remains available to the user in its current form.
- Development costs: Internal costs for ensuring that the data will be available in a different form in the future.
- Subscription costs: Payments to external parties for use of a dataset.

- Opportunity costs: Costs for the existence of the dataset in the system. Using one dataset might mean that this dataset takes the spot of another more relevant dataset. Opportunity costs capture this issue.

The costs are considered with a future perspective, ignoring sunk costs, meaning that if a dataset created costs for allocation in the past, but will not in the future, then its costs are 0. However, if there is a large investment to be made to acquire this dataset, then the costs will be included.

All of these costs can be estimated using established approaches for cost estimation in software project management and with questions of experienced experts. The opportunity costs are set as a low fixed value initially, but further research could use more elaborate techniques to identify them.

6.6 The data allocation problem

The previous three subsections covered utility and costs of datasets. This section will address how these are combined to ensure the most relevant data is presented to a user.

The problem of identifying which datasets should be presented to the user requires identifying individual dataset valuations. Therefore, a breakdown of valuations from the Value Map, which are based on dataset combinations as well as individual datasets, is needed. Section 6.4 showed that it can be difficult to allocate relevance values to individual datasets even for a small number of datasets. There are a large variety of evaluations for each dataset, depending on the comparison to the other datasets. Each evaluation is potentially valid, making it difficult to decide which one to use for further calculation.

Besides these computational problems, there are additional issues with the Value Map based on the recommender system and the user's budgets:

A. **It is continuously updated:** The values in the Value Map are continuously changing whenever the user looks at different datasets, gives different feedback for recommendations, or changes the user's preferences.
B. **It is incomplete**. The Value Map does not contain valuations for all individual users and all dataset combinations, because not all of these are shown to the user.

The algorithm to address this problem therefore needs to fulfil additional requirements (see Table 11).

Limitation	Requirements for algorithm solving the Value Map evaluation
A. The Value Map is continuously updated.	Due to the continuous updating, the algorithm needs to be able to handle this additional information while still computing relatively accurate results without a complete restart of the whole calculation process.
	Due to the continuous updating of the Value Map, an algorithm needs to update its calculation relatively quickly within a couple of hours. It cannot take weeks or months to be completed. However, the processing is still independent of the Value Map creation, so it does not require an update within seconds.
B. The Value Map is incomplete.	The algorithms cannot rely on relative comparison of all kinds of dataset combinations because these are not always available.

Table 11: Describing the impact that the limitations of the Value Map have on algorithms using the Value Map for individual dataset evaluations

This thesis proposes the use of market approaches, introduced in the previous chapters, to overcome this problem and find individual dataset evaluations[16]. While potentially offering some time improvements, market approaches focus especially on the benefit of not having to do individual comparison and of being sensitive to new incoming data without requiring that the whole calculation be redone.

6.7 Market approaches for solving the data allocation problem

The previous sections discussed the difficulty of finding a good solution for the individual dataset evaluations from the Value Map with dataset combinations. The literature review showed that market approaches could help to solve these types of problems.

The market approach component of the RecorDa approach needs to manage the interactions between costs for individual datasets and the utility of combinations of datasets used by the user.

The challenge of evaluating dataset combinations covered in the Value Map need to be broken up for individual datasets. The market-based algorithm needs to find the price that each individual dataset contributes to the different combinations of each user. This price represents the relevance that this dataset provides, and can be compared to the costs of offering this dataset. Market approaches use auctions for this challenge of price determination. An auction is based on two types of participants: buyers and sellers [109], [112]. The buyers are interested in acquiring a product and have a specific utility (or value) for this product. Sellers are interested in selling a product (in this case data) for as much as possible to cover their costs.

[16] Which are often required to make specific data management decisions and select which datasets to present.

Buyers continuously look for other options that they are interested in buying, and sellers continuously look for other people to whom they can sell their product. One can transfer the problems described within the Value Map to a market approach problem, where auctions manage these transactions. This gives the benefit of faster calculation times [63].

Data sources want to sell data, and the users are interested in buying data. Therefore, the following types of sellers and buyers are used for the auction mechanism.

Buyers
The data buyer is the user. The data buyer's evaluation of a data combination is given by a utility function, which influences the data buyer's willingness to pay for a specific data combination. The buyer will participate in a variety of auctions to find the highest gain in utility given the available data combinations. Data buyers will pay based on the individual asking price of the data sellers.

Seller
The data seller is interested in selling data in order to best cover the cost associated with offering this dataset. Data sellers are individual datasets. They have to sell themselves to the data buyers (the users) by bidding for each individual user. The data seller tries to maximize its revenue in order to obtain a high relevance level for its data. If the revenue or income of a data seller is lower than the costs, the seller will be put out of business and will no longer be part of the market.

The data buyer therefore evaluates combinations of datasets, while the seller tries to sell them individually to each data buyer. The difficulty is that the user buys a combination of datasets to use for decision-making. The auction mechanisms deal with situations in which a buyer evaluates and ultimately selects a combination of individual products (in this case datasets), while the data seller only sells individual product.

The following algorithms describe how data sellers and data buyers operate within this market approach component (see Algorithm 6.3 and Algorithm 6.4). It runs a series of iterations until the price for all datasets is no longer changing or until a fixed iteration cut-off of 1000 iterations is reached.

Algorithm 6.3 AuctionMechanism(): Data buyer's algorithm for evaluating price offers from the auction

```
Values given at start of market
V: Value of data combination
DC: Dataset combination for this seller

Variables
```

P: Total price

AP: Auction prices from sellers

d and z: Generic variables for datasets

pd: Price for dataset

Key functions

Auction(): Obtains a set of all auction prices for all datasets

Get_Price(d,AC): Obtains price for dataset d from auction prices AC

Add_Buyer(z,b): Adds buyer b to seller z

Calculation at each market iteration

```
For each auction iteration
      P=0
      AP=Auction()
      For each dataset d in DC
            pd=Get_Price(d,AP)
            P=P+pd
      If P<V
            For each dataset z in DC
                  Add_Buyer(z,this)
```

Algorithm 6.4 AuctionMechanism(): Data seller's algorithms for deciding whether a certain price and number of data buyers accepting this price would make the data buyer profitable

Values given at start of market

C: Costs for datasets

B: List of buyers as dataset combinations

Variables

SP: Price(s) set by the auction mechanism for the dataset

DCB: Dataset combinations and their willingness to buy certain datasets

R: Revenue for dataset

Profit: Profit that this dataset (or seller) generates

Key functions

Get_Price(): Obtains price for this dataset set by the auction mechanism

Buyers_Decision(SP): Returns the results from the different dataset combinations on their buying decision given specific prices SP (see Algorithm 6.3 for further details)

Buys(b,DCB,s): Returns true if the buyer b buys from this seller in the DCB; otherwise it returns false

Calculation at each market iteration

```
For each auction iteration
      SP=Get_Price()
      R=0
      DCB=Buyers_Decision(SP)
      For each buyer b in B
            If Buys(b,DCB)
                  R=R+SP(b)
      Profit=R-C
```

Both the data buyer and data seller receive price offers from the market maker, who executes the auction mechanism. The data buyer determines the interest in purchasing the dataset for this price, while the data seller calculates the data seller's profitability and evaluates whether the data buyer should stay in the market.

The following section will go into further details on the auction mechanism mentioned in the algorithms for the data buyer and data seller.

6.8 Auction mechanisms

In order to manage the price negotiations between users and combinations of datasets, market approaches use auctions the third key functionality outlined in chapter 4. As described in chapter 4 the market component uses a market maker who manages the price setting. The market maker takes price offerings from the different datasets and checks which dataset combinations are able to pay for their dataset combination (or have the relevance allocated from the user via the utility function). If a data combination does not have the resources to pay for the prices demanded by the dataset, it drops out of the bidding process. The algorithm overall tries to increase the prices that data combinations are willing to pay for a dataset stops if the revenue for a dataset starts to decrease.

As briefly shown in subsection 4.3.2 there are a series of auction mechanisms that can generally be used. The most commonly used auctions are English and Dutch auctions.

- In **English auctions,** [109] a price is continuously raised until only one bidder is still willing to pay the next higher price [109].
- In **Dutch auctions,** [109] a high price that none of the bidders is willing to pay is initially set and continuously lowered until the revenue of the bidder no longer increases.

Further details can be found in Chapter 2 and various research papers [109], [111], [112]. The RecorDa approach uses these well-established forms of auctions. Further research might use different auction types.

The English auction is based on a continuous increase of the price for a certain dataset. Then the data buyers decide whether they want to purchase the product for this price, and the data seller knows how much revenue and profit or loss is made depending on the number of buyers willing to pay the price. The auction is an adaption of the English Auction [109] for this particular purpose of dealing with datasets.

Besides the general auction, an alternative approach uses price discrimination in which the price is set differently for each of the users until a price point is reached that this particular dataset is no longer willing to pay. This might be unfair to the user, but potentially increases the revenue generated by the dataset and therefore the overall outcome of the market.

Unlike English auctions, Dutch auctions rely on a continuous reduction of the price in every iteration. The data buyers decide whether they want to buy the product, depending on the offered price, as they did in the English auction. In the same way, the data sellers also calculate their revenue based on the number of data buyers willing to purchase the dataset for a given price. The algorithm stops when all data combinations together offer the less total revenue for a dataset.

The auction is an adaption of the Dutch Auction [109] for this particular purpose of dealing with datasets.

6.9 Influencing the recommender system

After evaluating the relevancy of the different datasets individually, this section focuses on how this impacts the presentation of additional relevant datasets. The aim of the RecorDa approach is to provide the user with more relevant data. Therefore, after the datasets have been evaluated and the profit or loss of each specific dataset has been identified, this evaluation needs to affect which data is presented to the user as described in chapter 4 as the fourth key functionality.

Overall, the evaluations in the form of profit or revenue of individual datasets (as discussed on chapter) can impact the presented data in two different ways.

1. **They can impact which data is not shown.** This means that non-profitable data is not presented to the user.
2. **They can impact in which order the data is presented or which confidence is used to order the data**. This means that the order in which the data is presented to the user is impacted by the profit or loss of a certain dataset.

For the first types of impact, all data that has a negative profit is no longer shown to the user regardless of its specific relevance. For the second method of impact, the data is ranked using the recommender system component and the market approach component ranking, or its evaluation score. Then both of these rankings or evaluation scores are combined into a new ranking with a specific weighting.

```
Final_Ranking = weighting_Factor * MarketApproach_Based_Ranking + (1-weighting_Factor) * Recommender_Ranking
```

As shown in chapter 4 the ranking can be done based on the overall profit/loss of the datasets or the revenue generated from a specific user in the case where the price discrimination method was used. Both of these impact types and their influence on the presented datasets are evaluated in chapter 7.

6.10 Summary

This chapter showed how market approaches can be adjusted to overcome the problem of data allocation when combined with recommender systems.

The market approach component uses the input from the recommender system to evaluate the relative relevance of datasets using a utility function.

This research uses a utility function which assigns each dataset combination a valuation based on its usage. Costs are estimated based on expert interviews. An auction algorithm is used to combine utility and costs. The value map forms the basis for aggregating the dataset combination valuations and costs per user. Based on this market-based evaluation with auctions the datasets are then ranked to influence the recommendations from the recommender system. This way the market component to the RecorDa approach influences the data that is presented to the user.

7. Evaluation

7.1 Introduction

The previous two chapters addressed the first research question; this chapter will now focus on testing the main hypothesis of this thesis; that *based on the characteristics identified in chapter 2, recommender systems and market approaches can be used to identify the relevant data for users in a company and increase the amount of relevant data allocated to the user while reducing the problem of data overload*.
This will be tested by answering the second research question: *Can recommender system and market approach individually or in combination identify relevant data better than potential alternative techniques?*

As outlined in Chapter 3, this thesis will answer this question by using two methods: experiments and case studies.

This chapter tests whether the specific architecture of the RecorDa approach is able to provide more relevant data compared to alternative techniques. In order to achieve this goal, section 7.2 will first discuss how relevance of data is measured for both case studies and experiments. Section 7.3 will then introduce alternative methods to which recommender systems and market approaches will be compared. Following this section, section 7.4 will first introduce the experiments in more detail, while section 7.5 will do the same for the two case studies. Section 7.6 will then discuss the setup times for the different approaches. Finally, results both from the experiments and the case studies will be summarised in section 7.7.

7.2 Evaluation measures

This thesis has already introduced different types of data, some of which is relevant and some of which is not (see Table 2 in Chapter 1). It has also demonstrated that these different types of data need different methods of evaluation. Recommender systems have similar evaluation challenges and the existing work in this area can be used for this thesis's evaluation.

Based on the structure of known and unknown data introduced in Chapter 1 the following subchapters will describe their evaluation in more detail.

7.2.1 Evaluation of known data

For known data, this chapter investigates two types of measures: categorisation, and rank accuracy [207], [208] to evaluate the allocation of relevant data.

- **Categorisation**: Evaluates how many items are in the right category of recommended items and non-recommended items. Typical measures include precision and recall.
- **Rank accuracy**: Evaluates whether the recommendations are given in the right order. Typical measures include Spearman rank correlation.

Precision measures if the user actually gets relevant data and how much irrelevant data a user is still presented with. Recall measures what share of datasets that a user sees that are not relevant for him. In addition, rank accuracy tests how good an approach is in sorting the data to match user preferences.

A measure often used to evaluate categorisation accuracy in precision and recall is the F-Measure. However, this thesis is not adopting F-Measure. F-Measure compute the harmonic mean between precision and recall. However, this means it is biased against extreme values in either one of them [209] and prefers both measure to be sufficiently large in order to cause a good F-Measure. The harmonic mean shows a bad score when precision or recall take on extreme measures (e.g. Precision 0.9 and Recall 0.01). These extreme measures could still be desirable properties in terms of data allocation, especially in cases of a large number of relevant datasets that go beyond the user capability to assess them. The user might not be able to comprehend all the relevant datasets due to limited time [201]. In this situation F-Measure would not show good performance, because it biases against extreme value. Therefore, this thesis is using precision and recall individually to adjust for this sort of situations and identify the most suitable combination of precision and recall based on its individual values.
In addition, considering precision and recall individually allows considering them in combination with other variables, i.e. precision for rows as the precision for recall on a row level cannot be computed due to the computational complexity[17].

7.2.2 Evaluation of unknown data

The second type, organisationally unknown data, is more difficult to evaluate, because nobody in the organisation would know which data that is. It is not possible to check whether the 'relevant' data is presented to the user. For example, a certain dataset bought from an external data supplier might have a few additional relevant data points without anyone in the organisation knowing about them. For recommendations (or additional relevant datasets) that are completely unknown, the evaluation relies on proxy values. They measure whether the system would be able to present all the data to the user and give the user the opportunity to evaluate it. Hence, an important indirect criterion of relevance is that all datasets are

[17] Calculating the recall for rows would require going through all rows of all potentially available datasets and identify which rows are relevant. This would be too complex for a large number of combinations.

presented to the user. Within recommender systems this is typically measured with coverage and novelty [208], [210].

- **Coverage**: Describes which percentage of the existing recommendation items have been presented to a user.
- **Novelty**: Describes which percentage of datasets in a given recommendation a user has not seen beforehand.

A detailed definition of the measures can be found in Attachment D.

The additional benefit of these measures is that they also capture an aspect of robustness and flexibility in the user's understanding of relevant data. If these datasets or the user's interests change, a high coverage and novelty would ensure that the user is presented with new data for changing requirements. However, high measures for coverage and novelty also have the disadvantage that whenever a novel piece of data is presented to the user, a more relevant piece of data categorised as valuable is not presented to the user.

7.2.3 Evaluation of computation time

Beside these issues, there are a series of measures that capture important technical and economic aspects related to the company environment in which the system will operate. Within software, this is usually cost of development and computation time.

In this study, computation time was measured with java implementations on the same computer. A record was made of the time it took the compared approaches to supply the additional data to the user for different requests. A Mac Book Pro with 2.4 GHz Intel Core 2 Duo was used for the computation time evaluation together with an Open CSV library [211] to conduct the search through the csv files given the presented terms that were potentially relevant for the user. A separate subchapter will look into the setup times.

7.3 Methods for comparison

Section 2.3 identified three main alternative types of methods for providing the user with additional relevant data. These three types of methods are:

1. Search: Comparing the syntax and semantics of an entered search term against the content of various databases.
2. Requirement analysis: Analysing the data of interest to the user by identifying the user's specific requirements.

3. Decision theory or VoI based techniques: Analysing the specific decision of a user, identifying the impact of data on this decision, and thereby identifying the relevance that a specific dataset has for a user.

The implementation of the comparison of each of these methods will be introduced in the following sections.

7.3.1 Search

Based on the type of search discussed in section 2.3.2, three types of different user search behaviour were assumed for the evaluation:

- Directed search: The user knows directly where to find data. The user knows which table to search. For example, if the user looks for data about a certain known employee, the user can find this data in the Human Resources database and will search for this user directly in this database.
- Undirected search: Here the user does not know where to look for additional data, or whether additional data in fact exists.
- Learning search: This is a combination of the two search types above, and represents the behaviour of a new user. The user starts with an undirected search. If the user obtains the required result in the top search results, the user will use this type of search more often. If the user does not find the required result in the top 10 results, the user will stop searching for this term. If the user finds the result in a table, the user will always look in this specific table.

This research does not focus on different syntactic or semantic differences during search and identifying these, because there are already solutions dealing with this type of problem. Therefore, the search (as well as the recommender system) will only find direct syntactic overlaps and present these to the user.

7.3.2 Requirement analysis

Requirement analysis is based on asking users for their interests or identifying the users' interest by asking experts in the users' domain about the latter's interests. This is typically done via intensive interactions with the user using surveys or interviews [212]. These methods often use a series of tools such as whiteboards, brainstorming techniques, or prototyping [212]. There are various approaches to identifying and capturing the user requirements for further development, such as model-driven requirements engineering [212] or business process modelling [213] .The main issues with these techniques are that:

1. They require someone in the company to specifically know the exact user requirements.
2. They cannot deal with changing requirements from the user without having to repeat the process.
3. They also require the specific implementation of delivering specific data to specific users, which requires the availability of developer resources.

These are limitations that require a large and constant effort to ensure that the current information system always provides the most current data to the user.
On precision and recall, this approach performs well and reaches 100% when executed correctly. It captures knowledge regarding the current user's interests and ensures that these are presented to each user, similarly to the analysis conducted to find a comparison for this evaluation (see Attachment A and B). However, requirement analysis does poorly with regard to novelty. Coverage is high for a one-off analysis. However, it changes after the analysis is completed. Updating the analysis can be a very complex process which can be similar in effort to the initial analysis. It does not discover 'organisationally unaware data' (see Table 2) because it can only capture the current organisational requirements. This makes requirement analysis much less adaptable and flexible with regard to data allocation. The issue it that it is too complex to be updated all the time.

7.3.3 Decision-theory-based techniques

Chapter 2 presented an overview of different decision-theory-based techniques. Their main current limitation is their lack of scalability to a larger number of users and larger number of datasets. It is therefore not feasible to conduct a detailed analysis of data relevancy using these techniques. In addition, there are a large variety of configurations. Executing this analysis would be highly complicated with a large set of users. Assuming it would be feasible, however, it would result in the ideal allocation and identification of the different datasets for each user, and in a perfect precision to recall ratio as well as a perfect rank accuracy. Coverage would be high for the one-off analysis, but similarly to requirement analysis would become less accurate over time with evolving data needs of users.

The variable coverage would be at 100% because all datasets would have been considered. However, the measures for novelty would be not applicable because all of the datasets would be analysed as part of this hypothetical ideal VoI analysis, making this approach highly inflexible to changes in the required data.

7.4 Experimental evaluation

As an initial analysis, a series of experiments were conducted in this study. Due to the large number of variables in the configuration of RecorDa (standalone recommender system and with market approach component), different experiments were introduced to find a more suitable set of variables that could be applied within the case study settings that will be presented in section 7.5. The purpose of these experiments was to narrow the scope of ideal settings to then test them within the case studies.

Three main requirements needed to be met by the different experiments:

A. Industrial relevance: The experiments needed to represent industrial environments in order to show that this approach works within industrial environments.
B. Test data allocation: The experiments had to enable the collection of the variables outlined in section 8.2 to collect the values for the evaluation.
C. Repeatability: Due to the large number of configurations and components for the RecorDa approach (see Chapters 5 and 6) it had to be possible to run a large number of these experiments repeatedly.

To ensure that the first criterion was fulfilled, the experiments were designed with industrial experts from a large manufacturing company. The detailed method is outlined in section 7.4.1 to address requirement A. For each experiment, data was collected about:

- **Users**: Details on the users involved in the decision-making process. The task of the user and which type of decision the user is supposed to take.
- **Databases**: Details on the databases that could potentially be presented to the user. Detailed content in the form of the different tables, columns and the type of data that they contain, including examples for each of the different columns.
- **User's current data**: Details on the data presented to each separate user. This only contains the data that the user is currently seeing, but not additional datasets that could be relevant.
- **Additional relevant data for the user**: Details on which data (besides the data currently presented to each user) would also be of interest for the user. This includes the priority order of this additional relevant data. It describes which additional datasets the user should be seeing according to domain experts, but that is not currently being presented to that user. The user either needs to search for it in other systems, or is not aware that the data exists within the company. Ideally, the information system would provide some of this data to the user, however due to limitations of current systems this is not always possible.

The experiments were also based on three different experimental environments, which will be introduced in section 7.4.2.

Requirement B was fulfilled by collecting details about the data in which the users were interested, as well as the available datasets. This made the experiments repeatable (requirement C) because user requirements are known by system experts and it is possible to verify whether this data is in fact allocated to the user without requiring user input. However, there are various key variables involved in the user decision-making as well as the RecorDa approach component setup, which will be introduced in section 7.4.3. The results of the different experiments are presented in section 7.4.4.

7.4.1 Experimental design method

The experiments presented in this chapter are based on realistic industrial situations. The data and tables in the experiments, were datasets automatically generated according to the information collected while working through iterations with industrial experts. The industrial experts were from the research and development field and focused on IT projects within a large industrial manufacturing company. These experts provided a broader understanding of the complex data situation and all available datasets.

For the detailed experiment design and scenario understanding, the approach outlined by Jess et al. [214] (see Table 12) was followed. The approach was successfully used for a similar problem, in which the challenge was to develop experimental data for a recommender system. It was also specifically designed for cases in which direct access to data from the company was not possible [214], as was the case for these experiments. A detailed description about the execution for each of these steps can be found in Table 12.

Step	Description	Application for these experiments
1 Identify contacts	'The key contacts help in providing the additional information for following steps. They need to have a broad understanding of the company to provide the access for understanding the industrial data management problems or provide links to the right contacts in the organization. For data management in industry these could often be IT or Research divisions. But dependent on the problem and the organisation the criteria for the key contacts can vary' [214].	Based on previous work with the manufacturing company, a key contact in the company's R&D division was identified. The contact knew the main experts in the different areas in which data allocation could be improved and had previously worked with university research. The contact therefore had a good understanding to provide the required information and contacts internally for the experiments, while being able to understand the needs of university research.

2 Problem domain	‘Using the contact(s) identified in step 1 the general questions about the problem domain need to be understood and clarified—what are the operational goals that the company wants to achieve and how does their data problem affect these goals?’ [214]	During a series of around 12 personal meetings and monthly phone calls for three years, as part of a separate research project with the same company, areas in which data allocation was an issue were identified. Using this understanding, three separate industrial experimental scenarios were identified and further detailed in two phone calls of about one hour each with the industrial experts.
3 Problem characteristics	‘The key problem characteristics vary based on the industrial data management problem. These characteristics can be column headings, specific data types, data profile, error types, number of data occurrences and so on. They need to be sufficient to accurately translate the underlying problem into a mock-up dataset. These characteristics are the basis for the target environment and the mock data generation in the following chapters’ [214].	Following these initial interviews, a series of eight interviews were conducted with the industrial expert to detail the problem characteristics. These interviews were held with different experts within the company. The industrial expert also gathered further internal feedback from domain experts focusing on the characteristics of table structure, data types, data format, data volume, and user behaviour from Jess et al. [214].
4 Data environment	‘The ideal representative environment is defined based on the understanding of the domain of the data, and its key data characteristics’ [214]. A detailed table with these characteristics can be found in Jess et al. [214].	Based on the interviews and a continuous feedback loop with monthly calls, a set of tables, their content, and structure were specified to develop a mock-up environment for the experiments.
5 Mock-up	‘Based on the target environment the researcher can now develop a mock-up data environment to “simulate” the problem existing in the industrial company they are working with’ [214].	An automatic data generator to generate the previously identified datasets was developed using a combination of public datasets, random data generation, knowledge about business logic and links between data, and lists of expected values for certain fields. It automatically generated the table structure for the three experimental

		settings. The data generator ensured that a representative volume of data was generated and that it was consistent across different tables (e.g. matching of primary and foreign keys). These were the key requirements for designing a representative data environment.
6 Validate	'After generating the data environment it should be shared with the key contacts, so that they can evaluate whether it is a suitable and correct representation of the real problem' [214].	All automatically generated datasets were shared with the key contacts. Five additional interviews with domain experts in these areas further verified the collected datasets and whether they were representative of the companies' datasets for the purpose of analysis.
7 Solve	'In this step the researcher solves the actual problem utilising the data environment' [214].	The solution to the specific problem in the recommender system component and the market approach component are shown in Chapters 5 and 6 of this thesis. The solution was then tested on these environments (see sections 7.4.2-7.4.4).
8 Feedback	'This step applies the developed method or tool in the industrial company. The key contacts and their contacts in the company will test the solution under the guidance of the researcher. They can then provide feedback for the identified tool' [214].	Privacy concerns and legal restrictions did not permit the testing of the approach within the company's IT infrastructure. However, the presented case studies and their results in section 7.5 provide an idea of the potential of using these applications.

Table 12: Experimental design steps from Jess et al. [214] and an analysis for the three experiments

The understanding of the industrial situation was presented to the industrial experts after each step to verify it. Based on the feedback of the industrial experts, the output was adjusted in an iterative manner until a common understanding was reached to create a more realistic experiment. Each of these experiments was captured in an experiment description sheet (See Attachment A).

7.4.2 Experimental environments

The experiments and their development were described in the previous section. They addressed different problems within an industrial company, such as procurement, production, and support. They also covered different tasks of industrial companies. In this way, this thesis ensures a broader applicability of RecorDa in industrial data allocation problems. The specific experiments settings were the following:

- Procurement: A large manufacturing company has various procurement challenges, such as performance-based logistics (PBL) contracts with the military or procurement for its own production. For both, part availability is critical. The payment of the company and the company's rate of production depends on the level of part availability that they can achieve. Various datasets are used to help with the procurement process. The question is how additional datasets could help improve the decisions of the different users involved (see below) and what the increased availability (and therefore financial benefit) of these datasets would be.
- Production: Within the production of a large manufacturing company various datasets are used from a variety of different manufacturing steps. In each of these steps, different datasets are required for the coordination of the production. Various issues need to be addressed on a continuous basis. Among these issues are the following: managing missing parts within the production process; managing machine failures; managing replacements of parts between different large manufacturing parts; and managing specific requirements for each individual large manufacturing part. Addressing each one requires a variety of datasets, and addressing all of them requires an even greater variety. Identifying the relevant data for the users managing the production process is an important task.
- Support: A large manufacturing company works with various customers who require support for their products. This support is mainly in repairs, the management of spare parts in warehouses, and the management of personnel for this support. In order to manage this support in the most effective way, the users could leverage various internal and external datasets. Identifying the relevant data to use for this problem is a challenge for them.

The completed experiment description sheets for all three experiments can be found in Attachment A. They provide a detailed overview of each experiment.

For each environment, the applicability of the RecorDa approach was tested by asking a series of questions that generally identify whether it is within the spectrum of applications. This was done to ensure the validity of the experiment (see Table 13).

Does the dataset fulfil the following characteristics?	**Description**
User is using data from multiple (possibly changing) data sources	The RecorDa approach helps to identify which datasets are valuable for a user. Datasets will not provide any benefit if there are not a larger number of them potentially relevant for the user.
User has a set of offers (from data providers) to acquire more or different data to improve the user's decisions	If the user does not have to choose between different datasets that are potentially relevant, a mechanism that helps the user in selecting these datasets will not be beneficial
User knows the relevance of a certain piece of data or a combination of data pieces in terms of contribution to a decision	If the user cannot decide if a dataset provides a benefit once it has been presented to the user, the user cannot give good ratings, which is the main influence for the RecorDa approach
Data has costs associated with its allocation	If the costs of the dataset are unknown, it is difficult to determine whether it provides a positive benefit for the organisation
Partial information with data users and / or data providers	If the data provider is the data user, the user usually knows about all the datasets, thus taking away one of the main benefits of the two approaches, which is finding previously unknown information. These three criteria ensure that this is not the case.
Heterogeneous environment for data users and / or data providers	
Distributed decision-making between data user and data providers	

Table 13: Experiment validation questionnaire

7.4.3 Evaluation variables

There are a series of different variables to consider for the evaluation of the RecorDa approach. Analysing them is a key part of this chapter to ensure that a prime setup of RecorDa is selected for comparison to alternative solutions. They can be divided into two types of variables:

A. RecorDa with Market approach and standalone recommender system component variables: There are variables used for setting up the RecorDa approaches
B. Environmental measures: These are other external variables potentially influencing the outcome of the experiment due to different user reactions.

The internal measures are the following; they have already been outlined in Chapters 5 and 6. They can be divided between the different elements of the recommender system component and the elements of the market component.

Variables	Description	Variations[18]
Recommender system (functions and aggregation)		
Recommender system similarity functions (outlined as a key decision in chapter 4)	Method for comparing the different recommendations in the item and user recommender system. These recommendations rely on the input from the user.	The Mahout library contains the following standard methods: • Log Likelihood Similarity • City Block similarity • Euclidean Distance Similarity • Pearson Correlation Similarity • Spearman Correlation Similarity • Tanimoto Coefficient Similarity • Uncentered Cosine Similarity
Recommender aggregation method (outlined as a key decision in chapter 4)	Method for combining the recommendations from the different recommender systems	Min: Minimum of the three recommender systems Max: Maximum of the three recommender systems Avg: Average of the three recommender systems
Recommender system (number of recommendations)		
Number of recommendations per page	Number of recommendations that are presented to the user	Due to limited space and the fact that users can only comprehend limited number of datasets the following variations were selected: 2, 3, 5, and 7
Recommender system (thresholds)		
User recommender system threshold	Threshold for not considering recommendations and confidence values from the user recommender system for the further aggregation of results	The confidence values were usually in the range between 0 and 0.9; hence the following variables were chosen for further testing: 0; 0.2; 0.4; 0.8
Item recommender system threshold	Threshold for not considering	The confidence values were usually in the range between 0

[18] All variations were based on the different configuration of the RecorDa approach. The reason for these decisions can be found in chapter 5 and 6. There were two types of variations that had to be selected. For nominal values such as different aggregation methods this thesis shows all different combinations in the cases that presented themselves. For quantitative values such as thresholds this thesis took the spectrum of sensible values (e.g. thresholds can only be set between 0 and 1 if the value that has to pass the value is only between o and 1) and divided this spectrum into 4 to 5 intervals.

	recommendations and confidence values from the item recommender system for the further aggregation of results	and 0.8; therefore the following variables were chosen for further testing: 0; 0.2; 0.4; 0.6
Confidence threshold	Threshold for not considering recommendations and confidence values from the combined recommender system for the further aggregation of results	The confidence values were usually in the range between 0 and 1.0; therefore the following variables were chosen for further testing: 0; 0.25; 0.5; 0.75
Market approach analysis (auction method)		
Auction method (outlined as a key decision in chapter 4)	Auction method for identifying which is the best method for solving the Value Map	As introduced in Chapter 6, English and Dutch auctions are used
Price discrimination	Give each user its own individual price	For this case, the auction mechanisms can either be Discriminate or Not discriminate
Recommender system influenced by market approach		
Influence order or confidence evaluation (outlined as a key decision in chapter 4)	As introduced in Chapter 6, there are two ways in which the evaluation can influence the recommender system: it can change the order or the relative confidence	The two methods are therefore order, for changing the order; and confidence, for changing the confidence
Market weighting	Weighting between the recommender system and the market in selecting the order of the presented data	A broad range was chosen to find the amount of influence that the market should provide: 0; 0.25; 0.5; 0.75; 1
Remove unprofitable datasets	Selection if datasets that are found to be unprofitable by the market approach should still be presented to the user	The combination function has two options: 'Remove' or 'Do not remove'

Table 14: Different variation variables for the RecorDa approach with standalone recommender system and with market approach component

Environmental measures mainly concern user behaviour. The latter is the only external input critical for the evaluation. User experiments could be strongly influenced by variables such as user knowledge, requiring a large number of user interactions. Ranking behaviour as the main user input is therefore the main user variable to evaluate. There are two main variables influencing the user's ranking behaviour:

- Frequency: The number of times that a user provides a ranking when presented with recommendations
- Accuracy: Accuracy in the user's rankings

Therefore, in the experiments the following rating behaviours were used for different users (see Table 15) to test different variations in the way a user might interact with the RecorDa approach system.

Variable	**Description**	**Variations**
Rating frequency	Rating frequency describes how often the user ranks a dataset that is presented to the user. For example, does the user provide a ranking every time a dataset is shown, or does the user only do this, every other time?	The user gives recommendations for every 1, 3, 5, or 10 recommendation(s) presented
Accuracy – selecting the correct recommendations	Rating accuracy for selecting the correct recommendations describes how many and which of the presented recommendations the user is rating. For example, does the user provide ratings for all recommendations, just the top recommendations, or only the recommendations that have been correctly presented to the user?	Variations in the user behaviour for selecting the recommendations: • All recommendations • Top one recommendation • Top three recommendations • Correct recommendation • Wrong recommendation • Random recommendation • No recommendation
Accuracy – giving the correct rating	Rating accuracy describes how effective the user is in providing the correct types of rating for the different datasets in which the user should be interested. A user might not recognize data that is relevant. This variable tests whether the recommender system can deal with this type of user inaccuracy.	Types of accuracy the user can have in giving the correct rating: • Extreme • Strong • Medium • Neutral bias • Positive bias • Negative bias • Neutral

Table 15: Variations in user's rating behaviour

Due to the large variety in the variables and the time needed to run each experiment, the simplest, most intuitive, earliest, and most commonly used heuristics for optimisation of a large number of variables was used to identify the best combination: the greedy algorithm [215]. The greedy algorithm is further helpful as an approach to find a good combination of variables for RecorDa given the system-by-system encapsulation of its architecture, making it possible to draw conclusions regarding the functionality of each of these systems.

Each subsystem was improved separately, initially using optimal user behaviour. Potential behavioural variations were tested at the end, not in combination but each one separately, in order to identify the influence of each different user reaction. Overall, the following experiments were conducted using the following experimental scenarios.

1. Recommender system: The recommender system component is the main part of the RecorDa approach. It also forms the basis for the main utility input for the market component, and therefore required initial optimisation following the step-by-step system-wide optimisation approach.
2. Market approach analysis: Based on the recommender system following the step-by step variable optimisations, the market approach component needed to be improved next before the market's specific influence on the recommender system component output could be improved.
3. Recommender system influenced by market approach component: After optimising the market component, different types of its influence on the recommender systems were the next step in this optimisation approach.
4. User behaviour: After improving the system using ideal user behaviour, different types of user input were tested.

The overview of the experiments conducted on the architecture of the RecorDa approach can be found in Figure 10. It follows the key functional elements outlined in chapter 4. This figure shows how each component of the RecorDa approach was initially evaluated and compared to alternative techniques.

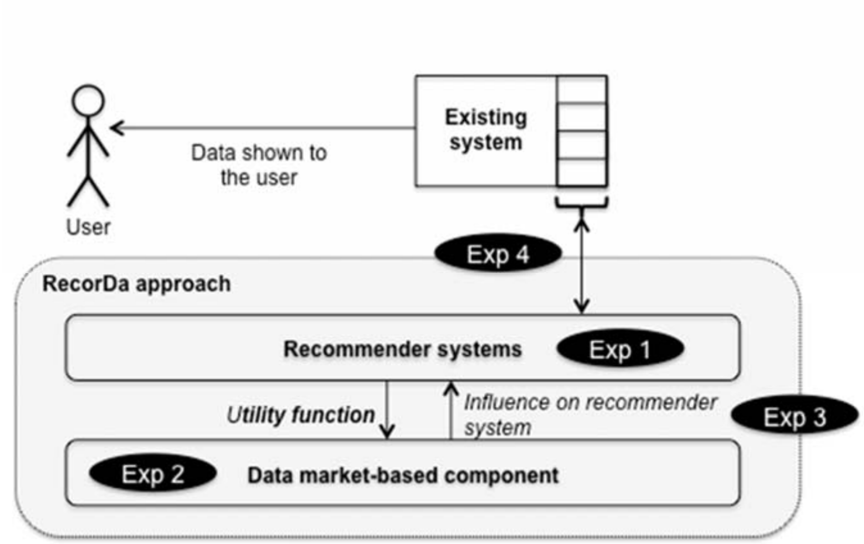


Figure 10: Description of the experiments along the flow of the RecorDa approach architecture

The evaluation results of these experiments will be discussed in the following section.

7.4.4 Experimental results – RecorDa approaches

The previous sections identified the main evaluation measurements (see section 7.2) and evaluation environments. The present section is structured along the different variables regarding the recommender system, the market approach, and the user behaviour. Its aim is to find the best performing configuration of the RecorDa approach. This includes different types of configurations, such as configurations with a strong focus on the recommender system component or a strong focus on the market approach component.

For each of these different configurations outlined in the previous section (see section 7.4.3), the following evaluation measurements were used:

- Categorisation (with precision and recall)
- Rank accuracy (with Spearman Correlation)
- Coverage
- Novelty
- Computation time

These measures were used for the specific setting of the RecorDa approach in the different evaluation settings. The best performing RecorDa settings (described at a variation of RecorDa) will now be compared to the performance of directed and undirected search in the following section.

<u>1a. Recommender system (functions and aggregation)</u>

Overall, most aggregation functions and similarity functions show similar performance (see Table 16). The Log Likelihood Similarity function and the maximum for aggregating the different sub-recommender system exhibit the best performance for categorisation by a slight margin, as well as good performance for coverage and novelty. Over the different iterations, there are also only slight differences in the learning speed of the different RecorDa variations (see Attachment C and Figure 12 - Figure 17).

The results show that the similarity function selection does not have a major impact on the system’s performance. This is probably due to the fact that all these functions are typically used in recommender systems and all measure similarity characteristics in the data.

The results further indicate that if one of the sub-recommender systems shows a high evaluation for a dataset, it is probably a sign that this dataset is relevant;

hence the maximum function is a good choice for aggregating the different sub-recommender systems.

There are no major differences in the time that most of these RecorDa variations take. It typically takes around 1 sec for the system to respond with a recommendation.

Settings [19] for variation	**Novelty**	**Coverage**	**Precision for rows**	**Precision for tables**	**Recall for tables**	**Computation time**
1) Minimum, Log Likelihood Similarity	0.46	0.28	0.6	0.48	0.43	1080ms
2) Maximum, Log Likelihood Similarity	0.46	0.28	0.61	0.48	0.43	1037ms
3) Average, Log Likelihood Similarity	0.47	0.28	0.59	0.48	0.43	1009ms
4) Minimum, City Block Similarity	0.46	0.28	0.59	0.48	0.43	1111ms
5) Maximum, City Block Similarity	0.47	0.28	0.6	0.48	0.43	1039ms
6) Average, City Block Similarity	0.46	0.28	0.59	0.48	0.43	1032ms
7) Minimum, Euclidean Distance Similarity	0.48	0.28	0.58	0.48	0.42	1019ms
8) Maximum, Euclidean Distance Similarity	0.49	0.29	0.6	0.47	0.41	1051ms
9) Average, Euclidean Distance Similarity	0.48	0.28	0.59	0.47	0.41	1029ms
10) Minimum, Pearson Correlation Similarity	0.46	0.28	0.58	0.48	0.44	1056ms
11) Maximum, Pearson Correlation Similarity	0.47	0.28	0.6	0.48	0.43	1003ms
12) Average, Pearson Correlation Similarity	0.46	0.28	0.59	0.48	0.43	1044ms

[19] The first measure describes the aggregation mechanism used among the different recommender systems. It can either take the minimum, maximum, or average of the three recommender systems outlined in chapter 6.

13) Minimum, Spearman Correlation Similarity	0.46	0.28	0.59	0.49	0.44	1053ms
14) Maximum, Spearman Correlation Similarity	0.47	0.28	0.6	0.48	0.43	1027ms
15) Average, Spearman Correlation Similarity	0.46	0.27	0.59	0.48	0.44	1082ms
16) Minimum, Tanimoto Coefficient Similarity	0.46	0.28	0.6	0.48	0.43	1024ms
17) Maximum, Tanimoto Coefficient Similarity	0.46	0.28	0.59	0.48	0.43	1104ms
18) Average, Tanimoto Coefficient Similarity	0.46	0.28	0.6	0.48	0.43	1018ms
19) Minimum, Uncentered Cosine Similarity	0.46	0.28	0.59	0.48	0.43	1003ms
20) Maximum, Uncentered Cosine Similarity	0.47	0.28	0.61	0.48	0.42	994ms
21) Average, Uncentered Cosine Similarity	0.47	0.28	0.6	0.48	0.42	1025ms

Table 16: Experiment 1a evaluations, average of experiment results

For the following experiments in 1b ID 2 will be used as functional setting upon which the further setting variations are tested and selected.

1b. Recommender system (number of recommendations)

All numbers of recommendations show good performance, with a continuous increase in accuracy over time (see Table 17). The results show that with more recommendations, the coverage and precision continuously increase until around five recommendations. The novelty remains higher for a smaller number of recommendations due to the fact that less data is shown in every step, leading to higher novelty in the results.

Over different recommendation iterations, a higher number of recommendations usually shows a faster increase in coverage and recall because more datasets can be shown and evaluated by the user (see Attachment C and Figure 18 - Figure 23). Precision however increases fastest for 5 recommendations with the strongest improvements coming from 3 to 5 recommendations. This particular result can be influenced by the input for the experiment: most users in this experimental setting only needed three to six additional data items, which is why having more than six recommendations does not result in higher accuracy. However, it also indicates 5 as being a suitable number of recommendations.

The computation time for all these RecorDa variations is similar, with a slight tendency to increase with the number of recommendations due to the additional computational effort in presenting more recommendations.

Settings[20] for variation	Novelty	Coverage	Precision for rows	Precision for tables	Recall for tables	Computation time
1) 2 Reco.	0.60	0.15	0.47	0.42	0.3	1143ms
2) 3 Reco.	0.57	0.21	0.5	0.42	0.35	1190ms
3) 5 Reco.	0.46	0.28	0.61	0.48	0.43	1213ms
4) 7 Reco.	0.34	0.3	0.59	0.46	0.44	1233ms

Table 17: Experiment 1b evaluations, average of experiment results

The following experiments will work with 5 recommendations.

1c. Recommender system (thresholds)

A broad range of different threshold settings were tried for the user. The results (see Table 18 and Figure 24-Figure 29) show a trade-off between precision and other measures, such as recall and novelty. This can be found in many recommender systems. With a higher overall confidence threshold, the precision of the datasets presented to the user increases, but the coverage, recall, and novelty are reduced. A higher threshold eliminates irrelevant recommendations. However, this also increases the chance of fewer relevant datasets being presented to the user.

This effect is clear for the overall confidence threshold, but less clear for user and item thresholds, showing that these thresholds have a lower impact. The results of experiment 1a show that the Maximum is the best method of aggregating the different sub-recommender systems. These results indicate that the user should be presented with a dataset if at least one of the three sub-recommender systems will identify it and will give it a high evaluation. The high thresholds for user and item sub-recommender systems indicate a similar tendency.

Computation time is similar for the different recommender systems, which is consistent with the previous results and expected, given that the computation process is almost the same for these different approaches.

[20] Describing the number of recommendations being shown to the user

Overall, the thresholds of 0.6 for the item recommendations, 0.8 for the user, and 0.25 for the confidence achieve the most suitable trade-off between precision and recall. Providing high precision and maintaining relatively high recall values for the other variables (i.e. novelty and coverage). They will hence be used as basis for the experiments from 2 onwards.

ID	Settings[21] of variations			No-velty	Cover-age	Precision for rows	Precision for tables	Recall for tables	Compu-tation time
	Item Rec. thresh.	User Rec. thresh.	Overall Conf. thresh.						
1	0	0	0	0.46	0.28	0.61	0.48	0.43	1253ms
2	0.3			0.46	0.28	0.61	0.48	0.43	1256ms
3	0.6			0.46	0.28	0.61	0.49	0.43	1041ms
4	0	0.2		0.47	0.28	0.6	0.48	0.42	1167ms
5	0.3			0.46	0.28	0.61	0.48	0.43	1165ms
6	0.6			0.46	0.28	0.61	0.49	0.43	1280ms
7	0	0.8		0.47	0.28	0.6	0.48	0.42	1145ms
8	0.3			0.46	0.28	0.6	0.48	0.43	1180ms
9	0.6			0.46	0.28	0.6	0.48	0.43	1138ms
10	0	0	0.25	0.46	0.28	0.61	0.49	0.43	1119ms
11	0.3			0.45	0.27	0.6	0.49	0.43	1074ms
12	0.6			0.44	0.27	0.62	0.49	0.41	1212ms
13	0	0.2		0.46	0.28	0.6	0.48	0.42	1251ms
14	0.3			0.45	0.28	0.6	0.49	0.43	1170ms
15	0.6			0.44	0.27	0.6	0.49	0.41	1225ms
16	0	0.8		0.46	0.28	0.6	0.48	0.42	1258ms
17	0.3			0.45	0.27	0.6	0.49	0.43	1270ms
18	0.6			0.44	0.27	0.63	0.49	0.41	1285ms
19	0	0	0.5	0.38	0.19	0.64	0.5	0.29	1028ms
20	0.3			0.37	0.19	0.66	0.5	0.3	1113ms
21	0.6			0.34	0.17	0.69	0.51	0.3	1021ms
22	0	0.2		0.38	0.19	0.64	0.5	0.29	994ms
23	0.3			0.37	0.19	0.66	0.5	0.3	1170ms
24	0.6			0.34	0.17	0.69	0.51	0.3	1226ms
25	0	0.8		0.38	0.19	0.64	0.5	0.29	1003ms
26	0.3			0.37	0.19	0.66	0.5	0.3	1022ms
27	0.6			0.33	0.17	0.7	0.51	0.3	1230ms

Table 18: Experiment 1c evaluations, average of experiment results

2. Market approach analysis (auction method)

Following the separate evaluation of the standalone recommender system component configuration is the evaluation of the market approach component

[21] Describing the thresholds being used by the item and the user recommendations systems, and the overall confidence threshold

configuration. It tests and compares further variations of the RecorDa approach (see Table 19) by including the market approach component.

ID	Settings[22] of variations
1	Don't run market (Standalone recommender system component)
2	Run market, Do same price for all users, and English auction method
3	Run market, Do individual prices for each user, and English auction method
4	Run market, Do same price for all users, and Dutch auction method
5	Run market, Do individual prices for each user, and Dutch auction method

Table 19: Experiment 2, market approach component settings on the auction mechanism being used

The average results (see show Table 20 and Figure 30-Figure 35) show that the RecorDa variation with the market component performs similar to the variation with the standalone recommender system. They have similar novelty, coverage, precision, and recall values with only a slight edge of the market approach in coverage and novelty after several iterations.

The market approach component variations have slightly higher recall for tables in the long run but a slightly smaller initial table recall. This indicates that it takes the market several iterations to identify the relevance evaluations of datasets but also that applying a market can help generate additional insights, by adding a different datasets relevancy valuation perspective. However, this would have to be further evaluated by follow up experiments.

Within the market variations there is little difference between the two variations with different auction mechanisms. The Dutch auction variations have a slightly bigger coverage, but a smaller precision at the beginning. Similar for overall and individual pricing. This shows that different market variations get to a similar price within data allocation domains. The main difference is in the computation time which is still within 0.5-1.2 sec. Due to the similarity in the performance of the market approach RecorDa variations, the standalone recommender and the English auction with individual pricing will be used for further experiments (ID 2 due to its slightly higher precision and faster computation time.)

Settings of variations	Novelty	Coverage	Precision for rows	Precision for tables	Recall for tables	Computation time
1) Standalone recommender system	0.44	0.27	0.62	0.49	0.41	1108ms

[22] Settings are for the different ways of managing the auction mechanism. First it defines the auction method and then if it is with each individual or with a set of all users. The first variable defines if the market is running or not

2) Market, English auction, individual prices	0.46	0.28	0.62	0.48	0.41	1687ms
3) Market, English auction, common prices	0.47	0.28	0.6	0.48	0.42	2219ms
4) Market, Dutch auction, individual	0.47	0.28	0.59	0.48	0.42	2144ms
5) Market, Dutch auction, common prices	0.47	0.28	0.59	0.48	0.42	2360ms

Table 20: Experiment 2 evaluations, average of experiment results

3. Recommender system influenced by market approach

This subsection evaluates which variation of the RecorDa approach performs best when testing the influence of the market approach component on the recommendations presented to the user.

ID	**Settings[23] of variations**			
	Run market	**Remove unprofitable datasets**	**Sorting method**	**Weighting of market**
1	No (standalone recommender system component)	-	-	-
2	Yes	No	Order	0.1
3				0.25
4				0.5
5				0.75
6			Confidence	0.1
7				0.25
8				0.5
9				0.75
10		Yes	Order	0.1
11				0.25
12				0.5
13				0.75
14			Confidence	0.1
15				0.25
16				0.5
17				0.75

Table 21: Experiment 3, influence of market mechanism on the recommender systems

[23] Settings describing the setup of how the results from the market based approach influence the recommendations. Additionally, the strength of the influence in terms of a weight is defined. The variable for running the market enables the comparison against the recommender systems approach.

The results indicate that the removal of unprofitable datasets results in worse performance. The reasons are twofold:

- Setting the budget and costs correctly can be difficult, and a slight mistake in settings can remove whole datasets; and
- It takes time for datasets to gain relevance, and preventing them from being presented does not give them the chance to gain relevance. Therefore, a slight reduction in relevance once means that it will not be presented again. This is in line with the previous observations that RecorDa with market approach component takes additional time before outperforming RecorDa with standalone recommender systems. Initially, the market is less accurate, and it takes several iterations for it to build up enough knowledge to start performing well.

Future research might be able to address this issue by improving the costs and budgeting process and finding solutions to keep temporarily irrelevant datasets in the process, similar to how a market might be able to keep temporarily insolvent companies in existence.

Both types of influence (order and confidence) seem to generate good results. However, the output is very similar in most cases. This shows that the results from the market approach can have a slight positive impact on the relevancy of the provided data but the impact is relatively small in this experimental setting. It is more an indicator that the market does not make the results worse and hence seems to be working properly for the right settings. Further experiments will test the positive benefit of using the market approach.

The weighting has a stronger impact on the confidence-based ordering when compared to the order based ranking. For the confidence-based evaluation, an overall lower weighting of the market of 0.1 shows worse performance than variations with higher weightings do.

The timing results show that the RecorDa variation with market approach component is slightly slower than the variation with standalone recommender system. The variations that require removal of datasets and influencing datasets based on the order from the market show the overall slightly worse performance, with a few negative outliers in computation time (ID 11 and ID 12). This is due to the additional computational effort required to analyse whether a dataset is lucrative and adjust the order.

ID	Settings			Novelty	Coverage	Precision for rows	Precision for tables	Recall for tables	Computation time
	Rec. presentations	Remove unprofitable.	Weighting						
1	Recommender system			0.44	0.27	0.62	0.49	0.41	1049ms
2	Order	No	0.1	0.46	0.28	0.61	0.48	0.41	1694ms
3			0.25	0.47	0.28	0.6	0.47	0.4	1662ms
4			0.5	0.46	0.27	0.63	0.48	0.42	1714ms
5			0.75	0.46	0.28	0.62	0.48	0.42	1701ms
6	Confid.		0.1	0.55	0.16	0.52	0.24	0.15	1893ms
7			0.25	0.44	0.2	0.6	0.41	0.27	1883ms
8			0.5	0.43	0.26	0.62	0.5	0.41	1655ms
9			0.75	0.45	0.27	0.61	0.49	0.43	1810ms
10	Order	Yes	0.1	0.55	0.26	0.55	0.3	0.25	1632ms
11			0.25	0.55	0.27	0.54	0.31	0.25	3335ms
12			0.5	0.56	0.26	0.52	0.29	0.24	2704ms
13			0.75	0.57	0.26	0.51	0.28	0.23	1964ms
14	Confid.		0.1	0.62	0.14	0.45	0.17	0.09	1859ms
15			0.25	0.54	0.16	0.56	0.24	0.14	1802ms
16			0.5	0.42	0.23	0.65	0.44	0.32	1762ms
17			0.75	0.54	0.26	0.54	0.34	0.27	1774ms

Table 22: Experiment 3 evaluations, average of experiment results

Overall, the results (see Table 22 and Figure 36-Figure 41) show that three variations of the RecorDa with the market approach component can potentially outperform the RecorDa approach variation with standalone recommender systems (IDs 5, 8, and 9) on Recall (ID 9), precision (ID 8), or coverage and novelty (ID 5). They show better performance on these variables without significantly sacrificing performance of other variables.

4. Influence of user behaviour

Four suitable variations were identified for further evaluation with different types of user behaviour (see Table 23). They are the different variations for the RecorDa approach that exhibited the best performance for the metrics of coverage, novelty, precision, recall, and rank accuracy based on the previous experiment.

ID	**Appr.**	**Previous experiment and experiment ID**	**Settings[24] of variations**
1	RecorDa standalone recommender system	Exp 2, ID 27	Max and Log Likelihood Similarity function, 5 recommendations, thresholds (0.6 for item recommender threshold, 0.8 for user recommender threshold, 0.8 for overall confidence threshold), and market not running

[24] See Experiments 1a, 1b, 1c, 2, and 3 for further details on these evaluations

2	RecorDa with market approach component	Exp 3, ID 5	Run market with recommender systems from ID 1 in Table 22, do individual prices for each user, English auction method, use order to present recommendations, and weighting of market of 0.75
3		Exp 3, ID 8	Run market with recommender systems from ID 1 in Table 22, do individual prices for each user, English auction method, use confidence to present recommendations, and weighting of market of 0.5
4		Exp 3, ID 9	Run market with recommender systems from ID 1 in Table 22, do individual prices for each user, English auction method, use confidence to present recommendations, and weighting of market of 0.75

Table 23: Experiment 4, settings of the different RecorDa approach setups used for further evaluation

The experiments, to test the influence of user behaviour, used the following settings.

ID	**Environmental Settings**[25]		
	Rating frequency	**Accuracy**	
		Selecting the correct recommendations	**Giving the correct rating in recommendations**
1	1	All recommendations	Extreme
2	3		
3	5		
4	10		
5	1	Top1 recommendation	
6		Top3 recommendations	
7		All correct recommendations	
8		All wrong recommendations	
9		Random recommendations	
10		No recommendations	
11		All recommendations	Strong
12			Medium
13			Neural bias
14			Positive bias
15			Negative bias
16			Neutral

Table 24: Experiment 4, influence of user behaviour on system performance

The results (see Table 25, Table 39, and Figure 36-Figure 41) can be broken down by the different environmental settings for user rating behaviour.

[25] Describing the frequency of intervals in which the user gives a rating, e.g. every 3rd set of recommendations presented to him, also describing the accuracy with regard to selecting the correct rating, e.g. a 5 rating when the dataset is actually relevant and the accuracy for selecting the correct datasets, e.g. giving recommendations to the first recommendation only, or only to the top three recommendations, or to all recommendations.

Rating frequency

In all four variations, less frequent ratings cause a reduction in recall, precision, and rank accuracy compared to more frequent ratings. This highlights the importance of user ratings to improve the system. More ratings mean that the RecorDa approach can learn and improve itself. Rating frequency has a similar impact on all approaches.

Accuracy in selecting the correct recommendations

Similarly, to the frequency, the accuracy in selecting which datasets receive ratings shows a similar tendency. Less accurate ratings lead to less precision, recall, and rank accuracy. The only exception to this tendency seems to be wrong recommendations. When the user provides feedback on which data should not be presented to the user, this seems to lead to the largest improvements in precision, recall, coverage, and novelty.

When there are only correct or no recommendations from the user, one of the RecorDa with market approaches component seems to slightly outperform the standalone recommender system, while the standalone recommender system performs better with regard to random recommendations.

Accuracy in giving the correct rating

Accurate recommendations have an important impact on the RecorDa approach's rating accuracy. Recommendations with a relatively strong accuracy divided between positive and negative ratings still performs relatively well. However, neutrally biased, positively biased, or strictly neutral ratings from the user reduce the system's performance. A negative bias, however, increases the performance. The strong and the negative bias ratings have in common that they provide many relatively negative ratings for datasets that are not relevant to the user. This verifies the importance of negative ratings from the user that was mentioned earlier.

ID	Settings			Novelty	Coverage	Precision for rows	Precision for tables	Recall for tables	Computation time
	Rec. presentations	Remove unprofitable.	Weighting						
1	Standalone recommender system			0.24	0.19	0.22	0.22	0.27	1567ms
5	Order	Strong	Strong	0.25	0.19	0.23	0.22	0.27	2234ms
8	Confid.			0.24	0.18	0.21	0.22	0.27	2222ms
9				0.25	0.19	0.22	0.22	0.27	2325ms

Table 25: Experiment 4 evaluations, average of experiment results

Overall, RecorDa variations using the market approach component tend to perform slightly better in the first iterations and after 10 iterations. They therefore tend to show good initial performance but then to learn more slowly. Compared to the standalone recommender system component, the market approach component also achieves a 10% higher Spearman correlation with regard to the order of the additionally presented data. The RecorDa with market approach component therefore provides some performance benefits compared to the standalone recommender system RecorDa approach. This is probably due to the additional knowledge involved in the market approach component by conducting further analysis on the datasets and the combinations of datasets with the use of mechanisms such as auctions.

These experiments specifically demonstrate the importance of receiving accurate ratings from the user. Especially negative ratings about data that the user does not want to see, seems to have an important impact. This shows that RecorDa builds up knowledge about which datasets it should filter as irrelevant. If the user provides more ratings, the user’s presented data will eventually converge to a fixed set of always presented datasets because the RecorDa approach learns the user’s current interests.

Overall, all RecorDa variations seem to provide results in a relatively short time frame of around 1.5 to 2.3 seconds, which is relatively efficient. The market approach component causes a slight computational disadvantage, but the difference is relatively small.

7.4.5 Experimental results – RecorDa vs. Search

This section compares the four best performing RecorDa variations (based on the experiments in the previous subsection) to the three different search approaches (directed search, undirected search, and learning search) with regard to the previously identified metrics and experiments (see Table 26 and Figure 42-Figure 47).

Novelty and coverage

For novelty and coverage, both directed and undirected search perform poorly. The datasets that are highest in the search results do not change. Hence there is not additional novelty after the first iteration for directed and undirected search. Learning search has the highest coverage and novelty because it eliminates search results that are not relevant in every iteration but ensures that a large number of them are still presented at least once.

The RecorDa with standalone recommender system and with market approach component perform between those learning and directed search. Except for a learning search approach in which the user also has to continuously provide feedback, the recommender systems and the market approaches therefore outperform search. Even for the learning search approach, the amount of input required from the user is much greater than the amount of input for the RecorDa approach. More input in the form of ratings can mean that the user is less patient when rating, and that it takes more time for the system to cover those datasets.

However, learning search still has the benefit of a higher coverage and novelty, and the positive effects of this, such as being able to react to changes in datasets or finding datasets that are potentially relevant for the user.

Precision and recall

For precision and recall, the directed search performs the best. It is immediately able to find the relevant data because the user is only looking for certain datasets. Conversely, undirected search shows the worse precision and recall performance for those values. It cannot find the specific datasets that are relevant to the user in the first search queries, and it therefore cannot show these datasets to the user at any time.

Learning search initially underperforms for recall compared to RecorDa with standalone recommender system and with market approach component, but then after a few iterations it starts outperforming these approaches. For precision, however, it is still outperformed by the RecorDa approach. The effect on precision has a similar reason as the effect seen for coverage. The learning search requires much input to find the relevant dataset and slowly improves, but requires more input and has a slower learning progress than the recommender system and the market approach components.

Computation time

For computation time, all approaches are reasonably quick and take within 0.1 to 3.5 sec of computation time. Undirected search exhibits the worst performance because it needs to go through a large number of datasets. Directed search is efficient because the datasets it needs to search are relatively small. The RecorDa is between directed search and undirected search in computation time. Learning search continuously reduces the datasets that it needs to search and therefore becomes faster with more iterations.

Summary

Overall, direct and learning search have a higher precision and recall, especially if the user knows where to search, but much lower coverage and novelty. Undirected search performs well in the cases in which the required data is in the top search results. If the required data is not in these results, it does worse than all other approaches including the two RecorDa approaches.

Compared to learning search, RecorDa has the additional advantage of knowing which datasets a user might need by looking for similar users. It takes learning search more time to obtain this knowledge. However, the RecorDa variations perform worse than directed search because in direct search the user provides the additional knowledge of knowing exactly where to look for data. This ability to narrow the databases shown to the user comes at the expense of smaller coverage with lower flexibility with regard to changing requirements.

ID	Settings			No-velty	Cover-age	Precision for rows	Precision for tables	Recall for tables	Compu-tation time
	Rec. pre-sen-tations	Re-move unpro-fitable.	Weigh-ting						
1	Directed search			0.1	0.08	1	1	0.97	113ms
2	Undirected search			0.12	0.12	0.04	0.07	0.09	3461ms
3	Learning search			0.76	0.56	0.43	0.25	0.47	751ms
4	Standalone recommender system			0.44	0.27	0.62	0.49	0.41	1120ms
5	Order	No	0.75	0.46	0.28	0.62	0.48	0.42	1697ms
6	Confid.		0.5	0.45	0.25	0.61	0.47	0.39	1667ms
7			0.75	0.45	0.28	0.61	0.48	0.43	1653ms

Table 26: Search evaluations, average of experiment results

7.5 Case study evaluation

As further evaluation of the RecorDa approach, two case studies were conducted to test the application of these approaches based on a real data systems environment. The case studies were conducted based on the knowledge that was obtained in the previous experiments to further verify the practical applicability of RecorDa. Overall, the thesis tested the following two different RecorDa variations based on the experimental performance (See Table 27).

ID	**Appr.**	**Previous experiment and experiment IS**	**Settings**[26]
1	Standalone recom-mender system	Exp 2, ID 27	Max and Log Likelihood similarity, 5 recommendations, thresholds (0.6 for item recommender threshold, 0.8 for user recommender threshold, 0.25 for overall confidence threshold), and market not running
2	With Market approach component	Exp 3, ID 5	Run market with recommender systems from ID 1, do individual prices for each user, English auction method, use order to present recommendations, and weighting of market of 0.75

Table 27: Case studies, settings of the different RecorDa with standalone recommender systems and with market approach component for the case study evaluation

Each case study was executed in a different company in a different industry and of a different size. Moreover, each case study addressed different organisational

[26] See Experiments 1a, 1b, 1c, 2, and 3 for further details on these evaluations

data management challenges to solve different problems. This demonstrates the broad applicability of the approaches developed in this thesis.

The following section will first introduce the procedure and the principles used for the case studies (see section 7.5.1). Next, the chapter will describe each of the case studies and go through their steps in section 7.5.2 for Case Study A about manufacturing part procurement, and in 7.5.3 for Case Study B about health care part catalogues for consumers and internal users.

7.5.1 Case study plan

The case studies followed the approach suggested by Kitchenham et al. [180], because this approach was specifically set up for information systems. In the context of the present study, these steps were as follows.

1. 'Define the hypothesis' [180]:

Based on the previous chapter, the following set of hypotheses for these case studies is based on the initial hypothesis.

A. Recommender systems and market approaches can be used to identify the relevant data for users in a company and increase the amount of relevant data allocated to the user while reducing the problem of data overload.

The respective null hypothesis is the following:

A. Recommender systems and market approaches cannot be used to identify the relevant data for users in a company and increase the amount of relevant data allocated to the user while reducing the problem of data overload.

2. 'Select the pilot projects' [180]:

A description of the case studies can be found in the following section (see sections 7.5.2 and 7.5.3). The following set of characteristics were checked in both case studies (see Table 28) to ensure their applicability.

#	Criteria name	Criteria description	Criteria value
1	Company size	The company needs to be large enough so that not every employee knows what the other employee is working on. This is typically the case between 100 to 230 people working in a company [216]–[218].	More than 200 employees
2	Task type	The RecorDa provide their main advantage by showing more data to the user and being capable of learning from the user. It is	Task needs to be data focused and repetitive

		therefore required that the user performs a task repetitively to test the performance of these approaches in such an environment.	
3	Problem type	The aim of the RecorDa approach is to overcome data overload and limited data availability. This needs to be a problem in a company in order to test the effect of these approaches in overcoming this problem.	Reason for parts of the problem is limited data availability
4	Amount of data	The number of datasets needs to be large so that it is not obvious which user is interested in which dataset.	More than 10 different datasets
5	Number of user types	The number of different users who in the case study needs to be large enough to create some variety and complexity, but still traceable within a case study to draw specific conclusions.	3-6 users should be sufficient
6	Number of users	There needs to be a large number of users for each user type to make learning between these different users possible, which is one of the impacts that the two approaches are supposed to achieve.	More than 10 users per user type

Table 28: Case study selection criteria

3. 'Identify the method of comparison' [180]:

Similar methods to RecorDa were identified (see section 7.3) which will be applied to the same project to compare their performance based on key measures, such as precision and recall. In these case studies, search, and requirement analysis are compared to the two types of RecorDa approaches.

4. 'Minimise the effect of confounding factors' [180]:

In order to minimise this effect, the following factors needed to be addressed.

#	Variable name	Description of influence	Variable control or elimination approach
1	User behaviour	The way in which the user provides search terms or ratings can have a strong influence on the performance of the system.	The main inputs from the user are ratings and search queries. This thesis makes assumptions about the user behaviour in these areas to reduce the impact of other factors, such as interface design for example.
2	User interface	The design and structure of the interface can potentially change the output that can be allocated to the user and the impact it has on the	Two different types of user interface were used for the two types of case study.

		user’s interaction with the system	
3	Data characteristics	The structure of the datasets, such as their type of columns and their content, the amount of data, and the quality of the data can have a strong impact on the system’s ability to provide this data to the user.	This study ignored the problem of data quality because there are existing solutions that would probably mitigate this problem. Therefore, the structure of the dataset and its content are the main variation to be considered. The characteristics of the dataset in the two case studies are different; this can be seen in the detailed description of the datasets (see Attachment A and B).
4	User interests in data	The specific interest of the user in different datasets can be important for the system’s performance because this is the data that the system needs to provide to the user.	This study examined users who had different tasks for their specific companies and who were working with a large variety of existing challenges, which resulted in a broad variety of dataset interests among them.
5	Information system environment	The environment of the information systems can have a strong influence on its performance. This variable describes the environment in which the system needs to operate.	The RecorDa approach works independently from current information systems. The only impact that these systems therefore have is on the data they provide to the user and the changes they make to the underlying dataset. Two different companies were chosen that varied in size, industry, and specific task of the user.

Table 29: Variables impacting the case study and how they were controlled or eliminated

Therefore, most aspects are maintained controlled within the experiments. By controlling most of the user behaviour aspects and using the same hardware as in the experiment, this study ensured that the treatment was the only variable impacting the differences in the results of the case studies.

5. ‘Plan the case study’ [180]:

Both case studies were not able to directly access the original data from the company. It was therefore necessary to use the approaches outlined in section 7.4.1 and Table 12. The steps are described in Jess et al. [214].

6. ‘Monitor the case study against the plan’ [180]:

The procedures and documents described in the previous point were applied and documented for all case studies.

7. ‘Analyse and report the results’ [180]:

The results of the case study can be found in sections 7.5.2 and 7.5.3. Subsequently, the chapter will show the performance of the two variations of RecorDa and compare them to the alternative solutions of requirement analysis and search.

7.5.2 Case Study A: Manufacturing part procurement

The first case study was conducted within a large manufacturing company with several hundreds of thousands of employees. The company manufactures large goods worth several tens to hundreds of millions of dollars. It therefore has a complex procurement challenge. Due to the large manufacturing products, the company has to order a large number of parts but in a relatively small batch size compared to other manufacturing companies.

This case study investigated a specific type of procurement with which this company deals. It has Performance-Based Logistics contracts (PBLs) with many of its customers. Within a PBL, the job of the company is to provide spare parts to the users of its machines. It needs to ensure that enough of the spare parts are allocated in time. The company is paid based on the time it takes to provide the spare part to the customer. Each PBL is held with a different company or along different military contracts. Government regulations and IT infrastructure of the companies with which it works often require the company to provide separate IT systems, including databases, for each of these different PBLs. This leaves it with a distributed infrastructure and datasets. The suppliers and parts for all of the PBLs are often the same in these systems. However, due to this distributed IT infrastructure, the company misses many opportunities to share data. It could receive discounts when ordering the same part or when ordering from the same supplier. At the centre of procurement for each PBL, there are five types of users:

- Asset Management: The Asset Manager decides how much of a certain spare part should be kept in the warehouse in case a customer needs it. The Asset Manager is therefore key to this decision-making process.
- Procurement: Procurement employees do the ordering of the parts from the supplier. After the order from the Asset Manager arrives, they handle the order and deal with the supplier throughout the ordering process.
- Supplier Management: The Supplier Manager stays in touch with suppliers to identify which type of parts they are allowed to deliver, so that procurement employees can order from them. They usually set up

framework contracts with major suppliers, which members of procurement then use to order parts. Supplier management conducts regular visits and questionnaires with the suppliers to ensure their capabilities.

- Risk Management: Risk Managers have a more strategic view of the supply chain. Their role is to evaluate supplier risk, especially with sub-tier suppliers.
- Supplier quality: Supplier quality employees are responsible for monitoring the quality of parts produced by different suppliers. They help with supplier management visits and check quality standards. They work with the supplier to improve its part quality, and with supplier management to select suitable suppliers.

This case study fulfils the requirements defined for case study selection. In order to execute the case study, the steps outlined in the previous section regarding the case study plan were followed (see Table 43 in Attachment E).

No specific datasets were received from the company for the study, however detailed representative mock-up datasets from the company were developed with a number of personnel typically included in two small-scale PBL contracts. For these case studies, there was the following number of users:

A. Asset Manager: Typically, one Asset Manager was responsible for each of these smaller PBLs; therefore, there were a total of two Asset Managers.
B. Procurement: Depending on the PBL and the parts ordered as part of a particular contract, a different number of Procurement Agents can be involved. For one of the PBLs, there were only two Procurement Agents involved, and for the other PBL there were mainly five.
C. Supplier Management: Supplier Management has a slightly different organisational structure. There could be 1-10 different Supplier Managers involved at any given point in time in a contract. This case study considered two for one of the PBLs, and four for the second PBL to be a reasonable assumption based on company experts' estimates.
D. Risk Management: Risk Management, like Supplier Management, has a slightly different organisational structure. However, the size of contract that this study examined for these two PBLs would probably involve only one to two people from Risk Management. Therefore, the case studies worked with two Risk Managers as an assumption.
E. Supplier Quality: Similar to Supplier Management, Supplier Quality could also involve a broad range of employees. For this case study, one for the first slightly smaller PBL and three for the second PBL were found to be a reasonable expert estimate.

The system experts' information was used to determine how many datasets each user would be working with on a monthly basis to study the medium- to longer-term effects of the RecorDa approach (see Table 30 for details on the number of interactions with the different datasets).

#	Role	Current dataset[27]	Number of interactions per month and user	Additional data needs
1	Asset Manager	*Order, Part Inventory, Procurement, Support and Service Parts*	150	*Historical Delivery Performance, DandB, Financial Health Assessment, IHS, Sub-tier Questionnaire*
2	Procurement	*Commodities for Suppliers, Supplier, Supplier to Parts, Order, Supplier*	150	*Historical Delivery Performance, DandB, Financial Health Assessment, IHS, Sub-tier Questionnaire*
3	Supplier Management	*Sub-tier Questionnaire, Supplier Management Visits, Commodities for Supplier, Supplier to Parts, Supplier*	8	*Historical Delivery Performance, Financial Health Assessment, IHS*
4	Risk Management	*Commodities for Supplier, Historical Delivery Performance, Order, Procurement, Supplier, Supplier Management Visits, Supplier to Parts, Order, Procurement*	30	*DandB, IHS,*
5	Supplier Quality	*Supplier Management Visits*	8	*DandB, Quality Control*

Table 30: Case Study A, user interaction with the dataset

[27] The datasets used for the case study are the same datasets used for the experiments (see Attachment A for further details)

The results of the case study (see Table 31 and Table 40 for additional details with different user behaviour, and Figure 48-Figure 53) are in close agreement with the results of the experiments. For novelty and coverage, directed and undirected search underperform compared to RecorDa approaches. Learning search achieves an overall higher novelty and coverage at the beginning, but at the expense of an initial slower increase in precision and recall. Directed search has high precision and recall, while undirected search has the worst performance of all methods in precision and recall. The RecorDa approach variations perform well on precision, but do not recall all of the datasets relevant for the user.

RecorDa approaches trade-off a slightly faster conversion to the relevant datasets for the user (higher precision) with slightly less required user input against an overall lower recall, and overall lower coverage of learning search. RecorDa uses the additional knowledge from other users and about data characteristics. Learning search takes several iterations to build this knowledge. Direct search performs well for precision because it already knows exactly where to search for the data.

System	Novelty	Coverage	Precision for rows	Precision for tables	Recall for tables	Computation time
Directed search	0.02	0.1	1	1	1	104ms
Undirected search	0.02	0.12	0.07	0.19	0.19	661ms
Learning search	0.12	0.75	0.95	0.9	0.92	101ms
RecorDa standalone recommender system	0.08	0.39	0.94	0.81	0.66	1387ms
RecorDa with market approach	0.08	0.41	0.93	0.81	0.66	2626ms

Table 31: Case Study A evaluations, average of experiment results

For the different user behaviours, the effect of user input frequency is only significant for the first iterations, but given the large number of iterations they decrease in importance. The frequency has no impact on direct and undirected search, but a strong impact on learning search. Less input from the user in the form of rating feedback on which search results are not relevant means a much slower conversion of precision, recall, and coverage, and also overall lower novelty. The RecorDa approach therefore outperforms learning search in situations with some but overall lower user input. The effect is similar for the number of presented datasets that are rated. Learning search also requires a higher input from the user and generally converges more slowly to a better performance in terms of precision, recall, and coverage. For less accurate ratings, the impact of wrong ratings is slightly stronger for learning search. However, even some wrong or less user input still keep the RecorDa variations ahead of learning search for the first iterations of the learning process.

When RecorDa variations with a standalone recommender system and with a market approach component are compared, the main difference is in rating accuracy. The standalone recommender system relies more on strong, correct, or wrong ratings from the user, while the RecorDa with market approach component outperforms the RecorDa with standalone recommender system in the other cases in which the user rating is less clear. The market approach is thus less sensitive to less accurate user ratings within companies for good performance on precision and recall.

The computation time is higher for the RecorDa with market approach component and the RecorDa standalone recommender system approach than for the search approach, especially over several iterations. The search approaches are continuously improving. However, the overall computation times are still relatively effective and within a few seconds of response time.

7.5.3 Case Study B: healthcare part catalogue for customers and internal users

The second case study was conducted within a large healthcare supply chain management company. It has around 200 employees in three different locations, and is a spin off with close relationships to a health care systems provider with up to 5,000 employees. The company offers services such as contract management, procurement, and analytics to hospitals and healthcare companies. It manages contracts, sourcing, procurement, storage, and delivery of healthcare products for these hospitals.

As part of its offering, the company has an internal catalogue of products (see Figure 11), which the customers or internal users can search to find additional data about their products. The company recently conducted a master data management initiative to potentially use the data and provide additional services and analytics to customers and employees. It found that it had a rich dataset of potentially relevant data for the different types of users. However, selecting which data was most relevant for the different users was a challenge. The company saw RecorDa as a solution to this problem. It collects large amounts of data. Identifying which of this data is relevant for certain customers was not intuitive for them. Many of its customers and internal users were also not aware of the potential datasets available to them.

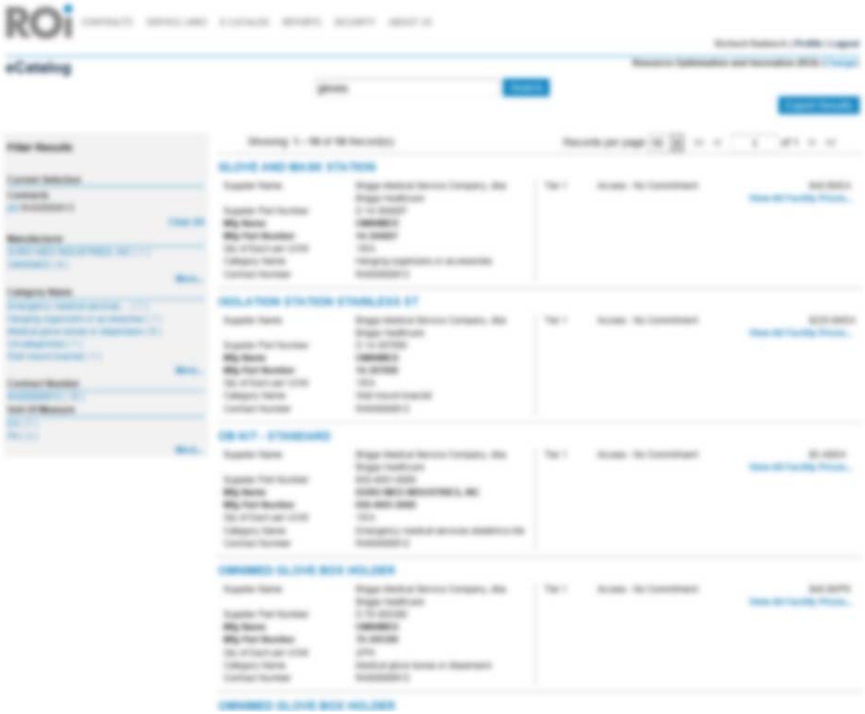


Figure 11: Blurred picture of the existing graphical user interface (GUI) for the electronic catalogue

Overall, this case study examined the following four types of users of the company’s electronic catalogue:

- Material Management (external customer): Managing the purchase of products and which type of products the company could consider buying.
- Purchasing (external customer): Purchasing of new material from the customer by looking through suppliers and available products.
- Internal Purchasing (internal user): Employees who execute purchasing requests for customers and ensure sufficient availability of parts in their warehouses.
- Internal Sourcing (internal user): Employees who work with suppliers to agree on contracts which they and their customers can use for future purchases. Longer-term contracts are critical and usual in healthcare. It is therefore important to manage these contracts a key offering of this healthcare service provider.

This case study fulfils the requirements defined for the case study selection.

The case study followed the steps outlined in the previous section regarding the case study plan (see Table 44 in Attachment E). It worked with sample datasets, a mock-up of the company’s GUI design, and the following number of users.

A. Material Management: For the dataset, the study included a total of five different customers, assuming that each customer was doing Material Management with one person.
B. Purchasing: Like for Material Management, for the purpose of the study, one single person also did the purchasing for each customer.
C. Internal Purchasing: The company's purchasing team comprised six people who interacted with the dataset.
D. Internal Sourcing: The company's sourcing team comprised five people who interacted with the dataset.

Expert interviews and personnel counts were used to estimate how many interactions with the system each user would have on a weekly basis (see Table 32 for details on the number of interaction with the different datasets).

#	Role	Current dataset[28]	Number of interactions per week and user	Additional data needs
1	Material Management	Electronic Catalogue	Each Material Management user on the customer side has an estimated 40 interactions per week	1) IMS Benchmarking Data_Unit and prices 2) Contract Mgmt System Data_ Contract description
2	Purchasing	Electronic Catalogue	Each purchasing customer has an estimated 60 interactions per week	1) IMS Benchmarking Data_Price comparison 2) IMS Benchmarking Data_Unit and prices
3	Internal Purchasing	Electronic Catalogue	Each user has an estimated 50 interactions per week	1) Contract Mgmt System Data_ Contract description 2) IMS Benchmarking Data_Price comparison
4	Internal Sourcing	Electronic Catalogue	Each user has an estimated 50 interactions per week	1) PO Spend Data_ Time and Facility details

Table 32: Case Study B, user interaction with the dataset

This case only permitted the use of two additional recommendations due to space limitations in the user interface (see Figure 11).

The results (see Table 33 and Table 41 for additional details with different user behaviour, and Figure 54-Figure 59) show a similar performance to previous experiments and case studies regarding the search, recommender system, and RecorDa approaches.

[28] Detailed list of datasets use for this case study can be found in Attachment B.

Novelty and coverage

Directed search and undirected search have poor coverage and novelty. This is because they continuously find and present the same tables based on number of search results found in these tables. This approach lowers the amount of data tables that a user sees. Conversely, learning search shows the highest coverage and high novelty, especially at the beginning of the case study. This is due to the fact that it slowly disregards more and more tables it searches. However, the benefit of having a high level of novelty and coverage for learning search is that this enables it to find good datasets, which is reflected in a smaller increase in precision and recall.

Precision and recall

Directed search again clearly outperforms all other approaches in precision and recall. This is because directed search already knows where it should look for data. The standalone recommender system and the market approach component variations both show similar results to learning search. Learning search has a slightly slower increase in precision than these two approaches and a smaller increase in recall than the RecorDa approach. In the first few iterations, both approaches outperform learning search, but after several iterations learning search slightly outperforms them. These initial results can have a large impact on the user's impression of the system. Learning search therefore benefits from searching through all datasets with a higher coverage, but it also suffers from a shorter learning rate and taking more time to find relevant data.

The recommender system and the RecorDa approach seem to clearly show better values than undirected search for users who do not know where to search for data. Compared to directed search, the performance of the recommender system component and the market approach component regarding precision and recall requires 10-30 user inputs to reach a similar level. Directed search again benefits from the user's knowledge with regard to determining where the user needs to search for specific data. However, RecorDa has much higher coverage and novelty, thus enabling the user to potentially find relevant datasets that the user does not know about yet. The RecorDa approach shows benefits with regard to novelty later in the case study. A continuous, relatively high novelty enables a continuous exploration of existing datasets.

When the RecorDa standalone recommender system and with market approach component are compared, the results show that all variations of user accuracy have an important impact on the performance of both systems. Less frequent ratings initially make the RecorDa standalone recommender system approach less accurate, but with an even smaller number of ratings the RecorDa with market approach component performs even worse. The rating accuracy again verifies the importance of giving wrong recommendations and identifying what is not of interest to the user. The reason for the stronger dependence of the RecorDa with market approach component than the standalone recommender systems on accurate ratings may be the smaller number of recommendations presented in this particular case study. This smaller number causes the system to have a low tolerance to data items that no longer appear for the user and that are therefore not rated or evaluated, which impacts future presentations to the user.

Computation time is similar for the market approach and the recommender system, and is longer than the time that most search approaches take. This is due to the additional computation of the recommender system and market approaches in evaluating the dataset and comparing other users' interests. Similarly, to the previous case study, the results are still relatively quick and significantly faster than in the previous case, due to the smaller number of user data interests.

System	Novelty	Coverage	Precision for rows	Precision for tables	Recall for tables	Computation time
Directed search	0.02	0.02	1	1	1	16ms
Undirected search	0.04	0.11	0.02	0.02	0.06	515ms
Learning search	0.35	0.79	0.83	0.65	0.9	54ms
RecorDa standalone recommender system	0.3	0.45	0.75	0.65	0.5	725ms
RecorDa with market approach	0.26	0.33	0.78	0.69	0.77	1257ms

Table 33: Case Study A evaluations, average of experiment results

7.6 Setup times

While computation time was covered within the different cases studies and experiments, another key aspect of time performance is the time to set up the system. Search and the RecorDa recommender system algorithms have both been developed; hence effort would only be needed to integrate the system into the current environment. However, for both approaches – search and RecorDa – the same type of access to the data would still be required. They are therefore similar in this area of development.

The RecorDa approach would require additional time to set up user budgets and costs for datasets. The case studies found that these costs were relatively easily obtained within most of the participating companies. In the case studies, this information was usually obtained within a few one-hour interviews with relevant experts. The costs for development and implementation are therefore not much higher for the RecorDa approach than for search approaches.

Decision theory and requirement analysis would take a long time for a large number of users. They require several interviews and user decision modelling for a series of users. Exact timings are difficult to estimate because they would vary with the user's ability and the time it takes to conduct this analysis. Decision theory and requirement analysis will also become too complex for large companies.

7.7 Evaluation summary

This chapter first identified the following specific measures to evaluate relevance by using the different categories of measures (see Table 2).

A. Relevance by presenting data known to the user. This is typically measured with precision and recall as the measures.
B. Relevance by presenting previously unknown but potentially relevant datasets to the user measured by evaluating the variety of available datasets shown to a user with coverage and novelty.
C. Relevance by providing the data in an efficient manner by measuring computation time.

These measures were used in a series of experiments and case studies designed with industry leaders. The first step identified the best performing RecorDa variations, which were (see Table 27 for additional details):

- The RecorDa standalone recommender system using Log Likelihood Similarity and the Max of all three recommendation parts with a threshold of 0.8 for user recommendations, 0.6 for item recommendations, and 0.25 as an overall confidence threshold.
- The RecorDa with market approach component using individual pricing for each user, English Auctions and influencing the order of the recommendations with a weighting of 0.75%.

These two variations were compared to different alternative methods:

- Search comparing the syntax and semantics of a search term. Different search approaches were used for comparison:
  - Directed search, which already knows in exactly which data table to look for the data.
  - Undirected search, which searches through all data tables.
  - Learning search, which uses the input from the user to limit the data tables that are searched for specific tasks over time.
- Requirement analysis, which uses information gathered from surveys, interviews, or user observations to design the information system in a way that the user receives all the required data that the user knows about.
- Decision- or value-based techniques to model the user's decision-making process and optimise the data required to improve this decision-making process.

These measures were tested in a series of experiments and two industrial case studies on the different metrics for relevance. With regard to the different key measures, the case study results indicate certain strengths and weaknesses of each of the different techniques and the RecorDa approach (see Table 34).

Measure of relevance	**Different approaches**			
	RecorDa	**Search**	**Requirement analysis**	**Decision theory**
Precision	Increasing over time to around 65-81%	Directed: 100% Undirected: 2-19% Learning: 65-90% (continuously increasing)	100% assuming a rigorous process considering all datasets	100% assuming a rigorous process considering all datasets
Recall	Increasing over time to around 50-77%	Directed: 100% Undirected: 6-19% Learning: 90-92% (continuously increasing)		
Novelty	26-30%	Directed: 2% Undirected: 2-4% Learning: 12-35%	0%	0%
Coverage	33-45%	Directed: 2-10% Undirected: 11-12% Learning: 75-79% (continuously increasing)	Typically just 10-30% depending on the subset of the data tables relevant for a user	100% because all datasets would have been considered
Computation time	0.7-2.6s	0.01-0.7s	Normal system performance but larger setup effort initially	Normal system performance but larger setup effort initially

Table 34: Describing the performance of different approaches with different relevance measures for the main case studies

The results show that direct search, decision theory, and requirement analysis outperform learning search and RecorDa on precision and recall. This is due to the additional data that they entail: the user knows where to search, and requirement analysis identifies the data needed for a specific user.

However, these approaches fall short when:

1. the requirements for a user change quickly; or
2. the user is not aware of all available datasets.

Both situations occur more frequently with the increasing amount of data and the more frequently changing requirements. The users need to adapt faster and will not always be aware of the data that they need.

When compared to learning search, RecorDa offers a trade-off between:

A. Quicker convergence to good precision and recall due to the leverage of similar users, and better novelty to keep the user interest in interacting with the new system; and
B. Broader coverage for learning search to ensure higher long-term precision and recall.

Learning search shows more datasets to the user, which enables it to collect more knowledge about the user’s interests. RecorDa captures this knowledge from other users and data characteristics enabling a faster initial increase but a higher likelihood of missing some datasets.

Search and recommender systems have specific advantages, which is why existing companies that have found similar problems in different situations often combine them, such as Google and Amazon. Exploring combinations of these two approaches would be interesting in future work.

8. Discussion and conclusion

8.1 Introduction

This chapter will summarise the course and the results of this research. It will present the main conclusions and summarise contributions to the scientific body of knowledge. It will further discuss the limitations of the research and the potential for future work in developing the RecorDa architecture.

8.2 Summary of research

This thesis used recommender systems and market approaches with the purpose of improving user data allocation by developing the RecorDa approach.

After providing an overview of current research and industry practises in data management (see Chapter 2), the thesis identified a key problem in the increasing amounts of data and user task complexity, indicating the necessity to find solutions to reduce the complexity of data allocation to users and to identify the most relevant datasets. Existing techniques for finding the relevant datasets for a user, such as decision theory and search, are limited with regard to scalability, their ability to assign valuations to specific datasets, and their flexibility when providing datasets to users. The literature review also shows the potential of recommender systems and market approaches to allocate resources (which are datasets in the context of this thesis) to people or other entities (e.g. machines on the shop floor).
This thesis therefore looked into architectures and functional aspects for using recommender systems and / or market approaches in a promising manner (see chapter 4). Based on this it developed the RecorDa approach in Chapters 5 and 6 build on these two techniques:

(a) Recommender systems: The recommender system component consists of three subsystems to generate each recommendation. The first subsystem uses the data characteristics of the different datasets that could be recommended to identify a similar dataset to the datasets currently presented to the user. In addition, a user- and item-based system uses ratings allocated by the user to find additionally relevant datasets that have received similar ratings (item-based recommender system) or datasets relevant for users who are similar to the current user (user-based recommender system). The combination of these three sub-systems determines which additional data is presented to the user.

(b) Market approaches: The market approach uses the data combinations presented with the recommender system component to further evaluate the relevance allocated to each dataset compared to its costs. Based on these combinations, the market approach then influences the recommender system's output in the datasets presented to the user.

Both variations of the RecorDa approach have shown their potential in domains with similar problem characteristics, such as supply chain management and ecommerce.

Chapter 7 compared RecorDa to approaches based on decision theory, requirements analysis, and search as alternatives to help provide datasets to users. This thesis used specific case studies and experiments regarding data allocation problems and tested the performance of these different techniques to solve those problems. It showed that RecorDa can outperform different search techniques by being faster in adjusting to changing user requirements in data allocation. RecorDa has the benefit of leveraging information from other users in selecting the data allocated to the user, and is therefore faster in adjusting based on additional knowledge. However, RecorDa shows worse performance in situations in which the user's requirements do not change.

8.3 Key results

The following two ways of using recommender systems and market approach that offer the most potential based on the analysis of different architectures are the following:

- Standalone recommender system, which provides data recommendation without any market approach
- Market approach based on a recommender system, which leverages the recommender system to determine the inputs for the market evaluation

They are adopted as variations of the RecorDa approach. Details of the key functionalities show that the RecorDa approach is robust against various functional variations and most settings of these functionalities show very similar performance. Overall, this thesis found that the following settings show to be the most promising:

- Standalone recommender system component using Log Likelihood similarity, taking the maximum of the sub-recommender system functions, and using thresholds of 0.6 for item, 0.8 for user, and 0.25 for the overall confidence threshold
- Market component working based on the aforementioned recommender systems component using English Auctions and have a weighting of 0.75 for the market

These two approaches were further used for comparison against alternative data allocation techniques.

The comparison results show that the performance of different approaches varies with the specific measure of relevance. Directed search, decision theory, and requirement analysis outperform all other techniques on precision and recall. They are effective in ensuring that the user is provided with the datasets that the user or someone designing the information systems already knows is relevant. This can be seen by the 100% precision and recall often achieved by these approaches compared to other techniques. The RecorDa approach performed at a precision and recall of only 50-80% in most case studies and experiments, and only reached these after a few iterations. The stronger performance of the alternative techniques on these measures is unsurprising due to the additional knowledge that they use.

However, when additional measures of relevance are used, such as novelty or coverage, then the RecorDa approach clearly outperforms these techniques. It has a higher coverage and novelty than all other approaches except for learning search. However, compared to learning search, it achieves a faster increase in precision and recall.

8.4 Conclusion

This section presents the conclusions of this thesis by reviewing and answering the initial hypothesis and research questions.

Research question 1*: What is the best way of using recommender systems and / or market approaches in industrial data allocation to improve performance in terms of precision, recall, novelty, coverage and computation time?*

The analysis and results from chapter 4, subchapter 7.4, and subchapter 7.5 show that two ways of using recommender systems and / or market approaches could perform best in terms of precision, recall, novelty, and coverage:

A. Standalone recommender system component which runs the recommendation engine and generate the suggested datasets combining three recommender systems functions (item, user, and content based). A detailed assessment showed that an item and user based recommender systems using the Log Likelihood similarity shows the best performance but only by a small margin compared to other approaches. For the content-based recommender system this thesis identified data characterisation as an approach describing the actual datasets well and helping to overcome the cold-start problem.
B. Optional Market component leveraging the output from the recommender systems component by using a utility function based on data usage. It further identifies the costs for datasets using expert interviews and combines these with an English auction using a market maker to determine the relevancy for a dataset.

Using the recommender system first in the architecture setup offers various benefits such as overcoming the cold-start problem and easier considerations of ratings from the user. These two characteristics are especially critical in an environment where the underlying user requirements are constantly evolving.

Future investigation of the different approaches could test other architectures or settings. This thesis explored two of these approaches and tested its performance in identifying relevant datasets compared to existing techniques. Using the RecorDa approach this thesis then addressed its second research question.

***Research question 2**: Can recommender system and market approach individually or in combination identify relevant data better than potential alternative techniques?*

The results of the experiments and case studies in Chapter 7 show that the RecorDa approach fills a gap between learning search and direct search. Direct search, requirement analysis, and decision theory are highly effective in providing the user with datasets that fit that user's current requirements. This is because they work based on the users' knowledge about the desired data.

However, given the limitations in data management mentioned in Chapter 2, such as a continuous increase in datasets and an increase in required flexibility in allocating the relevant datasets to the user, these techniques face challenges for two reasons: they are 1) not scalable and b) not adaptable enough. With several hundred datasets and users, requirement analysis, decision theory, and simple direct search in specific datasets struggle to include all variables in their model and design the system that provides data targeted for all these different types of users, which is reflected in current problems of information systems. This lack of adaptability can be seen when considering these approaches' measures for novelty and coverage. All three techniques only present a fraction of all datasets available to the user, which makes it difficult for the user to discover additional datasets that are potentially relevant. This type of discovery is essential when the task of the user evolves, thus creating the need for additional datasets.

Learning search, however, has higher novelty and coverage. It hence provides some of the adaptability and scalability required. However, it takes many iterations to reach high precision and recall, and is therefore much slower in adapting to changes. RecorDa shows a faster initial increase in precision and recall while still providing high values for novelty and coverage.

All of the results for the different types of search are made under the assumption that the user types in the correct keywords and makes the effort of looking for them. It also requires all datasets to be easily searchable. These are generous assumptions stacking the results in favour of search. In addition, RecorDa therefore has the added benefit of requiring less user input, hence making it less prone to error and requiring less interaction from the user, thereby potentially increasing the likelihood of the user employing this system.

It can therefore be concluded that the RecorDa approach can help provide the user with more relevant data assuming an environment where the user's needs change quickly and the amount of datasets and user tasks are widely different and constantly evolving.

8.5 Novelty

This thesis identified a series of novel or adapted concepts and approaches, metrics, and evaluations which can be summarized into the following four categories:

1. The concept of deployment of recommender system and market approach in data allocations is new. No prior research has used the potential of recommender systems and market approaches for data allocation. This thesis is the first to apply and analyse these techniques for industrial data allocation. It develops the RecorDa approach which provides additional data to users that helps them with their decision making.
2. This thesis identifies the most promising ways (in terms of precision, recall, novelty and coverage) of using market approaches and / or recommender systems for the problem of data allocation. There are various ways of using recommender systems and market approaches. This thesis identifies the key functional and architectural decisions within recommender systems, market approaches and in their combination. Such as the type of a) Auctions or use of a market maker for market approaches, b) type of recommendation function for recommender systems, or c) the level of data granularity (i.e. record, table, or database level) used by these systems. It further provides a qualitative assessment of the different architectures and a detailed quantitative assessment of most functional setups.
3. This thesis expands the current techniques for recommender systems and market approaches into the data allocation domain and develops the RecorDa approach. Both techniques have been suggested for data allocation. However, they have never been used in this domain. In order to make them applicable various changes needed to be made. The main adaptations are the following
    a. Introduction of an approach for content-based recommendations by identifying data characterisation or schema matching from a different domain and applying it towards content/based recommender systems in the domain of data allocation.
    b. Development of a utility function for the market approach based on data usage. The main principles for selecting this approach have been used but developing the specific algorithm for allocating different types of budgets and using concepts such as the ValueMap were adjustments to successfully implement this approach.

   c. Beside these main changes a series of small adjustments and implementations decisions were also required throughout this thesis.

4. Quantitative assessment for comparing recommender systems, market approaches and alternatives for industrial data allocation. Testing the benefit of a new method in data allocation requires a series of case studies, experiments and evaluation metrics. This thesis identified these for data relevancy to test the potential of these approaches and the specific benefits that a RecorDa approach can provide.

Following this evaluation of novelty, the following subsection will discuss the contribution of this thesis.

8.6 Contributions

This study has made the following main contributions to the existing literature.

a) **Motivation for alternative data allocation solutions**: Chapter 2 demonstrates the issue of a growing amount of datasets and a continuously increasing complexity for user tasks. This issue requires new solutions to deal with data allocation.

b) **Architecture**: This thesis analyses various architectures using recommender system and / or market approaches. A key contribution is the development of the specific RecorDa architecture using recommender systems component and market approach component as a solution to the problem of relevant data allocation. To date, existing recommender systems and market approaches have not been specifically designed individually or combined to address this type of problem.

c) **Application**: This thesis provides the first implementation of the RecorDa approach. While recommender systems and market approaches have been used in different domains, and their application has been proposed for data management problems, they have never been used for the specific problem of data allocation. This thesis contribution is to present various issues that need to be overcome for such an implementation, and ways in which this can be done. These issues include the need for a content-based approach in the recommender systems component to overcome the cold-start problem of recommender systems, or the importance of defining a utility function for market approaches. These issues, while known in other domains, have not been identified for the adaptation of data allocation with information systems.

d) **Concept of the Value Map and data utility function**: The Value Map and data utility function form the basis for the RecorDa approach, but they also describe ways to address problems in data management, such as the difficulty of assessing relevance on a larger scale and allocating this relevance to a specific dataset. This thesis develops this concept, which can be further extended to improve the data allocated to the user and the relevance evaluation accuracy. To date, no adaptation of recommender systems or market approaches for this type of problem has been developed.

e) **Addressing data combinations**: Existing research often neglects the importance of evaluating combinations of datasets instead of datasets on their own. A dataset on its own might be completely worthless but combined with another dataset it can provide additional value. By including market approaches or taking combinations and transferring them to individual datasets, this thesis overcomes the limitations of a number of data relevance evaluation methods.

f) **Tested benefits compared to alternative solutions:** This thesis identifies ways of comparing the RecorDa approach to existing solutions. By evaluating these solutions, this thesis demonstrates in which situations and applications they are beneficial, which provides insight into future applications of these techniques within industrial companies.

g) **Proof of concept for further development**: Based on the initial development of these two techniques, additional variations of the RecorDa approach can be applied. The architectures can be used for additional implementations. This thesis shows variations that could have an impact, and a route for further improving the RecorDa approach.

8.7 Limitations

This section discusses the limitations of this research. These can be divided into two points.

The first concerns the areas that are not addressed with regard to assessing the ways recommender systems and market approach are used separately or in combination. This thesis provides an initial architecture and functionality evaluation. However, additional work needs to validate and improve upon the specific RecorDa approach developed. Specifically, the following issues have not been fully captured:

- **Number of architectures explored:** This thesis only provides and initial assessment of different architectures and functionality decisions. It develops the most promising approach (in terms of precision, recall, novelty and coverage) based on this assessment. Further research needs to potentially test additional architecture combinations.
- **Architectural and setup selection**: There are a series of additional variables and variations possible for the current implementation. A series of these variations have currently not been tested and could potentially provide additional improvements. Examples of adjustments are the use of

multiplication instead of additions within the auction algorithms or a different type of recommendation engine using semantics in addition to syntax.

Second, this thesis is also limited with regard to the tests and evaluations conducted in order to compare the RecorDa approach to alternatives. The evaluation was based on extensive experiments and two case studies, but additional work needs to be done to fully ensure that these techniques offer a benefit in all types of industrial environments. Potential additions include the following limitations:

- **Unstructured datasets**: The current approach is not developed and tested for unstructured datasets, whereas unstructured data is becoming increasingly relevant for industrial companies.
- **Syntax and semantics similarity:** Currently the approach only compares identical word matches. It does not address similarity in semantics or syntax, such as simple typos or words with a similar meaning.
- **Full industrial roll-out:** The RecorDa approach needs to be tested in a life system to fully explore its potential.

Besides the limitations in the selected development approach, this research also omitted some issues with regard to fully testing the different approaches.

- **Privacy and security**: Many industrial companies are concerned about data privacy and data security. Not every user in a company can have access to every dataset. The current implementation of the RecorDa approach therefore has the risk of accidentally providing a user with a dataset that the user should not have access to. This risk can be mitigated with existing techniques, such as clustering users by access groups and including these access groups in the recommender system. This thesis does not address the issue of data security. However, various systems exist to control user access to data, as reviewed by Upadhyaya et al. [219], or to address the issue of data sharing [220].
- **Data duplication**: Besides the issue of privacy, there is also a limitation to how many users are allowed to use a specific dataset within a company. Companies enter into contracts with data suppliers that sometimes limit the number of people who are allowed to use a dataset. These kinds of limitations on the number of data uses are not considered in this thesis.
- **Accurate data relevance evaluation**: Estimating the accurate relevance of a dataset is a complicated problem [39], [73]. The current market approach aims to evaluate the impact of specific datasets using different utility functions. However, the specific valuations found for these datasets only include a limited number of influences that determine the relevance of a dataset, and therefore cannot provide the administrator with a complete answer to make further decisions about datasets.

- **User validation**: The evaluation presented in Chapter 7 did not use input directly from users, but instead assumes certain user reactions. This simplified the evaluation because it did not require access to the data systems and the issues of including a prototype into these systems. It also reduced the impact of user interface design on the results. However, future work should consider the influence of a user's interaction with the system.
- **Data reusability**: This thesis assumes that datasets can be easily reused for a different purpose. However, there are various challenges in reusing a dataset [221], which could potentially become an issue for the RecorDa approach. For example, the user might not always know how to read or interpret the additional data that is presented.

8.8 Future work

The results of this research suggest the potential of the RecorDa approach variations to overcome the complex problem of data allocation. The thesis provides initial architectures and evaluation mechanisms for these approach variations and develops a basis for additional developments. However, this research is only an initial step in applying these or similar but more flexible, dynamic, and robust data allocations in information systems. Additional work needs to be conducted to i) provide additional developments applicable for industrial use, and ii) further evaluate RecorDa's potential.

Regarding the first issue (i) of making RecorDa more applicable for industry, the following topics should be addressed.

a) **Unstructured Data**: RecorDa works with structured data (i.e. data in table format). However, unstructured data is becoming more important for companies and much of the data that is relevant for the user can be found in PDF or text files, for example. It would be interesting to convert unstructured data into structured data, or to think of alternative extensions of the current work to incorporate this unstructured data.

b) **Rating quality**: This thesis has established the importance of receiving accurate ratings from the user in order to provide good recommendations with the RecorDa approach. Additional work is required to identify the circumstances under which a user provides the best ratings. This could potentially even include the automatic capture of user interaction with a dataset to infer the user's rating, or providing the user with only two options (e.g. like and dislike).

c) **Data description:** Whenever data is presented, it usually helps to describe this data to the user to avoid problems of using data in a different context, as discussed by Woodall and Wainmann [221]. There are various approaches, such as the linked open data vocabulary [222], for example, that can be used to better describe the data and mitigate this problem.

d) **Search and recommender system combination**: In practice, recommender systems and search are often combined. Combining them makes sense given the complementary benefits that they seem to offer, with a high accuracy in providing specific datasets to the user in search and a high coverage and novelty of potentially relevant data in the RecorDa approach. A hybrid of search and recommender systems would likely provide the best data to the user.

e) **Utility functions**: The current utility function is based on the usage that specific datasets provide to the user. However, a series of additional measures could be used, such as data quality or an analytics-based solution, in which the system uses a combination of datasets to predict a certain relevance, and the accuracy of the prediction determines the relevance of this combination.

f) **Additional approach variations**: Besides the architecture and setup presented in Chapters 5 and 6, there are many other variations of recommender systems and market approaches. These include changes to the recommender system components and to the market component, such as different market setups for example.

More research is also required to further evaluate the potential of this research.

g) **Scalability**: All evaluations in this thesis were conducted with a relatively small sample dataset, which was not always representative of the large data volumes found in industrial companies. Therefore, RecorDa needs to be tested on larger datasets. Time performance is key for systems once real users use them within companies[29]. There are existing solutions for scaling recommender systems that could be used to extend this research.

h) **Additional testing**: This thesis has established that the RecorDa approach can provide benefits for a user. Besides the initial evaluation in a series of experiments and case studies, additional implementations in other areas and well thought-out experiments with real users could be used to further evaluate the potential of the RecorDa approach.

i) **Evaluation based on interface design**: The limitations already show the issues regarding data repurposing and data presentation in the GUI for the RecorDa approach. As part of one of the case studies, this idea was considered to work with user interfaces instead of datasets. There are always a large variety of potential ways in which a dataset could be presented to the user. Instead of evaluating datasets, then, future work could consider evaluating GUIs and potentially allowing different GUI or data presentations to be shown to the user. In this way, not only the data but also the data presentation could be improved.

[29] The increase of search results by 200ms caused a reduction in google search queries by a user which even continues for several weeks even when the reduction is removed [223], [224]. This indicates that users are sensitive to timing and effort they have to take to do searches.

This thesis has developed an initial architecture for the RecorDa approach to user data allocation. It has evaluated the initial potential of applying these techniques in data management research. Future work needs to build on these initial findings to develop these technologies to a stage where they can be used within industrial companies.

References

[1] "The Big Data Refinery: Distilling intelligence from Big Data," *Database Netw. J.*, vol. 42, no. 4, p.8, Aug. 2012.

[2] J. Manyika, M. Chui, B. Brown, J. Bughin, R. Dobbs, C. Roxburgh, and A. Hung Byers, "Big data: The next frontier for innovation, competition, and productivity," McKinsey Global Institute, Report, May 2011.

[3] T. Baumgartner, H. Hatami, and J. V. Ark, *Sales Growth: Five Proven Strategies from the World's Sales Leaders*, 1 edition. Wiley, 2012.

[4] S. Feldman, "Records Management in the Age of Information Overload: 5 Tips for Finding What you Need," Optical Image Technology, Inc., State College, PA, USA, White Paper, Jul. 2010.

[5] J. Bughin, M. Chui, and J. Manyika, "Clouds, big data, and smart assets: Ten tech-enabled business trends to watch," *McKinsey Quaterly*, Aug-2010.

[6] N. Rowe, "Big data trends in 2013 - Can you handle your unstructured data?," *Big data trends in 2013, Can you handle your unstructured data?*, Feb-2013. [Online]. Available: http://www.aberdeen.com/Aberdeen-Library/8244/RA-big-data-trends.aspx. [Accessed: 25-Aug-2013].

[7] Rolls Royce, "Rolls-Royce monitoring systems," *Rolls-Royce monitoring systems*. [Online]. Available: http://www.rolls-royce.com/about/technology/systems_tech/monitoring_systems.jsp. [Accessed: 27-Aug-2013].

[8] Computerworld UK, "Boeing 787s to create half a terabyte of data per flight, says Virgin Atlantic." [Online]. Available: http://www.computerworlduk.com/news/infrastructure/3433595/boeing-787s-to-create-half-a-terabyte-of-data-per-flight-says-virgin-atlantic/. [Accessed: 27-Aug-2013].

[9] C. Dominguez, D. Cox, W. G. Long, B. Moon, and G. Klein, "Helping Analysts Deal with Data Overload: Profiling Profilers," *Proc. Hum. Factors Ergon. Soc. Annu. Meet.*, vol. 49, no. 10, pp. 908–912, Sep. 2005.

[10] A. Borchers, J. Herlocker, J. Konstan, and J. Reidl, "Ganging up on information overload," *Computer*, vol. 31, no. 4, pp. 106–108, Apr. 1998.

[11] M. J. Eppler and J. Mengis, "The Concept of Information Overload: A Review of Literature from Organization Science, Accounting, Marketing, MIS, and Related Disciplines," *Inf. Soc.*, vol. 20, no. 5, pp. 325–344, Nov. 2004.

[12] P. Hemp, "Death by Information Overload," *Harvard business review*, Sep-2009.

[13] E. Sperling, "Coping With Data Overload," *Forbes*, Dec-2008. [Online]. Available: http://www.forbes.com/2008/12/28/cio-emc-lewis-tech-cio-cx_es_1229emc.html. [Accessed: 14-Jul-2014].

[14] Z. A. Sabeeh and Z. Ismail, "Effects of information overload on productivity in enterprises: A literature review," in *2013 International Conference on Research and Innovation in Information Systems (ICRIIS),* Kuala Lumpur, 2013, pp. 210–214.

[15] A. J. Stanley and P. S. Clipsham, “Information overload-myth or reality?,” *IEE Colloq. IT Strateg. Inf. Overload*, pp. 1–4, Dec. 1997.

[16] Y. Morieux and P. Tollman, *Six Simple Rules: How to Manage Complexity without Getting Complicated*. Boston, Massachusetts: Harvard Business Review Press, 2014.

[17] R. K. Lahti, E. D. Darr, and V. E. Krebs, “Developing the productivity of a dynamic workforce: The impact of informal knowledge transfer,” *J. Organ. Excell.*, vol. 21, no. 2, pp. 13–21, 2002.

[18] G. P. Cachon and M. Fisher, “Supply chain inventory management and the value of shared information,” *Manag. Sci.*, vol. 46, no. 8, pp. 1032–1048, 2000.

[19] K. Smith, L. Seligman, and V. Swarup, “Everybody Share: The Challenge of Data-Sharing Systems,” *Computer*, vol. 41, no. 9, pp. 54–61, Sep. 2008.

[20] K. Gordon, *Principles of Data Management: Facilitating Information Sharing*. British Informatics Society Ltd, 2007.

[21] M. K. Khurana, P. K. Mishra, and A. R. Singh, “Barriers to Information Sharing in Supply Chain of Manufacturing Industries,” *Int. J. Manuf. Syst.*, vol. 1, no. 1, pp. 9–29, Jun. 2011.

[22] V. Dignum and F. Dignum, “The knowledge market: Agent-mediated knowledge sharing,” in *Multi-Agent Systems and Applications III*, vol. 2691, Springer, 2003, pp. 168–179.

[23] Y. Crama, R. Pascual J, and A. Torres, “Optimal procurement decisions in the presence of total quantity discounts and alternative product recipes,” *Eur. J. Oper. Res.*, vol. 159, no. 2, pp. 364–378, Dec. 2004.

[24] K. Wei, J. Huang, and S. Fu, “A survey of e-commerce recommender systems,” in *2007 International Conference on Service Systems and Service Management*, Chengdu, China, 2007, pp. 1–5.

[25] E. Bothos, K. Christidis, D. Apostolou, and G. Mentzas, “Information market based recommender systems fusion,” in *Proceedings of the 2nd International Workshop on Information Heterogeneity and Fusion in Recommender Systems*, 2011, pp. 1–8.

[26] L. Zhen, G. Q. Huang, and Z. Jiang, “An inner-enterprise knowledge recommender system,” *Expert Syst. Appl.*, vol. 37, no. 2, pp. 1703–1712, Mar. 2010.

[27] Y. Xu, P. Scerri, K. Sycara, and M. Lewis, “Comparing market and token-based coordination,” in *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, Hakodate, Hokkaido, Japan., 2006, pp. 1113–1115.

[28] A. Koroni, T. Redman, and J. Gao, “Internal Data Markets: The Opportunity and First Steps,” in *COINFO ’09 Proceedings of the 2009 Fourth International Conference on Cooperation and Promotion of Information Resources in Science and Technology*, Beijing, China, 2009, pp. 127–130.

[29] P. Upadhyaya, M. Balazinska, and D. Suciu, “How to price shared optimizations in the cloud,” *Proc. VLDB Endow.*, vol. 5, no. 6, pp. 562–573, Feb. 2012.

[30] S. J. Rassenti, V. L. Smith, and R. L. Bulfin, “A Combinatorial Auction Mechanism for Airport Time Slot Allocation,” *Bell J. Econ.*, vol. 13, no. 2, pp. 402–417, Autumn 1982.

[31] O. Ercetin and L. Tassiulas, “Market-based resource allocation for content delivery in the internet,” *IEEE Trans. Comput.*, vol. 52, no. 12, pp. 1573–1585, Dec. 2003.

[32] R. Buyya, D. Abramson, and S. Venugopal, “The Grid Economy,” *Proc. IEEE*, vol. 93, no. 3, pp. 698–714, Mar. 2005.

[33] J. Stößer, *Market-Based Scheduling in Distributed Computing Systems*. 2009.

[36] T. Wang, Z. Lin, B. Yang, J. Gao, A. Huang, D. Yang, Q. Zhang, S. Tang, and J. Niu, “MBA: A market-based approach to data allocation and dynamic migration for cloud database,” Sci. China Inf. Sci., vol. 55, no. 9, pp. 1935–1948, Mar. 2012.

[35] K. Kwiat, “Using markets to engineer resource management for the information grid,” *Inf. Syst. Front.*, vol. 4, no. 1, pp. 55–62, Apr. 2002.

[36] R. Burke, A. Felfernig, and M. H. Göker, “Recommender systems: An overview,” *AI Mag.*, vol. 32, no. 3, pp. 13–18, Fall 2011.

[37] M. Mosley, M. Brackett, S. Earley, and D. Hendersson, *The DAMA Guide to the Data Management Body of Knowledge (DAMA-DMBOK) Print Edition*, First edition. Bradley Beach, New Jersey: Technics Publications, LLC, 2010.

[38] F. Machlup, *The Production and Distribution of Knowledge in the United States*. 1962.

[39] D. B. Lawrence, *The economic value of information*. New York: Springer, 1999.

[40] A. A. A. Ismat and M. Torres-Dela Cruz, “Maximizing Open Source Applications in Developing University Information Systems,” *Int. J. Inf. Syst. Eng.*, vol. 1, no. 1, Apr. 2015.

[41] R. Ramakrishnan and J. Gehrke, *Database Management Systems*, 3rd edition. McGraw-Hill Higher Education, 2002.

[42] Z. Zaier, R. Godin, and L. Faucher, “Evaluating Recommender Systems,” presented at the International Conference on Automated solutions for Cross Media Content and Multi-channel Distribution, Florence, Italy, 2008, pp. 211–217.

[43] “Data, data everywhere - A special report on managing information,” *The Economist*, 27-Feb-2010.

[44] G. Moore, “Excerpts from a Conversation with Gordon Moore: Moore’s Law,” *Standofrd university*, 2005. [Online]. Available: http://large.stanford.edu/courses/2012/ph250/lee1/docs/Excepts_A_Conversation_with_Gordon_Moore.pdf. [Accessed: 06-Sep-2012].

[45] G. E. Moore, “The Future of Integrate Electronics,” A Division of Fairchild Camera & Instrument Corporation, Palo Alto, CA, USA, Internal report.

[46] G. E. Moore, “Cramming More Components Onto Integrated Circuits,” *Proc. IEEE*, vol. 86, no. 1, pp. 82–85, Jan. 1998.

[47] A. Watters, “The age of exabytes - Tools and approaches for managing big data,” ReadWriteWeb, HP Networking, 4AA0-7725ENN, Jun. 2010.

[48] P. P. Tallon and R. Scannell, “Information Life Cycle Management,” *Commun. ACM*, vol. 50, no. 11, pp. 65–69, Nov. 2007.

[49] The Economist, “Building with big data - The data revolution is changing the landscape of business,” May-2011. [Online]. Available: http://www.economist.com/node/18741392. [Accessed: 25-Aug-2013].

[50] A. Brintrup, D. McFarlane, D. Ranasinghe, T. Sanchez Lopez, and K. Owens, “Will Intelligent Assets Take Off? Toward Self-Serving Aircraft,” *IEEE Intell. Syst.*, vol. 26, no. 3, pp. 66–75, May 2011.

[51] J.-L. Hou and C.-H. Huang, “Quantitative performance evaluation of RFID applications in the supply chain of the printing industry,” *Ind. Manag. Data Syst.*, vol. 106, no. 1, pp. 96–120, 2006.

[52] “Large Data Sets Repository | Public Data Sets with AWS,” *Amazon Web Services, Inc.* [Online]. Available: //aws.amazon.com/public-data-sets/. [Accessed: 12-Oct-2015].

[53] “Data.gov,” *Data.gov*. [Online]. Available: https://www.data.gov/. [Accessed: 13-May-2017].

[54] “data.gov.uk,” *data.gov.uk*. [Online]. Available: https://data.gov.uk/. [Accessed: 13-May-2017].

[55] T. Berners-Lee, J. Hendler, and O. Lassila, “The Semantic Web,” *Scientific American Magazine*, pp. 29–37, May-2001.

[56] L. Mearian, “World’s data will grow by 50X in next decade, IDC study predicts,” *Computerworld*, 28-Jun-2011. [Online]. Available: http://www.computerworld.com/article/2509588/data-center/world-s-data-will-grow-by-50x-in-next-decade--idc-study-predicts.html. [Accessed: 12-Oct-2015].

[57] S. Loertscher and M. H. Riordan, “Outsourcing, Vertical Integration, and Cost Reduction,” University of Columbia, Working Paper, Nov. 2014.

[58] T. H. Davenport and J. G. Harris, *Competing on Analytics: The New Science of Winning*, 1st ed. Harvard Business School Press, 2007.

[59] A. Berson and L. Dubov, *Master data management and data governance*, Second edition. New York: McGraw-Hill, 2011.

[60] Michael L. Brodie, “Data Management Challenges in Very Large Enterprises,” in *Proceedings of the 28th VLDB Conference*, Hong Kong, China, 2002.

[61] K. Eswaran, “Placement of Records in a File and File Allocation in a Computer Network,” in *In Proceedings of IFIP Conference*, Stockholm, Sweden, 1974, pp. 304–307.

[62] I. N. Wang, N. J. Fiddian, and W. A. Gray, “Market-based agent allocation in global information systems,” in *Proceedings of the fifth international conference on Autonomous agents*, New York, NY, USA, 2001, pp. 67–68.

[63] M. Brydon, “Economic metaphors for solving intrafirm allocation problems: What does a market buy us?,” *Decis. Support Syst.*, vol. 42, no. 3, pp. 1657–1672, Dec. 2006.

[64] P. A. Strassmann, “The Value Of Computers, Information and Knowledge,” *http://www.strassmann.com/*, Jan-1996. [Online]. Available: www.strassmann.com/pubs/cik/cik-value.shtml. [Accessed: 13-May-2017].

[65] W. R. King, V. Grover, and E. H. Hufnagel, “Using information and information technology for sustainable competitive advantage: some empirical evidence,” *Inf. Manage.*, vol. 17, no. 2, pp. 87–93, 1989.

[66] M. E. Porter and V. A. Millar, “How information gives you competitive advantage,” *Harvard business review*, pp. 149–174, Aug-1985.

[67] T. J. Mock, “Concepts of information value and accounting,” *Account. Rev.*, vol. 46, no. 4, pp. 765–778, 1971.

[68] C. Pfeifer, K. Schredelseker, and G. U. H. Seeber, “On the negative value of information in informationally inefficient markets: Calculations for large number of traders,” *Eur. J. Oper. Res.*, vol. 195, no. 1, pp. 117–126, 2009.

[69] P. O. Christensen and G. Feltham, *Economics of Accounting: Volume I: Information in Markets*. Springer, 2004.

[70] A. K. Parlikad and D. McFarlane, “Value of information in product recovery decisions: a Bayesian approach,” *Int. J. Sustain. Eng.*, vol. 3, no. 2, pp. 106–120, 2010.

[71] J. Marschak, “Economics of Information Systems,” *J. Am. Stat. Assoc.*, vol. 66, no. 333, pp. 192–219, Mar. 1971.

[72] C. Shapiro and H. R. Varian, *Information Rules: A Strategic Guide to the Network Economy*, 1ST ed. Harvard Business Review Press, 1998.

[73] M. Feeney and M. Grieves, *The Value and Impact of Information*. Bowker-Saur, 1994.

[74] R. A. Howard, “Information Value Theory,” *IEEE Trans. Syst. Sci. Cybern.*, vol. 2, no. 1, pp. 22–26, Aug. 1966.

[75] R. A. Howard, “The foundations of decision analysis,” *Syst. Sci. Cybern. IEEE Trans. On*, vol. 4, no. 3, pp. 211–219, 1968.

[76] A. Kulkarni, D. Ralph, and D. McFarlane, “Value of RFID in remanufacturing,” *Int. J. Serv. Oper. Inform.*, vol. 2, no. 3, pp. 225–252, 2007.

[77] N. Ahituv, “A metamodel of information flow: a tool to support information systems theory,” *Commun. ACM*, vol. 30, no. 9, pp. 781–791, 1987.

[78] T. Kelepouris and D. McFarlane, “Determining the value of asset location information systems in a manufacturing environment,” *Int. J. Prod. Econ.*, vol. 126, no. 2, pp. 324–334, 2010.

[79] M. Harrison, D. McFarlane, A. K. Parlikad, and C. Y. Wong, “Information management in the product lifecycle-the role of networked RFID,” in *Industrial Informatics, 2004. INDIN’04. 2004 2nd IEEE International Conference on*, 2004, pp. 507–512.

[80] R. Glazer, “Measuring the value of information: The information-intensive organization,” *IBM Syst. J.*, vol. 32, no. 1, pp. 99–110, 1993.

[81] U.S. Department of Transportation, “Value of Information and Information Services.” Office of Technology Applications, Federal Highway Administration, by Susan C. Dresley, EG&G Services, and Annalynn

Lacombe, Transportation Strategic Planning and Analysis Office, at the Volpe National Transportation Systems Center, Research and Special Programs Administration., Oct-1998.

[82] B. J. Epstein and W. R. King, “An experimental study of the value of information,” *Omega Lnternational J. Manag. Sci.*, vol. 10, no. 3, pp. 249–258, 1982.

[83] A. Ragowsky, N. Ahituv, and S. Neumann, “Identifying the value and importance of an information system application,” *Inf. Manage.*, vol. 31, no. 2, pp. 89–102, 1996.

[84] J. Matthews, “The Value of Information in Library Catalogs.” Special Libraries Association, Jul-2000.

[85] B. Szymanski, S. Y. Shah, S. C. Geyik, S. Das, M. Chhabra, and P. Zerfos, “Market mechanisms for value of information driven resource allocation in sensor networks,” in *2011 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops),* Seattle, WA, USA, 2011, pp. 20–25.

[86] S. C. Geyik, S. Y. Shah, B. K. Szymanski, S. Das, and P. Zerfos, “Market mechanisms for resource allocation in pervasive sensor applications,” *Pervasive Mob. Comput.*, vol. 8, no. 3, pp. 346–357, Jun. 2012.

[87] V. Avasarala, T. Mullen, and D. L. Hall, “A Market-Based Sensor Management Approach,” *Microsoft Academic Research - A Market-Based Sensor Management Approach*. [Online]. Available: http://tmullen.ist.psu.edu/pubs/masm_2007.pdf. [Accessed: 25-Aug-2013].

[88] Y. Yemini, A. Dailianas, and D. Florissi, “MarketNet: A Survivable, Market-Based Architecture for Large-Scale Information Systems,” DTIC Document, Final Technical Report ADA407306, Aug. 2002.

[89] F. Wijnhoven, C. Amrit, and P. Dietz, “Value-Based File Retention: File Attributes as File Value and Information Waste Indicators,” *ACM J. Data Inf. Qual.*, vol. 4, no. 4, p. Article No. 16, 1-17, May 2014.

[90] G. Shah, K. Voruganti, P. Shivam, and M. del Mar Alvarez Rohena, “ACE: Classification for Information Lifecycle Management,” IBM Research, Technical Paper RJ10372, 2006.

[91] J. Huber, M. Kirchler, and M. Sutter, “Is more information always better?: Experimental financial markets with cumulative information,” *J. Econ. Behav. Organ.*, vol. 65, no. 1, pp. 86–104, 2008.

[92] M. Balazinska, B. Howe, and D. Suciu, “Data markets in the cloud: An opportunity for the database community,” *Proc VLDB Endow.*, vol. 4, p. 12, 2011.

[93] A. Mowshowitz, “On the market value of information commodities. I. The nature of information and information commodities,” *J. Am. Soc. Inf. Sci.*, vol. 43, no. 3, pp. 225–232, Apr. 1992.

[94] N. Glance, D. Arregui, and M. Dardenne, “Making recommender systems work for organizations,” in *Proceedings of PAAM’99*, London, United Kingdom, 1999, pp. 19–21.

[95] L. Page, S. Brin, R. Motwani, and T. Winograd, “The PageRank citation ranking: bringing order to the web.,” Stanford InfoLab, 1999–66, 1999.

[96] J. M. Kleinberg, “Authoritative sources in a hyperlinked environment,” *J. ACM JACM*, vol. 46, no. 5, pp. 604–632, Sep. 1999.

[97] R. Lempel and S. Moran, “The Stochastic Approach for Link-Structure Analysis (SALSA) and the TKC Effect,” *Comput. Netw.*, vol. 33, no. 1, pp. 387–401, May 2000.

[98] “Google Enterprise Search,” *Google Enterprise Search*. [Online]. Available: http://www.google.com/enterprise/search/. [Accessed: 26-Aug-2013].

[99] L. Duan and L. D. Xu, “Business Intelligence for Enterprise Systems: A Survey,” *IEEE Trans. Ind. Inform.*, vol. 8, no. 3, pp. 679–687, Aug. 2012.

[100] H. P. Luhn, “A Business Intelligence System,” *IBM J. Res. Dev.*, vol. 2, no. 4, pp. 314–319, Oct. 1958.

[101] R. Sherman, *Business Intelligence Guidebook: From Data Integration to Analytics*. Amsterdam: Morgan Kaufmann, 2014.

[102] B. Shneiderman, “The eyes have it: a task by data type taxonomy for information visualizations,” in *Proceedings of IEEE Symposium on Visual Languages*, Boulder, CO, USA, 1996, pp. 336–343.

[103] Z. Wen and M. X. Zhou, “Evaluating the Use of Data Transformation for Information Visualization,” *IEEE Trans. Vis. Comput. Graph.*, vol. 14, no. 6, pp. 1309–1316, Nov. 2008.

[104] A. Smith, *The Wealth of Nations - An Inquiry into the Nature and Causes of the Wealth of Nations*, vol. 1. London: W. Strahan and T. Cadell, 1776.

[105] P. Tucker and F. Bermany, “On Market Mechanisms as a Software Technique,” Department of Computer Science and Engineering, University of California, San Diego, San Diego, California, USA, Technical Report CS96-513, Dec. 1996.

[106] S. H. Clearwater, *Market based control: a paradigm for distributed resource allocation*. World Scientific Publishing Company, 1996.

[107] Y. Yemini, A. Dailianas, and D. Florissi, “MarketNet: A market-based architecture for survivable large-scale information systems,” in *Proceedings of Fourth ISSAT International Conference on Reliability and Quality in Design*, 1998, pp. 8–13.

[108] T. Kaihara and S. Fujii, “A study on modeling methodology of oligopolistic virtual market and its application into resource allocation problem,” *Electr. Eng. Jpn.*, vol. 164, no. 1, pp. 77–85, Jul. 2008.

[109] V. Krishna, *Auction Theory, Second Edition*, 2 edition. Burlington, MA: Academic Press, 2009.

[110] J. K. MacKie-Mason, W. E. Walsh, M. P. Wellman, and P. Wurman, “Some Economics of Market-Based Distributed Scheduling,” in *The 18th International Conference on Distributed Computing Systems (ICDCS’98)*, Amsterdam, The Netherlands, 1998, pp. 612–621.

[111] P. Klemperer, “Auction Theory: A Guide to the Literature,” *J. Econ. Surv.*, vol. 13, no. 3, pp. 227–286, Jul. 1999.

[112] P. Klemperer, *Auctions: Theory and Practice*. Princeton University Press, 2004.

[113] T. Sandholm, “Algorithm for optimal winner determination in combinatorial auctions,” *Artif. Intell.*, vol. 135, no. 1–2, pp. 1–54, Feb. 2002.

[114] F. Kelly and R. Steinberg, “A combinatorial auction with multiple winners for universal service,” *Manag. Sci.*, vol. 46, no. 4, pp. 586–596, Apr. 2000.

[115] P. Cramton, Y. Shoham, V. L. Smith, and R. Steinberg, *Combinatorial Auctions*. MIT Press, 2010.

[116] D. Porter, S. Rassenti, A. Roopnarine, and V. Smith, “Combinatorial auction design,” *Proc. Natl. Acad. Sci.*, vol. 100, no. 19, pp. 11153–11157, Jul. 2003.

[117] S. de Vries and R. V. Vohra, “Combinatorial Auctions: A Survey,” *Inf. J. Comput.*, vol. 15, no. 3, pp. 284–309, Aug. 2003.

[118] M. H. Rothkopf, A. Pekec, and R. M. Harstad, “Computationally Manageable Combinatorial Auctions,” *Manag. Sci.*, vol. 44, no. 8, pp. 1131–1147, Aug. 1998.

[119] A. AuYoung, “Practical market-based resource allocation,” Doctoral disertation, University of California, San Diego, San Diego, California, USA, 2010.

[120] A. V. Goldberg, J. D. Hartline, A. R. Karlin, M. Saks, and A. Wright, “Competitive auctions,” *Games Econ. Behav.*, vol. 55, no. 2, pp. 242–269, May 2006.

[121] J. Q. Cheng and M. P. Wellman, “The WALRAS Algorithm: A Convergent Distributed Implementation of General Equilibrium Outcomes,” *Comput. Econ.*, vol. 12, no. 1, pp. 1–24, Aug. 1998.

[122] J. Huang, R. A. Berry, and M. L. Honig, “Auction-based Spectrum Sharing,” *Mob Netw Appl*, vol. 11, no. 3, pp. 405–418, Jun. 2006.

[123] B. Virginas, C. Voudouris, G. Owusu, and G. Anim-Ansah, “ARMS Collaborator—intelligent agents using markets to organise resourcing in modern enterprises,” *BT Technol. J.*, vol. 21, no. 4, pp. 59–64, Oct. 2003.

[124] C. Gerber, C. Ruß, and G. Vierke, “On the suitability of market-based mechanisms for telematics applications,” in *Proceedings of the third annual conference on Autonomous Agents*, New York, NY, USA, 1999, pp. 408–409.

[125] G. Jonker, J.-J. Meyer, and F. Dignum, “Towards a market mechanism for airport traffic control,” in *Progress in Artificial Intelligence*, vol. 3808, Springer, 2005, pp. 500–511.

[126] G. Confessore, S. Giordani, and S. Rismondo, “A market-based multi-agent system model for decentralized multi-project scheduling,” *Ann. Oper. Res.*, vol. 150, no. 1, pp. 115–135, Feb. 2007.

[127] M. B. Dias, R. Zlot, N. Kalra, and A. Stentz, “Market-Based Multirobot Coordination: A Survey and Analysis,” *Proc. IEEE*, vol. 94, no. 7, pp. 1257–1270, Jul. 2006.

[128] B. P. Gerkey and M. J. Mataric, “Sold!: auction methods for multirobot coordination,” *IEEE Trans. Robot. Autom.*, vol. 18, no. 5, pp. 758–768, Oct. 2002.

[129] H. Voos, “Market-based control of complex dynamic systems,” in *Proceedings of the 1999 IEEE International Symposium on*, Cambridge, MA, USA, 1999, pp. 284–289.

[130] Y. Yemini, A. Dailianas, D. Florissi, and G. Huberman, “MarketNet: protecting access to information systems through financial market controls,” *Decis. Support Syst. - Spec. Issue Inf. Comput. Econ.*, vol. 28, no. 1–2, pp. 205–216, Mar. 2000.

[131] Y. Yemini, A. Dailianas, and D. Florissi, “Marketnet: Using virtual currency to protect information systems,” *Res. Adv. Technol. Digit. Libr.*, pp. 518–518, 1998.

[132] A. Dailianas, Y. Yemini, D. Florissi, and H. Huang, “Marketnet: Market-based protection of network systems and services-an application to snmp protection,” in *Proceedings INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies.*, Tel Aviv, Israel, 2000, vol. 3, pp. 1391–1400.

[133] N. Haque, N. R. Jennings, and L. Moreau, “Scalability and robustness of a market-based network resource allocation system,” *NETNOMICS Econ. Res. Electron. Netw.*, vol. 7, no. 2, pp. 69–96, Apr. 2005.

[134] M. Fan, J. Stallaert, and A. B. Whinston, “Decentralized mechanism design for supply chain organizations using an auction market,” *Inf. Syst. Res.*, vol. 14, no. 1, pp. 1–22, Mar. 2003.

[135] P. M. Markopoulos, R. Aron, and L. Ungar, “Information markets for product attributes: A game theoretic, dual pricing mechanism,” *Decis. Support Syst.*, vol. 49, no. 2, pp. 187–199, May 2010.

[136] S. Ba, J. Stallaert, and A. B. Whinston, “Optimal Investment in Knowledge within a Firm Using a Market Mechanism,” *Manag. Sci.*, vol. 47, no. 9, pp. 1203–1219, Sep. 2001.

[137] Jason Magidson and Andrew E. Polcha, “Creating Market Economies Within Companies,” *J. Bus. Strategy*, vol. 13, no. 3, pp. 39–44, Mar. 1992.

[138] D. Lehmann, R. Müller, and T. Sandholm, “The winner determination problem,” *Comb. Auctions*, pp. 297–317, 2006.

[139] G. Koifman, O. Shehory, and A. Gal, “Multi-agent negotiation and price discrimination for information goods,” in *Proceedings of the 2004 IEEE International Conference on Systems, Man and Cybernetics.*, The Hague, Netherlands, 2004, vol. 2, pp. 1964–1971.

[140] M. Christoffel, “Information Integration as a Matter of Market Agents,” in *Proceedings of the 4th International Conference on Electronic Commerce Research*, Montreal, Canada, 2002.

[141] F. Wijnhoven, E. Van Den Belt, E. Verbruggen, and P. E. van der Vet, “Internal data market services: an ontology-based architecture and its evaluation,” *Informing Sci. J.*, vol. 6, pp. 259–271, 2003.

[142] P. Koutris, P. Upadhyaya, M. Balazinska, B. Howe, and D. Suciu, “Toward practical query pricing with querymarket,” in *Proceedings of the 2013 international conference on Management of data*, New York, NY, USA, 2013, pp. 613–624.

[143] P. Koutris, P. Upadhyaya, M. Balazinska, B. Howe, and D. Suciu, “Query-based Data Pricing,” in *Proceedings of the 31st Symposium on Principles of Database Systems*, Scottsdale, AZ, USA, 2012, pp. 167–178.

[144] M. Balazinska, B. Howe, P. Koutris, D. Suciu, and P. Upadhyaya, “A discussion on pricing relational data,” in *In Search of Elegance in the Theory and Practice of Computation*, vol. 8000, Springer, 2013, pp. 167–173.

[145] G. Linden, B. Smith, and J. York, “Amazon.com recommendations: item-to-item collaborative filtering,” *IEEE Internet Comput.*, vol. 7, no. 1, pp. 76–80, Feb. 2003.

[146] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry, “Using collaborative filtering to weave an information tapestry,” *Commun. ACM - Spec. Issue Inf. Filter.*, vol. 35, no. 12, pp. 61–70, Dec. 1992.

[147] D. H. Park, H. K. Kim, I. Y. Choi, and J. K. Kim, “A literature review and classification of recommender systems research,” *Expert Syst. Appl.*, vol. 39, no. 11, pp. 10059–10072, Sep. 2012.

[148] S. Perugini, M. A. Goncalves, and E. A. Fox, “Recommender systems research: A connection-centric survey,” *J. Intell. Inf. Syst.*, vol. 23, no. 2, pp. 107–143, Sep. 2004.

[149] R. M. Bell, Y. Koren, and C. Volinsky, “The BellKor solution to the Netflix Prize,” AT&T Labs – Research, Technical Report, Oct. 2007.

[150] L. Chen, M. de Gemmis, A. Felfernig, P. Lops, F. Ricci, and G. Semeraro, “Human Decision Making and Recommender Systems,” *ACM Trans. Interact. Intell. Syst.*, vol. 3, no. 3, pp. 1–7, Oct. 2013.

[151] D. Jannach, M. Zanker, M. Ge, and M. Gröning, “Recommender Systems in Computer Science and Information Systems – A Landscape of Research,” in *Lecture Notes in Business Information Processing*, vol. 123, Springer, 2012.

[152] C. Porcel, J. M. del Castillo, M. J. Cobo, A. A. Ruız, and E. Herrera-Viedma, “An improved recommender system to avoid the persistent information overload in a university digital library,” *Control Cybern.*, vol. 39, no. 4, pp. 899–924, Oct. 2010.

[155] I. Guy, A Jaimes, P. Agullo, P. Moore, P. Nandy, C. Nastar, and H. Schinzel, “Will recommenders kill search?: recommender systems-an industry perspective,” in Proceedings of the fourth ACM conference on Recommender systems, Barcelona, Spain, 2010, pp. 7–12.

[154] J. B. Schafer, J. Konstan, and J. Riedl, “Recommender Systems in e-Commerce,” in *Proceedings of the 1st ACM Conference on Electronic Commerce*, New York, NY, USA, 1999, pp. 158–166.

[155] H. Elsner and J. Krämer, “Managing corporate portal usage with recommender systems,” *Bus. Inf. Syst. Eng.*, vol. 5, no. 4, pp. 213–225, Apr. 2013.

[156] D. Bahls, G. Scherp, K. Tochtermann, and W. Hasselbring, “Towards a Recommender System for Statistical Research Data,” in *Proceedings of the 2nd International Workshop on Semantic Digital Archives*, Paphos, Cyprus, 2012, vol. 912, pp. 61–72.

[157] G. Chatzopoulou, M. Eirinaki, and N. Polyzotis, “Query recommendations for interactive database exploration,” in *Proceedings of the 21st International Conference on Scientific and Statistical Database Management*, New Orleans, Louisiana, USA, 2009, pp. 3–18.

[158] R. Meymandpour and J. G. Davis, “Recommendations using linked data,” in *Proceedings of the 5th Ph. D. workshop on Information and knowledge*, Maui, Hawaii, USA, 2012, pp. 75–82.

[159] A. Passant, B. Heitmann, and C. Hayes, “Using linked data to build recommender systems,” in *Proceedings of the 3rd ACM Conference on Recommender Systems*, New York, New York, USA, 2009.

[160] E. Costa-Montenegro, A. B. Barragáns-Martínez, and M. Rey-López, “Which App? A recommender system of applications in markets: Implementation of the service for monitoring users’ interaction,” *Expert Syst. Appl.*, vol. 39, no. 10, pp. 9367–9375, Aug. 2012.

[161] Y. Z. Wei, L. Moreau, and N. R. Jennings, “Learning users’ interests by quality classification in market-based recommender systems,” *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 12, pp. 1678–1688, Dec. 2005.

[162] Y. Z. Wei, L. Moreau, and N. R. Jennings, “Learning users’ interests in a market-based recommender system,” in *Intelligent Data Engineering and Automated Learning–IDEAL 2004*, vol. 3177, Springer, 2004, pp. 833–840.

[163] Y. Z. Wei, L. A. V. Moreau, and N. R. Jennings, “Market-based recommendations: Design, simulation and evaluation,” in *Agent-Oriented Information Systems*, vol. 3030, 2003, pp. 61–77.

[164] Y. Z. Wei, L. Moreau, and N. R. Jennings, “Recommender systems: a market-based design,” in *Proceedings of the second international joint conference on Autonomous agents and multiagent systems*, Melbourne, Australia, 2003, pp. 600–607.

[165] Y. Z. Wei, L. Moreau, and N. R. Jennings, “Market-based recommender systems: Learning users’ interests by quality classification,” in *Proceedings of the Six International Workshop on Agent-Oriented Information Systems (AOIS-2004)*, New York, New York, USA, 2004, pp. 119–133.

[166] S. M. Bohte, E. Gerding, and H. L. Poutré, “Market-based recommendation: Agents that compete for consumer attention,” *ACM Trans. Internet Technol. TOIT*, vol. 4, no. 4, pp. 420–448, Nov. 2004.

[167] S. Bohte, E. H. Gerding, and H. La Poutré, “Competitive Market-based Allocation of Consumer Attention Space: Concepts and Validation of Casy,” in *Proceedings of the 3rd ACM Conference on Electronic Commerce (EC-01)*, Tampa, Florida, USA, 2001, pp. 202–206.

[168] D. Melamed, B. Shapira, and Y. Elovici, “MarCol: A market-based recommender system,” *IEEE Intell. Syst.*, vol. 22, no. 3, pp. 74–78, Jun. 2007.

[169] P. Drineas, I. Kerenidis, and P. Raghavan, “Competitive recommendation systems,” in *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, Montreal, QC, Canada, 2002, pp. 82–90.

[172] L. Moreau, N. Zaini, J. Zhou, N.R. Jennings, Y.Z. Wei, W. Hall, D. De Roure, I. Gilchrist, M. O'Dell, S. Reich, T. Berka, and C. Di Napoli, “A Market-Based Recommender System,” presented at the The Fourth International Bi-Conference Workshop on Agent-Oriented Information Systems at AAMAS 2002 (AOIS’02), Bologna, Italy, 2002, pp. 50–67.

[171] X. Yuan, J.-H. Lee, S.-J. Kim, and Y.-H. Kim, “Toward a user-oriented recommendation system for real estate websites,” *Inf. Syst.*, vol. 38, no. 2, pp. 231–243, Apr. 2013.

[172] B. N. Miller, J. A. Konstan, and J. Riedl, “PocketLens: Toward a personal recommender system,” *ACM Trans. Inf. Syst. TOIS*, vol. 22, no. 3, pp. 437–476, Jul. 2004.

[173] J. M. Vidal, “A protocol for a distributed recommender system,” in *Trusting Agents for Trusting Electronic Societies*, vol. 3577, Springer, 2005, pp. 200–217.

[174] M.-C. Chen, L.-S. Chen, F.-H. Hsu, Y. Hsu, and H.-Y. Chou, “HPRS: A profitability based recommender system,” in *2007 IEEE International Conference on Industrial Engineering and Engineering Management*, Singapore, Singapore, 2007, pp. 219–223.

[175] Y.-M. Li and C.-P. Kao, “TREPPS: A Trust-based Recommender System for Peer Production Services,” *Expert Syst. Appl.*, vol. 36, no. 2, pp. 3263–3277, Mar. 2009.

[176] M. Easterby-Smith, R. Thorpe, P. Jackson, and A. Lowe, *Management Research: An Introduction*, Third Edition. Sage Publications, 2008.

[177] V. J. Wass, *Principles and Practice in Business and Management Research*. Brookfield, VT, USA: Dartmouth Publishing Company, 1994.

[178] I. Vessey, V. Ramesh, and R. L. Glass, “Research in Information Systems: An Empirical Study of Diversity in the Discipline and Its Journals,” *J. Manag. Inf. Syst.*, vol. 19, no. 2, pp. 129–174, Oct. 2002.

[179] R. K. Yin, *Case Study Research: Design and Methods*, vol. 5. SAGE, 2013.

[180] B. Kitchenham, L. Pickard, and S. L. Pfleeger, “Case studies for method and tool evaluation,” *IEEE Softw.*, vol. 12, no. 4, pp. 52–62, Jul. 1995.

[181] S. L. Pfleeger, “Experimental design and analysis in software engineering,” *Ann. Softw. Eng.*, vol. 1, no. 1, pp. 219–253, Dec. 1995.

[182] K. M. Eisenhardt, “Building Theories from Case Study Research,” *Acad. Manage. Rev.*, vol. 14, no. 4, pp. 532–550, Oct. 1989.

[183] C. Voss, N. Tsikriktsis, and M. Frohlich, “Case research in operations management,” *Int. J. Oper. Prod. Manag.*, vol. 22, no. 2, pp. 195–219, 2002.

[184] M. B. Miles, A. M. Huberman, and J. M. Saldana, *Qualitative Data Analysis: A Methods Sourcebook*, 3rd edition. Thousand Oaks, Califorinia: Sage Publications Ltd., 2013.

[185] T. D. Jick, “Mixing Qualitative and Quantitative Methods: Triangulation in Action,” *Adm. Sci. Q.*, vol. 24, no. 4, p. 602, Dec. 1979.

[186] V. R. Basili, R. W. Selby, and D. H. Hutchens, “Experimentation in Software Engineering,” *IEEE Trans. Softw. Eng.*, vol. 12, no. 7, pp. 733–743, Jul. 1986.

[187] L. Barroca, J. Hall, and P. Hall, *Software Architectures: Advances and Applications*, 2000 edition. 1999.

[188] K. Wiegers and J. Beatty, *Software Requirements*, 3erd edition. 2013.

[189] I. Sommerville, *Software Engineering*, 9 edition. Pearson, 2010.

[190] H. Cervantes and R. Kazman, *Designing Software Architectures: A Practical Approach*, 1st ed. Addison Wesley, 2016.

[191] G. Adomavicius and A. Tuzhilin, “Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions,” *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 6, pp. 734–749, Jun. 2005.

[192] D. Jannach, M. Zanker, A. Felfernig, and G. Friedrich, *Recommender Systems: An Introduction*. New York: Cambridge University Press, 2010.

[193] S. H. Clearwater, *Market-Based Control - A Paradigm for Distributed Resource Allocation*. Paolo Alto, CA, USA: Xerox Palo Alto Research Center, 1996.

[194] The Apache Software Foundation, “Mahout library,” *Apache Mahout: Scalable machine learning and data mining*, 2014. [Online]. Available: http://mahout.apache.org/. [Accessed: 24-Mar-2015].

[195] E. Rahm and P. A. Bernstein, “A survey of approaches to automatic schema matching,” *VLDB J.*, vol. 10, no. 4, pp. 334–350, Nov. 2001.

[196] N. Koudas, A. Marathe, and D. Srivastava, “Flexible String Matching Against Large Databases in Practice,” in *Proceedings of the Thirtieth International Conference on Very Large Data Bases - Volume 30*, Toronto, Canada, 2004, pp. 1078–1086.

[197] S. Owen, R. Anil, T. Dunning, and E. Friedman, *Mahout in Action*, 1st ed. Shelter Island, New York: Manning Publications, 2011.

[198] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, “Item-based Collaborative Filtering Recommendation Algorithms,” in *Proceedings of the 10th International Conference on World Wide Web*, Hong Kong, China, 2001, pp. 285–295.

[199] S.-H. Cha, “Comprehensive Survey on Distance/Similarity Measures between Probability Density Functions,” *Int. J. Math. Models Methods Appl. Sci.*, vol. 1, pp. 300--307, 2007.

[200] D. R. Raban, “User-centered evaluation of information: a research challenge,” *Internet Res.*, vol. 17, no. 3, pp. 306–322, Jun. 2007.

[201] G. A. Miller, “The magical number seven, plus or minus two: some limits on our capacity for processing information.,” *Psychol. Rev.*, vol. 63, no. 2, pp. 81–97, 1956.

[202] V. Köbberling, “Strength of Preference and Cardinal Utility,” *Econ. Theory*, vol. 27, no. 2, pp. 375–391, Feb. 2006.

[203] “Cardinal Utility,” *Am. Econ. Rev.*, vol. 43, no. 2, pp. 384–397, May 1953.

[204] D. Ellsberg, “Classic and Current Notions of ‘Measurable Utility,’” *Econ. J.*, vol. 64, no. 255, pp. 528–556, Sep. 1954.

[205] L. Blume, D. Easley, J. Kleinberg, R. Kleinberg, and E. Tardos, “Introduction to computer science and economic theory,” *J. Econ. Theory*, vol. 156, pp. 1–13, Mar. 2015.

[206] Bloomberg Finance L.P, “Optimize the Future,” 2013.

[207] A. Gunawardana and G. Shani, “A Survey of Accuracy Evaluation Metrics of Recommendation Tasks,” *J. Mach. Learn. Res.*, vol. 10, pp. 2935–2962, Dec. 2009.

[208] G. Shani and A. Gunawardana, “Evaluating recommendation systems,” Springer, TechReport MSR-TR-2009-159, Nov. 2009.

[209] Y. Sasaki, “The truth of the F-measure,” School of Computer Science, University of Manchester, Oct. 2007.

[210] M. Ge, C. Delgado-Battenfeld, and D. Jannach, “Beyond Accuracy: Evaluating Recommender Systems by Coverage and Serendipity,” in *Proceedings of the Fourth ACM Conference on Recommender Systems*, Barcelona, Spain, 2010, pp. 257–260.

[211] Sourceforge, “Open CSV Library,” Nov-2015. [Online]. Available: http://opencsv.sourceforge.net/. [Accessed: 26-Jan-2016].

[212] B. Berenbach, D. Paulish, J. Kazmeier, and A. Rudorfer, *Software & Systems Requirements Engineering: In Practice*, 1st ed. New York, NY, USA: McGraw-Hill, Inc., 2009.

[213] J. Holt, *A Pragmatic Guide to Business Process Modelling*, New edition. BCS, 2009.

[214] T. Jess, P. Woodall, and D. McFarlane, “Overcoming limited dataset availability when working with industrial organisations,” in *2015 IEEE 13th International Conference on Industrial Informatics (INDIN)*, 2015, pp. 826–831.

[215] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms, 3rd Edition*, 3rd edition. Cambridge, Mass: The MIT Press, 2009.

[216] J. de Ruiter, G. Weston, and S. M. Lyon, “Dunbar’s Number: Group Size and Brain Physiology in Humans Reexamined,” *Am. Anthropol.*, vol. 113, no. 4, pp. 557–568, Dec. 2011.

[217] B. Goncalves, N. Perra, and A. Vespignani, “Validation of Dunbar’s number in Twitter conversations,” *PLoS ONE*, vol. 6, no. 8, p. 8, Aug. 2011.

[218] J. Miller, “Dunbar’s Number, Span of Control and Lean Organization Design,” *Gemba Panta Rei*, 21-Jun-2011. .

[210] P. Upadhyaya, N. Anderon, M. Balazinska, B. Howe, R. Kaushik, R. Romomurthy, and D. Suciu, “The power of data use management in action,” in Proceedings of the 2013 international conference on Management of data, New York, NY, USA, 2013, pp. 1117–1120.

[220] R. Sarathy and K. Muralidhar, “Secure and useful data sharing,” *Decis. Support Syst.*, vol. 42, no. 1, pp. 204–220, Oct. 2006.

[221] P. Woodall and A. Wainman, “Data Quality in Analytics: Key Problems Arising from the Repurposing of Manufacturing Data,” presented at the ICIQ 2015, Cambridge, MA, USA, 2015.

[222] “Linked Open Vocabularies (LOV),” *Linked Open Data Vocabularies*, Mar-2016. [Online]. Available: http://lov.okfn.org/dataset/lov/about. [Accessed: 04-Mar-2016].

[223] J. Heer and S. Kandel, “Interactive analysis of big data,” *XRDS Crossroads ACM Mag. Stud.*, vol. 19, no. 1, p. 50, Sep. 2012.

[212] J. Brutlag, “Speed matters for google web search,” Google research blog, Jun-2009. Available: https://services.google.com/fh/files/blogs/google_delayexp.pdf. [Acceessed: 13-May-2017]

Attachments

Attachment A: Experiment description sheets

This thesis is looking into three experiment settings, which the following sheets describe in further detail.

- Procurement
- Production
- Support

A.1 Procurement

Name: *Procurement*	**ID:** *1*	**Date:** *02/02/2016*

Description:
A large manufacturing company has various procurement challenges such as PBL (Performance based logistics) contracts with the military or procurement for its own production. For PBLs and procurement for its own production, part availability is critical. The PBL payment or rate of production depends on the level of part availability. Various datasets are used to help with the procurement process. The question is how additional datasets could help improve the decisions of the different users involved (see below).

Datasets

Dataset name	**Data provider**	**Data costs** (annually)	**Data description**
Commodities for suppliers	*ERP system*	*(See procurement)*	*List containing the different commodities a supplier is qualified to deliver.*
DandB	*D and B*	*300Thd per year (Mock-up estimate)*	*External data from Duns and Bradstreet about supplier's bankruptcy likelihood. It can help with better supplier selection.*
Employee	*ERP system*	*50k (estimate)*	*List of employees*[30]
Financial health details	*Finance division using the typical financial*	*1Mn per year*	*Database containing details about the financials of a supplier.*

[30] This data is cheaper to acquire, because it needs to be collected for HR anyway. The costs just describe the additional costs of generating the dataset for Procurement purposes and maintaining them annually.

	tools and platforms		
Historical delivery performance	*ERP system*	*(See procurement)*	*Details about previous deliveries and their arrival times.*
HIS	*IHS*	*100Tsd (estimate but dependent on specific potentially changing contract with IHS. The price might vary dependent on the reports bought from IHS.)*	*External datasets with links to reports containing details about number and types of aircrafts, all the existing aircrafts, and the different airplane types, economic risk in certain countries, developments in certain countries, etc. This dataset is keyword based, certain keywords like aircraft types or locations link to specific reports and pages in these reports[31].*
Inspection types	*ERP system*	*(See quality inspections)*	*Links to quality inspections but contains further details about the specific tests performed on a part.*
Order	*ERP system*	*(See procurement)*	*List of currently open orders and their status.*
Part inventory	*ERP system*	*(See procurement)*	*List of parts in inventory.*
Procurement	*ERP system*	*2Mn per year (Mock-up estimate)*	*Details about the part lead times.*
Subtier questionnaire	*Risk management / Supplier*	*1Mn per year (Mock-up estimate)*	*List of questionnaires filled out by each supplier containing*

[31] In an actual data system, it would function like a keyword search for specific unstructured data in reports. However, to work within the experiments these keywords were predetermined.

			details about its sub tier suppliers[32]*.*
Supplier	*ERP system*	*(See procurement)*	*List of suppliers.*
Supplier management visits	*Supplier Management*	*5Mn per year (Mock-up estimate)*	*List of questionnaires filled out when a supplier is visited to qualify for a commodity. It ensures the supplier's production is suitable and fulfils all standards.*
Supplier to parts	*ERP system*	*(See procurement)*	*List matching parts to suppliers.*
Supports and services parts	*ERP system*	*1Mn per year (Mock-up estimate)*	*List with previous MTBD or MTBF times.*

User details:		
User name (and shortcut used in the rest of this experimental description)	**Use of the data**	**Datasets used**[33]
Procurement (P)	*P takes list of potential suppliers from Supplier management and list of qualified suppliers from Engineering. It then takes the incoming part orders and requests bids in accordance with the approved sourcing strategy (from supplier management). The sourcing strategy defines things like single source,*	*Dataset read:* *Commodities for suppliers (Data is used by P to know what commodities a supplier can deliver);* *Supplier (Data is used by P to have a list of possible suppliers for an order);* *Supplier to parts (used to identify parts*

[32] They are often just ~20% filled.

[33] Dataset read= datasets that are just read by this user, Dataset input= datasets where input is allocated to the dataset from this user, Additional= dataset that could be potentially relevant for this user but is currently not used from these users or the user is not aware of its existence. This data should also be considered in the data evaluation process. The additional data is ordered by preference.

	sole source, multiple source strategies etc.	*a supplier has on offer);* *<u>Dataset input:</u> Order (P makes the order); Supplier (P identifies suppliers that have been used for a specific part);* *<u>Additional:</u> 1. Historical delivery performance (Data could be used by P to see the past performance of a supplier and extrapolate its future performance); 2. Employee (Data could be used by P to identify if the order comes from someone with the permission for it);* *3. DandB and 4. Financial Health Assessment (Data could be used by P to identify if supplier will still able to deliver this part in the future);* *5. IHS (Data could be used by P to identify potential risk from a supplier's location or issues around an airplane type);* *6. Sub tier questionnaire (could be used to gather additional information about a supplier like its capacity etc. before making an order);*

Supplier management (SUMA)	*SUMA helps procurement to determine if the supplier is capable of producing specific parts and performs the supplier management visits. They need to select the pool of suppliers for the procurement agents. Their main output is the sourcing strategy.*	*Dataset read: Sub tier questionnaire and Supplier management visits (Data is used by SUMA to make a decision about which supplier is suitable);* *Dataset input: Commodities for supplier, Supplier to parts and Supplier (SUMA decides if a supplier can deliver certain commodities);* *Additional: 1. Historical delivery performance (Data could be used by SUMA to see the past performance of a supplier and extrapolate its future performance);* *2. DandB and 3. Financial Health Assessment (Data could be used by SUMA to identify if supplier will still able to deliver this part in the future);* *4. IHS (Data could be used by SUMA to identify potential risk from a supplier's location);*
Supplier quality (SQ)	*SQ are responsible for monitoring supplier quality. SQ helps with supplier management visits. They check against quality standards, part quality and correction plans. SQ potentially influences supplier management if*	*Dataset read:* None *Dataset input: Supplier Management Visits (SQ does the quality assessment on these visits);* *Additional: 1. DandB and 2. Financial Health Assessment*

	the quality of a supplier or product is not as expected.	*(Data could be used by SQ to identify, evaluate and discuss risks with this supplier's financial situation during the meeting and in their final evaluation report);* *3. IHS (Data could be used by SQ to identify, evaluate and discuss risk with this supplier's location during the supplier management meetings or once the supplier is regularly used);*
Asset managers/Materials management organization (AM)	*AM makes sure that the part pipeline is able to fulfil demands. Part pipeline consists of on-order, on-hand (inventory), and in-repair. They make one of the key decisions, by initiating the orders within procurement. When stock is below threshold they will initiate an order. They also identify issues in the supply chain.*	*Dataset read: Order, Part Inventory, Procurement and Support and Services parts (Data is used by AM to see incoming and outgoing parts in the future and make decisions about future orders that should be made);* *Dataset input: Order (AM initiates the order);* *Additional: 1. Historical Delivery performance, 2. DandB, 3. Financial Health Assessment, 4. IHS and 5. Subtier questionnaire (Data could be used by AM to infer issues or the non-existence of issues with some of the orders and react*

		earlier by adjusting orders);
Risk analyst (RA)	*RA are used to evaluate supplier risk, especially within the sub tier suppliers. They have usually a more strategic view on the supply chain, conduct root cause analysis for issues in the supply chain, loop towards supplier management to influence their supplier sourcing strategy, and gather various information about the supply chain. Most of their work is manual, they look at two types of issues (issues the company knows and issues it doesn't know yet) and then try to develop mitigation strategies together with Supplier management. RA is a trigger for changes in procurement.*	*Dataset read:* *Commodities for Supplier, DandB, Financial Health Assessment, Historical Delivery Performance, IHS, Order, Procurement, Supplier, Supplier Management Visits, and Supplier to Parts (Data is used by RA to be combined in an analytics tool to identify future issues with orders or suppliers);* *Dataset input:* *Order and Procurement (RA influence these indirectly-by warning procurement and supplier management about detected issues);* *Additional:* *None*
Manufacturing line (ML)	*ML is the line where the airplane or its parts are produced*	*Dataset read:* *None* *Dataset input:* *Historical delivery performance (ML know and record arrival of parts);* *Part inventory (ML checks stock of the parts);* *Procurement and Quality inspections (ML logs the arrival time and date of the part);* *Support and Services (ML logs the arrival*

		and leaving of these parts); *Additional:* None
Planners (PL)	*PL plans the production process and the parts required for this production*	*Dataset read:* None *Dataset input: Order and Supplier to parts (PL defines what parts need to be ordered and when in order to enable a proper production);* *Additional:* None
Supplier management Engineering (SME)	*SME decides on part specific acceptance criteria, which influence the supplier selection. They decide which suppliers are qualified, and whether alternate parts can satisfy the requirement of the engineering specification.*	*Dataset read:* None *Dataset input: Quality inspections and Supplier to parts (SME defines required parts and the criteria for inspection);* *Additional: 1. Sub tier questionnaire, 2. Commodities for supplier, and 3. Supplier (Data could be used by SME to help in better developing new parts easier to produce for suppliers);*
Product engineering (PE)	*PE does the drawings and designs specifications for the parts. (Note: They might over engineer a part and reduce the number of parts.)*	*Dataset read:* None *Dataset input: Support and Services Parts (SME develops the parts specifications);* *Additional: 1. Supplier to Parts, 2. Sub tier questionnaire, 3. Commodities for Suppliers, and 4. Supplier (Data could be used by PE in the product definition of parts requirements for easier procurement);*

<table>
<tr><td><i>Part quality (PQ)</i></td><td colspan="2"><i>PQ conducts the quality inspections of incoming parts.</i></td><td colspan="2"><i>Dataset read: None
Dataset input: Inspection types and Quality control (PQ does the inspections);
Additional: 1. Historical Part Delivery Performance, 2. Quality Control and 3. Inspection Types (Data could be used by PQ to get a better idea about issues with new incoming parts);</i></td></tr>
<tr><td colspan="5"></td></tr>
<tr><td colspan="5"><b>Scenario verification and detailing questions</b></td></tr>
<tr><td colspan="4">Are users aware of all the datasets?</td><td><i>No</i></td></tr>
<tr><td colspan="4">Are the users collaborating?</td><td><i>Yes</i></td></tr>
<tr><td colspan="5"><i>They are collaborating in the decision process. Especially supplier management, asset managers, procurement and supplier quality are working close in the supplier selection and ordering process. However, they still make independent decision potentially influencing the other divisions, without knowing further details about this impact.</i></td></tr>
<tr><td colspan="4">Are the users time- and / or resource-constrained in the data selection process?</td><td><i>Yes</i></td></tr>
<tr><td colspan="5"><i>Yes, they have to work on their operational tasks and do not have time to work through various datasets.</i></td></tr>
<tr><td colspan="3"><b>Does the dataset fulfil the following characteristics?</b></td><td><b>Yes or no</b></td><td><b>Comments</b></td></tr>
<tr><td colspan="3">User is using data from multiple (possibly changing) data sources</td><td><i>Yes</i></td><td><i>(See details in scenario description above)</i></td></tr>
<tr><td colspan="3">User has a set of offers (from data providers) to acquire more or different data to improve the user’s decisions</td><td><i>Yes</i></td><td><i>IHS and DandB data are just a few examples of further public, open or internal datasets that are considered.</i></td></tr>
</table>

User knows the value of a certain piece of data or a combination of data pieces in terms of contribution to a decision	*Yes / No*	*Once the user knows the impact on part availability the user can approximate it well.*
Data has costs associated with its allocation	*Yes*	*However, current values are estimates and not exact project measures.*
Partial information with data users and / or data providers	*yes*	*Yes, although they work together they are still not aware about all details the others are working on.*
Heterogeneous environment for data users and / or data providers	*Yes*	*(See details on various datasets and users)*
Distributed decision making between data user and data provider	*Yes*	*Some of the data inputs are made, not knowing about its detailed impact on other users decision making*

Details on the datasets

Dataset	**Columns and column details**
Commodities for suppliers	*CAGE: (CAGE identifies each military supplier and is a unique identifier of a company. It is used as a primary key among various tables.)* *Commodity: (Commodity that a company is allowed to produce for the company. It can be divided in various*

	groups such as Structures, Machine Parts, Interiors, Consumables, etc. One CAGE code can have different commodities.)
DandB	*CAGE: (CAGE code of company as a unique identifier)* *DUNS: (DUNS number of company)* *Name: (Name of company)* *Address: (Street number, Street name, City, State and Postcode of company’s headquarter)* *Telephone: (Phone number of company)* *Chief executive: (Name of company CEO)* *Year started: (Year in which the company was started)* *Employs: (Number of people employed by the company)* *Financial statement date: (Date on which company’s financial statement is released)* *Sales forecast: (Forecast of sales in the next year for the company)* *Net worth forecast: (Forecast for company’s net worth in the next year)* *Total liability: (Total debt of company)* *Financing: (Status of financing situation of company for the next year. It can either be “SECURED” or “UNSECURED”)* *DandB rating: (Score from 1 [low risk] to 9 [high risk] giving an indication for the risk of a company)* *Incidence of financial stress: (Percentage score between 0 and 100%. Based on the DandB Rating, which percentage of companies in this category did discontinue its operation in the next 12 months)* *Financial stress national average: (Gives a score of Financial Stress for the region the company is working in)* *Financial stress industry average: (Gives the average score of Financial Stress for the industry the company is working in)* *Credit delinquency score: (Score describing the likelihood of this company to not pay its debt in relation to all other companies. The score is from 0 to 1000. 1000 means 100% of companies are more likely to fail than this company. A 230 means 23% of companies are more likely than this company to fail)*

Employee	*Employee ID: (Individual ID for each employee as a unique identifier and primary key for this table)* *Last name: (Employee's last name)* *First name: (Employee's first name)* *Phone number: (Working phone number of employee)* *Office: (Office location, building and room number of employee's office. Not all employees have an office but most of them do.)* *Email: (Employee's email address)* *Department: (Department where the employee is working in, e.g. Part Quality, Procurement)*
Financial health details	*CAGE: (CAGE code of the company as the primary key)* *Company name: (Name of the company)* *Revenue: (Company's revenue in the last year)* *Profit: (Company's profit in the last year)* *Equity: (Total equity owned by company)* *Debt: (Total debt of company)* *Cash: (Total amount of cash the company owns)* *Profit margin: (Current profit margin of company)* *Book value per share: (Current book value of company divided by the number of shares of the company)* *Beta: (Beta factor of company's share)* *Profit to earnings: (Ratio of price to earnings of company)* *Dividend: (Dividend paid per share from company)*
Historical delivery performance	*Part number: (Number of part that was delivered in the past. Every part e.g. a specific screw has a separate number. However, two identical parts, e.g. the previously mentioned screw will have the same part number, but a different serial number.)* *CAGE: (CAGE code of supplier who is delivering this part [see Commodities for Suppliers for further details on the CAGE code])* *Serial number: (Identifies each separate produced part. It is the primary key for this table.)* *Expected delivery day: (Expected day when the company expected the part to arrive in its factory or warehouse.)* *Actual delivery day: (Actual day when the part arrived at the company. It is mostly before the expected delivery day. If the part is delayed it might arrive after the expected delivery day and can cause trouble in the production or the supply for potential customers).*

IHS	*Keyword: (Specific keyword as a connection with other tables*[34]*.* *Report: (Link to the report providing additional details)* *Page: (Page in document containing the keyword)* *Report number: (IHS number for the report)* *Report name: (IHS name for the report, e.g. In service Aircraft yearbook 2014/15)* *Publish date: (Date when the report was published)*
Inspection types	*Serial number: ([see Historical delivery performance table])* *Inspection method: (Method used for inspecting the part with the serial number above, e.g. Ultrasonic testing, Magnetic particle testing.)* *Metrics: (Metrics used in the inspection method, e.g. thickness, number of defects)* *Pass or fail: (yes or no - tells if the part has passed or failed the inspection)* *Employee ID: (Employee ID [see Employee table for further details])* *Date: (Date of the inspection)*
Order	*Order number: (Number of order as the primary key for this table.)* *CAGE: (CAGE code of supplier delivering this part.)* *Part number: (Number of part that was ordered [see Historical delivery performance])* *Quantity: (Number of parts that were ordered)* *Production step: (Step in the production process where the disruption occurred)* *Procurement manager: (Employee ID of Procurement Manager)* *Production manager: (Employee ID of Production Manager for this part, who started the order)* *Asset manager: (Employee ID of Asset Manager for this part who started the order)* *Status: (Status of the order like arrival or shipped)* *Expected arrival date: (Date, when the order is expected to arrive.)* *Actual arrival date: (If the order is already completed this contains the actual date the order arrived.)*

[34] It could be country, aircraft type, etc. This keyword is used to represent a search through a set of unstructured reports. With search the user would normally go through all these reports and find a match based on the data the user is currently seeing. By having these keywords this thesis has a similar solution, which enables easier experiments without having to implement a search function through various datasets.

Part inventory	Part number: (Number of part that is in inventory, it is used as the primary key of this table.) On hand quantity: (Quantity of this part that are on hand in the warehouse.) Backorder: (Quantity of this part that is already ordered.) In repair: (Quantity of this part that is being repaired.)
Procurement	Part number: (Number of part that can be repaired or ordered as a primary key of this table.) CAGE: (CAGE code of supplier for this part [see Historical Deliver Performance for details on the CAGE code.]) Administrative lead-time: (Administrative lead time [ALT] is the time it takes for the company to release an order of this part including additional work from engineering for example.) Production lead-time: (Production lead time [PLT] is the time it takes for the supplier to produce and deliver the product to the company after the order was send. The ALT and PLT combined are the total lead time it takes from the time of making an ordering decision to the time the product arrived at the company.) Part reparable or not: (Describes if this is part that can be repaired.) Repair turnaround time: (Average time it takes to repair this part.) Costs: (Historical costs for ordering this part.)
Sub tier questionnaire	CAGE: (CAGE code of company who completed the questionnaire [see Historical Deliver Performance for details on the CAGE code.] This is the primary key for this table.) Supplier name: (Name of supplier who completed this questionnaire.) Sub tier supplier: (Name of sub tier suppliers of this company) Delivery performance: (Supplier's announced delivery performance as mentioned in the questionnaire completed by this supplier.) On time PO releases: (Supplier's announced on time purchase order releases as mentioned in the questionnaire completed by this supplier.) Rate of rejection: (Supplier's announced rate of rejection [parts that were send to one of its customers

	but rejected] as mentioned in the questionnaire completed by this supplier.) *Sub tier staffing level/overtime/rate of attrition: (Supplier’s announced sub tier staffing level, amount of overtime, and the supplier’s rate of attrition as mentioned in the questionnaire completed by this supplier.)*
Supplier	*CAGE: (CAGE code of supplier [see Historical Deliver Performance for details on the CAGE code.]. This is the primary key for this table.)* *DUNS: (Number as a unique identifier for each company. Duns and Bradstreet manage it.)* *JCP certificate number: (Joint Certification Program number is a number given to each military supplier within Canada or the US. This number is given to a company from the Joint Certification Office between Canada and the US.)* *Company name: (Name of supplier)* *Status: (Divides the supplier into active and obsolete records. This describes if this company is still existent.)* *Parent CAGE: (CAGE code of a parent company (if existent) of this supplier.)* *Address: (Address of supplier’s main address)* *PO Box: (PO Box of supplier’s main address)* *City: (City of supplier’s main address)* *ZIP: (ZIP code of supplier’s main address)* *CAO-ADP: (Contract Administration Office – Automatic Data Processing is a number given to a supplier by the Office of Contract and Administration which is reviewing and signing sponsor projects such as some military projects. This is a number to automatically process certain suppliers.)* *State: (State of supplier’s main address)* *County: (County of supplier’s main address)* *Voice phone number: (Voice phone number that the company should call at this supplier.)* *Fax phone number: (Fax phone number that the company should use at this supplier.)* *Date CAGE code established: (Date when the CAGE code of this supplier was established.)* *Last updated: (Date of the last time this record was updated.)* *Point of contact: (Person to contact at this supplier.)*

Suppliermanagement visits	*CAGE: (CAGE code of supplier [see Historical Deliver Performance for details on the CAGE code.]. This is the primary key for this table.)* *Supplier name: (Name of supplier)* *Location: (Supplier location that was visited.)* *Date: (Date of the supplier visit.)* *Employee ID: (ID of employee leading the supplier visit.)* *Supplier contact: (Name of main contact at supplier.)* *Supplier derived delivery performance: (Delivery performance derived based on the visit at this supplier on a scale from 1 to 100)* *QMS grade: (Quality management system grade based on the visit at this supplier from 1-10)* *Part inspection method of supplier: (Part inspection method grade based on the visit at this supplier on a scale from 1-10)* *Lean grade of supplier: (Lean grade based on the visit at this supplier from 1-10)* *Disruption tolerance: (Disruption tolerance grade based on the visit at this supplier from 1-10)* *Capacity: (Production capacity grade based on the visit at this supplier from 1-10)* *Capability: (Production capability grade based on the visit at this supplier from 1-10)* *Tooling and capabilities of equipment: (Tool and equipment capabilities grade based on the visit at this supplier from 1-100)* *Manufacturing capability: (Manufacturing capability grade based on the visit at this supplier from 1-100)* *Master scheduling: (Master scheduling grade based on the visit at this supplier from 1-100)* *Procurement: (Procurement grade based on the visit at this supplier from 1-100)* *Business strategy: (Business Strategy grade based on the visit at this supplier from 1-100)* *Engineering capability: (Engineering capability grade based on the visit at this supplier from 1-100)* *Business systems: (Business systems grade based on the visit at this supplier from 1-100)* *Union or not: (Yes or no depend on the existence of a union at this supplier)* *Critical skills: (Note of skills that are critical at this supplier)*

Supplier to parts	*CAGE: (CAGE code of supplier [see Historical Deliver Performance for details on the CAGE code])* *Part number: (Number of a part. This is the primary key for this table)*
Supports and services parts	*Part number: (Number of part as a primary key for this table)* *CAGE: (CAGE code of company producing this part as a foreign key)* *Meantime between failures (actual): (Mean time between failures – actual [MATBFA]. This describes the actual value of the time it takes between a specific part type to fail. This value is measured during operations of the part.)* *Meantime between failures (predicted): (Mean time between failures – predicted [MTFBP]. This describes the predicted value of the time it takes between a specific part type to fail. This value is estimated from engineering and suppliers at the start of production of a new part.)* *Meantime between demand: (Mean time between demand [MTBD]. This value describes the average time until a new specific part from this part type will be needed)*

Table 35: Procurement experiment description

A.2 Production

<table>
<tr><td colspan="3"><b>Name:</b>
Production</td><td><b>ID:</b>
2</td><td><b>Date:</b>
02/02/2016</td></tr>
<tr><td colspan="5"><b>Description:</b>
Within the company’s production various datasets are used from a variety of different employees. In each of these steps different datasets are required for the coordination of the production. Various issues need to be addressed on a continuous basis. Among these issues are the following:
- Managing missing parts within the production process
- Managing machine failures
- Managing replacements of parts between different airplanes
- Managing specific requirements for each individual airplane
Addressing each issue requires a variety of datasets. Addressing all of them requires an even greater variety. Identifying the relevant datasets for the users managing the production process is an important task.</td></tr>
<tr><td colspan="5"></td></tr>
<tr><td colspan="5"><b>Datasets</b></td></tr>
<tr><td><b>Dataset name</b></td><td><b>Data provider</b></td><td><b>Data costs:</b></td><td colspan="2"><b>Data description</b></td></tr>
</table>

Disruption history	This data is separately collected for analysis purposes.	2Mn for collecting and sorting this information continuously.	*List of historical disruptions to the manufacturing process (types of disruption: Safety, manufacturing, part quality, part quantity, personal issues, environmental (e.g. snowstorm in Seattle), problems with existing products that cause changes in existing production process (e.g. FAA changes)).*[35]
Inspection types	*ERP system*	*(See quality control)*	*Links to quality inspections but contains further details about the specific tests performed on a part.*
Machine failures	(See Disruption History)	(See Disruption History)	*Records of all currently and previously broken machines.*
Machine status	(See Production Schedule)	(Same as Production Schedule)	*List that contains all machines and the plan for their regular maintenance.*
Machine use	(See Production Schedule)	(Same as Production Schedule)	*Table planning the use of different machines over time and the airplane production projects they are used for.*
Parts list	(See Production Schedule)	(Same as Production Schedule)	*List of parts per airplane that is in production.*
Part status	(See Production Schedule)	(Same as Production Schedule)	*Details about the current production stages of each part and details when parts will arrive.*
Personal plan	(See Production Schedule)	(Same as Production Schedule)	*Schedule of when and how employees are going to work.*
Production schedule	Internal manufacturing planning system.	50k per datasheet	*List containing a detailed plan for airplane production. There are various people*

[35] This data is mainly used for improving the decision-making and is not needed for continuous manufacturing operations. Therefore, there are higher costs in collecting it.

			keeping the data up to date[36]*.*
Production status	(See Production Schedule)	(Same as Production Schedule)	*Details about the current production stages of each airplane.*
Prognostics – Disruptions	External data analytics tool	1Mn one off (based on papers and industry estimates for similar types of projects in Big Data and ETL for providing the data) and 500k (reoccurring costs for data analysts and maintenance)	*Output from a tool that analyses the other datasets and predicts disruptions*[37]*.*
Prognostics - Machine failures	External data analytics tool	(Same as Prognostics Disruptions)	*Output from a tool that analyses the other datasets and predicts machine failures.*
Prognostics – Part arrival and quality	External data analytics tool	(Same as Prognostics Disruptions)	*Output from a tool that analyses the other datasets and predicts part arrival and part quality.*
Prognostics – Personal issues	External data analytics tool	(Same as Prognostics Disruptions)	*Output from a tool that analyses the other datasets and predicts personal issues.*
Prognostics – Safety related issues	External data analytics tool	(Same as Prognostics Disruptions)	*Output from a tool that analyses the other datasets and predicts safety related issues.*
Quality control	Collected from Quality control	5k for providing this information to	*List containing all quality controls. They are used to ensure parts and productions fulfil the quality*

[36]This data is already needed for running the internal production. The costs for providing this data are only the costs for maintenance in making it accessible to additional users annually.

[37] The main costs of this project are to ensure that the data gets to the data analytics tool and the costs for hiring data analysts.

		additional users[38]	*requirements and are based on inspections of various parts.*

User details:

User name	**Use of the data:**	**Datasets used**
Engineering development (ED)	ED designs the airplane. ED usually has the biggest influence at the start of a new airplane production.	*Dataset read: None* *Dataset input: Production schedule (ED helps initially shaping this plan as part of the airplane development process);* *Production status (ED influences the status by shaping the schedule for the production and reacting to problems);* *Parts list (ED defines what parts are needed on an airplane);* *Quality control and inspection types (ED sets the requirements for the quality control check);* *Additional: 1. Prognostics – Disruptions, Prognostics Part arrival/quality, Prognostics – Machine failures, Prognostics – Personal issues and Prognostics – Safety related issues (ED could make better decisions by knowing about various issues earlier);*
Foreman (F)	F is in charge of executing the day-to-day production at the separate machines. F monitors and executes details of production	*Dataset read: Production schedule (F needs this data to run the production);* *Dataset input: Production status (F update about the current status in the*

[38] Data is already collected regardless of additional data allocation because it is essential for quality control

	status, machine failures, and master schedule.	*production from F perspective);* *Personal plan (F decides who is doing which task);* *Disruption history (F inputs production disruption history from production line);* *Additional: 1. Prognostics – Personal issues (F could make better decisions by knowing about personal issues earlier);*
Line manager (LM)	LM is in charge of executing the day-to-day production at each production line. LM monitors production status, machine failures, and master schedule.	*Dataset read: Production schedule (LM needs this data to run the production);* *Dataset input: Production status (LM update about the current status in the production from LM perspective);* *Personal plan (LM uses it to decide who is doing which task);* *Disruption history (LM inputs production disruption history from production line);* *Additional: 1. Prognostics – Personal issues (LM could make better decisions by knowing about personal issues earlier);*
Machine manager (MM)	MM ensures the continuous repair of machines.	*Dataset read: Production schedule (MM needs this data to run the production);* *Production status (MM needs this data about the current state of production);* *Dataset input: Production status (MM might shutdown production if there are problems with a machine and therefore influences the schedule. In addition, MM update about the current*

		machine status in the production process); *Machine use (MM defines how different machines are used);* *Machine failure (MM records when and why a certain machine has failed);* *Machine status (MM gives regular update about the status of different machines);* *Personal plan (MM influences it to ensure everyone has the right training to use certain machines);* *Disruption history (MM inputs machine disruption history from machines);* *Additional: 1) Prognostics – Machine failures (MM could make better decisions by knowing about machine failures earlier) and Prognostics – Personal issues (MM could make better decisions by knowing about personal issues earlier);*
Machine planner (MaPl)	MaPl plans the usage of machines and fills the machine use plan	*Dataset read: Production schedule (MaPl need this data to identify the machines needed for the different steps);* *Production status (MaPl needs this data to know about the current state of production);* *Machine use (MaPl needs this data about past and current use of machines to monitor current production*

		and improve the user's future decision making); *Machine failure and Machine status (MaPl monitors current production and wants to improve future plans by knowing about machine failures and status changes);* *Disruption history (MaPl looks mainly for machine related disruptions to adjust the user's plan);* *Dataset input: Production schedule (MaPl helps with input of the machines planned for the different production steps);* *Additional: 1. Prognostics – Machine failures (MaPl could make better decisions by knowing about machine failures earlier);*
Master planner (MP)	MP plans the production process for an airplane. Ensures the right machines are free and the parts are ordered together with machine and part planner. MP is responsible for the master schedule.	*Dataset read: Machine use (MP needs this data about past and current use of machines to monitor current production and improve the user's future decision making);* *Parts list (MP use data to know what specific parts are needed for a certain airplane);* *Production status (MP stays updated about production status to monitor the process and improve future schedules);* *Machine failure and Machine status (MP monitors current production and wants to improve future plans by*

		knowing about machine failures and status changes); *Part status (MP monitors current orders and wants to improve future plans by knowing about part status);* *Personal plan (MP uses this data to monitor current production and improve future plans);* *Dataset input: Production schedule (Dataset is main input from MP);* *Additional: 1. Disruption history (MP Data could help to improve future plans by better considering likely disruptions);* *2. Prognostics – Disruptions, Prognostics – Part Arrival/Quality, Prognostics – Machine failure, and Prognostics – Personal Issues (MP could make better decisions by knowing about various issues earlier);*
Part planner (PPl)	PPl is responsible for making sure all parts for a new airplane are considered. Initializes the ordering process of parts on the parts list.	*Dataset read: Production schedule (PPl read initial rough plan to identify detailed parts that are needed);* *Parts list (PPl uses this data to know which specific parts are needed to produce a specific airplane);* *Production status (PPl stays updated about production status to monitor the process and improve future plans);* *Dataset input: Production schedule (PPl helps with input of the parts planned for*

		the different production steps); *Part status (PPl sets when specific parts need to be ordered);* *Additional: 1. Prognostics – Part Arrival/Quality (PPl could make better decisions by knowing about part arrival and quality earlier);*
Procurement (P)	P order the parts from the parts list.	*Dataset read: Production schedule (P needs data to make decisions about when and what parts to order);* *Production status (P stays updated about production status to monitor the process and ensure in time part arrival);* *Parts list (P uses this data to know what part to order);* *Quality control and inspection types (P uses past quality performance to select the right suppliers for the future);* *Disruption history (P monitors supplier induced disruptions);* *Dataset input: Part status (P updates about the current status of a part in the ordering process [e.g. "part order", "part in production", "shipping"]);* *Additional: None*
Production manager (PM)	PM is in charge of overall production management. PM monitors master schedule and production status.	*Dataset read: Production schedule (PM needs this data to run the production);* *Dataset input: Production status (PM update about the current status in the*

		production from PM perspective); *Disruption history (PM inputs production disruption history from production line);* *Additional: 1. Prognostics – Disruptions, Prognostics – Part Arrival/Quality, Prognostics – Machine failures, and Prognostics – Personal issues (PM could make better decisions by knowing about various issues earlier);*
Quality control (QC)	QC makes sure that supplier parts arrive on time.	*Dataset read: Production status (QC stay updated about the current production status to know which control need to be executed first);* *Part status (QC needs this data to identify issues with a part and to know when it arrives at QC);* *Dataset input: Quality control and inspection types (QC executes quality control check and records the results);* *Additional: 1. Prognostics – Part Arrival/Quality (QC could make better decisions by knowing about part arrival and quality earlier);*
Safety (S)	S ensures safety of manufacturing personal.	*Dataset read: Production schedule (S needs this data to identify safety relevant production steps and when they occur);* *Machine status (S needs to know about which machines are running to identify safety risks);* *Personal plan (S to know when safety relevant procedures are executed*

		and to know what people are doing them); *Dataset input: Production status (S might shutdown production if there is a risk in the production process);* *Disruption history (S inputs safety related disruption);* *Additional: 1. Prognostics – Safety related issues (S could make better decisions by knowing about safety related issues earlier);*
Supplier manager (SM)	SM manages suppliers. SM monitors performance and other criteria of different suppliers (e.g. delivery experience, training, and qualifications of personal).	*Dataset read: Parts list (SM reads this to know what parts are needed for certain airplanes);* *Part status (PM uses this data to analyse specific suppliers);* *Quality control and inspection types (SM monitors quality performance of its suppliers);* *Dataset input: Disruption history (SM inputs supplier induced disruptions);* *Additional: 1. Prognostics – Disruptions (SM could make better decisions by knowing about disruptions earlier);* *Prognostics – Part Arrival/Quality (SM could make better decisions by knowing about part arrival and quality earlier);*

Scenario verification and detailing questions	
Are users aware of all the datasets?	*No, especially on the prognostics side not all data is available to the users.*
Are the users collaborating?	*Yes*

Various departments have a set of interactions along the productions process and need to communicate regularly.		
Are the users time- and / or resource-constrained in the data selection process?		*Yes*
Limited resource in selecting data that can help to identify future disruptions. Currently the data in order to do these predictions is not available.		
Does the dataset fulfil the following characteristics?	**Yes or no**	**Comments**
User is using data from multiple (possibly changing) data sources	Yes	Not on the current production execution, which is already well planned and has most data sources available. However, they are missing data on predictions which is already existing.
User has a set of offers (from data providers) to acquire more or different data to improve the user's decisions	Yes	Additional data could help in the production planning by predicting failures of machines and parts
User knows the relevancy of a certain piece of data or a combination of data pieces in terms of contribution to a decision	Yes and no, dependent on the dataset	Only the output of an analytical tool is able to tell the impact of a dataset, but this can't be know before the actual analytics
Data has costs associated with its allocation	Yes	This domain experts are able to provide estimate for current and

		additional datasets
Partial information with data users and / or data providers	Yes	They don’t know exactly all available datasets and the knowledge of other users
Heterogeneous environment for data users and / or data providers	Yes	There are various different datasets and users
Distributed decision making between data user and data provider	Yes	Different users make decision that impact the production

Details on the datasets	
Dataset	**Columns and column details**
Disruption history	*Disruption number: (Number for each separate disruption as primary key for this table)* *Disruption type: (Definition of the type of disruption, it could be machine failure, part disruption, personal disruption or quality disruption)* *Disruption description: (Detailed description of the actual disruption)* *Production step: (Step in the production process where the disruption occurred)* *Machine failure: (Link to the machine failure as a foreign key it is NULL if there is no machine failure as the type of the disruption)* *Part disruption: (Link to the part list as a foreign key it is NULL if there is no part disruption as the type of the disruption)* *Personal disruption: (Link to the personal plan as a foreign key it is NULL if there is no personal disruption as the type of the disruption)* *Quality disruption: (Link to the inspection type as a foreign key it is NULL if there is no quality disruption as the type of the disruption)* *Date: (Date when the disruption occurred)* *Time: (Time when the disruption occurred)*

Inspection types	*Serial number: (Serial number of part that is inspected)* *Inspection method: (Method used for inspecting the part, e.g. Ultrasonic testing, Magnetic particle testing.)* *Metrics: (Metrics used in the inspection method, e.g. thickness, number of defects)* *Pass or fail: (yes or no tells if the part has passed or failed the inspection)* *Employee ID: (Employee ID which is linked to the HR table)* *Date: (Date of the inspection)* *[Note: This dataset is similar to the dataset for the procurement experiment. However, quality control also happens between different production steps and not just at the arrival of a part.]*
Machine failures	*Failure number: (Number of failure as a primary key for this table)* *Machine number: (Number of machine that failed)* *Failure type: (Type of failure that occurred)* *Production step: (Link to the actual production step in the production schedule as a foreign key)* *Date: (Date of machine failure)* *Time: (Time of machine failure)* *Estimated repair time: (Estimated time to repair the failure)* *Repair status: (Current status of repair. It could be identified, parts on order, in progress, final test runs, etc.)*
Machine use	*Machine usage number: (Individual usage number for the machine as the primary key for this table)* *Machine number: (Individual number for a machine as a foreign key for this table)* *Machine type: (Type of operation the machine is going to conduct)* *Production step: (Production step in which the machine is going to be used as a foreign key for this table)* *Machine planner: (Machine planner responsible for this planning of the machine)* *Date: (Date when the machine is going to be used in this step)*
Machine status	*Machine number: (Machine number to uniquely identify each machine as the primary key for this table)* *Status: (Current status of machine at the specific time and date, e.g. in production or maintained)* *Production step: (Production step in which the machine was or is used as a foreign key for this table.)* *Machine planner: (Machine planner responsible for this planning of the machine)* *Date: (Date when this machine was used for this production step)* *Time: (Time when this machine was used for this production step)*

Parts list	*Part number: (Number of part as a primary key for this table)* *Part description: (Description of part)* *Parts on hand: (Number of parts of this part number the company has in the warehouse or in production buffer places)* *Backorder: (Number of parts of this part number that are currently ordered with suppliers)*
Order	*(See Procurement scenario order table)*
Part status	*Order number: (Number of order as a primary key for this table)* *Production step: (Production step in which this part is going to be needed as foreign key for this table)* *Part number: (Number of the part as a foreign key)*
Personal plan	*Personal plan number: (Number of the personal plan as a primary key)* *Personal number: (Personal number for each person as a foreign key to an employee list)* *Production step: (Production step in which this person is going to be used)* *Foreman: (Foreman in charge of the employee during this production step)* *Date: (Date when this employee is working on this production step)* *Time: (Time when this employee is working on this production step)*
Production schedule	*Product number: (Number of product that is produced as the primary key for this table)* *Lead planner: (Person leading the planning process [Linked to employee table])* *Model number: (Number of model that is produced)* *Customer: (Customer name who is buying the product)* *Start date: (Date when the production of this product is planned to start)* *End date: (Date when the production of this product is planned to end)* *Line number: (There are multiple lines doing the assembly this number differentiates them)*

Production status	*Product number: (Number of the product that is being produced as a combined primary key [with production step] for this table)* *Production step: (Step in the production process as a combined production step)* *Status: (Status of this specific production step. It could be delayed, on time, in progress, or finished)* *Description: (Description of this production step)* *Start date: (Date when this specific production step for this product number is supposed to start)* *Start time: (Time when this specific production step for this product number is supposed to start)* *End date: (Date when this specific production steps for this product number is supposed to end)* *End time: (Time when this specific production step for this product number is supposed to end)*
Prognostics - Disruptions	*Disruption number: (Number of this predicted disruption as a primary key)* *Disruption type: (Type of disruption that is predicted)* *Likelihood: (Likelihood of this disruption happening)* *Consequences: (Potential influence this disruption would have on schedule)* *Mediation strategy: (Strategy to reduce the problems following from the disruption)*
Prognostics - Machine failures	*Machine number: (Number of this predicted machine failure as a primary key)* *Failure type: (Type of machine failure that is predicted)* *Likelihood: (Likelihood of this machine failure happening)* *Consequences: (Potential influence this disruption would have on schedule)* *Mediation strategy: (Strategy to reduce the problems following from the disruption)*
Prognostics – Part arrival and quality	*Supplier number: (Number of this predicted part arrival and quality as a primary key)* *Disruption type: (Type of part arrival and quality problem that is predicted)* *Likelihood: (Likelihood of this part arrival or quality problem happening)* *Consequences: (Influence on schedule)* *Mediation strategy: (Strategy to reduce the problems following from the disruption)*

Prognostics – Personal issues	*<u>Personal number:</u> (Number of this predicted personal issues as a primary key)* *<u>Issue type:</u> (Type of personal issue that is predicted)* *<u>Likelihood:</u> (Likelihood of this personal issue happening)* *<u>Consequences:</u> (Potential influence this disruption would have on schedule)* *<u>Mediation strategy:</u> (Strategy to reduce the problems following from the disruption)*
Prognostics – Safety related issues	*<u>Safety number:</u> (Number of this predicted safety related issues as a primary key)* *<u>Issue type:</u> (Type of safety related issues that is predicted)* *<u>Likelihood:</u> (Likelihood of this safety related issue happening)* *<u>Consequences:</u> (Potential influence this disruption would have on schedule)* *<u>Mediation strategy:</u> (Strategy to reduce the problems following from the disruption)*
Quality control	*<u>Part number:</u> (Number of part that is in the quality control process)* *<u>Arrival date:</u> (Date when the part is planned to arrive)* *<u>Production step:</u> (Each production step has separate quality issues)* *<u>Serial number:</u> (Linked to inspection types)* *[Note: This dataset is similar to the dataset for the procurement experiment. However, quality control also happens between different production steps and not just at the arrival of a part.]*

Table 36: Production experiment description

A.3 Support

Name: *Support*	**ID:** *3*	**Date:** *02/02/2016*

Description:
A large manufacturing company is working with various customers who require support for their airplanes. This support is mainly in repairs, management of spare parts in warehouses, and management of personal for this support.
In order to manage this support in the most effective way the users could leverage various internal and external datasets. Identifying the relevant datasets to use for this problem is a challenge for the large manufacturing company.

Datasets

Dataset name	**Data provider**	**Data costs** (Annually)	**Data description**
Airplanes	*Production and*	*50k a year for collecting initial*	*List of airplanes held by certain airlines, locations,*

	Customer providing a continuous update	*configuration, etc.*	*and airplanes configuration. This dataset is not essential for operations but can help in the decision making process[39].*
Airplane conditions	*Data collected from customers*	*100k for setup of customer input with additional 50k annual maintenance.*	*Table describing the conditions in which each airplane was used.*
Airplane part history	*Data could be collected internally and in a general registry with customers and suppliers*	*0.5 to setup the system, 15Mn for an internal tracking system going to different airplanes during build, in addition 0.5Mn annual running, maintenance and manual interference costs.*	*List of parts in the different airplanes and where they are coming from. Including details like previous repairs, supplier, or manufacturing site for example.*
Details warehouse parts	*Division running the PBL programs*	*50k annually Data is already collected for accounting and other internal purposes. Therefore it just need be maintained and made available*	*List of all specific parts stored in all the warehouses*
Engineering part life	*Provided by engineering department during the*	*50k annually The data has to be collected during the development*	*List containing the estimated part life from engineering.*

[39] Data is allocated from production whenever an airplane is delivered or potentially from external companies in case of rare changes

	development process	*process. Therefore, the main issue is making it accessible for a larger group of users.*	
FAA changes	*FAA online website*	*100k a year for continuous collection*	*List of required changes from the FAA for specific airplanes.*
Facilities	*Internal registry with all facilities*	*10k for regularly updating the list and making it available to all people*	*List of all facilities and their location.*
Future airplane plans	*Data collected from customers and analysed from industry or DoD reports*	*75k annual cost to subscribe to industry reports and making them available to various users.*	*Details about plans for the future of the different existing airplanes. The data includes details about future lifetime and expected usage of different airplanes.*
Future part costs	*System predicting the development of different types of parts*	*One off analytics platform: 950k. In addition, the company needs acquisition of roughly 25k annual cost to subscribe to industry reports and 500k for the data analysts*	*List with predictions about future costs of parts*
Incident reports	*Data provided by customer or from incident report websites*	*220k One off for a system that collects data online from websites and 50k*	*List of different incidents that occurred during the usage of the airport as it is recorded in the logbook for each airplane.*

		annually for maintenance of this system.	
Number of flights per airline	*Production and Customer providing a continuous update*	*50k a year for collecting initial configuration, etc.*	*List of flights per airline and the airports used that can indicate future support demand.*
Order	*Central order ERP system*	*50k annually. Orders have to be done for operational purposes, therefore the only additional costs are making it accessible and maintenance.*	*List of orders going out to procurement*
Part estimation	*Division running the PBL programs*	*2.5Mn dependent on the number of parts (5000$ per part is an estimate for 500 key components)*	*Details about the estimated number of parts that are needed. It is largely influenced by output of support requests.*
Parts	*Division running the PBL programs*	*100k a year*	*Contains a list of parts used within the company. It also contains the number of repairs and orders.*
Parts in warehouse	*(See Details warehouse parts)*	*(See Details warehouse parts)*	*List with quantities of parts stored in certain warehouses as an aggregated version of details warehouse parts.*
Repairs	*Division running the PBL programs*	*50k some data is already collected and just needs to be maintained and presented. However data could be*	*List with details about all the repairs that still need doing and that have already been completed.*

		improved for 9Mn (0.5Mn per repair Depot for 18 depots)	
Repair parts	(See Repairs)	(See Repairs)	List with all the parts needed for specific repairs.
Supplier	Central supplier ERP system (Asset Manager provides most of this data)	Normally just 50Thsd for data allocation and maintenance. [Note: Additional allocation for new suppliers: Per part: 1/100 parts have the issue costs for this are 50k (due to finding, contracts, etc.)]	List of suppliers, their location and their contact details. This dataset is not essential for operations but can help in the decision making process.
Support requests	Customer (e.g. Airline), ERP system	100k a year	List of support requests and their description from customers.
Warehouse	(See details Warehouse parts)	(See details warehouse parts)	List of all warehouses and their address.

User details:

User name	Use of the data	Datasets used
Asset manager (AM)	AM tries to keep the part inventory as small as possible while still fulfilling the demand. AM looks at incoming support requests and demand forecasts to make decisions about future orders of parts. His goal is an optimal number of parts in warehouse. The user makes one of the most	Dataset read: Parts in warehouse (Data is used by AM to know the current stock); Repairs and Repair parts (Data is used by AM to identify parts that are

	important decisions within this experiment.	*currently repaired and could be used later);* *Order (Data is used to identify how many orders have already been made);* *Dataset input: Part estimation (AM has the job to make these estimations);* *Order (AM triggers additional orders of a part);* *Additional: 1. Future airplane plans (Data could be used by AM to identify future usage of the airplane and its parts);* *2. Future part costs (Data could be used by AM to identify changes in price and react by adjusting order times);* *3. Airplane part history (Data could be used by AM to identify upcoming issues with parts);* *4. Supplier (Data could be used by AM to identify issues with suppliers);* *5. Number of flights per airline, 6. Incident reports and 7. Airplane conditions (Data could be used by AM to identify future usage of parts);*
Aircraft operations manager (AOM)	*AOM is a customer in the support role. The user is an external person (Airline or service provider). The user's contact within the company is the warehouse.*	*Dataset read: Parts and Part in warehouse (Data is used by AOM to know available spare parts in case they are needed);* *Dataset input: Support requests (AOM request the part from the manufacturer);* *Number of flights per airline (AOM provides details on*

		the flights flown for the airplane); Additional: None
Engineering team (ET)	ET gives part usage duration estimates for new programs. For established programs it has a smaller role.	Dataset read: None Dataset input: Engineering part life (ET provide further details on the estimated part life); Additional: 1. Repair parts, 2. Incident reports and 3. Repair (Data could be used by ET to identify typical part failures and consider this in future part designs); 4. Airplane part history, 5. Airplane conditions and 6. Number of flights per airline (Data could be used by ET to identify the typical usage of the parts);
First level mechanic (FLM)	FLM works directly at the airplane. The user does the maintenance and repairs at the airplane.	Dataset read: None Dataset input: Incident reports and Airplane conditions (FLM helps providing the incident reports and airplane conditions when they remove the part, notice that it needs repairing and then send it to the SLM); Additional: None
Flight certification manager (FCM)	FCM takes incoming FAA requests and ensures that they are executed	Dataset read: None Dataset input: Repairs and FAA changes (FCM is in touch with FAA and initializes the Repairs required for certain FAA changes); Additional: None
Second level mechanic (SLM)	SLM work in the company and receive the parts for further repair. They partially rely on data from FLM.	Dataset read: Parts in warehouse (Data is used by SLM to identify available replacement parts); Repairs and repair parts (Data is used by SLM to identify

		currently already repaired parts and potentially learn from these repairs or use the parts in repair); Incidents reports and airplane conditions (Data is used by SLM to get a better understanding of the circumstances in which the part operated); Airplane part history (Data is used to identify details about the usage); *Dataset input: 1. Repair parts and 2. Repairs (SLM repair the parts and therefore provide the input for this table);* *Additional: None*
Supplier (S)	*S receives the order from procurement. They mainly provide additional data that could help in this process.*	*Dataset read: Order (Data is used by S so that the user knows which part the user has to produce);* *Dataset input: Parts (S provides input on the duration to build a part); Engineering part life (S provides input on how long certain parts are likely going to last during operation);* *Additional: 1. Repair parts, 2. Incident reports and 3. Repair (Data could be used by S to identify typical part failures and consider this in future part designs);* *4. Airplane part history, 5. Airplane conditions and 6. Number of flights per airline (Data could be used by S to identify the typical usage of its parts);*
Warehouse manager (WM)	*Manages the operations within a warehouse. WM marks the reduction in*	*Dataset read: Details warehouse parts, Parts in warehouse, and*

	stock, which the asset manager is monitoring.	*Warehouse (Data is used by WM to know the current status of the warehouse that is managed); Repair and repair parts (Data is used by WM to get an idea of potentially arriving repaired parts); Order (Data is used by WM to know how many parts are going to arrive);* *Part estimation (Data is used by WM in order to know how many parts are predicted to arrive longer term);* *Dataset input: Warehouse, Details in warehouse and Parts in warehouse (WM observes the warehouse personal that collects and inputs these data);* *Additional: None*
Scenario verification and detailing questions		
Are users aware of all the datasets?		*Yes, they know all the datasets. But don't always have access and don't have all the details about the datasets.*
Are the users collaborating?		*Yes*
Along this process the users have to exchange various information and discuss various issues. They need to exchange details about fulfilling a specific supplier order or predict required level of storage levels. The asset manager requires detailed data form engineering and suppliers at the beginning of a new contract to predict part failure rate.		
Are the users time- and / or resource-constrained in the data selection process?		*Yes*
The users often have the problem that data does not exist, is incomplete, or finding it is a big effort. At the same time the users have only a limited time available to make their decisions.		
Does the dataset fulfil the following characteristics?	**Yes or no**	**Comments**

User is using data from multiple (possibly changing) data sources	*Yes*	*Internal users are using various data sources (such as suppliers and customers, or additional details about these) and their amount of detail and content are varying.*
User has a set of offers (from data providers) to acquire more or different data to improve the user's decisions	*Yes*	*Various datasets would require additional data or details (especially for support part obsolescence)*
User knows the relevancy of a certain piece of data or a combination of data pieces in terms of contribution to a decision	*Yes*	*With additional or the relevant data the user would be able to make better decisions and identify their contributions*
Data has costs associated with its allocation	*Yes*	*Exact values are difficult, but in most cases good approximations are possible*
Partial information with data users and / or data providers	*Yes*	*Most data is only partially available and the specific data provider is not always known*
Heterogeneous environment for data users and / or data providers	*Yes*	*There is a variety of datasets and data users with various different types of data and different user tasks*

Distributed decision making between data user and data provider	*Yes*	*Data comes from distributed sources. Users make their decision often (partially) independent from each other. Although they might discuss some of them*

Details on the datasets

Dataset	**Columns and column details**
Airplanes	*Airplane number: (Number for airplane. Each airplane has a unique number as the primary key for this table)* *Airplane types: (Specific type of airplane)* *Owner: (Name of current owner of the airplane)* *Airline: (Airline currently operating the airplane. It provides a link to the actual flights per airplane beside the airplane number.)* *Base: (Base airport from which this airplane is operating. Specifically, relevant for military contracts.)*
Airplane conditions	*Flight number: (Number of flight as primary key for this table)* *Airplane number: (Number of the airplane that was flying as a foreign key for this table)* *Weather: (Text describing the weather conditions during the flight)* *Wind: (Text describing the wind conditions during the flight)*
Airplane part history	*Serial number: (Number unique identify every specific part as the primary key for this table. Other data such as supplier can be found via this key.)* *Airplane number: (Number of airplane which uses this specific serial number of a part as a foreign key for this table. Other data*

	such as conditions in which the airplane was used can be found via this key.) *Part number: (Number identify the type of part as a foreign key for this table)* *Start date: (Date identifying when this serial number was put into the airplane)* *Expected end date: (Date when this serial number should be replaced. If NULL then there is no specific date when this part needs replacing)* *End date: (Date describing when the serial number was actually removed from the airplane)* *Source: (Text describing who provided this data, e.g. customer, internal approximation, or industry database)* *Support number: (Link to Support request table describing which support request is addressing this FAA change requirements)*
Details warehouse parts	*Warehouse number: (Unique number of each Warehouse. Provide a link to the Warehouse table.)* *Serial number: (Detailed serial number of the part. This field is the primary key for this table.)* *Status: (Describes the current operational status of a specific part in the warehouse. It can be either "Shipped", "Stored", or "In Progress")*
Engineering part life	*Part number: (Number of part as the primary key for this table)* *Estimated duration: (Time which it will take until this part will need replacing)* *Confidence: (Percentage giving the confidence in this estimated part life)* *Responsible engineer: (Name of engineer responsible for this analysis)*
FAA changes	*FAA number: (Number from the FAA identifying the specific change as the primary key for this table)* *FAA description: (Text description of FAA changes required)* *Due date: (Date when the FAA changes need to be implemented in all airplanes)* *Responsible manager: (Name of manager who is responsible for implementing the changes)* *Support number: (Link to Support request table describing which support request is addressing this FAA change requirements)*
Facilities	*Facility number: (Number of facility as primary key for this table. It identifies each specific facility.)* *Address (Specific address of facility)*

Future airplane plans	*Airplane number: (Number uniquely identifying the airplane as primary key for this table. It serves as link to other tables which might contain more data)* *Description of plan: (Text describing the future plan with this airplane)* *Running date: (Date until when it is expected that the airplane is going to operate)* *Source: (Text describing the source for this data, e.g. the customer, industry reports, or news)*
Future part costs	*Part number: (Number uniquely identify the type of part as the primary key for this table)* *Part description: (Text describing the part)* *Current cost: (Monetary value describing the current costs for a part)* *Expected future cost: (Monetary value describing what the expected future value for this part is likely going to be)* *Confidence: (Percentage giving the confidence that this expected future costs of a part is actually going to happen. They can vary by the analysis method)* *Timeline: (Time describing the expected timeline for this adjustment in price)* *Data analyst: (Name of data analysts developing the basis [e.g. the data analyst found the report or developed the price prediction tool] for this analysis)* *Basis for analysis: (Text describing the basic piece of evidence leading to this analysis about the future price. It could be an industry report or a specific tool analysis various datasets)*
Incident reports	*Incident number: (Number for each incident report as a primary key for this table)* *Airplane number: (Number uniquely identifying the airplane which was part of the incident as a foreign key)* *Incident description: (Text describing what happened during the incident)* *Incident category: (Text categorising the incident into different groups such as “near miss with other airplane”, “Instrument failure”, etc.)* *Date: (Date when the incident occurred)*

Number of flights per airline	Airplane number: (Number of each specific airplane as primary key for this table) Flight number: (Number for the flight that was used) Airline: (Airline operating the flight) Start date: (Date of flight start) End date: (Date of flight landing) Take off date and time: (Time of flight start) Landing date and time: (Time of flight landing) Start airfield: (Airfield from where the plane started) Land airfield: (Airfield where the plane landed)
Order	(See Procurement)
Part estimation	Part number: (Number of part as primary key for this table) Part description: (Description of this part) Number of ordered parts per year: (Number of this type of part that are currently in the ordering process) Number of repaired parts per year: (Number of this type of part that are currently being repaired) Asset manager: (Name of asset manager responsible for the estimation of future required number of parts of this part.) Estimated order: (Estimated order quantity per year)
Parts	Serial number: (Specific unique serial number of every part ever ordered or repaired for a support request as primary key for this table) Support number: (Specific support number from Support requests that this part is ordered or repaired for. It is linked to the Support request table) Parts number: (Part Number for this part) Repair number: (Number of repair to be done to this specific part. It is NULL if there is no repair done to the specific serial number of this part.) Order number: (Number of order for this part. It is NULL if there is no order done to the specific serial number of this part) Warehouse number: (Number of warehouse that can deliver this specific part. It is NULL if there is no delivery from the warehouse done to the specific serial number of this part) Expected arrival date: (Date when order, repaired part or part from warehouse is expected to arrive) Actual arrival date: (Date when order, repaired part or part from warehouse actually arrived) Number of parts: (Number of part with this specific serial number.)
Parts in warehouse	Warehouse number: (Number for each warehouse to uniquely identify it. It is linked to the Warehouse table) Part number: (Number of part stored in the warehouse) Quantity: (Quantity of part stored in the warehouse)

Repairs	*Repair number: (Number of a repair to uniquely identify it as a primary key for this table. It is linked to the Parts table.)* *Serial number: (Serial number of part that is being repaired.)* *Parts number: (Unique Part Number for this part)* *Failure code: (Code to identify the specific failure diagnosed for this specific part.)* *Failure description: (Description of failure with this part.)* *Status: (Current status of repair process “Arrived”, “In repair”, “Finished repair”, or “not repairable”)* *Responsible mechanic: (Name of Mechanics in charge of this repair.)* *Facility number: (Number of facility where this repair is completed. This table it linked to the facilities table.)*
Repair parts	*Serial number: (Specific serial number of the part that is being repaired.)* *Repair number: (Link to repairs identifying the specific repair that this part is needed for.)* *Parts number: (Number of the part that is needed for a repair.)* *Warehouse number: (Link to the Warehouse table, which is delivering this part. It is NULL if the company does not get it from a Warehouse)* *Order number: (Link to the order table, which describes the order for this part. It is NULL if there is no order)*
Supplier	*(See Supplier in Procurement)*
Support requests	*Support number: (Number identifying each specific request for support from a customer as primary key for this table.)* *Customer number: (Number of customer making the support request)* *Request description: (Description of the request for support given by the customer)* *Request code: (Code to identify the specific type of request)* *Requested date and time: (Date and time of request from customer)* *Estimated completion date and time: (Date of estimated completion of customer request)* *Need date: (Date when the supplier ideally needs this request to be fulfilled)*
Warehouse	*Warehouse name: (Name for this specific warehouse)* *Address: (Address of the warehouse)* *Warehouse number: (Number of a specific warehouse as primary key for this table and link to other tables.)*

Table 37: Support experiment description

Attachment B: Case Study B dataset description

This attachment describes the table structure used in the second case study.

Main tables	Sub tables	Description[40]	Columns
Contract Mgmt System Data	Actual effective date	Provides details on the effective date of a contract such as start date or dates for amendments.	4
	Additional notes	Provides additional details around a contract agreement such as details around extending the contracts under certain circumstances.	6
	Admin fee	Provides details around the administration fee for contract management.	10
	Annual revenue	Describes the revenue and the sources for revenue around each contract.	8
	Award type	Provides details around the sources for a certain contract.	4
	Brokerage fee	Provides details around the brokerage fees associated with a specific contract.	5
	Category details	Identifies the categories relevant for specific contracts.	7
	Channel fee	Describes the details of the fee structure such as frequency and due date of certain fees for a contract.	12
	Class of TRD	Describes the trade regulations around a specific contract.	4
	Contract ID	Provides additional identifiers for the contract.	7
	Contract administration	Details around the people responsible for managing this contract.	13
	Contract description	Description of contract and document number used for this contract.	10
	Contract number successor	Details around the following contracts and current contract negotiations.	6
	Contract value	Detailed overview about the value created with a specific product.	9

[40] The description includes the ideal information provided in these tables. Many of these tables are often only filled in rare cases with the described information.

	Cost avoidance	Details around cost avoidance achieved by a product.	4
	Cost center and budget	Provides cost centre and budget for a specific contract.	4
	Current rebate scale	Describes the current agreement on rebates.	4
	Current savings	Describes the current savings achieved with this contract.	4
	Distribution channel	Describes the typical distribution channel, e.g. via the vendor or via distribution centres.	5
	Effective data	Provides the date since when the contract exists within the company.	4
	Effective price date	Provides the date since when the current price is effective.	4
	Extension reason	Provides the date when the contract was extended the last time.	4
	FOB terms	Describes the freight terms for this contract.	4
	Freight terms markups	Describes the mark-ups for different freight terms.	12
	Freight terms	Describes additional details around the freight terms for this contract.	5
	Growth incentive	Describes incentives for the company to increase the contract volume and the impacts related to a contract volume increase.	5
	Internal contract number	Provides the internal contract number used by the vendor.	4
	LOC details	Describes additional details around the services provided for a certain contract.	5
	Margin	Provides the companies margins associated with a specific contract.	5
	Market value	Describes the market value of a specific contract.	4
	Markup	Provides the mark-up of a specific contract.	4
	Member design form	Describes details around the form requirements for certain contract orders.	4

	Members discount	Describes details about which users receive additional details for certain products.	4
	Min order	Describes minimum order requirements for a contract.	5
	Payment terms	Describes payment terms for a contract.	4
	Price escalator details	Describes details around future price developments for a certain contract.	5
	Price expiration date	Provides details until when the price of a certain product is valid.	4
	Price tier	Describes details around different price levels and their future applicability.	10
	Pricing dsh	Describes if pharmaceutical dsh pricing applies to this contract.	4
	Product details	Provides an additional product description.	5
	Project planned build date	Describes future plans with a specific contract.	5
	R contract value	Describes the value of a specific contract for the company.	4
	Rebate details	Provides details about the different rebate levels.	12
	Related contracts	Provides details about related contracts.	4
	Renewal details	Describes current renewal planning for a given product.	5
	Required additional member	Describes if additional members for a certain contract are needed.	4
	Required authentication	Describes if a specific authentication with a vendor are required.	4
	Savings	Describes different types of savings achieved with a contract.	6
	Supplier sales data and electronic signatures	Describes details around additional sales data requested and provided by the supplier for this contract.	6
	Support details	Describes details around the support for this contract.	4

	Surcharges defined	Describes additional surcharges that are potentially applicable for this contract.	5
	Tracing details	Describes details around the tracing required for a certain contract.	13
IMS Benchmarking Data	Contract dates	Describes the dates and product description levels around a contract for a specific item at a specific facility.	5
	Contract details	Provides an overview about the main contract details for a specific item at a specific facility.	7
	Facility details	Describes details around the facility used for a certain item.	5
	IMS description	Provides a detailed description of the item at a specific facility.	6
	National price comparison	Provides a price comparison data for an item with nationwide benchmarks.	44
	Packaging and product name	Provides a packaging details and the product name for an item at a specific facility.	4
	Price comparison	Provides national averages, lows and modes of prices as a benchmark for an item at a specific facility.	5
	Price dates	Provides details around the dates in which a certain price was effective for the company.	4
	Sourcing details	Provides details around the sourcing for a specific item at a specific facility.	6
	Unit and prices	Provides number of sold units and mode prices for a certain item at a specific facility.	5
Item Master Data	Alt description	Provides alternative numbers used for an item number.	8
	Buy quantity	Provides details around the quantity of items sold for a certain type of item number.	5
	Dates	Provides contract effectiveness dates for a certain item.	3
	Item description	Provides specific description for a certain item.	4
	Manufacturing description	Provides the descriptions from the manufacturer.	5

	Manufacturing details	Describes details about the items from the manufacturer perspective such as strategic impact for example.	5
	Purchasing and inventory	Provides number of current purchases and current inventory of this item.	10
	STK number	Provides the details of quantity of items sold at a specific UOM (Unit of Measure).	5
	Segment description	Provides further details about the specific product segment of this item.	7
	Sell details	Details about sales of this item.	5
	Sell prices	Details about the price for which this item was sold.	5
	Trademark	Describes trademarks associated with a particular item.	4
	User description	Provides the users description of the item.	8
	Volume data	Provides the volumes sold of this item.	2
PO Spend Data	Contract details	Describes additional details about a specific contract associated with a particular purchase.	6
	Supplier name	Provides details about the supplier for a specific purchase.	4
	Time and Facility details	Provides time and facility for a specific purchase.	5
	Item and price details	Provides item and price details of a specific purchase.	16
Price List Data	Contract dates	Provides details around the contract for items on a price list.	9
	External item number	Provides the external item number of an item at the supplier for a specific item on the price list.	8
	Item dates	Provides system specific identifiers for an item on the price list.	11
	Item description	Provides item description and quantity of item on the price list.	9
	Row insert	Provides details about when the row was inserted into the database.	8
	Tier description	Provides details about the level of this item in the supply chain.	8
	Vendor description	Provides item number of the vendor for item on the price list.	8

ECatalog	Overview	Consists of an overview of data to be presented to the user in the electronic Catalogue. The data consists of - Item description - Supplier name - Supplier part number - Manufacturer name - Manufacturer part number - Quantity per item in UOM - Category name - Contract number	8

Table 38: Description of different tables used for the Case Study B

Attachment C: Empirical evaluation

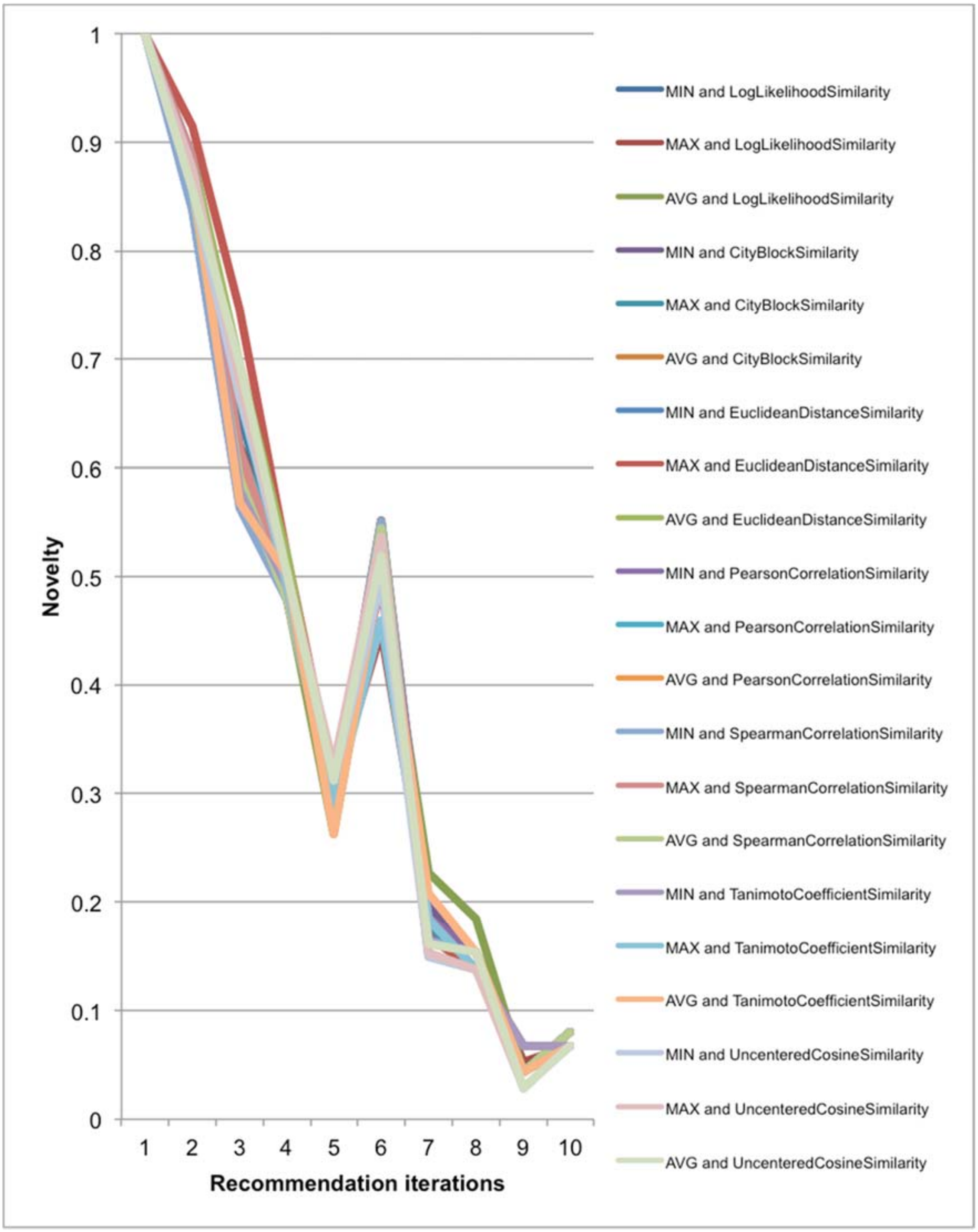


Figure 12: Experiment 1a: Average novelty over different recommendation iterations of different recommender system functions for all users

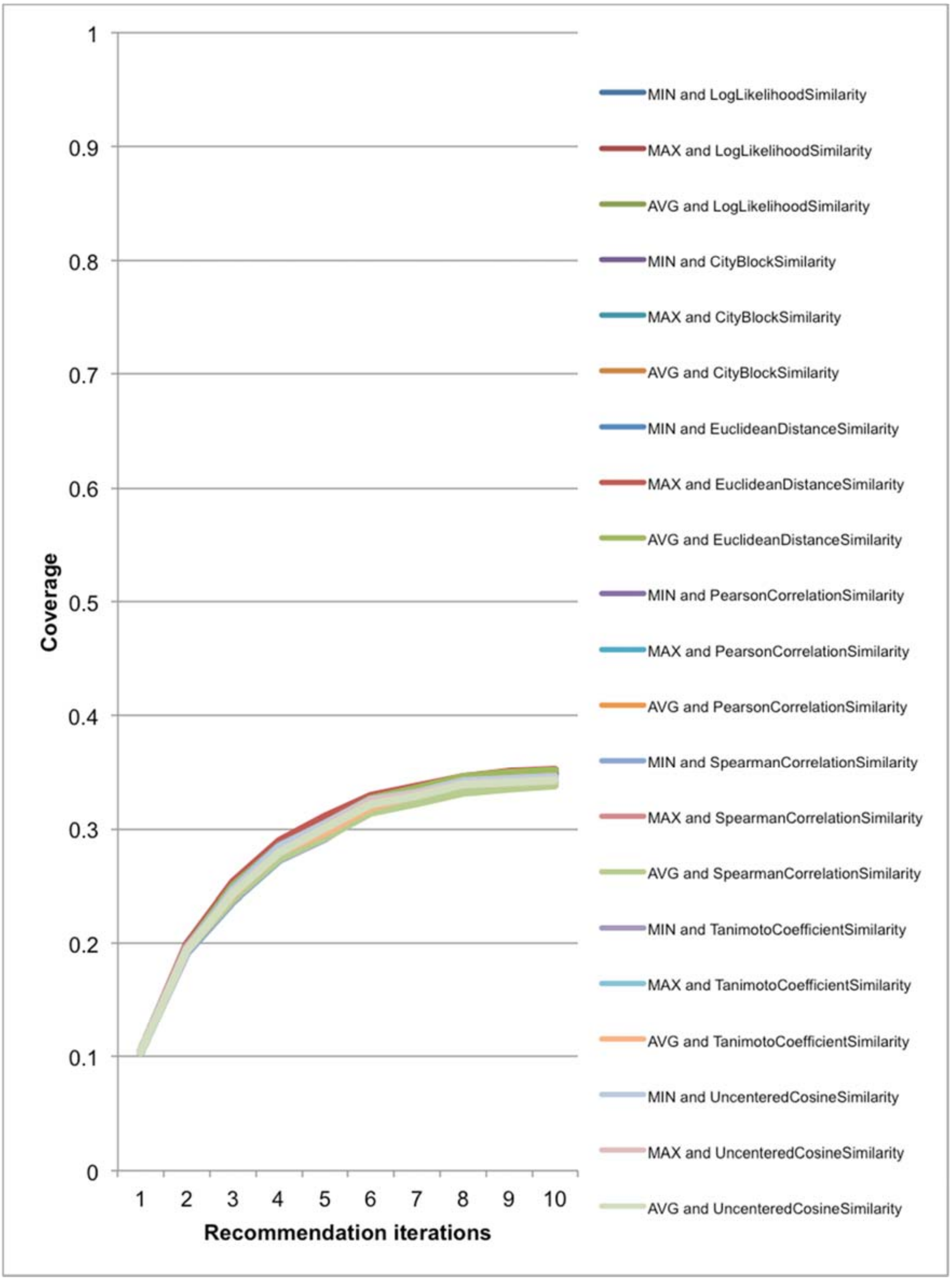


Figure 13: Experiment 1a: Average coverage over different recommendation iterations of different recommender system functions for all users

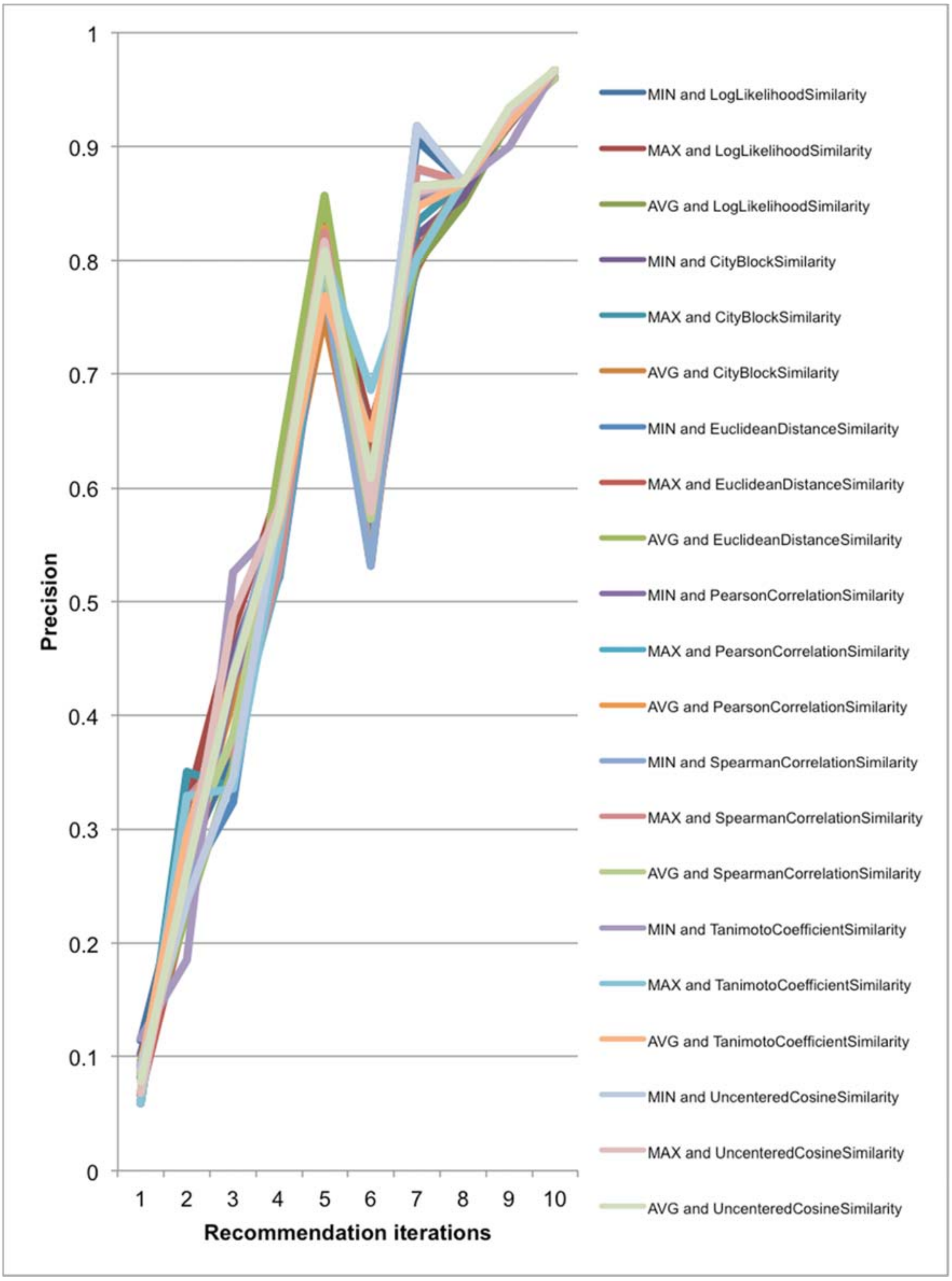


Figure 14: Experiment 1a: Average precision over different recommendation iterations of different recommender system functions for all users

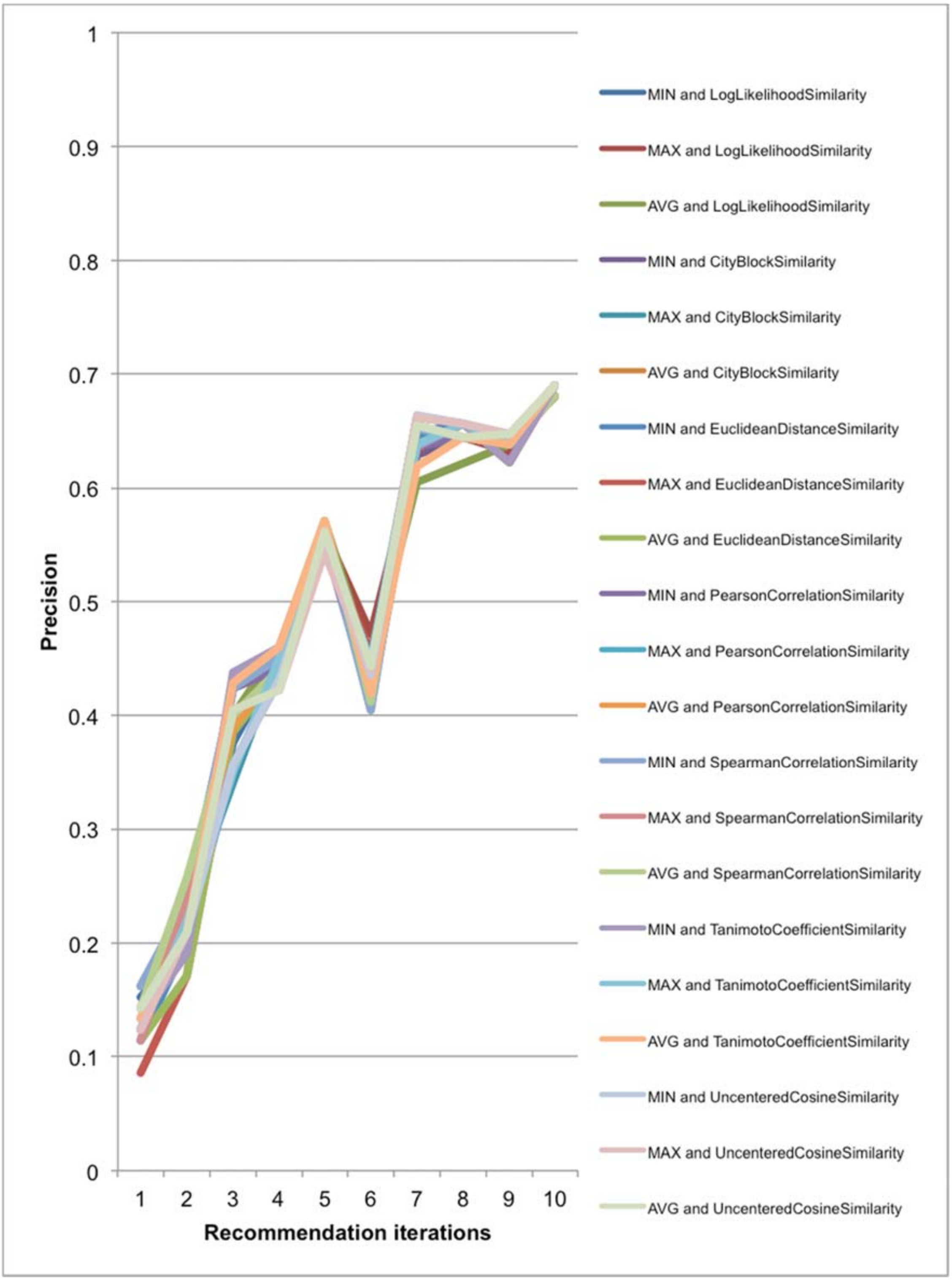


Figure 15: Experiment 1a: Average precision for tables over different recommendation iterations of different recommender system functions for all users

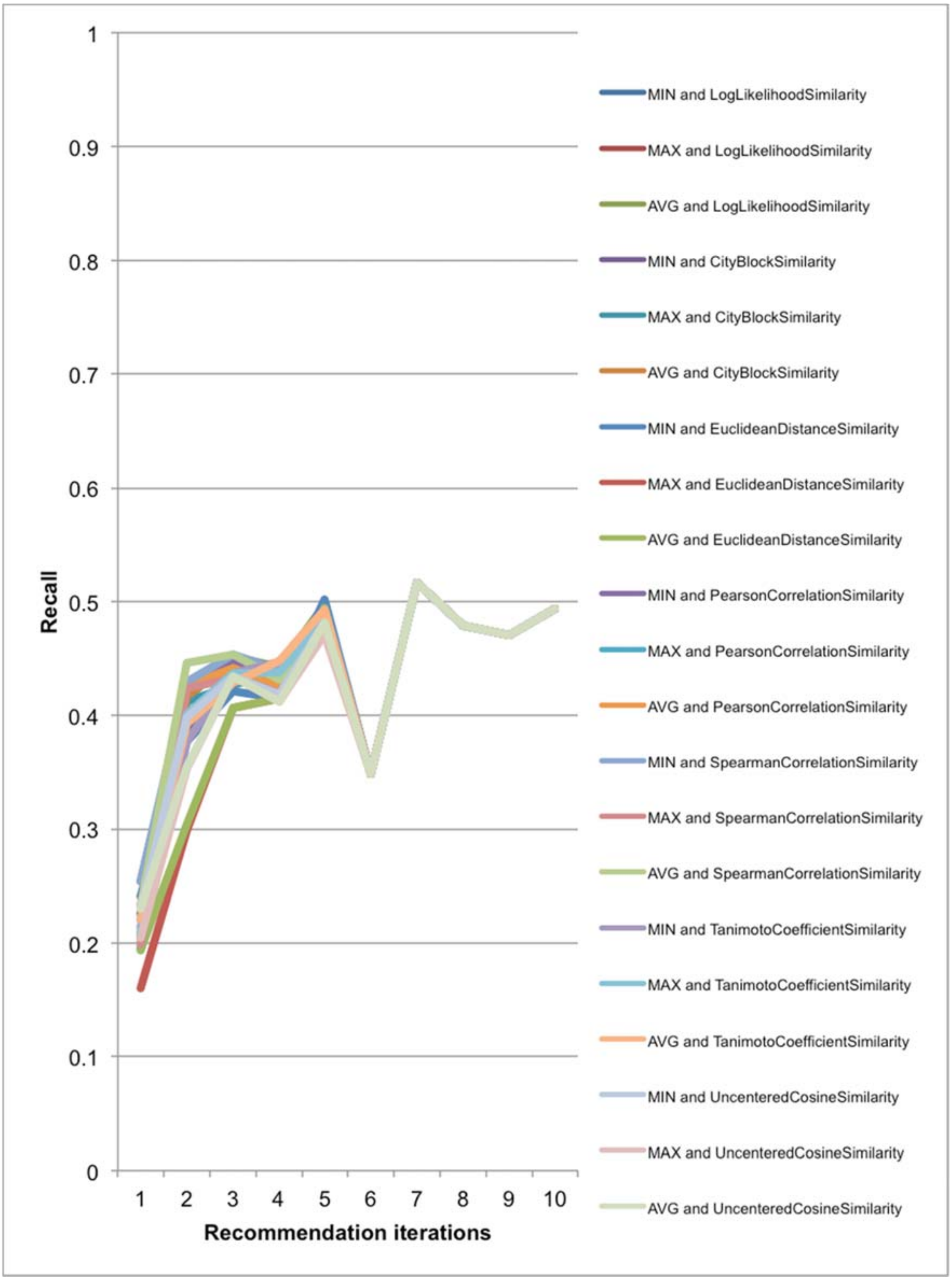


Figure 16: Experiment 1a: Average recall for tables over different recommendation iterations of different recommender system functions for all users

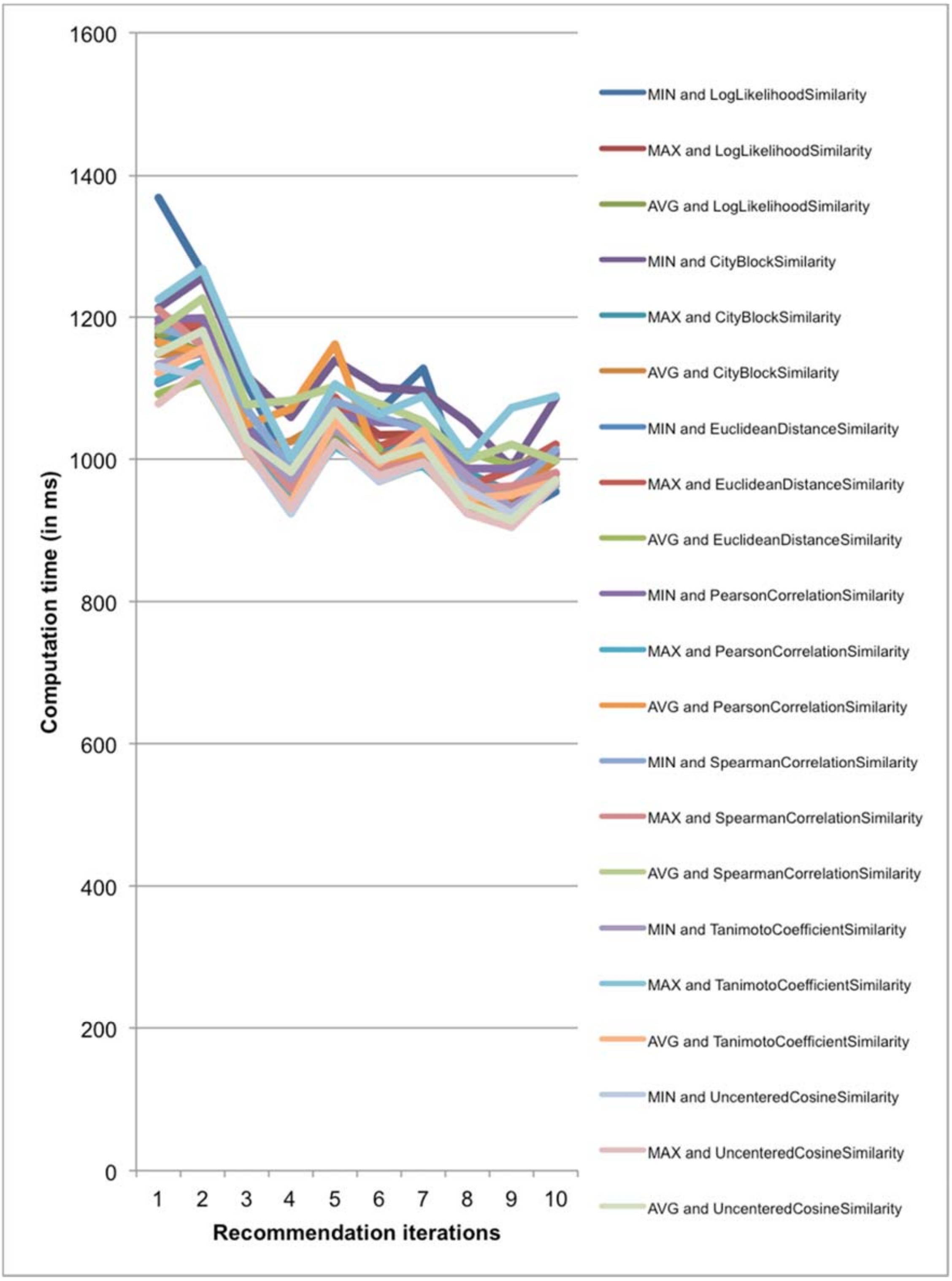


Figure 17: Experiment 1a: Average computation time over different recommendation iterations of different recommender system functions for all users

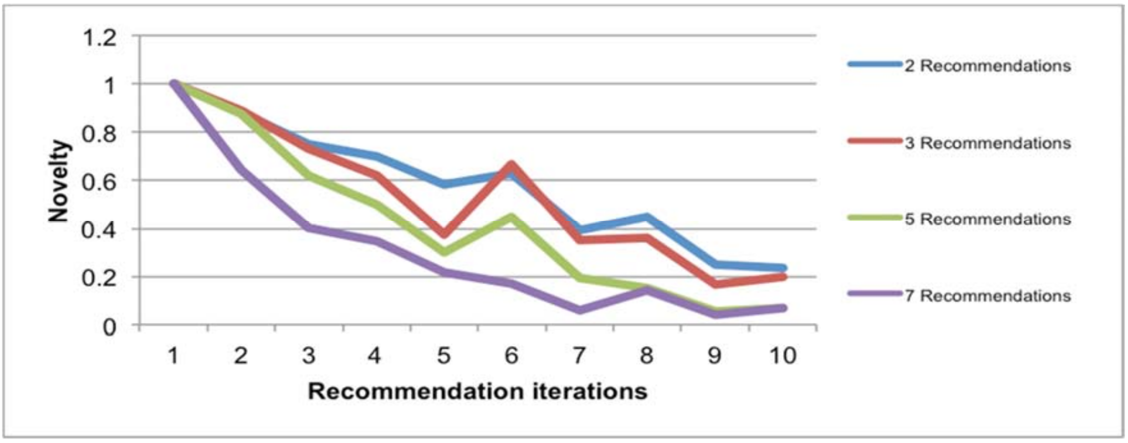


Figure 18: Experiment 1b: Average novelty over different recommendation iterations of different recommender system functions for all users

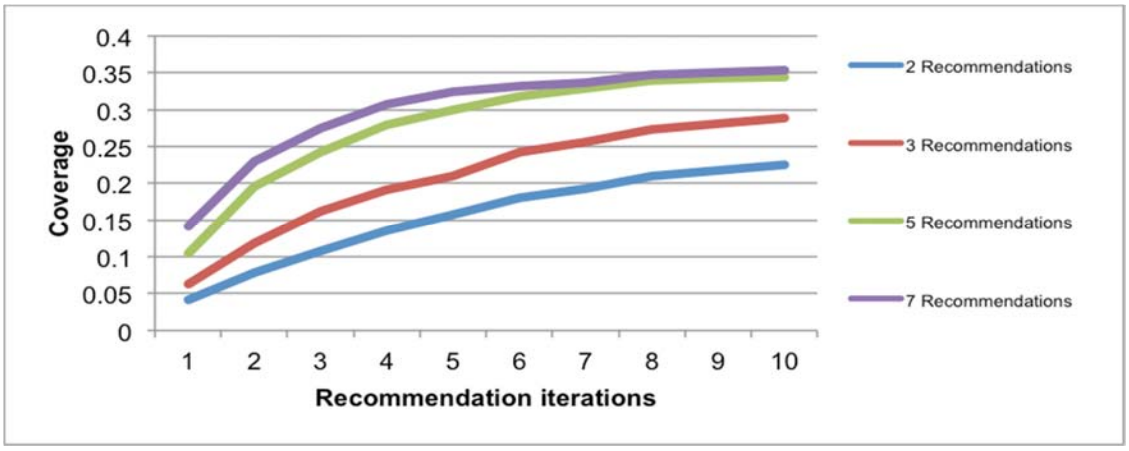


Figure 19: Experiment 1b: Average coverage over different recommendation iterations of different recommender system functions for all users

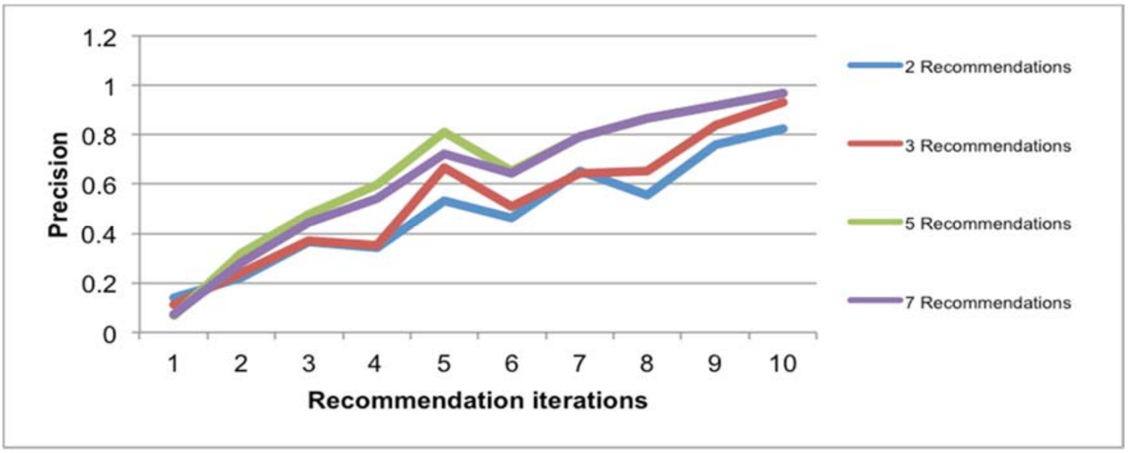


Figure 20: Experiment 1b: Average precision over different recommendation iterations of different recommender system functions for all users

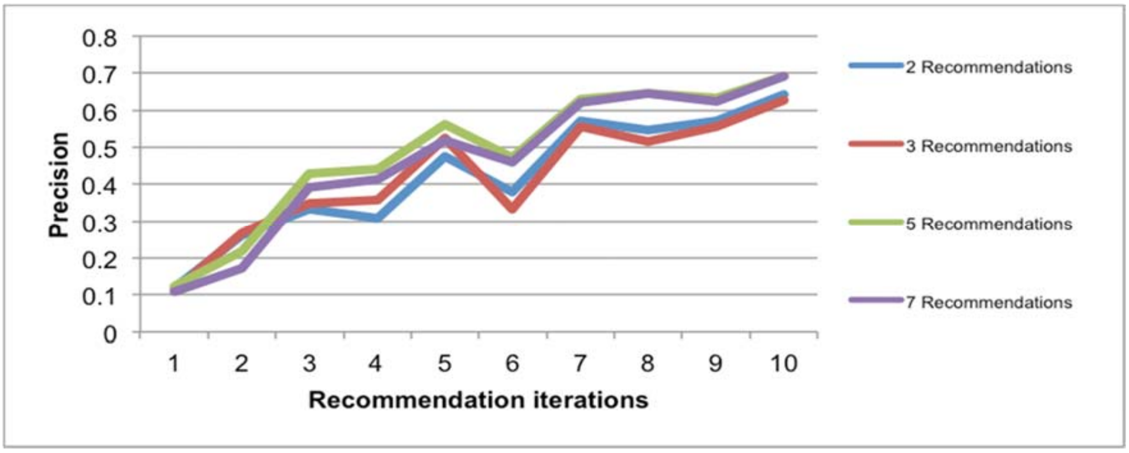


Figure 21: Experiment 1b: Average precision for tables over different recommendation iterations of different recommender system functions for all users

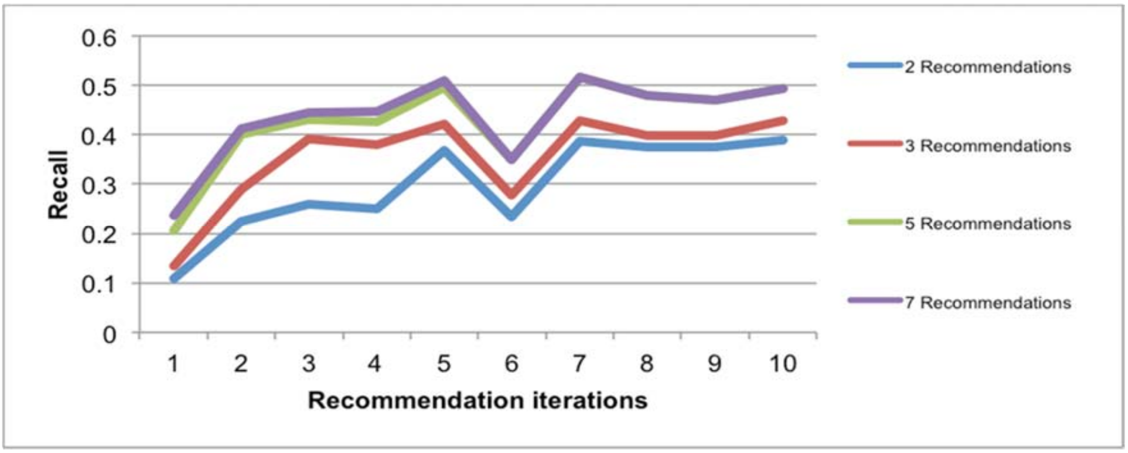


Figure 22: Experiment 1b: Average recall for tables over different recommendation iterations of different recommender system functions for all users

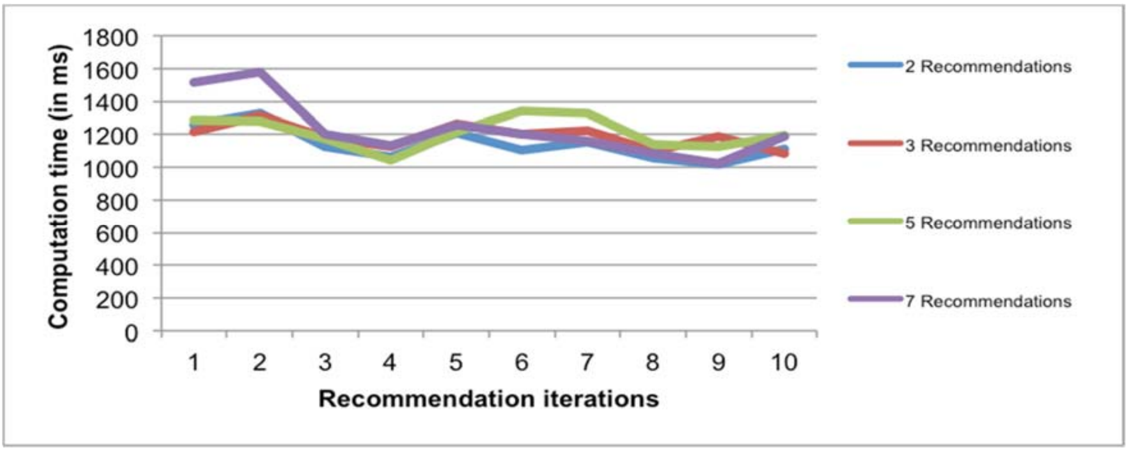


Figure 23: Experiment 1b: Average computation time over different recommendation iterations of different recommender system functions for all users

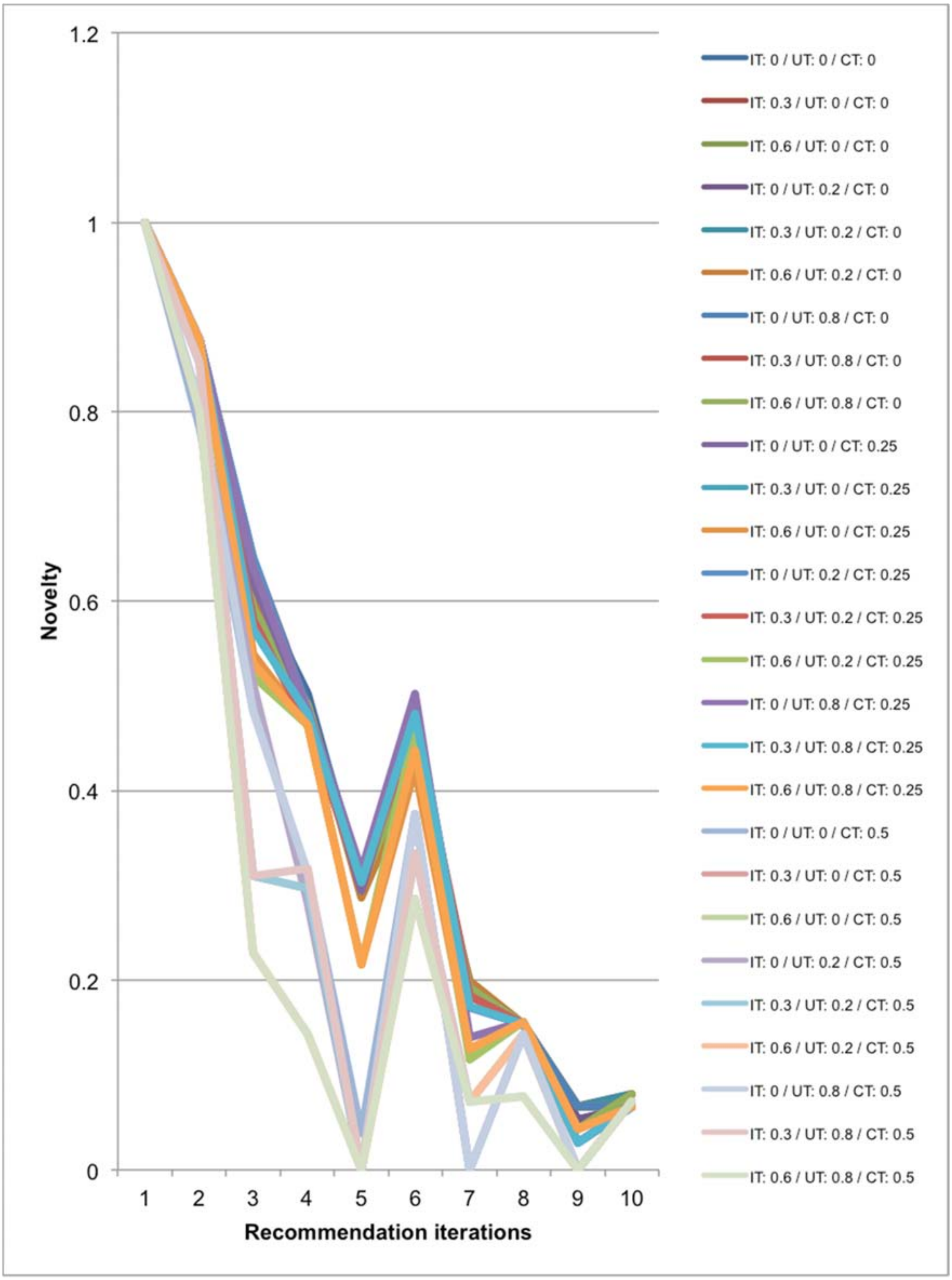


Figure 24: Experiment 1c: Average novelty over different recommendation iterations of different recommender system functions for all users

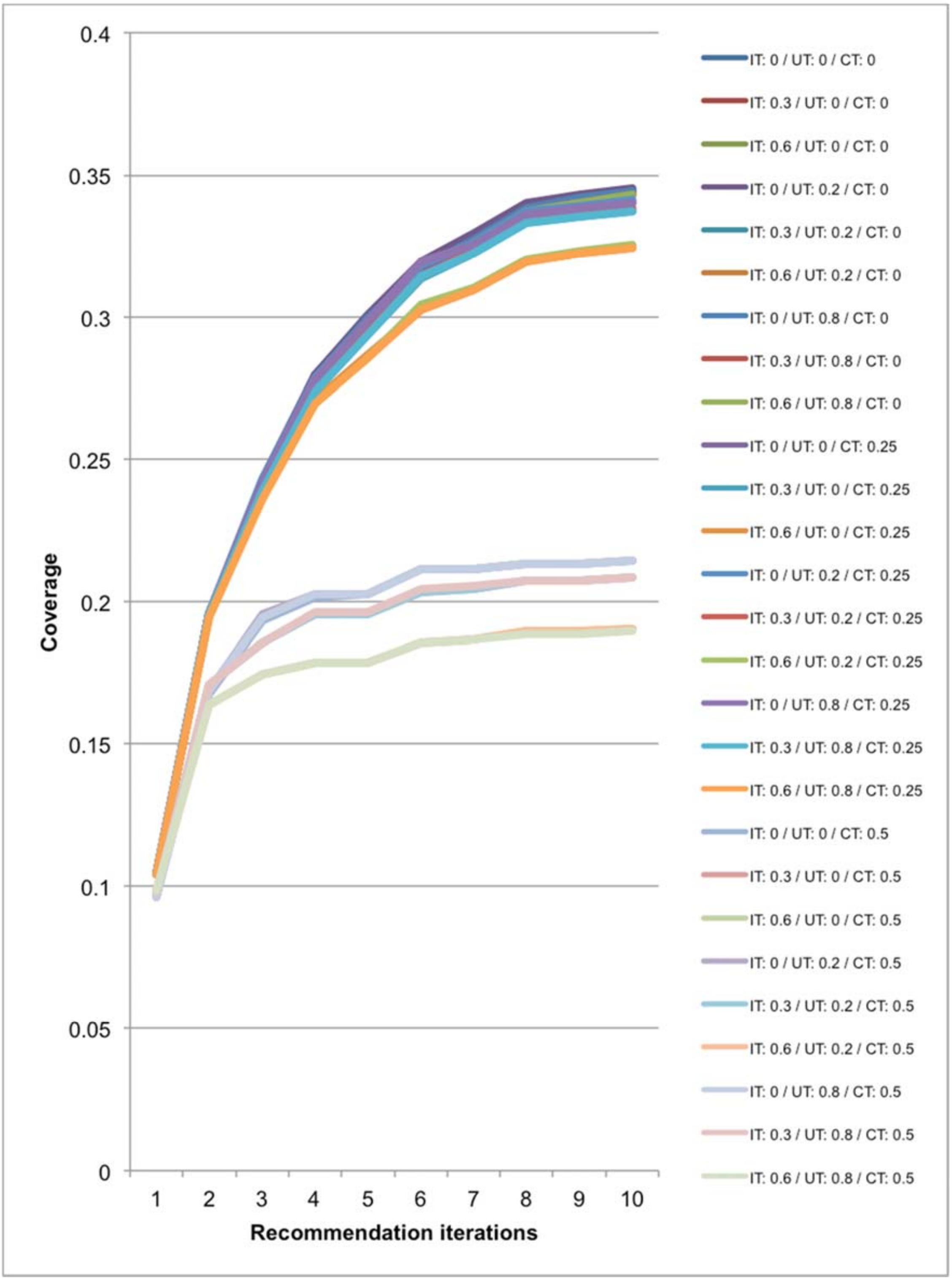


Figure 25: Experiment 1c: Average coverage over different recommendation iterations of different recommender system functions for all users

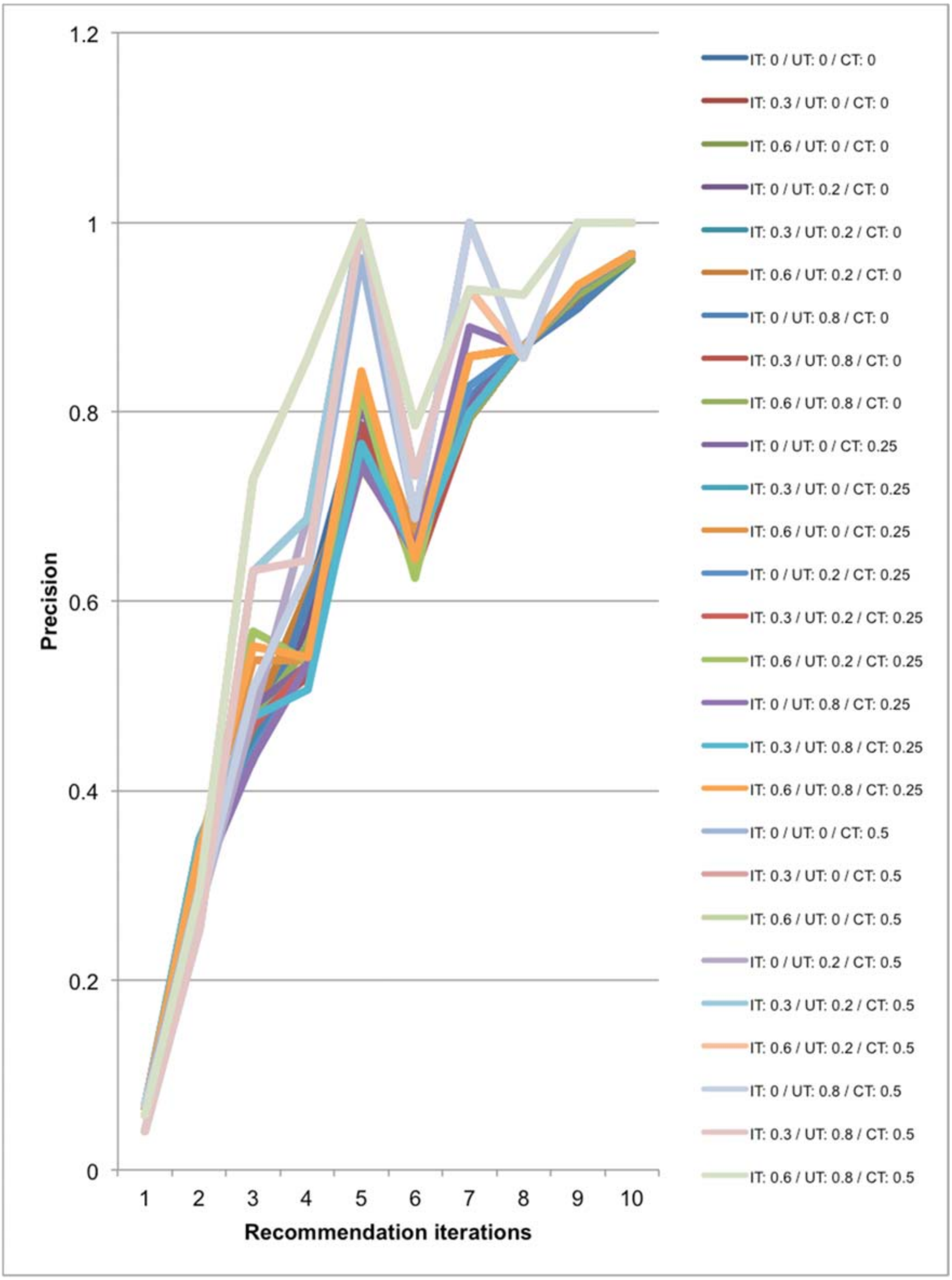


Figure 26: Experiment 1c: Average precision over different recommendation iterations of different recommender system functions for all users

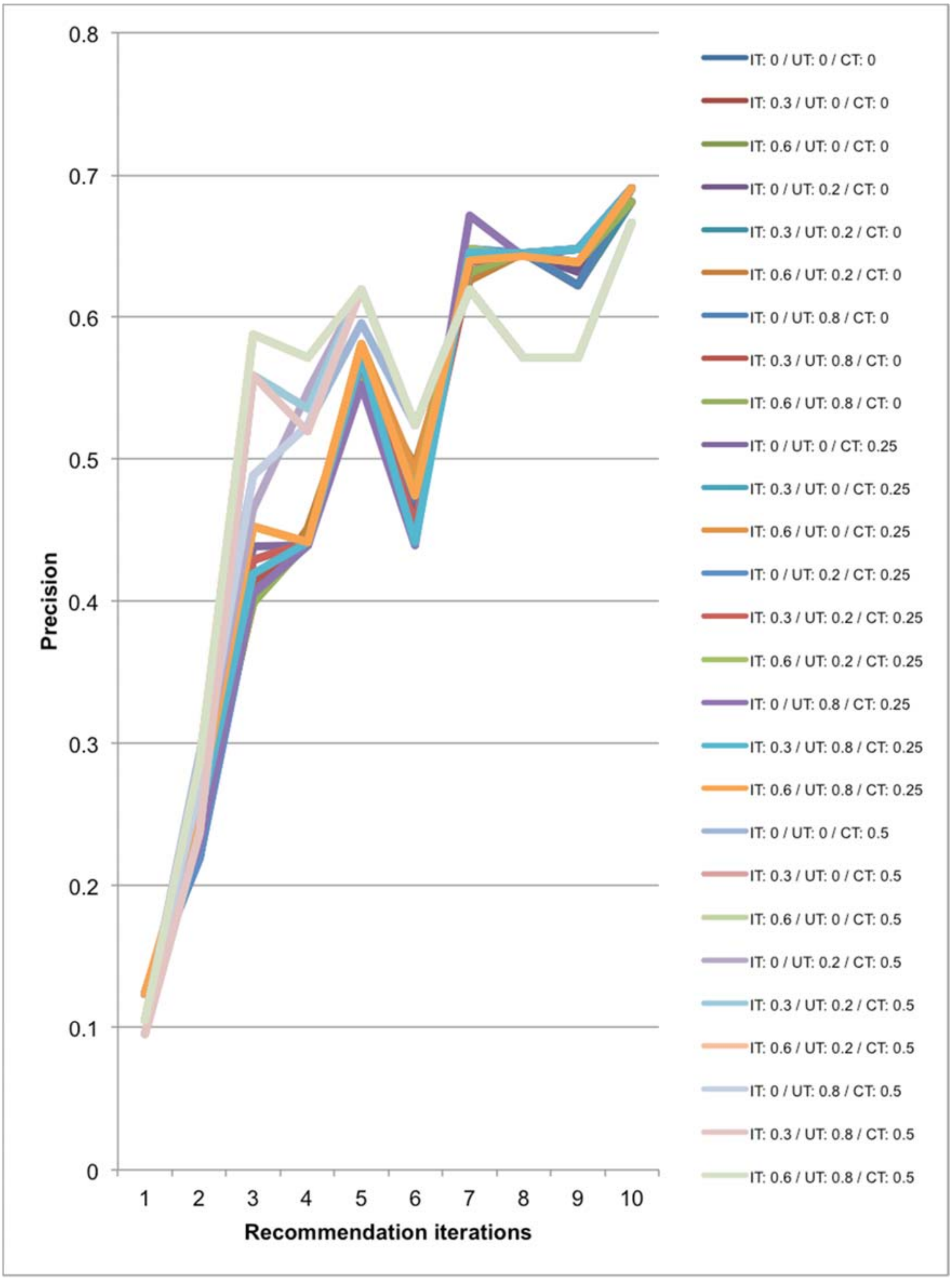


Figure 27: Experiment 1c: Average precision for tables over different recommendation iterations of different recommender system functions for all users

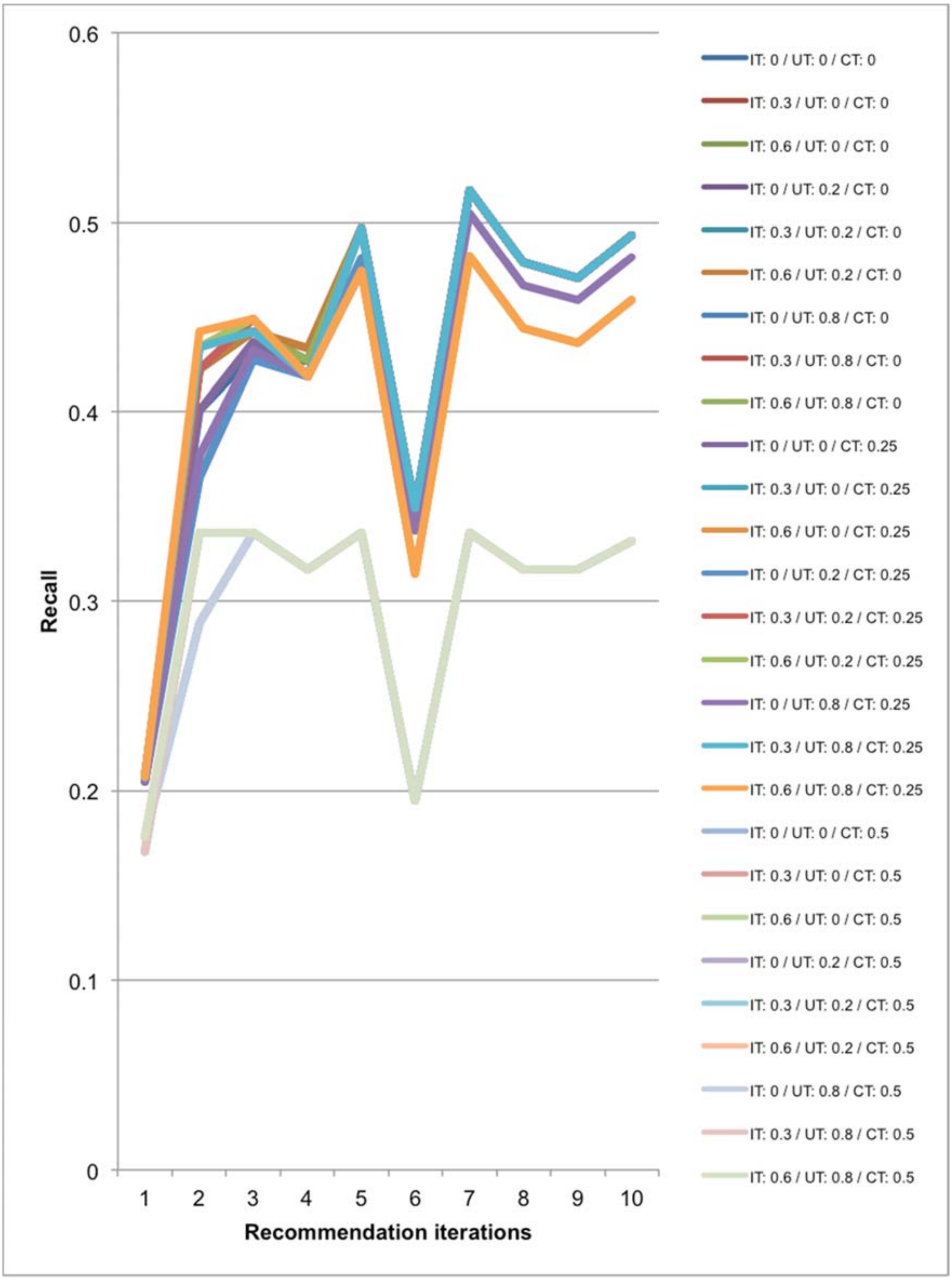


Figure 28: Experiment 1c: Average recall for tables over different recommendation iterations of different recommender system functions for all users

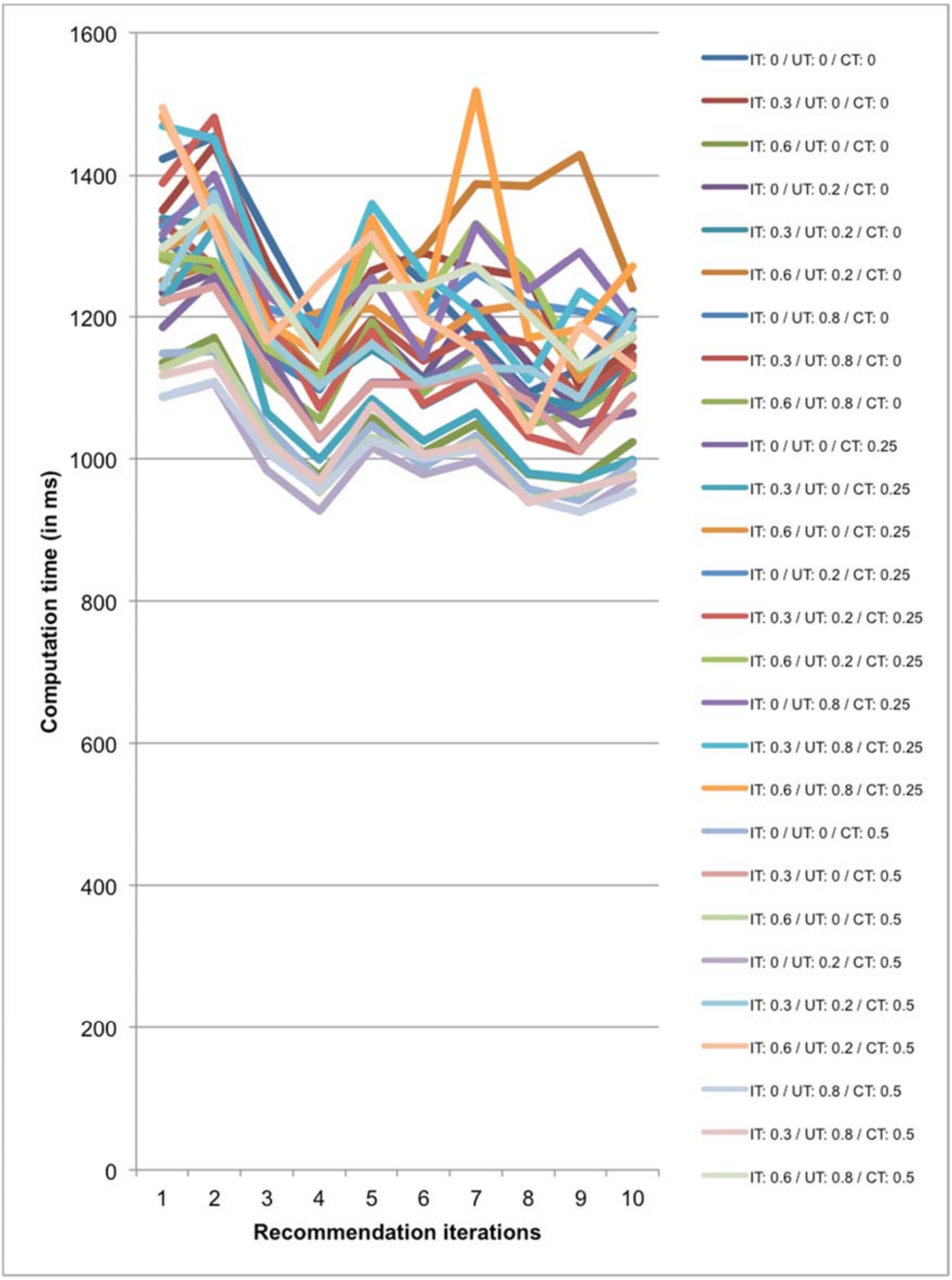


Figure 29: Experiment 1c: Average computation time over different recommendation iterations of different recommender system functions for all users

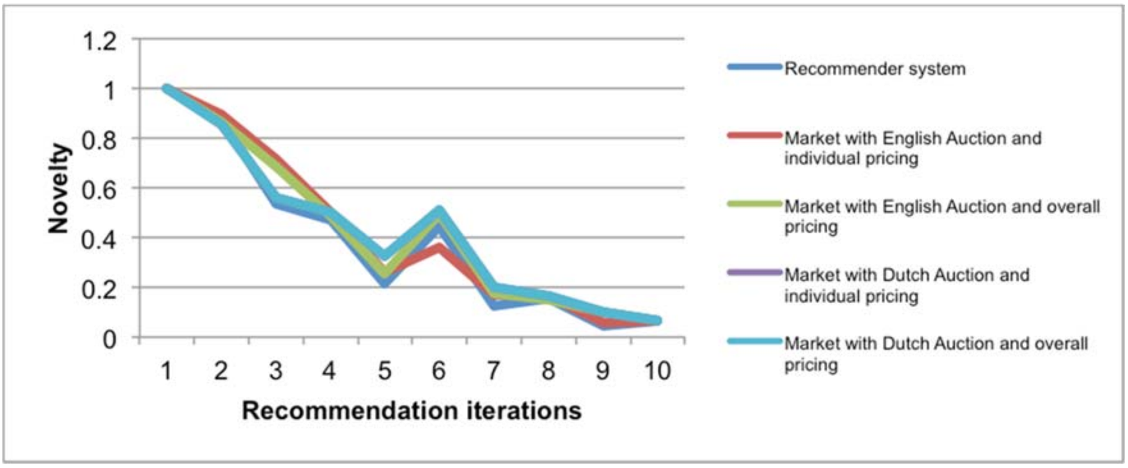


Figure 30: Experiment 2: Average novelty over different recommendation iterations of different recommender system functions for all users

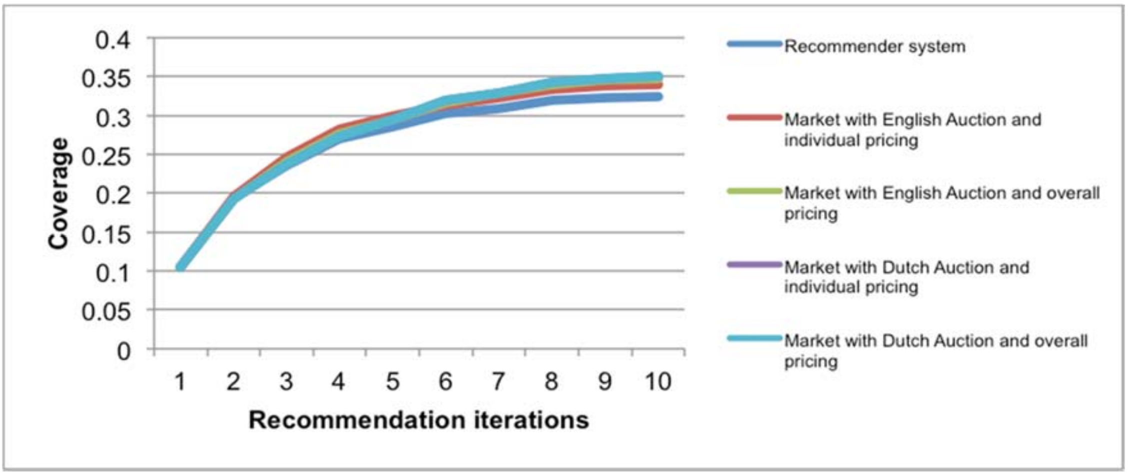


Figure 31: Experiment 2: Average coverage over different recommendation iterations of different recommender system functions for all users

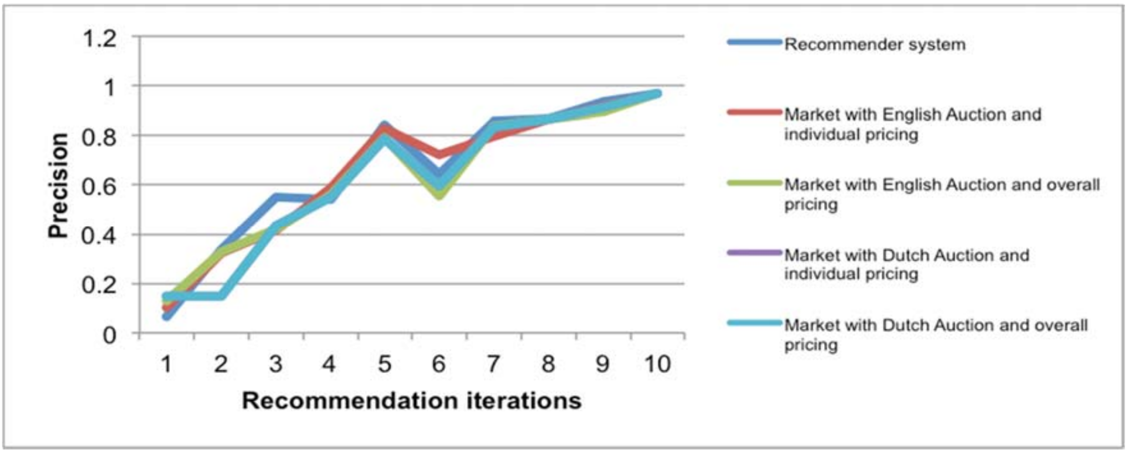


Figure 32: Experiment 2: Average precision over different recommendation iterations of different recommender system functions for all users

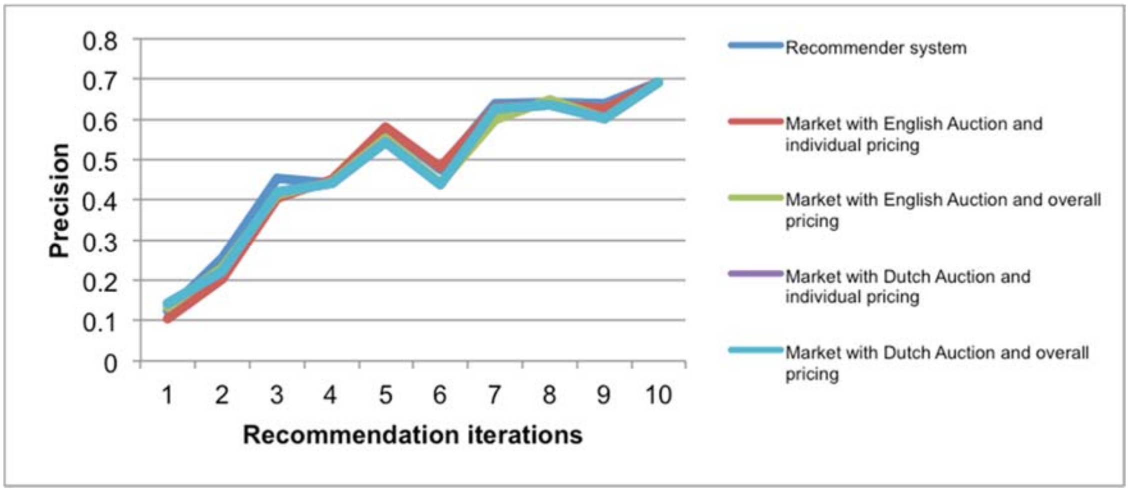


Figure 33: Experiment 2: Average precision for tables over different recommendation iterations of different recommender system functions for all users

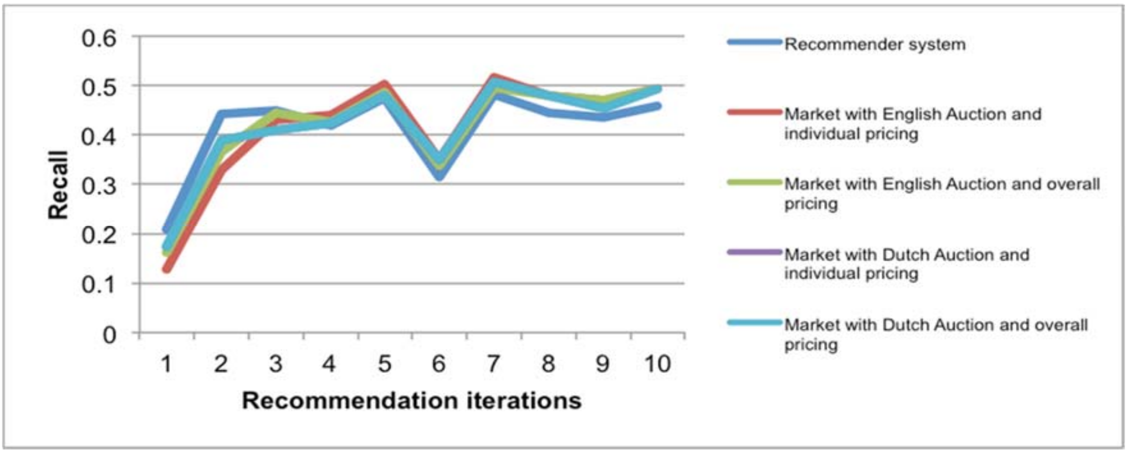


Figure 34: Experiment 2: Average recall for tables over different recommendation iterations of different recommender system functions for all users

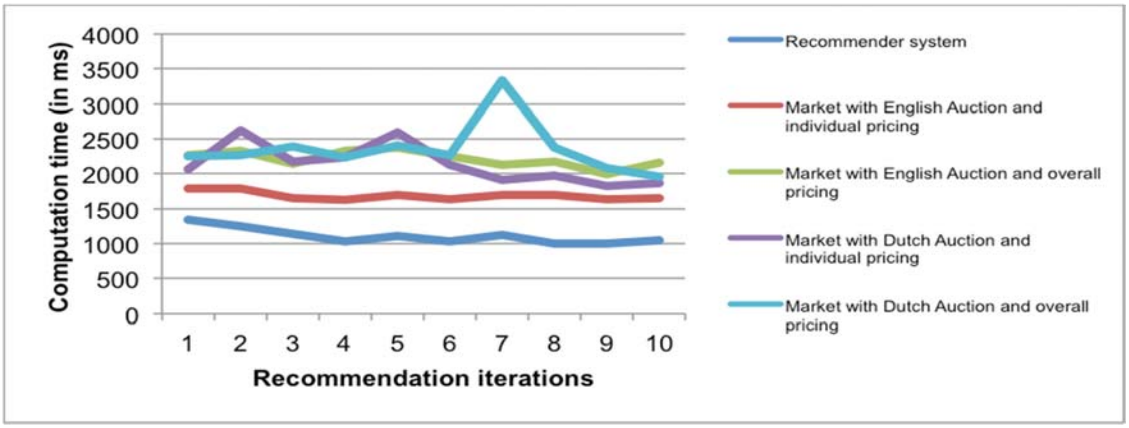


Figure 35: Experiment 2: Average computation time over different recommendation iterations of different recommender system functions for all users

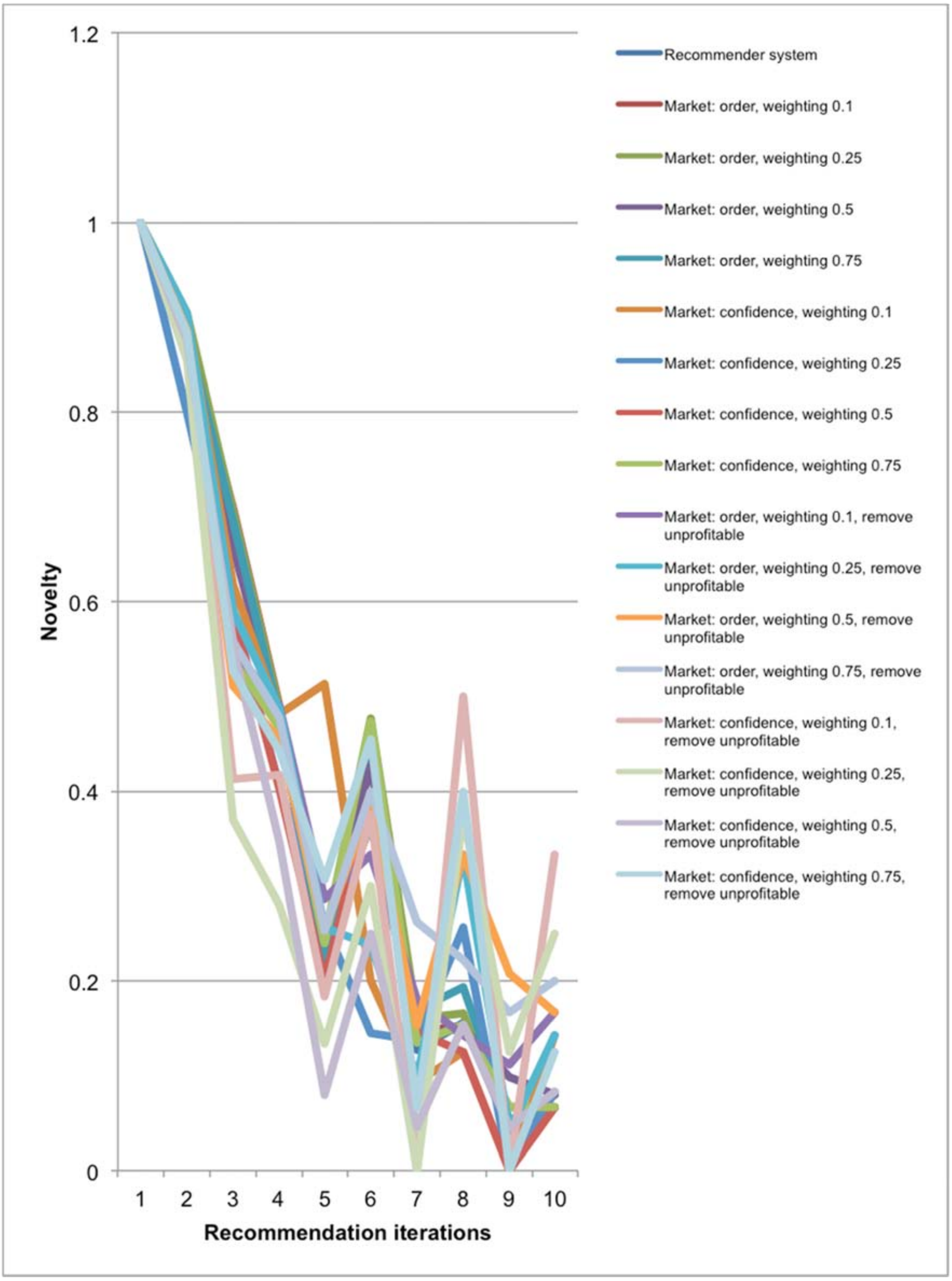


Figure 36: Experiment 3: Average novelty over different recommendation iterations of different recommender system functions for all users

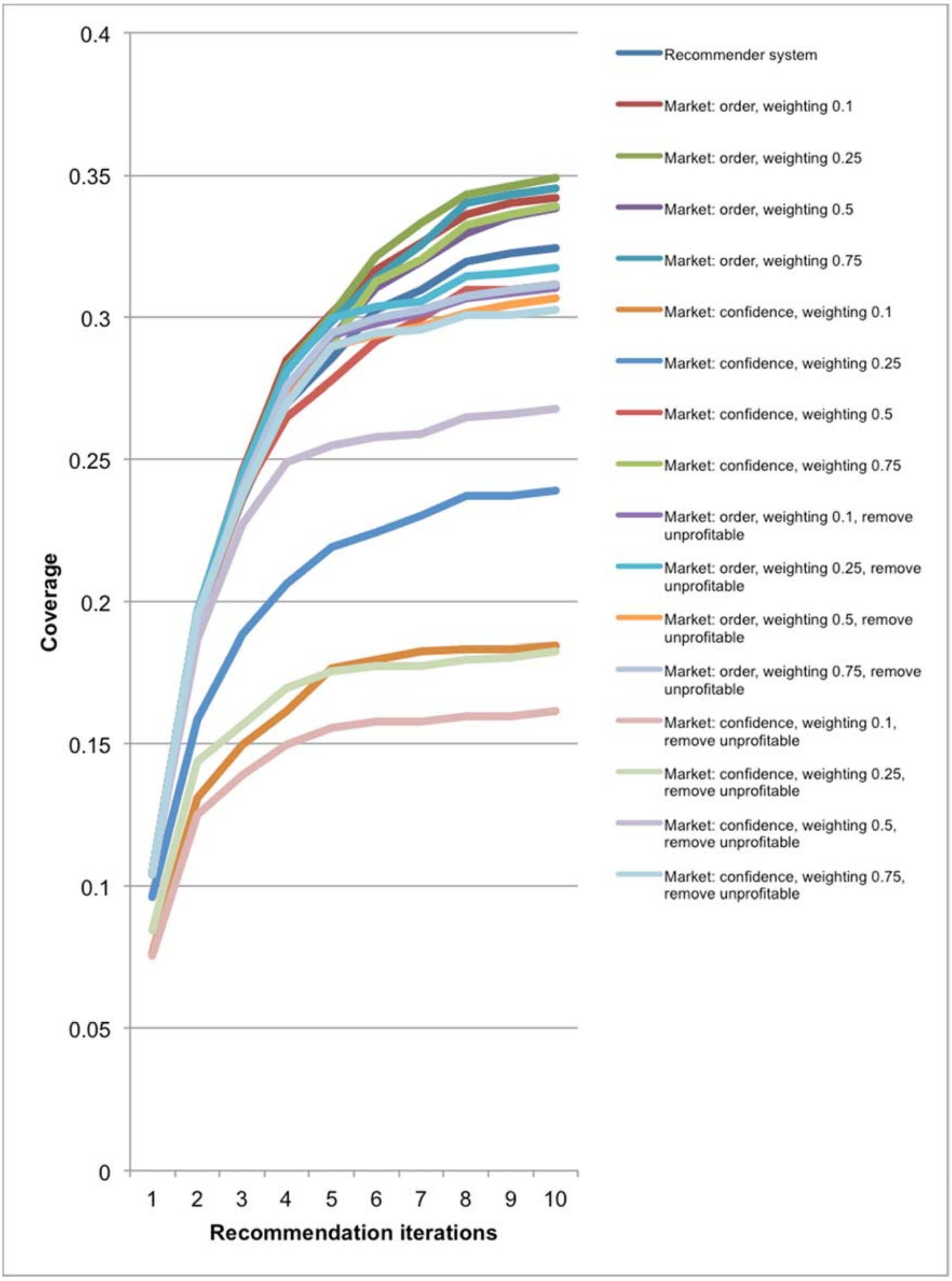


Figure 37: Experiment 3: Average coverage over different recommendation iterations of different recommender system functions for all users

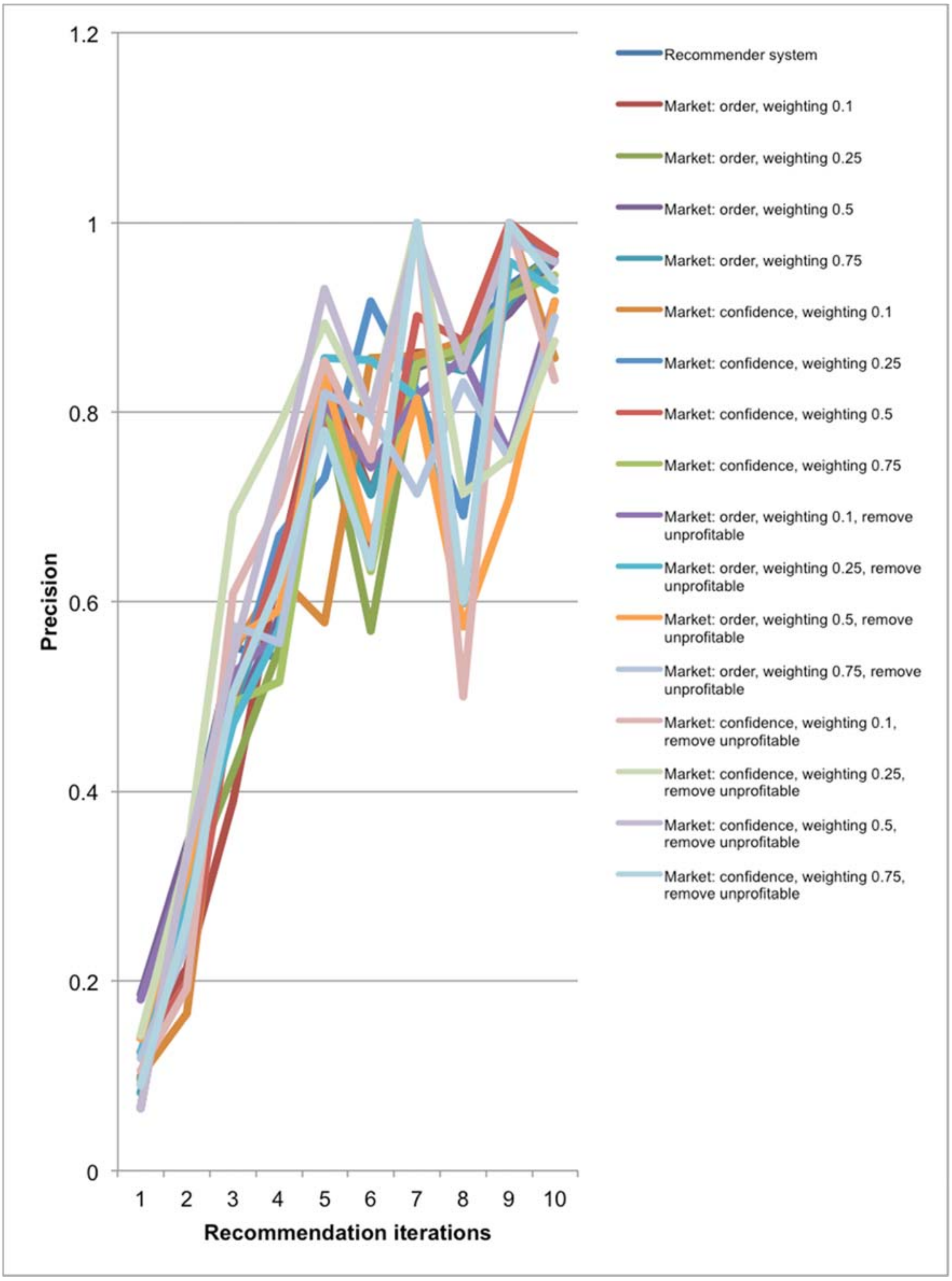


Figure 38: Experiment 3: Average precision over different recommendation iterations of different recommender system functions for all users

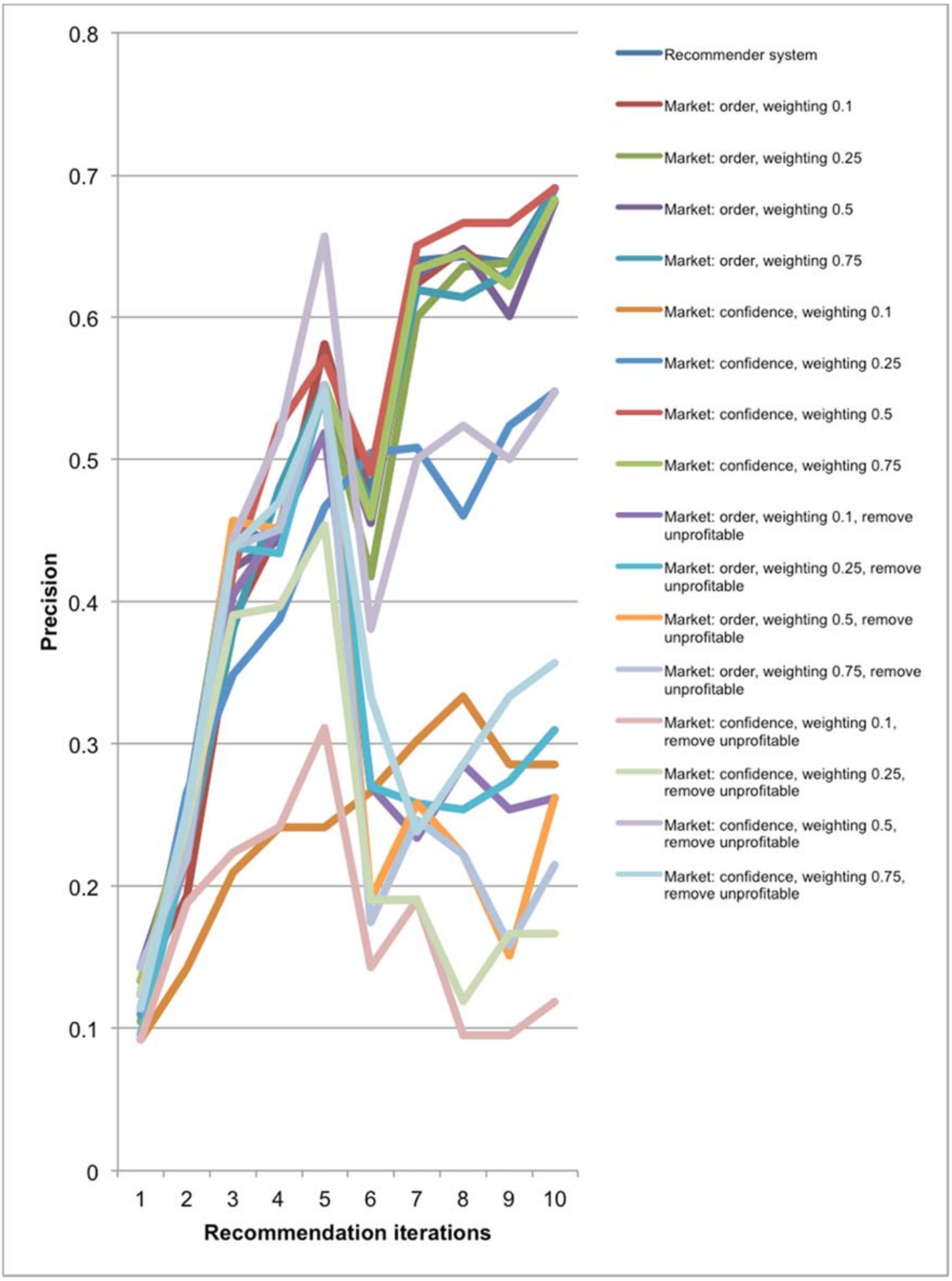


Figure 39: Experiment 3: Average precision for tables over different recommendation iterations of different recommender system functions for all users

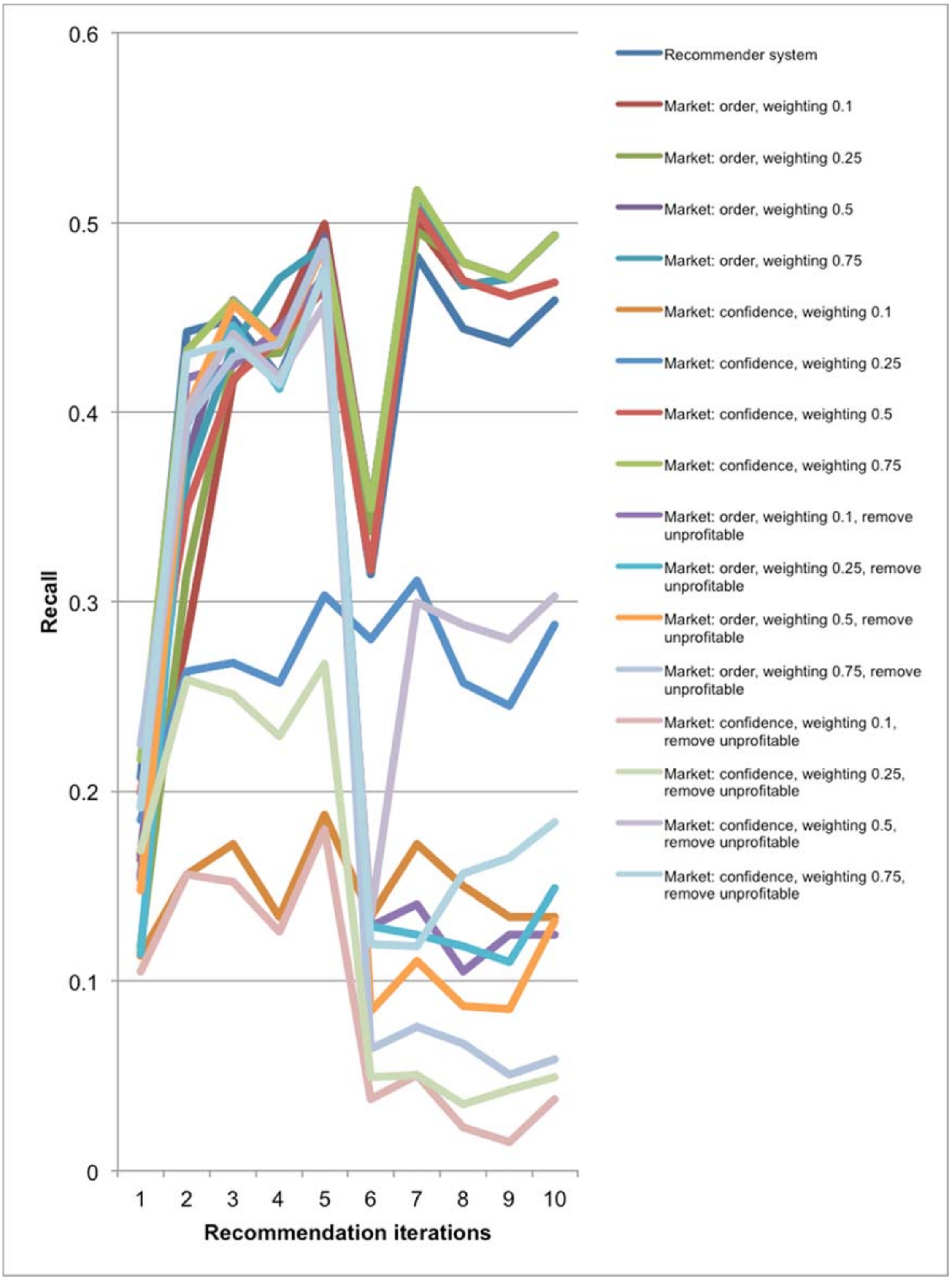


Figure 40: Experiment 3: Average recall for tables over different recommendation iterations of different recommender system functions for all users

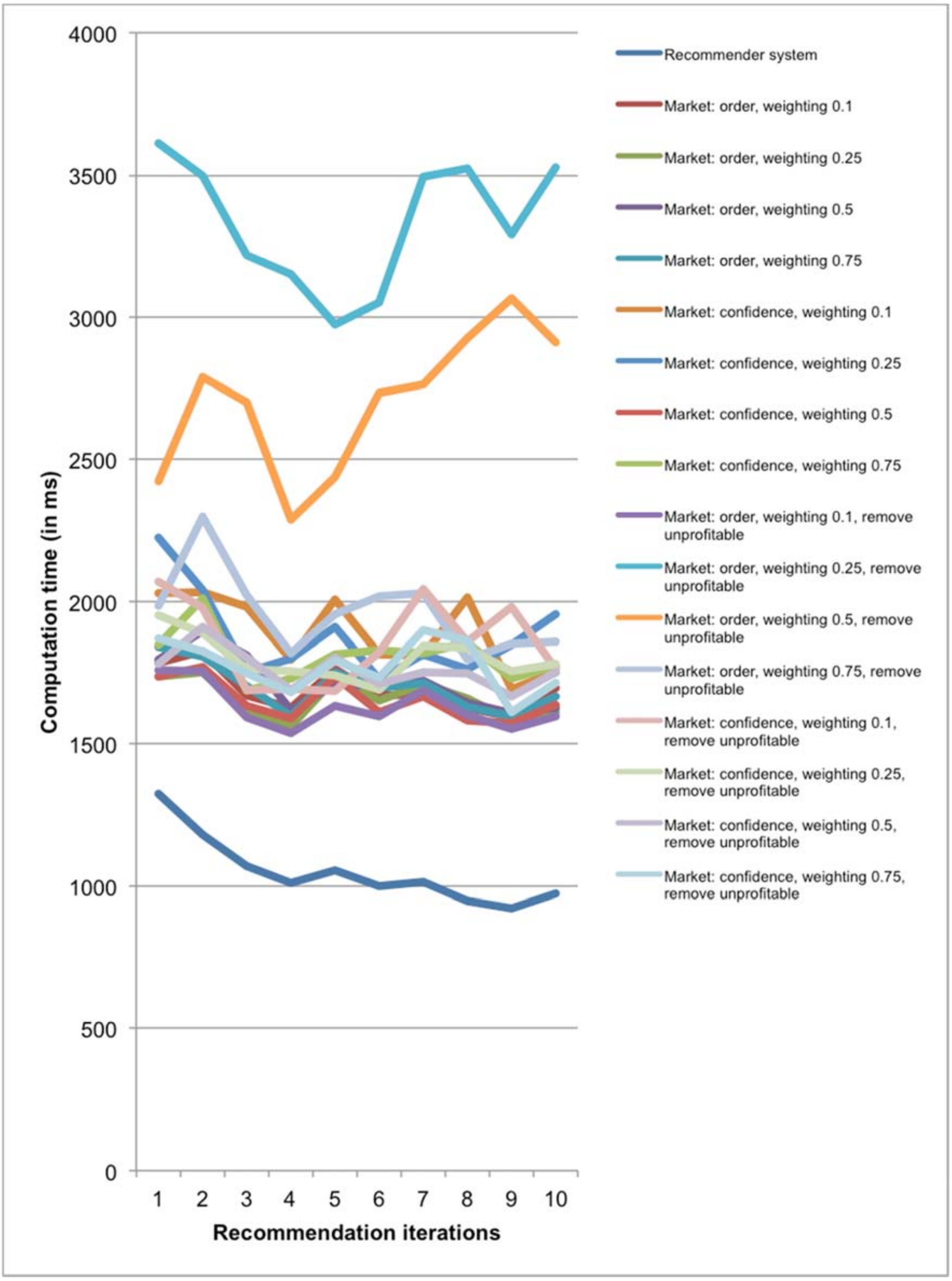


Figure 41: Experiment 3: Average computation time over different recommendation iterations of different recommender system functions for all users

System ID	Rating behaviour			Novelty	Coverage	Precision for Rows	Precision for Tables	Recall for Tables	Computation time
	Rating frequ.	# of rating reco.	Rating accura.						
1	1	All	Extreme	0.44	0.27	0.62	0.49	0.41	3575ms
2				0.46	0.28	0.62	0.48	0.42	1697ms
3				0.45	0.25	0.61	0.47	0.39	1667ms
4				0.45	0.28	0.61	0.48	0.43	1653ms
1	3			0.3	0.21	0.28	0.26	0.34	1034ms
2				0.31	0.2	0.30	0.26	0.34	1806ms
3				0.28	0.19	0.32	0.27	0.35	1824ms
4				0.3	0.	0.29	0.27	0.36	1828ms
1	5			0.22	0.17	0.16	0.17	0.32	1034ms
2				0.22	07	0.19	0.19	0.33	1744ms
3				0.23	0.16	0.2	0.2	0.3	1745ms
4				0.23	0.17	0.16	0.16	0.31	1808ms
1	10			0.14	0.14	0.07	0.1	0.24	1037ms
2				0.14	0.14	0.1	0.11	0.2	1624ms
3				0.14	0.14	0.05	0.1	0.24	1631ms
4				0.14	0.14	0.05	0.1	0.24	1627ms
1	1	Top		0.22	0.18	0.15	0.18	0.32	1039ms
2				0.23	0.19	0.18	0.19	0.27	1761ms
3				0.22	0.18	0.14	0.18	0.32	1742ms
4				0.22	0.18	0.14	0.18	0.32	1765ms
1		Top3		0.37	0.24	0.44	0.3	0.37	1029ms
2				0.38	0.24	0.44	0.35	0.38	1780ms
3				0.38	0.23	0.42	0.32	0.37	1781ms
4				0.37	0.24	0.42	0.34	0.38	1813ms
1		Right reco.		0.12	0.12	0.05	0.7	0.13	1058ms
2				0.12	0.12	0.05	0.09	0.17	1617ms
3				0.12	0.12	0.05	0.07	0.13	1652ms
4				0.12	0.12	0.05	0.07	0.13	1736ms
1		Wrong reco.		0.46	0.28	0.6	0.48	0.39	1034ms
2				0.49	0.29	0.6	0.4	0.4	2438ms
3				0.49	0.27	0.57	0.44	0.36	2606ms
4				0.47	0.28	0.58	0.47	0.38	3268ms
1		Rando m		0.27	0.21	0.08	0.09	0.17	1537ms
2				0.28	0.2	0.11	0.09	0.13	2420ms
3				0.25	0.19	0.11	0.11	0.15	4135ms
4				0.28	0.2	0.1	0.11	0.16	5245ms
1		No reco.		0.12	0.12	0.06	0.08	0.15	3404ms
2				0.12	0.12	0.05	0.1	0.17	2401ms
3				0.12	0.12	0.05	0.08	0.14	1903ms
4				0.12	0.12	0.06	0.08	0.15	1869ms
1		All	Strong	0.36	0.24	0.41	0.33	0.39	1133ms
2				0.37	0.24	0.39	0.31	0.39	2537ms
3				0.35	0.24	0.39	0.32	0.39	2301ms
4				0.36	0.24	0.45	0.37	0.39	2891ms
1			Medium	0.14	0.14	0.05	0.09	0.16	2092ms
2				0.14	0.14	0.08	0.12	0.19	3045ms
3				0.14	0.14	0.05	0.09	0.16	2668ms
4				0.14	0.14	0.05	0.09	0.16	3088ms

1			Neutral	0.15	0.14	0.06	0.12	0.26	1764ms
2				0.15	0.14	0.07	0.1	0.18	2896ms
3				0.15	0.15	0.06	0.12	0.27	3039ms
4				0.15	0.14	0.05	0.11	0.25	2008ms
1			Positive	0.14	0.14	0.05	0.1	0.18	1480ms
2				0.14	0.14	0.08	0.11	0.19	2725ms
3				0.14	0.14	0.05	0.1	0.18	2291ms
4				0.14	0.14	0.05	0.1	0.18	1955ms
1			Negative	0.45	0.28	0.59	0.48	0.4	1164ms
2				0.46	0.28	0.61	0.48	0.42	2719ms
3				0.42	0.25	0.64	0.51	0.39	2503ms
4				0.45	0.27	0.61	0.48	0.43	2756ms
1			Neutral	0.14	0.13	0.04	0.07	0.14	1664ms
2				0.13	0.13	0.04	0.09	0.12	2536ms
3				0.14	0.13	0.04	0.07	0.14	2058ms
4				0.14	0.13	0.04	0.07	0.14	1895ms

Table 39: Experiment 4 evaluations, Average experiment results

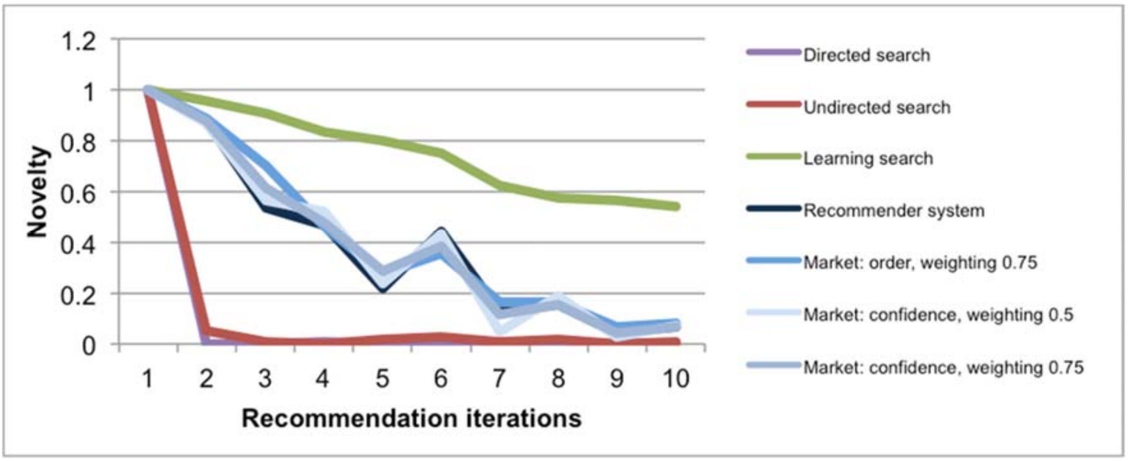


Figure 42: Search: Average novelty over different recommendation iterations of different recommender system functions for all users

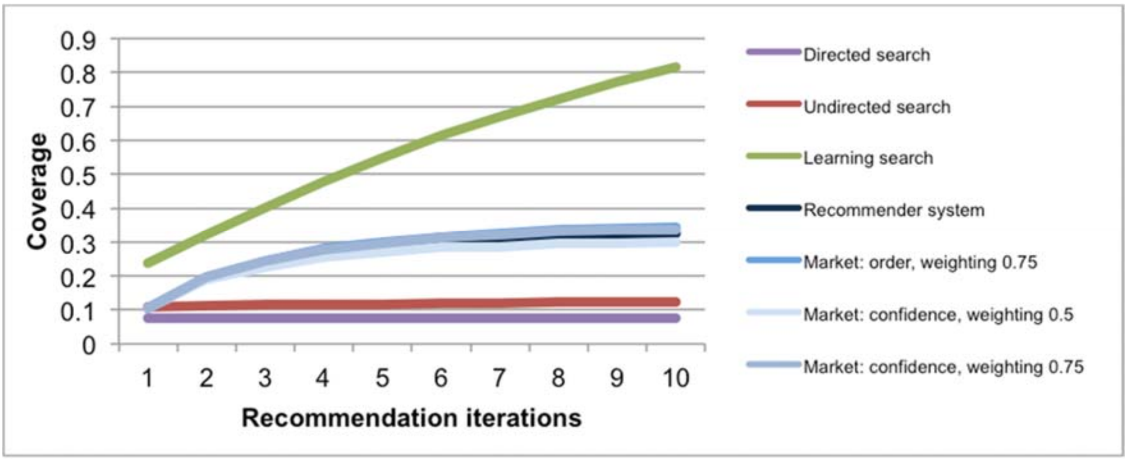


Figure 43: Search: Average coverage over different recommendation iterations of different recommender system functions for all users

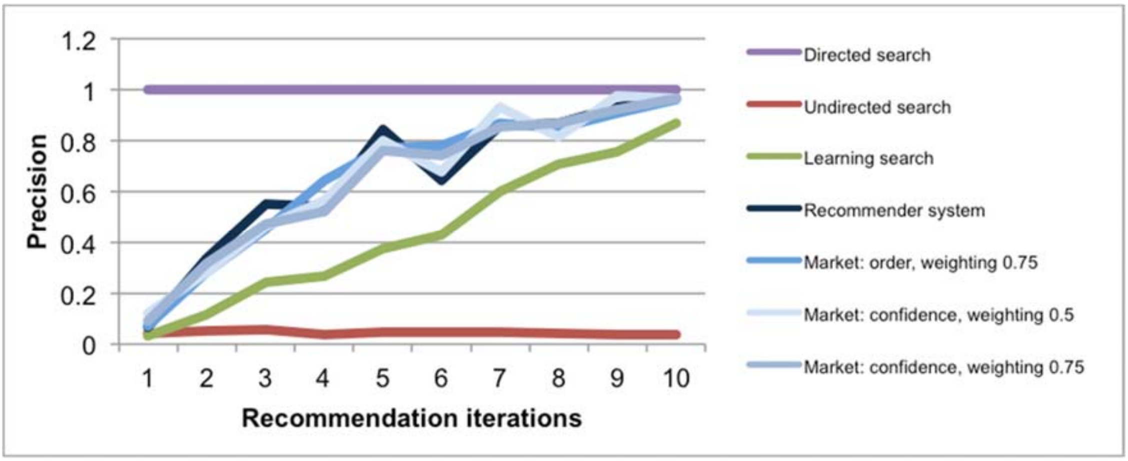


Figure 44: Search: Average precision over different recommendation iterations of different recommender system functions for all users

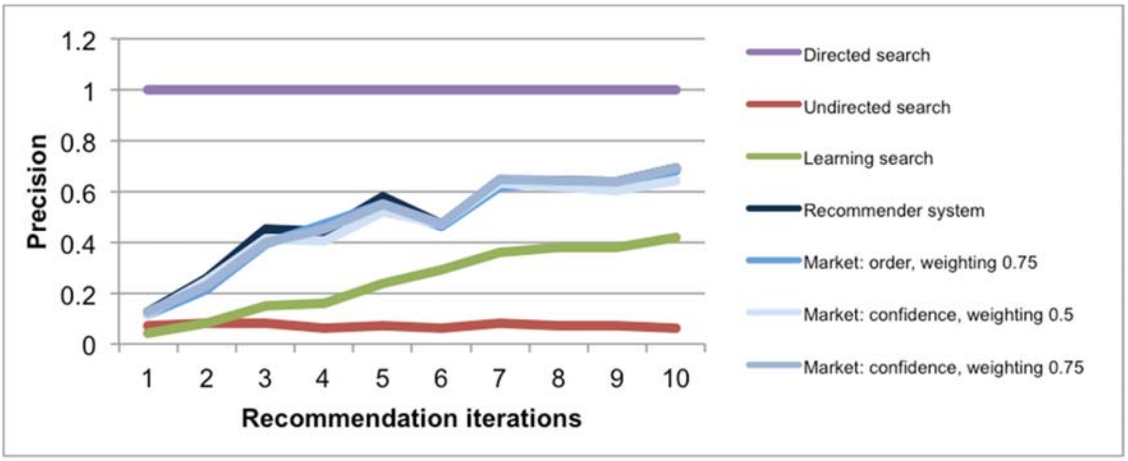


Figure 45: Search: Average precision for tables over different recommendation iterations of different recommender system functions for all users

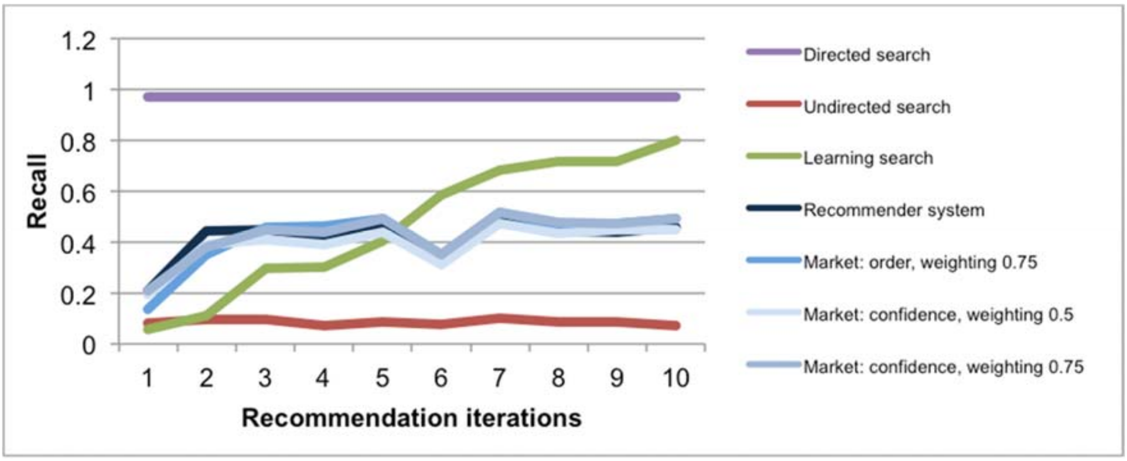


Figure 46: Search: Average recall for tables over different recommendation iterations of different recommender system functions for all users

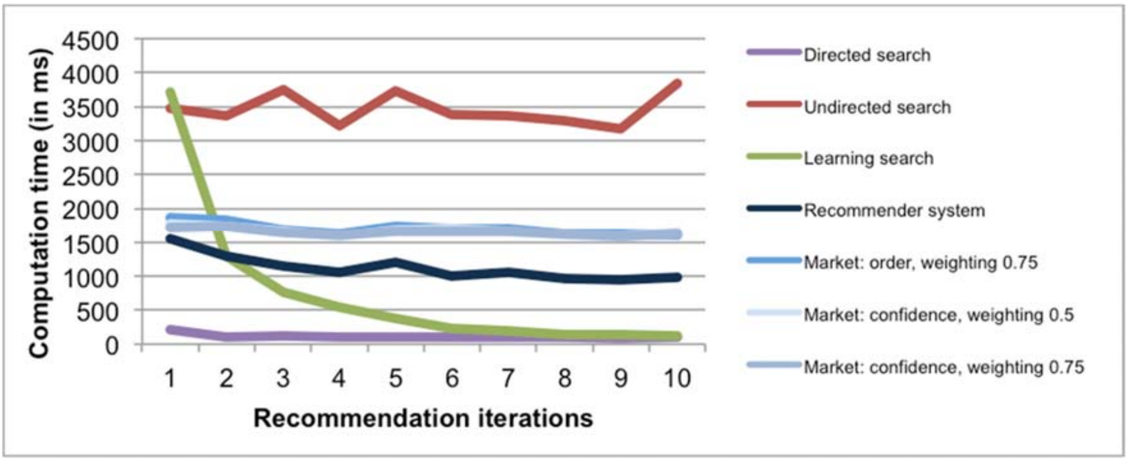


Figure 47: Search: Average computation time over different recommendation iterations of different recommender system functions for all users

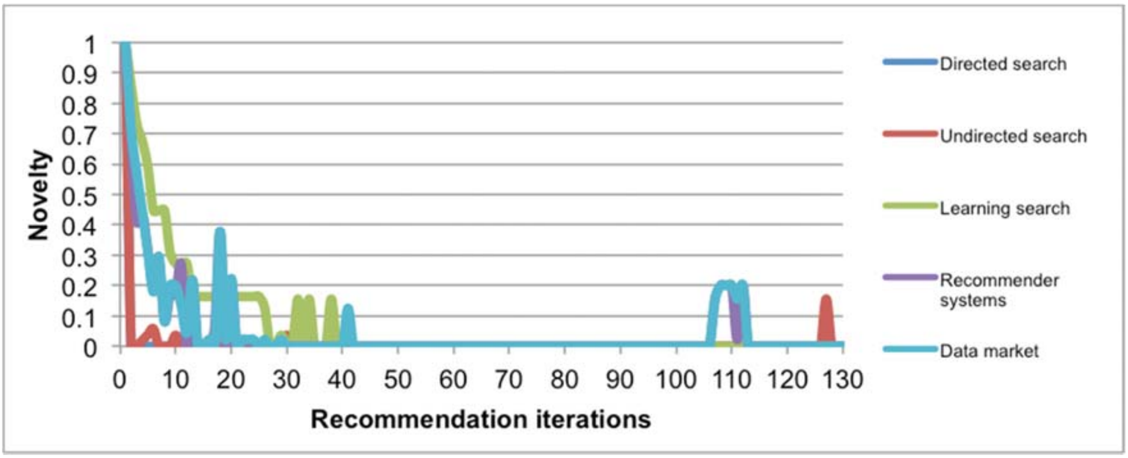


Figure 48: Case Study A: Average novelty over different recommendation iterations of different recommender system functions for all users

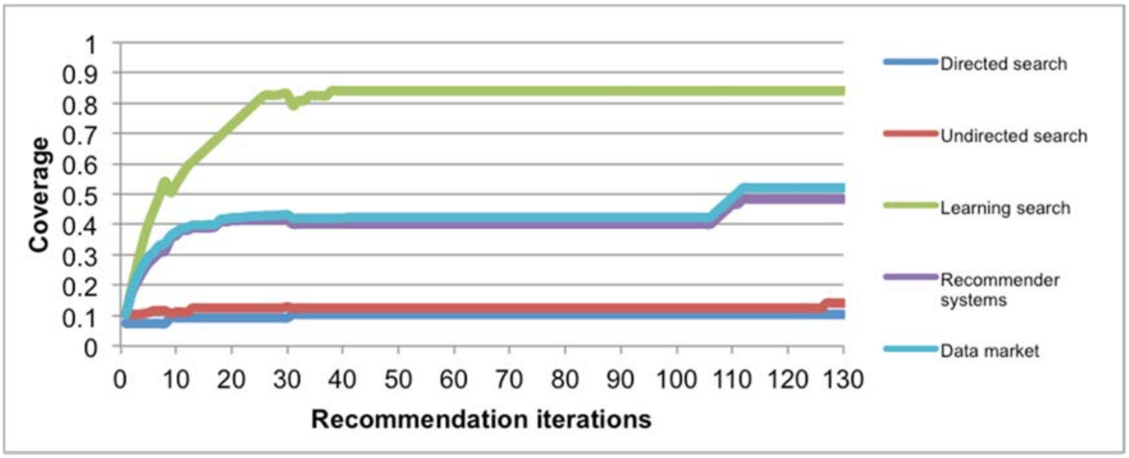


Figure 49: Case Study A: Average coverage over different recommendation iterations of different recommender system functions for all users

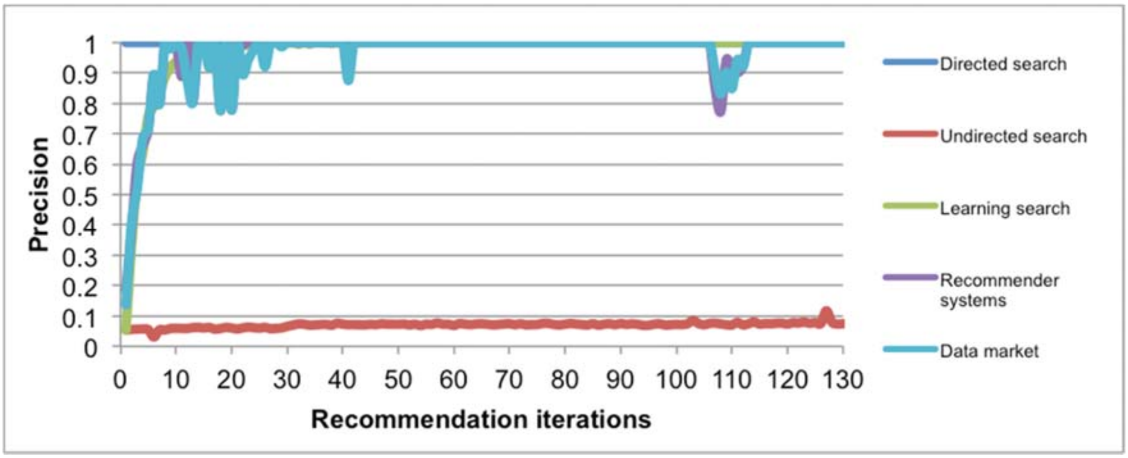


Figure 50: Case Study A: Average precision over different recommendation iterations of different recommender system functions for all users

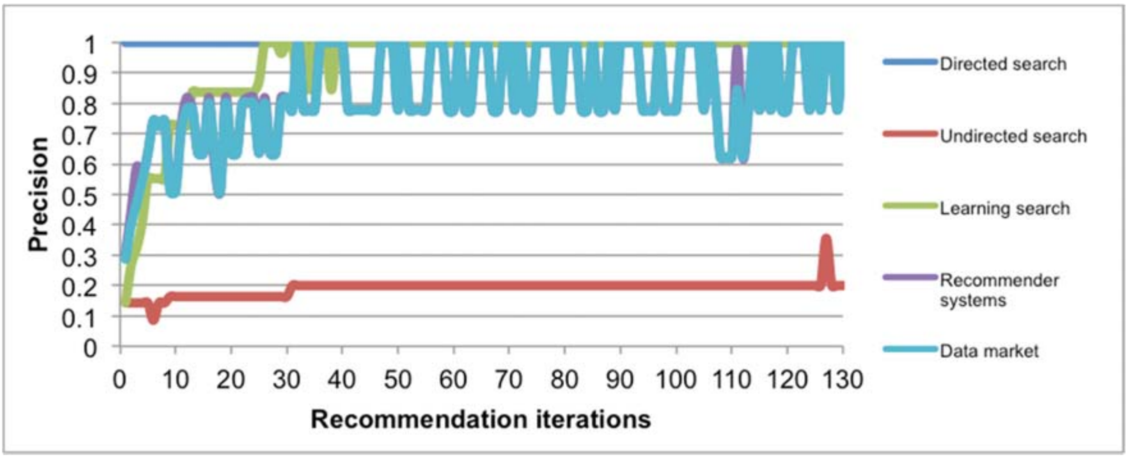


Figure 51: Case Study A: Average precision for tables over different recommendation iterations of different recommender system functions for all users

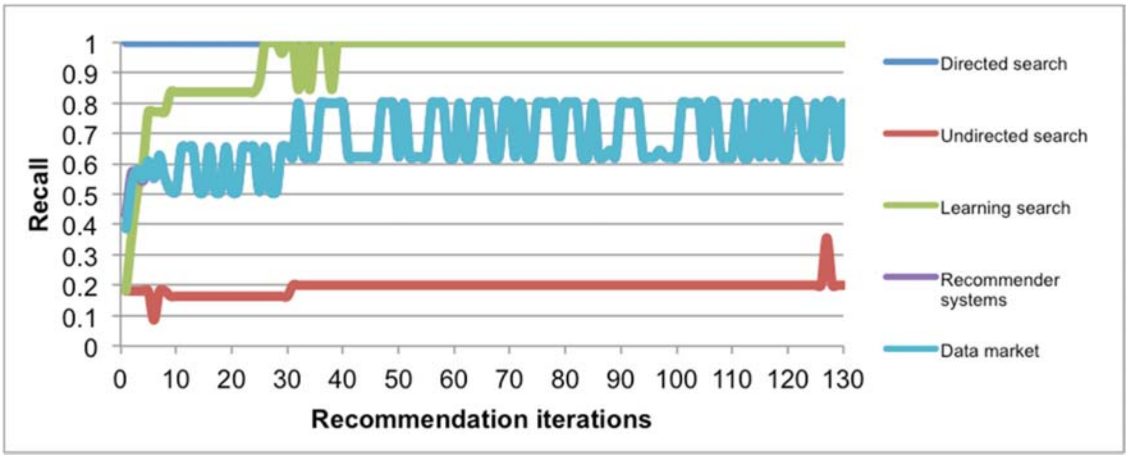


Figure 52: Case Study A: Average recall for tables over different recommendation iterations of different recommender system functions for all users

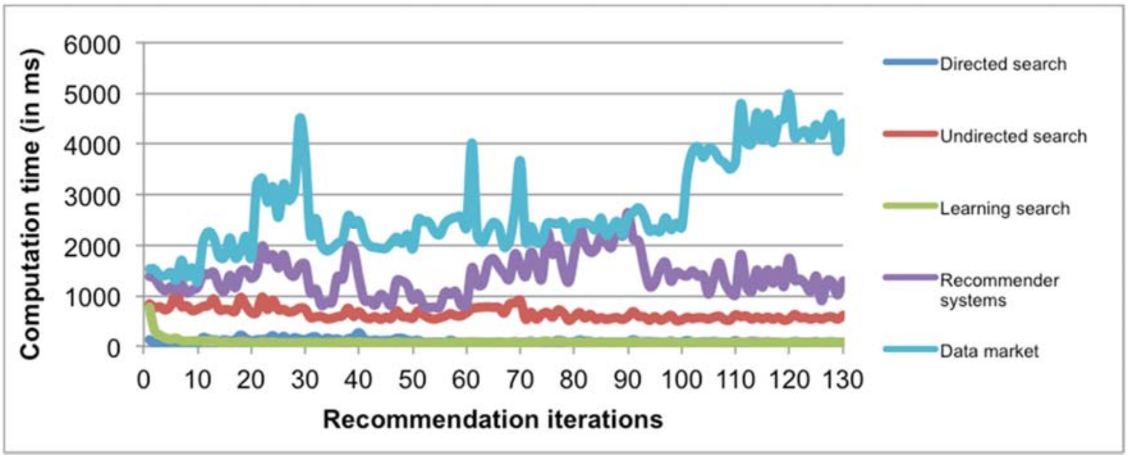


Figure 53: Case Study A: Average computation time over different recommendation iterations of different recommender system functions for all users

System	Rating behaviour			Novelty	Coverage	Precision for Rows	Precision for Tables	Recall for Tables	Computation time
	Rating fre.	# of rating reco.	Rating accura.						
Direct				0.02	0.1	1	1	1	104ms
Undirect				0.02	0.12	0.07	0.19	0.19	661ms
Learning				0.12	0.75	0.95	0.9	0.92	101ms
Recom.	1	All	Extreme	0.08	0.39	0.94	0.81	0.66	1387ms
Market				0.08	0.41	0.93	0.81	0.66	2626ms
Recom.	3			0.07	0.38	0.86	0.74	0.65	1451ms
Market				0.07	0.39	0.86	0.73	0.65	2218ms
Recom.	5			0.06	0.36	0.79	0.68	0.64	1038ms
Market				0.06	0.37	0.8	0.67	0.65	2228ms
Recom.	10			0.05	0.34	0.65	0.59	0.57	1047ms
Market				0.06	0.35	0.74	0.6	0.6	2119ms
Recom.	1	Top		0.05	0.27	0.47	0.29	0.3	1055ms
Market				0.04	0.25	0.69	0.31	0.32	2128ms
Recom.		Top3		0.06	0.32	0.78	0.65	0.65	1047ms
Market				0.06	0.29	0.8	0.5	0.51	3430ms
Recom.		Right reco.		0.03	0.18	0.05	0.11	0.11	1648ms
Market				0.02	0.15	0.09	0.17	0.18	4593ms
Recom.		Wrong reco.		0.08	0.35	0.93	0.81	0.5	1444ms
Market				0.09	0.42	0.93	0.81	0.66	2366ms
Recom.		Random		0.17	0.4	0.18	0.05	0.04	1052ms
Market				0.17	0.41	0.09	0.03	0.03	3853ms
Recom.		No reco.		0.03	0.18	0.05	0.11	0.11	1282ms
Market				0.02	0.15	0.09	0.17	0.18	2318ms
Recom.		All	Strong	0.07	0.4	0.88	0.77	0.66	1169ms
Market				0.08	0.4	0.89	0.77	0.66	2515ms
Recom.			Medium	0.03	0.18	0.13	0.27	0.27	1275ms
Market				0.02	0.17	0.58	0.32	0.33	2371ms
Recom.			Neutral	0.03	0.22	0.09	0.16	0.17	1271ms
Market				0.03	0.19	0.26	0.24	0.25	2570ms
Recom.			Positive	0.03	0.18	0.14	0.25	0.26	1296ms
Market				0.02	0.16	0.58	0.32	0.33	2373ms
Recom.			Negative	0.08	0.39	0.94	0.81	0.65	1165ms
Market				0.09	0.42	0.93	0.81	0.66	2316ms
Recom.			Neutral	0.03	0.19	0.05	0.1	0.11	1309ms
Market				0.02	0.14	0.1	0.17	0.18	2379ms

Table 40: Case A evaluations (Details), Average of experiment results

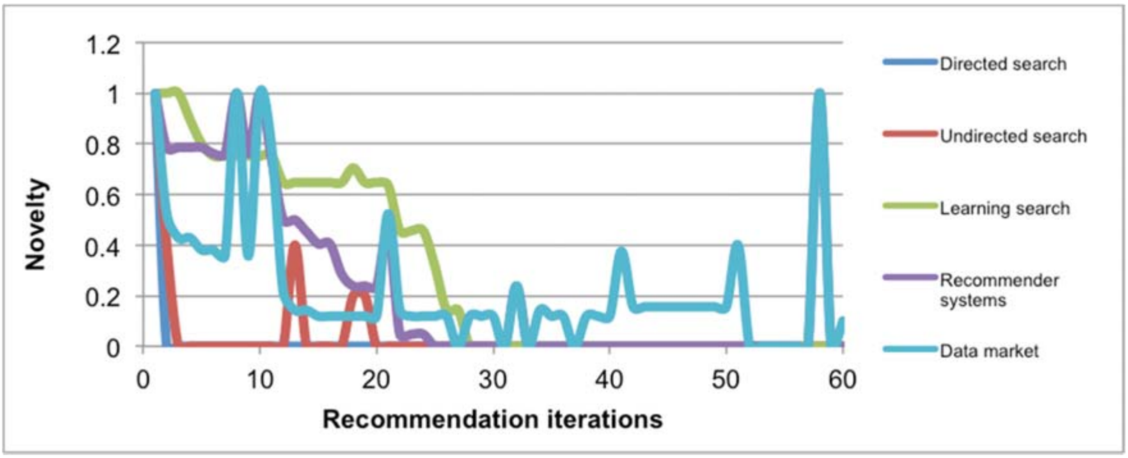


Figure 54: Case Study B: Average novelty over different recommendation iterations of different recommender system functions for all users

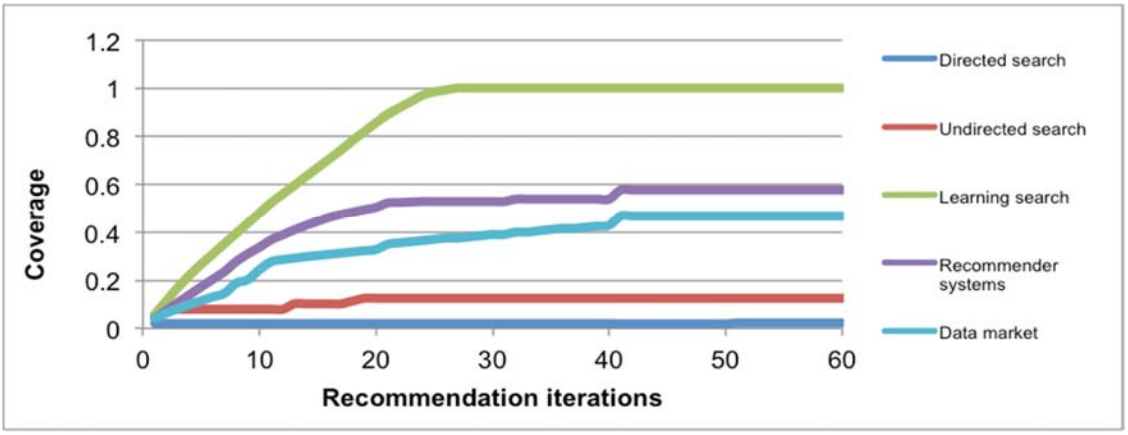


Figure 55: Case Study B: Average coverage over different recommendation iterations of different recommender system functions for all users

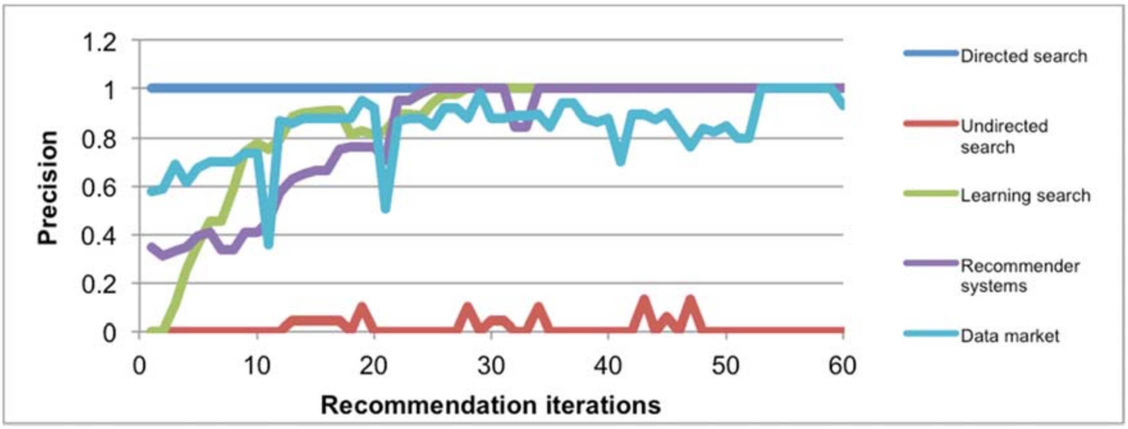


Figure 56: Case Study B: Average precision over different recommendation iterations of different recommender system functions for all users

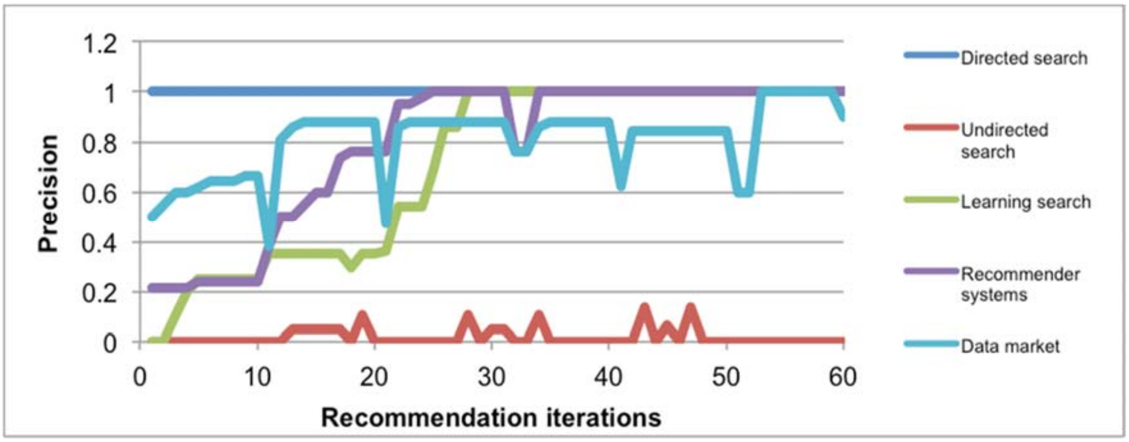


Figure 57: Case Study B: Average precision for tables over different recommendation iterations of different recommender system functions for all users

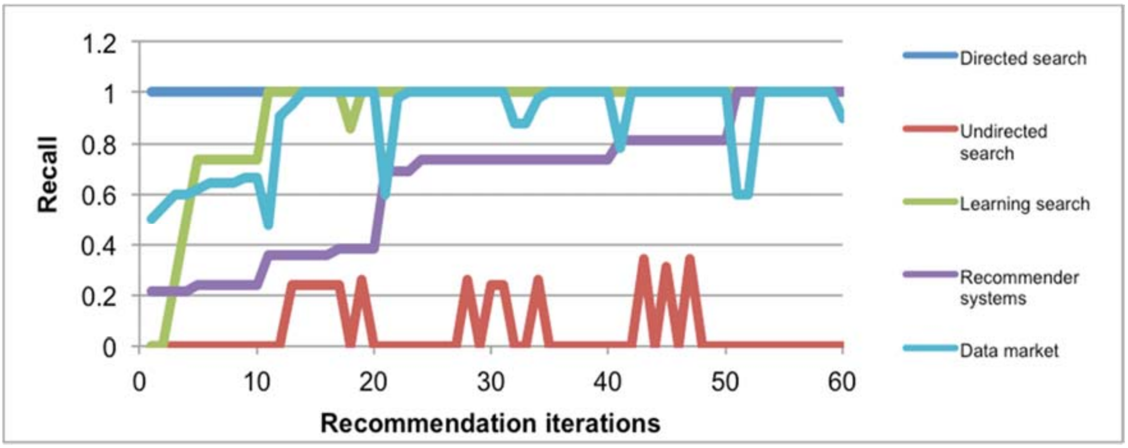


Figure 58: Case Study B: Average recall for tables over different recommendation iterations of different recommender system functions for all users

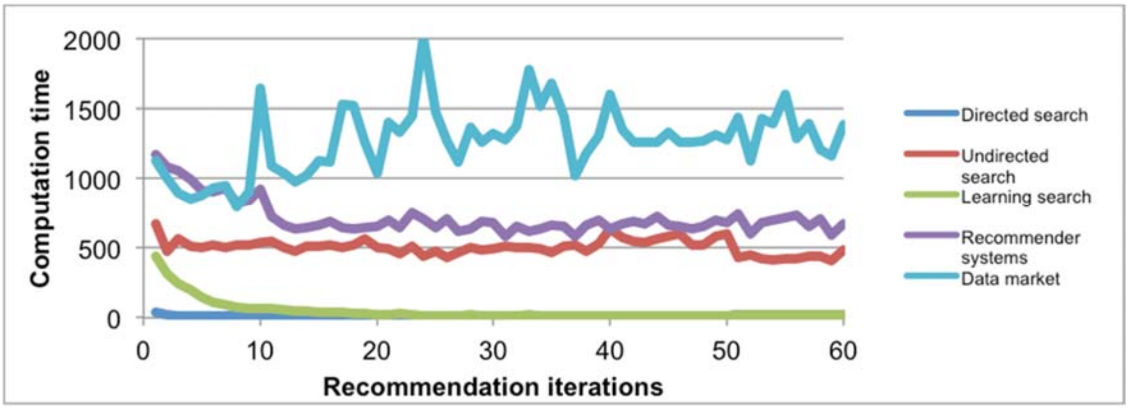


Figure 59: Case Study B: Average computation time over different recommendation iterations of different recommender system functions for all users

System	Rating behaviour			No-velty	Cover-age	Pre-cision for Rows	Pre-cision for Tables	Recall for Tables	Compu-tation time
	Ra-ting fre.	# of rating reco.	Rating accura.						
Direct				0.02	0.02	1	1	1	16ms
Undirect				0.04	0.11	0.02	0.02	0.06	515ms
Learning				0.35	0.79	0.83	0.65	0.9	54ms
Recom.	1	All	Extreme	0.3	0.45	0.75	0.65	0.5	725ms
Market				0.26	0.33	0.78	0.69	0.77	1257ms
Recom.	3			0.17	0.22	0.6	0.55	0.55	741ms
Market				0.18	0.24	0.62	0.57	0.61	2260ms
Recom.	5			0.2	0.25	0.29	0.2	0.2	1230ms
Market				0.19	0.25	0.3	0.24	0.24	1893ms
Recom.	10			0.11	0.16	0.19	0.15	0.15	927ms
Market				0.12	0.16	0.02	0.01	0.01	1205ms
Recom.	1	Top		0.1	0.17	0.57	0.42	0.52	762ms
Market				0.18	0.27	0.56	0.41	0.47	1240ms
Recom.		Top3		0.3	0.45	0.75	0.65	0.5	1106ms
Market				0.26	0.33	0.78	0.69	0.77	1880ms
Recom.		Right reco.		0.04	0.08	0	0	0	1203ms
Market		Right reco.		0.04	0.08	0	0	0	1163ms
Recom.		Wrong reco.		0.6	0.65	0.56	0.38	0.44	709ms
Market		Wrong reco.		0.67	0.72	0.44	0.31	0.34	996ms
Recom.		Random		0.22	0.24	0.06	0.03	0.03	680ms
Market				0.22	0.28	0.02	0.02	0.02	1000ms
Recom.		No reco.		0.04	0.08	0	0	0	645ms
Market				0.04	0.08	0	0	0	941ms
Recom.		All	Strong	0.3	0.38	0.48	0.39	0.39	668ms
Market				0.31	0.35	0.55	0.4	0.4	1024ms
Recom.			Medium	0.07	0.12	0.42	0.42	0.42	672ms
Market				0.05	0.1	0	0	0	893ms
Recom.			Neutral	0.06	0.12	0.2	0.18	0.18	637ms
Market				0.06	0.11	0.03	0.02	0.02	904ms
Recom.			Positive	0.04	0.08	0	0	0	633ms
Market				0.04	0.08	0	0	0	895ms
Recom.			Negative	0.42	0.53	0.65	0.55	0.6	652ms
Market				0.28	0.36	0.78	0.68	0.76	997ms
Recom.			Neutral	0.04	0.08	0	0	0	650ms
Market				0.04	0.08	0	0	0	878ms

Table 41: Case B evaluations (Details), Average of experiment results

Attachment D: Formulas for evaluation measures

Measure	Formula	Description
Categorisation	$Precision = \frac{tp}{tp + fp}$ $Recall = \frac{tp}{tp + fn}$ $tp = true\ possitive\ (correct\ recommendations)$ $fp = false\ possitive\ (wrong\ recommendations)$ $fn = false\ negative\ (missed\ recommendations)$	Categorisation is typically measured with precision and recall. Precision measures the number of recommendations that are relevant to the user. Recall measures that number of recommendations out of all recommendations the user would want that are being presented to him. [41]
Rank accuracy	$Spearman's\ corelation$	Correlation is the established measure for

[41] The two main RecorDa variations present tables containing the rows matching the rows already presented to the user from the operational system. So when the user looks into a specific part and the RecorDa approach suggest orders as a relevant table it will not show all orders from this table, but just the specific rows that match for example. Precision and recall could therefore be measured based on the rows or the tables presented. The following example illustrates the difference between these two. *Example: There are 3 tables recommended Table A with 2 rows, Table B with 1 row, and Table C with 3 rows, and assuming Table A and B are correct recommendations while Table C is not a correct recommendation. However, Table D with 2 rows would have been the correct recommendation. This thesis would then get the following value for precision and recall:*
Table level:
Precision: {Table A, Table B} / {Table A, Table B, Table C} = 2/3
Recall: {Table A, Table B} / Table A, Table B, Table C} = 2/3
Row level:
Precision: {Row 1 and 2 Table A, Row 1 Table B} / {Row 1 and 2 Table B, Row 1 Table B, Row 1-3 Table C} = 1/2
Recall: {Row 1 and 2 Table A, Row 1 Table B} / {Row 1 and 2 Table B, Row 1 Table B, Row 1 and 2 Table D} = 2/5
While this calculation is possible for precision by just looking into all recommendations it is difficult for recall, because it would require calculating which rows would be presented if they were a recommendation. This would require a complete change of the design of the recommender system component, which only checks the top recommendations for the matching rows. It would also require a complex computational effort for all experiments and the results from precision indicate probably only small differences to the comparison on the table level.
This thesis is therefore evaluating precision on the row and table basis and recall only on the table basis.

	$= \frac{\sum_i^n (r_{i,A} - \bar{r}_A) \times (r_{i,B} - \bar{r}_B)}{n \times (\sigma_A \sigma_B)}$ $n = Number\_of\_ideal\_recommendations$ $\sigma_A = Standard\_diviation\_of\_recommendations\_rank$ $\sigma_B = Standard\_diviation\_of\_ideal\_recommendations\_rank$ $r_{i,x} = Rank\_of\_(ideal)\_recommendation$ $\bar{r}_x = Average\_rank\_of\_(ideal)\_recommendation$	identifying if the presented additional data is presented in the correct order. Rank accuracy is measured by spearman correlation.
Coverage	$Coverage = \frac{\#\_of\_presented\_datasets}{\#\_of\_all\_datasets}$	Coverage measures the number of tables recommended at some point out of all potential tables that could be recommended.
Novelty	$Novelty = \frac{\_of\_previously\_unseen\_datasets}{\#\_of\_presented\_datasets}$	Novelty measures the amount of currently presented datasets that have not been previously shown to a user.
Computation Time	$Computation\_time = Approach\_start\_time - Approach\_finish\_time$	Measures the time it takes from the start of a new recommendation request until the systems responds with the recommendations

Table 42: Evaluation measures based on Shani and Gunawardana [208]

Attachment E: Case study process steps

Step	Description
1 Identify contacts	Interviews with 12 people working in this domain were conducted. They were mainly research staff and experts working with the underlying data, helping to develop the system, and conducting research on future systems development. Leading up to the main interviews were a series of interviews with one domain expert, to develop an initial understanding of the users and datasets for this case study. The main interviews then verified and adjusted the initial insights.
2 Problem domain	The problem domain was initially identified during the interviews leading up to the main interviews by analysing the different case study criteria. The domain was further clarified during the main interviews with the various company experts.
3 Problem characteristics	After identifying the domain additional characteristics like the specific users involved, the specific characteristics of the datasets like table columns, content of tables and so on were clarified. The interview process continuously improved these characteristics with various interview loops with the different company's domain experts.
4 Data environment	Based on these characteristics mock-up datasets were generated.
5 Mock-up	Based on the input form the experts a mock-up GUI of the system was developed, containing the main datasets the user was working with in the company.
6 Validate	The dataset and GUI was validated by presenting the mock-up of the mock-up information system to the domain experts.
7 Solve	The solution applied for these problems were the different RecorDa approaches presented in chapter 5 and 6.
8 Feedback	Feedback was gathered iteratively for steps 2 to 7 from different experts within the company. They continuously improved the understanding of the databases and information system, while developing and testing the RecorDa approaches.

Table 43: Case study A: Process steps descriptions

Step	Description
1 Identify contacts	Two initial meetings were used to first describe the approach to an initial contact and a vice president of the company. In another meeting the approach was presented to two technical experts in the company to verify the potential of the solution and discuss the outline of the case study.

2 Problem domain	The electronic catalogue was found to be a good system to consider for the RecorDa approach, because the company was already exploring different options to present more data to the users in this system.
3 Problem characteristics	The specific problem was that there are a series of users, who could need additional data to improve their decision-making process. It was discussed during an interview with the company’s domain experts.
4 Data environment	The case study worked with a couple of datasets from their database and a series of screenshots from the electronic catalogue, which were used in a mock-up information system.
5 Mock-up	To conduct the case study a mock-up system of the electronic catalogue was developed using their data and GUI.
6 Validate	To verify the understanding a prototype was setup and presented to the company’s experts.
7 Solve	The case study was conducted by using the different RecorDa approaches on the mock-up information system.
8 Feedback	In a final presentation this research showed the findings and described the conducted case study to the key contacts in the company to ensure the correct understanding of the particular problem.

Table 44: Case study B: Process steps descriptions