AN INDUSTRIAL DATA RECOMMENDER SYSTEM TO SOLVE THE PROBLEM OF DATA OVERLOAD

Research in Progress

Jess, Torben, University of Cambridge, Cambridge, UK, tj282@eng.cam.ac.uk
Woodall, Philip, University of Cambridge, Cambridge, UK, pw325@eng.cam.ac.uk
Dodwani, Vijay, Indian Institute of Technology, Bombay, India, vijay_dodwani@iitb.ac.in
Harrison, Mark, University of Cambridge, Cambridge, UK, mgh12@eng.cam.ac.uk
McFarlane, Duncan, University of Cambridge, Cambridge, UK, dcm@eng.cam.ac.uk
Nicks, Eric, Boeing, USA, eric.l.nicks@boeing.com
Krechel, William, Boeing, USA, william.e.krechel@boeing.com

Abstract

Getting the right data to the right decision-maker is a significant problem for many industrial companies. One of the main reasons is an overload of data. With the increasing amounts of industrial data this problem is becoming a bigger problem in the future. In order to address this challenge we propose the use of an Industrial Data Recommender System (IDRS). An IDRS recommends additional data to append to the data the decision-maker is currently working with, using techniques from the recommender systems domain like content-based and collaborative filtering. Using industrial cases we found that an IDRS is capable of suggesting useful information to the decision-maker. This additional information should help them to improve their decision-making.

Keywords: Data Recommender Systems, Data Overload, Information Recommender Systems, Recommender Systems, Industrial Data Recommender Systems.

1 Introduction

The amount of industrial data is increasing by around 40% every year (Manyika et al., 2011). Combined with historically grown software architectures and databases, this makes it difficult for the decision-maker to identify useful information. In order to get the most value out of their data, the challenge for industrial companies is to ensure the right data is getting to the right employees. This challenge means addressing the problem of data overload (Eppler and Mengis, 2004), where the large number of datasets make it difficult for the users to be aware of them or even look through them separately.

In order to mitigate this problem we propose an Industrial Data Recommender System (IDRS) and test whether it can recommend other useful data to decision-makers. Recommender systems were successfully used in application areas of information overload such as online shopping or online movie selection (Burke et al., 2011; Park et al., 2012). We investigate the application of IDRS to industrial data overload and consider what modifications, if any, are needed in the future.

As an initial step we use the recommender system to suggest other useful records of data (also known as database rows or tuples) in addition to the records normally presented to the user for their

task/decision. These records may originate from various tables that reside in multiple databases throughout the organisation. However, they can also come from a larger combined integrated system. In this paper we performed an experiment to compare the recommender system against searching for data using automated keyword-based lookup. We assumed two cases for the search: 1.The user knows what tables should be searched (referred to as informed search) and 2. The user has no idea what tables contain useful records (referred to as blind search). After doing and initial evaluation on realistic, fictitious industrial cases, generated with the help of our industrial partners, the results indicate that IDRS outperforms blind search, but performs worse than informed search.

The paper is structured as follows. The second section presents the research background. Section 3 contains a technical description of IDRS. Section 4 then evaluates our system, followed by the conclusion in section 5.

2 Recommender systems and alternatives

Recommender systems are known to address the information overload problem (Jannach et al., 2012),(Porcel et al., 2010) and reduce search effort (Chen et al., 2013). They were used for similar applications such as Knowledge Recommender systems (Zhen et al., 2010) mainly addressing recommendation of documents instead of specific data. Others were applied to internal documents (Jannach et al., 2012) or corporate services (Elsner and Krämer, 2013). Further approaches for recommending datasets to a user in the field of economics (Bahls et al., 2012) or SQL query recommendation to decision-makers (Chatzopoulou et al., 2009) are addressing similar issues of data overload and lack of ability in finding relevant data. However, none of them specifically identified the relevant records within a dataset that would help the decision-maker. Recommender systems typically use collaborative-based filtering and content-based filtering. Collaborative filtering relies on the decision-makers to rank recommended items, while content-based filtering uses the description of the item. Various papers give a further overview about recommender systems (Burke et al., 2011; Park et al., 2012).

The main alternative to recommender systems for information overload is search (Smyth et al., 2011). Search was used successfully for the Internet, one of the biggest data overload problems. Search has the problem that it can take time (ibid) and users only search things they know. Even companies that mainly use search such as Amazon or Google use recommender systems to enhance search (ibid).

Additional alternatives try to improve the process of data selection for example by finding better user interfaces (Ives, 1982) or requirement definition in the design process.

3 Technical approach

An IDRS takes the information (e.g. table and database) of the presented record (called operational record) to generate recommendations (see figure 1, Step 1). It uses the table to ask the recommendation engine for additional tables of interest to the decision-maker (Step 2). The recommendation engine is based on three separate engines.

User recommender system: Identifies additional items by looking for decision-makers with similar rankings for other items. We use the Mahout recommender system library (The Apache Software Foundation, 2014) and Pearson correlation to find similar decision-makers. We then use the decision-makers' other rankings to find additional items.

Item recommender system: Recommends tables (the items in IDRS) to the decision-maker by looking for items that are similar to the currently presented table in a way that they received similar ratings as this item. We used the Mahout library (ibid) for this approach combined with Log likelihood similarity.

Content-based approach: Uses data characterisation (Rahm and Bernstein, 2001) to identify similar records. It takes all columns from a table and generates meta data 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. Tables with columns having a high likelihood of matching are recommended. The benefit of the content-based approach is that it does not require any input from the decision-maker.

The results of these three separate recommender systems are aggregated into a single list of recommended tables (Step 3) using the average of the individual calculated recommendation scores. Then the operational record is used to identify records in the recommended tables by accessing the database (Step 4). Records with an identical join to the operational record are extracted from the system (Step 5) and presented to the decision-maker in descending order of the rating (Step 6). The decision-maker is presented with tables that are relevant to the tables they are working on. However, only the relevant records from this table are shown, significantly reducing the search effort. In the current setting, the decision-maker is initially presented with the first five recommended tables on the side and has the option to click through to additional recommendations. The decision-maker has the opportunity to give ratings for the data, which are then used to further improve the user and item recommender systems and have an influence on the data presented to them.

The initial implementation is kept relatively simple and based on standard recommender systems libraries and kept simple. This is done on purpose to analyse the general applicability of recommender systems towards the data overload problem, which has not been done before. Future work needs to further develop these initial implementations.

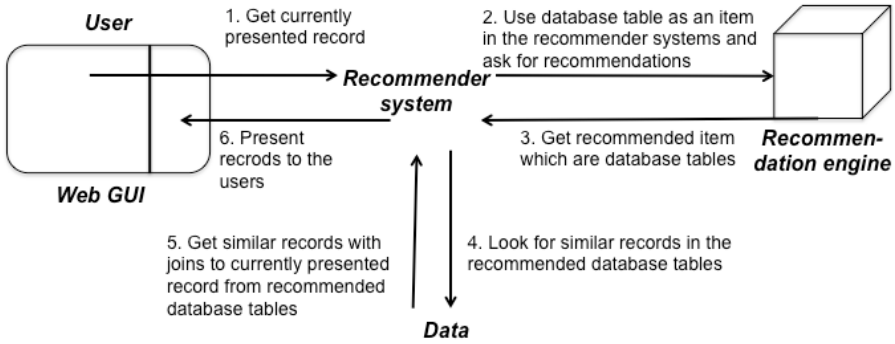

Figure 1. Description of the different process steps for the IDRS.

4 Evaluation

Unlike other application areas of recommender systems like movies no dataset with various users and their recommendations exist for industrial companies. To evaluate IDRS we therefore generated a fictitious industrial data environment with help from our industrial partners. We identified three industrial decisions where additional records, not currently presented to the user, can improve this decision. We used precision and recall for the evaluation. Precision is the number of datasets relevant for the user that were retrieved by the recommender system divided by the total number of retrieved datasets. It shows what percentage of datasets shown by the recommender system was actually relevant to the user. Recall is the number of datasets relevant for the user that were retrieved by the recommender system divided by the total of datasets that would actually be relevant for the user. It shows the percentage of datasets the user wants for their decision-making that the recommender system actually did retrieve.

4.1 Industrial data environment

In the fictitious data environment the decision-makers have to ensure that new parts are procured with the right quality and time for the cheapest possible price. The datasets used can be found in Table 1.

Name of dataset	# of records	Description
Employee list	6467	Contains details like phone numbers, date of birth, job title, etc. about each employee in the company
Procurement	1498	Contains details for the procurement department, that can help in the part ordering process, such as time estimates for time for delivery, previous costs, serial numbers, etc.
Part inventory	1465	Contains the inventory, backorder and # of parts in repair for each part.
Supplier details	1273	Contains a list of suppliers and details about the supplier, such as addresses, zip codes, phone numbers, etc. (Supplier details)
Commodities for Suppliers	1418	Contains a list of commodities a supplier is qualified to deliver (Commodities for suppliers).
Supplier management visits	721	Contains details about visits conducted at each supplier, including details about its capabilities and various ratings
Supplier financial assessment	606	Contains available financial details about a supplier, submitted from the supplier.
Order	3904	Contains a list of orders for parts and delivery times and costs
Order history	69112	Contains a list of previous completed orders
Quality inspections	48813	Contains documentation of part quality inspections conducted at the arrival of past orders
Inspection types	103425	Contains the specific inspections performed in each case (Inspection types).
Support and services parts	1498	Contains details on the Mean-time-between-failure and Mean-time-between-demand for each part
Supplier to parts data	1498	Contains a match of suppliers to parts to clarify which supplier has delivered which part.
Sub-tier questionnaire	6449	Contains the result of a questionnaire from the supplier

Table 1. Datasets used within the industrial data environment used for the test cases

Various parties in the procurement process use and edit these datasets. We identified three types of decision-makers.

Procurement: Asks for proposals from potential suppliers based on orders from other divisions (e.g. Asset Management). Decisions are based on variables like sources strategy, prices, or delivery times.

Supplier management: Determines the supplier sourcing strategy (like single, sole or multiple sourcing). The sourcing strategy bases on supplier management visits, historical performance, and additional criteria.

Asset Manager: Responsible for ensuring enough parts are available as spare parts or for production. They make most of the orders. Decisions are based on inventory, orders, in-repair parts, etc.

4.2 Evaluation cases

The problem for the evaluation of IDRS is lack of existing industrial datasets containing details on recommendations; whereas in other areas such (e.g. movies or books) there are existing datasets (such as the Netflix dataset for the Netflix price for example). We therefore rely on a small set of cases developed with industry and domain experts. The cases in Table 2 describe the decision-maker, the decisions they need to make, the current record the decision-maker is looking at for their decision-making, the additional data that would improve their decision-making (but that they currently cannot access) and the impact this additional data would have on the decision. Table 2 further contain a sample of the current record and the additional data. For this evaluation we are currently assuming that there are no data quality issues and the data is consistent among datasets, e.g. a name or a part description is identical in two separate tables.

4.3 Decision-maker behaviour

Beside the cases, we considered the influence of the decision-maker behaviour. Each of the two approaches (search and IDRS) delivers different outcome based on the decision-maker interaction with the system. For the IDRS the rating behaviour of the user is significant. It impacts the data seen in the future, because decision-maker can influence the data they are seeing by rating. For the search approach the user might be able to notice certain patterns (like specific questions always coming from a specific table) and adjust their search behaviour. We therefore test 3 types of decision-maker rating behaviour.

1) Rates nothing: The decision-maker gives no feedback by rating the recommended data

2) Rates correct subset: The decision-maker only rates the data they like very positively

3) Rates everything: The decision-maker rates all the data that is presented to them (both positively and negatively)

They represent a range of possible decision-maker reactions. For this paper we assume the decision-maker rates 5 if the recommendation is needed, and 1 if it is not needed. Future work needs to evaluate the influence of the different possible ratings. For the search strategy we use two types of behaviour:

1) Blind search: The user always searches through all tables

2) Informed search: The user knows that certain information is in a table and only looks at this table.

For the search we used the CAGE code for case 1, Company name for case 2 and Part No. for case 3 as the keywords. These keywords cover all roles identified and the main join columns of this data management environment, which are often used for searches by decision-makers.

4.4 Empirical results

We evaluated the datasets with the 3 test cases using precision and recall (see Figure 2 for these results). Each of these test cases was used for the three types of rating and the two types of search behaviour. For the context of finding additional data, precision describes the accuracy of the presented or searched additional data. Low precision means the user has a higher effort finding the data. Recall

describes if the additional data was actually found. A high recall indicates that all the helpful additional data was found. These values were measured for the top 5 recommendations. We ran 10 trial batches for each case using records randomly selected from the tables for each case. For each batch we ran 4 trials with the described user rating behaviour until the recommendations did not change further, so that further trials would not change the results found in the last iteration. For search and no-rating recommender behaviour we only ran one trial, because the results obtained by repeating the experiment would not change. See figure 2 for the results.

Decision-maker	**Decision**	**Current record**	**Additional data**	**Impact on decision**
Procurement agent	Does the supplier deliver in time based on past performance?	Supplier details	Historical delivery performance	Decide if order should be sent to this supplier

Sample current data presented to the user:

CAGE	*Comp. name*	*City*	*ZIP*	*State*	*County*	*Phone #*	*Fax #*	*Person of Contact*
F7060	*Brimont SA*	*Rethel*	*8300*	*IL*	*Cook*	*708-343-6837*		*Colleen X. Thomas*

Sample additional data:

Part No.	*CAGE*	*Serial No.*	*Expected delivery day*	*Actual delivery day*
186D1	*F7060*	*11*	*27/6/2013*	*23/6/2013*

Supplier management	Is the supplier still existent in 6 months?	Supplier details	Supplier financial assessment	Decide if orders should be sent to this supplier

Sample current data presented to the user: (see previous decision description)

Sample additional data:

Company name	*CAGE*	*Solvency rating*
Brimont SA	*F7060*	*8*

Asset manager	Should additional parts be ordered?	Part inventory	Procurement	Decide if order for this part should be placed.

Sample current data presented to the user:

Part No	*CAGE*	*On hand quantity*	*Backorder*	*In repair*
811-BATT-1	*26884*	*0*	*1*	*0*

Sample additional data:

Part No.	*CAGE*	*Administrative lead time*	*Production lead time*	*Part repairable*	*Repair turnaround time*	*Costs*
811-BATT-1	*26884*	*17*	*229*	*No*		*11963.06*

Table 2. Details on the data that should be recommended to the decision-maker for the specific task

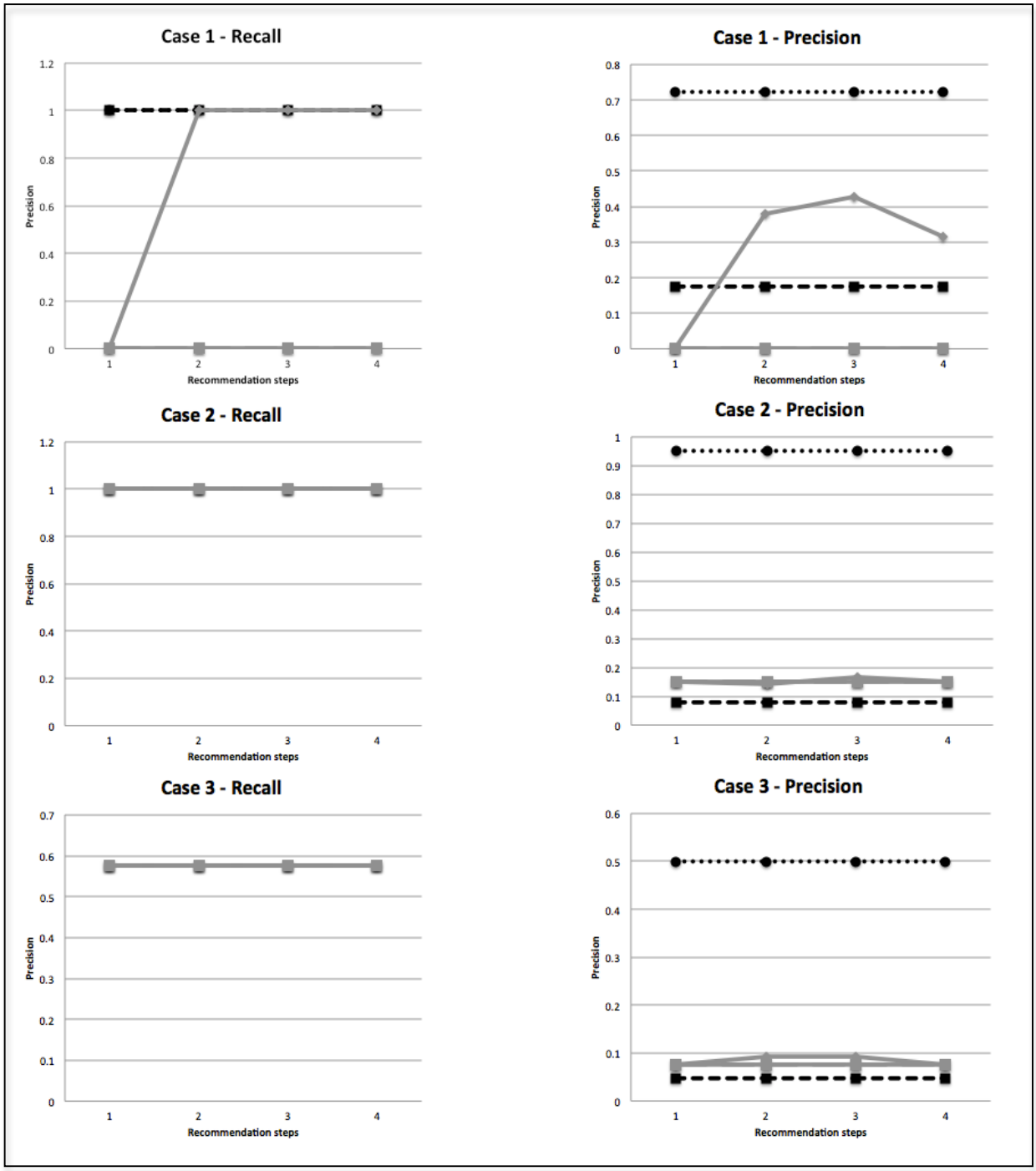

Figure 2. *Describing the average precision and recall values for the different search (black lines) and recommendation approaches (grey lines). Blind search (square markers, stripped lines), Informed search (circle markers, dotted lines), All Rating Recommender (grey, diamond markers), Positive Rating Recommender and No Rating Recommender (grey, square markers)*

For case 2 and 3 we found that the recall of recommender systems is always as good as search. Search will always have a recall of 1 assuming the correct search term is used. For case 1 we found that the performance of the recommender system strongly depends on the rating behaviour of the users

(especially in case 1). The no rating and only positive rating behaviour show identical results. The reason is that if the data is already presented with no rating input then a positive rating would only cause the same information to remain presented to the user. If the data is not presented with the no rating behaviour then the user never gets to rate the data. Therefore the recommender system requires a negative ranking of presented data. Future work needs to address this problem.

The average precision analysis shows a strong benefit of informed search (but not blind search) in comparison to the recommender system. A further breakdown into the separate batches for each of the cases shows that the individual precision varying from 0.02 to 0.2 strongly depends on the specific records. Currently the recommender system cannot reach a precision of 1 because 5 recommendations are present, each with at least 1 record. Given our cases with just 1 record that needs to be identified, the recommender system will always present at least 4 false results. Future work needs to address this problem potentially by eliminating recommendations that received bad ratings. In the cases 1 and 2, IDRS identified the correct information already as the first recommendation. For the all rating behaviour we found that the precision can get slightly worse in some cases with increasing number of ratings from the user. The reason is that the user gives a low rating for a table that contained only 1 record in the recommendation. Due to this recommendation, another dataset is presented to the user, which has more then 1 record, causing precision to be reduced overall due to the higher number of false predictions.

5 Conclusion and future work

This research has developed an Industrial Data Recommender System (IDRS) to address the problem of industrial data overload. IDRS supports decision-makers by providing them with additional useful data (data to which they don't normally have direct access). We took test cases where the decision-maker required additional records, which they are currently not aware of or has to search for. In this situation information overload is a significant problem for decision-makers because finding the relevant data becomes time-consuming as the amount of data increases. IDRS outperforms blind user search but clearly performs worse than informed search with regard to precision. The measures for recall are very similar for all approaches. The performance of the recommender system strongly depends on the rating behaviour of the user. The IDRS in its current state can help to improve the data overload problem, when the user does not know where to search or is not aware of the data.

The implementation outlined in this paper is an initial application of recommender systems in order to determine if they provide benefits for the problem of data overload. More advance recommender systems in the future can improve upon the initial results, especially given that a simple implementation can already create the benefits shown in this paper. They could explore different user and decision types, look into impact of ratings on other users data provision

The initial evaluation was based on industrial experts to identify which data would improve the decision-making. In the future, additional testing which actually measures the decision-making and if it is improving or not should further advance this approach. This would allow comparing our approach with alternatives such as better interface design (Ives, 1982) for example.

References:

Bahls, D., Scherp, G., Tochtermann, K., Hasselbring, W., 2012. Towards a Recommender System for Statistical Research Data, in: Proceedings of the 2nd International Workshop on Semantic Digital Archives. CEUR Workshop Proceedings, Paphos, Cyprus, pp. 61–72.

Borchers, A., Herlocker, J., Konstan, J., Reidl, J., 1998. Ganging up on information overload. Computer 31, 106–108.

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

Burke, R., Felfernig, A., Göker, M.H., 2011. Recommender systems: An overview. AI Mag. 32, 13–18.

Chatzopoulou, G., Eirinaki, M., Polyzotis, N., 2009. Query recommendations for interactive database exploration, in: Proceedings of the 21st International Conference on Scientific and Statistical Database Management. Springer, New Orleans, Louisiana, USA, pp. 3–18.

Chen, L., de Gemmis, M., Felfernig, A., Lops, P., Ricci, F., Semeraro, G., 2013. Human Decision Making and Recommender Systems. ACM Trans. Interact. Intell. Syst. 3, 1–7.

Elsner, H., Krämer, J., 2013. Managing corporate portal usage with recommender systems. Bus. Inf. Syst. Eng. 5, 213–225.

Eppler, M.J., Mengis, J., 2004. The Concept of Information Overload: A Review of Literature from Organization Science, Accounting, Marketing, MIS, and Related Disciplines. Inf. Soc. 20, 325–344.

Ives, B., 1982. Graphical User Interfaces for Business Information Systems. MIS Q. 6, 15–47. doi:10.2307/248990

Jannach, D., Zanker, M., Felfernig, A., Friedrich, G., 2010. Recommender Systems: An Introduction. Cambridge University Press, New York.

Jannach, D., Zanker, M., Ge, M., Gröning, M., 2012. Recommender Systems in Computer Science and Information Systems – A Landscape of Research, in: Lecture Notes in Business Information Processing. Springer.

Manyika, J., Chui, M., Brown, B., Bughin, J., Dobbs, R., Roxburgh, C., Byers, A.H., 2011. Big data: The next frontier for innovation, competition, and productivity (Report). McKinsey Global Institute.

Park, D.H., Kim, H.K., Choi, I.Y., Kim, J.K., 2012. A literature review and classification of recommender systems research. Expert Syst. Appl. 39, 10059–10072.

Porcel, C., del Castillo, J.M., Cobo, M.J., Ruız, A.A., Herrera-Viedma, E., 2010. An improved recommender system to avoid the persistent information overload in a university digital library. Control Cybern. 39, 899–924.

Rahm, E., Bernstein, P.A., 2001. A survey of approaches to automatic schema matching. VLDB J. 10, 334–350.

Smyth, B., Freyne, J., Coyle, M., Briggs, P., 2011. Recommendation as collaboration in web search. AI Mag. 32, 35–45.

The Apache Software Foundation, 2014. Mahout library [WWW Document]. Apache Mahout Scalable Mach. Learn. Data Min. URL http://mahout.apache.org/ (accessed 3.24.15).

Zhao, Y., Tang, L.C.M., Darlington, M.J., Austin, S.A., Culley, S.J., 2008. High value information in engineering organisations. Int. J. Inf. Manag. 28, 246–258.

Zhen, L., Huang, G.Q., Jiang, Z., 2010. An inner-enterprise knowledge recommender system. Expert Syst. Appl. 37, 1703–1712.