Proceedings of the ACM RecSys 2010 Workshop on User-Centric Evaluation of Recommender Systems and Their Interfaces (UCERSTI),
Barcelona, Spain, Sep 30, 2010
Published by CEUR-WS.org, ISSN 1613-0073, online ceur-ws.org/Vol-612/paper6.pdf

A Novel Multidimensional Framework for Evaluating Recommender Systems

Artus Krohn-Grimberghe
Information Systems and Machine Learning Lab
University of Hildesheim, Germany
artus@ismll.de

Alexandros Nanopoulos
Information Systems and Machine Learning Lab
University of Hildesheim, Germany
nanopoulos@ismll.de

Lars Schmidt-Thieme
Information Systems and Machine Learning Lab
University of Hildesheim, Germany
schmidt-thieme@ismll.de

ABSTRACT

The popularity of recommender systems has led to a large variety of their application. This, however, makes their evaluation a challenging problem, because different and often contrasting criteria are established, such as accuracy, robustness, and scalability. In related research, usually only condensed numeric scores such as RMSE or AUC or F-measure are used for evaluation of an algorithm on a given data set. It is obvious that these scores are insufficient to measure user satisfaction.

Focussing on the requirements of business and research users, this work proposes a novel, extensible framework for the evaluation of recommender systems. In order to ease user-driven analysis we have chosen a multidimensional approach. The research framework advocates interactive visual analysis, which allows easy refining and reshaping of queries. Integrated actions such as drill-down or slice/dice, enable the user to assess the performance of recommendations in terms of business criteria such as increase in revenue, accuracy, prediction error, coverage and more.

The ability of the proposed framework to comprise an effective way for evaluating recommender systems in a business-user-centric way is shown by experimental results using a research prototype.

Keywords

Recommender Systems, Recommendation, Multidimensional Analysis, OLAP, Exploratory Data Analysis, Performance Analysis, Data Warehouse

1. INTRODUCTION

The popularity of recommender systems has resulted in a large variety of their applications, ranging from presenting personalized web-search results over identifying preferred multimedia content (movies, songs) to discovering friends in social networking sites. This broad range of applications, however, makes the evaluation of recommender systems a challenging problem. The reason is the different and often contrasting criteria that are being involved in real-world applications of recommender systems, such as their accuracy, robustness, and scalability.

The vast majority of related research usually evaluates recommender system algorithms with condensed numeric scores: root mean square error (RMSE) or mean absolute error (MAE) for rating prediction, or measures usually stemming from information retrieval such as precision/recall or F-measure for item prediction. Evidently, although such measures can indicate the performance of algorithms regarding some perspectives of recommender systems' applications, they are insufficient to cover the whole spectrum of aspects involved in most real-world applications. As an alternative approach towards characterizing user experience as a whole, several studies employ user-based evaluations. These studies, though, are usually rather costly, difficult in design and implementation.

More importantly, when recommender systems are deployed in real-world applications, notably e-commerce, their evaluation should be done by business analysts and not necessarily by recommender-system researchers. Thus, the evaluation should be flexible on testing recommender algorithms according to business analysts' needs using interactive queries and parameters. What is, therefore, required is to provide support for evaluation of recommender systems' performance based on popular online analytical processing (OLAP) operations. Combined with support for visual analysis, actions such as drill-down or slice/dice, allow assessment of the performance of recommendations in terms of business objectives. For instance, business analysts may want to examine various performance measures at different levels (e.g., hierarchies in categories of recommended products), detect trends in time (e.g., elevation of average product rating following a change in the user interface), or segment the customers and identify the recommendation quality with respect to each customer group. Furthermore, the interactive and visual nature of this process allows easy adaptation of the queries according to insights already gained.

In this paper, we propose a novel approach to the evaluation of recommender systems. Based on the aforementioned motivation factors, the proposed methodology builds on multidimensional analysis, allowing the consideration of various aspects important for judging the quality of a recommender system in terms of real-world applications. We describe a way for designing and developing the proposed extensible multidimensional framework, and provide insights into its applications. This enables integration, combination and comparison of both, the presented and additional, measures (metrics).

To assess the benefits of the proposed framework, we have implemented a research prototype and now present experimental results that demonstrate its effectiveness.

Our main contributions are summarized as follows:

- A flexible multidimensional framework for evaluating recommender systems.

Proceedings of the ACM RecSys 2010 Workshop on User-Centric Evaluation of Recommender Systems and Their Interfaces (UCERSTI),
Barcelona, Spain, Sep 30, 2010
Published by CEUR-WS.org, ISSN 1613-0073, online ceur-ws.org/Vol-612/paper6.pdf

- A comprehensive procedure for efficient development of the framework in order to support analysis of both, dataset facets and algorithms' performance using interactive OLAP queries (e.g., drill-down, slice, dice).
- The consideration of an extended set of evaluation measures, compared to standards such as the RMSE.
- Experimental results with intuitive outcomes based on swift visual analysis.

2. RELATED WORK

For general analysis of recommender systems, Breese [5] and Herlocker et al. [11] provide a comprehensive overview of evaluation measures with the aim of establishing comparability between recommender algorithms. Nowadays, the generally employed measures within the prevailing recommender tasks are MAE, (R)MSE, precision, recall, and F-measure. In addition further measures including confidence, coverage and diversity related measures are discussed but not yet broadly used. Especially the latter two have attracted attention over the last years as it is still not certain whether today's predictive accuracy or precision and recall related measures correlate directly with interestingness for a system's end users. As such various authors proposed and argued for new evaluation measures [22, 21, 6]. Ziegler [22] has analyzed the effect of diversity with respect to user satisfaction and introduced topic diversification and intra-list similarity as concepts for the recommender system community. Zhang and Hurley [21] have improved the intra-list similarity and suggested several solution strategies to the diversity problem. Celma and Herrera [6] have addressed the closely related novelty problem and propose several technical measures for coverage and similarity of item recommendation lists. All these important contributions focus on reporting single aggregate numbers per dataset and algorithm. While our framework can deliver those, too, it goes beyond that by its capability of combining the available measures and, most importantly, dissecting them among one or more dimensions.

Analysis of the end users' response to recommendations and their responses' correlation with the error measures used in research belongs to the field of Human-Recommender Interaction. It is best explored by user studies and large scale experiments, but both are very expensive to obtain and thus rarely conducted and rather small in scale. Select studies are [13, 14, 4]. Though in the context of classical information retrieval, Joachims et al [13] have conducted a highly relevant study on the biasing effect of the position an item has within a ranked list. In the context of implicit feedback vs. explicit feedback Jones et al [14] have conducted an important experiment on the preferences of users concerning recommendations generated by unobtrusively collected implicit feedback compared to recommendations based on explicitly stated preferences. Bollen et al. [4] have researched the effect of recommendation list length in combination with recommendation quality on perceived choice satisfaction. They found that for high quality recommendations, longer lists tend to overburden the user with difficult choice decisions. Against the background of those results we believe that for initial research on a dataset, forming an idea, checking if certain effects are present, working on collected data with a framework like the one presented is an acceptable proxy. With findings gained in this process, conducting meaningful user studies is an obvious next step.

Recent interesting findings with respect to dataset characteristics are e.g. the results obtained during the Netflix challenge [3, 17] on user and item base- effects and time-effects in data. When modeled appropriately, they have a noteworthy effect on recommender performance. The long time it took to observe these properties of the dataset might be an indicator for the fact that with currently available tools proper analysis of the data at hand is more difficult and tedious than it should be. This motivates the creation of easy-to-use tools enabling thorough analysis of the datasets and the recommender algorithm's results and presenting results in an easy to consume way for the respective analysts.

Notable work regarding the integration of OLAP and recommender systems stems from the research of Adomavicius et al. [2, 1]. They treat the recommender problem setting with its common dimensions of users, items, and rating as inherently multidimensional. But unlike this work, they focus on the multidimensionality of the generation of recommendations and on the recommenders themselves being multidimensional entities that can be queried like OLAP cubes (with a specifically derived query language, RQL). In contrast, our work acknowledges the multidimensional nature of recommender systems, but focusses on their multidimensional evaluation.

Existing frameworks for recommender systems analysis usually focus on the automatic selection of one recommendation technique over another. E.g., [10] is focussed on an API that allows retrieval and derivation of user satisfaction with respect to the recommenders employed. The AWESOME system by Thor and Rahm [20], the closest approach to that presented here, shares the data warehousing approach, the description of the necessary data preparation (ETL), and the insight of breaking down the measures used for recommender performance analysis by appropriate categories. But contrary to the approach presented here, the AWESOME framework is solely focussed on website performance and relies on static SQL-generated reports and decision criteria. Furthermore, it incorporates no multidimensional approach and does not aim at simplifying end-user-centric analysis or interactive analysis at all.

3. FRAMEWORK REQUIREMENTS

3.1 The Role of a Multidimensional Model

Business analysts expect all data of a recommender systems (information about items, generated recommendations, user preferences, etc.) to be organized around business entities in form of dimensions and measures based on a multidimensional model. A multidimensional model enforces structure upon data and expresses relationships between data elements [19]. Such a model, thus, allows business analysts to investigate all aspects of their recommender system by using the popular OLAP technology [7]. This technology provides powerful analytical capabilities that business analysts can query to detect trends, patterns and anomalies within the modeled measures of recommender systems' performance across all involved dimensions.

Multidimensional modeling provides comprehensibility for the business analysts by organizing entities and attributes of their recommender systems in a parent-child relationship (1:N in databases terminology), into dimensions that are

Proceedings of the ACM RecSys 2010 Workshop on User-Centric Evaluation of Recommender Systems and Their Interfaces (UCERSTI),
Barcelona, Spain, Sep 30, 2010
Published by CEUR-WS.org, ISSN 1613-0073, online ceur-ws.org/Vol-612/paper6.pdf

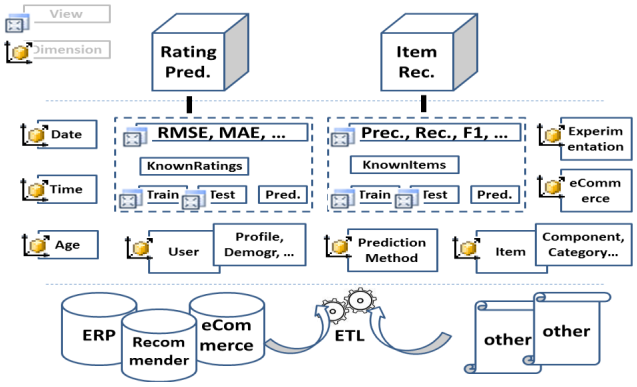

Figure 1: The recommender evaluation framework. The dimensions specified are connected with both fact table groups (dashed boxes in the center) and are thus available in both resulting cubes. End users can connect to the Rating Prediction and Item Recommendation cubes.

identified by a set of attributes. For instance, the dimension of recommended items may have as attributes the name of the product, its type, its brand and category, etc. For the business analyst, the attributes of a dimension represent a specific business view on the facts (or key performance indicators), which are derived from the intersection entities. The attributes of a dimension can be organized in a hierarchical way. For the example of a dimension about the user of the recommender systems, such a hierarchy can result from the geographic location of the user (e.g., address, city, or country). In a multidimensional model, the measures (sometimes called facts) are based in the center with the dimensions surrounding them, which forms the so called star schema that can be easily recognized by the business analysts. The star schema of the proposed framework will be analyzed in the following section.

It is important to notice that aggregated scores, such as the RMSE, are naturally supported. Nevertheless, the power of a multidimensional model resides in adding further derived measures and the capability of breaking all measures down along the dimensions defined in a very intuitive and highly automated way.

3.2 Core Features

Organizing recommender data in a principled way provides automation and tool support. The presented framework enables analysis of all common recommender datasets. It supports both rating prediction and item recommendation scenarios. Besides that, data from other application sources can and should be integrated for enriched analysis capabilities. Notable sources are ERP systems, eCommerce systems and experimentation platform systems employing recommender systems. Their integration leverages analysis of the recommender data by the information available within the application (e.g., recommender performance given the respective website layouts) and also analysis of the application data by recommender information (e.g., revenue by recommender algorithm).

Compared to RMSE, MAE, precision, recall, and F-measure, more information can be obtained with this framework as, first, additional measures e.g. for coverage, novelty, diversity analysis are easily integrated and thus available for all datasets. Second, all measures are enhanced by the respective ranks, (running) differences, (running) percentages, totals, standard deviations and more.

While a single numerical score assigned to each recommender algorithm's predictions is crucial for determining winners in challenges or when choosing which algorithm to deploy [8], from an business insight point of view a lot of interesting information is forgone this way. Relationships between aspects of the data and their influence on the measure may be hidden. One such may be deteriorating increase in algorithmic performance with respect to an increasing number of rating available per item, another the development of the average rating over the lifetime of an item in the product catalog. A key capability of this framework is exposing intuitive ways for analyzing the above measures by other measures or related dimensions.

From a usability point of view, this framework contributes convenient visual analysis empowering drag-drop analysis and interactive behavior. Furthermore, convenient visual presentation of the obtained results is integrated from the start as any standard conforming client can handle it. Manual querying is still possible as is extending the capabilities of the framework with custom measures, dimensions, or functions and post-processing of received results in other applications. Inspection of the original source data is possible via custom actions which allow the retrieval of the source rows that produced the respective result. Last but not least, aggregations allow for very fast analysis of very large datasets, compared to other tools.

The following section elaborates on the architecture of the multidimensional model that is used by the proposed framework, by providing its dimensions and measures.

4. THE ARCHITECTURE OF THE MULTIDIMENSIONAL FRAMEWORK

Figure 1 gives an overview of the architecture of the framework. The source data and the extract-transform-load (ETL) process cleaning it and moving it into the data store are located at the bottom of the framework. The middle tier stores the collected information in a data warehouse manner regarding facts (dashed boxes in the center) and dimensions (surrounding the facts). The multidimensional cubes (for rating recommendation and item prediction) sitting on top of the data store provide access to an extended set of measures (derived from the facts in the warehouse) that allow automatic navigation along their dimensions and interaction with other measures.

4.1 The Data Flow

The data gathered for analysis can be roughly divided into two categories:

***Core data*:** consisting of the algorithms' training data, such as past ratings, purchase transaction information, online click streams, audio listening data, ... and the persisted algorithms' predictions.

***Increase-insight data*:** can be used as a means to leverage the analytic power of the framework. It consists roughly of user master data, item master data, user transactional statistics, and item transactional statistics. This data basically captures the metadata and

Proceedings of the ACM RecSys 2010 Workshop on User-Centric Evaluation of Recommender Systems and Their Interfaces (UCERSTI),
Barcelona, Spain, Sep 30, 2010
Published by CEUR-WS.org, ISSN 1613-0073, online ceur-ws.org/Vol-612/paper6.pdf

usage statistics data not directly employed by current recommender algorithms (such as demographic data, geographic data, customer performance data...).

In case of recommender algorithms employed in production environments, relational databases housing the transactional system (maybe driving an e-commerce system like an ERP system or an online shop) will store rich business master data such as item and user demographic information, lifetime information and more, next to rating information, purchase information, and algorithm predictions. In case of scientific applications, different text files containing e.g. rating information, implicit feedback, and the respective user and item attributes for training and the algorithms' predictions are the traditional source of the data.

From the respective source, the master data, the transactional data, and the algorithm predictions are cleaned, transformed, and subsequently imported into a data warehouse. Referential integrity between the elements is maintained, so that e.g. ratings to items not existing in the system are impossible. Incongruent data is spotted during insert into the recommender warehouse and presented to the data expert.

Inside the framework, the data is logically split into two categories: measures (facts) that form the numeric information for analysis, and dimensions that form the axes of analysis for the related measures. In the framework schema (figure 1), the measures are stylized within the dashed boxes. The dimensions surrounding them and are connected to both, the rating prediction and the item recommendation measures.

4.2 The Measures

Both groups of measures analyzed by the framework—the measures for item recommendation algorithms and the measures for rating prediction algorithms—can be divided into basic statistical and information retrieval measures.

***Statistical measures*:** Among the basic statistical measures are counts and distinct counts, ranks, (running) differences and (running) percentages of various totals for each dimension table, train ratings, test ratings and predicted ratings; furthermore, averages and their standard deviations for the lifetime analysis, train ratings, test ratings, and predicted ratings.

***Information retrieval measures*:** Among the information retrieval measures are the popular MAE and (R)MSE for rating prediction, plus user-wise and item-wise aggregated precision, recall and F-measure for item prediction. Novelty, diversity, and coverage measures are also included as they provide additional insight. Furthermore, for comparative analysis, the differences in the measures between any two chosen (groups of) prediction methods are supported as additional measures.

In case a recommender system and thus this framework is accompanied by a commercial or scientific application, this application usually will have measures of its own. These measures can easily be integrated into the analysis. An example may be an eCommerce application adding sales measures such as gross revenue to the framework. These external measures can interact with the measures and the dimension of the framework.[1]

[1]E.g., the revenue could be split up by year and recommendation method, showing the business impact of a recommender.

4.3 The Dimensions

The dimensions are used for slicing and dicing the selected measures and for drilling down from global aggregates to fine granular values. For our framework, the dimensions depicted in figure 1 are:

***Date*:** The Date dimension is one of the core dimensions for temporal analysis. It consists of standard members such as Year, Quarter, Month, Week, Day and the respective hierarchies made up from those members. Furthermore, Year-to-date (YTD) and Quarter/Month/Week/Day of Year logic provides options such as searching for a Christmas or Academy Awards related effect.

***Time*:** The Time dimension offers Hour of Day and Minute of Day/Hour analysis. For international datasets this dimension profits from data being normalized to the time zone of the creator (meaning the user giving the rating).

***Age*:** The Age dimension is used for item and user lifetime analysis. Age refers to the relative age of the user or item at the time the rating is given/received or an item from a recommendation list is put into a shopping basket and allows for analysis of trends in relative time (c.f. section 6).

***User*:** User and the related dimensions such as UserProfile and UserDemographics allow for analysis by user master data and by using dynamically derived information such as activity related attributes. This enables grouping of the users and content generated by them (purchase histories, ratings) by information such as # of ratings or purchases, # of days of activity, gender, geography...

***Item*:** Item and the related dimensions such as ItemCategory and ItemComponent parallel the user-dimensions. In a movie dataset, the item components could be, e.g., actors, directors, and other credits.

***Prediction Method*:** The Prediction Method dimension allows the OLAP user to investigate the effects of the various classes and types or recommender systems and their respective parameters. Hierarchies, such as Recommender Class, Recommender Type, Recommender Parameters, simplify the navigation of the data.

***eCommerce*:** As recommender algorithms usually accompany a commercial or scientific application (e.g., eCommerce) having dimensions of its own, these dimensions can easily be integrated into and be used by our framework.

***Experimentation*:** In case this framework is used in an experiment-driven scenario [8], such as an online or marketing setting, Experimentation related dimensions should be used. They parallel the PredictionMethod dimension, but are more specific to their usage scenario.

Proceedings of the ACM RecSys 2010 Workshop on User-Centric Evaluation of Recommender Systems and Their Interfaces (UCERSTI),
Barcelona, Spain, Sep 30, 2010
Published by CEUR-WS.org, ISSN 1613-0073, online ceur-ws.org/Vol-612/paper6.pdf

5. PROTOTYPE DESCRIPTION

This section describes the implementation of a research prototype for the proposed framework. The prototype was implemented using Microsoft SQL Server 2008 [18] and was used later for our performance evaluation.

In our evaluation, the prototype considers the Movielens 1m dataset [9], which is a common benchmark for recommender systems. It consists of 6.040 users, 3.883 items, and 1.000.209 ratings received over roughly three years. Each user has at least 20 ratings and the metadata supplied for the users is userId, gender, age bucket, occupation, and zipcode. Metadata for the item is movieId, title and genre information.

Following a classical data warehouse approach [15, 12], the database tables are divided into dimension and fact tables. The dimension tables generally consist of two kinds of information: static master data and dynamic metadata. The static master data usually originates from an ERP system or another authoritative source and contains e.g. naming information. The dynamic metadata is derived information interesting for evaluation purposes, such as numbers of ratings given or time spent on the system. To allow for an always up to date and rich information at the same time, we follow the approach of using base tables for dimension master data and views for dynamic metadata derived through various calculations. Further views then expose the combined information as pseudo table. The tables used in the warehouse of the prototype are Date, Time, Genre (instantiation of Category), Item, ItemGenre (table needed for mapping items and genres), Numbers (a helper table), Occupation, PredictedRatings, PredictedItems, PredictionMethod, TestRatings, TestItems, TrainRatings, TrainItems, and User. The Item and User table are in fact views over the master data provided with the Movielens dataset and dynamic information gathered from usage data. Further views are SquareError, UserwiseFMeasure, AllRatings, and AgeAnalysis.

On top of the warehouse prototype, an OLAP cube for rating prediction was created using Microsoft SQL Server Analysis Services. Within this cube, the respective measures were created: counts and sums, and further derived measures such as distinct counts, averages, standard deviations, ranks, (running) differences and (running) percentage. The core measures RMSE and MAE are derived from the error between predicted and actual ratings. The most important OLAP task with respect to framework development is to define the relationships between the measures and dimensions, as several dimensions are linked multiple times (e.g. the Age dimension is role-playing as it is linked against both item age and user age) or only indirect relationships exist (such as between category and rating the relationship is only established via item). Designing the relationships has to be exercised very carefully, as both correctness of the model and the ability to programmatically navigate dimensions and measures (adding them on the report axes, measure field or as filters) depend on this step. Linking members enables generic dimensions such as Prediction Method A, and Prediction Method B, that can be linked to chosen dimension members. This renders unnecessary the creation of the $n(n-1)/2$ possible measures yielding differences between any two prediction methods A and B (for, say, RMSE or F-measure). Furthermore, this approach allows choosing more than one dimension member, e.g. several runs of one algorithm with different parameters, as one linked member for aggregate analysis.

Before we go on to the evaluation of our prototype, let us state that our framework describes more than simply a model for designing evaluation frameworks. The prototype serves well as a template for other recommender datasets, too. With nothing changed besides the data load procedure, it can be used directly for, e.g., the other Movielens datasets, the Netflix challenge dataset or the Eachmovie dataset. Additional data available in those datasets (e.g. the tagging information from the Movielens 10m dataset) are either ignored or require an extension of the data warehouse and the multidimensional model (resulting in new analysis possibilities).

6. PERFORMANCE EVALUATION

In the previous section we have described the implementation of a research prototype of the proposed framework using the Movielens 1m dataset. Building on this prototype, we proceed with presenting a set of results that are obtained by applying it.

We have to clarify that the objective of our experimental evaluation is not limited to the comparison of specific recommender algorithms, as it is mostly performed in works that propose such algorithms. Our focus is, instead, on demonstrating the flexibility and easiness with which we can answer important questions for the performance of recommendations. It is generally agreed that explicitly modelling the effects describing changes in the rating behavior over the various users (user base-effect), items (item base-effect), and age of the respective item or user (time effects) [3, 16, 17]. For this reason, we choose to demonstrate the benefits of the proposed framework by setting our scope on those effects followed by exemplary dissecting the performance of two widely examined classes of recommender algorithms, i.e., collaborative filtering and matrix factorization. We also consider important the exploratory analysis of items and users, which can provide valuable insights for business analysts about factors determining the performance of their recommender systems. We believe that the results presented in the following demonstrate how easy it is to obtain them by using the proposed framework, which favors its usage in real-world applications, but also can provide valuable conclusions to motivate the usage of the framework for pure research purpose, since it allows for observing and analyzing the performance by combining all related dimensions that are being modeled.

All results presented in the remainder of this section could easily be obtained graphically by navigating the presented measures and dimensions using Excel 2007 as multidimensional client.

6.1 Exploratory Data Analysis

Using the framework, the first step for a research and a business analytics approach is exploring the data. As an example, the Calendar dimension (Date) is used to slice the average rating measure. Figure 2 presents this as pivot chart. The sharp slumps noticeable in March and August 2002 together with a general lack of smoothness beyond mid 2001 arouse curiosity and suggest replacing average rating by rating count (figure not shown). Changing from counts to running percentages proves that about 50 percent of the ratings in this dataset are spent within the first six months

Proceedings of the ACM RecSys 2010 Workshop on User-Centric Evaluation of Recommender Systems and Their Interfaces (UCERSTI),
Barcelona, Spain, Sep 30, 2010
Published by CEUR-WS.org, ISSN 1613-0073, online ceur-ws.org/Vol-612/paper6.pdf

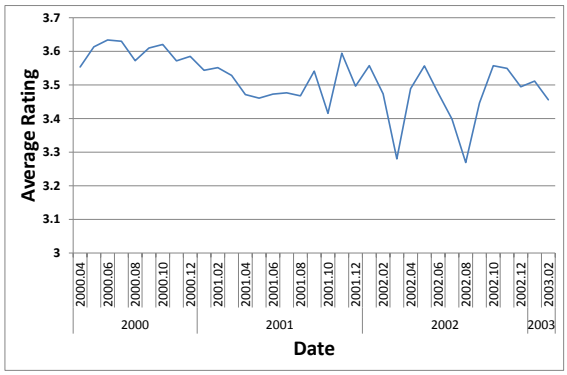

Figure 2: Average rating by date

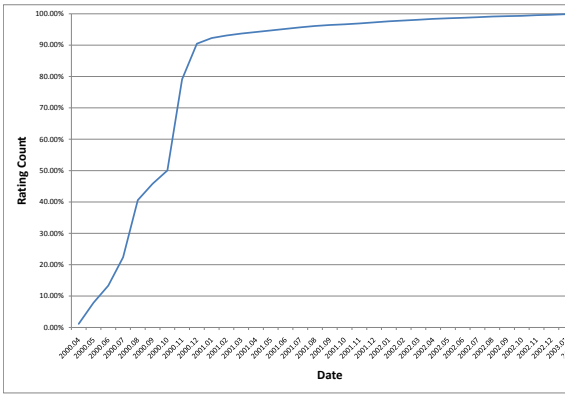

Figure 3: Rating count by date (running percentages)

out of nearly three years. Within two more months 90 percent of the ratings are assigned, roughly seven percent of the data for 50 percent of the time (figure 3).

6.1.1 *Item Analysis*

The framework allows an easy visualization of the item effect described e.g. in [16], namely that there usually is a systematic variation of the average rating per item. Aditionally, other factors can easily be integrated in such an analysis. Figure 4 shows the number of ratings received per item sorted by decreasing average rating. This underlines the need for regularization when using averages, as the movies rated highest only received a vanishing number of ratings.

Moving on the x-axis from single items to rating count buckets containing a roughly equal number of items, a trend of heavier rated items being rated higher can be observed (figure omitted for space reasons). A possible explanation

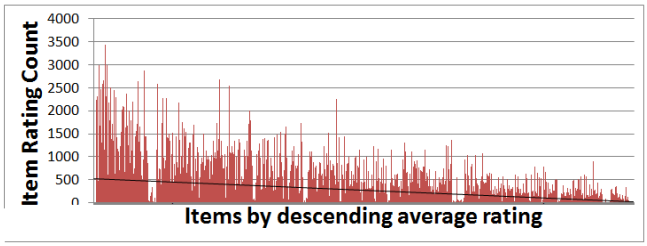

Figure 4: Item rating count sorted by decreasing average rating

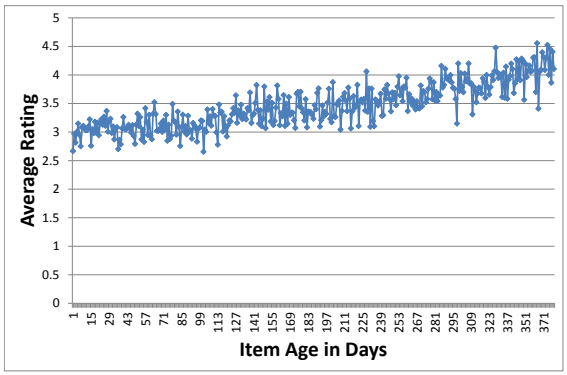

Figure 5: The all-time classics effect. Ratings tend to increase with the age of the movie at the time the rating is received. Age is measured in time since the first rating recorded.

might be that blockbuster movies accumulate a huge number of generally positive ratings during a short time and the all-time classics earn a slow but steady share of additional coverage. That all-time classics receive higher ratings can nicely be proved with the framework, too. Consistent with findings during the final phase of the Netflix competition by Koren [17], figure 5 shows a justification for the good results obtained by adding time-variant base effects to recommender algorithms. Besides the all-time classics effect, the blockbuster effect can also be observed (figure 6), showing that items who receive numerous ratings per day on average also have a higher rating.

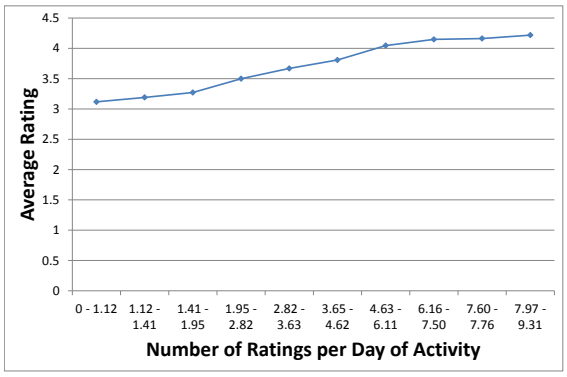

Figure 6: The blockbuster effect. Increasing average item rating with increasing number of ratings received per day.

Slicing the average rating by Genre shows a variation among the different genre with Film-Noir being rated best (average rating 4.07, 1.83% of ratings received), and Horror being rated worst (3.21, 7,64%). Of the Genres with at least ten percent of the ratings received Drama scores highest (3.76, 34.45%) and Sci-Fi lowest (3.46, 15.73%). Figure not shown.

6.1.2 *User Analysis*

The user effect can be analyzed just as easy as the item effect. Reproducing the analysis explained above on the users,

Proceedings of the ACM RecSys 2010 Workshop on User-Centric Evaluation of Recommender Systems and Their Interfaces (UCERSTI),
Barcelona, Spain, Sep 30, 2010
Published by CEUR-WS.org, ISSN 1613-0073, online ceur-ws.org/Vol-612/paper6.pdf

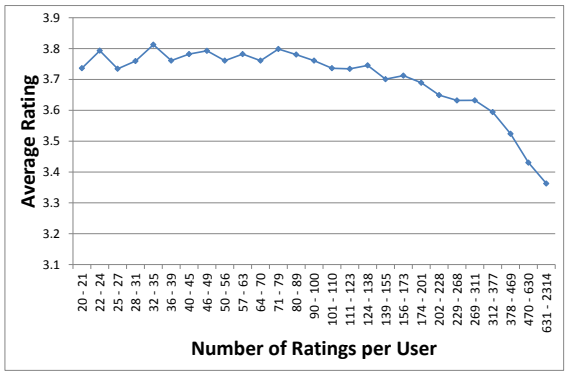

Figure 7: The effect of the number of ratings per user on the average rating

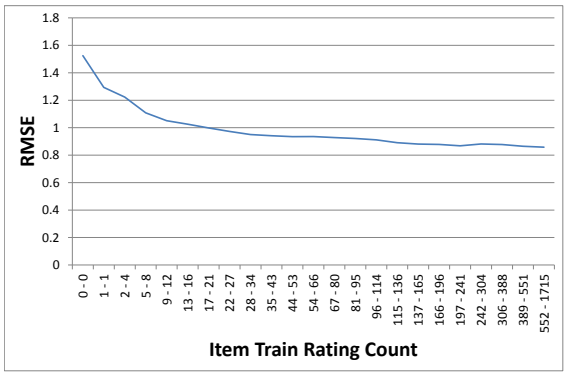

Figure 8: Item rating count effect on a factor model. Buckets created on roughly equal item count.

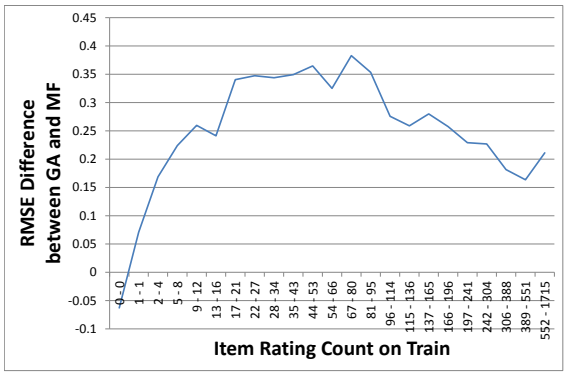

Figure 9: Difference in RMSE between Matrix Factorization (MF) and Global Average (GA) vs. ratings available per item on the train dataset.

it is interesting to notice that for heavy raters the user rating count effect is inverse to the item rating count effect described above (figure 7): the higher the amount of ratings spent by a given user, the lower his or her average rating. One explanation to this behavior might be that real heavy raters encounter a lot of rather trashy or at least low quality movies.

6.2 Recommender Model Diagnostics

For algorithm performance comparison, the Movielens 1m ratings were randomly split into two nearly equal size partitions, one for training (500103), and one for testing (500104 ratings). Algorithm parameter estimation was conducted on the training samples only, predictions were conducted solely on the test partition. Exemplarily, a vanilla matrix factorization (20 features, regularization 0.09, learn rate 0.01, 56 iterations, hyperparameters optimized by 5-fold cross-validation) is analyzed.[2]

For a researcher the general aim will be to improve the overall RMSE or F-Measure, depending on the task, as this is usually what wins a challenge or raises the bar on a given dataset. For a business analyst this is not necessarily the case. A business user might be interested in breaking down the algorithm's RMSE over categories or top items or top users as this may be relevant information from a monetary aspect. The results of the respective queries may well lead to one algorithm being replaced by another on a certain part of the dataset (e.g. subset of the product hierarchy).

In figure 8, RMSE is plotted vs. item rating count in train.This indicates that more ratings on an item do help factor models. Interpreted the other way around, for a business user, this implies that this matrix factorization yields best performance on the items most crucial to him from a top sales point of view (though for slow seller other algorithms might be more helpful).

The same trend can be spotted when RMSE is analyzed by user rating count on the training set (figure omitted for space reasons), though the shape of the curve follows a straighter line than for the item train rating count (where it follows more an exponential decay).

Due to the approach taken in the design of the OLAP cube the number of recommender algorithms comparable as A and B is not limited; neither does it have to be exactly one algorithm being compared with exactly one other, as multiple selection is possible. Furthermore—given the predictions are already in the warehouse—replacing one method by another or grouping several methods as A or B can nicely be achieved by selecting them in the appropriate drop-down list. Exemplarily, the matrix factorization analyzed above is compared to the global average of ratings as baseline recommendation method. Figure 9 reveals that for this factor model more ratings on train do increase the relative performance, as expected, up to a point from which the static baseline method will gain back roughly half the lost ground. Investigation of this issue might be interesting for future recommender models.

All results presented could be obtained very fast: when judging the time needed to design query and report (chart)—which was on average seconds for construction of the query and making the chart look nice—, and when judging execution time—which was in the sub-second timeframe.

[2]The matrix factorization yielded an RMSE of 0.8831 given the presented train-test split.

7. CONCLUSIONS

We have proposed a novel multidimensional framework for integrating OLAP with the challenging task of evaluating recommender systems. We have presented the architecture of the framework as a template and described the implementation of a research prototype. Consistent with the other papers at this workshop, the authors of this work

Proceedings of the ACM RecSys 2010 Workshop on User-Centric Evaluation of Recommender Systems and Their Interfaces (UCERSTI),
Barcelona, Spain, Sep 30, 2010
Published by CEUR-WS.org, ISSN 1613-0073, online ceur-ws.org/Vol-612/paper6.pdf

believe that the perceived value of a system largely depends on its user interface. Thus, this work provides an easy to use framework supporting visual analysis. Our evaluation demonstrates, too, some of the elegance of obtaining observations with the proposed framework. Besides showing the validity of findings during the recent Netflix prize on another dataset, we could provide new insights, too. With respect to the recommender performance evaluation and the validity of RMSE as an evaluation metric, it would be interesting to see if a significant difference in RMSE concerning the amount of ratings present in the training set would also lead to significant effects in a related user study.

In our future work, we will consider the extension of our research prototype and develop a web-based implementation that will promote its usage.

8. ACKNOWLEDGMENTS

The authors gratefully acknowledge the co-funding of their work through the European Commission FP7 project MyMedia (grant agreement no. 215006) and through the European Regional Development Fund project LEFOS (grant agreement no. 80028934).

9. REFERENCES

[1] G. Adomavicius, R. Sankaranarayanan, S. Sen, and A. Tuzhilin. Incorporating contextual information in recommender systems using a multidimensional approach. *ACM Trans. Inf. Syst.*, 23(1):103–145, 2005.

[2] G. Adomavicius and A. Tuzhilin. Multidimensional recommender systems: A data warehousing approach. In *WELCOM '01: Proceedings of the Second International Workshop on Electronic Commerce*, pages 180–192, London, UK, 2001. Springer-Verlag.

[3] R. Bell, Y. Koren, and C. Volinsky. Modeling relationships at multiple scales to improve accuracy of large recommender systems. In *KDD '07: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 95–104, New York, NY, USA, 2007. ACM.

[4] D. Bollen, B. P. Knijnenburg, M. C. Willemsen, and M. Graus. Understanding choice overload in recommender systems. In *RecSys '10: Proceedings of the 2010 ACM conference on Recommender systems*, New York, NY, USA, 2010. ACM.

[5] J. S. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *MSR-TR-98-12*, pages 43–52. Morgan Kaufmann, 1998.

[6] O. Celma and P. Herrera. A new approach to evaluating novel recommendations. In *RecSys '08: Proceedings of the 2008 ACM conference on Recommender systems*, pages 179–186, New York, NY, USA, 2008. ACM.

[7] E. Codd, S. Codd, and C. Salley. Providing OLAP to user-analysts: An it mandate. Ann Arbor,MI, 1993.

[8] T. Crook, B. Frasca, R. Kohavi, and R. Longbotham. Seven pitfalls to avoid when running controlled experiments on the web. In *KDD '09: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1105–1114, New York, NY, USA, 2009. ACM.

[9] GroupLens. Movielens data sets. http://www.grouplens.org/node/73.

[10] C. Hayes, P. Massa, P. Avesani, and P. Cunningham. An on-line evaluation framework for recommender systems. In *In Workshop on Personalization and Recommendation in E-Commerce (Malaga*. Springer Verlag, 2002.

[11] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.*, 22(1):5–53, 2004.

[12] W. H. Inmon. *Building the Data Warehouse*. Wiley, 4th ed., 2005.

[13] T. Joachims, L. Granka, B. Pan, H. Hembrooke, and G. Gay. Accurately interpreting clickthrough data as implicit feedback. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 154–161, New York, NY, USA, 2005. ACM.

[14] N. Jones, P. Pu, and L. Chen. How users perceive and appraise personalized recommendations. In *UMAP '09: Proceedings of the 17th International Conference on User Modeling, Adaptation, and Personalization*, pages 461–466, Berlin, Heidelberg, 2009. Springer-Verlag.

[15] R. Kimball and M. Ross. *The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling*. Wiley, 2nd ed., 2002.

[16] Y. Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *KDD '08: Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 426–434, New York, NY, USA, 2008. ACM.

[17] Y. Koren. Collaborative filtering with temporal dynamics. In *KDD '09: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 447–456, New York, NY, USA, 2009. ACM.

[18] Microsoft. Microsoft SQL Server 2008 homepage. http://www.microsoft.com/sqlserver/2008/.

[19] J. O'Brien and G. Marakas. *Management Information Systems*. McGraw-Hill/Irwin, 9th ed., 2009.

[20] A. Thor and E. Rahm. Awesome: a data warehouse-based system for adaptive website recommendations. In *VLDB '04: Proceedings of the Thirtieth international conference on Very large data bases*, pages 384–395. VLDB Endowment, 2004.

[21] M. Zhang and N. Hurley. Avoiding monotony: improving the diversity of recommendation lists. In *RecSys '08: Proceedings of the 2008 ACM conference on Recommender systems*, pages 123–130, New York, NY, USA, 2008. ACM Press.

[22] C.-N. Ziegler, S. M. McNee, J. A. Konstan, and G. Lausen. Improving recommendation lists through topic diversification. In *WWW '05: Proceedings of the 14th international conference on World Wide Web*, pages 22–32, New York, NY, USA, 2005. ACM.