Model Driven Approach to Risk Aware Web Recommender System

Contents:

What is Big Data and Big Data Analytics?

Big data usually includes data sets with sizes beyond the ability of commonly used software tools to capture, curate, manage, and process data within a tolerable elapsed time. Big data "size" is a constantly moving target, as of 2012 ranging from a few dozen terabytes to many petabytes of data.

Big Data Analytics involves a **new set of approaches** for analysing data sets that were not previously accessible because they posed challenges across one or more of the “3 V’s” of Big Data:

- **Volume** - too Big – Terabytes and more of Credit Card Transactions, Web Usage data, System logs
- **Variety** - too Complex – truly unstructured data such as Social Media, Customer Reviews, Call Centre Records
- **Velocity** - too Fast - Sensor data, live web traffic, Mobile Phone usage, GPS Data

Operations on Big Data:

- Aggregation and Statistics
  - Data warehouse and OLAP
- Indexing, Searching, and Querying
  - Keyword based search
  - Pattern matching (XML/RDF)
- Knowledge discovery
  - Data Mining
  - Statistical Modeling

Big Data for Effective Decision Making :

Effective Decision making helps to utilise the available resources (information) for achieving the objectives. Big Data has created three distinct types of data driven products for effective decision making:

Benchmarking

Benchmarking is often the first quick win when embarking into the world of big data. It helps us to find solutions to the problems like: Why is A performing better then B? Why did the curve drop?

Predictions

Predictions provides solution to the problems by evaluating patterns from the past. Who will perform better in the future, A or B ?

Recommendation and Filter systems

The fame of data products is driven by something else: Recommendation Engines. Recommendations narrow what could become a complex decision to just a few recommendations. Big Data allowed us to do recommendations on a new scale that we did not see before. The most well-known example is how the Google search algorithm trumped Altavista by recommending the best websites to view. Another well-known example is the recommendation from Amazon based on the reading behaviour from other readers. Both of those systems are based on algorithms that “learn” from past data.

Model-driven Engineering

Model-driven engineering (MDE) is a software development methodology which focuses on creating and exploiting domain models, which are conceptual models of all the topics related to a specific problem. Hence it highlights and aims at abstract representations of the knowledge and activities that govern a particular application domain, rather than the computing (f.e. algorithmic) concepts.

Domain Specific Modelling and Domain Specific Language

Domain-specific modeling is a software engineering methodology for designing and developing systems, such as computer software. It involves systematic use of a domain-specific language to represent the various facets of a system.

Domain-specific modeling languages tend to support higher-level abstractions than general-purpose modeling languages, so they require less effort and fewer low-level details to specify a given system.

Model-driven Big Data Analytics

Data analytics is getting more and more important in all businesses. If you want to have brilliant results of high quality, then you need data analyst experts. Such an analyst must be an expert at the data domain and at the technical domain at the same time. And this is hard to find. The MDBDA Tool Suite will help you to close the gap.

- **Graphical**
  The Tool Suite allows you to draw your data analytics instead of writing them programmatically. As the proverb says, a picture is worth a thousand words, MDBDA increases the learning curve dramatically and improves the documentation.

- **Model-driven**
  The model-driven approach separates the analytics logic from the technical big data platform. This allows you to change platform components later, without rewriting your analytics.

Recommender Systems

Recommendation systems (RS) help to match users with the items of possible interest to the user.

- **Collaborative filtering**

Collaborative filtering methods are based on collecting and analyzing a large amount of information on users' behaviours, activities or preferences and predicting what users will like based on their similarity to other users. Algorithms such as the k-nearest neighbour (k-NN) approach and the Pearson Correlation are used.

- **Content-based filtering**

Recommendations are based on information on the content of items rather than on other users' opinion. Uses a machine learning algorithm to induce a profile of the users preferences from examples based on a feature description of content.

- **Hybrid Recommender Systems**

It combines collaborative filtering and content-based filtering.

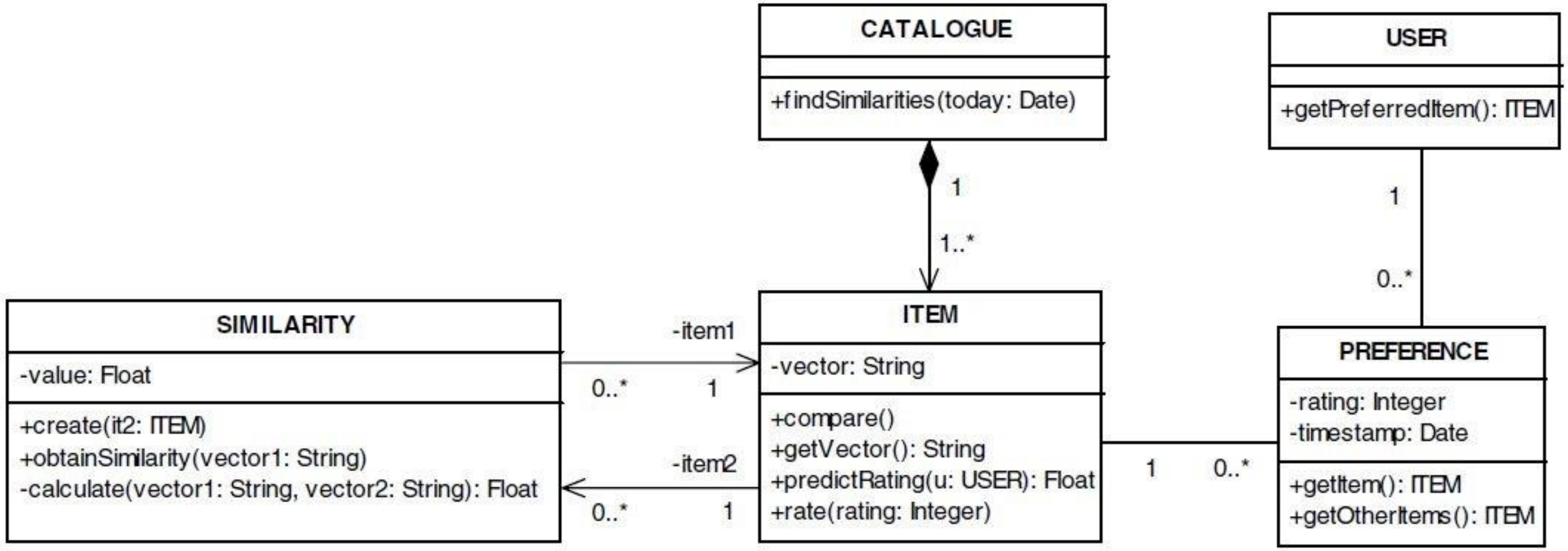
Hybridization techniques:

- Weighted: The score of different recommendation components are combined numerically.
- Switching: The system chooses among recommendation components and applies the selected one.
- Mixed: Recommendations from different recommenders are presented together.
- Feature Combination: Features derived from different knowledge sources are combined together and given to a single recommendation algorithm.

Model-Driven Approach to Web Recommender systems

Recommendation techniques have been increasingly incorporated in e-commerce applications, supporting clients in identifying those items that best fit their needs. Unfortunately, little effort has been made

to integrate these techniques into methodological proposals of Web development, discouraging the adoption of engineering approaches to face the complexity of recommender systems.



Class diagram of a generic recommender system

Risk Aware Recommendation System

In many applications, such as recommending personalized content, it is also important to consider the risk of upsetting the user so as not to push recommendations in certain circumstances, for instance, during a professional meeting, early morning, late-night. Therefore, the performance of the recommender system depends in part on the degree to which it has incorporated the risk into the recommendation process.

Method

In response to these challenges, a dynamic risk sensitive recommendation system called **DRARS (Dynamic Risk-Aware Recommender System**), which models the context-aware recommendation as a bandit problem. This system combines a content-based technique and a contextual bandit algorithm. They have shown that DRARS improves the Upper Confidence Bound (UCB) policy, the currently available best algorithm, by calculating the most optimal exploration value to maintain a trade-off between exploration and exploitation based on the risk level of the current user's situation. The authors conducted experiments in an industrial context with real data and real users and have shown that taking into account the risk level of users' situations significantly increased the performance of the recommender systems.

Risk Aware Web Recommendation System (RAWRS)

The system should provide more relevant information to users in risky and dynamic environments.

There is a need to consider various requirements for an accurate risk aware web recommender system, considering the dynamicity of the content and the situation risk level. This model of a recommender system aims at proposing more relevant information to the user with the help of user specific data such as web logs, user data, dwell time, search requests and click through rate (CTR). By establishing the relationship between the user specific data and the content of the website, this system evaluates the risk associated with the recommendation results. It further establishes that tuning the recommendation results with the corresponding risk factor enhances the user experience.

Applications of RAWRS

- **Health Product Recommendation**

Recommending products by keeping in mind the risk associated.

- **Portfolio Recommendation**

Such recommendations includes specific index funds and ETFs in which to invest, portfolio risk/return statistics, a summary of your risk profile, and other helpful tools and tips.

Model Driven Approach to RAWRS

In order to model RAWRS we need to understand its process.

Inputs:

User model (ratings, preferences, demographics, situational context)

Logs (website navigation data, dwell time, search requests and click through rate (CTR))

Items (with or without description of item characteristics)

Outputs:

Relevance score. Used for ranking.

Risk Score

Outcome:

Recommend items that are assumed to be relevant

Constraints:

Remember that relevance might be context-dependent

Characteristics of the list itself might be important (diversity)