TV Predictor: Personalized Program Recommendations to be displayed on SmartTVs

Christopher Krauss
Fraunhofer Institute FOKUS
Kaiserin-Augusta-Allee 31
Berlin, Germany
christopher.krauss@fokus.fraunhofer.de

Lars George
Humboldt-Universität zu Berlin
Unter den Linden 6
Berlin, Germany
lars.george@hu-berlin.de

Stefan Arbanowski
Fraunhofer Institute FOKUS
Kaiserin-Augusta-Allee 31
Berlin, Germany
stefan.arbanowski@fokus.fraunhofer.de

ABSTRACT

Switching through the variety of available TV channels to find the most acceptable program at the current time can be very time-consuming. Especially at the prime time when there are lots of different channels offering quality content it is hard to find the best fitting channel.

This paper introduces the TV Predictor, a new application that allows for obtaining personalized program recommendations without leaving the lean back position in front of the TV. Technically the usage of common Standards and Specifications, such as HbbTV, OIPF and W3C, leverage the convergence of broadband and broadcast media. Hints and details can overlay the broadcasting signal and so the user gets predictions in appropriate situations, for instance the most suitable movies playing tonight. Additionally the TV Predictor Autopilot enables the TV set to automatically change the currently viewed channel. A Second Screen Application mirrors the TV screen or displays additional content on tablet PCs and Smartphones.

Based on the customers viewing behavior and explicit given ratings the server side application predicts what the viewer is going to favor. Different data mining approaches are combined in order to calculate the users preferences: Content Based Filtering algorithms for similar items, Collaborative Filtering algorithms for rating predictions, Clustering for increasing the performance, Association Rules for analyzing item relations and Support Vector Machines for the identification of behavior patterns. A ten fold cross validation shows an accuracy in prediction of about 80%.

TV specialized User Interfaces, user generated feedback data and calculated algorithm results, such as Association Rules, are analyzed to underline the characteristics of such a TV based application.

Keywords

Hybrid TV, SmartTV, recommendation, algorithms, content-based Filtering, collaborative Filtering, offline, online.

BigMine'13 , August 11 - 14 2013, Chicago, IL, USA

1. INTRODUCTION

With the rapid growth of available content via the internet the users are not able to consume all offered products at once - nor in a life time. So they have to find a considered selection of media items they want to consume. A Recommendation System helps its users to find a pre selected list of items they might be interested in. Therefore these engines use a set of different approaches to get a prediction of the users' interests.

In a closed system like a web platform there exists a fixed amount of items and a group of users is trying to find the best fitting item within this platform. In general an item can be every media a user is searching for. This can be for instance a text document, a picture, an audio file, an on demand video file or a live TV program.

The term SmartTV (also called hybrid TV) is a made-up word in allusion to the term SmartPhone and has no official definition. Today TV sets are called hybrid TVs, when they perform more than just showing moving pictures or teletext. For instance they are able to render websites overlaying the regular TV program as they are connected to the internet. The Hybrid Broadcast Broadband Television Standard (HbbTV) allows for enriching the regular linear TV signal with an Application-URL and so users can open specific websites by pressing the red button on their remote controls. As broadcasters are only able to enrich their own signals, this CE-HTML based website is called broadcaster depended App.

The Fraunhofer Institute for Open Communication Systems (FOKUS) has developed a system, called TV Predictor, that uses the benefits of both technologies, SmartTVs and Recommendation Engines, and so allows for overlaying the linear TV program with personalized recommendations and rating predictions. It consists of a typical client-server architecture in which the thin client is only used for rending the User Interface and delegating the user inputs to the server side recommendation engine. The Java-based server is able to calculate program recommendations for different situations depending on the users request.

The Personalized Program Guide (PPG) is a feature of this recommendation system allowing to create a "personal channel". It is not an actual TV channel, but an aggregation of the best fitting parts of all available media items. As HbbTV allows for changing to specific channels, the HbbTV frontend can automatically change the channel for the user without further user input. This results in a lean back situation for the user where the TV automatically shows the best fitting program at any time.

In order to explain the data mining aspects of such a recommendation system, this paper is structured into 6 sections. The second section introduces basics of hybrid TV sets and related recommendation projects. The third section will focus on the use cases, architecture and main components of the TV Predictor. Afterwards the main recommendation aspects as well as the theory becomes determined, followed by an evaluation. A summary and future research will conclude this paper.

2. STATE OF THE ART

In this section some basics for a better comprehension of this paper will be elucidated. Thereby this paper will refer to standards of the consumer electronics and different approaches for recommender systems.

2.1 Hybrid TV

The hybrid TV is the result of the increasing convergence between television and internet. The TV is not necessarily connected to the world wide web directly. It can also be upgraded by incorporating so called set top boxes, additional devices as middleware.

The internet connection of hybrid television is often only used to deliver additional contents to the device and not the whole live program. The live program in contrast (called linear TV) is delivered using the Digital Video Broadcasting (DVB) standard. In general this DVB signal is still delivered by satellite, cable or in a terrestrial way. SmartTV applications must be accepted by the broadcaster. So they can guarantee that the web application is optimized for interpretation on a television set and only shows appropriate content. Therefore the application URL has to be entered in the Application Information Table (AIT) of the DVB stream. This is an additional digital container to deliver applications. This signal containing the TV program and the application will be received, decrypted and interpreted by the corresponding connected device. So a consumer is able to use the application functions directly on his TV set.

In order to fulfill all requirements for displaying and interacting with web pages on TV screens, a specification was published in June 2010 by The European Telecommunications Standards Institute (ETSI), called Hybrid Broadcast Broadband Television. The HbbTV specification is based on different standards of the Open IPTV Forum (OIPF), the Consumer Electronics Association (CEA) and the DVB-Project (DVB). Consumer Electronics – HTML (CE-HTML) is a TV web technology based on current W3C standards, like XHTML 1.0, CSS TV Profile 1.0 and JavaScript. In addition this technology allows for using native TV functions, such as changing the TV channel, by using OIPF components.

2.2 Recommendation Systems

Today there are a lot of Web 2.0 services providing recommendation engines. In Germany only few of the leading services are popular or known. First of all the big players like Facebook, Youtube, Amazon and MySpace are often in use. New emerging SVoD and TVoD services, such as Maxdome, Lovefilm and Watchever, didn't play a big role in Germany when starting this project in November 2011. Today they try to fill this gap – even on SmartTVs, using HbbTV or native applications – but with mostly Content Based Recommendation Engines predicting similar videos. So there was a lack of personalized Recommendation Systems for TV programs to be displayed on SmartTVs in Germany.

Other services focus on the US market, for instance Hulu and Netflix. Moreover the US company Rovi demonstrated a white label solution in beta state for TV program guides (cf. [6]) to be shown on TV sets. Academic recommendation services such as queveo.tv (cf. [2]) or the Content-boosted Collaborative Filtering (cf. [16]) have introduced possible solutions to recommend TV related content, but only in a regular web 2.0 environment.

It seemed that a Recommendation System predicting TV programs directly on the target device, that is easy to use, based on the user behaviour and automatic feedback was needed in Germany.

3. TV PREDICTOR SOLUTION

The TV Predictor is a German cooperation project between Arvato RTV (content provider and subsidiary company of Bertelsmann) and the Fraunhofer FOKUS. This prototype was introduced at the IFA Consumer Electronics in September 2012 and was designed to recommend programs on SmartTVs and connected Second Screen devices. Since December 2012 a productive system was established to be used for free in the regular desktop web environment on the website:
`http:\\www.rtv.de`

The recommendation system uses a set of different criteria to make recommendations which correspond to the users' viewing behavior. When users watch TV Predictor enabled channels, they can open the recommendation menu by pressing the according button on their remote control. A set of the best and most relevant programs for the current user will be shown. These personalized recommendations are based on the automatically tracked viewing behavior and explicitly defined program ratings of the registered user or - in case they did not sign up - they will get averaged or well-selected recommendations.

In order to generate the best and most accurate recommendations, the recommendation system combines the best fitting algorithms in a hybrid way. The usage of these algorithms depends on the user's request:

- Find similar programs to the selected one by using common content-based filtering algorithms, such as the Cosine Similarity, and by using unsupervised learning algorithms, such as Association Rules
- Get program highlights for a specific time period based on the favorite programs of similar users (Pearson Correlation Coefficient) and predictions of program ratings (Slope One)
- Calculate a personalized program guide changing the channel automatically by using Clustering to pre-select programs best fitting the user's interests and rating predictions
- Overlay upcoming program recommendations while watching TV based on recognized behavior patterns (calculated by a Support Vector Machine) to find user interests, such as genres and categories, favored actors, directors and producers or even the preferred channels, weekdays or times to watch specific content

3.1 TV specialized User Interface

The User Interface on a SmartTV differs from other devices (such as PCs, tablets and Smartphones) in a lot of features. The according criteria are focussing on the navigation with a remote control and the display restrictions for older devices and distant viewers. It is based on CE-HTML, CSS 2 and JavaScript.

The TV Predictor Menu will open when users press the red button on their remote control. As shown in figure 1 there is a main navigation on the top of the screen. The part below is used to display the contents of the current section – in this case the program highlights of the current day.

Figure 1: TV Predictor Menu

The top 7 programs are displayed in the content area. A circle indicates the users affection for the shown program. This is visualized by a number in the middle of the circle in the range of [1, 10] representing the prediction value, where 1 is the lowest and 10 is the best value. A dark blue circle points to an already rated program and a light blue circle shows a real predicted rating.

The content menu opens when the users select a specific recommendation. On the following screen users can get detailed information about the program contents, rate them, see the recommendation value and where applicable get an explanation of the predicted rating. Moreover they can ask for similar items calculated on demand.

The colored keys of the remote control allow to log into the application (green button), to connect the TV set with a second screen device (yellow button) and to hide the application and watching the linear program again (red button).

Basically the TV Predictor was designed to be shown on SmartTVs, but when users want to watch the linear program again, they can push the contents to a tablet PC or a Smartphone and control the same User Interface by touching on the corresponding screen element. Alternatively the application can be mirrored on the TV set and the second screen device in order to provide a smart remote control application.

3.2 Software Architecture

According to figure 2, the client consists of four different front end modules: the Consumer Electronics User Interface, the Second Screen Interface, the statistics front end allowing for analyzing user and usage behaviour, as well as the admin console that allows the administrator to set up and adjust the server system. The server offers 18 different REST-APIs. The interfaces are classified into the 5 main groups: SubmitClientInput to provide automatic and manually inputs, GetRecommendations to retrieve recommendations, DoCronJobs to activate operations in continuous time intervals, EngineSettings to adjust the engines behavior and EngineStatistics to offer graphical analysis. Each Interface is a Java Servlet. The return type depends on the requested data and can be XML or CSV formated data.

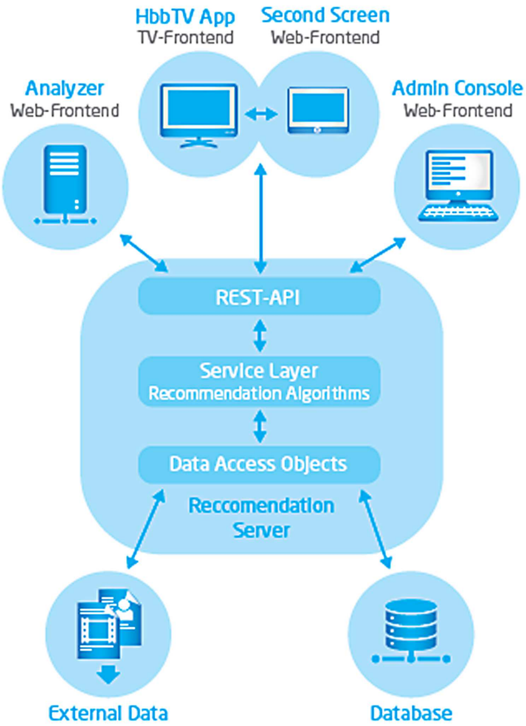

Figure 2: TV Predictor - Architecture

The server infrastructure is hosted in the Amazon cloud using a load balancer, scalable engine instances and distributed database nodes. The hardware costs are comparatively low as one virtual machine (Amazon EC2 server: c1.medium) can handle up to 400 users at the same time. Another tiny instance (Amazon RDS class: db.m1.small) is needed for the Amazon Relational Database Service. In order to allow a synchronization of the TV set and the second screen device, node.js as a middleware server infrastructure was introduced.

3.3 Domain Model

The core of the recommendation engine focuses on program recommendations. So the core of the domain model focuses on it as well. Figure 3 shows an Entity Relationship Model (ERM) containing Users, Programs, their relation and meta data of a Program. A Program on television, such as a series episode, has a lot of meta data. It does not matter if it is a moving picture, a series episode, news or a documentation, a program can be identified by its content. It may belong to one or more Categories and Genres. Each is encapsulated by an association (ProgramHasCategory and ProgramHasGenre) as it is the rule for ERMs. Of course there are a lot of people involved in the production of a program, such as the author, the director or the actors.

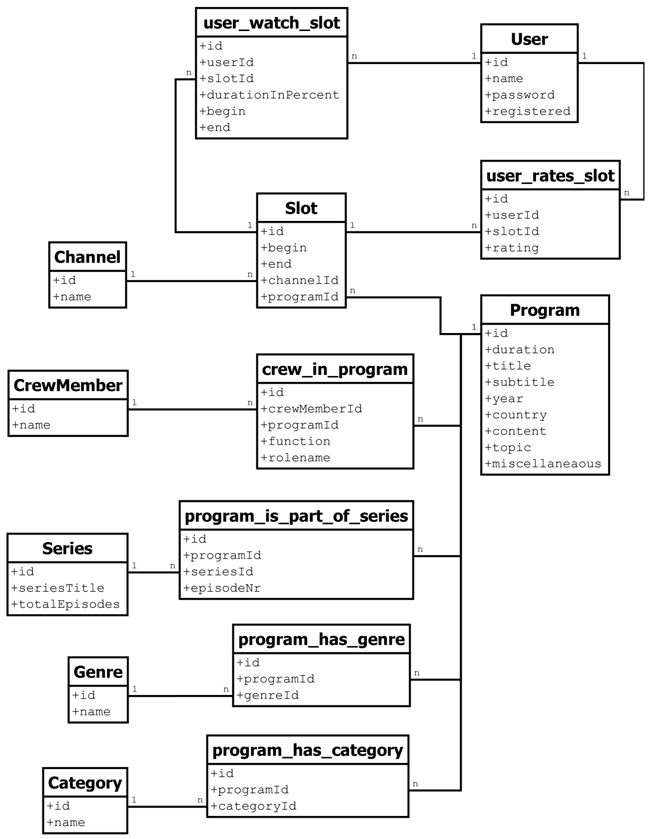

Figure 3: SmartTV Predictor – ERM of the user-program relationship

Those people are defined as CrewMembers.

Besides a Program may belong to a Series. That is not only the case if it is a regular TV series, but also if a movie is part of a multi-parted TV event, like a trilogy, or it is repeated like daily news. A Program can be broadcasted on multiple Channels at multiple times. This construction is defined as Slot representing a time interval on a specific Channel. So a Program can be shown on different Slots, for instance a specific episode of a series is broadcasted between 10:00 p.m. and 10:30 p.m. on one channel and between 3:00 p.m. and 3:30 p.m. on another channel.

Primarily a User does not give feedback to a Program, but to a Slot. That is the direct relation to the content when users watch TV. What kind of Program a User is interested in will be analyzed by Filtering algorithms afterwards thus preparing additional relations between users and situations.

4. RECOMMENDATION ALGORITHMS

The TV Predictor Recommendation Engine has the goal to filter the most relevant items from the set of all items. Top-N Filtering means that the resulting collection of elements does not contain the whole data set, but only the first N elements of an ordered list. So only the Top-7 programs (e.g. with highest predicted rating) are finally offered to a user.

4.1 Content Based Filtering

In the TV Predictor Content Based Filtering is used to find similar items. Therefore an element is just compared to another element by respecting their content information on meta data. So an item for instance can be very similar or dissimilar to another item. Therefore each attribute has to be explored and compared to the according attributes of the other elements. The Cosine-based Similarity is used to calculate the similarity of two elements by treating them as vectors:

$$cSim(e_1, e_2) = cos(\vec{E}_1, \vec{E}_2) = \frac{\vec{E}_1 \cdot \vec{E}_2}{|\vec{E}_1| \times |\vec{E}_2|} \quad (1)$$

e_1 and e_2 are the elements to be compared like items or users. $\vec{E}_1$ and $\vec{E}_2$ are vectors representing all features of this element, so when n is the number of all attributes of an element, then $\vec{E}_1 = (a_{1,e_1}, a_{2,e_1}, a_{3,e_1}, ..., a_{n,e_1})$. (cf. [4, p. 619], [18, p. 929])

Content Based Filtering is used to find the best fitting item to the current, for instance other videos or even advertisements similar to the watched content. But feedbacks like user-item relations are not directly provided by these algorithms. Approaches made on the basis of user-item relations are called Collaborative Filtering.

4.2 Collaborative Filtering

Collaborative Filtering (short: CF) is the most common approach for Web 2.0 technologies. The simple comparison of elements is extended by data on consumer behaviour. So the recommendation engine is able to predict items by characteristics of other users.

In CF the user is related to items – here by a rating value. It is possible and more widely spread that not each user rates each item. Actually some items may be never rated by a user and some users may never provide ratings at all. This issue is caused by the sparsity problem (cf. [24, p. 579]) and the cold start problem (cf. [7, p. 238]).

In order to calculate recommendations the TV Predictor must know about the users interests. Therefore the engine differs between two different feedback types (see figure 3):

- Automatically tracked watch behaviour: The client sends in regular intervals messages indicating the watched channel, so the server can lookup for the current playing program in the database.
- Manually given ratings: The user can explicitly provide ratings for single programs in range of $[1, 10]$.

4.2.1 Slope One

The Slope One algorithm (Item-based Filtering) calculates the Top-N items by taking into account the ratings of all other users. It is divided into two parts. (cf. [20, p. 153]) First of all there is a method to get the average deviation of two items:

$$dev(i_1, i_2) = \frac{\sum_{u \in U_{i_1 i_2}} (r_{u,i_1} - r_{u,i_2})}{|U_{i_1 i_2}|} \quad (2)$$

Where i_1 and i_2 are the items, r_{u,i_1} is the items rating user u gave. $U_{i_1 i_2}$ is the set of users who rated both items and $|U_{i_1 i_2}|$ is its cardinality. So the result is the ratings deviation of an item. If the average rating of item i_1 is higher than the average rating of item i_2 the value is positive, if it is lower the value is negative, or if the average ratings are equal the value is $dev(i_1, i_2) = 0$.

The prediction value $pre(u,j)$ for user u and item j is defined as

$$pre(u,j) = \frac{\sum_{i \in Ij}(r_{u,i} - dev(i,j))}{|Ij|} \tag{3}$$

Ij is the set of all relevant items to be compared with item j and $|Ij|$ is its cardinality. The higher $|U_{i_1 i_2}|$, the better the prediction. $r_{u,i}$ is the rating of user u for item i. (cf. [13, p. 3]) The resulting value is the predicted rating of this user u for the current item j.

The result will be limited to the specific rating interval, just in case the predicted rating is not in that range. This might happen when the predicted rating is not in the range of [1,10], for instance when the current user always rated the items worse than the average user (e.g. a standard deviation of 3.5) and the requested item is rated extremely bad (e.g. 2.1). The predicted rating will be -1.4, so it must be mapped to the original range – in this case to 1.

4.2.2 Pearson Correlation Coefficient

In contrast to Item-based Filtering, User-based Top-N Recommendation Filtering focuses on finding similar users – called neighbours. So the objective of Neighbourhood-based Collaborative Filtering is to find the nearest neighbour. (cf. [5, p. 550]) Afterwards the Top-N items of the nearest neighbours are predicted.

The Pearson correlation coefficient is used to calculate the similarity of users by considering the items both users rated.

$$pSim(u_1,u_2) = \frac{\sum_{i \in I_{u_1 u_2}}(r_{i,u_1} - \overline{r}_{u_1})(r_{i,u_2} - \overline{r}_{u_2})}{\sqrt{\sum_{i \in I_{u_1 u_2}}(r_{i,u_1} - \overline{r}_{u_1})^2 (r_{i,u_2} - \overline{r}_{u_2})^2}} \tag{4}$$

r_{i,u_1} is the rating of user u_1 for the item i of item set $I_{u_1 u_2}$ (a set with existing ratings of both users for each item). $\overline{r}_{u_2}$ is the average rating of all the items of user u_2. (cf. [1, p. 738], [20, p. 153], [4, p. 619])

4.3 Model-based Algorithms

The most valuable recommendation approach is a combination of Memory-based (online) and Model-based (offline) algorithms. Memory-based Filtering predicts items depending on the given feedback data set, such as the two introduced CF approaches do. So the recommendation is calculated on the fly by using just the current database values and no retrospectively transformed data sets. On the fly calculations are also called online calculations. The according algorithm uses only the available set of this data and no transformations of it. (cf. [12, p. 49]) Model-based Filtering algorithms mostly perform offline calculations, which is often an expensive operation. (cf. [12, p. 49]) Only a few tasks are performed on demand. The goal is to prepare the data sets in a special way - for instance by finding patterns of user behaviour or finding whole neighbourhoods - so just a minimum of tasks has to be done online.

Calculations can become time consuming when too many users and too many items are available. The concrete numbers depend on the size of the user-item-matrix, the used algorithm and the available hardware-resources. In order to increase the performance, offline algorithms first segment users or items considering their ratings.

4.3.1 Clustering

Cluster Analysis is also called Data Segmentation and has the goal to divide a set into subsets. It aims at finding groups of elements that are similar in one or more criteria. (cf. [9, p. 454]) Amazon for instance uses a not further named greedy cluster generation, which starts with a set of randomly chosen users and searches for their nearest neighbours. Some of their algorithms classify the users into multiple clusters depending on the users' behaviours. (cf. [14, p. 77])

The TV Predictor uses K-Means (cf. [11, p. 675]), a partitional Clustering algorithm, in order to group users by respecting their likings on different attributes, such as preferred genre, category and channel. Partitional means that the resulting clusters consists of an predefined amount of 15 subsets.

4.3.2 Association Rules

Association Rules have the goal to find highly represented relations (so called transactions) between a user or even a user's attribute and the items or their attributes. (cf. [14, p. 441], [15, p. 500]) The resulting set of frequent items may, in addition, be scanned for some rules and afterwards the list of transaction can be divided in causations and consequence. For instance when a defined number of users have watched the same programs, the resulting frequent item set could be "Program 1, program 2 and program 3 are often watched together." and moreover if a user has watched a subset of these transactions, an association rule could be: "You watched program 1 and program 2, but you may also like program 3." and can be written as $\{Program1, Program2\} \implies \{Program3\}$ or more generally

$$X \implies Y \tag{5}$$

The item set X implies the item set Y. (cf. [8, p. 1455]) The implemented algorithm consists of two parts:

1. The generation of frequent item sets uses the Apriori approach. Therefore the watch-feedbacks of each user are represented as transaction. It is also possible to use other user-item-relations, such as the best rated items (analysed with the help of a rating threshold) for each user, or the watched items for each user and day. The threshold of total watched time (for genres, programs etc.) is 20 minutes for the prototype. When using the frequent item sets for more than only 300 users, the threshold has to adjust dynamically to a sound value. So the frequent item sets are found for the following domain data objects (the given minimum support values work fine for the test data set):
   - Categories (minSupp: 10%)
   - Channels (minSupp: 20%)
   - Crew Members (minSupp: 11%)
   - Genres (minSupp: 10%)
   - Programs (minSupp: 6%)
   - Series (minSupp: 6%)

   The results are sets of items that can be associated with each other, in this case they are often watched by the same users.

2. The generation of association rules for the calculated frequent item sets is done by a AssociationRuleGenerator that looks for the strongest rules. Some thresholds

are needed to adjust this algorithm. Especially the support value (*supp* in percent or sometimes as absolute value) is necessary and defines the frequency of an item set.

$$supp(X \implies Y) = \frac{supp(X \cup Y)}{|T|} \geq minsupp \quad (6)$$

$|T|$ is the total number of transactions, so $supp(X \implies Y)$ is the fraction of all transactions that contain X and Y.
The confidence value expresses the conditional probability of Y knowing X and is defined as:

$$conf(X \implies Y) = \frac{supp(X \cup Y)}{supp(X)} \geq minconf \quad (7)$$

The minimum confidence value ($minconf$) and the minimum support value ($minsupp$) are the according thresholds, so the goal is to find rules with a support value equal to or greater than $minsupp$ and a confidence value equal to or greater than $minconf$.

All results are based on the values: minimum Confidence (0.5) minimum Lift (1.1) and minimum Cosine (0.66). Over 500 interesting association rules resulted. Some of the strongest rules are stated below:

- Genre: "Docu-Soap", "Late-Night-Show" $\implies$ "Soccer",
- Category: "Entertainment", "Series", "Other" $\implies$ "Report", "Sport",
- Series: "The Simpsons", "Scrubs" $\implies$ "How I Met Your Mother",

4.4 Preference Filtering

Moreover the TV Predictor implements another approach representing a mixture of Content-based and Collaborative Filtering, the so called Preference Filtering. (cf. [19, p. 809]) Its goals are to find items that fit to the users' preferences. These preferences originate from manual user input or by learning them automatically. Therefore user interests (represented as attributes) are mapped to the item attributes, such as a sports interest of a user is mapped to the according category of an item (e.g. a sports program). (cf. [23, p. 244])

Support Vector Machines (SVMs) create hyperplanes that divide specific space with according points into two spaces. Each point represents an item and the according coordinates are its attributes. The Support Vector Machine will find the best fitting hyperplane that is the borderline between two classes (spaces). This can be done very effectively in multidimensional spaces. (cf. [10, p. 147], [21, p. 401-402])

The TV Predictor uses the Support Vector Machine Framework LIBSVM [3], that is also used by the common data mining tool WEKA. It is published as Java Archive (JAR) file and so it can easily be implemented into the engine. In addition there was an SVM training class that read data sets from ARFF Files. This `SVMTrainer.java` file was adapted to load the needed data sets from the according database and convert them to the needed domain structure.

The idea is to train a single SVM for each user with his rated slots respectively their features. The input data (the training as well as the prediction data set) are slot meta data of a watched program that must be converted into a list of numbers. So it is only applicable to use features that can be mapped to the required numeric range of [-1,1]:

- Slot time: The weekday (0-6), the duration of the slot (0-240 min) as well as the begin and end time (0-24 hours) are used.
- Program data: The production year (1920-2020) and the country String is used. The usage of Strings requires a mapping to more than one feature. So there is a feature for each possible country (D, GB, USA, E, F etc.) with the value 1 (it is this country) or 0 (it is not).
- Channel data: The same procedure as for the countries applies to the main 22 channels.
- Genre data: Same as above for 69 genres.
- Category data: Same as above for 6 categories.

So at all there are 109 features for each slot. To get a prediction, each training slot has to be labelled. In this case the label is numeric and either a 1, when the rating of the current user for that slot is higher than 7.5, or apart from that it is a -1.

4.5 Hybrid Filtering

The results of several calculations are combined to a final one. The TV Predictors recommendation engine uses the benefits of Content-based, Collaborative and Preference Filtering as well as the introduced offline learning algorithms. Depending on the request type the Hybrid Filtering algorithms the following approaches:

1. The Cascade Filtering uses the different algorithms in a sequential way. So each approach curtails the resulting set in order to avoid unnecessary calculations on already rejected items.
2. The Switching Filtering uses just one available approach.
3. And finally the Weighted Filtering merges the single results by weighting them using the following formula:

$$RV = \frac{\sum_{i=1}^{N} w_i * r_i}{\sum_{i=0}^{N} w_i} \quad (8)$$

   The recommendation value RV results from the summation of each single recommendation value r multiplied with the according weight w of the N different algorithms and afterwards divided by the summation of all weights. Which type of hybrid engine should be used, depends on the domain structure, the data sets, the context and the accuracy of the final prediction.

The starting point for a typical prediction is the set of all relevant items for the specific situation (e.g. a group of channels, genres or a specific time period). The system analyses all items and existing user-item-relations. If such a relation exists, the engine skips the rating prediction. Otherwise the introduced algorithms consecutively calculate predictions for the current item . If the results fit into a specific range (so considered as good proposals), the results are merged in a weighted way. If not, the result set is decreased by this item.

5. EVALUATION

769 users registered at the system and provided 20,368 ratings in total until May 15th 2013 – so in average 26 ratings per customer. In fact, 17 users rated over 100 items (the frontrunner even rated 487 different slots), whereas 102 customers didn't provide any rating. As a result they only receive average ratings.

The total available item set counts almost 150,000 slots, 17,000 unique series and 66,000 crew members. These numbers increase every day. As a result, 10,184 slots were rated. The most rated item is the motion picture 'The Silence of the Lambs' (with 130 ratings), that was played 13 times in the given time period, followed by a weekly German prime time series 'Tatort' (rated 127 times), 'Die Hard' (124 ratings), 'Who Wants to Be a Millionaire?' (118 ratings) and 'Miami Vice' (114 ratings).

An interesting result is the scatter of the rating values. While the lowest rating value (1) is used in 39 %, the highest value (10) is only used in 4.8 %. This may be caused by the fact, that people prefer to dislike instead of liking items or the initial prediction displays to much items, that users don't want to see again.

The easiest way to calculate the error of the final rating prediction is by subtracting the real value from the predicted value (cf. [17, p. 290], [22, p. 63]). The Root Mean Squared Error (RMSE) is calculated as:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} |p_i - q_i|^2}{N}} \quad (9)$$

$< p_i, q_i >$ is a prediction-real-value pair of N where p_i is the predicted value and q_i is the real value for item i. The lower the error the better the algorithm. One method to train and evaluate algorithms is the use of the n-fold cross validation. This cross validation type is repeated ten times. During every iteration the whole data set is split into another 10% of evaluation data and 90% of training data. The average value of errors defines the accuracy.

The rating prediction works very well for most of the users (average accuracy from 78.0% to 83.7%). But single users brings trouble for the system, as the accuracy is for one user about 62.7% on an average. The total rating accuracy is 80.2%.

The according Mean Absolute Error (MAE) is 1.98 and the Root Mean Squared Error is 3.77. So the errors seem to be very scattered, because MAE and RMSE have a great deviation. Simplified, predicted ratings range in intervals ± 1 (MAE) respectively ± 1.9 (RMSE).

Another challenge was the conformance of all local legal aspects. In general it is possible to develop web contents for SmartTVs when observing all potential laws applicable, as no judgement has been delivered in Germany so far that explicitly handles SmartTV applications. Especially the Data Protection Acts of Germany have to be respected for a productive Recommendation Application. Users must give their consent before being offered personalized recommendations. If they do not agree they will only be able to receive the anonymous Top-N Recommendations. But if they give their consent they must be able to see all data collected, to correct and delete them. In contrast, the usage data must be handled separately from the personalized data.

The offered recommendations and advertisements must not go beyond the legally allowed limitations. Showing recommendations as an overlay is possible if the broadcast program does not contain editorial or children's content. Since the application developer is not able to identify or even predict where and how these contents might be specifically used, the service provider must have the opportunity to deactivate and/or adjust the recommendations in an administrative section.

Just as every businesslike website has to provide information about the company behind it a SmartTV application must offer an About section as well.

6. CONCLUSION

This paper analyzed the problem of TV program recommendations to be displayed on SmartTVs. Due to the variety of recommendation approaches that can be classified in multiple ways, the search for and validation of the most suitable algorithms was a challenge. The most used web services offering recommendations are using Collaborative and Hybrid Filtering. There exists no recommendation system that allows showing program information including predicted ratings on TV sets, that can automatically collect usage data and is based on a common standard.

HbbTV is a "young" standardized specification that was successfully established in Germany and some countries of Mid-Europe. With 18 % of the German households already owning a SmartTV or connected Set-Top Boxes, the German market seems to be the best place to introduce technologies that benefit from this new infrastructure.

The broadcasters or content providers are hosting the Backend System. This leads to the problem that they may influence the recommendations requested. This can be an advantage, as humans should know best about other humans' likings. And it might happen that – when consumers addict to such a system – the recommended items will lose their variety and so for instance users will only see a very specific genre or category. Then the service provider can offer other items with an according explanation, such as "You never watched this!". But any influence bears the risk of manipulating the users' subconscious, when the users concerned do not know that they are influenced.

There are a lot of other possibilities to extend the TV Predictor. This can be a reminder that informs the customer when selected programs begin or a role system enabling different users and groups having their own settings while sharing one TV set. For instance, the family father wants to get other recommendations than the mother. It is important that when the whole family including the kids watch TV, only recommendations that are suitable for children are shown.

Retrieving feedback data like user ratings can be enhanced using gamification. Using this approach users may not feel like they are only supporting the server system in the background. An award system for the most given ratings or highscores may increase the interest for example. A badge systems for users that mark shows as currently whatching can be combined with small benefits. This may be a coupon for articles in the show or any addition that a broadcaster can offer to enhance the interest in using such a system.

Another aspect is the extension of the Personalized Program Data. Together with Video-On-Demand and Catch-up services the inevitable pauses between two programs may be filled with recommendations for other media.

With information about the users interests another interesting future work may be personalized commercial breaks. New technologies like the upcoming MPEG standard Dynamic Adaptive Streaming over HTTP (MPEG-DASH) enable opportunities where the bidirectional connection with the internet can be used to send personalized media data to the consumers. New versions of the HbbTV standard already support MPEG-DASH, although only few TV sets support it yet.

7. ACKNOWLEDGMENTS

The authors would like to especially thank Prof. Dr. Agathe Merceron of Beuth University of Applied Sciences for her support in structuring the main ideas, making practical suggestions for carrying on this work and helping with words and deeds. Moreover this paper would not have been possible without the aid of Robert Seeliger.

8. REFERENCES

[1] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, VOL. 17, NO. 6:734–749, 2005.

[2] A. B. Barragáns-Martínez, E. Costa-Montenegro, J. C. Burguillo, M. Rey-López, F. A. Mikic-Fonte, and A. Peleteiro. A hybrid content-based and item-based collaborative filtering approach to recommend tv programs enhanced with singular value decomposition. *Information Sciences*, 180(22):4290–4311, 2010.

[3] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[4] Y.-n. Chen and M. Yu. A hybrid collaborative filtering algorithm based on user-item. *IEEE Computer Society*, pages 618–621, 2010.

[5] A. Dattolo, F. Ferrara, and C. Tasso. The role of tags for recommendation: a survey. *IEEE*, pages 548–555, 2010.

[6] C. Dawes. Rovi hbbtv program guide. http://tilden.rovicorp.com/hbbtv/index.html, 2011.

[7] O. H. Embarak. A method for solving the cold start problem in recommendation systems. *IEEE - International Conference on Innovations in Information Technology*, pages 238–243, 2011.

[8] J. Han. Learning fuzzy association rules and associative classification rules. *IEEE - International Conference on Fuzzy Systems*, pages 1454–1459, 2006.

[9] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer Science+Business Media, 2001.

[10] M. F. Hornick, E. Marcadé, and S. Venkayala. *Java data mining - strategy, standard, and practice - A Practical Guide for Architecture, Design and Implementation*. Morgan Kaufmann Publishers, 2007.

[11] E. Kim, S. Pyo, E. Park, and M. Kim. An automatic recommendation scheme of tv program contents for (ip)tv personalization. *IEEE Transactions on Broadcasting*, VOL. 57, NO. 3:674–684, 2011.

[12] J. Kunegis and S. Albayrak. Adapting ratings in memory-based collaborative filtering using linear regression. *IEEE*, pages 49–54, 2007.

[13] D. Lemire and A. Maclachlan. Slope one predictors for online rating-based collaborative filtering. *Proceedings of SIAM Data Mining (SDM'05)*, 2005.

[14] G. Linden, B. Smith, and J. York. Amazon.com recommendations - item-to-item collaborative filtering. *IEEE Internet Computing*, pages 76–80, 2003.

[15] G. Liu, H. Jiang, R. Geng, and H. Li. Application of multidimensional association rules in personal financial services. *IEEE - International Conference On Computer Design And Appliations*, pages 500–503, 2010.

[16] P. Melville, R. J. Mooney, and R. Nagarajan. Content-boosted collaborative filtering for improved recommendations. In *AAAI/IAAI*, pages 187–192, 2002.

[17] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In *GroupLens Research Group/Army HPC Research Center*, 2001.

[18] C. Shin and W. Woo. Socially aware tv program recommender for multiple viewers. *IEEE Contributed Paper*, pages 927–932, 2009.

[19] N. Tintarev and J. Masthoff. A survey of explanations in recommender systems. *Data Engineering Workshop, 2007 IEEE*, 23rd International Conference:801 – 810, 2007.

[20] P. Wang and H. W. Ye. A personalized recommendation algorithm combining slope one scheme and user based collaborative filtering. *International Conference on Industrial and Information Systems*, pages 152–154, 2009.

[21] J. A. Xu and K. Araki. A svm-based personal recommendation system for tv programs. *IEEE*, pages 401–404, 2006.

[22] S. A. Yu, Kai, V. Tresp, X. Xu, and H.-P. Kriegel. Probabilistic memory-based collaborative filtering. *IEEE Transactions on Knowledge and Data Engineering*, VOL. 16, NO. 1:56–69, 2004.

[23] H. Zhang and S. Zheng. Personalized tv program recommendation based on tv-anytime metadata. *IEEE Consumer Electronics. Proceedings of the Ninth International Symposium*, pages 242–246, 2005.

[24] J. Zhou and T. Luo. Towards an introduction to collaborative filtering. *International Conference on Computational Science and Engineering*, pages 576–581, 2009.