PREDICTION STRATEGIES IN A TV RECOMMENDER SYSTEM – METHOD AND EXPERIMENTS

Mark van Setten, Mettina Veenstra
Telematica Instituut
P.O. Box 589, 7500 AN, Enschede, The Netherlands
mark.vansetten@telin.nl, mettina.veenstra@telin.nl

Anton Nijholt, Betsy van Dijk
University of Twente
P.O. Box 217, 7500 AE, Enschede, The Netherlands
anijholt@cs.utwente.nl, bvdijk@cs.utwente.nl

ABSTRACT

Predicting the interests of a user in information is an important process in personalized information systems. In this paper, we present a way to create prediction engines that allow prediction techniques to be easily combined into prediction strategies. Prediction strategies choose one or a combination of prediction techniques at the moment a prediction is required, taking into account the most up-to-date knowledge about the current user, other users, the information and the system itself. Results of two experiments show that prediction strategies improve both the accuracy and stability of prediction engines. One of these experiments involves a TV recommender system. This paper describes the method of prediction strategies, how they have been applied in the TV recommender system and results of the experiment in detail.

KEYWORDS

Personalization, Adaptive Systems, Recommender Systems, User Modeling

1. INTRODUCTION

Personalization is a very common issue in many aspects of our lives: labor contracts in which you can choose your own benefits (child support, lease car, vacation days, etc.), diets tuned to your personal needs, personalized pension plans, etc. Also with the delivery of information to users, personalization is important, as there is so much information available. One aspect of personalization is determining how interested a user will be in a certain piece of information, e.g. how interesting is this TV program for the viewer? How interesting is this book for the customer? How interesting is this news article for the reader?

Determining how interesting information is to a user is basically a form of predicting. Most of the currently available personalized information systems and research into these systems focus on the use of a single prediction technique or a fixed combination of two or three techniques (Smyth & Cotter, 2000) (Herlocker & Konstan, 2001). We believe that combining multiple techniques in a more dynamic and intelligent way can provide more accurate and stable predictions.

By dynamic and intelligent combinations we mean that the combination of prediction techniques should not be fixed within a system, that the combination ought to be based on knowledge about strengths and weaknesses of each prediction technique, and that the choice of prediction techniques to use should only be made at the moment an actual prediction is required, taking into account the most up-to-date knowledge about the current user, other users, the information and the system itself. We call such a combination of prediction techniques a prediction strategy.

This paper first describes our method of using prediction strategies (section 2). Section 3 introduces two experiments we performed and describes one, the TV recommender system, in more detail. Section 4 provides results of the TV recommender experiment, while the final section provides conclusions and directions for future research.

2. TECHNIQUES AND STRATEGIES

2.1 Prediction Techniques

A prediction technique calculates how interested a certain user will be in a piece of information, using some sort of algorithm. The resulting prediction is a numerical value representing the amount of expected interest for the user. Examples of prediction techniques are social filtering (Shardanand & Maes, 1995) (Herlocker, 2000), techniques from case-based reasoning (CBR) (Jackson, 1990), techniques from information filtering (Houseman & Kaskela), item-item filtering (Rashid et.al., 2002), and genre Least Mean Square (genreLMS) (van Setten, 2002). This section introduces a generic model of such prediction techniques, which allows us to easily combine prediction techniques into prediction strategies.

Even though there are different types of prediction techniques, it is possible to create a generic model due to the basic nature of each prediction technique: each technique can calculate a predicted interest value, simply called a *prediction*, of a piece of *information* for a given user, based on knowledge stored in the *user profile*, data and metadata of the information and profiles of other users. This forms the basis of our model (see Figure 1). Naturally, each technique must normalize its predictions in order to be comparable and to combine predictions. We use the bipolar range from –1 to +1 (zero being neutral).

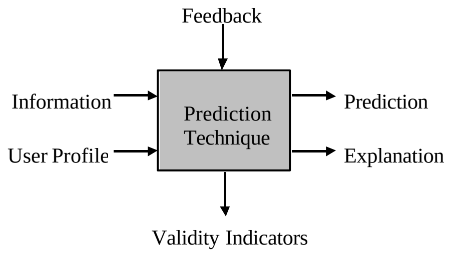

Figure 1. Generic model of a prediction technique.

Several techniques, such as social filtering, CBR and genreLMS, are capable of learning from users in order to optimize future predictions. They learn from *feedback* provided by users, whether being explicit or implicit feedback. For this reason, feedback is also incorporated in the model.

Optionally, prediction techniques can provide *explanation* data. Explanations provide transparency, exposing the reasoning and data behind a prediction and can increase the acceptance of prediction systems (Herlocker, Konstan & Riedl, 2000). Explanations help users to decide whether to accept a prediction or not and to understand the incorrect reasoning when a prediction is inaccurate. This in turn helps to increase the trust a user has in the prediction engine. Our current focus lies not on explanations but on predictions.

In order to make informed decisions about when techniques are useable, each technique exposes so-called *validity indicators*. Validity indicators are features of a prediction technique that provide information about the state of the technique and can be used to determine how useful the technique will be in predicting the user's interests. Validity indicators are analogous to the concept of reliability indicators as described by Toyama & Horvitz (2000) and Bennett, Dumais & Horvitz (2002) but have been developed independently. Because of differences in prediction techniques, most techniques have unique and different validity indicators. E.g. where social filtering provides the number of similar users that rated the information, CBR provides the number of similar rated items by the user. These indicators can be used by strategies (see section 2.2) to decide to what extent a technique is likely to give a good prediction. Examples of prediction techniques and their validity indicators are given in section 3, where we describe the TV recommender system in more detail.

2.2 Prediction Strategies

Prediction strategies generate an interest prediction for a given piece of information and user, not using some sort of prediction algorithm but by selecting between and combining prediction techniques based on the most up-to-date knowledge about the current user, other users, the information and the personalized information system itself. To make decisions about which prediction techniques to use and how to combine them, a strategy uses a decision approach. Examples of possible approaches are the use of hard decision rules (if … then … else …), fuzzy rules, artificial neural networks, Bayesian networks, etc. (Mitchell, 1997). Prediction strategies use data stored in the user profile, data and metadata of the information, system status data and most importantly the validity indicators of the prediction techniques to decide about what prediction techniques to select and/or combine.

From a black box perspective, prediction strategies are no different than prediction techniques. Both predict the interest in a piece of information for a user: both are predictors. This means that the generic model of a prediction technique also applies to prediction strategies.

However, when looking into the black box, prediction techniques actually generate predictions based upon the user profile and information, whereas prediction strategies only choose one or more predictors (prediction techniques and/or other prediction strategies) that generate predictions on behalf of the strategy.

One of the main advantages of the distinction between prediction techniques and prediction strategies is that it allows for the development of prediction techniques independent of the domain in which they will be used. These techniques can then easily be used and combined by others into prediction strategies, where they are tuned to the specific domain in which they must operate. E.g. a prediction technique based on CBR can be developed without looking at the domain in which it will be used, leaving only the similarity calculation between two information items open for implementation when it is to be applied in a specific domain.

3. EXPERIMENT

3.1 TV Recommender

In order to validate the usage of prediction strategies and our prediction framework, we performed experiments on two different datasets:

- MovieLens (http://www.grouplens.org). This is the publicly available dataset from a movie recommendation system developed at the University of Minnesota. The dataset consists of 100,000 ratings by 943 users for 1682 movies. The results of this experiment have already been reported in van Setten, Veenstra & Nijholt (2002).
- TiV (see Figure 2). In this experiment, we asked 24 people to rate four weeks of TV programs from Dutch television (broadcasted between 15 August 2002 and 14 September 2002), containing 40,539 broadcasts from 47 different channels. Participants, of course, only rated those programs they had an opinion about, resulting in a total of 31,368 ratings. The four weeks include a transition from the summer TV season to the winter season at September 1st. This transition has been included deliberately as it helps to show that prediction strategies are more stable than prediction techniques, even when a large change in the information source takes place.

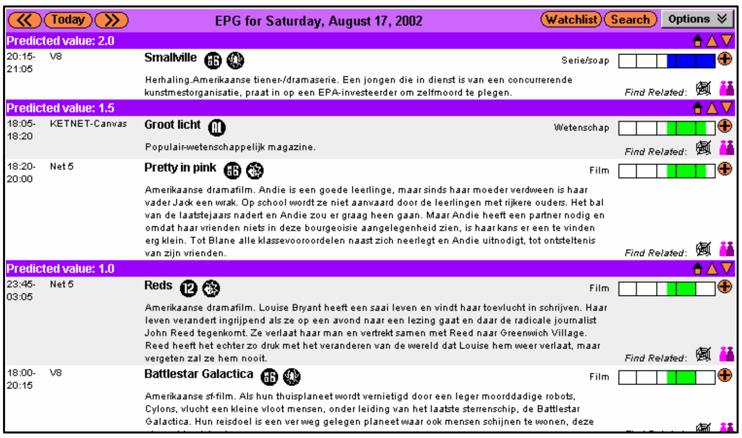

Figure 2. Screenshot of TiV.

In MovieLens, data has been gathered via real usage of the system, while in TiV, data has been gathered by asking people to rate all programs for which they have an opinion. Both methods have its advantages and disadvantages. The main drawback of asking people to just rate a long list of programs is that it does not represent a normal usage situation. On the other hand, it avoids one of the problems of gathering data for research in a real recommender system: people tend to only give feedback on incorrect predictions. Imagine the perfect prediction engine, one where every prediction reflects the actual interest of the user. In such a system, users no longer give feedback, as predictions are already correct. From the user's point of view, this is an ideal situation, but not for research purposes, as there is no way to quantifiably validate the performance of such a prediction engine. However, the same problem occurs in less perfect, but more realistic, prediction engines. In such systems, there is no way of knowing whether a prediction for which no feedback has been received was either correct, whether the user has not seen the information or whether he has no opinion about it. By asking people to rate a list of programs, there is either a large chance that the user has no opinion about those programs for which no rating is available or that he has already rated that same program multiple times.

In the experiments, we only used hard decision rules as a strategy decision approach, as our research focus was to only get a first indication of the power of prediction strategies. Future research will focus on using different strategy decision approaches. The rules for TiV were created manually based on known strengths and weaknesses of the different used prediction techniques. In this experiment, we used the following prediction techniques:

- *AlreadyRated* returns the rate of an item that the user has already rated from the user profile. The validity indicator "Known" returns true if the user already rated the item, otherwise false is returned.
- *UserAverage* returns the average of all ratings provided by the user. No validity indicators of this technique are used, although it has the indicator "NumberOfRatedItemsByUser", which returns the number of items that have already been rated by the user.
- *TopNDeviation (TopNDev)* returns a prediction based on all predictions from other users that already rated this item. The exact algorithm used is the deviation-from-mean average over all users as described by Herlocker (2000). The validity indicator "NumberOfUsersThatRatedItem" of this technique returns the number of users that have already rated the information item.
- *Social Filtering (SF)* is based on the idea that people that have rated the same items the same way in the past probably also have similar interests patterns. Based on this knowledge one can predict how much a person likes an unseen item when similar users already rated that item. We use the SF algorithm as described by Herlocker (2000) with Pearson correlations and using the deviation-from-mean SF algorithm. The two validity indicators are: the number of items that the user has already rated ("NumberOfRatedItems ByUser") and the number of similar users that have already rated the item ("NumberOfSimilarUsersWho RatedItem"), where users are similar when they have rated at least the same 50 items.
- *Case Based Reasoning (CBR)* is based on the idea that if two items are similar and if a rating is known for one of them, the rating for the other will probably be the same. This idea comes from the overall research in CBR (Jackson, 1990). Our algorithm calculates the weighted average, using the similarities between items as weights, over all already rated items with a similarity of 0.5 or more. CBR has one validity indicator called "NumberOfSimilarItemsWithSimilarity(*t*)" that returns the number of already rated items that have a similarity value of at least *t* with the current item.
- *GenreLMS*, this prediction technique learns how interested a user is in the main genres of a TV program using a linear function between the possible genres (van Setten, 2002). The validity indicator of this technique is "Certainty", which is the average certainty of each genre in the information item as indicated by the user profile, which is a number between 0 and 1. Each time the user rates an item with a specific genre, the certainty of that genre is altered: when the prediction was positive and the weight of that genre in the user profile is positive or when the prediction was negative and the weight of that genre in the user profile was negative, certainty is increased by 0.1, otherwise it is decreased with 0.1.
- *SubGenreLMS*, this is the same technique as GenreLMS, using the same validity indicator. The only difference is that GenreLMS works on the main TV genres (comedy, serial, nature, news, movie, etc.), whereas SubGenreLMS works on an alternate set of sub genres (*English* comedy, *action* movie, *French Science-Fiction* movie).
- *Information Filtering (InfoFilter)*. This prediction technique is similar to GenreLMS, except that it uses all words (with stop words removed and stemmed) from the TV program descriptions and their frequency as weights in calculating the prediction and only those words for which an interest is known in the user

profile. The validity indicator of this technique is "Certainty", which is the average certainty for each word in the information item as indicated by the user profile, which is a number between 0 and 1. Learning the interests of the user in the words and the certainty of these interests is done as described for GenreLMS.

- *Default* returns a neutral prediction value of 0 (which is how users are likely to see a non prediction) and has no validity indicators.

Based on this set of prediction techniques, we created one main TV strategy that uses two sub strategies: the TV Fallback Strategy and Genre Strategy. The decision rules for these three strategies are shown in Figure 3.

TV Strategy

```
IF AlreadyRated.Known = true THEN USE AlreadyKnown
   ELSEIF SF.NumberOfRatedItemsByUser ≥ 25 AND
          SF.NumberOfSimilarUsersWhoRatedItem ≥ 20 THEN USE SF
          ELSEIF CBR.NumberOfItemsWithSimilarity(0.9) ≥ 1 OR
                 CBR.NumberOfItemsWithSimilarity(0.7) ≥ 5 OR
                 CBR.NumberOfItemsWithSimilarity(0.5) ≥ 11 THEN USE CBR
                 ELSEIF InfoFilter.Certainty > 0.5 THEN USE InfoFilter
                        ELSE DO TVFallbackStrategy
```

TV Fallback Strategy

```
IF GenreLMS.Certainty ≥ 0.35 OR SubGenreLMS.Certainty ≥ 0.35 THEN
   DO GenreStrategy
   ELSEIF TopNDev.NumberOfUsersThatRatedItem ≥ 10 THEN USE TopNDev
          ELSE USE UserAverage
```

Genre Strategy

```
IF GenreLMS.Certainty ≥ 0.35 AND GenreLMS.Certainty ≥ 0.35 THEN
   USE WeightedAverageOf(GenreLMS, SubGenreLMS)
   ELSEIF GenreLMS.Certainty ≥ 0.35 THEN USE GenreLMS
          ELSEIF SubGenreLMS.Certainty ≥ 0.35 THEN USE SubGenreLMS
                 ELSE USE Default
```

Figure 3. Decision rules of the strategies used in the TiV Experiment.

3.1 Performance Measures

According to Herlocker (2000), the two best measures to evaluate predictions in recommender systems are:

1. The mean absolute error (mae): this measures the average absolute deviation between a predicted rating and the user's actual rating. The lower the mae, the better the performance of a predictor;
2. Coverage: percentage of items for which a prediction could be generated. Some techniques cannot always provide a prediction. E.g. social filtering only generates a prediction when similar users can be found for the current user.

We believe that in some systems, coverage is more important than in others. E.g. in a rental movie recommendation system, it is less important if a prediction cannot be made for a specific movie, as long as predictions that could be made are correct: it is not important if users rent a certain movie this week or in several months, as long as they will like the currently recommended movies. However, in a TV system, coverage is more important as most TV programs are only available at the moment of broadcasting. This means that we are interested in both prediction quality and coverage at the same time. For this reason, we combined both measures into a new measure: the global mean absolute error (gmae). This measure is the same as mae, but when a prediction cannot be made the default value (zero) is assumed.

To compare the prediction techniques and strategies throughout the usage period of the system, we divided the set of ratings in each dataset into different validation sets. Set A consisted of the first set of ratings (in time), Set B of the next set of ratings, etc. When testing each set, the ratings of all previous sets were used for training. The TiV dataset was divided into four sets, one set per week (with 8867, 8843, 6890, and 6768 ratings respectively), with the transition from summer to winter season between week 2 and 3.

We also evaluated the prediction techniques throughout the user usage cycle. Because all users in the TiV system started using the system at the same time, we only looked at the first 100 ratings of each user within the whole time period to see how well the prediction strategies and techniques performed for new users.

For all six sets in the TiV dataset, we calculated the gmae for each prediction technique and strategy and performed paired samples T-tests to determine if differences between the results are statistically significant (using a 95% confidence interval, meaning that if $p < 0.05$ then differences are statistically significant). In

TiV, we did not use the original mae, because techniques like social filtering could not produce enough ratings for valid statistical analysis due to the small number of users.

3.2 Validation Process

The validation process is carried out using the following steps: The ratings provided by the user are fed to the system one by one, in the logical order of the system (in TiV the logical order is the order in which the programs were broadcasted). When a rating is provided during validation, first all prediction strategies and prediction techniques are invoked to provide a prediction for the current user and the current item. The absolute difference between the prediction provided by each prediction technique and strategy and the actual rating is the prediction error. After the errors for all prediction strategies and techniques have been calculated, the actual rating from the user is given as feedback to all the prediction strategies and techniques and is also stored in the user profile. Several prediction strategies and techniques use this actual rating to learn. This way, when a new rating is processed, the system and all predictors know and have learned from all previous ratings. This process is repeated for all ratings in the specific test set. At the end the gmae can be calculated by averaging all absolute errors that fall within each of the sets.

4. RESULTS

Table 1 shows that the TV Strategy used for the TiV system out-performs all prediction techniques, in each of the four weeks. This better performance of the TV strategy is statistically significant. We noticed that the CBR prediction technique is also very good. This can be contributed to a specific characteristic of the TV guide data we received for this experiment. In this data, two programs that most people would see as the same program, e.g. two episodes of Friends that are broadcasted on different days are actually listed as two different programs with different identifiers. For this reason, if a user has rated Friends highly, another episode of Friends does not have a known rate, but needs to be predicted by other prediction techniques. In this case, CBR is especially successful in predicting the interest, as the similarity between those programs is of course very high.

One can also see, that the decrease in accuracy when switching from the TV summer season to the TV winter season is only 9% for the TV Strategy, while there is a decrease of 14% in the best individual technique: CBR. This indicates that the TV Strategy is capable of falling back on other techniques when otherwise excellent prediction techniques begin to falter.

Table 1. Accuracy of the predictors in TiV (The lower value, the better the performance of a predictor).

	Overall	**Week 1**	**Week 2**	**Week 3**	**Week 4**
TV Strategy	**0.2100**	**0.2935**	**0.1774**	**0.1934**	**0.1602**
CBR	0.2376	0.3530	0.1899	0.2171	0.1693
Information Filtering	0.2919	0.3679	0.2598	0.2769	0.2495
TV Fallback Strategy	0.3835	0.4128	0.3645	0.3832	0.3701
Genre Strategy	0.3870	0.4211	0.3662	0.3850	0.3716
GenreLMS	0.3975	0.4105	0.3751	0.4036	0.4035
SubGenreLMS	0.4129	0.4558	0.3982	0.4105	0.3781
Social Filtering	0.4820	0.4850	0.4764	0.4852	0.4822
TopNDeviation	0.5241	0.5199	0.5202	0.5286	0.5300
UserAverage	0.5398	0.5301	0.5294	0.5512	0.5544
AlreadyRated	0.6126	0.6086	0.6084	0.6234	0.6126

When looking at the performance for both new users and established users (see Table 2), it appears that the TV strategy again is out-performing the prediction techniques (statistically significant).

Table 2. Accuracy of predictors for new and established users.

	New	**Established**		**New**	**Established**
TV Strategy	**0.4145**	**0.1913**	SubGenreLMS	0.5174	0.4042
CBR	0.4790	0.2175	Social Filtering	0.4877	0.4815
Information Filtering	0.4748	0.2768	TopNDeviation	0.4995	0.5261
TV Fallback Strategy	0.4902	0.3746	UserAverage	0.5412	0.5397
Genre Strategy	0.4995	0.3777	AlreadyRated	0.6021	0.6135
GenreLMS	0.4906	0.3898			

Furthermore, we also looked at the influence of removing individual prediction techniques from the TV prediction strategy. This shows the necessity of using multiple prediction techniques within prediction strategies. In Table 3, we show the impact over the whole four-week period when removing several prediction techniques and prediction strategies from the TV Strategy. From these results, one can conclude that removing CBR, UserAverage, the Genre Strategy and the TV Fallback Strategy negatively influences the accuracy of the predictions and that this negative influence is statistically significant. Removing AlreadyRated has a positive influence on the prediction accuracy, which is logical because people only give the same program another rating when they previously entered an incorrect rating. This however does not mean that the AlreadyRated prediction technique should be removed from the TV Strategy. An already given rating is, most of the times, the best indication of the user's interest as the user explicitly specified it. This does not show up in the accuracy of the AlreadyRated technique, as users will not re-rate an item if the given rating is already correct. In our current dataset, removing SF or TopNDeviation did not make any difference, because the number of users was not enough for these two techniques to be used in the first place (SF was never used, hence the significance value *p* could not be calculated, TopNDeviation was only used eleven times for all 31368 predictions). However, in the MovieLens experiment, removing these two techniques did have a significant influence. Removing Information Filtering decreased the accuracy only a little but not statistically significant. For this reason, we should remove that technique from the strategy, try to improve the technique itself or try to optimize the decision rule for Information Filtering within the TV Strategy (this process is not described in this paper).

Table 3. Influence of removing a single prediction technique or sub strategy on the accuracy.

Removed technique	gmae	p	Removed strategy	gmae	p
CBR	0.3064	.000	Social Filtering	0.2100	-
Information Filtering	0.2104	.404	AlreadyRated	0.2058	.000
UserAverage	0.2114	.000	Genre Strategy	0.2223	.000
TopNDeviation	0.2101	.101	TV Fallback Strategy	0.2318	.000

The results of the experiments show that using multiple prediction techniques combined into prediction strategies indeed provide more accurate predictions than a single or a fixed combination of two or three prediction techniques. Although improvements in individual prediction techniques are still necessary (and encouraged as they will also improve the performance of strategies), the conclusion of this experiment and the MovieLens experiment (van Setten, Veenstra & Nijholt, 2002) is that prediction strategies are indeed more capable of predicting user interests and they also provide more stable predictions.

5. CONCLUSIONS AND FUTURE RESEARCH

In this paper, we described a new way of looking at prediction engines in personalized information systems. We described our method of using prediction strategies, which is based on a generic model for prediction techniques. This method makes it is possible to quickly create, use and test different prediction strategies using several prediction techniques and allows for the independent development of prediction techniques and their application in different domains.

The results of the experiments show that, because prediction strategies decide only at the moment a prediction is required what (combination of) prediction techniques to use, prediction strategies indeed improve the accuracy and stability of prediction engines. Because of the generic nature of our method and the success in two different systems, we believe that our approach can be applied in many different personalized information systems, such as news recommendation systems, personalized radio broadcasts, personalized electronic travel guides and recommendations systems for digital libraries.

One must be aware of one of the drawbacks of using prediction strategies: there is a performance penalty (in processing time) due to the decisions made within strategies. In systems that use a fixed (combination of) technique(s), no decisions have to be made about what prediction techniques to use. The performance penalty is determined by the complexity of the validity indicators used, as they have to be calculated by the different prediction techniques before a prediction strategy can make a decision. A first good solution is to only calculate a validity indicator when it is absolutely necessary for the strategy. Another solution would be to use as many simple (requiring the least processing power) validity indicators as possible. More research is needed in determining per prediction technique the best and most simple validity indicators.

In the current experiment, we only used hard decision rules as an approach for prediction strategies. Creating these decision rules requires expert knowledge about the different prediction techniques and the domain in which the prediction strategies are to be applied. For this reason, the focus of our future research lies on experimenting with different learning approaches, such as fuzzy rules that better combine prediction techniques in situations where more than one prediction technique is valid, artificial neural networks or Bayesian networks that learn themselves when to use what prediction techniques, etc. Also the difference between using static approaches and dynamic approaches will be investigated. With static approaches, the decision model is fixed within the strategy during actual usage and only updated every now and then (e.g. a learned neural network is implemented and is not updated dynamically during usage). With dynamic approaches, the decision model is directly updated based on feedback received by users. We therefore expect that dynamic approaches (perhaps bootstrapped with a learned static decision model) are better capable of anticipating unforeseen situations, but have a larger performance penalty than static approaches.

ACKNOWLEDGEMENTS

This research is part of the PhD project Duine (http://duine.telin.nl) at the Telematica Instituut (http://www.telin.nl). This work has been performed in the GigaPort project GigaCE (http://www.gigaport.nl) and the Freeband projects Xhome and WASP (http://www.freeband.nl). I would like to thank Wolf Huijsen and Wojtek Jamroga for their helpful comments. Also thanks to all participants of the TiV experiment and to the researchers at the university of Minnesota for making the MovieLens dataset publicly available.

REFERENCES

Bennett, P. N., Dumais, S. T. and Horvitz, E. (2002). Probabilistic Combination of Text Classifiers Using Reliability Indicators: Models and Results. *Proceedings of SIGIR'02*. Tampere, Finland, pp. 207-214.

Herlocker, J. (2000). *Understanding and Improving Automated Collaborative Filtering Systems*. University of Minnesota.

Herlocker, J. and Konstan, J.A. (2001). Content-Independent Task-Focused Recommendation. *IEEE Internet Computing*, Vol. 5, pp. 40-47.

Herlocker, J., Konstan, J.A. and Riedl, J. (2000). Explaining Collaborative Filtering Recommendations. *Proceedings of CSCW'2000*. ACM, Philadelphia, pp. 241-250.

Houseman, E. M. and Kaskela, D. E. (1970). State of the art of selective dissemination of information. *IEEE Trans Eng Writing Speech III*, pp. 78-83.

Jackson, P. (1990). *Introduction to Expert Systems*. Addison-Wesley, Reading, MA.

Mitchell, T.M. (1997). *Machine Learning*. McGraw-Hill, Singapore.

Rashid, A.M., Abert, I., Cosley, D., Lam, S.K., McNee, S.M., Konstan, J.A. and Riedl, J. (2002). Getting to Know You: Learning New User Preferences in Recommender Systems. *Proceedings of ACM Intelligent User Interfaces 2002*. ACM, New York, pp. 127-134.

Shardanand, U. and Maes, P. (1995). Social information filtering: algorithms for automated "Word of Mouth". *Proceedings of Human factors in computing systems 1995*. ACM, New York, pp. 210-217.

Smyth, B. and Cotter, P. (2000). A personalised TV listings service for the digital TV age. *Knowledge-Based Systems*, Vol. 13, pp. 53-59.

Toyama, K. and Horvitz, E. (2000). Bayesian Modality Fusion: Probabilistic Integration of Multiple Vision Algorithms for Head Tracking. *Proceedings of ACCV 2000*, Fourth Asian Conference on Computer Vision.

van Setten, M., Veenstra, M. and Nijholt, A. (2002). Prediction Strategies: Combining Prediction Techniques to Optimize Personalization. *Proceedings of the workshop Personalization in Future TV'02*, Malaga, Spain, pp. 23-32.

van Setten, M. (2002). Experiments with a recommendation technique that learns category interests. *Proceedings of IADIS WWW/Internet 2002*, Lisbon, Portugal, pp. 722-725.