Case-Based Reasoning as a Prediction Strategy for Hybrid Recommender Systems

Mark van Setten[1], Mettina Veenstra[1], Anton Nijholt[2], Betsy van Dijk[2]

[1] Telematica Instituut, P.O. Box 589, 7500 AN, Enschede, The Netherlands
{Mark.vanSetten, Mettina.Veenstra}@telin.nl
[2] University of Twente, P.O. Box 217, 7500 AE, Enschede, The Netherlands
{anijholt, bvdijk}@cs.utwente.nl

Abstract. Hybrid recommender systems are capable of providing better recommendations than non-hybrid ones. Our approach to hybrid recommenders is the use of prediction strategies that determine which prediction technique(s) should be used at the moment an actual prediction is required. In this paper, we determine whether case-based reasoning can provide more accurate prediction strategies than rule-based predictions strategies created manually by experts. Experiments show that case-based reasoning can indeed be used to create prediction strategies; it can even increase the accuracy of the recommender in systems where the accuracy of the used prediction techniques is highly spread.

1 Introduction

Recommender systems are intelligent systems that are capable of helping people to quickly and easily find their way through large amounts of information by determining what is interesting and what is not interesting to a user. Recommenders employ prediction techniques to determine what is and what is not interesting by learning from the user and sometimes other users. Examples of such techniques are information filtering [3], social filtering [6] [12], genre-based recommendations [15], case-based reasoning (CBR) [13], and item-item filtering [7]. Current recommender research focuses on using a mixture of prediction techniques. Such hybrid recommenders are capable of providing better recommendations than individual techniques [1] [4] [5] [16] [17]. Our hybrid recommender approach uses prediction strategies that determine which prediction technique(s) should be used at the moment an actual prediction is required. Initially we used manually created rule-based strategies. Creating these manual strategies requires expert knowledge; because this is a major drawback, we now focus on strategies that teach themselves when to use which techniques.

This paper determines whether CBR can provide more accurate prediction strategies than manually created rule-based predictions strategies. We first describe hybrid recommenders in general and our hybrid recommender approach in particular. We then explain how we use CBR as a prediction strategy. This is followed by a description of the experiments in which CBR as a prediction strategy is compared with manually created rule-based strategies. We then present and discuss the results of the experiments and conclude this paper on using CBR as a prediction strategy.

J. Favela et al. (Eds.): AWIC 2004, LNAI 3034, pp. 13-22, 2004.

2 Hybrid Recommenders and Prediction Strategies

Burke [5] describes seven different types of hybridization methods for recommender systems: weighted, mixing, switching, feature combination, cascade, feature augmentation, and meta-level. He also provides an example of a cascade hybrid recommender system called EntreeC that recommends restaurants to users using both CBR and social filtering as prediction techniques.

Buczak, Zimmerman and Kurapati describe a weighted hybrid TV recommender that combines three prediction techniques [4]: two techniques that analyze implicitly gathered viewing data using Bayesian learning and Decision Tree learning, and one technique that contains explicitly provided interests about the user for different aspects of TV programs, such as times, channel and genres. Predictions of these three techniques are fused with an artificial neural network.

Another example of a weighted hybrid TV recommender is described by Ardissono, Gena, Torasso, Bellifemine, Chiarotto, Difino and Negro [1]. In this system, three prediction techniques are combined: a stereotype-based technique, a technique based on explicitly provided interests from the user, and a technique that employs a Bayesian belief network that learns from implicitly gathered user behavior data. The weights used to combine the predictions are based on confidence scores provided by the individual techniques.

Our approach employs switching hybridization by deciding which prediction technique is most suitable to provide a prediction. The decision is based on the most up-to-date knowledge about the current user, other users, the information for which a prediction is requested, other information items and the system itself [16] [17]. Such a hybrid is called a prediction strategy. Where prediction techniques actually generate predictions, strategies only choose one or more predictors that generate predictions on their behalf. Predictors are either prediction techniques or other prediction strategies.

The way prediction strategies retrieve information for their decision is similar to the ensemble method approach used in visual analysis [14] and text classification [2]. Each prediction technique exposes a set of reliability or validity indicators providing information that can be used to decide whether to use the technique or not. Two examples of validity indicators are ‘the number of similar users that rated the information’ for social filtering and ‘the number of similar rated items by the user’ for CBR.

Prediction strategies can use one of several decision techniques to make a decision based on the validity indicators; examples of such decision techniques are decision rules, neural networks, Bayesian networks and CBR. In [16] [17] we have shown that prediction strategies indeed provide more accurate predictions when using manually created decision rules. These rules were created using expert knowledge on the different prediction techniques and the two domains for which the prediction strategies were created: a TV recommender and a movie recommender.

As the need for expert knowledge is a drawback of manually created rule-based strategies, we have been investigating the possibility to create prediction strategies that teach themselves when to use which prediction techniques. Several machine learning algorithms can be used as a decision technique, including neural and Bayesian networks. However, due to the nature of prediction strategies, specifically the usage of validity indicators, CBR is the most promising and hence investigated in this paper. The other algorithms are topics for future research.

3 Case-Based Reasoning Based Strategy

CBR is a method to solve new problems by adapting solutions that were used to solve past problems [10]. With CBR, one searches for past cases that are analogous to the current case; the solutions of the most analogous past cases are then used to create a solution for the current case. Because all prediction strategies try to determine which prediction technique can best provide a prediction for a specific prediction request, a prediction strategy using CBR must consequently do so. A prediction strategy can only learn how well the different prediction techniques performed when feedback has been received from the user. This feedback represents the actual interest of the user, which the strategy can compare with the predictions of the individual techniques in order to determine which technique retrospectively predicted best. For this reason, when using CBR as a prediction strategy, a case represents a specific prediction request for which feedback of the user has been received. As validity indicators provide information that can be used to decide whether to employ a technique or not, these indicators can also be used by CBR to describe the case of a prediction request. E.g. the three validity indicators of the CBR prediction technique are: number of similar items rated by the user where sim > 0.5, where sim > 0.7 and where sim > 0.9.

For prediction strategies, it is best to use a case-base per prediction technique instead of a global case-base with all validity indicators of all techniques. In a global case-base problems arise when techniques are added to, changed within or removed from a strategy; all old cases become invalid as they are based on a set of techniques that no longer exists. With a case-base per technique, only for the new or changed technique must the case-base be rebuilt. Furthermore, with a global case-base, a larger case-base is necessary before accurate decisions can be made: the probability that a similar set of validity indicators occurs for one technique is higher than the probability that a similar set of indicators occurs for many techniques at the same time.

The outcome of a prediction technique is a score indicating the predicted interest of the user; the outcome of the decision within a strategy is not the prediction but an indication which prediction technique can best be used to provide the prediction. In order to keep the set of techniques flexible, a case solution is needed that not only reflects the performance of a technique, but that is also independent of the other techniques. When using an indication such as a rank that relates different techniques to each other, any change to the set of techniques would render every case-base invalid. We use the prediction error of techniques as the score. The prediction error is the absolute difference between the prediction and the feedback of the user; the lower the error the more suitable that technique was, in retrospect, to generate the prediction.

For every prediction request, the goal of the CBR-based strategy is twofold. First, determine the expected error of each prediction technique using those stored cases that have similar validity indicators as the current situation. Second, choose the technique with the lowest expected error as the one that should provide the prediction.

3.1 Determining Analogous Cases

The key element of CBR is determining which old cases are analogies of the current prediction request [18]. Traditional CBR systems calculate the distance between two cases; old cases with the least distance from the current case are retrieved to determine the outcome for the current case. The closer a case is to the current situation, the more important that case should be in determining the outcome for the current request; the importance should be used as a weight.

The four most frequently used distance measures in CBR are [8]: unweighted Euclidean distance (ued), weighted Euclidean distance (wed), maximum measure (mms) and mean squared difference (msd). We also explored an information retrieval measure that calculates similarity instead of distance, namely cosine similarity (cs) [11]. With cosine similarity, the similarity between two cases is determined by the cosine of the angle between two vectors containing the values of the case attributes.

Using distances as weights is difficult due to the non-fixed upper level of distance; the maximum possible distance is unknown or may differ per distance measure. However, similarity measures are in the range of [0, 1] [18], one means that two cases are exactly the same and zero means that two cases are completely different. As traditional CBR systems use distances, it is necessary to convert distances into similarities. We explored three types of conversion functions for the conversion of distance to similarity: linear functions, sigmoidal functions and the inverted function. The linear function *linear(d, m)* converts the distance *d* to similarity *s* linear over the distance from 0 to *m*. Distances equal to or larger than *m* all have a similarity of 0. Sigmoidal functions are often used in machine learning for smooth transitions [9]. The sigmoidal function *sigmoidal(d, k, m)* has a tuning parameter *k,* which is used to determine the flatness of the smoothing function. The sigmoidal function, including transformations to place it in the distance domain of [0, *m*] and in the similarity range of [0, 1] is:

$$s = \frac{1}{1 + e^{k(d - \frac{1}{2}m)}} \tag{1}$$

Using 0.5*m* assures that similarity is halfway between zero distance and the maximum affordable distance *m*; in our experiments we choose this midpoint in order to determine the smoothening effects of a sigmoidal function compared to linear functions that also have a midpoint at 0.5*m*. However, the inverted function *inverted(d)* does not have its similarity midpoint at 0.5*m*. The function used is the inverted function, transposed to have a value of 1 at a distance of 0:

$$s = \frac{1}{(d + 1)} \tag{2}$$

3.2 Limiting Case-Base Size

One of the drawbacks of using CBR as a prediction strategy is the scalability issue of CBR: the larger the case-base, the more time it takes to make a decision; for every prediction request, similarity has to be calculated between the current request and all cases in the case-bases. For this reason, we also experimented with limiting the size of the case-bases using the first-in/first-out method, which allows the system to keep learning from recent situations. Recent situations have a higher probability of resembling future prediction requests than much older cases.

However, removing old cases from the case-bases has a risk: some of the removed cases may represent situations that, although they do not occur often, they do occur every now and then; e.g. a new user registering system. If the case-base is too small, the cases representing such an event may have been removed by the time such an event reoccurs. Because the frequency in which special situations occur differs per system, the optimal size of the case-bases also differs per system; the more frequently special situations occur, the smaller the necessary case-base size. For this reason, it is necessary to experiment with case-base sizes in order to determine the optimal size: optimal with regard to both accuracy and speed.

4 Experiments

In order to determine how well CBR-based strategies perform compared to manually rule-based prediction strategies, we performed experiments on two datasets:

- The TiV dataset [17]. This dataset contains 31,368 ratings of 24 people for 40,539 broadcasts from four weeks of TV programs from Dutch television.
- The 1 million ratings MovieLens dataset. This is the publicly available dataset from the movie recommendation system developed at the University of Minnesota (http://www.grouplens.org). The dataset contains 1,000,000 ratings by 6,040 users for 3,592 movies. We only used the first 160,000 ratings in this experiment.

Both the rule-based and the CBR-based prediction strategies use the same set of prediction techniques. For details on the used techniques and the rule-based strategies, we refer to [17] for the TiV dataset and to [16] for the MovieLens set.

4.1 Accuracy Measure

As a measure for the accuracy of the different prediction strategies, we use the accuracy of the resulting predictions; better prediction strategies result in better predictions. As an accuracy measure we use a combination of the often-used mean absolute error (mae) and coverage. Herlocker [6] compared mae, coverage and other possible accuracy measures for recommender systems, including measures like precision, recall and ROC curves, and concludes that mae is a good choice for systems that generate predictions per item, like our hybrid recommender systems.

We combine mae and coverage because in many systems, such as TV systems, it is not only important that a good prediction can be made, measured by mae, it is also important that a prediction can actually be made, measured by coverage. Most TV programs are only available at the moment of broadcast. For this reason, we combine both measures into the global mean absolute error (gmae); it is the same as mae, but when a prediction technique or strategy cannot make a prediction the default value zero is assumed as the prediction. The lower the gmae the better the accuracy[1].

---

[1] Accuracy presented in this paper is based on a rating scale from -1 to +1, where -1 indicates a very strong negative interest and +1 a very strong positive interest; 0 indicates a neutral or indifferent interest.

To evaluate the accuracy of the prediction strategies throughout the usage period of the systems, we divided the set of ratings in each dataset into different 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. At the end also the overall accuracy over the whole dataset was calculated. The TiV dataset was divided into four sets, one set per week (with 8867, 8843, 6890, and 6768 ratings respectively). Since there is no logical unit to divide MovieLens, like the weeks in TiV, the MovieLens dataset was divided into 16 sets of 10,000 ratings each. X-validation was chosen over randomly drawn train and test sets as x-validation represents the order in which users actually accessed and rated the information.

4.2 Performed Experiments

First, the TiV dataset was used to experiment with different combinations of distance measures, distance to similarity conversion functions and case selection functions. We used all four distance measures, ued, wed, mms and msd, in combination with the three different distance to similarity functions, linear, sigmoidal and inverted, using different parameter values. We also experimented with the cosine similarity function. As case selection functions we used: all cases, only the most similar two, only the most similar three, the eldest most similar case, the newest most similar case, cases with similarity > 0.30, cases with similarity > 0.50, and cases with similarity > 0.70.

While exploring different similarity functions, we quickly discovered that steeply descending similarity functions provide the best results, e.g. *inverted*, *linear(d, m)* and *sigmoidal(d, k, m)* where $m < 4$. With such small distance ranges, especially when combined with a threshold value, both inverted and sigmoidal functions can easily be approximated by computationally simpler linear functions. Of all distance measures, ued is always one of the best, no matter what distance to similarity functions we use. Of the different distance to similarity conversion functions, a linear function with a low value for m performs best. In order to determine what value to use for m, we ran several other simulations resulting in $m = 2$ and sim > 0.50 providing the best results.

This best combination was then used in the CBR-based prediction strategy that was compared with the manually created rule-based prediction strategy. Furthermore, we examined the impact of using limited case-base sizes on the accuracy of the CBR-based strategy. Finally, the results were cross-validated with the MovieLens dataset.

5 Results

To determine whether CBR-based prediction strategies can provide more accurate predictions than manually created rule-based strategies we compared the results of both strategies. The rule-based strategy used was the best performing strategy determined in previous experiments [17]. The CBR-based strategy used the best performing combination of distance measure, distance to similarity function and case selection method: ued, *linear(d, 2)* and sim > 0.50. The results are listed in Table 1.

The results show that the CBR-based prediction strategy outperforms the manually created rule-based strategy, except in the first week. This is caused by the fact that

CBR needs time to build up case-bases; without enough cases, the strategy has too little data to make good decisions. But even in the first week, the CBR-based strategy still outperforms the best individual prediction technique. The increased accuracy is statistically significant (using a paired samples T-test and 95% confidence).

Table 1. Results of CBR-based versus rule-based strategy (bold indicates the best predictor)

	Week 1	Week 2	Week 3	Week 4
Rules-Based Strategy	**0.2935**	0.1774	0.1934	0.1602
CBR-Based Strategy	0.3031	**0.1710**	**0.1861**	**0.1535**
Best Technique	0.3530	0.1899	0.2171	0.1693

5.1 Impact of Limiting Case-Base Size

In order to determine the impact of limiting the size of the case-bases, which improves the speed and scalability of a CBR-based strategy, we experimented with several sizes. The results are shown in Table 2.

Table 2. Results of limiting the case-base size (bold indicates better predictions than no limit)

Case-base size	Week 1	Week 2	Week 3	Week 4
No limit	0.3031	0.1710	0.1861	0.1535
1000	0.3056	0.1784	0.1965	0.1593
2500	0.3047	0.1772	0.1917	0.1614
5000	**0.3029**	0.1743	0.1881	0.1568
7500	0.3043	0.1728	**0.1842**	0.1535
10000	0.3043	0.1720	**0.1831**	**0.1523**
12500	0.3043	**0.1708**	**0.1830**	**0.1506**
15000	0.3043	0.1713	**0.1841**	**0.1502**
17500	0.3043	0.1712	**0.1853**	**0.1501**

These results show that using a limited case-base can further improve the accuracy. We hypothesize that the removal of old cases made the strategy more accurate, since these cases represented old situations that did not occur again in the system. Furthermore, some prediction techniques behave differently early on in a system than they do later on, even under the same conditions according to the validity indicators. For example, in two situations A and B, a validity indicator of social filtering indicates that there are 40 similar users that have rated the item for which a prediction is necessary; however, in situation A, early in the system, the similarity of these 40 users is based on less rated items by each user than in the later situation B; hence the probability that social filtering provides an accurate prediction is higher in situation B than in A.

However, the improved effect of limited case-base sizes may also be influenced by the time-based characteristic of TiV. Because all users started using TiV at the beginning of week 1, no special situations like a new user occurred in later weeks. On the other hand, there is one special occasion between week 2 and 3: at that time almost all channels changed their programming drastically because at that time the new TV season started; this makes existing users similar to new users. The limited case-base size did not have any negative effects on the accuracy, on the contrary, accuracy increased more with limited case-base sizes after the second week than with unlimited sizes.

5.2 Cross-Validation

Cross-validation in the MovieLens dataset (see Table 3) shows that the CBR-based strategy does perform well in MovieLens, although not as well as in the TiV dataset. Sometimes, the rule-based strategy still out-performs the CBR-based strategy; however, even in those situations the CBR-based strategy is still better than the best individual prediction technique, making it an adequate prediction strategy.

Table 3. Results of the cross-validation of CBR-based versus rule–based strategy in the MovieLens dataset (bold indicates the best predictor in that set)

Set	10000	20000	30000	40000	50000	60000	70000	80000
Rule-Based	**0.3830**	**0.3953**	0.3880	0.4006	**0.3889**	**0.3803**	0.3973	0.3667
CBR-Based	0.3887	0.3968	**0.3830**	**0.3904**	0.3912	0.3815	**0.3954**	**0.3638**
Best Technique	0.3835	0.4050	0.3911	0.3966	0.3932	0.3820	0.4033	0.3678
Set	90000	100000	110000	120000	130000	140000	150000	160000
Rule-Based	0.3786	0.3737	**0.3688**	**0.3943**	**0.3924**	0.3787	0.3670	**0.3591**
CBR-Based	**0.3780**	**0.3721**	0.3714	0.3961	0.3963	**0.3782**	**0.3668**	0.3625
Best Technique	0.3826	0.3782	0.3715	0.3970	0.3950	0.3835	0.3713	0.3622

We have formulated two hypotheses for the lesser performance of the CBR-based strategy in MovieLens. The first is that the rule-based strategy created for the TiV dataset was not as good as the rule-based strategy created for the MovieLens dataset. Because both rule-based strategies have been designed by the same experts and several tuning experiments have been performed to optimize the rule sets in both strategies, we believe this hypothesis to be invalid.

Table 4. Results of the cross validation of limiting the case-base size in the MovieLens dataset (bold indicates better predictions than no limit)

Set	10000	20000	30000	40000	50000	60000	70000	80000
No limit	0.3887	0.3968	0.3830	0.3904	0.3912	0.3815	0.3954	0.3638
CB Size 12500	0.3887	**0.3963**	**0.3829**	0.3904	0.3944	0.3822	0.3964	0.3669
CB Size 25000	0.4057	0.4235	**0.3827**	0.3917	0.3924	0.3823	0.3963	0.3650
CBSize 50000	0.3887	0.3968	0.3830	0.3904	0.3912	0.3815	**0.3951**	**0.3634**
CB Size 100000	0.3887	0.3968	0.3830	0.3904	0.3912	0.3815	0.3954	0.3638
Set	90000	100000	110000	120000	130000	140000	150000	160000
No limit	0.3780	0.3721	0.3714	0.3961	0.3963	0.3782	0.3668	0.3625
CB Size 12500	0.3799	0.3778	0.3728	0.3997	0.3990	0.3813	0.3719	0.3667
CB Size 25000	0.3790	0.3747	0.3715	0.3966	0.3972	0.3789	0.3701	0.3668
CB Size 50000	0.3784	0.3729	0.3717	0.3967	**0.3961**	0.3790	0.3673	0.3648
CB Size 100000	0.3780	0.3721	0.3716	0.3964	0.3965	0.3785	0.3669	0.3630

The second hypothesis has to do with the observation that the prediction techniques for the TiV dataset have a much higher spread in their accuracy than the techniques for the MovieLens dataset. Spread is defined as the difference between the accuracy of the best performing technique and the accuracy of the worst performing technique. The average spread over the 16 validation sets in MovieLens is 0.0666, while the average spread over the four weeks in TiV is 0.3076. This means that the expected errors calculated by the CBR-based prediction strategy in the MovieLens dataset tend to be situated close together, making the probability of a wrong decision larger as the decisions are based on these expected errors. Since the spread is higher in the TiV dataset, the probability of a wrong decision is smaller.

Limiting the size of the case-base in MovieLens resulted in a decreased accuracy for the CBR-based prediction strategy (see Table 4). Only in a few situations did accuracy improve slightly, for example when using a size of 50000; in other subsets using the same case-base size accuracy decreases again.

We believe that the small spread of accuracy in the prediction techniques of MovieLens is also the reason why limiting the case-base size results in such different results. In order to confirm the influence of the spread of accuracy in prediction techniques, additional research is necessary; either experimenting with two different datasets and prediction strategies that show similar spread patterns or by developing one or two prediction techniques for MovieLens that increases the spread in accuracy.

6 Conclusions

In this paper, we determined whether CBR could provide more accurate prediction strategies than manually created rule-based prediction strategies. Experiments have shown that in systems where prediction techniques have a large spread in accuracy, CBR can indeed provide more accurate prediction strategies. However, in systems where the prediction techniques have a small spread in accuracy, the accuracy of a CBR-based strategy becomes more unreliable; sometimes accuracy is better than the rule-based strategy, sometimes worse. However, even with a small spread, a CBR-based strategy still outperforms the best individual prediction technique.

One of the main benefits of using CBR instead of the manually created rules in rule-based prediction strategies is that no expert knowledge is required to create the prediction strategies. A downside of using CBR as a prediction strategy is the performance and scalability penalty of CBR. Rule-based systems are very fast and scalable because they are model based. Since CBR is memory based, the speed of CBR-based strategies depends on the size of the case-bases.

In some systems, limiting the size of the case-bases not only improves speed and makes the system more scalable, it can also improve the accuracy of a CBR-based prediction strategy. However, more research is needed to determine under which conditions improved accuracy can be expected with limited case-base sizes.

All things considered, we conclude that CBR can indeed be used to create prediction strategies for hybrid recommender systems. However, one must be aware of the conditions under which CBR will increase accuracy over manually created rule-based prediction strategies.

Acknowledgments

This research is part of the PhD project Duine (http://duine.telin.nl) at the Telematica Instituut (http://www.telin.nl) and the Freeband projects Xhome and WASP (http://www.freeband.nl). The authors like to thank Mannes Poel and Erik Boertjes for their comments, support and help in this research. Also thanks to the researchers at the university of Minnesota for making the MovieLens datasets publicly available.

References

1. Ardissono, L., Gena, C., Torasso, P., Bellifemine, F., Chiarotto, A., Difino, A., Negro, B.: User Modeling and Recommendation Techniques for Personalized Electronic Program Guides. In: Ardissono, L., Maybury, M. T., Kobsa, A. (eds.): Personalization and User-Adaptive Interaction in Digital TV. Kluwer (to appear)
2. Bennett, P. N., Dumais, S. T., Horvitz, E.: Probabilistic Combination of Text Classifiers Using Reliability Indicators: Models and Results. In: Proceedings of SIGIR'02 (Tampere, Finland). ACM (2002) 207-214
3. Billsus, D., Pazzani, M. J., Chen, J.: A Learning Agent for Wireless News Access. In: Proceedings of IUI 2000 (New Orleans, USA). ACM (2000) 33-36
4. Buczak, A. L., Zimmerman, J., Kurapati, K.: Personalization: Improving Ease-of-Use, Trust and Accuracy of a TV Show Recommender. In: Proceedings of the Workshop Personalization in Future TV'02 (Malaga, Spain). 3-12
5. Burke, R.: Hybrid Recommender Systems: Survey and Experiments. User Modeling and User-Adapted Interaction 12, 4 (2002) 331-370
6. Herlocker, J.: Understanding and Improving Automated Collaborative Filtering Systems. University of Minnesota (2000)
7. Herlocker, J., Konstan, J. A.: Content-Independent Task-Focused Recommendation. IEEE Internet Computing 5, 6 (2001) 40-47
8. Mendes, E., Mosley, N., Watson, I.: A Comparison of CBR Approaches to Web Hypermedia Project Cost Estimation. In: Proceedings of WWW2002 (Honolulu, Hawaii, USA). ACM (2002) 272-280
9. Mitchell, T. M.: Machine learning. McGraw-Hill, New York, USA (1997)
10. Riesbeck, C. K., Schank, R.: Inside CBR. Lawrence Erlbaum Associates, Northvale, NJ, USA (1989)
11. Salton, G., McGill, M.: Introduction to Modern Information Retrieval. McGraw-Hill, New York, USA (1983)
12. Shardanand, U., Maes, P.: Social information filtering: algorithms for automated "Word of Mouth". In: Proceedings of Human factors in computing systems 1995 (New York, USA). ACM (1995) 210-217
13. Smyth, B., Cotter, P.: A personalised TV listings service for the digital TV age. Knowledge-Based Systems 13, 2-3 (2000) 53-59
14. Toyama, K., Horvitz, E.: Bayesian Modality Fusion: Probabilistic Integration of Multiple Vision Algorithms for Head Tracking. In: Proceedings of ACCV 2000, Fourth Asian Conference on Computer Vision (2000)
15. Van Setten, M.: Experiments with a recommendation technique that learns category interests. In: Proceedings of IADIS WWW/Internet 2002 (Lisabon, Portugal). 722-725
16. Van Setten, M., Veenstra, M., Nijholt, A.: Prediction Strategies: Combining Prediction Techniques to Optimize Personalisation. In: Proceedings of the Workshop Personalization in Future TV'02 (Malaga, Spain). 23-32
17. Van Setten, M., Veenstra, M., Nijholt, A., van Dijk, B.: Prediction Strategies in a TV Recommender System: Framework and Experiments. In: Proceedings of IADIS WWW/Internet 2003 (Algarve, Portugal). 203-210
18. Watson, I. D.: Applying CBR: Techniques for Enterprise Systems. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1997)