Addressing Uncertainty in Implicit Preferences

A Case-Study in TV Program Recommendations

Sandra Gadanho
Applications Research
Motorola Labs
Basingstoke, U.K.
sandra.gadanho@motorola.com

Nicolas Lhuillier
Applications Research
Motorola Labs
Gif-sur-Yvette, France
nicolas.lhuillier@motorola.com

ABSTRACT

The increasing amount of content available via digital television has made TV program recommenders valuable tools. In order to provide personalized recommendations, recommender systems need to collect information about user preferences. Since users are reluctant to invest much time in explicitly expressing their interests, preferences often need to be implicitly inferred through data gathered by monitoring user behavior. Which is, alas, less reliable.

This article addresses the problem of learning TV preferences based on tracking the programs users have watched, whilst dealing with the varying degrees of reliability in such information. Three approaches to the problem are discussed: use all information equally; weight information by its reliability or simply discard the most unreliable information.

Experimental results for these three approaches are presented and compared using a content-based filtering recommender built on a Naïve Bayes classifier.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; I.2 [**Artificial Intelligence**]: Learning

General Terms

Algorithms

Keywords

Recommender, Reliability, Implicit Preferences

1. INTRODUCTION

Digital television offers viewers with more content than one can reasonably consume. Profusion of content however is not at all a guarantee of user satisfaction if finding which content to watch becomes a tedious task and damages the user experience. Systems which deliver users with personalized recommendations are therefore seen as valuable tools to link users with content of potential interest to them [11]. The efficiency of recommender systems, however, greatly depends not only on good prediction algorithms but also on the amount and the accuracy of the user data available to train these algorithms. There exist basically two main user profiling strategies: explicit and implicit.

RecSys'07, October 19–20, 2007, Minneapolis, Minnesota, USA.
ACM 978-1-59593-730-8/07/0010.

Explicit user modeling requires users to manually provide information about their tastes. This may take various forms, from bootstrap questionnaires to "on-the-fly" feedback about content via a rating scale. Though seen as very accurate, the main problem with explicit preferences, in addition to varying interpretation of the same rating scale, is the extra burden they place on users who generally consider television to be a passive activity. As such, explicit profiles are often too scarce to be really useful because users for instance do not rate enough items. Additionally, such profiles do not conveniently evolve with time. The claimed accuracy of explicit preferences is also questionable as users have a psychological tendency to describe themselves more as what they would like to be than what they actually are.

Implicit user modeling, on the other hand, relies on data mining techniques to automatically extract user interests and preferences from their actions. Such a mechanism therefore does not impose any extra effort on users. However, due to their lack of user control, implicit preferences are often perceived as more intrusive and less accurate than explicit ones [8]. Indeed, it is often difficult to accurately map user actions into indications of preferences. However, thanks to greater amounts of data it allows to be collected which significantly increases the size of the training sets used by the prediction algorithms, recommendations based on implicit preferences can be at least as accurate as those relying on explicit user preferences [10].

In the end, the best system is likely to be a hybrid which uses both modeling strategies.

Implicit preferences, which are based on assumptions, inherently suffer from uncertainty. As quoted from Kelly and Teevan [6]: *"More tools that allow for the accurate and reliable collection of data [...] need to be developed"*. Indeed, the objective of this paper is to investigate the best practices to address such uncertainty. Three different approaches to deal with unreliable implicit preference inputs are therefore proposed and compared: (1) the first one does not perform any particular treatment to the data and will serve as reference; (2) the second approach weights preference inputs by their perceived confidence and (3) the last approach authoritatively discards preference inputs which do not ensure enough reliability.

The next section presents a literature review of implicit preferences used in TV recommender systems. Section 3 discusses the issues linked to uncertainties in implicit preference indicators and discusses the proposed strategies to address them. Section 4 compares experimental results for the proposed approaches to deal with uncertainty. The final section discusses these results and opens future work directions.

2. RELATED WORK

Implicit preferences have initially been applied to browsing the Internet [4]. Various forms of implicit modeling have been developed. Oard and Kim first [9], Kelly and Teevan later [6], proposed a classification of implicit feedback techniques along two axes: the type of user action (*e.g.*, examine, retain, annotate) and the scope of the action (*i.e.*, only a portion of the content, the entire object or even its class).

Implicit preferences have also already been used in the television domain with some success [1, 10, 2]. The commercial and popular TiVo system uses an explicit 7-point rating scale (ranging from 3 thumbs-down to 3 thumbs-up) to collect explicit user preferences, but it also considers any user recording of a program as a one thumb-up implicit rating [1]. TiVo researchers further point out that actions on TiVo's "Season's pass"[1] may be a good indicator of user preferences as well. Kim et al. [7] developed a slightly more complex approach which combines statistics about watched programs with sequences of user actions, such as: "Play → Fast Backward → Play" to create their model of user preferences.

Baudisch and Brueckner [2] took an interesting and original approach with TV Scout, a web-based recommendation system providing personalized TV schedules. This system continuously and unobtrusively gathers information about user interests from implicit feedback. However, in order to lower the entry barrier for new users ("cold-start" issue), the system initially presents itself as a retrieval system, so that all user effort leads to an immediate result. Eventually, when enough user preferences have been collected, it gradually evolves into a filtering system. Their live experiments strongly supported this phased approach.

Zimmerman et al. [12] evaluated an implicit recommender system with two different implicit methods: Bayesian statistics and Decision Trees. Their results showed that different methods perform better for different users or even for different types of shows, but fusing the results of the two methods improved the robustness of the system.

Regarding uncertainty in implicit preferences, Claypool et al. [4], developed the "Curious Browser", which has been designed to capture both explicit feedback and implicit preference indicators in order to evaluate which implicit indicators are correlated with the stated user interest. Their experiments showed that time spent on a web page is a reliable sign of user's interest. This was an interesting study in the domain of web browsing that would be worth replicating in the domain of TV recommendation.

3. ADDRESSING UNCERTAINTY

The implicit preference indicator we used in our TV recommender system is based on the amount of a given program the user has watched. Such information has been suggested as one of the best possible indicators for TV recommenders [1] and seems to be an appropriate alternative for the successful web-page reading time indicator. This is somehow different from previous work where, probably due to a lack of enough detailed information, only the fact that a program has been watched or not-watched is taken into account (*e.g.*, [12]). Bearing in mind that users do not have an exhaustive knowledge of the TV schedule, we think that the fact one user did not watch a particular program does not necessary indicate that this user does not like it. The user may simply not have been aware of the program. On the contrary, we believe that the fact a user watched a program only briefly is a strong indication of dislike, especially, if the user zapped to another channel.

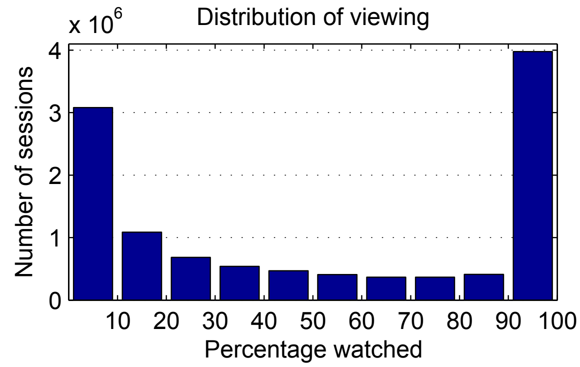

Figure 1: Distribution of the viewing sessions according to percentage of the program watched. This is based on six-months of viewing information for 5000 representative viewers.

The information of how much of a program a user has watched is often difficult to capture. In a normal set-top-box scenario, the basic information that is known is the channel that is on. From the EPG, the corresponding program can be identified with some accuracy (note that there is often some uncertainty associated with advertisement breaks). More difficult still is identifying which members of the household are actually viewing the program or even whether someone is actually watching the television or the television has been left on as background noise. The data[2] we use in our experiments already identifies the viewers and advertisement breaks and is therefore free from such issues. However, in real systems, these issues may also need to be addressed.

Despite the fact that user and program are correctly identified, uncertainty issues still remain. The fact that one of the users watched a program does not guarantee this user liked it. Did you never regret having watched a bad program? The basic assumption behind our implicit preference mechanism hence is that the amount of a program the user has watched depends on user's appreciation of the program. This postulation may sensibly and statistically prove valid for programs either almost entirely or hardly watched by users. Intermediate situations, however, are not so straightforward to correctly categorize. The distribution of the percentage of program viewed in our viewing data, demonstrates that the viewing patterns of users actually mostly fall within the most extreme and therefore more reliable cases (see Figure 1). This again suggests that this measure is potentially a very useful preference indicator. However, there is still about half of the viewing sessions that fall within the more uncertain categories.

[1]Allows to record all episodes from a series.

[2]See Section 4.1 for details.

To deal with this uncertainty, we suggest three strategies:

Neutral — One straightforward approach to the problem is to treat all examples equally and simply define a threshold to decide when a program should be considered as being liked by the user. The advantage is that it is computationally a simple approach which makes use of all information. It relies on the assumption that the underlying learning algorithm is able to cope with noisy information and that the more numerous correct examples available will help to contravene the negative impact of the less numerous uncertain examples.

Weighted — A slightly more expensive approach is to weight examples according to their perceived uncertainty so that more reliable examples have a higher impact on the learning algorithm than those associated with more uncertainty. This strategy has the advantage of still using all the information but not being as negatively affected by incorrect examples [5].

Selective — Since there is usually quite a lot of implicit information available, it might be reasonable to simply discard the most unreliable information. This is actually the most computationally-effective approach since the underlying learning algorithm is only required to learn on a subset of the total available examples. However, the risk with this strategy is that it may discard information the algorithm may never have the opportunity to learn about. For example, one user may be interested in music programs but never watches them in full. In such a case the recommender will never have the chance to learn these preferences because related viewing sessions will always be discarded.

The following section compares these different approaches via an empirical evaluation of the predictions and recommendations that can be obtained with each one of them.

4. EXPERIMENTS

This section presents the different experiments made to test the approaches proposed to deal with uncertainty in implicit preferences. This section first describes the dataset and recommender algorithm used. Next it describes in detail how the different approaches to uncertainty were implemented. This is followed by the description of the evaluation method and finally the results obtained in the experiments.

4.1 Dataset

Considering the difficulty in collecting user data for test purposes [12], we opted in our experiments to use a BARB[3] dataset. BARB collects TV viewing data from a representative set of UK households in order to provide minute-by-minute estimates of the number of people watching the different terrestrial, cable and satellite broadcasting channels at any one time. Its main difference compared to standard viewing data collected on a set-top-box, is that viewing sessions identify the users that were actually watching TV.

Our BARB dataset comprises 6 months of viewing information, from January to June 2005, for approximately 3000 households (5000 individual viewers). This viewing information was combined with program schedule information for the same period to produce viewing sessions which indicate how much of a given program a given individual has watched. For performance reasons, experiments were conducted with a subset of 30 "typical" users (*i.e.*with no obvious abnormal viewing patterns like unusual amount of time spent watching TV or switching channels) selected for their tendency to watch TV alone (to limit the influence of others in their program choices). The idea is that these users can potentially provide more accurate implicit preferences. Each viewing session in the dataset describes the person, the genre of the program watched and the percentage of this program that has been watched.

Evaluating a recommender with such a dataset would mean that only programs watched by the user (however briefly) could be considered in the evaluation process. However, in a live system, the recommender has to select programs from the whole schedule, so it is important to make sure that programs that are not usually watched at all by the user are also handled correctly. For this reason, programs from the schedule which had not been viewed by the user were added to the dataset with no viewing information. Since the whole schedule is quite large, it was decided to only use a sample of the schedule consisting of 20 random programs per day. This results in special examples with no viewing information which are ignored during training and handled specially while testing.

4.2 Recommender algorithm

The recommender algorithm selected for the experiments follows a machine learning approach. A Naïve Bayes (NB) classifier is trained to learn the classification of the examples using a set of training examples taken from the dataset. Each example is derived from a TV viewing session and it is characterized by a genre and a class (matching/non-matching) attribute. Genre is a nominal attribute from BARB specification that takes 1 out of 18 possible categories and is used as the description of the program. The class attribute determines whether the item is of interest to the user or not. In other words, the NB classifier learns the probability that a program matches user's interest given its Genre.

For the evaluation a subset of testing examples (which are distinct from the training ones) is taken from the dataset. The recommender algorithm sorts these test items according the NB classifier's estimated probability of them matching user preferences. The top 100 items are selected for the recommendation shortlist. The reason for this limit is that we believe it is not sensible to present a larger recommendation list to the user. Furthermore and in order to ensure high precision in recommendations, elements with less than 80% of matching probability are discarded from the recommendation shortlist. This threshold is referred to as the *minimum confidence* of the recommender.

4.3 Different implicit-preference strategies

The interpretation of user behaviors provides implicit information about their preferences. Different assumptions can be taken to determine whether the user liked or disliked a program. These determine the examples' classification. As mentioned previously, the assumption behind our experiments is that users watch much of the programs they like and quickly skip those that they do not. However, this leaves the question of how much of the program constitutes an in-

[3]Broadcasters' Audience Research Board LTD.

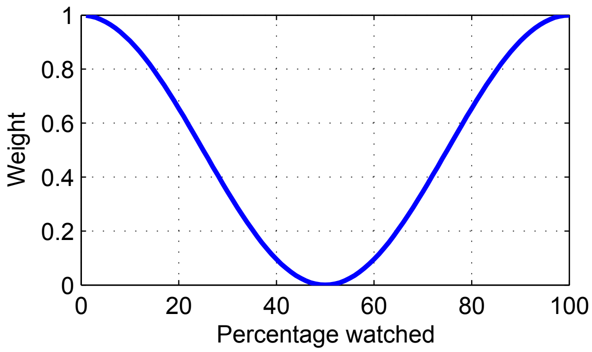

Figure 2: The weight function.

dication of preference or non-preference. Furthermore, different examples have different associated uncertainty. While a program watched in totality is likely a good indication of preference, a program only half-watched can be considered as ambiguous. The different approaches proposed take different strategies to handle this underlying uncertainty. They are described in detail next.

Regardless of the strategy chosen, examples with no viewing information, which have been randomly picked from the schedule are never classified.

4.3.1 Neutral

The Neutral strategy simply treat all viewing examples equally. It classifies an example as matching user preferences if the user watched the program for over 50% of its duration and it classifies it as non-matching otherwise.

4.3.2 Weighted

The Weighted strategy uses the same classification strategy as the Neutral strategy but it weights examples according to their uncertainty. Examples with high level of uncertainty are given a smaller weight so that they have less impact on the recommender's learning. For example, it takes two learning examples of 0.5 weight to influence the classifier's preference model as much as a single training example of weight 1. The training examples are weighted by a function (f) of the percentage watched (w):

$$f(w) = \cos(\frac{w\pi}{100})^2 \tag{1}$$

This function is illustrated in Figure 2.

4.3.3 Selective

The selective strategy simply discards examples that are too ambiguous. With this strategy, examples are classified such that:

- examples with percentage watched above or equal to 90% are classified as matching user preferences;
- examples with percentage watched less or equal to 10% are classified as NOT matching user preferences;
- other examples are left unclassified.

Similarly to the Neutral strategy, all training examples have the same learning impact (*i.e.*, same weight).

4.4 Evaluation method

The best way to evaluate these different approaches would be to have the feedback from users themselves. Unfortunately, this information was not available to us, as it is frequently the case.

Following a typical machine learning evaluation, the same 'classification scheme' would be used for both the training and the testing data. This would not be a concern since usually what is under evaluation is the classifier and the classification is a given. Here we are actually looking into different 'classification schemes' so, in a certain way, we are training the classifier for different problems. Using a typical approach would result in evaluating which one is the easiest problem to learn. Our preliminary experiments showed that the Selective classification is easier to learn than the Neutral classification. This indicates that extreme viewing behavior is easier to learn but does not say much about the recommendation quality of the system from a user satisfaction point of view.

Thus, our three approaches are first all evaluated using the *Neutral Classification* and then using the *Selective Classification*. As such, these two evaluations were respectively named Neutral and Selective. This allows comparing the approaches in equal grounds and avoiding a biased result that would be obtained by preferring one classification over the other.

For all experiments, 4 evaluation metrics were taken:

Accuracy — Measures the proportion of test examples correctly classified as matching or as not matching user preferences. This measures the accuracy of the NB classifier and does not take into account the notion of recommendation shortlist. Unclassified test examples are ignored by this measure.

Breese Score — Measures the sum of the probabilities of seeing items matching user preferences in the recommendation list. This metric was adapted from the original score suggested by Breese et al. [3]. The experiments use a half life value of ten[4].

Precision — Considering all the matching or non-matching items that were shortlisted by the recommender, it measures the proportion of those that actually match user preferences (unclassified items that were shortlisted are ignored).

Recall — Measures the proportion of item examples that match user preferences which were shortlisted.

In the evaluation process, the dataset is divided into 26 week periods. The recommender iteratively and cumulatively learns one week of data and is then tested against data of the following week. The first evaluation is performed without the recommender being trained and the last evaluation is tested against the last week which is never used for training.

All evaluation metrics are calculated individually for each user and each week. Individual user results are averaged to obtain final results. Graphs and tables also show the 95% confidence interval (assuming a Normal distribution) for each measure.

[4]This assumes there is a 50% chance of the user reaching item ten in the list.

Table 1: Summary of experimental results obtained using the Neutral and the Selective evaluations. The table shows the mean of the last 20 weekly evaluations.

Evaluation	Strategy	Accuracy (%)	Breese score	Precision	Recall
Neutral	Neutral	67.9 ± 2.5	0.58 ± 0.07	0.76 ± 0.09	0.12 ± 0.07
	Weighted	67.7 ± 2.5	0.58 ± 0.07	0.78 ± 0.06	0.21 ± 0.09
	Selective	67.1 ± 2.6	0.58 ± 0.07	0.74 ± 0.05	0.35 ± 0.09
	Baseline	50	—	0.48 ± 0.04	0.40 ± 0.02
Selective	Neutral	73.4 ± 3.2	0.48 ± 0.07	0.85 ± 0.07	0.13 ± 0.08
	Weighted	74.0 ± 3.0	0.49 ± 0.07	0.88 ± 0.05	0.23 ± 0.10
	Selective	73.9 ± 3.0	0.49 ± 0.07	0.85 ± 0.05	0.39 ± 0.10
	Baseline	50	—	0.55 ± 0.06	0.41 ± 0.02

Further to the results presented for each implicit preference strategy, results are also presented for a random recommender. The random recommender randomly selects one hundred items for recommendation and randomly classifies items as either matching or not matching user preferences. These latter results are used as baseline for our implicit learning mechanisms.

4.5 Results

The first set of experiments made uses the Neutral evaluation. It compares the performance of different approaches when the test examples are classified with the Neutral classification. This means the Neutral strategy should have a slight advantage since this was the problem it was trained for. Results for the three approaches are shown in Figure 3 and summarized in Table 1. Baseline results for a non-learning recommender system are also shown.

The second set of experiment uses the Selective evaluation to compare performances of the three approaches. In this case, the Selective approach is expected to benefit from a slight advantage. Table 1 also shows this second set of results.

Results obtained with the two different evaluation approaches are consistent. All the implicit preferences strategies are viable in that they all provide a significant improvement towards the baseline recommender (random recommender). Note that the random recommender has a very good recall evaluation compared to the trained recommenders. This is explained by the fact that the random recommender always suggests one hundred recommendations while the trained recommenders restrict their shortlists to items with a high confidence value, which often results in shorter lists. Everything else being equal shorter lists will have a smaller proportion of total number of matching items. However by using shorter lists, the learning recommenders can ensure much better precision which is usually much more important than recall.

All three strategies are very similar in terms of accuracy, Breese score and precision. Their evaluation only differs significantly in terms of recall. The Selective approach recommends a higher proportion of matching items whilst the Neutral Evaluation recommends a smaller proportion.

This suggests that the Selective approach is more useful in that it can generate more recommendations with the same level of precision. Note that the discriminative level of a recommender can be adjusted by the choice of minimum confidence value required for short-listing an item for recommendation. So varying this value can potentially change the differences in performance between the different approaches. For understanding the influence of the minimum confidence values, the graphs in Figures 4, 5 and 6 show an analysis of the use of different minimum confidence values. This analysis is based on the Selective evaluation, but similar results are obtained using the Neutral evaluation.

The first graph shows recall and precision results for different values of the minimum confidence. Results are shown both for the Neutral and Selective approaches. This analysis is consistent with the results presented previously: it shows that recall for the Selective approach is always better or identical compared to the Neutral approach and that precision values are similar[5].

The last couple of graphs show the distribution of items according to their estimated confidence value when learning with the Neutral strategy (see Figure 5) and with the Selective strategy (see Figure 6). Both show the distribution of items matching and not matching user's preferences according to the classifier's confidence in recommending the item (*i.e.*, the probability of this item matching user's preferences according to the classifier). The distribution of unclassified items is also shown. These are the items that either do not have any viewing information or which were left unclassified by the Selective classification. Graphs show the percentage of programs in each classification category that fall within the different confidences ranges. A mean is taken of the values calculated for the recommendation outputs of all users during the latest 4 weeks.

An ideal recommender should have all matching items with high confidence values (matching bars towards the right in the histogram) and all non-matching items with low confidence values (non-matching bars towards the left in the histogram). The true value of unclassified items being unknown, a uniform distribution (or a concentration around middle confidence values) is expected. Arguably, these items should neither be simply discarded nor preferred over all other items. This would mean that the recommender (1) was not giving preference to items the user has shown clear disinterest over items that the user has not seen at all and (2) was not preferring items not seen by the user over items that the user has shown interest in.

The recommenders shown in the graph have a reasonable distribution of the items according to confidence. Matching items have a tendency to have higher confidence values and

[5]Note that neither accuracy nor the Breese score were affected by the minimum confidence value.

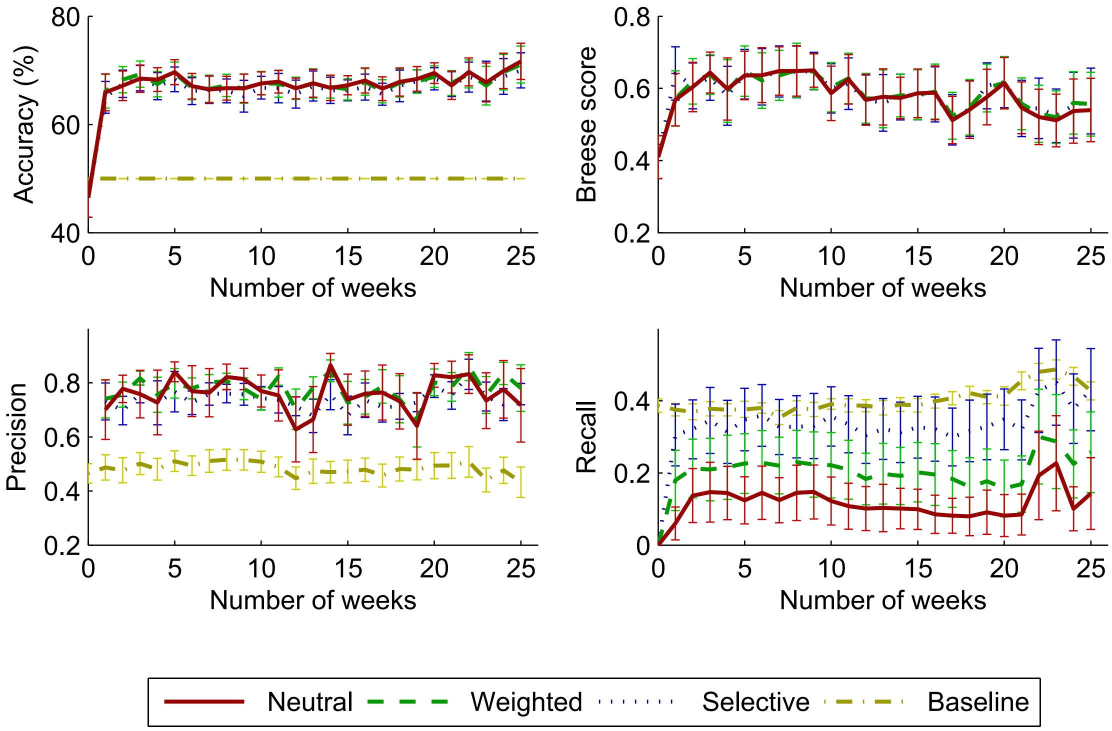

Figure 3: Experimental results obtained using the Neutral evaluation.

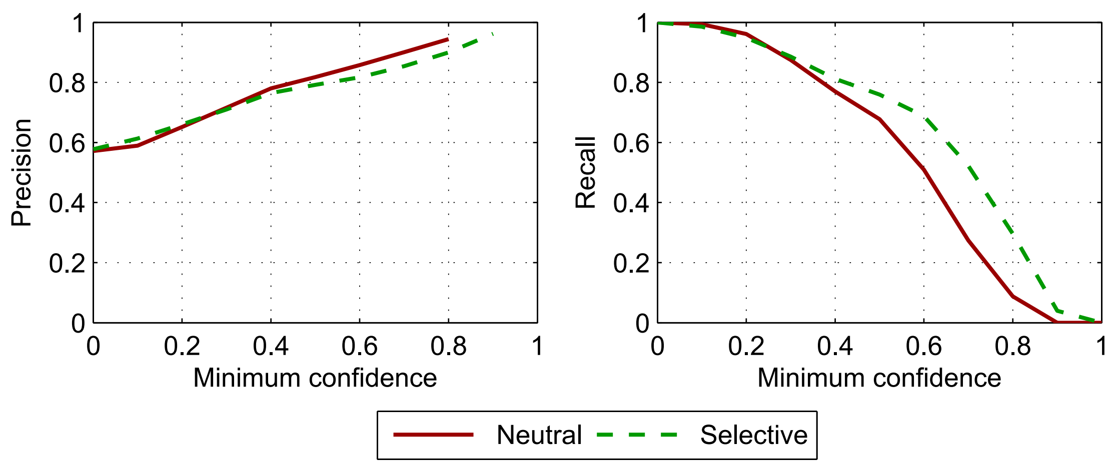

Figure 4: The values of precision and recall depending on the minimum confidence set for recommending an item with no limit on list size.

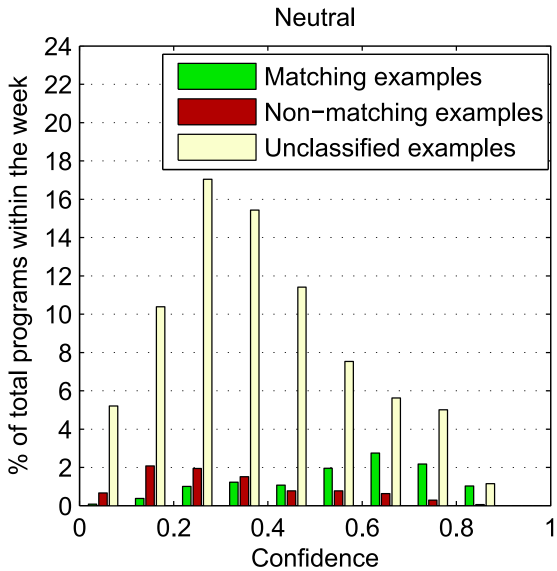

Figure 5: The distribution of sessions according to the Neutral classifier's confidence in selecting them for recommendation.

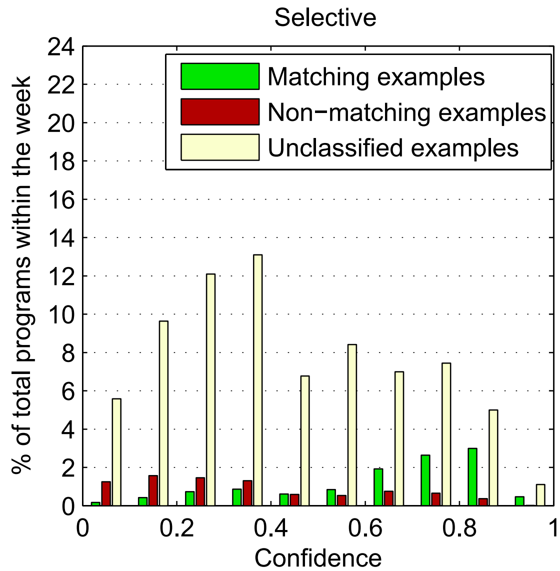

Figure 6: The distribution of sessions according to the Selective classifier's confidence in selecting them for recommendation.

non-matching items have a tendency to have smaller confidence values. Non-classified items are neither preferred nor unfavoured. The graphs suggest though that the Selective approach is better at attributing higher confidence values to matching items, thus confirming this approach as advantageous.

As a final point, note that if the recommenders had been evaluated under their own classification scheme, then the Selective approach would have a clear advantage towards the Neutral approach in term of accuracy (73.9 *vs.* 67.9), precision (0.85 *vs.* 0.76) and recall (0.39 *vs.* 0.12). This comparison is much more beneficial to the Selective approach than the ones we use confirming our preliminary assertion that this comparison is not really fair.

5. CONCLUSIONS

The results of our implicit preference experiments are first a supplementary proof, if ever required, of the validity of implicit preferences for TV program recommendations. The precision of 88% for the Weighted strategy for instance indicates that theoretically, on average, only 1 out of 10 recommendations would not meet user's preferences, which is a very good score. Taking into account that the prediction algorithm was a simple Naïve Bayes using only the Genre metadata as parameter, we believe there is additional space for improvement by using information such as time or broadcast channel in addition.

Further to the point, results suggest that either the Weighted or Selective approaches are better than the Neutral approach. With respect to which one of the Weighted or the Selective approaches works best to address uncertainty, we feel that our experimental results are not entirely clear. This decision also needs to be motivated by the use-case. On the one hand, results suggest that the Weighted approach may give a slightly better precision score which is very important as the cost of making a wrong recommendation is often high. On the other hand, the Selective approach misses less recommendation opportunities (*i.e.*, has a significantly better recall) and is the most resource-effective (it relies on less learning examples to learn and does not need the extra weighting of the examples). From a strict observation of our results and taking into account that precision values in our experiments are statistically identical, the Selective approach is better. However, we think there may exist scenarios for which the Weighted approach will be preferable.

The main difficulty we found with our experiments relied in the evaluation methodology. All our metrics remain theoretical since the assumptions used to test recommendations are the same as the ones used to infer the user preferences. Future experiments shall be designed so that users are directly able to evaluate the recommendation lists or even their learnt preferences. This should also allow comparing implicit recommendations with others built on explicit user preferences. In the meanwhile, we proposed some solutions for experiments with implicit preferences. These may be useful when explicit preferences are not available as it is often the case. These included, using different implicit approaches in the testing phase, adding in examples not covered in the implicit preference examples (*i.e.*, in our case using examples from the full schedule and not being restricted to those programs actually watched by the users) and different visualization techniques.

Bearing in mind that our BARB dataset is not strictly identical to data that can be logged on a commercial set-top-box, future work will also aim at defining filtering strategies to remove noise from collected data and evaluate the impact of having multiple users using the same implicit recommendation system.

6. REFERENCES

[1] K. Ali and W. van Stam. TiVo: Making show recommendations using a distributed collaborative filtering architecture. In *Proceedings of the 2004 ACM*

SIGKDD international conference on Knowledge discovery and data mining, pages 394–401, New York, USA, August 2004. ACM Press.

[2] P. Baudisch and L. Brueckner. TV scout: Lowering the entry barrier to personalized TV program recommendation. In *Proceedings of the* 2^{nd} *International Conference on Adaptive Hypermedia and Adaptive Web Based Systems (AH'2002)*, volume 2347 of *Lecture Notes in Computer Science*, pages 58–67, Malaga, Spain, May 2002. Springer-Verlag Heidelberg.

[3] J. S. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *In Proceedings of the* 14^{th} *Conference on Uncertainty in Artificial Intelligence*, pages 43–52, 1998.

[4] M. Claypool, P. Le, M. Wased, and D. Brown. Implicit interest indicators. In *IUI '01: Proceedings of the* 6^{th} *international conference on Intelligent user interfaces*, pages 33–40, New York, NY, USA, 2001. ACM Press.

[5] S. Gadanho. TV recommendations based on implicit information. In *ECAI'06 workshop on Recommender Systems*, 2006.

[6] D. Kelly and J. Teevan. Implicit feedback for inferring user preference: a bibliography. *SIGIR Forum*, 37(2):18–28, 2003.

[7] M. Kim, J. Lim, K. Kang, and J. Kim. Agent-based intelligent multimedia broadcasting within mpeg-21 multimedia framework. *ETRI Journal*, 26(2):136–148, 2004.

[8] D. M. Nichols. Implicit rating and filtering. In *Proceedings of* 5^{th} *DELOS Workshop on Filtering and Collaborative Filtering*, pages 31–36. ERCIM, 1998.

[9] D. W. Oard and J. Kim. Modeling information content using observable behavior. In *Proceedings of the* 64^{th} *Annual Meeting of the American Society for Information Science and Technology*, pages 38–45, 2001.

[10] D. O'Sullivan, B. Smyth, D. Wilson, K. M. Donald, and A. F. Smeaton. Interactive television personalisation. From guides to programmes. In *Personalized Digital Television: Targeting Programs to Individual Viewers*, volume 6 of *Human-Computer Interaction Series*, chapter 2, pages 73–91. Kluwer Academic Publishers, 2004.

[11] B. Smyth and P. Cotter. Personalized electronic program guides for digital TV. *AI Magazine*, 22(2):89–98, 2001.

[12] J. Zimmermana, K. Kurapati, A. L. Buczak, D. Schaffer, S. Gutta, and J. Martino. TV personalization system. Design of a TV show recommender engine and interface. In *Personalized Digital Television: Targeting Programs to Individual Viewers*, volume 6 of *Human-Computer Interaction Series*, chapter 2, pages 27–51. Kluwer Academic Publishers, 2004.