

Recommendation of Television Programs

André Pinção Lucas

Instituto Superior Técnico, INESC-ID,
Av. Professor Cavaco Silva, 2744-016 Porto Salvo, Portugal
`andre.lucas@ist.utl.pt`

Abstract. The growing volume of TV shows offered by networks makes harder for viewers to select relevant content. In this setting, recommendation systems are among the possible solutions, although their actual application to television shows remains relatively unexplored. In my MSc project, I studied the television show recommendation problem, specifically through hybrid approaches combining collaborative filtering and techniques based on similarities in the shows' descriptive metadata. Experimental results based on data extracted from the Web confirm the effectiveness of the proposed methods, thus showing that a combination of different approaches may result in an improvement of around 7% compared to the best collaborative techniques.

Key words: Hybrid Recommendation Systems, Learning-to-Rank, Television Show Recommendation

1 Introduction

The current volume of television shows offered by networks makes hard for viewers to select relevant content. Traditional television content search tools, e.g. printed or electronic TV guides, are extensive and do not efficiently meet the viewers' current information needs [14]. The problem is getting harder, due to the growing amount of contents brought on by the generalization of digital television.

In this setting, recommendation systems are among the likely solutions. These systems assist users in searching relevant content according to their interests, through explicit feedback and through the monitoring of past user behavior [1].

Despite the growing popularity of such systems, their actual application in the context of recommending television shows remains relatively unexplored in the related literature. In this paper, I present a study on the problem of television show recommendation through content-based systems, collaborative filtering techniques and hybrid approaches combining both types of techniques. In addition, I analyzed the use of machine learning in order to combine the results of various recommendation algorithms, effectively learning how to rank TV shows so that they might be recommended according to their relevance to users' interests. To evaluate the various techniques, I developed a prototype recommendation system using the Duine Framework [29].

In order to compare the studied techniques, I began by collecting and statistically defining a test data set, given that no specific standard data set

exists for TV shows. This data set was built with basis on publicly available information from the Web, through the following social networks where users rate television shows: TV.com¹, IMDb² and Living Social³.

The remaining content in this article is organized as follows. Section 2 presents basic concepts in the field of recommendation systems, and reviews the main related papers. Section 3 describes basic prediction techniques employed in this paper, based on content analysis and collaborative filtering. Section 4 describes the different hybrid approaches tested. The data set is described in Section 5. Section 6 presents both the evaluation methodology and the results obtained, along with a discussion of the latter. Finally, Section 7 summarizes the main conclusions of the paper and points possible paths for future research.

2 Concepts and Related Work

Generally speaking, recommendation systems are information systems with the ability to recommend potentially interesting items to users, employing artificial intelligence techniques in order to predict the users' interest in items based on their behavior patterns and on similarities between items and/or users [1].

According to the reading of [1], recommendation systems can be typically classified along the following categories:

- **Content-based systems** – These recommend items with features similar to others for which the user has shown interest for in the past. The similarities between items are measured, in most cases, through information retrieval techniques [19]. According to [23], content-based recommendation systems mostly rely on user profiles, which fit into one of two types:
 - **A history of user interactions with the system**, which may include the set of items that a user accessed in the past [10], or items that the user explicitly reviewed [20]. These histories may be used directly (i.e., memory-based systems), for instance through a nearest-neighbors approach applied to items [6], or indirectly through the creation of a classification model employing supervised machine learning techniques (i.e., model-based systems), such as the Naive Bayes classifiers [22].
 - **A user preference model**, i.e. a description of the types of items the user is interested in. There are several possible options for this description, but among the most common is a function that, for each item, returns an estimate of the user's probability of being interested in it [23].
- **Collaborative filtering systems** – Based on the items' user ratings, these systems infer similarities between users or items and then recommend items potentially of interest to users [12]. These are the most commonly used recommendation systems, despite their requirement of a vast amount of users with a high participation rate in order to provide useful recommendations. Some studies, such as [1] and [24], reviewed various collaborative filtering systems. These systems are subdivided into the following subtypes:

¹<http://www.tv.com/>

²<http://www.imdb.com/>

³<http://tv.livingsocial.com/>

- **User-to-user collaborative filtering** – Based on users’ preferences, items preferred by users with similar preferences are recommended [31].
- **Item-to-item collaborative filtering** – Based on users’ preferences, item similarity is inferred (e.g., similarities in item ratings) and items similar to those that users preferred in the past are suggested [31].
- **Hybrid systems** – Hybrid systems combine collaborative and content based techniques in order to improve performance [28] and fill in some gaps in classic approaches [1]. In [7] different hybrid system application methods are presented, the most common of which as follows:
 - **Weighted** – The recommendations from each technique are weighted and combined in order to produce a single set of recommendations. In [9] content-based and collaborative filtering components are combined through a four-layer Bayesian network, containing item features, items, users and a layer of user relationships.
 - **Switching** – For each recommendation offered, only one technique is used, depending on the current context.
 - **Mixed** – Recommendations suggested by various techniques are presented to the user simultaneously, that is to say, they are jointly presented.
 - **Cascade** – One recommendation technique refines another’s recommendations. An instance of this approach is the RACOFI framework, in which recommendations are first computed by the collaborative filtering component – named Cofi - and then used as an input for RALOCA, a rule-based content-based system, which refines recommendations [3].
 - **Meta-level** – The model corresponding to a recommendation technique is used to provide the input data to another technique. The Fab system is a meta-level hybrid recommender system, aiming to suggest Web pages according to user preferences [4]. It includes collection agents associated to interest clusters that employ user profile models maintained by selection agents to carry out its search operations for new pages relevant to the interest topic they represent, through collaborative filtering techniques.

Recommendation systems are also classified according to the technique employed for analyzing the data on which the recommendations are based. These systems fall into one of two general groups of techniques:

- **Memory-based techniques** – Also known as heuristic techniques [1], they rely on heuristics and make predictions based on the full collection of previously user-rated items, inferring which items may potentially interest users [28].
- **Model-based techniques** – These employ the collection of user ratings to learn a model offline, later used to make online recommendations [1]. Methods such as Bayesian networks [9] or Boltzmann machines [25] are frequently used.

Ratings may be explicit (i.e., directly offered by users) or implicit (i.e., the system deduces ratings by monitoring users’ activities). These ratings allow a calculation of similarities between items and/or users [28]. There are several ways to compute these similarities, although the Pearson correlation and the cosine measure are the most frequently employed.

Digital television opened up possibilities for interactive television environments that allow system customization, as per the user's preferences. So far, however, few applications for recommendation systems in this field exist.

In these systems, implicit ratings are more commonly used, for instance by recording a show's view (e.g., in TV Genius⁴) or through preset tapings [30]. In this field, items are generally very information-rich, which facilitates the use of content-based techniques. TiVo, a popular item-to-item recommendation system [2], TV Genius, Lighthouse [30] and Dynamic TV [27] are four instances of recommendation systems applied to the television domain.

Each paper mentioned in the preceding paragraph devised a specific recommendation strategy and, in this specific application domain there is a lack of a detailed analysis of different recommendation techniques. This paper proposes to carry out that analysis and present new feasible approaches, namely through the use of machine learning techniques.

3 Recommendation techniques employed

This section describes several recommendation techniques employed in this paper, both content- and collaborative filtering-based. Section 3.1 begins by describing some basic prediction techniques that rely on a single prediction variable (e.g., user or item). Content-based techniques relying on programs description will appear in Section 3.2, and lastly the collaborative filtering techniques that rely on crossing user and item data are presented in Section 3.3.

3.1 Basic prediction techniques

Two basic and more efficient techniques may be used to predict ratings based on aggregate data alone, thus avoiding potential privacy concerns. They are: user rating averages and item rating averages. The first measure consists in computing the average of the user's ratings; the second is the average of ratings awarded to a given item.

3.2 Content-based recommendation techniques

These recommendation techniques are based on an estimate of the similarity in terms of items' features. This paper considered each item as having the following features: show title, genres, production companies, creators, directors, writers, actors or presenters and description.

If J is the set of items rated by the user u whose similarity to the item i is above a certain threshold, the prediction for u 's rating of the show i can be computed through the following equation [29]:

$$p_{u,i} = \frac{\sum_{j \in J} r_{u,j} * w_{i,j}^2}{\sum_{j \in J} w_{i,j}^2} \quad (1)$$

⁴<http://www.tvgenius.net/>

where $w_{i,j}$ is the weighting factor, which in this case exclusively matches the similarity value (from 0 to 1) between items i and j , and $r_{u,j}$ is the rating given by user u to item j .

The similarity between two items is calculated based on the similarity of each of its features a ($sim_a(i,j)$), then weighted by various p_i weight factors, in order to reach a single similarity value $w_{i,j}$, as the following equation shows

$$w_{i,j} = \frac{\sum_{a \in A} sim_a(i,j) * p_a}{\sum_{a \in A} p_a} \quad (2)$$

Dice's coefficient is used to measure the genre similarity between two shows, since the genre feature is a group, in which rank is irrelevant [19]. On the other hand, since the production companies, creators, directors, writers and actors/presenters features include a rank reflecting the importance of the individual item in each set, the inverse rank measure will be used to measure these features' similarity. This measure confers varying weights according to each element's rank (i.e., the first elements are given greater relevance) [5]. Each element is seen as a chain of characters.

Regarding descriptions, with a sequence of words for each item, the number of occurrences of each word in each sequence is computed, and results are registered in the \mathbf{A}' and \mathbf{B}' vectors. Each of these vectors has $|A \cap B|$ elements, so that, for instance, A_1 and B_1 represent the number of times that the same word appears in the word sequences A and B , respectively. From this, the cosine of the two vectors is calculated to determine their similarity [19].

3.3 Collaborative recommendation techniques

Regarding user-to-user collaborative recommendation techniques, if V is the set of users except user u , $r_{v,i}$ is the rating given by user v to item i , $w_{u,v}$ is the value of the similarity between users u and v , σ_u is the standard deviation of u 's ratings, and \bar{r}_u is the average of his ratings, then the prediction of user u 's rating of the item i is given by [11]:

$$p_{u,i} = \bar{r}_u + \sigma_u \frac{\sum_{v \in V} \left[\left(\frac{r_{v,i} - \bar{r}_v}{\sigma_v} \right) w_{u,v} \right]}{\sum_{v \in V} w_{u,v}} \quad (3)$$

Research in [11] shows that the accuracy of the prediction will improve if the similarity value is multiplied by a weighting factor associated to the number of items that these users jointly rated. In practice, if the number of items rated by both users ($commonItems$) is smaller than a certain value ($minCommonItems$), weighting $commonItems/minCommonItems$ is applied. Thus, the first metric is:

$$w_{u,v} = \begin{cases} sim_{u,v} * \frac{commonItems}{minCommonItems} & \text{if } commonItems < minCommonItems \\ sim_{u,v} & \text{otherwise} \end{cases} \quad (4)$$

Equation 3 considers users with any sort of similarity, even a low and inconclusive one, which may skew predictions. As such, the inclusion of a threshold (i.e., a minimum required similarity value) was also tested [11].

The $w_{u,v}$ similarity between users u and v may also be computed through various correlation measures [21]. All these measures return a value between -1.0 and 1.0. The measures tested in this paper were the Pearson correlation, Spearman's rank correlation coefficient, Kendall's correlation coefficient and covariance. A variation of the Pearson correlation, which we will name as the fixed Pearson correlation, was also used, wherein each user's average is replaced by the midpoint value of the range of possible ratings (in a 0.0 to 1.0 range, the midpoint rating is 0.5) [11]. Thus, the ratings' polarity does not depend on the user's ratings but on the absolute range of ratings. The equation is:

$$\rho_{u,v}^{fixed} = \frac{\sum_{i \in I} (r_{u,i} - 0.5)(r_{v,i} - 0.5)}{\sqrt{\sum_{i \in I} (r_{u,i} - 0.5)^2 \sum_{i \in I} (r_{v,i} - 0.5)^2}} \quad (5)$$

Item-to-item collaborative techniques compute the similarity between items, which will be used to infer how a given user u will rate item i , knowing that he has already rated similar items [17]. The calculation is the same as in user-to-user collaborative recommendation techniques. As such, user u 's rating of item i is given by the following equation:

$$p_{u,i} = \bar{r}_i + \sigma_i \frac{\sum_{j \in J} \left[\left(\frac{r_{u,j} - \bar{r}_u}{\sigma_u} \right) w_{i,j} \right]}{\sum_{j \in J} w_{i,j}} \quad (6)$$

The recommendation systems literature also presents simpler and more efficient collaborative filtering techniques whose results are often as good as the more complex techniques previously mentioned. Therefore, the Top-N Deviation [29] and Weighted Slope One [16] techniques will be tested, as well as two variations employing the standard deviation (σ_u for user u) in each user's ratings to adapt to each user. The first technique, the Relative Top-N Deviation, corresponds to the following equation:

$$p_{u,i} = \bar{r}_u + \sigma_u \frac{\sum_{v \in V} \left(\frac{r_{v,i} - \bar{r}_v}{\sigma_v} \right)}{|V|} \quad (7)$$

On the other hand, the Relative Weighted Slope One's $D_{i,j}$ is computed by Equation 8, and the prediction value is given by Equation 9.

$$D_{i,j} = \sum_{u \in S_{i,j}} \frac{\left(\frac{r_{u,j} - r_{u,i}}{\sigma_u} \right)}{|S_{i,j}|} \quad (8)$$

$$p_{u,i} = \frac{\sum_{j \in R_i} (D_{i,j} * \sigma_u + u_j) |S_{i,j}|}{\sum_{j \in R_i} |S_{i,j}|} \quad (9)$$

where $D_{i,j}$ represents the average deviation between the ratings of item i and the classifications made for item j .

4 Hybrid strategies

This section presents some strategies that aim to combine several techniques described in the previous chapter. We'll begin by looking at strategies that combine techniques through heuristics, followed by an example of a combination strategy that employs machine learning techniques, seeking to optimize ranked recommendation lists.

- **Fixed rank switching combination** – This strategy consists in going through a ranked list of techniques, questioning each one about its prediction for a given rating. When a technique is able to present its prediction, the value is used, and the algorithm is completed without going through the remaining techniques.
- **Variable rank switching combination** – In this strategy, the ranking of techniques changes for each prediction, according to the accumulated Mean Absolute Error (MAE) obtained from items previously ranked by the user. In other words, for each user a prediction is inferred through each technique (ignoring the actual rating) and, according to results, the MAE for each technique is determined. When a prediction is needed for a potential recommendation, the techniques are ranked in ascending order according to each technique's MAE for the given user. Each technique will then be accessed according to its rank in order to present a prediction value.
- **Fixed weighting combination with three sets of switching combination techniques** – In this strategy, three types of predictions are considered, namely: (i) collaborative filtering, (ii) item popularity-based and (iii) content-based. Each of these contains one or two techniques combined through fixed weighting, meaning that in each of these types only one technique will present its prediction, and later the three will be combined through an arithmetic mean. This strategy aims to insure that the three types of techniques are weighted equally in all predictions, so that all three types may always contribute towards the predictions.
- **Variable weighting combination** – In this case, each technique's weight in the final prediction varies according to the same variable switching combination standard (i.e. MAE). The smaller a given technique's MAE is according to the history, the larger that technique's weight will be in predicting. The goal is to dilute the potential errors of individual techniques through improved predictions obtained by others.
- **Fixed weighting combination** – In weighting combination strategies, the main challenge is finding the ideal weights. A strategy in which each technique's weights are adjusted through time was previously mentioned. However, to create a fixed weighting strategy, it is necessary to determine the fixed weights for each technique. A frequent is to use supervised machine learning, namely learning-to-rank algorithms. Through these techniques, training data may be employed in order to learn a combination model for various recommendation techniques, optimizing results according to measures such as the normalized discounted cumulative gain (nDCG) [18].

The input for the learning-to-rank process is the training set consisting in n users $q_i (i = 1, \dots, n)$, each of them associated to $m^{(i)}$ recommendation technique

vectors $x^{(i)} = \{x_j^{(i)}\}_{j=1}^{m^{(i)}}$, each of them corresponding to an item associated to user q_i . For each user, documents are also ranked through $y^{(i)}$. By receiving this training set, the learning system trains the model, which will be the input for the rating system, allowing it to be applied in a test set with the same structure as the training set. As a result, the items associated to each input user are ranked, and this rank may thus be compared to the ideal rank. In this paper the SVM^{ndcg} learning-to-rank algorithm [8] was used to train the model.

5 Validation Data Set

Since no public data exists for experiments on the field of television show recommendation, a data set was built for this paper from the following three sources (corresponding to Web sites where human users describe, comment and declare preferences for TV shows): TV.com, IMDb and Living Social.

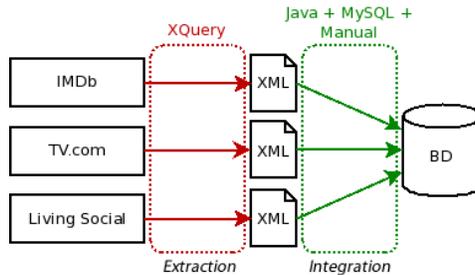


Fig. 1. Building a data set for the television show field.

The process of building the data set is summarized in Figure 1. Each of these three Web sites was accessed through a data extraction script, which ran through the list of the most popular shows, returning an XML file with all information pertaining to them. In order to build a large data set and to ensure a greater diversity in ratings, these three sources were integrated into a database. The data set was filtered so that it would only include users that rated five items or more. After filtering, the data set presents the distributions shown in Figure 2.

Each rating's proportion is shown in Figure 3, where we can see that more than half of the ratings correspond to values of 0.8 or above. This suggests that users overwhelmingly rate their preferred items.

Table 1 presents a more detailed statistical description of the created data set.

6 Experimental Evaluation

To evaluate the different recommendation techniques, all ratings were registered into the system. Later, every rating was inferred through each recommendation technique, ignoring the existing real rating. Once the prediction process was completed and the database was filled, two types of evaluations were performed:

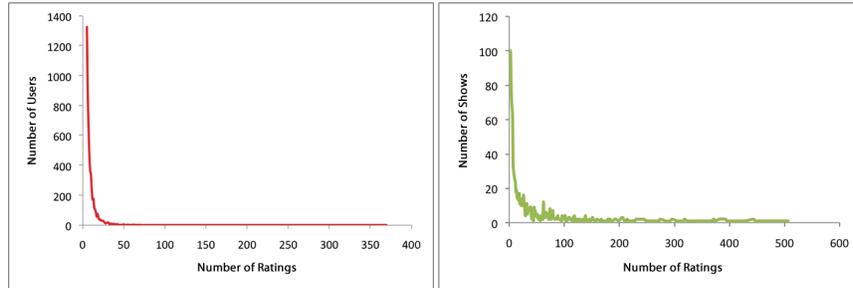


Fig. 2. User and item ratings distribution.

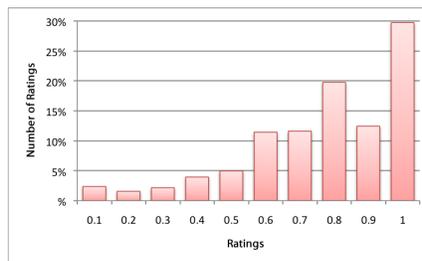


Fig. 3. Each rating's share of the data set.

- **Evaluation of the agreement between predictions and real values** – This evaluation is based on the difference between the predictions' values and the actual ratings' values, employing measures such as Mean Absolute Error (MAE) and Mean Squared Error (MSE) [13, 26].
- **Evaluation of lists containing recommended items** – In this evaluation, items are first ranked in two separate lists:
 - In one list, items are ranked according to their prediction value;
 - In the other list, they're ranked according to their actual rating value.

Both resulting lists are compared according to the normalized discounted cumulative gain (nDCG) measure, consisting in the accumulation of the relevance value of recommended items (i.e., the actual value of the user's rating) and also considering their rank. This means that the larger the cumulative gain, the greater interest the list will have for the user [15]. This measure is standardized by a ratio with the ideal cumulative gain (i.e., the cumulative gain value of the list of user's n preferred items). If p is the number of items considered in the list of recommended items and rel_i is the actual classification (i.e., the relevance) assigned by a user to the item of position i in the ordered list presented, DCG (no standardized) is given by:

$$DCG_p = rel_1 + \sum_{i=2}^p \frac{rel_i}{\log_2 i} \quad (10)$$

And nDCG is calculated as follows:

Name	TV Shows Data Set
Field	Television shows
Ratings	58.878
Items included	1.302
Users	5.638
Minimum ratings per user	5
Distribution type	Power-Law Distribution
Ratings domain	$\{0.1, \dots, 1.0\}$
Item information	Title, Genres, Companies, Description, Directors, Authors, Actors/Presenters

Table 1. TV Shows Data Set description.

$$nDCG_p = \frac{DCG_p}{IDCG_p} \quad (11)$$

Despite the nDCG measure’s standardization, the range of user ratings is ignored. For instance, a 7 rating given by a user whose top rating is 7 and whose lowest is 3 must be awarded greater relevance than a 7 rating given by another user whose top rating is 10 and whose lowest is 7. As such, the ratings given by users must not be evaluated as absolute values but as relative ratings within the user’s ratings profile. To fill this gap in nDCG, this paper presents an alternate way to evaluate each rating’s relevance in the context of the evaluation of a recommendation system. Thus, if r_u is u ’s set of ratings, we suggest that the relevance value for each rating (rel_i) be computed as follows:

$$rel_i = \begin{cases} \frac{rel_i - \min(r_u)}{\max(r_u) - \min(r_u)} & \text{if } \max(r_u) \neq \min(r_u) \\ 1 & \text{otherwise} \end{cases}$$

As a consequence, by changing the calculation of rel_i , an alternative to nDCG appears, which will be named nDRCG (normalized relative discounted cumulative gain).

6.1 Evaluation of the agreement between predictions and real values

Figure 4 presents results obtained for the evaluation of the agreement between predicted values and real ratings.

The poor results of the content-based strategy show that most users’ interests do not depend on a few of the show’s features; instead, they mostly depend on its perceived quality, which isn’t necessarily explicit in the descriptive features. This strategy’s low coverage should also be noted, as it reveals the difficulty in measuring similarities between items based on content, despite the data set’s wealth in descriptive information.

Regarding user-to-user collaborative filtering techniques, the Pearson correlation, Spearman’s correlation and covariance presented very similar results. Although the Pearson correlation is the most frequently used, these results show

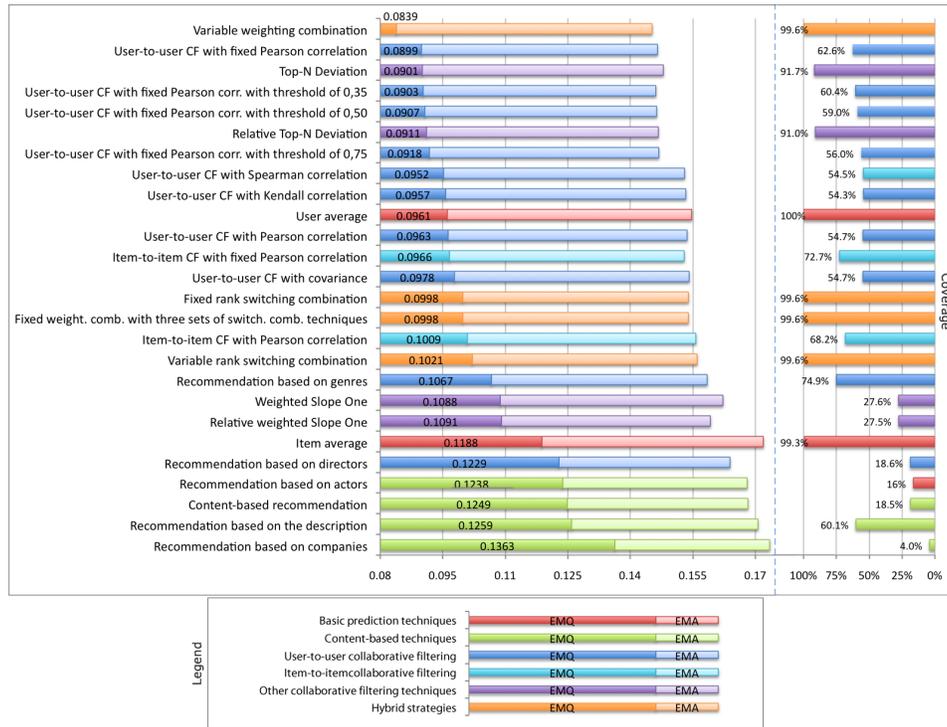


Fig. 4. Summary of results according to the evaluation of the agreement between predictions and real values, and of each prediction technique's coverage.

that other similarity techniques can be good alternatives. Indeed, Spearman's and Kendall's correlation actually present better results.

In the case of item-to-item collaborative filtering (Equation 6), this technique's results are 5% worse than the corresponding user-to-user technique, albeit with much greater coverage.

The user-to-user and item-to-item fixed Pearson correlation variations produced an MSE improvement of 8% and 5%, respectively, which shows that a below-average rating doesn't necessarily have negative implications, and it is more useful to describe as negative ratings only those below the midpoint value.

The inclusion of a threshold was also tested and, as should be expected, the larger the applied threshold, the smaller the coverage. However, even with a 75% threshold, the coverage loss was only around 14%. Including a threshold did not lead to vastly improved results, thus showing that less similar users do not have a significant impact on results, considering that the calculation of predictions in collaborative filtering is weighted according to similarity, and the values of low-similarity users have a proportional weight (i.e., not much) in the final result.

Regarding Slope One, its coverage is relatively low, since this technique requires a correspondence of at least one item between the user for whom predictions are made and the user upon whom similarity is computed; it is also necessary that

the similar user has rated the item upon which the prediction is to be made. The latter is the only requirement for Top-N deviation. Top-N deviation comes out as one of the better techniques – a fine alternative to the more complex collaborative filtering techniques, as it allows the system to become more efficient, thanks to the simplicity in its calculations and the breadth in its coverage. The relative Top-N deviations and relative Slope One variations presented very similar results to the original techniques.

It is difficult to compare hybrid strategies to other techniques since these strategies have much greater coverage. Looking at the results, the conclusion is that including several techniques (in the case of the variable weighting strategy) in a prediction will lead to much better results than predictions relying solely on the potentially better technique. It should be noted that the variable weighting hybrid strategy obtained excellent results, with improvements above 20% in the mean squared error relative to the other hybrid strategies, except the fixed weighting strategy. Therefore, this strategy obtains both total coverage and excellent results. The collective employment of several techniques allows some techniques' errors to be mitigated by the other's accuracy, instead of just relying on one of them.

The remaining hybrid strategies present similar results among themselves and, as they aim to include every rating, some accuracy is lost, since in many cases they have to resort to worse techniques, as these are the only ones capable of making these predictions. The variable weighting strategy overcomes this problem by combining several techniques simultaneously, diluting eventual mistakes that some techniques might make in certain scenarios.

6.2 Evaluation of lists containing recommended items

Figure 5 presents the evaluation results of lists of items recommended by each technique to each user. In this case, lists with the top 5 ranked items.

In the evaluation of lists recommended by each technique, the item rating average stands out with the best results. This shows that when it comes to users' preferred items, there is a high level of consensus, even if that consensus does not appear in lower rated items. Hence the poor result of this technique at evaluating errors.

Also patent is the fact that item-to-item collaborative filtering techniques are better at this sort of approach, meaning that there is a high correlation between the most popular item rating vectors, and therefore the similarity between items is better in this case than the calculation of the similarity between users.

Once again it is shown that the variable weighting combination strategy reduces errors better than other hybrid strategies (except that the learning-to-rank algorithm was 1% better) , albeit with worse results than the technique corresponding to using the average rating scores.

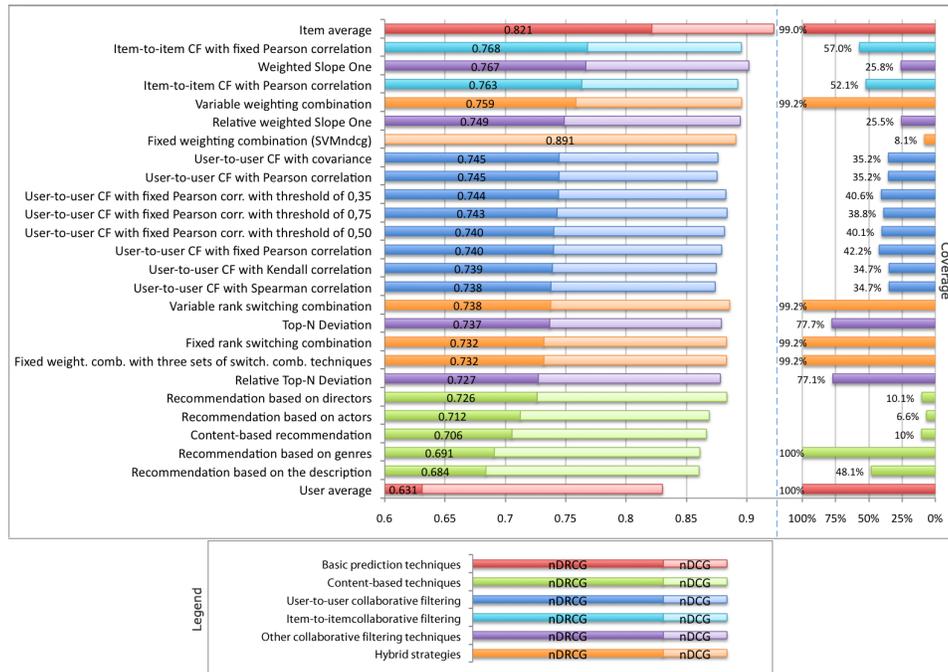


Fig. 5. Summary of results obtained according to the evaluation of the 5-preferred items lists.

7 Conclusions

This paper studied the field of recommendation systems, namely through their application in the context of television show recommendation, a domain that remains relatively unexplored despite its critical importance.

A comparative analysis of different recommendation techniques was made, and different results were obtained according to the evaluation strategy considered (i.e., evaluation based on ratings or based on lists of ranked items):

- By evaluating the agreement between predictions and real values, the conclusion was that user-to-user collaborative techniques employing the fixed Pearson correlation and Top-N Deviation produce the best results, while content-based techniques not only have low coverage but also present the worst results.
- Faced with another evaluation technique, the evaluation of lists containing n preferred items, item ratings averages stood out as the best technique, while the worst results were achieved by content-based techniques and by using the average scores of ratings.

After reviewing the various techniques, these were combined through hybrid strategies, employing different approaches, and the conclusion was that variable weighting combination and fixed weighting combination had the best results.

The main challenges for future research relate to performing real time test sessions with actual users testing the approaches presented here with other data sets, and experimenting with model-based techniques, which in this study were only considered when reviewing related research.

Bibliography

- [1] G. Adomavicius and E. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, (6), 2005.
- [2] K. Ali and W. Van Stam. TiVo: making show recommendations using a distributed collaborative filtering architecture. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2004.
- [3] M. Anderson, M. Ball, H. Boley, S. Greene, N. Howse, D. Lemire, and S. McGrath. Racofi: A rule-applying collaborative filtering system. *Proceedings of Collaboration Agents Conference (COLA)*, 2003.
- [4] M. Balabanovic and Y. Shoham. Fab: content-based, collaborative recommendation. *Communications of the ACM*, (3), 1997.
- [5] J. Bar-Ilan, K. Keenoy, E. Yaari, and M. Levene. User rankings of search engine results. *Journal of the American Society for Information Science and Technology*, (9), 2007.
- [6] D. Billsus, M.J. Pazzani, and J. Chen. A learning agent for wireless news access. In *Proceedings of the 5th international conference on Intelligent user interfaces*, 2000.
- [7] R. Burke. Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, (4), 2002.
- [8] S. Chakrabarti, R. Khanna, U. Sawant, and C. Bhattacharyya. Structured learning for non-smooth ranking losses. In *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2008.
- [9] L.M. de Campos, J.M. Fernandez-Luna, and J.F. Huete. A Bayesian network approach to hybrid recommending systems. In *Proceedings of the 11th International Conference of Information Processing and Management of Uncertainty in Knowledge-Based Systems*, 2006.
- [10] X. Fu, J. Budzik, and K.J. Hammond. Mining navigation history for recommendation. In *Proceedings of the 5th international conference on Intelligent user interfaces*, 2000.
- [11] J.L. Herlocker. Understanding and improving automated collaborative filtering systems. 2000.
- [12] J.L. Herlocker, J.A. Konstan, and J. Riedl. Explaining collaborative filtering recommendations. In *Proceedings of the 2000 ACM conference on Computer supported cooperative work*, 2000.
- [13] J.L. Herlocker, J.A. Konstan, L.G. Terveen, and J.T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS)*, (1), 2004.
- [14] S. Hsu, M. Wen, H. Lin, H. Lee, and C. Lee. Aimed - a personalized tv recommendation system. In *Proceedings of the 5th European Interactive TV Conference*, 2007.
- [15] K. Jarvelin and J. Kekalainen. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems (TOIS)*, (4), 2002.
- [16] D. Lemire and A. Maclachlan. Slope One Predictors for Online Rating-Based Collaborative Filtering, 2005.
- [17] G. Linden, B. Smith, and J. York. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet computing*, (1), 2003.
- [18] T.Y. Liu. Learning to rank for information retrieval. *Foundations and Trends in Information Retrieval*, (3), 2009.
- [19] C. Manning, P. Raghavan, and H. Schtze. *Introduction to Information Retrieval*. Cambridge University Press, 1 edition, 2008.
- [20] R.J. Mooney and L. Roy. Content-based book recommending using learning for text categorization. In *Proceedings of the 5th ACM conference on Digital libraries*, 2000.
- [21] J.L. Myers and A. Well. *Research design and statistical analysis*. Lawrence Erlbaum, 2003.
- [22] M. Pazzani and D. Billsus. Learning and revising user profiles: The identification of interesting web sites. *Machine learning*, (3), 1997.
- [23] M.J. Pazzani and D. Billsus. Content-based recommendation systems. 2007.
- [24] P. Resnick and H.R. Varian. Recommender systems. *Communications of the ACM*, (3), 1997.
- [25] R. Salakhutdinov, A. Mnih, and G. Hinton. Restricted Boltzmann machines for collaborative filtering. In *Proceedings of the 24th international conference on Machine learning*, 2007.
- [26] G. Shani and A. Gunawardana. Evaluating Recommendation Systems. Technical report, Microsoft Research, 2009.
- [27] R. Simeoni, M. Geymonat, R. Antonini, R. Montanari, and M. Perrero. DynamicTV: a new Entertainment paradigm for television. 2009.
- [28] X. Su and T. M. Khoshgoftaar. A Survey of Collaborative Filtering Techniques. *Advances in Artificial Intelligence*, 2009.
- [29] M. Van Setten. Supporting people in finding information: hybrid recommender systems and goal-based structuring. 2005.
- [30] M. Van Slobbe. Lighthouse: A distributed recommendations system for television programs. *Communications of the ACM*, (3), 2005.
- [31] J. Wang, A.P. De Vries, and M.J.T. Reinders. Unifying user-based and item-based collaborative filtering approaches by similarity fusion. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, 2006.