Recommender System Based on Purely Probabilistic Model from Pooled Sequence Statistics

Javier A. Kreiner[1] and Eitan Abraham[2]

[1] Center for Brain and Mind Sciences (CIMeC),
University of Trento, Rovereto (TN), Italy
javkrei@gmail.com
[2] School of Engineering and Physical Sciences,
Heriot-Watt University, Edinburgh, UK
E.Abraham@hw.ac.uk

Abstract. In this paper we present a method to obtain a recommendation ranking for items in a collection using a marginalization technique to estimate conditional probabilities. The method uses no content-related information and rests on a probabilistic model based on implicitly collected data from past user behaviour. Given a query triplet of items for which a list of recommended results is required, the technique uses estimates for the conditional probabilities of items appearing after the three doublets defined by the triplet. The technique leads to the evaluation of a score function which takes the simple form of a sum of these conditional probabilities. Results show that the approach has good performance with respect to other methods.

1 Introduction

Given the exponential growth of content availability that current technology provides, it is usually impossible for a user to even skim over each item in a collection of e.g. webpages, movies, books or products in order to make a choice. This was recognized early in the internet era and great effort has been devoted to the development of systems that assist the user in this task. The ability to separate the wheat from the chaff is, in fact, the defining element of a number of technology companies, e.g. google or yahoo, and the recommendation of items of interest is of paramount importance for many others like netflix and amazon. In a nutshell, recommender systems attempt to provide a list of elements that are likely to be of relevance to a user. The ranking of elements is based on characteristics of the items and their relationships (content-based information), information about the user in question, and information explicitly or implicitly provided by other users (collaborative-type information) [4].

The ECML/PKDD Discovery Challenge 2011 [1] had as its purpose the improvement of VideoLectures.net's [3] recommender system. The challenge was set up on the tunedit.org platform [2], which provides functionality to easily organize data mining competitions. VideoLectures.net is an open access multimedia repository of video lectures available on the internet. The videos are recordings of lectures given by researchers in diverse areas of science during scientific events such as conferences and workshops. There were two main tasks and a workflow contest during the challenge. This paper

describes a solution for task 2 which obtained second place. This task simulated the situation in which a particular user is known to have seen a set of three videos. Based on this knowledge, the system should recommend 10 videos in descending order of relevance.

The paper is organized as follows. §2 describes the available data. In §3 the main idea of the solution is introduced. §4 contains details about the implementation. §5 shows the performance of the method proposed. Finally, in §6 appear some remarks and conclusions.

2 Data and evaluation

The contestants were provided with a number of datasets containing information about the lectures and past viewing behaviour to construct a recommender system. The datasets contained two types of information. The first type was information about the content-related features of each lecture such as the author and category to which the lecture belongs. The second type was statistical information extracted from the viewing sequences of site users. Given the lack of explicit profiles, the users were identified by a cookie left in their browsers. Each sequence was determined by the stream of videos seen with a uniquely cookie-identified browser. The viewing sequences were not actually provided. Instead, aggregated information about user behaviour was given. The solution presented here disregards content-related data, and uses only statistics obtained from pooled viewing sequences.

In what follows, we introduce notation that will allow us to refer to the statistics used by the method. Let (v_1, v_2, v_3) be a triplet of three different lectures' id's, and t the id of a 'target' lecture , that is, a lecture seen after the triplet. Thus, we define the following:

1. **pairs frequencies**: the number of distinct sequences in which a pair of lectures was viewed together (not necessarily consecutively and regardless of order). We denote the co-viewing frequency of v_1 and v_2 by $f(v_1, v_2)$.
2. **triplets frequencies**: given a triplet of video lectures, the number of viewing sequences in which the three videos that define the triplet appear. $f(v_1, v_2, v_3)$ denotes this frequency.
3. **triplets' targets frequencies**: given a triplet of video lectures, the number of viewing sequences in which a given target video lecture has been seen *after* viewing the three videos in the triplet (available only for the ten targets most frequently viewed after each triplet). The notation used in this case is: $f(v_1, v_2, v_3; t)$. A semicolon to separate the target from the rest because this frequency is counted differently to the previous two.

An example follows that clarifies the last one of these definitions (adapted from [1]). Consider the following viewing sequence:

$$v_1 \to v_7 \to v_2 \to v_1 \to v_4 \to v_5 \to v_6 \to v_3$$

The first operation is to remove duplicates, after which the sequence becomes:

$$v_1 \to v_7 \to v_2 \to v_4 \to v_5 \to v_6 \to v_3$$

Suppose that we want to obtain the target viewing frequencies for the triplet (v_1, v_4, v_5). Given that only v_6 and v_3 appear after all of v_1, v_4 and v_5, we increment the two frequencies $f(v_1, v_4, v_5; v_3)$ and $f(v_1, v_4, v_5; v_6)$ by one.

Now suppose that there is another sequence:

$$v_4 \to v_5 \to v_8 \to v_1 \to v_6 \to v_3 \to v_7$$

Then, given that v_6, v_3 and v_7 appear after all of v_1, v_4 and v_5 we increment the three frequencies $f(v_1, v_4, v_5; v_3)$, $f(v_1, v_4, v_5; v_6)$ and $f(v_1, v_4, v_5; v_7)$ by one.

Hence, if these were the only two sequences containing (v_1, v_4, v_5) we would have:

$$f(v_1, v_4, v_5; v_3) = 2$$

$$f(v_1, v_4, v_5; v_6) = 2$$

$$f(v_1, v_4, v_5; v_7) = 1$$

The whole triplets' targets frequencies dataset was divided by the organizers in two parts. The first part, consisting of the target frequency information for 109044 triplets, was provided to the contestants to train their models. For the rest of the triplets, 60274, the target frequency information was retained by the organizers as a test set to score the submitted solutions. As is customarily done in data mining competitions, part of this test set was used to rank the contestants in the competition leaderboard, while the complete test set was used to construct the final ranking. The performance measure used was the Mean Average R-Precision (MARp, see [1]).

3 Approach used in the solution

Consider a given triplet (v_1, v_2, v_3) and a target t for that triplet available in the training set. Then let:

$$p(t|v_1, v_2, v_3) = \frac{f(v_1, v_2, v_3; t)}{f(v_1, v_2, v_3)} \tag{1}$$

denote the conditional probability of seeing t given that v_1, v_2, v_3 have been seen previously in any order. In a similar way, if we had available $f(v_1, v_2; t)$, the number of sequences in which t appears after (v_1, v_2), we could calculate

$$p(t|v_1, v_2) = \frac{f(v_1, v_2; t)}{f(v_1, v_2)} \tag{2}$$

i.e. the conditional probability of seeing t after having seen the doublet (v_1, v_2).

The solution is based on the observation that it is possible to estimate this conditional probability of seeing a video after having seen a doublet of videos, based on the triplets' information available in the training set:

$$p(t|v_1, v_2) \approx \hat{p}(t|v_1, v_2) := \frac{\sum_v f(v_1, v_2, v; t)}{\sum_v f(v_1, v_2, v)} \tag{3}$$

As a matter of consistency with the numerator as regards the counting of sequences, we chose the denominator as a sum over the triplets in the training set. Consequently, this is an approximation for the following reasons: (a) there is an unknown overlap between the set of sequences counted by $f(v_1, v_2, v; t)$ and $f(v_1, v_2, v'; t)$, and also between the set of sequences counted by $f(v_1, v_2, v)$ and $f(v_1, v_2, v')$ for two different videos v and v', (b) there is no information available for target videos that are not in the top ten most frequently seen videos for each triplet, and (c) there are triplets that have been retained for the test set for which there is no target information. In any case, this "marginalization" over the third video would seem to provide a reasonable estimate.

Consider now a query triplet (q_1, q_2, q_3). The task is to identify, in descending order of viewing frequency, the ten videos most frequently viewed after having seen q_1, q_2, q_3 in any order. Using conditional probability estimates $\hat{p}(t|q_1, q_2)$, $\hat{p}(t|q_1, q_3)$, $\hat{p}(t|q_2, q_3)$, the ranking is constructed based on some function of them, i.e. let

$$\text{score}(t) = F(\hat{p}(t|q_1, q_2), \hat{p}(t|q_1, q_3), \hat{p}(t|q_2, q_3)) \tag{4}$$

be the score assigned to video t. It is reasonable to postulate some restrictions for F. Namely, that it should be increasing, or at least non-decreasing, in each of its arguments, and also that it should obtain the same value for any permutation of its arguments. A number of options were tested for F, among them the product of the arguments. In the end, the sum of the conditional probabilities happened to give a very good result, i.e.

$$\begin{aligned} &F(\hat{p}(t|q_1, q_2), \hat{p}(t|q_1, q_3), \hat{p}(t|q_2, q_3)) = \\ &= \hat{p}(t|q_1, q_2) + \hat{p}(t|q_1, q_3) + \hat{p}(t|q_2, q_3) \end{aligned} \tag{5}$$

Since the original submission of these results we found, however, that there is another option that provides a superior ranking. This is an entropy-like function defined as,

$$\begin{aligned} &F(\hat{p}(t|q_1, q_2), \hat{p}(t|q_1, q_3), \hat{p}(t|q_2, q_3)) = \\ &= -\sum_{1 \le i < j \le 3} \hat{p}(t|q_i, q_j) \log(\hat{p}(t|q_i, q_j) \end{aligned} \tag{6}$$

which also enjoys the same permutation symmetry.

4 Implementation details

In order to render the method computationally viable, as a first step, the target conditional probability estimates given a doublet are calculated and stored in an index. In the index, each doublet points to a list of targets that appeared after the doublet with the corresponding conditional probability estimate (see Equation (3)). This is done by traversing the triplets' training set once and accumulating the frequencies for each doublet and target.

Once this index is constructed, given a query triplet (q_1, q_2, q_3), consider the three doublets (q_1, q_2), (q_1, q_3), and (q_2, q_3). The index contains for each of these doublets a list of target videos with corresponding conditional probabilty estimates. Let L_1, L_2 and L_3 be the list of targets in the index for (q_1, q_2), (q_1, q_3), and (q_2, q_3), respectively.

Additionally, let L be the list of videos appearing simultaneously in L_1, L_2 and L_3, i.e. $L = L_1 \cap L_2 \cap L_3$. For these targets, which would seem particularly relevant for the query, we can readily use the scoring function F to rank them. In the case that the number of results thus obtained is at least ten (the number of required recommendations for the task), the top ten sorted recommendations are written to an output file and we are done with this query triplet.

For some query triplets it may happen that there are less than ten targets in the intersection list L (in the query file this happened for 20688 of the total of 60274 query triplets). If this happens, consider the targets t that are in the intersection of exactly two of the lists i.e.

$$t \in L' := (L_1 \cap L_2) \cup (L_1 \cap L_3) \cup (L_2 \cap L_3) \setminus L$$

for these t's two of the probability estimates are available. They are ranked, again using the function F, and appended after the previous recommendations.

If at this point the recommendation list still has less than ten results (3372 of the 60274 queries), consider the targets that appear exactly in one of the three lists, i.e.

$$t \in L'' := L_1 \cup L_2 \cup L_3 \setminus ((L_1 \cap L_2) \cup (L_1 \cap L_3) \cup (L_2 \cap L_3))$$

For these targets one probability estimate is available. Once again, they are ranked using F and added to the recommendation list.

There are still some query triplets for which a list of at least ten recommendations cannot be obtained (412 of 60274 queries), for these the conditional probability estimates of targets appearing after single videos $\hat{p}(t|q_1)$, $\hat{p}(t|q_2)$, and $\hat{p}(t|q_3)$ are calculated "marginalizing" over two videos:

$$p(t|q) \approx \hat{p}(t|q) := \frac{\sum_{v,w} f(q,v,w|t)}{\sum_{v,w} f(q,v,w)} \tag{7}$$

Now, the score for ranking given to a target for query triplet (q_1, q_2, q_3) is:

$$\hat{p}(t|q_1) + \hat{p}(t|q_2) + \hat{p}(t|q_3) \tag{8}$$

Finally, there are some triplets in the query file for which after carrying out the previous steps still less than 10 recommendations are obtained (29 of 60274), in this case the video pairs co-viewing frequencies are used to generate the remaining recommended video lectures.

The method described above obtained a score of 0.60749 on the leaderboard and a score of 0.61134 on the complete test set. The final solution submitted, which obtained a score of 0.60791 on the leaderboard and a score of 0.61172 on the complete test set, included a coefficient per doublet that was fitted and incorporated into the scoring function. The rationale behind these coefficients was that they might adjust for some of the inaccuracies in calculating the conditional probability estimates discussed above.

Consider a model introducing these coefficients:

$$\begin{aligned} &F(p(t|q_1,q_2), p(t|q_1,q_3), p(t|q_2,q_3)) = \\ &= \lambda_{q_1,q_2} \times p(t|q_1,q_2) + \lambda_{q_1,q_3} \times p(t|q_1,q_3) + \lambda_{q_2,q_3} \times p(t|q_2,q_3) \end{aligned} \tag{9}$$

To fit these coefficients a greedy grid search method was used. Accordingly, each coefficient was initialized to 1. Given a doublet for which the coefficient needs to be fitted, a greedy search was conducted over a range (range used for the solution: $[0.5-1.5]$) and the coefficient value was selected for which the evaluation metric (MARp) over the available training triplets that contained the doublet was maximal. After greedy fitting these coefficients sequentially for each doublet and calculating the recommendations for the test set using Equation (9) for the score function, the leaderboard score improved to 0.60791. The Wilcoxon signed rank test was used to assess whether the difference was statistically significant. The p-value obtained was 0.01462, which does not lend strong support to the hypothesis that the means are different.

5 Results

Table 1 contains the scores for the leaderboard and complete test sets obtained using the methods described. The table also shows the score obtained when using a solution based on single videos conditional probability estimates only (see Equation (7)). Additionally, the scores obtained by the winner solution and the third place solution are included for comparison.

Again, the Wilcoxon signed rank test was used to assess statistical significance of the difference in scores between the original method using the simple sum as ranking function and the one using an entropy-like ranking function. The test yielded a p-value in the order of 10^{-16}, confirming that the difference in scores is statistically significant.

Table 1. Scores for the three methods described

Method	Leaderboard Score	Complete Test Set Score
Singles Cond. Probs.	0.41844	0.42057
Doublets Cond. Probs.	0.60749	0.61134
Doublets Cond. Probs. with Coeffs.	0.60791	0.61172
Doublets Cond. Probs. entropy-like F	0.60910	0.61285
Winner Solution (D'yakonov Alexander)	0.62102	0.62415
Third Place Solution (Vladimir Nikulin)	0.58727	0.59063

6 Summary and Conclusions

The method presented here uses a purely probabilistic approach to construct a recommender system from pooled viewing sequences statistical data. To do this we introduced an approximate marginalization technique leading to an estimate of the conditional probabilities of viewing target videos given that a doublet has been seen. These are in turn combined using a fully symmetric function to calculate a ranking score.

Introduction of doublet-dependent coefficients did not improve the performance in a statistically significant amount according to the Wilcoxon signed rank test. On the other hand, replacing a simple sum by an entropy-like function for the ranking function yielded

a statistically significant higher performance, again as assessed by the same statistical test.

The technique is straightforward, intuitively sound, being based on a simple insight, and easy to implement. Furthermore, it displayed better performance compared to other methods, obtaining second place on task 2 of the ECML/PKDD Discovery Challenge 2011.

References

1. ECML/PKDD Discovery Challenge 2011, http://tunedit.org/challenge/vlnetchallenge
2. tunedIT.org website, http://www.tunedit.org
3. VideoLectures.net website, http://www.videolectures.net
4. Adomavicius, G., Tuzhilin, A.: 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 17, 734–749 (2005)