Experimenting Analogical Reasoning in Recommendation

Nicolas Hug, Henri Prade, Gilles Richard

Experimenting Analogical Reasoning in Recommendation

Nicolas Hug(✉), Henri Prade, and Gilles Richard

Institut de Recherche en Informatique de Toulouse, Université Paul Sabatier,
31062 Toulouse Cedex 09, France
{nicolas.hug,henri.prade,gilles.richard}@irit.fr

Abstract. Recommender systems aim at providing suggestions of interest for end-users. Two main types of approach underlie existing recommender systems: content-based methods and collaborative filtering. In this paper, encouraged by good results obtained in classification by analogical proportion-based techniques, we investigate the possibility of using analogy as the main underlying principle for implementing a prediction algorithm of the collaborative filtering type. The quality of a recommender system can be estimated along diverse dimensions. The accuracy to predict user's rating for unseen items is clearly an important matter. Still other dimensions like *coverage* and *surprise* are also of great interest. In this paper, we describe our implementation and we compare the proposed approach with well-known recommender systems.

1 Introduction

In a world of information overload, automatic filtering tools are essential to extract relevant information from basic noise. In the field of e-commerce, recommender systems play the role of search engines when surfing the entire web: they filter available items to provide relevant suggestions to customers.

Besides, analogical reasoning is widely acknowledged as an important feature of human intelligence [4,9]. It is a powerful way for establishing parallels between apparently non related objects, and then guessing relations, properties, or values, on the basis of the observed similarities and dissimilarities. As a consequence, we may consider an analogy-based system as a suitable candidate for providing a user with relevant, and possibly surprising, recommendations. Indeed, providing the user with both accurate and surprising recommendation has become a key challenge. This is the option that we investigate in this paper.

Our paper is organized as follows. In the next section, we provide a brief survey of recommender system technologies. We also review diverse dimensions along which such a system can be evaluated. In Sect. 3, we provide the necessary background about analogical proportions and how they can be the basis of an inference process. In Sect. 4, we investigate how analogical proportions may provide a clean underlying framework to design a recommender system. In Sect. 5, we report the results obtained on a benchmark and we compare them to those of well-known existing approaches. Finally, we conclude and provide lines for future research in Sect. 6.

2 Background on Recommender Systems

The aim of a recommendation system is to provide users with lists of relevant personalized items. Let us now formalize the problem.

2.1 Problem Formalization

Let U be a set of users and I a set of items. For some pairs $(u, i) \in U \times I$, a rating r_{ui} is supposed to have been given by u to express if he/she likes or not the item i. It is quite common that $r_{ui} \in [1, 5]$, 5 meaning a strong preference for item i, 1 meaning a strong rejection, and 3 meaning indifference, or just an average note. Let us denote by R the set of ratings recorded in the system. It is well known that, in real systems, the size of R is very small with regard to the potential number of ratings which is $|U| \times |I|$, as a lot of ratings are missing. In the following, U_i denotes the set of users that have rated item i, and I_u is the set of items that user u has rated.

To recommend items to users, a recommender system will proceed as follows:

1. Using a prediction algorithm A, estimate the unknown ratings r_{ui} (i.e. $r_{ui} \notin R$). This estimation $A(u, i)$ is usually denoted $\hat{r}_{ui}$.
2. Using a recommendation strategy S and in the light of the previously estimated ratings, recommend items to users. For instance, a basic yet common strategy is to suggest to user u the items $i \notin I_u$ with the highest $\hat{r}_{ui}$.

The two main prediction techniques are commonly referred to as *content-based* and *collaborative filtering*, that we both briefly review below.

2.2 Content-Based Techniques

Content-based algorithms use the metadata of users and items to estimate a rating. Metadata are external information that can be collected. Typically:

– for users: gender, age, occupation, location (zip code), etc.
– for items: it depends on the type of items, but in the case of movies, it could be their genre, main actors, film director, etc.

Based on these metadata, a content-based system will try to find items that are similar to the ones for which the target user has already expressed a preference (for instance by giving a high rating). This implies the need for a similarity measure between items. A lot of options are available for such metrics. They will not be discussed here as our method is of a collaborative nature.

Indeed, a well-known drawback of content-based techniques is their tendency to recommend only items that users may already know, and therefore the recommendations lack in novelty, surprise and diversity. In the following, we only use collaborative filtering techniques, so we do not consider the use of any metadata.

2.3 Collaborative Filtering Techniques

By collaborative filtering, we mean here algorithms that only rely on the set of known ratings R to make a prediction: to predict $r_{ui} \notin R$, the algorithm A will output $\hat{r}_{ui}$ based on R or on a carefully chosen subset. The main difference between collaborative and content-based method is that in the former, metadata of items and users are not used, and in the latter the only ratings we may take into account are that of the target user. A popular collaborative filtering technique is neighbourhood-based, that we describe here in its simplest form.

To estimate the rating of a user u for an item i, we select $N_i^k(u)$, the set of k users that are most similar to u and that have rated i. Here again, there is a need for a similarity measure (between users, and based on their respective ratings). The estimation of r_{ui} is computed as follows:

$$\hat{r}_{ui} = \underset{v \in N_i^k(u)}{\text{aggr}} \; r_{vi},$$

where the aggregate function aggr is usually a mean weighted by the similarity between u and v. A more sophisticated prediction, popularized by [2] is as follows:

$$\hat{r}_{ui} = b_{ui} + \underset{v \in N_i^k(u)}{\text{aggr}} \; (r_{vi} - b_{vi}),$$

where b_{ui} is a baseline (or bias) related to user u and item i. It is supposed to model how u tends to give higher (or lower) ratings than the average of ratings μ, as well as how i tends to be rated higher or lower than μ. As it uses the neighbourhood of users to output a prediction, this technique tends to model local relationships in the data.

Note that it is perfectly possible to proceed in an item-based way. Indeed, rather than looking for users similar to u, one may look for items similar to i, which leads to formulas dual of the above ones.

2.4 Recommender System Evaluation

Providing an accurate measure of the overall quality of a recommender system is not a simple task and diverse viewpoints have to be considered (see [14] for an extensive survey).

Accuracy. The performance of the algorithm A is usually evaluated in terms of accuracy, which measures how close the rating prediction $\hat{r}_{ui}$ is to the true rating value r_{ui}, for every possible prediction. To evaluate the accuracy of a prediction algorithm, one usually follows the classical machine learning framework: the set of ratings R is divided into two disjoint sets R_{train} and R_{test}, and the algorithm A has to predict ratings in R_{test} based on the ones belonging to R_{train}.

The Root Mean Squared Error (RMSE) is a very common indicator of how accurate an algorithm is, and is calculated as follows:

$$\text{RMSE}(A) = \sqrt{\frac{1}{|R_{test}|} \sum_{r_{ui} \in R_{test}} (\hat{r}_{ui} - r_{ui})^2}$$

To better reflect the user-system interaction, other precision-oriented metrics are generally used in order to provide a more informed view.

Precision and Recall. Precision and recall help measuring the ability of a system to provide relevant recommendations. In the following, we denote by I_S the set of items that the strategy S will suggest to the users using the predictions coming from A. For ratings in the interval $[1, 5]$, a simple strategy could be for example to recommend an item i to user u if the estimation rating $\hat{r}_{ui}$ is greater than 4.

$$I_S = \{i \in I | \exists u \in U, \hat{r}_{ui} \geq 4\}.$$

Let I_{relev} be the set of items that are actually relevant to users (i.e. the set of items that would have been recommended to users if all the predictions made by A were exact). The precision of the system is defined as the fraction of recommended items that are relevant to the users:

$$\text{Precision} = \frac{|I_S \cap I_{relev}|}{|I_S|},$$

and the recall as the fraction of recommended items that are relevant to the users out of all possible relevant items:

$$\text{Recall} = \frac{|I_S \cap I_{relev}|}{|I_{relev}|},$$

If accurate predictions are crucial, it is widely agreed that it is insufficient for deploying an effective recommendation engine. Indeed, still other dimensions are worth estimating in order to get a complete picture of the performance of a system. [5,6,8]. For instance, one may naturally expect from a recommender system not only to be accurate, but also to be surprising, and to be able to recommend a large number of items. When evaluating the recommendation strategy, one must keep in mind that its performance is closely related to that of the algorithm A, as the recommendation strategy S is based on the predictions provided by A.

Coverage. Coverage, in its simplest form, is used to measure the ability of a system to recommend a large amount of items: it is quite easy indeed to create a recommender system that would only recommend very popular items. Such a recommender system would drop to zero added value. Coverage can be defined as the proportion of recommended items out of all existing items:

$$\text{Coverage} = \frac{|I_S|}{|I|}.$$

Surprise. Users expect a recommender system to be surprising: recommending an extremely popular item is not really helpful. Following the works of [6], surprise of a recommendation can be evaluated with the help of the pointwise mutual information (PMI). The PMI between two items i and j is defined as follows:

$$PMI(i, j) = -\log_2 \frac{p(i, j)}{p(i)p(j)} / \log_2 p(i, j),$$

where $p(i)$ and $p(j)$ represent the probabilities for the items to be rated by any user, and $p(i,j)$ is the probability for i and j to be rated together : $p(i) = \frac{|U_i|}{|U|}$ and $p(i,j) = \frac{|U_i \cup U_j|}{|U|}$. PMI values fluctuate between the interval $[-1,1]$, -1 meaning that i and j are never rated together and 1 meaning that they are always rated together. To estimate the surprise of recommending an item i to a user u we have two choices:

- either to take the maximum of the PMI values for i and all other items rated by u, with $Surp^{max}(u,i) = \max_{j \in I_u} PMI(i,j)$
- or to take the mean of these PMI values with $Surp^{avg}(u,i) = \frac{\sum_{j \in I_u} PMI(i,j)}{|I_u|}$

Then the overall capacity of a recommender to surprise its users is the mean of the surprise values for all predictions.

3 Background on Analogical Proportions

An analogical proportion "a is to b as c is to d" states analogical relations between the pairs (a,b) and (c,d), as well as between the pairs (a,c) and (b,d). It is only rather recently that formal definitions have been proposed for analogical proportions, in different settings [7,10,16]. In this section, we provide a brief account of a formal view of analogical relations that underlie their use in the proposed algorithm. For more details, see [11–13].

Formal Framework. It has been agreed, since Aristotle time, that an analogical proportion T, as a quaternary relation, satisfies the three following characteristic properties:

1. $T(a,b,a,b)$ (reflexivity)
2. $T(a,b,c,d) \implies T(c,d,a,b)$ (symmetry)
3. $T(a,b,c,d) \implies T(a,c,b,d)$ (central permutation)

There are various models of analogical proportions, depending on the target domain. When the underlying domain is fixed, $T(a,b,c,d)$ is simply denoted $a : b :: c : d$. Standard examples are:

- Domain $\mathbb{R}$: $a : b :: c : d$ iff $a/b = c/d$ iff $ad = bc$ (geometric proportion)
- Domain $\mathbb{R}$: $a : b :: c : d$ iff $a - b = c - d$ iff $a + d = b + c$ (arithmetic proportion)
- Domain $\mathbb{R}^n$: $\boldsymbol{a} : \boldsymbol{b} :: \boldsymbol{c} : \boldsymbol{d}$ iff $\boldsymbol{a} - \boldsymbol{b} = \boldsymbol{c} - \boldsymbol{d}$. This is just the extension of arithmetic proportion to real vectors. In that case, the 4 vectors $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}, \boldsymbol{d}$ build up a parallelogram.
- Domain $\mathbb{B} = \{0,1\}$: $a : b :: c : d$ iff $(a \wedge d \equiv b \wedge c) \wedge (a \vee d \equiv b \vee c)$
- Domain $\mathbb{B}^n$: $\boldsymbol{a} : \boldsymbol{b} :: \boldsymbol{c} : \boldsymbol{d}$ iff $\forall i \in [1,n], \quad a_i : b_i :: c_i : d_i$.

Other definitions are available when dealing with matriceces, formal concepts or lattices, etc. (see [7,16] for other options). In this paper, we are interested in evaluating analogical proportions between ratings, which might be Boolean (*like*/*dislike*), or in our case study, integer-valued (using the scale $\{1,2,3,4,5\}$).

Equation Solving. Starting from such an analogical proportion, the equation solving problem amounts to finding a fourth element x to make the incompletely stated proportion $a : b :: c : x$ to hold. As expected, the solution of this problem depends on the domain on which the analogy is defined. For instance, in the case of extended arithmetic proportions, the solution always exists and is unique: $\boldsymbol{x} = \boldsymbol{b} - \boldsymbol{a} + \boldsymbol{c}$. In terms of geometry, this simply tells us that given 3 points, we can always find a fourth one to build a parallelogram.

The existence of a unique solution is not always granted: for instance in the Boolean setting, the solution may not exist [10].

Analogical Inference. In this perspective, analogical reasoning can be viewed as a way to infer new plausible information, starting from observed analogical proportions. The *analogical jump* is an unsound inference principle postulating that, given 4 vectors $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}, \boldsymbol{d}$ such that the proportion holds on some components, then it should also hold on the remaining ones. This can be stated as (where $\boldsymbol{a} = (a_1, a_2, \cdots a_n)$, and $J \subset [1, n]$):

$$\frac{\forall j \in J, a_j : b_j :: c_j : d_j}{\forall i \in [1, n] \setminus J, a_i : b_i :: c_i : d_i} \quad (analogical\ inference)$$

This principle leads to a prediction rule in the following context:

- 4 vectors $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}, \boldsymbol{d}$ are given where $\boldsymbol{d}$ is partially known: only the components of $\boldsymbol{d}$ with indexes in J are known.
- Using analogical inference, we can predict the missing components of $\boldsymbol{d}$ by solving (w.r.t d_i) the set of equations (in the case they are solvable):

$$\forall i \in [1, n] \setminus J, \quad a_i : b_i :: c_i : d_i.$$

In the case where the items are such that their last component is a label, applying this principle to a new element $\boldsymbol{d}$ whose label is unknown leads to predict a candidate label for $\boldsymbol{d}$.

4 Analogical Recommendation

The main idea is that if an analogical proportion stands between four users a, b, c, d, meaning that for each item j that they have commonly rated, the analogical proportion $r_{aj} : r_{bj} :: r_{cj} : r_{dj}$ holds, then it should also hold for an item i that a, b, c have rated but d has not (i.e. r_{di} is the missing component). This leads us to estimate r_{di} as the solution $x = \hat{r}_{di}$ of the following analogical equation:

$$r_{ai} : r_{bi} :: r_{ci} : x.$$

Given a pair (u, i) such that $r_{ui} \notin R$ (i.e. there is no available rating from user u for item i), the main procedure is as follows:

1. find the set of 3-tuples of users a, b, c such that an analogical proportion stands between a, b, c, and u and such that the equation $r_{ai} : r_{bi} :: r_{ci} : x$ is solvable.

2. solve the equation $r_{ai} : r_{bi} :: r_{ci} : x$ and consider the solution x as a candidate rating for r_{ui}.
3. set $\hat{r}_{ui}$ as an aggregate of all candidate ratings.

The first step simply states that users a, b, c and u, considered as vectors of ratings, constitute a parallelogram. This condition is a bit strong and we may want to relax it by allowing some deformation of this parallelogram. This can be done by choosing another condition, such as $||(a - b) - (c - d)|| \leq \lambda$ where λ is a suitable threshold and $||.||$ denotes the Euclidean norm.

Another modification to the algorithm would be to only search for the users a, b, and c in a subset of U. One may consider the set of the k-nearest neighbours of d, using the assumption that neighbours are the most relevant users to estimate a recommendation for d.

Obviously, analogical proportion may be applied in an item-based way rather than in a user-based way, as in the case of standard techniques. Both views will be considered in the experimentation.

Implementation. In our implementation, we consider the basic definition of analogy using the arithmetic definition in $\mathbb{R}^n$: $\boldsymbol{a} : \boldsymbol{b} :: \boldsymbol{c} : \boldsymbol{d} \iff \boldsymbol{a} - \boldsymbol{b} = \boldsymbol{c} - \boldsymbol{d}$. Given 4 users, they are represented as real vectors of dimension m, where m is the number of item they have rated in common. This dimension can change with the 4-tuples of users that we consider. Then, the threshold λ has to be a function of this dimension m, as the range of values that $||.||$ may take depends on it. Using cross-validation, we have found that $\lambda = \frac{3}{2} \cdot \sqrt{m}$ showed the best results and acts as a kind of normalization factor. Our pseudo-code is described in Algorithm 1.

5 Experiments and Results

Our algorithm *Analogy* ($k = 20$) is compared to the neighbourhood-based algorithms described in Sect. 2.3, referred to as *kNN* for the basic model and *Bsln-kNN* for the extended model using a baseline predictor (with $k = 40$). For each of the algorithms, we have estimated the metrics described in Sect. 2.4. The recommendation strategy S is to recommend i to u if $\hat{r}_{ui} \geq 4$.

Dataset. We have tested and compared our algorithm on the Movielens-100K dataset[1], composed of 100,000 ratings from 1000 users on 1700 movies. Each rating belongs to the interval $[1, 5]$.

Evaluation Protocol. In order to obtain meaningful measures, we have run a five-folds cross-validation procedure: for each of the five steps, the set of ratings R is split into two disjoint sets R_{train} and R_{test}, R_{train} containing five times more ratings than R_{test}. The reported measures are averaged over the five steps.

Results and Comments. Table 1 shows the performances of the algorithms applied in a user-based way. Similar experiments have been led in a movie-based setting, exhibiting very similar results, slightly worse for RMSE (about 5‰ higher).

[1] http://grouplens.org/datasets/movielens/.

Algorithm 1. Analogy

Input: A set of known ratings R, a user u, an item i such that $r_{ui} \notin R$.
Output: $\hat{r}_{ui}$, an estimation of r_{ui}
Init:
$C = \varnothing$ // list of candidate ratings
for all users a, b, c such that

1. $r_{ai} \in R, r_{bi} \in R, r_{ci} \in R$
2. $r_{ai} - r_{bi} = r_{ci} - x$ is solvable // i.e. the solution $x \in [1, 5]$
3. $||(a - b) - (c - d)|| \leq \lambda$ // Analogy almost stands between a, b, c, d considered as real vectors

do
$x \leftarrow r_{ci} - r_{ai} + r_{bi}$
$C \leftarrow C \cup \{x\}$ // add x as a candidate rating
end for
$\hat{r}_{ui} = \underset{x \in C}{\text{aggr}}\ x$

Table 1. Performance of algorithms

	RMSE	Prec	Rec	Cov	$Surp^{max}$	$Surp^{avg}$	Time
Analogy	.898	89.1	43.3	31.2	0.433	0.199	2 h
kNN	.894	89.1	44.1	27.8	0.432	0.198	10 s
Bsln-kNN	.865	88.4	44.0	44.7	0.431	0.199	10 s

As expected, the Bsln-kNN algorithm is more accurate than the basic collaborative filtering method (KNN). The two classical collaborative algorithms perform better than the new proposed analogy-based method in terms of RMSE. Still, there seems to be some room for improvement for the analogical approach, with the help of a careful analysis of the behaviour of the algorithm.

As for performances other than RMSE, we see that the figures obtained by the three algorithms are quite close. Regarding surprise, which is a delicate notion to grasp, one may also wonder if the used measure is fully appropriate.

As usual, analogy-based algorithms suffer from their inherent cubic complexity. In the case of recommender systems where millions of users/items are involved, this is also a serious issue.

6 Conclusion and Future Research

This is clearly a preliminary study of an analogical approach to prediction in recommendation, a topic that has never been addressed before. First results are not better than the ones obtained with standard approaches. Even if they do not look that far, the difference is still significant enough to have a genuine impact on the users' experience. However, it is interesting to observe that approaches based on quite different ideas may lead to comparable results.

Besides, it should be recognized that the prediction part of the recommendation problem, although somewhat similar to a classification problem (for which analogical proportion-based classifiers are successful), presents major differences, since grades on items (playing here the role of descriptive features) may be both quite redundant and somewhat incomplete for providing a meaningful profile of a user. This may explain that the application of an analogical proportion-based approach is less straightforward in recommendation than in classification.

Analogical proportion-based methods have also been used recently for predicting missing Boolean values in databases [3]. The recommendation problem can be also viewed as a problem of missing values, but here the proportion of unknown data is very high, and data are not Boolean. This again suggests that the recommendation task is more difficult.

There are quite a number of issues to further explore, such as understanding on what types of situation an analogical proportion-based approach would perform better, and when another view is preferable. Another basic issue is the fact that users likely use the rating scale $\{1, 2, 3, 4, 5\}$ in an ordinal way rather than in an absolute manner, since the meaning of a rating may change with users. This calls for the use of analogical proportion between ordinal data [1].

Lastly, the ideas of the exploitation of the creative power of analogy for (i) proposing items never considered by a user, but having some noticeable common features with items (s)he likes [15], (ii) of the explanation power of analogy for suggesting to the user why an item may be of interest for him, are still entirely to explore.

References

1. Barbot, N., Miclet, L.: La proportion analogique dans les groupes. applications à l'apprentissage et à la génération. In: Proceedings Conference Francophone sur l'Apprentissage Artificiel (CAP), Hammamet, Tunisia (2009)
2. Bell, R.M., Koren, Y.: Lessons from the netflix prize challenge. SIGKDD Explor. Newsl. **9**(2), 75–79 (2007)
3. Correa Beltran, W., Jaudoin, H., Pivert, O.: Estimating null values in relational databases using analogical proportions. In: Laurent, A., Strauss, O., Bouchon-Meunier, B., Yager, R.R. (eds.) IPMU 2014, Part III. CCIS, vol. 444, pp. 110–119. Springer, Heidelberg (2014)
4. Gentner, D., Holyoak, K.J., Kokinov, B.N.: The Analogical Mind: Perspectives from Cognitive Science. Cognitive Science, and Philosophy. MIT Press, Cambridge (2001)
5. Herlocker, J.L., Konstan, J.A., Terveen, L.G., Riedl, J.T.: Evaluating collaborative filtering recommender systems. ACM Trans. Inf. Syst. **22**(1), 5–53 (2004)
6. Kaminskas, M., Bridge, D.: Measuring surprise in recommender systems. In: Adamopoulos, P., et al. (ed.) Proceedings of the Workshop on Recommender Systems Evaluation: Dimensions and Design (Workshop Programme of the 8th ACM Conference on Recommender Systems) (2014)
7. Lepage, Y.: De l'analogie rendant compte de la commutation en linguistique. Habilit. à Diriger des Recher., Univ. J. Fourier, Grenoble (2003)

8. McNee, S.M., Riedl, J., Konstan, J.A.: Being accurate is not enough: how accuracy metrics have hurt recommender systems. In: Olson, G.M., Jeffries, R., (eds.) Extended Abstracts Proceedings of the 2006 Conference on Human Factors in Computing Systems (CHI 2006), Montréal, Québec, Canada, 22–27 April, pp. 1097–1101 (2006)
9. Melis, E., Veloso, M.: Analogy in problem solving. In: Handbook of Practical Reasoning: Computational and Theoretical Aspects. Oxford University Press (1998)
10. Miclet, L., Prade, H.: Handling analogical proportions in classical logic and fuzzy logics settings. In: Sossai, C., Chemello, G. (eds.) ECSQARU 2009. LNCS, vol. 5590, pp. 638–650. Springer, Heidelberg (2009)
11. Prade, H., Richard, G.: Analogical proportions and multiple-valued logics. In: van der Gaag, L.C. (ed.) ECSQARU 2013. LNCS, vol. 7958, pp. 497–509. Springer, Heidelberg (2013)
12. Prade, H., Richard, G.: From analogical proportion to logical proportions. Log. Univers. **7**(4), 441–505 (2013)
13. Prade, H., Richard, G.: Homogenous and heterogeneous logical proportions. IfCoLog J. Log. Appl. **1**(1), 1–51 (2014)
14. Ricci, F., Rokach, L., Shapira, B., Kantor, P.B.: Recommender Systems Handbook. Springer, Cambridge (2011)
15. Sakaguchi, T., Akaho, Y., Okada, K., Date, T., Takagi, T., Kamimaeda, N., Miyahara, M., Tsunoda, T.: Recommendation system with multi-dimensional and parallel-case four-term analogy. In: Proceedings of IEEE International Conference on Systems, Man, and Cybernetics (SMC 2011), pp. 3137–3143 (2011)
16. Yvon, F., Stroppa, N.: Formal models of analogical proportions. Technical report D008, Ecole Nationale Supérieure des Télécommunications, Paris (2006)