Using Graph Partitioning Techniques for Neighbour Selection in User-Based Collaborative Filtering

Alejandro Bellogín
Information Retrieval Group
Department of Computer Science
Universidad Autónoma de Madrid
alejandro.bellogin@uam.es

Javier Parapar
Information Retrieval Lab
Department of Computer Science
University of A Coruña
javierparapar@udc.es

ABSTRACT

Spectral clustering techniques have become one of the most popular clustering algorithms, mainly because of their simplicity and effectiveness. In this work, we make use of one of these techniques, Normalised Cut, in order to derive a cluster-based collaborative filtering algorithm which outperforms other standard techniques in the state-of-the-art in terms of ranking precision. We frame this technique as a method for neighbour selection, and we show its effectiveness when compared with other cluster-based methods. Furthermore, the performance of our method could be improved if standard similarity metrics – such as Pearson's correlation – are also used when predicting the user's preferences.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Information Filtering, Clustering

General Terms

Algorithms, Experimentation, Performance

Keywords

Recommender systems, Collaborative Filtering, Clustering, Normalised Cut

1. INTRODUCTION

Collaborative Filtering (CF) is a particularly successful form of personalized Information Retrieval which suggests interesting items to users based on the preferences from similar-minded people [4, 9]. In CF, the most common form of ground user preference evidence consists of ratings, which are explicit relevance values given by users to items of interest. CF algorithms exploit the active user's ratings to make predictions, and thus it has the interesting property that no item descriptions are needed to provide recommendations, since it merely exploits information about past ratings between users and items. Moreover, it has the salient advantage that a user benefits from others' experience, being exposed to novel recommendations with respect to the latters' personal preferences. Note that this cannot be produced in general by other (content-based) approaches that tend to reproduce the user's past, insofar as they examine the preferences of individual users in isolation [1].

Collaborative filtering approaches can be classified into two main categories: model-based approaches and memory-based approaches. Model-based approaches learn user/item rating patterns to build statistical models that provide rating estimations. Memory-based approaches, on the other hand, compute user/item similarities based on distance and correlation metrics [3], and use these similarities to find similar-minded people of the active user. These people are usually called neighbours, and their preferences are used to predict ratings for the active user.

Memory-based CF algorithms are based on the principle that a particular user's rating records are not equally useful to all other users as input to provide them with item suggestions [4]. Thus, central aspects in these algorithms are how to identify which neighbours form the best basis to generate item recommendations for the active user, and how to properly account for the information provided by them. Typically, neighbourhood identification is based on selecting those users who are most similar to the active user according to a certain similarity metric. In this context, the similarity of two users generally consists of finding a set of items that both users have interacted with, and examining to what degree the users displayed similar behaviours on these items.

Once the active user's neighbours are identified, the more similar a neighbour is to the active user, the more her preferences are taken into account as input to make up recommendations. For instance, a common memory-based approach consists of predicting the relevance of an item for the active user by a linear combination of her neighbours' ratings, which are weighted by the similarity between each neighbour and the user. It is also a common practice to set a similarity threshold (or a maximum number of most similar users) to restrict the set of neighbours, in order to avoid the noisy disruption of long tails of dissimilar users. An instantiation of this algorithm can be formulated as follows [10]:

$$\hat{r}(u,i) = \bar{r}(u) + C \sum_{v \in N_k(u,i)} \text{sim}(u,v)(r(v,i) - \bar{r}(v)) \qquad (1)$$

This method establishes that the preference of a user u for a particular unseen item i is given by a numeric rating $r(u,i)$ estimated in the form of $\hat{r}(u,i)$. To provide that estimation,

RecSys 2012, September 9 - 13, Dublin, Ireland

the method takes into account the ratings $r(v,i)$ provided by the k users v who are most *similar* to u, usually called neighbourhood and denoted here as $N_k(u,i)$. The function $\text{sim}(u,v)$ measures the similarity between two users u and v, and the constant C is a normalization factor. Thus, the predicted rating of user u for item i is computed over the average rating $\bar{r}(u)$ (this step is not always required [1]), and the sum of u's similarities with her neighbours v, weighted by the deviations of v's ratings for i and average ratings $\bar{r}(v)$. The item-based algorithm can be described analogously [11].

Different ways for building the neighbourhood $N_k(u,i)$ have been proposed: some authors use the concept of *trust* by selecting only the most trustworthy users with respect to some trust metric [9]; other authors split the set of users or items in order to improve the scalability of the recommender systems and their accuracy [8, 15]. Most of the latter approaches use old-fashioned clustering methods such as k-Means or hierarchical clustering. Furthermore, in some situations, external information is used for the data partition, such as the content of the item (genres or tags, in the movie domain).

Cluster algorithms for neighbour selection in CF have not been widely exploited. The few existing approaches [15] produced good results but at the expenses of lower coverage. In this paper, we propose a clustering method which has shown good empirical performance properties in the fields of Information Retrieval and Data Mining and apply it to Recommender Systems. In this process, we provide a general formulation for cluster-based CF methods, framed as neighbour selection methods. We report empirical results confirming where our method outperforms both standard CF methods and other cluster-based algorithms, with the additional advantage that we rely exclusively on data extracted from the rating matrix, that is, no external information is used.

2. GRAPH PARTITIONING BASED CLUSTERING: NORMALISED CUT

Spectral clustering algorithms [14] use graph spectral techniques to tackle the clustering problem transforming it into a graph cut problem. The dataset to be clustered is typically represented as a weighted graph, $G = (V, E, W)$, where V is the set of objects to cluster, E is the set of edges between objects, and W denotes a diagonal matrix whose elements are the weights e_{ij} between vertices $v_i, v_j \in V$. The Normalised Cut ($NCut$) value of a certain cut (a partition of V) of a given graph was introduced by [13]. For a certain cut $w = \{A_1, A_2, ...A_k\}$ of a graph G, $NCut$ is defined as:

$$NCut(A_1, ...A_k) = \sum_{i=1}^{k} \frac{cut(A_i, \bar{A}_i)}{vol(A_i)} \quad (2)$$

where

$$cut(A,B) = \sum_{i \in A, j \in B} w_{ij}; \;\; vol(A) = \sum_{i \in A} \sum_{j=1}^{n} w_{ij} \quad (3)$$

here, A_1 to A_k are the connected components (ideally, the clusters) in which the graph has been divided and $\bar{A}_i$ are the vertices which are not included in A_i.

A graph cut with a low $NCut$ would represent a cut of the graph in which the weights of the edges which join vertices in different connected components are as low as possible while keeping the volumes of the resulting connected components as high as possible. This last condition ensures a certain balance between the connected components, trying to avoid trivial solutions. So, a cut of G with a low $NCut$ would correspond to a good clustering of the data.

The minimisation of $NCut$ can be presented as a matrix trace minimisation problem [14]. Let $H = (h_{i,j})$ be a $n \times k$ matrix which will be used to encode the membership of data points to the connected components. The j^{th} column of H contains the membership of connected component A_j (the indicator vector) encoded as follows:

$$h_{ij} = \begin{cases} \frac{1}{\sqrt{vol(A_j)}} & \text{if } v_i \in A_j \\ 0 & \text{else} \end{cases} \quad (4)$$

Also, let D be a $n \times n$ diagonal matrix such that $d_{i,i} = \text{degree}(v_i) = \sum_{j=1}^{n} w_{ij}$ and let L be the Laplacian matrix of graph G (that is, $L = D - W$). Using these matrices, it can be shown that the minimisation of NCut can be written as in Eq. (5)

$$\min_{A_1,...A_k} \text{Tr}(H^T L H) \text{ s.t. } H^T D H = I \quad (5)$$

It can be demonstrated that the condition of discreteness on the values of H makes the minimisation problem NP-Hard. If this discreteness of the values of H is relaxed, allowing the indicator columns composing that matrix to have any value in $\mathbb{R}^n$, and the substitution $Y = D^{\frac{1}{2}} H$ is performed, the expression in Eq. (6) is obtained:

$$min_{Y \in \mathbb{R}^{n \times k}} \text{Tr}(Y^T \left[D^{\frac{-1}{2}} L D^{\frac{-1}{2}} \right] Y) \text{ s.t. } Y^T Y = I \quad (6)$$

The above equation is in the standard form of a trace minimisation problem. Therefore, it can be demonstrated that this equation is minimised by the matrix Y which contains as columns the eigenvectors corresponding to the smallest eigenvalues of $D^{\frac{-1}{2}} L D^{\frac{-1}{2}}$.

Due to the relaxation of the condition of discreteness of the values of H to reduce the complexity of the problem and make it computationally affordable, instead of having an indicator vector for each connected component, we would have a vector in $\mathbb{R}^k$ for each datapoint (the rows of matrix Y). Thus, we have a projection of each data instance in $\mathbb{R}^k$ based on its similarity to the other instances. Hence, some other technique (such as k-Means) should be used to find a discrete segmentation of this space. Once this segmentation has been performed, we can backtrace each projected datapoint to the original one, obtaining the final outcome of the clustering algorithm.

3. NORMALISED CUT FOR RECOMMENDATION

The use of $NCut$ in the recommendation process is proposed as a tool for neighbour selection. Now, we focus on the methods derived when user clusters are used, alternatively, item clusters could be used in a straightforward way and the problem would be casted as an item-based CF method. Hence, in a user-based method, the users will play the role of the nodes of the graph to cut. A good set of neighbours for each user in the collection would be obtained by finding a good $NCut$ of the graph. Besides, in order to perform the complete process of recommendation, we also need to

establish a procedure for weighting the edges in the graph (e_{ij}), i.e. the distances among users. We decided to adopt the well-known – and traditionally used – Pearson's correlation similarity. Thus, for each cluster, we obtain a list of users belonging to such cluster. Then, we build a recommender which predicts the rating for user u and item i in the following way:

$$\tilde{r}(u,i) = \frac{\sum_{e \in NC(u)} \text{sim}(u,e) r(e,i)}{\sum_{e \in NC(u)} |\text{sim}(u,e)|} \quad (7)$$

where $NC(u)$ outputs the elements who belong to the same cluster as the target user u, $\text{sim}(u,e)$ represents the similarity between the element e and the current user; finally, $r(e,i)$ is the rating given by user e to item i.

Thus, we summarise our neighbour selection problem for a user-based method as follows. We use the Normalised Cut algorithm to create different user clusterings. Then, we use the information generated by this clustering for the neighbourhood formation, such that for each user u, we find the cluster c_u that user belongs to, then, the rest of the users belonging to c_u, are selected as the potential set of u's neighbours.

4. EXPERIMENT AND RESULTS

In order to empirically compare whether the clusters generated by the Normalised Cut technique are able to enhance standard collaborative recommendations, we have performed the following experiments. Firstly, we have plugged those clusters into a standard user-based CF method and compared their performance against some well-known state-of-the-art recommenders: a user-based CF with Pearson's correlation as similarity measure (UB [10]) and a matrix factorization algorithm with a latent space of dimension 50 (MF [6]). Secondly, in order to show whether the improvement comes from using a cluster-based method or the specific technique that we propose, we compare the performance of our technique against a standard clustering method (k-Means [7, 15]).

These experiments have been carried out using the publicly available dataset called *Movielens 100K*[1]. This dataset contains 943 users, 1,682 items and 100,000 ratings. We performed a 5-fold cross validation using the splits contained in the public package, these splits retain the 80% of the data for training, and the rest for testing. The methodology used in the evaluation corresponds to the *TestItems* approach described in [2], although alternative methodologies (such as the one described by Koren in [6]) have also been evaluated and similar results were obtained. More specifically, the TestItems methodology generates for each user a ranking by predicting a score for every item in the test set. Then, the performance of this ranking is measured using, for instance, the *trec_eval* program[2]. In this way, standard IR metrics such as precision, normalised Discounted Cumulative Gain (nDCG) or Mean Reciprocal Rank could be used.

4.1 Normalised Cut for Neighbour Selection

In order to asses our technique's ability to select good neighbours, we evaluate the *NCut* clustering technique (denoted as NC+P, since Pearson's correlation is used as similarity) by the Equation 7. We compared it with the standard UB method where the neighbourhood is selected among the set of most similar users. Besides, to put our results in perspective, we also include a well-known method which does not use any neighbour selection and which is typically among the best performing recommenders (MF method).

Table 1: Results for cluster-based algorithm using Normalised Cut (NC+P). In brackets, the number of eigenvectors used in the Normalised Cut for each k. Statistical significant improvements according to Wilcoxon Test ($p < 0.05$) w.r.t. MF and UB are superscripted with m and u respectively, best values are bolded.

Method	P@5	nDCG@5	Coverage
MF	0.081^{u}	0.076^{u}	100%
UB	0.026	0.020	100%
NC+P 50 (100)	0.075^{u}	0.062^{u}	100.00%
NC+P 100 (150)	0.101^{mu}	0.085^{u}	97.82%
NC+P 150 (200)	0.111^{mu}	0.095^{mu}	93.68%
NC+P 200 (250)	$\mathbf{0.112}^{mu}$	$\mathbf{0.097}^{mu}$	79.74%
NC+P 250 (300)	0.111^{mu}	$\mathbf{0.097}^{mu}$	69.06%
NC+P 300 (350)	0.108^{mu}	0.096^{mu}	59.26%
NC+P 350 (400)	0.103^{mu}	0.088^{mu}	54.25%
NC+P 400 (450)	0.094^{mu}	0.083^{u}	43.36%
NC+P 450 (500)	0.086^{u}	0.074^{u}	40.52%
NC+P 500 (550)	0.079^{u}	0.070^{u}	35.95%
NC+P 550 (600)	0.079^{u}	0.068^{u}	32.03%
NC+P 600 (650)	0.079^{u}	0.070^{u}	25.93%

The results are presented in Table 1, where different values for the size of the cluster (k) have been evaluated against the baselines. In fact, the quality of the results of the spectral methods improve considerably if, in the projection phase, instead of taking only the first k eigenvectors, a higher number of vectors (d) is used [5], for this reason we have taken higher number of eigenvectors (specified between brackets). Due to space constraints, only the results for a specific cut-off ($N = 5$) are reported, but other cut-offs and metrics such as P@50 and nDCG@50 were also tested and a similar trend was observed. Here, we show that cluster-based methods outperform baseline methods for specific values of the cluster size, namely $k = 150$ and 200, and these improvements are statistically significant. In fact only MF and in only one occasion (k=50) achieved statistical significant improvements over our approach. Furthermore, a larger number of clusters does not necessarily mean a better performance, probably because most of the clusters would contain zero or one element when this number is very large.

More importantly, a larger number of clusters tends to produce lower coverage values[3], since the information available to the recommender is not enough, which would produce a well performing method, but a not very interesting one, since it would only be able to suggest items – for some configurations – to less than the 30% of the system's population.

4.2 Sensitivity to the Number of Clusters

In this experiment, we analyse the recommendation performance variations that result when different cluster sizes

[1]Available at http://www.grouplens.org/node/73

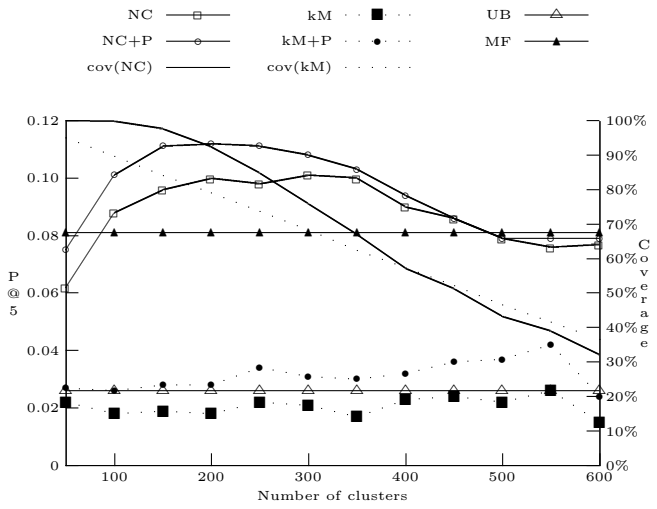
[2]Available at http://trec.nist.gov/trec_eval/

[3]We follow the definition given in [12] of *user space coverage*, that is, the number of users for which the system can recommend items.

are used in a cluster-based recommender method. More specifically, we compare our *NCut* method with a state-of-the-art algorithm where the clusters are generated using the k-Means method [15]. Furthermore, since our method has shown good performance once it has been plugged in the standard formulation of user-based CF (previous section), now we also evaluate its performance when no information about the user-neighbour similarity is used.

Figure 1 summarises our results. We denote as *NC* when the cluster-based recommender with $w_1(u, e) = 1$ and the NCut clustering technique are used, as in the previous section, when this technique is used in combination with the standard Pearson's similarity coefficient, we denoted it as *NC+P*. Similarly, *kM* and *kM+P* denote the corresponding k-Means cluster-based methods when a constant similarity $w_1 = 1$ and Pearson values are used, respectively.

Figure 1: Performance obtained (P@5) between the baseline methods (standard CF and cluster-based using k-Means alone or in combination with Pearson's similarity: *kM* and *kM+P*), the NCut method without a similarity function (*NC*), and the combination of a Pearson's similarity function and NCut as the neighbour selection method (*NC+P*). Coverage values are plotted along the secondary axis.



We can observe in the figure, first, that the coverage of both cluster-based methods is very similar, although our method obtains improvements up to 15% when less than 300 clusters are used. Interestingly enough, the performance of our technique reaches a maximum also at that point, whereas the k-Means cluster-based recommender shows a flat performance along the different number of clusters evaluated. It should be noted also that the performance of our method when no similarity information is used (curve *NC*) is comparable to that of *NC+P*, and thus, it becomes apparent that our method alone is improving the way the neighbours are being selected, which leads to a major performance enhancement.

5. CONCLUSIONS AND FUTURE WORK

In this paper, we have presented a new approach for neighbour selection in Recommender Systems. *NCut* is a clustering method based on graph partitioning which is naturally applied to the graph essence of the recommender systems communities. We tested two approaches of exploiting the neighbourhoods determined by the algorithm and both of them greatly improved the performance of standard CF methods. Furthermore, when compared against other cluster-based methods, our method obtained large improvements in performance, while the coverage was comparable to that of those methods.

Although our approach is general enough to work for user- and item-based CF methods, we have decided to focus on user-based methods since they provided better performance in initial experiments, but we aim to further investigate in the future also item clusters along with item-based approaches. We would also explore the behaviour of our proposal when working in larger datasets and the use of other spectral clustering techniques. Besides, we have observed that cluster-based methods tend to reduce the coverage of the system at the expense of its performance, hence, we aim to investigate alternative clustering methods which do not suffer from this behaviour.

6. ACKNOWLEDGMENTS

This work is supported by the Spanish Government (TIN2011-28538-C02-01), and the Government of Madrid (S2009TIC-1542).

7. REFERENCES

[1] Adomavicius, G., and Tuzhilin, A. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. Knowl. Data Eng. 17*, 6 (2005), 734–749.

[2] Bellogín, A., Castells, P., and Cantador, I. Precision-oriented evaluation of recommender systems: an algorithmic comparison. In *RecSys* (2011), pp. 333–336.

[3] Desrosiers, C., and Karypis, G. A comprehensive survey of neighborhood-based recommendation methods. In *Recommender Systems Handbook*. 2011, pp. 107–144.

[4] Herlocker, J. L., Konstan, J. A., and Riedl, J. An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms. *Inf. Retr. 5*, 4 (2002), 287–310.

[5] Jin, R., Ding, C. H. Q., and Kang, F. A probabilistic approach for optimizing spectral clustering. In *NIPS* (2005).

[6] Koren, Y. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *KDD '08* (2008), ACM, pp. 426–434.

[7] MacQueen, J. B. Some methods for classification and analysis of multivariate observations. In *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability* (1967), L. M. L. Cam and J. Neyman, Eds., vol. 1, University of California Press, pp. 281–297.

[8] O'Connor, M., and Herlocker, J. Clustering items for collaborative filtering. In *ACM SIGIR Workshop on Recommender Systems* (1999).

[9] O'Donovan, J., and Smyth, B. Trust in recommender systems. In *IUI* (2005), pp. 167–174.

[10] Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., and Riedl, J. Grouplens: An open architecture for collaborative filtering of netnews. In *CSCW* (1994), pp. 175–186.

[11] Sarwar, B., Karypis, G., Konstan, J., and Riedl, J. Item-based collaborative filtering recommendation algorithms. In *WWW '01* (2001), ACM, pp. 285–295.

[12] Shani, G., and Gunawardana, A. Evaluating recommendation systems. In *Recommender Systems Handbook*. 2011, pp. 257–297.

[13] Shi, J., and Malik, J. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell. 22*, 8 (2000), 888–905.

[14] von Luxburg, U. A tutorial on spectral clustering. Tech. Rep. TR-149, Max Planck Institute for Biological Cybernetics, 2006.

[15] Xue, G. R., Lin, C., Yang, Q., Xi, W., Zeng, H. J., Yu, Y., and Chen, Z. Scalable collaborative filtering using cluster-based smoothing. In *SIGIR* (2005), ACM, pp. 114–121.