International Journal of Scientific Research in Science, Engineering and Technology (www.ijsrset.com)
© 2020 IJSRSET | Volume 7 | Issue 2 | Print ISSN: 2395-1990 | Online ISSN : 2394-4099
DOI : https://doi.org/10.32628/IJSRSET207249

Analysis on Research Paper Publication Recommender System with Authors-Conferences Matrix

Htay Htay Win
Faculty of Information Science/University of Computer Studies (Taungoo) / Taungoo City, Bago Region, Myanmar

ABSTRACT

For years, achievements and discoveries made by researcher are made aware through research papers published in appropriate journals or conferences. Many a time, established s researcher and mainly new user are caught up in the predicament of choosing an appropriate conference to get their work all the time. Every scientific conference and journal is inclined towards a particular field of research and there is a extensive group of them for any particular field. Choosing an appropriate venue is needed as it helps in reaching out to the right listener and also to further one's chance of getting their paper published. In this work, we address the problem of recommending appropriate conferences to the authors to increase their chances of receipt. We present three different approaches for the same involving the use of social network of the authors and the content of the paper in the settings of dimensionality reduction and topic modelling. In all these approaches, we apply Correspondence Analysis (CA) to obtain appropriate relationships between the entities in question, such as conferences and papers. Our models show hopeful results when compared with existing methods such as content-based filtering, collaborative filtering and hybrid filtering.

Keywords : Recommender Systems, Machine Learning, Dimensionality Reduction, Correspondence Analysis, Topic Modelling, Linear Transformation, Author Social Network, Content Modelling

I. INTRODUCTION

With the advent of the Internet and the growing amount of information available therein, people are increasingly resorting to finding information online. This in rotate has resulted in a handful of challenges, one of the principal a single for users being finding perfectly what they are looking for or for researchers to keep up to date on information of whose existent they may be unconscious and other in [11].

In order to address this problem, we aim to build a recommender system that recommends the most appropriate publication venues for an author. This system is exceptionally useful to budding researchers who have very little knowledge about the research world and also to experienced researchers by saving a lot of their time and effort.

In this work, we aim to approach this problem in the settings of dimensionality reduction and topic modeling. We propose three different methods to recommend conferences for researchers to submit their paper based on the content of the paper and the social network of the authors: two of them involving content-analysis and the third one involving social network of the authors. Our approach is evaluated speculatively using the dataset of recent ACM conference publications and, to compare with existing methods such as content-based filtering, collaborative

filtering and hybrid filtering with promising results. However, there are several obstacles that need to be addressed in advanced. We list out the challenges along based on the massive literature survey.

1. Challenges**:** We face several challenges when working in this domain, as illustrated.

(a) In all the previous work done related to our problem, only a model using the social network of the authors has been employed. Content analysis of the papers in consideration, to the best of the authors' knowledge, has never been attempted. Just using the network of authors, without even looking at the paper, is not sufficient to decide where the paper should go to. We do incorporate content into our work.

(b) Suggesting conferences to new authors is a very tricky business. If the author has not published any paper before, he does not have a social network. Hence, the current systems would yield a poor recommendation. We are considering content of the paper lead to better results.

2. Main Claims: The abstracts of the papers will be in consideration for content analysis. The challenges raised above are systematically addressed as follows:

(a) It would be problematic on suggesting recommending conferences to authors with no prior social network. However, this problem might not arise during content-analysis as the author's social network is not in consideration. Just relying on the content of the abstract, we recommend suitable conferences. In our experiments, to suggest conferences to new authors, we observed that this method far supersedes the one relying on only his/her social network.

(b) Maximum essence of the relationship between the attributes in a table is obtained only in lower dimensional subspaces. Thus, when reducing the dimension of the matrices using CA, we essentially throw out the redundant information while maintaining the crucial and important part of them that are responsible for the relationships. As an added bonus, the reduced dimension increases the efficiency of the methods.

(c) In order to avoid such a confusion, our third method does not compose the two matrices. Instead a linear transformation is defined between the two spaces after reduction of dimension. In essence, after constructing the Paper × Words and Words × Conference matrices, we apply CA to each of them to reduce their dimension and then define a linear transformation from one subspace to the other for the process of recommendation.

3. Key tasks of the methods: The key tasks of each of the method proposed are as follows:

Method : Considering the content of the paper and composition of matrices.

- ✓ We construct a Paper × Words matrix and a Words × Conference matrix, where the $(i,j)^{th}$ entry of each of the matrices indicate the frequency of occurrence of word_j in paper_i and word i in the papers published in conference_j respectively.
- ✓ Then, we compose these two matrices and apply CA to obtain the principal column co-ordinates corresponding to the conferences.
- ✓ We obtain the principal row co-ordinates of the paper in need of a recommendation by computing it's tf-idf vector, composing with the Words × Conference training matrix and subsequent CA.
- ✓ The conference nearest to the paper in the bi-plot is recommended as the most suitable one.

II. TECHNICAL APPROACH

Different approaches can be taken to solve the considered problem of attempting to recommend conferences to authors. Outline of ideas are provided and their pitfalls, if any, are mentioned. This recommender system unlike most commercial ones like recommending books, movies etc.. involves people in some sense. Thus, there is an emotional connection involved. What this means is, if a conference suggested by our system gets a paper

rejected, it is highly unlikely that he will use this system again. This is not that case with books or movie recommenders. So, there is no room for errors and less accuracies. Some previous work on this has been done by H. Luong et al. [25] who have recommended conferences to authors using the social network i.e. the co-author network with the same dataset. Exploring the possibility of using CA has not been attempted before.

We have implemented a total of 6 methods for this application and have done a comprehensive evaulation of the results. Three of the methods use Correspondence Analysis and three of them don't. The first method uses the Author-conference relation without taking into account the content of the paper. The next two methods use the content along with an application of CA to arrive at the results. The abstracts of the paper are used for content-analysis. This makes sense because the essence of the entire paper is contained in the abstract.

The last three methods are respectively: Content-based filtering, Collaborative filtering and Hybrid filtering. Content-based filtering and Hybrid filtering use the content of the paper but none of these methods employ CA.

Content is obtained in two ways: term frequency-inverse document frequency (tf-idf) and topics. LDA has been used for the latter. For each content-method, number of topics used: 100, 200, 400, 600, 800 and 1000. However, only results for 400 topics are displayed in the evaluation, due to there being a very vast multitude of results and it would be too cumbersome to list all of them. Number of words used in tf-idf: 14082. For computing the resultant conferences, three methods of similarity have been used: euclidean distance, cosine similarity and pearson correlation.

In all the methods, 2008–2009 set of papers have been used for training and 2010 papers have been used for testing. There are a total of 5447 papers for the years 2008–2010, 3572 for 2008–2009 and 1875 for 2010.

There are a total of 16 conferences.
The various similarity metrics used in the experiments are given below:
Euclidean distance

$$d(x,y) = \sqrt{\sum_{k=1}^{n}(x_k - y_k)^2}$$

where n is the number of attributes and x_k and y_k are the k^{th} attributes of the data points x and y, respectively.

- ✓ Cosine Similarity: In this similarity measure, items are considered as n-dimensional document vectors and their similarity is measured as the cosine of the angle that they form between them. Thus, if the cosine measure is close to 1, i.e. the angle between the two vectors is close to 0, the items are considered to be very similar.

$$\cos(x,y) = \frac{(\mathrm{x}.\,y)}{\mathrm{II}x\mathrm{II}\ \mathrm{II}y\mathrm{II}}$$

where · indicates vector dot product and IIxII is the norm of vector x. This similarity is also known as the L_2 Norm.

- ✓ Pearson Correlation: Correlation between items can also measure their similarity, linear relationship in this case. Although several correlation coefficients can be used, the most commonly used one is the Pearson Correlation. Given the covariance of data points x and y, Σ, and their standard deviation σ, we compute the Pearson correlation using:

$$Pearson(x,y) = \frac{\Sigma(x,y)}{\sigma_x \times \sigma_y}$$

A. Using Authors-Conferences Matrix

a. Data Construction

From the data collected in the DBLP database, we construct the author-conference matrix, where each row represents a single author. Here f_{ij} represents the number of times author ai has published in conference c_j. We construct two such matrices: one training, say M_{train} and the other a test matrix, say M_{test}. The training matrix M_{train} is constructed from 2008–2009 papers (a total of 3572) and the test matrix M_{test}t is constructed from the 2010 papers (a total of 1875). There are a total of 16 conferences.

b. Applied Method

The algorithm followed is given in the following steps:

- ✓ We compute the standardized residual matrix S_{train} from M_{train}.
- ✓ We then obtain the coordinate matrices (both standard and principal for rows and columns), after decomposing S_{train} using SVD.
- ✓ Using the matrix M_{test}as a supplementary row matrix, we compute its principal coordinates using the standard column coordinates of M_{train}.

$$\begin{array}{c} \\ a_1 \\ a_2 \\ \vdots \\ a_N \end{array} \begin{array}{c} \begin{array}{cccc} c_1 & c_2 & \dots & c_M \end{array} \\ \begin{bmatrix} f_{11} & f_{12} & \cdots & f_{1M} \\ f_{21} & f_{22} & \dots & f_{2M} \\ \vdots & \vdots & & \vdots \\ f_{N1} & f_{N2} & \dots & f_{NM} \end{bmatrix} \end{array}$$

Figure 1: The Author-Conference matrix

- ✓ The rows of the supplementary test matrix M_{test} represent individual authors. So, to recommend a conference to a paper, which may be written by multiple authors: we take all the authors of that particular paper and compute the similarity (euclidean distance/cosine/ pearson) with each of the 16 conferences. For this purpose, we use the principal coordinates of the authors and the principal coordinates of the conferences.
- ✓ We sort the conferences, which maximize the sum of the similarity to all the authors of the paper in consideration, in decreasing order. Maximizing similarity means: minimizing euclidean distance/maximizing cosine similarity/maximizing pearson correlation.
- ✓ We then get a ranked list of recommendations for each paper.

This method has several drawbacks. For one, all the new authors (new to these conferences) are all recommended the same conference. Thus, this approach fails if the author has no publication history. Also, this does not capture the essence of the problem because we are recommending without even looking at the content of the paper in question. Thus, we need to look at the content of the paper as well in order to make better and more appealing recommendations.

Here, we have considered each row to be a single author. It can also be changed to comprise of multiple authors i.e. who have co-authored a paper. In this case, there will be more number of entries in the matrix and also it will be more sparse. Even in this case, the same limitations as above apply and in addition, the sparsity, in some sense, also reduces the "meaningfulness" between the authors and conferences. Applying a dimensionality reduction technique like SVD or CA will bring it to a lower-dimensional subspace which will capture the essence of the relation better, rendering the matrix less sparse.

III. EVALUATION AND RESULTS

In this section, we detail the evaluation procedures and discuss the results obtained. We have used a total of 7 metrics to evaluate the performance of the algorithms described above. They were applied on the ranked list of recommendations generated by the above methods:

- ✓ Mean Precision at *K* (MP@*K*): The mean Precision at K for a set of queries is defined as the mean of the Precision at K values for each of those queries. Precision at *K*, *P(K)*, is defined as:

$$P(K)=\frac{\text{No.of relevant documents retrieved in the top K results}}{K}$$

- ✓ Mean Recall at K (MR@K): The mean Recall at K for a set of queries is defined as the mean of the Recall at K values for each of those queries. Recall at K, $R(K)$, is defined as:

$$\text{R(K)}=\frac{\text{No.of relevant documents retrieved in the top K results}}{\text{Total number of relevant documents}}$$

- ✓ Mean Average Precision at K (MAP@K): Mean average precision at K for a set of queries is the mean of the average precision at K values for each of those queries.

$$\text{MAP} = \frac{\sum_{q=1}^{Q} \text{AveP}(q)}{Q}$$

where Q is the number of queries. Here $AveP(q)$ is the average precision for the q^{th} query. Average precision is defined as:

$$\text{AveP} = \frac{\sum_{k=1}^{n} \text{AveP}(P(k) \times rel(k))}{\text{no. of relevant documents}}$$

where rel(k) is an indicator function equaling 1 if the item at rank k is a relevant document, zero otherwise. $P(k)$ is the precision at k.

- ✓ Mean Normalized Discounted Cumulative Gain at P (MNDCG@P): Discounted Cumulative Gain (DCG) at P is defined as:

$$DCG_P = \sum_{i=1}^{p} \frac{2^{rel_i} - 1}{log_2(i + 1)}$$

where irel_i is the relevance score of result i. DCG uses a graded scale of relevance and this allows us to have preferences in the predicted results. Let us assume an ideal sequence of predicted results which would yield the maximum DCG_P. We call this the ideal DCG_P, denoted by $IDCG_P$. The normalized DCG_P, $NDCG_P$, is the ratio of the obtained $DCG_P$with that of the ideal $IDCG_P$. This would thus always yield a value between 0 and 1. The mean normalized DCG_P for a set of queries is then the mean of the $NDCG_P$ values for each of those queries.

- ✓ Mean Reciprocal Rank (MRR): The reciprocal rank of a query response is the multiplicative inverse of the rank of the first correct answer. The mean reciprocal rank is the average of the reciprocal ranks of results for a sample of queries Q:

$$\text{MRR} = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{rank_i}$$

- ✓ Mean F-Measure at K (MF-M): The mean F-measure at K for a set of queries is the mean of the F-measures at K for each of those queries. F-measure is defined as the harmonic mean of precision and recall:

$$F=\frac{2.\text{precision.recall}}{(\text{Precision+recall})}$$

This is the balanced F-score, where the weights of precision and recall in the harmonic mean are equal. We can also have cases of uneven weights.

- ✓ Mean *R-Precision* (MR-P): The mean R-Precision for a set of queries is the mean of the R-Precision values for each of those queries. R-Precision is defined as the Precision at R, where R is the number of relevant documents. At this position, the precision and recall values become equal.

For the experiments, we have chosen the value of K and P to be 5. This means that the measures are evaluated (which are @K and @P) considering only the top 5 of the returned results. For the purpose of calculating the metrics, we have defined relevant conferences in two cases:

1. A predicted conference is relevant if it is same as the actual conference the paper was originally published in (we have that information from the 2010 data set). For computing DCG in this case, the relevant conference (which is the original conference) is given a score of 1 and the rest are given scores 0.

2. A predicted conference is relevant if it belongs to the Special Interest Group (SIG) of the actual conference the paper was originally published in. For computing DCG in this scenario, the original

conference is given a score of 2, the other conferences in the SIG are given a score of 1 as they are considered to be partially relevant. The rest of the conferences get a score of 0.

For calculating similarity to determine the ranking of the retrieved results, we have used three different metrics as previously mentioned:

- ✓ Euclidean Distance
- ✓ Cosine Similarity
- ✓ Pearson Correlation

Table 1: Experimental Parameters for LDA

Parameter	Parameter
Number of Iterations	1000
Dirichlet Prior α	0.5
Number of Topics	400
Number of Training Papers	3572
Number of Test Papers	1875

Earlier it was explained that the dimension of the lower-dimensional subspace for an I ×J matrix is ≤ min{I −1, J −1}. Since, we have only 16 conferences and more than 1000 papers, the minimum is always 15. Although the experiments were evaluated for more than one subspace, due to lack of space and vast multitude of results, we only show the results for a 10-dimensional subspace. We call this d. In the case of third method (Linear Transformation), we reduce two matrices independently using CA and hence each can be reduced to a different dimensional subspace. So, for that method, we show the results for d1 = 10, d2 = 10,100, where d1 is the dimension of the subspace that the Conference x Words/Topics is reduced to and d2 is the dimension of the subspace that the Paper x Words/Topics is reduced to.

The experimental parameters used for LDA . For tf-idf, 14082 words were used. For displaying the results of the experiments, the following conventions are used

- ✓ MAP@5: Mean Average Precision at 5
- ✓ MNDCG@5: Mean Normalized Discounted Cumulative Gain at 5
- ✓ MRR: Mean Reciprocal Rank
- ✓ MR-P: Mean R-Precision
- ✓ MF-M: Mean F-Measure
- ✓ MP@5: Mean Precision at 5
- ✓ MR@5: Mean Recall at 5

Using the above conventions, the evaluations of the experiments are given below:

Method: Using Author-Conference Matrix

Here, we present the results for the first method. In this case, evaluation has been conducted with two matrices. The first matrix is the one constructed from the 2010 test dataset. The second matrix is a null matrix (all entries are 0). The second matrix is required for testing because many authors are common in the training and testing set and it is highly likely that an author, if published in a certain conference, would prefer to publish in it again.

Metrics	Euclid		Cosine		Pearson	
	Actual	SIG	Actual	SIG	Actual	SIG
MAP@	0.948	0.638	0.943	0.638	0.948	0.608
MNDCG@55	0.961	0.839	0.963	0.839	0.961	0.839
MRR	0.948	0.991	0.944	0.991	0.948	0.961
MR-P	0.905	0.657	0.900	0.657	0.905	0.617
MF-Mt 5	0.332	0.585	0.338	0.585	0.332	0.505
MP@5	0.199	0.525	0.197	0.525	0.199	0.525
MR@5	0.998	0.651	0.995	0.651	0.998	0.631

Table 2 : Results for Method: Considering the test matrix to be built from 2010 papers, *d*=0

Metrics	Euclid		Cosine		Pearson	
	Actual	SIG	Actual	SIG	Actual	SIG
MAP@	0.202	0.223	0.202	0.223	0.202	0.253
MNDCG@55	0.304	0.322	0.302	0.322	0.304	0.392
MRR	0.254	0.422	0.258	0.422	0.254	0.432
MR-P	0.019	0.331	0.016	0.331	0.019	0.311
MF-M@ 5	0.201	0.390	0.203	0.390	0.201	0.340
MP@5	0.120	0.356	0.128	0.356	0.120	0.346
MR@5	0.604	0.443	0.600	0.443	0.604	0.433

Table 3 : Results for Method: Considering the test matrix to be a zero (null) matrix, $d = 10$

Hence, this gives very high accuracy. The only way to really put the method to the test it to consider a new paper, which has not been published in any of the conferences mentioned and then recommend. This is why we considered a null matrix. The results are given below:

- ✓ Case 1: Using 2010 test matrix, d = 10. The results are displayed in Table 2.
- ✓ Case 2: Using null test matrix, d = 10. The results are displayed in Table 3.

As can be seen from the above results, when the input is a null matrix, the method performs poorly.

IV. CONCLUSION

Although each of the aforementioned procedures has its own advantages, from the surveys obtained, we observe the following:

The content-based methods proposed easily beat popular methods like collaborative filtering. This shows that for this system, considering content is vital. Computing similarities with content in hybrid filtering also did not prove to be very helpful, as the remainder of the procedure is identical to collaborative filtering.

Content-based filtering is seen to outperform the CA-based methods. This may be attributed to the fact that there is a certain amount of information loss during the dimensionality reduction phase, while content-based filtering utilizes the "pure" raw content.

In the results obtained, using tf-idf for content proved to be better than using topics. This may be due to considering a much larger number of words in tf-idf representation (14082) than it's topic counterpart (400). Also, the method of generating the topic matrices may have influenced the results.

Lastly, we observe that cosine similarity proves to be the best measure to calculate the similarities.

V. REFERENCES

[1]. Marko Balabanovi´c and Yoav Shoham. Fab: content-based, collaborative recommendation. Communications of the ACM, 40(3):66–72, 1997.

[2]. Michael Pazzani and Daniel Billsus. Learning and revising user profiles: The identification of interesting web sites. Machine learning, 27(3):313–331, 1997.

[3]. Xun Zhou, Jing He, Guangyan Huang, and Yanchun Zhang. A personalized recommendation algorithm based on approximating the singular value decomposition (approsvd). In Proceedings of the The 2012 IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology-Volume 02, pages 458–464. IEEE Computer Society, 2012.

[4]. Robert M Bell, Yehuda Koren, and Chris Volinsky. The bellkor solution to the netflix prize, 2007.

[5]. G´abor Tak´acs, Istv´an Pil´aszy, Botty´an N´emeth, and Domonkos Tikk. A unified approach of factor models and neighbor based

methods for large recommender systems. In Applications of Digital Information and Web Technologies, 2008. ICADIWT 2008. First International Conference on the, pages 186–191. IEEE, 2008.

[6]. Arkadiusz Paterek. Improving regularized singular value decomposition for collaborative filtering. In Proceedings of KDD cup and workshop, volume 2007, pages 5–8, 2007.

[7]. ON Osmanli and IH Toroslu. Using tag similarity in svd-based recommendation systems. In Application of Information and Communication Technologies (AICT), 2011 5th International Conference on, pages 1–4. IEEE, 2011.

[8]. Ken Goldberg, Theresa Roeder, Dhruv Gupta, and Chris Perkins. Eigentaste: A constant time collaborative filtering algorithm. Information Retrieval, 4(2):133–151, 2001.

[9]. Manolis G Vozalis and Konstantinos G Margaritis. Using svd and demographic data for the enhancement of generalized collaborative filtering. Information Sciences, 177 (15):3017–3037, 2007.

[10]. Manolis G Vozalis and Konstantinos G Margaritis. A recommender system using principal component analysis. Current Trends in Informatics, 1:271–283, 2007.

[11]. Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Application of dimensionality reduction in recommender system-a case study. Technical report, DTIC Document, 2000.

[12]. Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Incremental singular value decomposition algorithms for highly scalable recommender systems. In Fifth International Conference on Computer and Information Science, pages 27–28. Citeseer, 2002.

[13]. Panagiotis Symeonidis. Content-based dimensionality reduction for recommender systems. In Data Analysis, Machine Learning and Applications, pages 619–626. Springer, 2008.

[14]. Markus Zanker, Matthias Fuchs, Wolfram H¨opken, Mario Tuta, and Nina Mu¨ller. Evaluating recommender systems in tourism—a case study from austria. Information and communication technologies in tourism 2008, pages 24–34, 2008.

[15]. Steven Bethard and Dan Jurafsky. Who should i cite: learning literature search models from citation behavior. In Proceedings of the 19th ACM international conference on Information and knowledge management, pages 609–618. ACM, 2010.

[16]. C Lee Giles, Kurt D Bollacker, and Steve Lawrence. Citeseer: An automatic citation indexing system. In Proceedings of the third ACM conference on Digital libraries, pages 89–98. ACM, 1998.

[17]. Kannan Chandrasekaran, Susan Gauch, Praveen Lakkaraju, and Hiep Phuc Luong. Concept-based document recommendations for citeseer authors. In Adaptive Hypermedia and Adaptive Web-Based Systems, pages 83–92. Springer, 2008.

[18]. Xiang Chen, Cheng-Zen Yang, Ting-Kun Lu, and Hojun Jaygarl. Implicit social network model for predicting and tracking the location of faults. In Computer Software and Applications, 2008. COMPSAC'08. 32nd Annual IEEE International, pages 136–143. IEEE, 2008.

Cite this article as :

Sh

Htay Htay Win, "Analysis on Research Paper Publication Recommender System with Authors - Conferences Matrix ", International Journal of Scientific Research in Science, Engineering and Technology (IJSRSET), Online ISSN : 2394-4099, Print ISSN : 2395-1990, Volume 7 Issue 2, pp. 195-202, March-April 2020. Available at doi : https://doi.org/10.32628/IJSRSET207249
Journal URL : http://ijsrset.com/IJSRSET207249