Paper 1887-2014

Recommending TV Programs using Correlation

Rajendra LVN[1], Qing Wang[2] and John Dilip Raj[1]
[1]GE Capital Retail Finance, [2]Warwick Business School

ABSTRACT

Personalized recommender systems are being used in many industries to increase customer engagement. In the TV industry, this is primarily used to increase viewership, which in turn increases market share, revenue and profit. This paper attempts to develop a recommender system using the Correlation procedure under Collaborative Filtering Methodology. The only data requirement for this recommendation system would be past viewership of customers for a given time period.

INTRODUCTION

TV channels find much value in recommending programs/movies etc. to customers. Recommendation systems for the same serve three key functions:

(i) *Drive customer engagement*: Many customers find it convenient when they are recommended things to view. Knowing what other like customers have viewed gives a certain direction on next program/movie viewership.

(ii) *Hastening viewership of other programs*: Though many customers may eventually view the other programs during the repeat telecast or after a certain period of time, tailoring offers, advertisements may lead them to view these earlier than usual.

(iii) *Keeping viewership within the channel*: While customers may have planned what they are going to view, it is not certain that they will do so only in one particular channel. Recommendation systems can therefore encourage customers to not look outside the channel for such programs.

There have not been many publications where SAS® has been used for increasing viewership through recommendation engines and this is an attempt to address the same.

COLLABORATIVE FILTERING

Of the many techniques available to recommend programs to customers, collaborative filtering has proved to be one of the most successful techniques in recent years (Li, Zhang and Wang 2013). Huang, Zeng and Chen (2007) mention other areas such as algorithm design, human- computer interaction design, consumer incentive analysis, and privacy protection where Collaborative Filtering techniques are being advanced.

PROC CORR TO DEVELOP A RECOMMENDATION ENGINE

In SAS®, a great tool for data screening, exploratory analysis and modeling is PROC CORR (Hill 2009).

The following is a step by step approach in developing a recommendation system using PROC CORR and other functions in SAS®.

Step 1: Input the transaction level dataset

We use a hypothetical dataset with information of 5 customers with respect to their viewership across 4 programs as in Table 1. The absolute number of times they viewed a program is not taken but only data to indicate whether the customer has viewed a particular program or not in the given time period.

Table 1: Customer – Prior TV Program Viewership Matrix

Customer ID	Program 1	Program 2	Program 3	Program 4
1	0	1	0	1
2	0	1	1	1
3	1	0	1	0
4	1	0	0	1
5	1	0	0	0

Step 2: Apply PROC CORR to find relationship between Programs

A correlation gives us the results as in Table 2.

Table 2: TV Program Level Similarity

	Program 1	Program 2	Program 3	Program 4
Program 1	-1	-1	-0.16667	-0.66667
Program 2	-1	-1	0.16667	0.66667
Program 3	-0.16667	0.16667	-1	-0.16667
Program 4	-0.66667	0.66667	-0.16667	-1

A few facts worth noting emerge from the table above:

Sets of Programs (in the above example, Program 1 and 2), that have not been viewed by any customer in the given time period, but either of the two Programs having been viewed by all the customers, are imputed to get a correlation value of '-1'. If a set of

Programs have been viewed by all customers then they get a value of ‘1’. In the real time environment, both these extremes may not happen.

Only one customer (with Customer ID 3) viewed Programs 1 and 3. Similarly, only one customer (with Customer ID 4) in the above example viewed Programs 1 and 4. At the outset, it seems that the relationship of Program 1 with Programs 3 and 4 are same. However, the correlations are not the same as not only similarity in view but also similarity in non-view is considered. There was one customer (with Customer ID 1) who did not view both Program 1 and 3. But there is no such customer who did not make a view in both Programs 1 and 4. Therefore, the similarly between Programs 1 and 3 is higher (-0.16667) than that of Programs 1 and 4 (-0.66667).

Step 3: Targeting customers using recommendation strength of Programs

Now that the similarities of Programs are estimated, the next step is to roll up these into recommendations at customer level. For this purpose, we use the algorithms as detailed by Deshpande and Karypis (2004) and convert them into SAS® codes using a matrix multiplication (Table 1 * Table 2). The recommendations are the sum of similarities of Programs they viewed previously. The recommendation strength of each Program at customer level is given in Table 3.

Table 3: Recommendation Strength of Programs at Customer Level

Customer	Program 1	Program 2	Program 3	Program 4
1	-1.66667	-0.33333	0	-0.33333
2	-1.83333	-0.16667	-1	-0.5
3	-1.16667	-0.83333	-1.16667	-0.83333
4	-1.66667	-0.33333	-0.33333	-1.66667
5	-1	-1	-0.16667	-0.66667

Table 4 further provides information at a level where marketers can choose how many Programs need to be targeted for each customer. Ranked list of items also can help consumers in making view decisions as per their personal taste (Ning and Karypis 2012). This technique does not just recommend Programs viewed in the previous time period. Evidence of this is seen by the first Program preference of customer 4, who had not viewed Program 2 in the previous time period, but which comes out as the first preference through this recommendation.

Table 4: Customer Targeting by Recommendation Strength

Customer ID	First Program Preference	Second Program Preference	Third Program Preference	Fourth Program Preference	Strength of First Program Preference	Strength of Second Program Preference	Strength of Third Program Preference	Strength of Fourth Program Preference
1	3	2	4	1	0	-0.33333	-0.33333	-1.66667
2	2	4	3	1	-0.16667	-0.5	-1	-1.83333
3	2	4	1	3	-0.83333	-0.83333	-1.16667	-1.16667
4	2	3	1	4	-0.33333	-0.33333	-1.66667	-1.66667
5	3	4	1	2	-0.16667	-0.66667	-1	-1

Research has shown offering additional/new Programs are important for an organization's success (Zboja and Hartline 2012). Marketers wanting to encourage cross-viewership may well find this technique of much use. If the business decision is to encourage view of Programs not viewed before/recently, table 5 shows how the recommendation engine can be customized to target customers for this purpose. Here, items viewed before/recently are dropped from the recommendations and only those not viewed before/recently and having recommendation strength are recommended to customers. This enables channels to estimate the unknown preferences of customers and try to make their channel(s) the choice of view for these.

Table 5: Targeting Programs not Viewed Before/Recently by Recommendation Strength

Customer ID	First Non-viewed Program Preference	Second Non-viewed Program Preference	Third Non-viewed Program Preference	Strength of First Non-viewed Program Preference	Strength of Second Non-viewed Program Preference	Strength of Third Non-viewed Program Preference
1	3	1	.	0	-1.66667	.
2	1	.	.	-1.83333	.	.
3	2	4	.	-0.83333	-0.83333	.
4	2	3	.	-0.33333	-0.33333	.
5	3	4	2	-0.16667	-0.66667	-1

CONCLUSION and FURTURE DIRECTIONS

With channels looking to increase engagement of customers, Program recommendation systems are becoming a must have feature for effective marketing campaigns. The procedure detailed is one way of implementing such a system using SAS®.

Recent research has also shown new approaches outperforming several collaborative filtering and attribute-based preference models (Chung and Rao 2012; Hennig-Thurau, Marchand and Marx 2012). A powerful tool like SAS® can add value in further providing these services for channels and others.

REFERENCES

Chung, J. & Rao, V. R. (2012), ‘A General Consumer Preference Model for Experience Programs: Application to Internet Recommendation Services’, *Journal of Marketing Research*, 49(3), 289-305.

Deshpande, M. & Karypis, G. (2004), ‘Item-Based Top-*N* Recommendation Algorithms’, *ACM Transactions on Information Systems*, 22(1), 143-177.

Hennig-Thurau, T., Marchand, A. & Marx, P. (2012), ‘Can Automated Group Recommender Systems Help Consumers Make Better Choices?’, *Journal of Marketing*, 76(5), 89-109.

Hill, A. (2009), ‘Customizing a Correlation Matrix that would make Pearson Proud’, *Proceedings of the 2009 NESUG Conference*.

Huang, Z., Zeng, D. & Chen, H. (2007), ‘A Comparison of Collaborative-Filtering Recommendation Algorithms for E-commerce’, *IEEE Intelligent Systems*, 22(5), 68-78.

Li, H., Zhang, S. & Wang, X. (2013), ‘A Personalization Recommendation Algorithm for E-Commerce’, *Journal of Software*, 8(1), 176-183.

Ning, X. & Karypis, G. (2012), ‘Sparse Linear Methods with Side Information for Top-N Recommendations’, *6th ACM Recommender Systems Conference*.

SAS Institute Inc. "PROC CORR Procedure" Base SAS® 9.3 Procedures Guide: Statistical Procedures. http://support.sas.com/documentation/cdl/en/procstat/63963/HTML/default/viewer.htm#procstat_corr_sect004.htm

Zboja, J. J., & Harline, M. D. (2012), ‘An Examination of High-Frequency Cross-Selling’, *Journal of Relationship Marketing*, 11, 41-55.

APPENDIX:

```
/*------------------------------------------------------------------------------*/
/*                  DATA REQUIRED/ INPUT DATASET               */
/*------------------------------------------------------------------------------*/

DATA Data;
Input customer_id m_1 m_2 m_3 m_4;
Cards;
1 0 1 0 1
2 0 1 1 1
3 1 0 1 0
4 1 0 0 1
5 1 0 0 0
;

/*Recommendation - Using Correlations*/

Proc Corr data = data
                    outp=Corr(drop=_TYPE_
rename=(_NAME_ = Variable)  where=(Variable~='')) noprint;
Var M_1 - M_4 ;
Run;
```

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Rajendra LVN
GE
iLabs Building, Block 1A, Hitech City,
Madhapur, Hyderabad – 500081
India
Email: rajendra.ledallavenkatanaga@ge.com