CASE STUDY 3 : SOCIAL RECOMMENDATION

RATINGS, TRUST & PERFORMANCE EVALUATION

Author : *Paula Dwan*
Email : *paula.dwan@gmail.com*

Student ID : *13208660*

Course : *MSc Advanced Software Engineering*
Module : *COMP-47270 Computational Network Analysis and Modelling*

Lecturer : *Dr. Neil Hurley*
Email : *neil.hurley@ucd.ie*

Due Date : *11 May 2015*

TABLE OF CONTENTS

1 Case Study Requirements

1.1 Case Study Requirements

Devise and evaluate an algorithm that combines ratings data and trust data to make a recommendation.

Epinions data is provided (... other data can be downloaded from the web).

1. Compare your algorithm with and without the trust data?
2. Does the trust data data improve the recommendation?

1.2 Understanding of Requirements from Class Discussions

1. Use the Collaborative Filtering algorithm as covered in class with any updates made as ***Rating*** recommendation algorithm.
2. Device another for Trust Ratings.
3. Evaluate the results.
4. Provide suggestions / conclusions.

2 Introduction

2.1 Social Recommendation

To start, what is Social Recommendation?

In real-life, the equivalent is personal recommendations from friends, colleagues and family on such items as whether or not to try a new restaurant, what hotel to use if visiting an area visited previously by someone known to the individual, what school to send a child to, what book to read next, if the latest block-buster film is worth seeing in cinema or should one wait on the DVD.

In social networks, social recommendation is using personal information on the user (publicly available or harvested by the social network) to provide a unique and personalized recommendation to the user. There is a wealth of information available on anyone with a social media / web presence (email, purchasing history, browser history, facebook, twitter, instagram, blogs, newspaper comments on articles, queries / answers on professional sites). This data may be mined and learned from to provide strong, detailed recommendation to the user.

2.2 Datasets Used

Each of the directed graphs chosen deal with Social media and vary in the number of nodes present. For more information on each, please see the section : References – Datasets

2.3 How Recommendations Work?

Recommendations are usually created by :

Collaborative Filtering	Collects and analyse a large amount of user information covering common user behaviours and on-line activities. Predicts what users will then prefer using similarity to other users.
Content Based Filtering	Uses the item description and users known preferences to suggest other items the user would like. So if a user liked ITEM X previously and item is similar to ITEM Y then the user will also probably like ITEM Y.

Recommendations are usually improved by adding **Trust**. Existing user rate other users on a pre-agreed scale indicating level of trust he/she has in other users. If USER 1 trusts USER 2 and USER 3 also trusts USER 2, then USER 1 may also trust the recommendations of USER 2.

There are many ways of doing this including :

Local Trust	Use the personal views of the users to predict different trust values in all other users. Examples include : MoleTrust (Paolo Massa et al, 2007) and TidalTrust (Jennifer Ann Golbeck, 2005).
Global Trust	Global reputation value for each user is used. Basically, an *average* rating for each user based on how the entire community views that user. Examples include : PageRank (Google) and Eigentrust ().

Performance results are evaluated using :

	Actual (Expectation)	
Predicted (Observed)	**TP** ***(true positive)*** Correct result	**FP** ***(false positive)*** Unexpected result
	FN ***(false negative)*** Missing result	**TN** ***(true negative)*** Correct null result

Recall

Fraction of the recommended set that includes relevant items.

$$\textbf{Recall} = \frac{TP}{TP + FN}$$

Precision

Fraction of the recommended set that includes relevant items.

$$\textbf{Precision} = \frac{TP}{TP + FP}$$

For precision and recall, as N increases recall increases towards 1 but precision decreases towards 0. Performance is also calculated using :

Mean Absolute Error (MAE)

How close is the predictions to the eventual outcomes.

$$MAE = \sum_{(a,i)\in T} \frac{|r_{a,i} - p_{a,i}|}{|T|}$$

Root Mean Squared Error (RMSE)

Differences between predicted values by a model and the actual values.

$$RMSE = \sqrt{\sum_{(a,i)\in T} \frac{(r_{a,i} - p_{a,i})^2}{|T|}}$$

2.3.1 RATINGS : COLLABORATIVE FILTERING

Overview	Use a simple Collaborative Filtering algorithm.
Calculation	Simple Collaborative Filtering as completed in Jan 2015.

2.3.2 RATINGS : COLLABORATIVE FILTERING

Overview	Use a Top N Collaborative Filtering algorithm for both training sets test sets.
Calculation	Collaborative Filtering with Pearson's Coefficient included. Calculate Pearson's for all test data sets against training data set. Use the test data set with the best Pearson's coefficient result. Calculate Top N for the test data set. Compare the results to those for the training data set and calculate overall MAE.

Overview	Use a Top N Collaborative Filtering algorithm for both training sets test sets.

Pearson's Coefficient

If r =

+ .70 or higher	Very strong positive relationship
+ .40 to +.69	Strong positive relationship
+ .30 to +.39	Moderate positive relationship
+ .20 to +.29	weak positive relationship
+ .01 to +.19	No or negligible relationship
- .01 to -.19	No or negligible relationship
- .20 to -.29	weak negative relationship
- .30 to -.39	Moderate negative relationship
- .40 to -.69	Strong negative relationship
- .70 or higher	Very strong negative relationship

2.3.3 TRUST : PAGE-RANK (GOOGLE)

Overview

Page rank of a web page is a number which assigns weight or importance of the page bsed on in-bound and out-bound connections. According to the Page Rank algorithm a webpage is more important if many other pages point to it.

Say we have : six web pages [P1, P2, , P3, P4, P5, P6]

each of which connects to itself and to all others, thus the L and A matrices are :

$$A = \begin{array}{c|cccccc} & P1 & P2 & P3 & P4 & P5 & P6 \\ \hline P1 & 1 & 0 & 0 & 0 & 0 & 0 \\ P2 & 1 & 1 & 0 & 0 & 0 & 0 \\ P3 & 1 & 1 & 1 & 0 & 0 & 0 \\ P4 & 1 & 1 & 1 & 1 & 0 & 0 \\ P5 & 1 & 1 & 1 & 1 & 1 & 0 \\ P6 & 1 & 1 & 1 & 1 & 1 & 1 \end{array} \qquad L = \begin{array}{c|cccccc} & P1 & P2 & P3 & P4 & P5 & P6 \\ \hline P1 & 0 & 1 & 1 & 1 & 1 & 1 \\ P2 & 0 & 0 & 1 & 1 & 1 & 1 \\ P3 & 0 & 0 & 0 & 1 & 1 & 1 \\ P4 & 0 & 0 & 0 & 0 & 1 & 1 \\ P5 & 0 & 0 & 0 & 0 & 0 & 1 \\ P6 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}$$

Adjacency matrix *Inverse Adjacency matrix*

Page rank formula is :

$$\mathbf{Pi} = (1-d) + d \sum_{i=1}^{N} \left(\frac{L_{ij}}{C_j}\right) P_j$$

where :

L_{ij} :	Link exists from i → j? if true = 1, else = 0
N :	Total number of pages
Cj :	Number of outbound links
d :	Positive constant = 0.85

2.3.4 TRUST : SIMPLE MAPPING – USER [I] TRUSTS USER [J]?

Overview

Page rank applies on a global level, in this instance a more localised approach would be better. Thus I went for the simplest possible implementation :

> Does USER [I] trust USER [J]?
>
> If [YES] then use the rating, if [NO] don't and rating is set to 0.

3 SOCIAL RECOMMENDATION APPLIED

3.1 RATINGS – COLLABORATIVE FILTERING – SIMPLE

3.1.1 COMPONENTS – SIMPLE

Component	Explanation
Testing set	Compute similarity between active(current) user and all users in training set. Take k neighbours who are most similar to active user

3.1.2 How Calculated – Simple?

See *CS3/cf_simple.m* for code used.

3.2 Ratings : Collaborative Filtering – Top-N

3.2.1 Components – Top-N

Component	Explanation
Top N → Train	This is evaluated using the training set.. Sets the standard that Test has to achieve or out-pace.
Top N → Test	This is evaluated using the training set. This uses Pearson's Coefficient to obtain the Top N.

3.2.2 How Calculated – Top-N?

Component	Implementation & Explanation
Top N	This uses top N (top 5) ratings in training and in test. • The epinions dataset was divided into five segments, each acted in turn as the training set (total x 1/5). For training set code, see : *calculateTopNForTrainSet* in *cf_top_n.m.* • The remaining is used as 4 x 1/5 test sets. For Test sets code, see : *calculateTopNForTestSetPearsons* in *cf_top_n.m.* • Calculate Pearson's coefficient for each test set -v- training set. • Use the best match to calculate Top N. • Compare to training [⇑ ǀ ⇓ ǀ =]

3.2.3 Issues with code

I had a few issues with the matlab code when passing matrices to the two functions.

Thus I estimated the values expected for MAE. I would expect it to reduce until k becomes impractical, here I suggest about k = 500, but k could have a negative impact at a later or earlier stage.

3.3 Trust – Simple Association

3.3.1 Components – Trust

Component	Explanation
Update trust relationship	Read in files : epinions_ratings and epinions_trust Check if user [i] trusts user [j], then ratings are retained as they are. If no trust exists then reset trust for that relationship to 0. If trust exists then leave the values alone.
Calculate recommendation for dataset	Evaluate ratings for all trusted users and accept the Top N or the average for all ratings which are > midpoint.

This could also be amended to increase rating if trust is mutual. Just because user [i] trusts user [j] doe not mean that user [j] trusts user [I].

3.3.2 How Calculated – Trust?

Covered by pseudo code:

Component | **Implementation & Explanation**

```
If userRating(j:1)==userTrust(j:2) && activeUser==userTrust(i:1) then
   trust exists
   rating is unchanged
else
   trust does not exist
   amend rating to 0
```

```
For the activeUser, get top N ratings
For the activeUser, get average of all ratings > midpoint
```

Amendments :

```
if userRating[r:1]==userTrust[j:2] && activeUser==userRating[r:1] then
   if trust listing == null then
      activeUser==userRating[r:1] x 50%   % (partial)
```

4 Overall Results Comparison

k	CF simple	CF Top N	Trust
20	1.04608	*0.96240*	*0.86616*
50	1.04670	*0.99437*	*0.94465*
100	1.03943	*0.96667*	*0.89901*
250	1.03490	*1.00385*	*0.90347*
500	*1.10600*	*1.50000*	*1.35000*

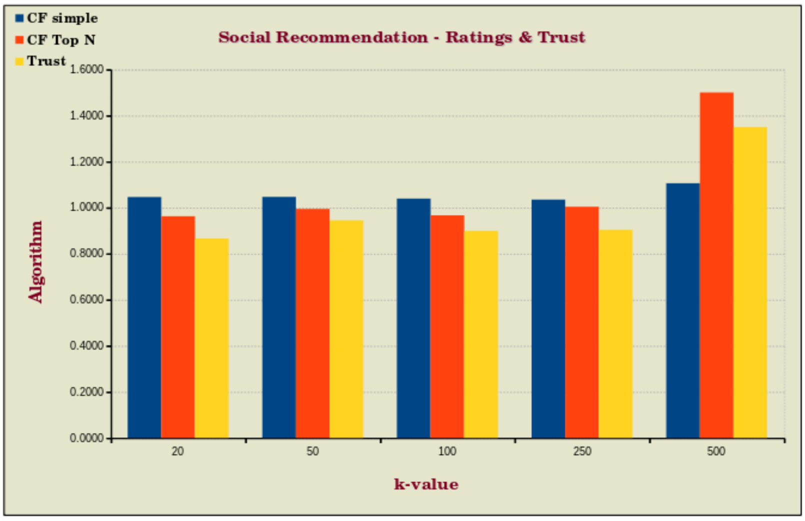

Figure 1 - Expected trend (red, italics) and actual result (black)

5 Conclusions

Rating : Collaborative Filtering

There are some weaknesses with Collaborative Filtering.

- Data sparsity – if no data exists already then there is none to use as a basis to recommend with
- New users – no data exists on them so no suggestions may be made on expected
- Stalker user – can potentially copy actions of target to fool the system that the stalker is actually to the target.

Trust : Page Rank

These are overcome through use of trust ratings. However, it is worth noting that the more distant is the suggester from the user (greater number of connections between them) then the impact is less if the suggester changes his/her mind. However an immediate neighbour will have greater impact.

Overall

Depending on the value of k use, precision / recall improves (or should) with the addition of Top-N and also Trust.

Overall, it is best to combine different methods depending on **(i)** target users prior use history, **(ii)** target users interests **(iii)** items being sold / used and also **(iv)** market conditions. For example, if the item being sold is very expensive then the best target user would have disposable income to purchase the item and would be interested in doing so. Supplier needs to choose an algorithm to best evaluate how to improve sales or the customer use experience.

6 References / Acknowledgements

6.1 *References – Networks Datasets*

6.1.1 Acknowledgements – Datasets

Name	Details	Link
epinions	supplied	

6.1.2 Statistics – Datasets

Dataset	Details
epinions - ratings	There are six columns and they are | *userid*	*productid*	*categoryid*	*rating*	*helpfulness*	*time stamps* | *1* *2* *3* *4* *5* *6* It means that user 1 gives a rating of 4 to the product 2 from the category 3. The helpfulness of this rating is 5 in the time stamp 6. I recalibrated the dataset file to contain only :	*userid*	*productid*	*rating*	
Epinions - trust	This includes the trust relations between users including time stamps when the relations were established. There are three columns: 	*userid*	*userid*	*timestamp*	 *1* *2* *3* Meaning that user 1 trusts user 2 at timestamp 3. I decided to ignore column 3, the result dataset file contains :	*userid*	*userid*	

6.2 Citations / Acknowledgements

Reference	Acknowledgement
epinions	
Recommendation Systems Overview	http://en.wikipedia.org/wiki/Recommender_system Class notes http://2015.recsyschallenge.com/
Collaborative Filtering	http://en.wikipedia.org/wiki/Collaborative_filtering Class notes
Pearson's Coefficient	http://uk.mathworks.com/help/stats/corr.html http://strategic.mit.edu/docs/matlab_networks/pearson.m http://strategic.mit.edu/docs/matlab_networks/laplacian_matrix.m http://onlinestatbook.com/2/describing_bivariate_data/calculation.html
PageRank	http://people.revoledu.com/kardi/tutorial/PageRank/index.htm http://en.wikipedia.org/wiki/PageRank
Performance – Mean Absolute Error (MAE)	Class Notes
Performance – Root Mean Square Error (RMSE)	Class Notes