Local Learning of Item Dissimilarity Using Content and Link Structure

Abir De, Maunendra Sankar Desarkar, Niloy Ganguly, Pabitra Mitra
Department of CSE, IIT Kharagpur, India
abir.iitkgp@gmail.com, {maunendra,niloy,pabitra}@cse.iitkgp.ernet.in

ABSTRACT

In the *Recommendation Problem*, it is often important to find a set of items *similar* to a particular item or a group of items. This problem of finding similar items for the recommendation task may also be viewed as a link prediction problem in a network, where the items can be treated as the nodes. The strength of the edge connecting two items represents the similarity between the items. In this context, a central challenge is to suitably define an appropriate *dissimilarity function* between the items. For content based recommender systems, the dissimilarity function should take into account the individual attributes of the items. The same attribute may have different importances in different parts of the underlying network. We focus on the problem of *learning a suitable dissimilarity function between items* and address it by formulating it as a constrained optimization problem which captures the local weightages of the attributes in different regions of the graph. The constraints are imposed in such a way that the non-connected nodes show higher value of dissimilarity than the connected nodes. The local tuning of the weights learns the optimal value of weights in various parts of the network: from the portions having rich graph information to the portions having only content information. Detailed experimentation shows the superiority of the proposed algorithm over the Adamic Adar metric as well as logistic regression methodology.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Information Filtering; H.4 [**Information Systems Applications**]: Miscellaneous

General Terms

Algorithms, Experimentation, Performance

Keywords

Content based recommendation, collaborative filtering, information retrieval.

RecSys'12, September 9–13, 2012, Dublin, Ireland.

1. INTRODUCTION

In the recommendation task, the goal is to produce a list of items that a target user might be interested in. A common approach for solving the task is to obtain a set of items that are similar to the items that the user has liked in the past. In this approach, often the items are viewed as the nodes of an underlying network. The strength of the edge connecting two nodes in the network indicates the similarity between the corresponding items. From a social networking perspective, the problem of finding (and recommending) similar items from this network may also be viewed as a link prediction problem.

Many researchers have analyzed the problem of recommending similar items in a pure link-based approach [4]. In [4], the authors suggest different metrics (e.g. Adamic Adar, Jaccard coefficients etc.) to estimate the strength of a candidate edge in the network.

An alternate approach is to make a content based recommendation where each item is associated with a feature vector or attribute profile. The features may hold numeric or nominal values and represent certain aspects of the item (e.g. director, genre, release date for a movie; author, genre, language for a book etc.) In this approach, an important task is to suitably construct a dissimilarity function between the feature vectors for computing the closeness between two items. In a dissimilarity function, the different item attributes are assigned different weights depending on their importances. Many researchers have devised methodologies to derive these weights. [3] uses a poisson regression model to find suitable attribute weights using clickstream data. Some researchers have exploited the knowledge of link information to learn the weights. For example, a hybridization of collaborative filtering and content based recommendation has been presented on a linear regression model based framework in [2]. The method proposed in [1] learns to bias a PageRank-like random walk using supervised approach, so that the walk visits the connected nodes more frequently than other nodes.

In these schemes, the trend is to associate an optimum global weight to each attribute. But importance (or weight) of one attribute may vary widely over items. For example, the role of schooling plays an insignificant role in two celebrities getting connected. However, in case of two common persons, their schooling may possibly play a vital role in recommending each other as a friend. We therefore emphasize that the weights assigned to an attribute over a network should not be constant, instead its importance should be determined taking locality into consideration.

In this technical note, we have addressed the content based recommendation problem from a novel optimization based framework. The framework identifies important attributes and assigns higher weights to those attributes while computing the similarity between the nodes (item). Also, the importance is specific to a region in a network. Finally, depending on the local weights, we develop the sorted list of dissimilarity values of items with a particular item. This is essentially a ranked list of recommendation for that item, or for an user who have liked that item.

2. PROPOSED APPROACH: CONSTRAINED LOCAL LEARNING

Problem Definition: Let x be a particular item. The objective is to make a ranked list of recommendations to x. So, for a given x, the algorithm should generate $A_x(y, S)$, where y's are the recommended items and S's are the corresponding scores. More is the score, higher is the rank of the corresponding item in the list. To find A_x, for a particular item x, we consider all the nodes which have common neighbours with x, as candidates. So, we construct a hypothetical item-item network where two items are connected by an edge if they have been previously accessed/accepted by a certain number of users. The task is to predict whether a presently non existing link may appear in the future. Hence the problem that whether an item y belongs to the list A_x, reduces to the problem of predicting a future edge between nodes x and y in the underlying graph.

In order to understand/predict the dynamics of link formation, we assume that *the dynamics primarily depends on the corresponding locality of the graph.* In other words, the possibility of a node (x) being connected to another node (y) rarely is determined by the nodes that are *far apart* from x and y in the underlying graph. Keeping this in mind, the first step of the algorithm learns the amount of dissimilarity between x and y and their common neighbours. (We term this process as the reference dissimilarity function.) We then use this learning to predict the chance of a link arriving between x and y.

Definitions: We formally define some important terms in Table 1.

Table 1: Important Definitions:

Neighbourhood of a node i	$\Gamma(i)$
Attribute vector of node i	θ_i
Dissimilarity function between nodes i and j	$\Delta_w(i,j) = w^T\lvert(\theta_i - \theta_j)\rvert$ [1]

1. Local Weights and Reference Dissimilarity Function: Computation of an appropriate dissimilarity function relies on finding suitable weights associated with each attribute. Weight of one individual attribute may vary widely over the network. Thus the choice of suitable weights specific to a particular region of the graph is crucial. In order to predict an edge between x and y (assuming they are not connected yet), we consider the locality $N = \Gamma(x) \cup \Gamma(y)$ and restrict our discussion to N throughout this section. If there is an edge between two items, we assume that the edge has

[1] $|.|$ denotes the term by term absolute value of a vector (e.g $|[-2,3]| = [2,3]$) and w is the weight vector.

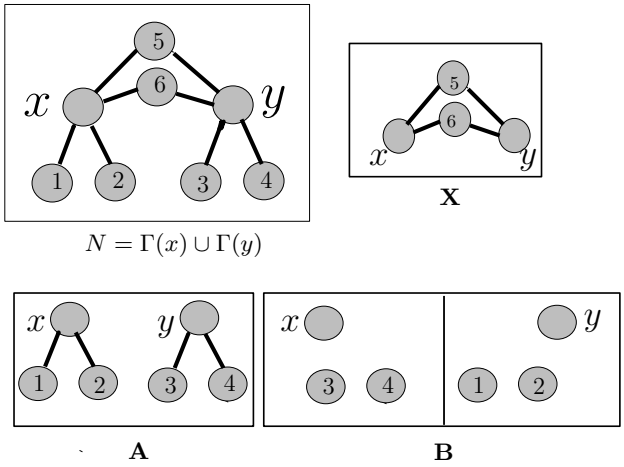

Figure 1: Sample graph to determine the possible score between x and y. X : Edges of x and y with their common neighbours give an estimate of minimum value of the reference dissimilarity function Δ_w^{xy}, *under the imposition of* A: ($\Delta_w(1,x)$ and $\Delta_w(2,x)$) are not very high (Similarly for the pair $(3,y)$ and $(4,y)$). B: For non existing edges, the dissimilarity can be very high w.r.t the reference. So, $\Delta_w(1,y)$, $\Delta_w(2,y)$, $\Delta_w(3,x)$ and $\Delta_w(4,x)$ are relatively high with respect to Δ_w^{xy}.

arrived because the attributes of the two nodes are similar. Hence we assume that the value of the dissimilarity function is low for node pairs which have an edge connecting them and relatively higher for node pairs which are not neighbours. Under these impositions, we wish to minimize the sum of the dissimilarity values of x and y with their common neighbours. This is because this minimization would in turn give the maximum possibility of an edge between x and y appearing in future. It can also be termed as *reference dissimilarity function.* So our objective is to find optimal w, so that, the dissimilarity

$$\Delta_w^{xy} = \sum_{i \in \Gamma(x) \cap \Gamma(y)} \Delta_w(i,x) + \Delta_w(i,y)$$

should be minimum w.r.t w, *assuming that,* the dissimilarity between linking item pairs are low ,*i.e.* $\Delta_w(i,x)$ for $i \in \Gamma(x) \setminus \Gamma(y)$ and $\Delta_w(i,y)$ for $i \in \Gamma(y) \setminus \Gamma(x)$ should be low w.r.t the reference dissimilarity Δ_w^{xy} and for not existing edges, dissimilarity (e.g. $\Delta_w(i,y)$ for $i \in \Gamma(x) \setminus \Gamma(y)$) should be relatively high.

Following the above arguments, the problem of determining the reference dissimilarity function can be cast as the following optimization problem.

$$\Delta^{xy} := \min_w \widetilde{\Delta}_w^{xy}$$

subject to, Ω_w^{xy} :

$$\sum_{i \in \Gamma(x) \setminus \Gamma(y)} \widetilde{\Delta}_w(i,x) \leq \alpha \widetilde{\Delta}_w^{xy} \quad (1)$$
$$\sum_{i \in \Gamma(y) \setminus \Gamma(x)} \widetilde{\Delta}_w(i,y) \leq \alpha \widetilde{\Delta}_w^{xy} \quad (2)$$
$$\sum_{i \in \Gamma(y) \setminus \Gamma(x)} \widetilde{\Delta}_w(i,x) \geq \beta \widetilde{\Delta}_w^{xy} \quad (3)$$
$$\sum_{i \in \Gamma(x) \setminus \Gamma(y)} \widetilde{\Delta}_w(i,y) \geq \beta \widetilde{\Delta}_w^{xy} \quad (4)$$

where $\sum_{i \in G} \widetilde{\Delta}_w(i,x) = \frac{1}{|G|} \sum_{i \in G} \Delta_w(i,x)$,

and α and β are suitable parameters. Symbolically Ω_w^{xy} denotes the set of all constraints.We use standard LP method to solve the optimization problem. The algorithm gives the outputs: *reference dissimilarity value* (Δ^{xy}) and *optimal weights* (w^*).

2. Computation of Actual Dissimilarity Function: Using the optimal weight w^* so derived, we calculate the dissimilarity between x and y,

$$\delta^{xy} = w^{*T}|\theta_x - \theta_y|.$$

3. Computation of Score: The goodness of δ^{xy} needs to be compared w.r.t its locality which is quantified by the value Δ^{xy}. The difference between δ^{xy} and Δ^{xy} would then actually indicate how more similar is x and y than their surroundings and in turn tell us the possibility of formation of new edge. So the score is given as,

$$score(x,y) = (\Delta^{xy} - \delta^{xy}).$$

Choice of α and β: α and β are the parameters that control the inequality constraints. Smaller (larger) value of α (β) allows a lower (higher) dissimilarity between the connected (disconnected) nodes. Here, we have experimentally selected α and β. We have experimented with random items and the optimum values of α and β is found by maximizing the harmonic mean of *true negative and false positive*.

3. EXPERIMENTAL RESULTS

Experimental Setup : We consider part of Movielens, Citeseer, Cora and WebKb datasets for experimentation. As baselines, we have chosen Adamic Adar distance which is a pure link based metric and a logistic regression based recommendation model which is an unconstrained classification model [5].

Datasets used: **Movielens [7]:** It has 6040 users and 3952 movies. Each user has rated at least one movie. Each movie has features which are a subset of a set of 18 nominal attributes (e.g. animation, drama etc.). We have constructed a hypothetical network where two movies have an edge if they have at least a certain number of common viewers. By choosing the minimum number of common viewers to be 100, we obtain a network with 3952 nodes and 5669 edges.

CiteSeer [6]: The CiteSeer dataset consists of 3312 scientific publications and the citation network consists of 4732 links. Each publication is tagged with a set of keywords. Total number of keywords is 3703.

Cora [6]: The Cora dataset consists of 2708 scientific publications and the citation network consists of 5429 links. Here the total number of keywords is 1433.

WebKb [6]: This dataset consists of 877 scientific publications and the citation network consists of 1608 links. Here the total number of keywords is 1703.

For these four datasets, we generated recommendations with three algorithms, (a) Constrained Local learning (CLL) algorithm proposed in this paper, (b) Adamic Adar metric, (c) Logistic regression. We considered same set of items as queries for all the three algorithms. To build the ground truth, we have removed some edges from the graphs and have noted whether those edges can be predicted back.

Adamic Adar (AA) : Using the method described in [4], we obtain the recommendation scores. **Logistic Regression (LR):** In this method the difference in attributes between x and y is defined by $\theta_{xy} = (0.5\hat{\mathbf{1}} - |\theta_x - \theta_y|)$. Here

Table 2: Summary of the datasets , where N is the total number of items, E is the total number of links, $n(a)$ is the number of features, d_{max} is the maximum degree and d_{avg} is the average degree.

Dataset	N	E	$n(a)$	d_{avg}
Movielens	3952	5669	18	2.8689
CiteSeer	3312	4732	3703	2.7391
Cora	2708	5429	1433	3.89
WebKb	877	1608	1703	2.45

$\hat{\mathbf{1}}$ is a vector having all elements equal to 1 and dimension same as θ. Since $|\theta_x - \theta_y|$ is a binary vector in all datasets, elements of the term $(0.5\hat{\mathbf{1}} - |\theta_x - \theta_y|))$ will be positive or negative depending on whether the features are similar or not. The score between x and y is given by $score(x,y) = 1/(1 + exp(-w^T\theta_{xy}))$. So more is the $score(x,y)$, higher is the rank of y in the list of recommended items to x. The logistic regression algorithm is carried out in two different ways. In the first approach, we randomly choose a part of network as the training data. Using these data, the regression model generates the optimum attribute weight vector w. In the second method, to provide recommendation to an item x, the training data consists of x and it's relation (link or no link) with randomly selected 30 percent nodes. So for each node, the optimum w is different, i.e. w is local to x in this case. Interestingly we observe that, for all the four datasets, the performance of global logistic regression model is very poor. Hence we present the results given by second(local) method of logistic regression for comparison with our method.

For performance comparison we use the following performance metrics.

Metrics 1 and 2: $MeanPrecision(k) = \frac{1}{q}\Sigma_{i=1}^{q} P_i(k)$; $MeanRecall(k) = \frac{1}{q}\Sigma_{i=1}^{q} R_i(k)$, where q is the number of queries, $P_i(k)$ is Precision@k for i^{th} query and $R_i(k)$ is Recall@k for i^{th} query. So $MeanPrecision(k)$ is the average of all Precision@k values over the set of queries. Same is with $MeanRecall(k)$.

Metric 3: $AvP(i) = \frac{1}{L}\Sigma_{k=1}^{n} P_i(k).r_i(k)$, where n is the total number of items, L is the number of retrieved relevant items and $r_i(k)$ is an indicator taking value 1 if the item at rank k is a relevant item or zero otherwise. Thus we obtain $MAP = \frac{1}{q}\Sigma_{i=1}^{q} AvP(i)$.

Comparison of Results : Figure 2 indicates variations of $MeanPrecision$ against $MeanRecall$ and Table 3 gives a comparative analysis of MAP (Mean Average Precision) values for all datasets. We observe that in all the datasets the overall performance of the proposed approach is much superior to the other two methods. In all these datasets, where the nodes with high degree are connected together, Adamic-Adar metric fails to provide high score. Due to its very poor performance in case of popular items, it produces a poor overall MAP value in all four datasets.

Apart from locality, consideration of constraints is found to be very crucial. Because in the local logistic regression model, although the weights are obtained locally, its performance is substantially poorer than CLL. It is because, even though it is based on local behaviour, no constraint is

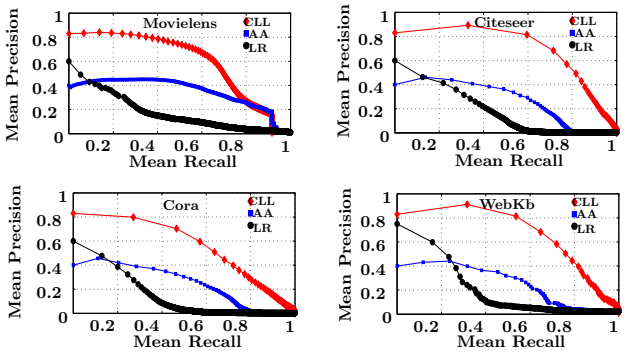

Figure 2: Mean Precision Recall curve

Table 3: Mean Average Precision(MAP), for different algorithms and different datasets

Dataset	CLL	AA	LR
Movielens	0.7740	0.4720	0.2313
CiteSeer	0.6313	0.4161	0.3783
Cora	0.5620	0.3910	0.2923
WebKb	0.6427	0.4029	0.3339

considered. So a suitable formulation of constraint plays a deciding role.

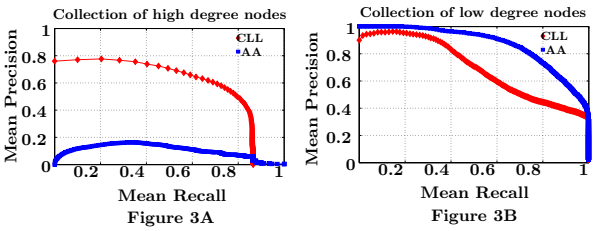

Figure 3: A comparison between CLL and AA algorithm on Movielens dataset

CLL vs Adamic Adar : Figure 3 shows a comparison of Adamic Adar and CLL on Movielens dataset over two parts of the network: on a dense part and on a sparse part. It is clear that in the dense part, high degree nodes being connected together, Adamic Adar shows extremely poor performance [Figure 3A] and CLL performs much better. Interestingly in case of low degree items the results are comparable although Adamic Adar performs a bit better [Figure 3B]. Please note that CLL operates efficiently at a very important zone i.e. the zone where the popular items are being bought along with some other items. In most of the recommendation algorithms this is considered as a superfluous information and given least importance. However there are some definite semantics behind choice of such popular items, which this method (CLL) clearly brings forward. On the other hand, CLL can quickly adjust to regions where link plays a predominant role and performs as good as a pure link based strategy(AA).

Variations on datasets: If we carefully check the MAP values in the Table 3, we observe that MAP value for CLL is much higher in Movielens than the other three datasets. This is because the attributes in Movielens are much more well structured. It actually consists of the genre and it is observed that people usually like movies of similar genre. On the other hand in the citation document space the keywords get diluted through polysemy, synonymy etc. However more important to note that CLL can really take advantage of such attribute structure and increase its MAP much sharply (20%) than Adamic Adar (13%) from the corresponding second best in the list.

4. CONCLUSION AND DISCUSSIONS

A method for local learning of item dissimilarity via a constrained optimization framework has been proposed in this paper. The algorithm assigns more weights to the important features of a particular region of the graph. The important features and the reference score of similarity are estimated in this optimization portfolio. The overall performance of the algorithm is found to be significantly better than two baseline algorithms, namely Adamic Adar and Logistic Regression. An interesting property of our algorithm is that it can also adjust between regions where content is dominating and where link is more important. However, these are initial results, a more detailed theoretical as well as experimental work need to be launched to realize the full potential of the algorithm.

5. REFERENCES

[1] L. Backstrom and J. Leskovec. Supervised random walks: predicting and recommending links in social networks. In *Proceedings of the fourth ACM international conference on Web search and data mining*, WSDM '11, pages 635–644, New York, NY, USA, 2011. ACM.

[2] S. Debnath, N. Ganguly, and P. Mitra. Feature weighting in content based recommendation system using social network analysis. In *Proceedings of the 17th international conference on World Wide Web*, WWW '08, pages 1041–1042, New York, NY, USA, 2008. ACM.

[3] M. Kagie, M. van Wezel, and P. J. Groenen. Choosing attribute weights for item dissimilarity using clikstream data with an application to a product catalog map. In *Proceedings of the 2008 ACM conference on Recommender systems*, RecSys '08, pages 195–202, New York, NY, USA, 2008. ACM.

[4] D. Liben-Nowell and J. Kleinberg. The link prediction problem for social networks. In *Proceedings of the twelfth international conference on Information and knowledge management*, CIKM '03, pages 556–559, New York, NY, USA, 2003. ACM.

[5] A. Popescul, R. Popescul, and L. H. Ungar. Towards structural logistic regression: Combining relational and statistical learning. In *Proceedings of the Workshop on MultiRelational Data Mining*, KDD'02, pages 130–141, 2002.

[6] www.cs.umd.edu/projects/linqs/projects/lbc.

[7] www.grouplens.org.