NetRec: Product Recommendation With Heterogeneous Networks

McQuillan, Daniel
dmcquil2@illinois.edu

McConnell, John
mcconne7@illinois.edu

May 10, 2016

Abstract

Picking a good recommendation strategy is extremely important because a bad strategy can cause users to lose faith in application. Accordingly, the bad recommendations make it impossible to deliver a nice personal experience to the end user. Collaborative recommender systems can be used to present similarities between users as well as products in order to create a generalized relationship between entities based on basket analysis. There currently exists a multitude of recommendation strategies utilizing networks. A large focus is placed on link prediction. In this paper we examine, our own algorithm to predict consumer purchases and rank products.

1 Introduction

1.1 Network Overview

Definition 1.1. Network Schema A network schema displays the types of relationships that will be exhibited within the graph.

Definition 1.2. Network instance A network instance is a network that is created that follows a given network schema.

Definition 1.3. Meta Path A meta path $P$ represents a sequence of relations $R_1, \ldots, R_t$ with constraints such that $\forall i \in 1, \ldots, t - 1, \text{range}(R_i) = \text{domain}(R_{i+1})$. The meta path $P$ can also be written as $P : T_1 \stackrel{R_1}{\rightarrow} T_2 \stackrel{R_2}{\rightarrow} \cdots \stackrel{R_t}{\rightarrow} T_{t+1}$, i.e. $P$ corresponds to a composite relation $R_1 \times R_2 \times \cdots \times R_t$ between node type $T_1$ and $T_{t+1}$.

domain($P$) = domain($R_1$) and range($P$) = range($R_t$). The length of $P$ is $\ell$, i.e., the number of relations in $P$.

Definition 1.4. Homogeneous Networks (Information Network). An information network is defined as a directed graph $G = (V, E)$ such that $V$ represents the set of vertices within the graph and $E$ is the set of edges in the given graph such that $E \in V \times V$. A entity type mapping function is not needed for this type of information network since the there will only be one type of node and one type of edge.

Homogeneous networks are usually created from a larger heterogeneous network in order to simplify the network model. This usually can be done by navigating a path in order to represent connectivity of two homogeneous vertices within a network.

Definition 1.5. Heterogeneous Networks (Information Network). An information network is defined as a directed graph $G = (V, E)$ with an entity type mapping function $\phi : V \rightarrow A$ and a link type mapping function $\phi : E \rightarrow R$. Each entity $v \in V$ belongs to an entity type $\phi(v) \in A$, and each link $l \in E$ belongs to a relation type $\phi(l) \in R$.

Homogeneous networks typically don’t retain enough information about the relationships that are lost when created from an existing homogeneous network. Many real world systems have a complicated network schema with multiple distinct types of inter-related objects and relationships. Structures of this type are broadly defined as heterogeneous networks, a term which encapsulates multiple more specific types of networks.

1. Multi-mode [10] is a network that typically consists of multiple heterogeneous social actors among which various types of interactions could occur

2. Multi-dimensional/relational [3] are networks with multiple types of relationships between types of nodes

3. Bi-partite [5] is a network that can be divided into 2 disjoint sets. In the case of a heterogeneous network the two sets usually consist of two different types of nodes such as items and users

4. K-partite [5] is a natural extension to that of bi-partite networks such that instead of two sets it has $K$ disjoint sets

In order to induce recommendations from a graph a homogeneous network is induced from that of the heterogeneous structure. Instead we would like to maintain this full featured structure in order to minimize information loss.
1.2 Recommendation Systems

1.2.1 Why do we need good recommendations

Picking a good recommendation strategy is extremely important since when using a bad strategy users can lose faith in your recommendations altogether and the ability to deliver a personal experience with it. [1]

Scenario

1. A large supermarket has just recently opened a website in order to enable them to sell their groceries online. When a user shops at the store the cashiers know to promote items that commonly go along with the items that they are purchasing. Similarly, using basket analysis the stores are able to organize the store in a way that the items that are purchased more frequently together are also physically located near each other. The designers of this online store have identified this challenge and would like to bring the same upselling capabilities to their online store. Currently they possess a few years worth of shopping data from in-store transactions. How would the store managers be able to accomplish this without any physical interaction with the store?

Recommendation systems can help to improve the results present on a website in order to personalize the page to the current user. Using the results of a recommendation algorithm a system could suggest results either based on a category a user is viewing, an item in order to suggest possible pairings, as well as full on prediction of their order when it comes to basket based orders such as online grocery shopping.

1.3 Related Work

In our research we discovered multiple existing efforts for developing a quality recommendation. One of which, is called HeteRec [2]. HeteRec is a meta-path based learned model that is used to create recommendation results based on the guidance of a meta-path in order to maintain a relationship in the recommendation. This could be as simple as finding users who are similar based on the fact that they tend to buy items that are on sale. A personalized and global model is also proposed utilizing a latent variable to represent the learned model as well as bayesian ranking optimization to estimate the given model. Personalized entity recommendation through the use of user feedback in heterogeneous information networks were studied and the displayed algorithm utilized meta-paths to generate latent features for users and items. Other efforts surrounding recommendations have also been pursued in the form of link prediction.

SimRank is another algorithm which is related to our work [4]. SimRank calculates the pairwise similarity between two nodes in a network based of the nodes topology. SimRank is defined recursively which makes the complete calculation often very expensive. The high cost of the algorithm and its inability to deal with large complex heterogenous networks makes an algorithm like PathSim a viable candidate [8]. This algorithm resulted in accurate recommendations but has room for improvement as a result of other similar network based research. The HeteRec algorithm creates both a global and personalized recommendation model to create recommendations. In their experiments, they utilized the K-means clustering algorithm to determine similar user neighborhoods in their network. We believe a possible place of improvement could be to utilize NetClus [9]. The procedure of which will be described in a later section.

2 Algorithm

The algorithm we will be using to perform recommendations will be very similar to that of the HeteRec algorithm. Most elements of the HeteRec algorithm will remain the same with some simple modifications. The algorithm will utilize the diffusion score and global recommendation models used in HeteRec. The main change is that we will require as input to the algorithm a star-schema based network (defined in definition 2.1) to be used as input to the NetClus [9] algorithm in order to provide a soft clustering of the target user nodes. If the specified network does not conform to a star schema then the algorithm will produce a failure. In order for this algorithm to run it must receive a network in a star schema in order to calculate the personalized clustering model.

Definition 2.1. Star Network Schema An information network $G = (V, E, W)$ on $T + 1$ types of objects $\chi = \{X_t\}_{t=0}^T$ is called a star network schema, if $\forall e = (x_i, x_j) \in E, x_i \in X_0 x_j \in X_t (t \neq 0)$, or vise versa, $G$ is then called a star network. Type $X_0$ is called the target type. [9]

Example 2.1. Star Network Schema Example Below is an example of a star network schema where there is 6 types. 5 of the types will simply be labeled $X_{1...5}$ and the target type will be labeled $X_T$.
2.1 NetClus

An intermediate step of our algorithm is performing NetClus, in order to find clusters and relative rankings for products. NetClus requires that the network follows a star schema as defined above [2.1]. For our experiment that meant creating a graph with all nodes connecting to the central user node. The NetClus algorithm provides ranking and cluster though a generative probabilistic model and applying the EM algorithm. The algorithm works as follows [3]:

1. Partition the target nodes into different clusters.
2. Build the ranking based generative model
   \[ \{ \mathcal{P}(x|C_k) \}^K \] (1)
3. Calculate the posterior probabilities for the target objects
   \[ \{ p(C_k|x) \} \] (2)
4. Calculate mean attributes of each cluster
   \[ \{1/\|X_k\| \sum_{x \in X} \{ p(C_k|x) \}^K \} \] (3)
5. Adjust clusters from the updated attributes
6. Repeat until convergence
   \[ \{ C_k^+ \}^K \} = \{ C_k^- \}^K \} = \{ C_k^{t-1} \}^K \} \] (4)

NetClus is used in our algorithm to cluster related products together and calculate the relative importance of the products. The ranking in the algorithm is based off of preferential attachment. This is an important distinction from degree. Measuring the node based off of rank tries to model the importance of the node in the network. Using the node’s degree as the metric would find local importance but will not correlate well with global importance in the graph structure. The phenomena occurs naturally in the real world. A good example is the DBLP dataset.

The top authors publish the majority of papers, which only appear in the top conferences. The ranking model of NetClus models the real world behavior well for these types of networks.

2.2 Generative Model

For our model, we use a star schema which derives a generative probabilistic model that highly ranked objects which co-appear are more likely to generate a center object.

- Our shopping networks, \( G \), consists of \(<V,E,W>\).
- Our nodes, \( V \), consist of four distinct types: \( P, B, C, U \), meaning products, brands, categories, and users respectively. \( V \) contains all nodes of all other types.
- \( E \) is the edges already represented in the dataset.
- \( W \) is a weighted adjacency matrix as defined as follows:
  \[ W_{x_i,x_j} = \begin{cases} 1 & \text{if } x_i(x_j) \in P \cup B \cup C, \text{and } x_i \text{ has link to } x_j \\ 0 & \text{otherwise} \end{cases} \] (5)

In order to handle the many different types that the network nodes can have the algorithm needs to make a couple of key assumptions:

- First, assume the probability of visiting objects of different types is independent.
- Even with the independence assumption, it does not negate the factorization that can be done on the probability of visiting an object, \( p(x_i|G) \).
- Example, \( p(x_i|G) \) can be factored into:
  \[ p(x_i|T_{xi}, G) \times p(T_{xi}|G) \] (6)

\( p(T_{xi}|G) \) is the probability of visiting the type of node which belongs to node \( xi \) given \( G \).

- One more key assumption is that visiting two nodes jointly, \( p(x_i,x_j|G) \) which have the same type are independent.

- Giving us the equation:
  \[ p(x_i,x_j|T_{x}, G) = p(x_i|T_{x}, G) \times p(x_j|T_{x}, G) \] (7)

With the above assumption and the equation \( p(x_i|G) \), we have a generative model which can be used in the EM algorithm.

Now we must create the generative model in the reverse direction: building a generative model for the target objects given the rankings of the attribute objects. With the
given assumptions we have made, the calculation is as follows:

\[ p(d_i|G) = \prod_{x \in N_G(d_i)} p(x|G)^{W_{d_i,x}} \]  

where \( N_G(d_i) \) is the neighborhood around \( d_i \). Under this model, a target object with many high ranking attribute objects is more likely to be generated. An important consequence of these equations is that the different types of attribute objects are factorized.

### 2.3 Posterior Probability

The problem is now reversed. We have the generative model, a target object with many high ranking attribute objects is more likely to be generated. An important consequence of these equations is that the different types of attribute objects are factorized.

NetClus uses the conditional rankings of the attribute types as well as a smoothing technique to produce the final equation:

\[ p(d|G_k) = \prod_{x \in N_G(d)} p(x|G_k)^{W_{d,x}} p(T_k|G_k)^{W_{d,x}} \]  

\[ p_{\text{smoothed}}(d|G_k) = \lambda p(x|G_k) + (1 - \lambda) p(x|G) \]

Smoothing is important because objects will be almost guaranteed to be misclassified. If any ranking score for the given cluster is zero the entire probability will result in zero and none of the information will be propagated through the network.

NetClus gets the potential size of each cluster, \( k \), by maximizing the log-likelihood of generating the target objects and taking local maxima for \( k \). This process is a natural product of following the EM algorithm. The centers of the clusters can be inferred using the mean vector of target objects in the clusters. Naturally, the question becomes: how to represent the target objects as a vector? NetClus represents the target objects as a vector of probabilities for each cluster:

\[ \vec{d} = < p(k_1|d), \ldots, p(k_n|d) > \]

Now using the cosine similarity between target objects and cluster vectors. New clusters can be assigned for each target object based on the minimum distance to the center of each cluster.

### 2.4 Network Diffusion

Network diffusion captures the underlying mechanism of how events propagate throughout a complex network.

Whether the subject is a virus spreading across a computer network or a viral marketing message among a network of people, the questions are the same: How fast will it spread? How will the system as a whole react? What are the hubs in the network that are more connected and therefore more important than others? In the context of this recommendation strategy the diffusion score will be calculated in terms of the PathSim similarity and an indicator matrix to reflect user interest. The diffusion score is defined in equation [13]

\[ R_{u_i,e_j}^{(l)} = s(u_i, e_j) p^{(l)} = \sum_{e \in I} \frac{2 \cdot R_{u_i,e \cdot} M_{e,e_j}^{(l)}}{M_{e,e}^{(l)} + M_{e,j}^{(l)}} \]  

where \( p^{(l)} \) is the \( l \)-th meta path, \( M_{ij}^{(l)} \) represents the transition matrix where \( i, j \in V \) and \( M_{ij}^{(l)} \) will represent the number of distinct paths from node \( i \) to node \( j \), and \( R_{u_i,e \cdot} \) is a relationship relationship matrix as follows in equation [14]

\[ R_{u_i,e \cdot} = \begin{cases} 1 & \text{if interaction is observed} \\ 0 & \text{otherwise} \end{cases} \]

### 2.5 Global Recommendation Model

The global recommendation model will be similar to that of the original HeteRec global model with the exception that the network diffusion matrix will not be factorized into a user and item preference matrix. The global recommendation will be defined as the following in equation [15]

\[ r(u_i,e_j) = \sum_{l=1}^{L} \theta_l \cdot R_{ij}^{(l)} \]  

As in the HeteRec model the parameter \( \theta_l \) will represent the weight for the \( l \)-th user and item low-rank representation pair.

### 2.6 Personalized Recommendation Model

The main difference between our method and the original HeteRec algorithm will exist within the personalized recommendation model. Instead of simply using the K-means algorithm to estimate user cluster similarity we will utilize a more graph oriented approach known as NetClus. K-means and other homogeneous network based clustering algorithms usually work by inducing a simpler homogeneous network from a larger, more complex heterogeneous network. However, real world networks are usually much too complex to be represented in this format due to the network’s sparsity as well as the number of possible types of nodes and links. Like other heterogeneous network based clustering methods this will help to minimize information loss in the network as well as create a way to
rank nodes within a given cluster. Other clustering methods also do a hard clustering upon the data entries so that there will be no ambiguity in which cluster a node should belong too. Since this method does more of a soft clustering it can create a better membership distribution across nodes in the network.

Our personalized recommendation model will be defined in terms of the global model and the cluster similarity score retrieved from the NetClus algorithm. The final equation will be similar to that of the original personalized recommendation model with the exception that the similarity score will be retrieved using NetClus and is defined in equation 18:

$$ r^s(u_i, e_j) = \sum_{k=1}^c \text{netsim}(C_k, u_i) \sum_{\ell=1}^c \theta_{\ell} \cdot f^{(\ell)}_{ij} $$

where netsim is defined as the NetClus based similarity of a user, $u_i$ to a cluster, $C_k$. Defined in terms of the NetClus algorithm the netsim will be defined as the probabilistic likelihood of a cluster given a user. More succinctly, netsim is defined in equation 17:

$$ \text{netsim}(C_k, u_i) = p(C_k | u_i) $$

### 2.7 Parameter Estimation

The parameter $\theta$ will be estimated for each meta-path and cluster pair using the methodology described in HeteRec. The objective function described in HeteRec will work for our variation as well since it simply tries to estimate weights such that the calculated score of an item that is purchased is greater than that of an item that was not purchased by a given user. The objective function is defined as follows.

$$ O = -\ln p(\theta | R) = -\ln p(R | \theta) p(\theta) $$

$$ = -\sum_{u_i \in U} \sum_{(e_a > e_b) \in R_i} \ln p(e_a > e_b, u_i(\theta)) + \lambda ||\theta||_2^2 $$

$$ = -\sum_{u_i \in U} \sum_{(e_a > e_b) \in R_i} \ln \sigma(r(u_i, e_a) - r(u_i, e_b)) + \lambda ||\theta||_2^2 $$

where $\sigma$ is the logistic sigmoid function $\sigma(x) = \frac{1}{1+e^{-x}}$, $p(e_a > e_b, u_i(\theta))$ is probability that an item a user purchased is more likely to be purchased than another defined as $p(e_a > e_b, u_i(\theta)) = \sigma(r(u_i, e_a) - r(u_i, e_b))$, $(e_a > e_b) \in R_i$ is the pairs of items in user i’s relation matrix such that item a has a greater value than item b in the relation matrix, and $\lambda ||\theta||_2^2$ is a data dependent $L_2$ regularization term.

The gradient of the above objective function is:

$$ \frac{\delta O}{\delta \theta} = -\sum_{u_i \in U} \sum_{(e_a > e_b) \in R_i} \frac{e^{-r_{i,ab}}}{1 + e^{-r_{i,ab}}} \delta r_{i,ab} + \lambda \theta $$

In order to estimate the parameter $\theta$ we used a standard SGD approach. Our update rule is defined as follows:

$$ \theta = \theta - \alpha \left[ \frac{1}{|U|} \frac{\delta O}{\delta \theta} \right] $$

Using SGD we only needed to sample a small subset of our data randomly sampled from $R$ in order to find a good estimation of $\theta$.

### 3 Experiment

Our experiment will be used to see how well NetRec models transactional data and also do a comparison between multiple variations of NetRec as well as HeteRec. We will first give an overview of our data schema in section 3.1, then go over our network creation process in section 3.2, then we will present the results of the experiment in section 3.3 and finally we will give an overview of the NetClus clustering results in section 3.4.

#### 3.1 Data Overview

The experiment utilized supermarket purchase data from 1998. In order to create our data we extracted the following fields from the database dump.

1. product_id - The unique identifier for a given product in the database
2. time_id - A year relative time identifier to identify when the product is purchased
3. customer_id - The unique id of the customer that purchased the item
4. brand_name - The brand that the product is associated with
5. product_name - The descriptive name for the given product
6. product_class_id - The roll-up id for a given tuple of a category and a sub category
7. product_subcategory - The leaf category that the product belongs to in the category tree
8. product_category - The roll-up category that the product is associated with

In order to group our purchases into transactions we grouped by customer_id and then by time_id in order to form a set of user transactions. Since the experiment we are performing relates to network evolution we want to identify if the items that are purchased from January to November can be used to create a model to recommend the product purchases in December.
3.2 Network Setup

In order to perform recommendations we need to set up two networks. The first will be a K-partite network and the second will be a star network. The network schemas will be called the relationship and the user cluster network respectively.

**Definition 3.1. Purchase Information Network** The relationship network will be the main backing data model used in conjunction with the HeteRec algorithm. The schema will consist of 4 node types: Users, Transactions, Items, and Brands. Users create Transactions, Transactions belong to Users, Transactions contain an Item, Items reside within a Transaction, Items resides in a Brand, and Brands contain items. This network can also be seen as a 3-Partite network.

![Diagram of relationship network]

**Definition 3.2. User Cluster Information Network** The purchase network will be the data model used to create the network structure used to find user clusters using the NetClus algorithm. The schema will consist of 4 node types: Users, Items, Brands, and Categories. Users will purchase from a Category, Categories will be purchased by Users, Users will purchase from a Brand, Brands will be purchased by Users, Users will purchase Items (or products), and Items will be purchased by Users.

![Diagram of user cluster network]

In order to perform HeteRec we also require meta paths in order to find optimal recommendations. The meta paths we will be using for the purpose of our experiment are listed in table [I].

### 3.3 Experiment Results

Upon creation of our network it resulted in a network that could be modeled via the power law. (Table 2)

<table>
<thead>
<tr>
<th>Nodes</th>
<th>Edges</th>
</tr>
</thead>
<tbody>
<tr>
<td>2787</td>
<td>27855</td>
</tr>
</tbody>
</table>

Table 2: Node/Edge distribution

By using this network we ran two variations of NetRec’s and HeteRec’s personalized model to determine if the NetRec algorithm produced higher quality results. The experiment resulted in the NetRec results being higher than both HeteRec as well as the inclusion of NMF into NetRec. We used a binary classification measure in order to indicate correctness of our algorithm. We simply calculated the average score for each user and then if the score of a purchased item was greater than the average user score it was viewed as correct (as shown in equation group [21]). The accuracy results are shown in table [3].

\[
C_{ui} = \sum_{e \in T_{ui}} I \left[ r(u_i, e) > \frac{1}{|T_{ui}|} \sum_{e' \in T_{ui}} r(u_i, e') \right] \\
C = \sum_{u_i \in U} F = \left[ \sum_{u_i \in U} \sum_{e \in T_{ui}} 1 \right] - C \\
\text{Accuracy} = \frac{C}{C + F} \tag{21}
\]

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>NetRec</td>
<td>99.23</td>
<td>77.5</td>
</tr>
<tr>
<td>NetRec-NMF</td>
<td>88.55</td>
<td>77.4</td>
</tr>
<tr>
<td>HeteRec</td>
<td>97.40</td>
<td>75.5</td>
</tr>
</tbody>
</table>

Table 3: Experiment Precision/Recall

### 3.4 NetClus

Running NetClus with a value of K=4 yielded the following results. First is the product rankings, for clusters 1, 2, 3, and 4 (Table 4 & 5). Looking at the results of NetClus, we can hypothesize that the clusters 1,2,3,4 have the rough semantic meanings of: vegetables, household goods, soup, and simple foods. Of course, more rigorous testing is needed to prove the algorithm is making semantically meaningful clusters.

We also ran performance tests to see how the NetClus portion of our algorithm scales.
### Meta Path | Semantic Meaning
--- | ---
U - T - I - B - I - T - U - T - I | User to item similarity based on brands purchased and brands similar users have purchased
U - T - I - T - U - T - I | User to item similarity based on similar users
U - T - I | User to item similarity based on items the user has purchased

Table 1: Meta Paths for HeteRec

<table>
<thead>
<tr>
<th>NetClus Products</th>
<th>Cluster 1</th>
<th>Cluster 2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Top 3</strong></td>
<td>Tri-State Corn on the Cob: 0.0065&lt;br&gt;High Top Cron on the Cob: 0.0063&lt;br&gt;Hermanos Garlic: 0.0060</td>
<td>Cormorant Scented Toilet Tissue: 0.0042&lt;br&gt;Red Wing Soft Napkins: 0.0036&lt;br&gt;Booker Low Fat Cottage Cheese: 0.0036&lt;br&gt;Hermanos Garlic: 0.0000&lt;br&gt;High Top Asparagus: 0.0000&lt;br&gt;Ebony Beets: 0.0000</td>
</tr>
<tr>
<td><strong>Bottom 3</strong></td>
<td>Bravo Turkey Noodle Soup: 0.0003&lt;br&gt;Blue Label Rice Soup: 0.0003&lt;br&gt;Better Noodle Soup: 0.0000</td>
<td></td>
</tr>
</tbody>
</table>

Table 4: Results of NetClus for clusters 1 and 2

<table>
<thead>
<tr>
<th>NetClus Products</th>
<th>Cluster 3</th>
<th>Cluster 4</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Top 3</strong></td>
<td>Better Vegetable Soup: 0.0125&lt;br&gt;Blue Label Chicken Ramen Soup: 0.0109&lt;br&gt;Pleasant Turkey Noodle Soup: 0.0105&lt;br&gt;Hermanos Broccoli: 0.0000&lt;br&gt;Fast Golden Raisins: 0.0000&lt;br&gt;Tell Tale Mandarin Oranges: 0.0000</td>
<td>Better Noodle Soup: 0.0030&lt;br&gt;Super Chunky Peanut Butter: 0.0028&lt;br&gt;Great Rye Bread: 0.0028&lt;br&gt;American Foot-Long Hot Dogs: 0.0012&lt;br&gt;Red Wing Scented Toilet Tissue: 0.0012&lt;br&gt;Walrus Light Beer: 0.0012</td>
</tr>
</tbody>
</table>

Table 5: Results of NetClus for clusters 3 and 4

The algorithm appeared to scale linearly with the numbers of objects in the network. Unfortunately, we experienced high variance in our execution times. We suspect this is due to IO and other parts of our system with high constant execution time complexity.

### 4 Conclusion

In conclusion, our algorithm NetRec performed better than the two tested alternatives, NetRec-NMF and HeteRec. The created algorithm likely performed better due to the information loss of using both NMF as well as only clustering users based on the factorized user matrix in HeteRec. In our algorithm we minimize loss by utilizing the full diffusion matrices rather than the factorized estimations and also do not use the induced homogeneous data from NMF to create user clusters. We believe that there are still other refinements to be had that could enhance the performance of the recommendation score further. The algorithm itself could use improvement as far as having to use two different network schemas in order to create recommendations so we could enhance this by utilizing an algorithm that would use a single network schema. This can further be described in the future work section below.

#### 4.1 Future Work

In the future we would like to find a better method that creates a mutually enhancing algorithm that would both learn user clusters as well as meta paths at the same time.
We would also like to create a way of predicting a time-based score that would focus on items that could be purchased periodically. Meta path induced prediction\cite{7} described by Sun, Han, Aggarwal, and Chawla who developed a link prediction algorithm in order to find out not only whether a link will occur but also when it should be created or added could be used as a resource for enhancing the algorithm for a time based model. As for the periodic recommendation we could learn from algorithms such as eperiodicity.\cite{12}

References


[7] Yizhou Sun, Jiawei Han, Charu C. Aggarwal, and Nitesh V. Chawla. When will it happen?: Relationship prediction in heterogeneous information networks. In Proceedings of the Fifth ACM International Conference on Web Search and Data Mining, WSDM ’12, pages 663–672, New York, NY, USA, 2012. ACM.

[8] Yizhou Sun, Jiawei Han, Xifeng Yan, Philip S. Yu, and Tianyi Wu. Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. In In VLDB 11, 2011.

[9] Yizhou Sun, Yintao Yu, and Jiawei Han. Ranking-based clustering of heterogeneous information networks with star network schema. In Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD ’09, pages 797–806, New York, NY, USA, 2009. ACM.

