Abstract
In customer relationship management (CRM), online recommender assumes an important role of suggesting the right product or information to the right customer automatically. Hence customers are empowered with the choices that are predicted to be preferred by the system. The underlying technique is often a collaborative filtering (CF) algorithm that harvests both information from similar products and peer users for inferring a suggested item out of many for a user. CF and its variants have been studied extensively in the literature on online recommender; however, most of the works were based on Web 1.0 where all the information necessary for the computation is by default assumed to be always available, as if it were readily stored in a database. In the distributed environment of Web 2.0 such as social networks, the required information by CF may either be incomplete or scattered over different sources. This poses certain computational challenges for Web 2.0 recommender.

The contribution of this paper is a novel model of CF that attempts to meet these challenges. This model uses a trust-network as well as emergence of information from multiple sources for utilizing CF for a recommender in a social network. This integrated model is called Multi-Collaborative Filtering Trust Network by various sources, M-CFTN in short.

Keywords: Collaborative filtering, Online recommender, Social Network, Facebook.

1 Introduction
Social networks, for instance Facebook, have recently gained a tremendous amount of popularity and they are still undergoing an exponential growth [1]. This new platform commonly known as Web 2.0 is characterized by cultivating open communication among users and sharing information with friends. Many companies have already established their virtual presence in social networks and started outreaching to potential customers via networks of friends of friends.

In such a distributed environment, CRM would face additional difficulties, though the Web marketing principles may be the same as those of CRM for websites, such as recommending preferred products for customers according to their tastes [2]. CF is a well-known technique for implementing such a recommender that is based on peer users information and product information. The technical challenges pertaining to the availability of data required in CF computation nevertheless come in two ways. Firstly, peer users information may not be fully available for CF from a social network. In contrast, an e-business in Web 1.0 is often equipped with a relational database that contains a full set of well-structured customer records. The users who present similar activities can be easily grouped together by SQL queries. The mechanism of social networks, however, is a partially connected graph with levels of trust supposedly proportional to the distance apart between any pair of users. The main challenges lie not only in the differential trust level, but the fact that the product
information may be scarce in any given clique of friends, especially when the product under consideration for recommendation is an unpopular or uncommon item. For example, an accountant wishes to pick up a new hobby in piccolo. In Facebook, his circle of friends including his trusted ones like family members and cohorts at work may have insufficient or zero information about piccolos. Information about some specific products however would be more abundant in other websites such as epinions.com or ebay.com. Therefore it makes sense that product information should be referred from other sources, instead of from the social network alone. In theory, the more the better it is that the CF algorithm can tap on to other sources for gathering product information. This technical challenge on the scarcity of product information leads us to combine social networks which are strong in expressing social relations between users (but not products) with product-oriented websites, for operating CF. The second challenge is known as Cold Start in CF. Inherent from the nature of partially connect graph, Cold Start may get more severe in social networks when a user has few friends to start with. Product information is extracted from posts and comments from users in a social network. When a user has a small circle of friends, the amount of comments and posts interacted around them would proportionally be little. The chance that the contents of the posts would involve a specific product is extremely slim.

In the light of the above-mentioned limitations in a social network environment, a new version of CF model is needed to overcome the challenges so that online recommender in Web 2.0 can be made efficient. The focus of this paper is a new design of CF model that infuses of convergence of data from multiple sources, trust-network for inferring the relations between users, and a mechanism for solving the Cold Start problem. This integrated model aims at providing possibilities for implementing online recommender in Web 2.0.

2 Research Motivations
Online recommender is a demanding research topic in a research community. To the best of the authors’ knowledge no precedent work has emerged on designing a CF model specifically for social networks, although social networks are growing fast to become an important platform for e-marketing. The core structure of CF basically remains unchanged in the new version for Web 2.0; the challenges mentioned in the previous section however are needed to be dealt with. We propose in this paper an integrative model of several techniques, which is an enhanced version of CF for this purpose.

From a high level, the operational flow of an online recommender is depicted in Figure 1. The workflow is consistent with that of business intelligence analytics [3, 4], which pulls in information from various sources for estimating a result in the form of a predicted item appealing to the user. Rules are derived from processing the relevant information, and they govern about the choice of the recommended items to be displayed to the user.

Figure 1: Work flow of a typical online recommender
In the current version of Facebook at the time of writing, its layout design displays recommended items in the form of advertisements in the right column. As an example shown in Figure 2, the advertisements were carefully selected for display according to the profile of the user (www.facebook.com/fongsimon).

The criteria of choice of advertisements, adopted by the recommender or some similar engine, include the country where the user is from, his occupation, his interests and something that is mutually liked by the user and his friends – as indicated by the history of “Like” by his friends. Besides these criteria that are mainly based on the user’s profile information, the CF algorithm is able to pool in new products that are similar to those associated with the user plus the products that are favored by his similar peers. From there, the algorithm derives an appropriate choice of new items that the user has not experienced before.

The future of Web 2.0 recommender should be based on CF, hence it can provide more refined and probably more accurate recommendations than now in Web 2.0.

Recently, a new Open Graph Protocol (OGP) is announced to be available by Facebook through f8 (apps.facebook.com/feightlive). That means, in a Facebook application, it is technically possible to integrate contents from other websites via the same user. Now products and consumers can be connected semantically, and thus all the data related to the same user can be shared by all the connected websites.

With this kind of interoperable connectivity between Facebook and other websites, extraction of data from other sources with the user centered is feasible. However, interoperable connectivity still brings up a series of questions regarding how such available data by OGP could be used in the CF model at the algorithmic level. For example, how to share all the data related to the user via the same user? How to model these websites quantitatively in the order of relevance to the subject, and more importantly integrate them into the CF algorithm? How to take into account of the temporal proprieties of relations between users? These are prominent questions to be answered before a Web 2.0 recommender system powered by our integrated CF algorithm can be effectively deployed in e-marketing applications on a social network.

Both missing values and trust metrics have been studied for decades, while recently a temporal issue becomes a hot research topic. Temporal information could be incorporated into widely used prediction algorithms so it can be adapted to changes in several characteristics over time. By considering temporal elements, it helps to improve the accuracy of recommendation prediction over an ever changing social network environment.

This paper is organized as follow: Section 3 reviews related works on mutual information extraction, temporal issue and current recommendation algorithms. Section 4 shows our proposed CF model which we call it M-CFTN. Section 5 is a discussion about the technical implementation. A conclusion is drawn at the end.

3 Related Works
To deal with Cold Start, sometimes known as Cold Boot-up problem, Thomas Sandholm, Hang Ung, Christina Aperjis, Bernardo A. Huberman presented their geo-aware rating and incentive mechanism using Kendall Rank Correlation Coefficient algorithm [5], which both improves the ranking and aggregate rating
of a series of location-dependent Web pages. Maunendra S. Desarkar, Sudeshna Sarkar, and Pabitra Mitra proposed a memory based algorithm [6] by comparing performances with four existing approaches: User based Pearson Correlation (UPCC), Item based Pearson Correlation Coefficient (IPCC), Random Walk Recommender (RWR) and Somers Coefficient based algorithm. While P. Massa considered both cold boot-up and trust issue, he extended recommendation systems to Trust-aware recommendation systems [7] by propagating a trust matrix in addition to the ratings matrix. Based on Massa’s research, we measured the trust both by relation and reputation last year in [8].

Recently, researchers increasingly realized temporal relation: Pedro G. Campos, Alejandro Bellogin, Fernando Diez, J. Enrique Chavarriaga proposed simple Time-Biased KNN-based recommendation algorithm [9], which consider simple strategies for taking into account the temporal context for recommendations, mainly based on variations of the KNN algorithm. Antoine Brnner, Bruno Pradel, Nicolas Usunier proposed time-dependent collaborative personalized recommendation algorithm [10], which modeled the short-term evolution of the probability and forecasted these probabilities on the test week. Nathan N. Liu, Min Zhao, Evan Xiang, Qiang Yang proposed a Time-aware Matrix Factorization Model (Time SVD) [11], which extended the widely used neighborhood based algorithms by incorporating temporal information and developed an incremental algorithm for updating neighborhood similarities with new data.

<table>
<thead>
<tr>
<th>Authors</th>
<th>Work Title</th>
<th>Area of Work</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thomas Sandholm, etc.</td>
<td>Global Budgets for Local Recommendations[14]</td>
<td>✓</td>
</tr>
<tr>
<td>Zeno Gantner, etc.</td>
<td>Factorization Models for Context-Time-Aware Movie Recommendations[20]</td>
<td>✓</td>
</tr>
<tr>
<td>Pedro G. Campos, etc.</td>
<td>Simple Time-Biased KNN-based recommendations[17]</td>
<td>✓</td>
</tr>
<tr>
<td>Maunendra S. Desarkar, etc.</td>
<td>Aggregating Preference Graphs for Collaborative Rating Prediction[15]</td>
<td>✓</td>
</tr>
<tr>
<td>Antoine Brenner, etc.</td>
<td>Predicting Most Rated Items in Weekly Recommendation with Temporal Regression[18]</td>
<td>✓</td>
</tr>
<tr>
<td>Nathan N. Liu, etc.</td>
<td>Online Evolutionary Collaborative Filtering[19]</td>
<td>✓</td>
</tr>
<tr>
<td>P. Massa, etc.</td>
<td>Trust-aware Recommender Systems[16]</td>
<td>✓</td>
</tr>
<tr>
<td>Wei Chen, Simon Fong</td>
<td>M-CTFN</td>
<td>✓</td>
</tr>
</tbody>
</table>

Table 1: Summary of Literature Review

4 Proposed Model and Algorithm Details
This section describes details of our proposed model which includes mutual information extraction, similarity and weights calculations, and applying Hopfield net spreading activation.

4.1 Proposed Model
This general model represents the inter-activities and relationships between Facebook and other websites. Each oval is one individual website, which can be viewed as a two-layer model. The upper layer is user layer, representing different users who are connected
with one another. The bottom layer is called the items layer, standing for different items under several categories. Users are associated by relationships in the upper layers, and items are connected by similarities – items of the same kind are grouped in the same category. Inter-activities between the users in the upper layer and the items in the lower layer do exist. For example, in Facebook, user A may trade an item B via an e-marketplace or posted a comment about item B. In epinions.com, for another example, user C wrote comments about item D, etc. Figure 3 shows an example of Facebook as a centered site that has an internal view of relations between its users and products, as well as their relations to other websites.

\[ MI_c = \frac{f_c}{f_{left} + f_{right} - 2f_c}, \] where \( f \) represents the frequency of a given word or phrase, and \( MI_c \) stands for the mutual information, which is the probability of co-occurrence of the pattern \( c \) (Dianye Private Middle School), relative to its left sub-phrase (Dianye Private Middle) and the right sub-phrase (Private Middle School). If \( MI_c \) is high and close to 1, it stands for the phrase \( c \) is more likely to form a phrase than its left and right sub-phrases alone. On the contrary, if \( MI_c \) is low and close to 0, phrase \( c \) is unlikely to form a phrase.

4.3 Person Correlation Coefficient

PCC was used in many recommendation systems [5], because it can be easily implemented and can achieve high accuracy, compared to other similarity algorithms.

In user-user (at the upper layer of the two-layer model) collaborative filtering, PCC is used to define the similarity between two users \( a \) and \( u \) based on the items they rated or related in common:

\[
\text{Sim}(a, u) = \frac{\sum_{i \in I(a) \cap I(u)} (r_{a,i} - \bar{r}_a) \cdot (r_{u,i} - \bar{r}_u)}{\sqrt{\sum_{i \in I(a) \cap I(u)} (r_{a,i} - \bar{r}_a)^2} \cdot \sqrt{\sum_{i \in I(a) \cap I(u)} (r_{u,i} - \bar{r}_u)^2}}
\]

where \( \text{Sim}(a, u) \) stands for the similarity between user \( a \) and user \( u \), and \( i \) belongs to the subset of items which user \( a \) and user \( u \) both rated or related. \( r_{a,i} \) is the rate user \( a \) gave item \( i \), and \( \bar{r}_a \) represents the average rate of user \( a \). From this definition, user similarity \( \text{Sim}(a, u) \) is ranging from [0,1], and a larger value means user \( a \) and \( u \) are more similar.

In item-item (at the bottom layer of the two-layer model) collaborative filtering, the similarity between two items \( i \) and \( j \) can be expressed as:

\[
\text{Sim}(i, j) = \frac{\sum_{u \in U(i) \cap U(j)} (r_{u,i} - \bar{r}_i) \cdot (r_{u,j} - \bar{r}_j)}{\sqrt{\sum_{u \in U(i) \cap U(j)} (r_{u,i} - \bar{r}_i)^2} \cdot \sqrt{\sum_{u \in U(i) \cap U(j)} (r_{u,j} - \bar{r}_j)^2}}
\]
where Sim(i,j) is the similarity between item i and item j, and u belongs to the subset of users who both rated item i and item j. r_{ui} is the rate user u gave item i, and \bar{r}_i represents the average rate of item i. Like user similarity, item similarity Sim(i,j) is also ranging from [0,1].

Considering the temporal issue, we adopt a modified version from [10]. The strength of a relation is supposed to gradually fade off over time. In other words, the more recent the relation activity occurred the stronger it is. A comment made today has a stronger impact than last month in this temporal version of CF. Similarity computation can be expressed as follow:

\[
\text{Sim}(i,j) = \frac{\sum_{u \in U, r_{ui} > 0} (f^u_{ai}(t) \cdot r_{ui})(f^u_{aj}(t) \cdot r_{uj})}{\sqrt{\sum_{u \in U, r_{ui} > 0} (f^u_{ai}(t) \cdot r_{ui})^2 \sum_{u \in U, r_{uj} > 0} (f^u_{aj}(t) \cdot r_{uj})^2}}
\]

Neighborhood Computation, for each item i, find the k items that are most similar to i and as follow: modified version from [10]. The strength of a relation is also ranging from [0,1].

Score Prediction can be expressed as follow:

\[
\hat{r}_{ui}(t) = \bar{r}_u(t) + \frac{\sum_{j \in N_i \cap N_u} S_{ij}(t) \cdot f^\beta_{ij}(t) \cdot r_{uj}}{\sum_{j \in N_i \cap N_u} S_{ij}(t) \cdot f^\beta_{ij}(t)}, \quad \text{with } \alpha \leq 10, \beta \leq 10
\]

where the parameter \(\alpha\) controls the decaying rate.

\[
f^{\alpha}_{ui}(t+1) = \gamma \cdot f^{\alpha}_{ui}(t) \quad \text{where the constant } \gamma = e^{-n}
\]

denotes the constant decay rate. Thus, as we compared in Table 1, we consider a combination of temporal relevance, trust factors, cold boot-up and multiple sources those four areas. The final algorithm is as below.

\[
\text{Pr}_u(g) = \alpha \bar{r}_u(t) + \beta \text{sim} + \gamma \mathcal{T}
\]

where Pr_\(u(g)\) is the total prediction score of one products or advertisement for user i, and \(\alpha, \beta, \gamma\) are different weights we issue to different areas, (temporal, trust, etc.) for indicating their relative importance. We use the same algorithm \(T\) as in [8], which both represents trust by relation and reputation.

### 4.4 Hopfield Algorithm

Based on [14, 15], Hopfield net algorithm was found to be suitable for semantic information retrieval. The Hopfield algorithm (or Hopfield Net) performs a parallel search, where each node is initiated in parallel and initiation values from different sources are combined for each individual node. Neighboring nodes are traversed in order until the initiation levels of nodes converge in the network.

In this present model our embedded two layer network of user layers and item layers can be viewed as interconnections of neurons and synapses in the Hopfield Net, where neurons stand for users or items and synapses represent weighted links between pairs of users and items. Moreover, Hopfield Net Algorithm could adapt to our M-CFTN model, where neurons represent each individual websites and synapses represent weighted links between those websites.

When there is no difference in terms of output between two iterations, this algorithm terminates.

**Input Initialization:** It first retrieves all the items that used to have connections with users as the starting nodes in iteration 0.

\[
\mu_i(0) = x_i, 0 \leq i \leq n - 1
\]

\[
\mu_i(t) \quad \text{is the weight of node } i \text{ at iteration } t \text{ and } x_i,\text{ which ranges from 0 to 1, indicates the input weight for node } i. \text{ At iteration 0 (when } t=0), \text{ all starting nodes are assigned weight of 1, all other nodes are assigned weight 0.}
\]

**Activation, weight calculation and iteration:**

The output of each node is computed as:

\[
\mu_i(t+1) = f_{\mu} \left[ \sum_{j=0}^{n-1} t_{ij} \mu_j(t) \right], 0 \leq j \leq n - 1, \quad \text{where } t_{ij}
\]

is the association weight between node i to node j when there is a link that points from node i to node j, otherwise \(t_{ij}\) is 0.

Within a certain amount of time, we keep top 100 nodes (ranked by \(\mu_i(t)\) as activated nodes for the next
iteration. Thus \( n = 50 \). The \( f_i \) is the SIGMOID transformation function \([16, 17]\) and is shown below:

\[
f_i(net_i) = \frac{1}{1 + \exp \left[ \frac{\theta_j - net_i}{\theta_b} \right]},
\]

\( net_j = \sum_{t=0}^{n-1} t_i \mu_i(t), \theta_j \)

where \( \theta_0 \) is used as a threshold or bias, and \( \theta_b \) is used to modify the shape of the SIGMOID function. \( \theta_j \) and \( \theta_b \) need experiments to be adjusted according to the characteristics of the graph and magnitude of the association weights in the graph.

**Convergence Criteria**: The above process is repeated until there is no change in output between two iterations, which can be checked by

\[
\sum_{j=0}^{n-1} \left| \mu_j(t+1) - \mu_j(t) \right| \leq \epsilon,
\]

where \( \epsilon \) is the maximal accepted error, a small number. In the final iteration, only top 100 items that had not been connected with that user are retrieved as the list of recommendation.

5 **Experiment**

We launched three projects to observe inter-activities and relationships between multiple social networks via IssueCrawler. For project 1, we named it as Multi-social network, we set starting points as Figure 4 shows. Those websites are individual websites, without mutual information being connected. Their inter-activities and relationships are shown in Figure 5.

For project 2, we extracted mutual information for a user account under test called “joeholland”, between epinions.com and facebook.com. The user Joe Holland had posted actively in both sites. The experimental results are showed as follows.
For project 3, we extracted mutual keywords "obama" to observe the inter-activities and relationship among CNN, YouTube, Facebook, and Yahoo!, by using the same techniques. Results are shown as below.
As observed from the above graphs, we could find that, (1) by comparing project 1 and project 2, there are more inter-activities and stronger relations between websites with mutual information or keywords; (2) in project 2 and project 3, higher term frequency is yielded. This may be explained by the impact of reputation and popularity in the networks, e.g. Obama is much more famous and popular than joeholland, there are closer links are found among them.

In this experiment, we analyzed the whole process of data extraction from multiple sources, and considered other three areas of works, which are temporal relation, trust network, and cold boot-up. We integrated these four features into one algorithm. The experiment demonstrated that our model is technically possible.

6 Conclusion

Online recommender aims at assisting online users to make decisions in e-commerce by offering them recommended products or information. In this new era of Web 2.0 where users have their online presences that are connected as networks of friends, online recommender naturally would resort to gathering information from a user’s close associates in a social network, and infer from those information for making a recommendation to the user. So far little work has been done on investigating into this technique. This paper proposed a collaborative filtering model especially for recommenders to be used in social networks. Several technical challenges are identified that are inherent from the nature from social networks such as sparsity of product information in a partially connected social network, Cold Boot-up and referring information from other websites. Solutions were presented in this paper for solving these problems. The model is called M-CFTN in short that is an integrated version of collaborative filtering algorithm combined with trust-network, and enabled by referring information from other websites should the information within the social network is incomplete. Some experimental tasks were performed by using IssueCrawler software program to check out the performance of M-CFTN. The temporal issue of the link strengths was taken into account as well. In the experiment, Facebook Open Graph was used as a case tool here to help illustrating the concepts.

REFERENCES


