

Providing Recommendations in SCENS

Rong Zhang¹, Sheng Zhang¹, Song Ye¹, Yan Zhao¹,
James Ford¹, and Fillia Makedon¹

Department of Computer Science, Dartmouth College

Abstract. SCENS (Secure Content Exchange Negotiation System) is a web based data sharing system that enables data owners to maintain control of their original data while negotiating the conditions under which sharing can be done. This allows for data tracking, data provenance and data copyright enforcement. In this paper, we introduce our current work on incorporating a recommendation component in SCENS. The objective of this component is to help users find the most reliable, valuable, important and interesting data quickly and easily. Four implemented recommendation algorithms (Naive item average, User-based, Item-based, and Singular Value Decomposition based) in our recommendation component are discussed.

Keywords: SCENS, recommender systems, collaborative filtering

1 Introduction

Data sharing of sensitive or highly valuable informational resources requires new models of negotiation to promote communication with built-in incentives, secure authentication, new metadata standards, and new metrics of evaluation. SCENS, a Secure Content Exchange Negotiation System, is a system we have been building to enable the exchange or sharing of private (sensitive) multimodal digital data that reside in distributed digital repositories. These data may include raw data, derived data, tools, methods or services.

SCENS was originally designed to support data sharing in neuroscience research, where sharing is important in promoting discovery and collaboration. Researchers have resisted sharing primary data in this field traditionally because neuroscience data are not only private but also of high value due to production costs. To encourage the collaboration while protecting the right of the owner, SCENS provides a flexible, user-centered alternative to large, transnational archiving of published only results that provide limited access control to the data owner once he has given the data. SCENS negotiation services provide a secure mechanism of reaching agreements on conditions for primary data sharing. In this respect, our system offers a novel way for digital libraries composed of sensitive data to grow while enhancing collaboration of users and providing data usage tracking facilities.

When there are too many data resources in SCENS, it becomes very difficult for users to find resources that are really valuable and interesting to them. In

order to ameliorate such information overload, we introduce a recommendation component in SCENS. The recommendation component allows users to leave feedback (ratings) on the data resources they have requested and used, and to share their ratings with each other. Based on these ratings, the recommendation component predicts which items have the highest likelihood of being useful and valuable to users.

Four recommendation algorithms are implemented in our recommendation component. A naive item average algorithm recommends items based on their rating averages. A user-based algorithm and an item-based algorithm make recommendations by exploiting information from those users and items that are similar to the current user and the current item. A Singular Value Decomposition (SVD) based algorithm computes a low-dimensional linear model from all observed ratings and uses the computed model for recommendations.

2 Secure Content Exchange Negotiation System (SCENS)

The aim of SCENS is to facilitate data sharing through negotiation on metadata information derived from data of interest (primary data). Primary data exchange is done between two parties once they have reached agreement, and, until then, primary data remain in the control of data owners.

The SCENS framework contains the following three functional units.

1. A front-end data entry interface for data providers to enter or upload metadata representations of the data repositories to be shared. In this case, metadata descriptions of the original data are used to describe different aspects of the original data, *e.g.*, origin, size, location, type, date, complexity, and methods of collection.
2. The negotiation system that forms the core of the SCENS framework and is described in further detail below.
3. A suite of functionalities that add value to the collected information and make the system appear more intelligent, thus motivating the user to enter data sharing requests through the system. Examples of such functionalities include tools to enable query-by-example in metadata format type queries, and the recommendation component. In addition, These SCENS services also constitute strong incentives for its users.

SCENS negotiation system has a flexible 3-layer service structure; each layer implements distinct functions and providing different levels of negotiation services for different types of users [8].

Layer 1 is a traditional web-based negotiation system for human beings. It provides a user friendly interface for negotiating parties to negotiate with each other and if possible, to help them reach the agreement on data sharing conditions. The services provided by Layer 1 to support the negotiation process include Negotiation Conditions Registration (NCR) and Negotiation Strategy Support (NSS). NCR enables users to register the conditions that can be used for the

negotiation process. Layer 1 also provides some basic negotiation agents, which are actually user customizable utility functions. NSS allows users to customize these negotiation agents by multiple parameters, such as the weights assigned to different conditions. Layer 1 is the fundamental layer of the whole negotiation system; some services, such as NCR, can only be provided by layer 1.

Although Layer 1 provides negotiation service in most conditions, it is not fully customizable. If the strategy is very complex, it simply can not be expressed by parameters. Layer 2 is designed to support fully customizable negotiation strategy. Users are allowed to have their own negotiation agents to implement any negotiation strategies. The negotiation agents, which are treated as web service consumers and run on the client side, conduct negotiations with other negotiation agents or human beings through web services. Layer 2 communicates with negotiation parties through SOAP (Simple Object Access Protocol), a lightweight protocol for exchange of information in a decentralized and distributed environment. Figure 1 shows the relation between Layer 1 and Layer 2.

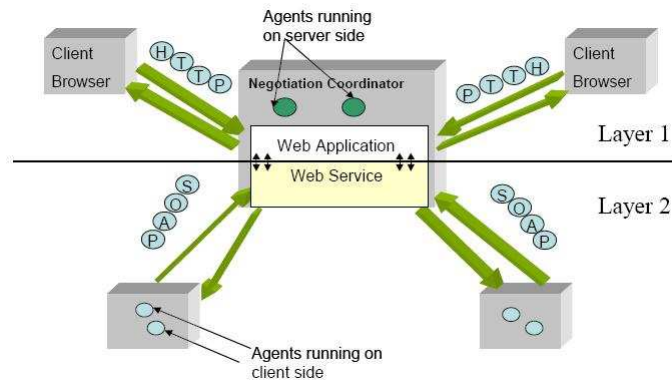


Fig. 1. Layer 1 provides web-based negotiation services for human beings; Layer 2 interacts with negotiation agents through web services

Layer 3 is designed to provide an open and automated negotiation environment. DAML+OIL, a language for creating ontology and marking up information, is used to define a negotiation ontology, which allows agents to acquire knowledge includes negotiation protocols, negotiation proposals and conditions, *etc.* Agents communicating with Layer 3 can be used in any negotiation activities given the proper negotiation ontology. In Layer 2, in contrast, the knowledge about negotiation rules is actually hard-coded into the agents.

3 Recommendation Component

The objective of our recommendation component is to help users to find the most reliable, valuable, important and interesting information quickly and easily.

Recommendation algorithms are usually classified according to how recommendations are made into the following two categories. **Content-based recommendations:** the user will be recommended items judged similar to the ones the user preferred in the past. **Collaborative filtering recommendations:** the user will be recommended items that people judged to have similar tastes and preferences liked in the past. Our recommendation component mainly focuses on collaborative filtering (CF) recommendations because it is the most popular category.

In its most common formulation, the recommendation problem is reduced to the problem of predicting ratings for the items that a user has not seen before. Once we can estimate a user's ratings for all unrated items, we can recommend the items predicted to receive the highest ratings. Figure 2 gives a simple recommendation scenario composed of 8 users and 6 items. User preference on items are expressed using discrete numerical values from 1 to 5, where 5 represents the fact that a user likes the corresponding item very much. In order to make recommendations to user Alice, we compute predicted ratings for Alice on those items that she has not rated yet and then recommend items that have the highest predicted ratings.

	Item 1	Item 2	Item 3	Item 4	Item 5	Item 6
Alice	5		3			?
User1			4		4	
User2		3			2	1
User3	4			1		
User4	3			3		2
User5		3				1
User6	5	3				
User7				4		2

Fig. 2. A simple recommender system scenario. In order to recommend items to user Alice, predictions to those missing entries corresponding to the row of Alice are computed first, and then items are recommended to Alice based on their predictions.

4 Collaborative Filtering Algorithms

In this section, we introduce four collaborative filtering algorithms that are used for providing users with recommendations in SCENS.

4.1 Naive Item Average Algorithm

In the naive item average algorithm, rating average is computed for each item based on all users' ratings. Those items that get highest rating averages will be recommended. Note that recommendation results of this algorithm will be always the same to all users.

4.2 User-based Algorithm

The user-based CF algorithm [2, 4] first computes the correlations between users using a mean-adjusted Pearson correlation, and then combines a weighted average of the k nearest neighbors' ratings to produce a prediction. More precisely, a predicted rating $P_{i,j}$ for user i on item j is computed by

$$P_{i,j} = \bar{A}_i + \frac{\sum_{u=1}^k w_{i,u} \cdot (A_{u,j} - \bar{A}_u)}{\sum_{u=1}^k |w_{i,u}|},$$

where $A_{u,j}$ is user u 's rating on item j , \bar{A}_i is user i 's average rating, and $w_{i,u}$ is the correlation between users i and u .

Several different similarity weighting have been used to compute $w_{i,u}$. The most common weighting measure used is the *Pearson correlation coefficient*. Pearson correlation measures the degree to which a linear relationship exists between two variables. It is derived from a linear regression model that relies on a set of assumptions regarding the data, namely that the relationship must be linear, and the errors must be independent and have a probability distribution with mean 0 and constant variance for every setting of the independent variable [2, 3]. If the Pearson correlation coefficient is used, we have

$$w_{i,u} = \frac{\sum_{l=1}^h (A_{i,j_l} - \bar{A}_i)(A_{u,j_l} - \bar{A}_u)}{h\sigma_i\sigma_u},$$

in which h is the number of common items that both user i and u have rated, and j_l is the index of the l th common item. Both rating averages (\bar{A}_i , \bar{A}_u) and rating standard deviations (σ_i , σ_u) are computed based on those common items only.

4.3 Item-based Algorithm

The item-based algorithm [5] computes and uses similarities between items rather than users. The formula used to compute a prediction is:

$$P_{i,j} = \frac{\sum_{v=1}^k s_{v,j} \cdot A_{i,v}}{\sum_{v=1}^k |s_{v,j}|}.$$

Here, $s_{v,j}$ is the similarity between items v and j . Our implementation uses an *adjusted cosine correlation* to compute similarities. That is,

$$s_{v,j} = \frac{\sum_{u \in U} (A_{u,v} - \bar{A}_u)(A_{u,j} - \bar{A}_u)}{\sqrt{\sum_{u \in U} (A_{u,j} - \bar{A}_u)^2} \sqrt{\sum_{u \in U} (A_{u,v} - \bar{A}_u)^2}},$$

where U denotes the set of all users.

4.4 Singular Value Decomposition based Algorithm

Singular Value Decomposition (SVD) was first introduced into recommendation systems in [1] and [6]. The underlying assumption of applying SVD to a rating matrix is that observed ratings $A_{i,j}$ are combinations of ratings from a low-dimensional linear model (denoted as X) and Gaussian noise (with zero mean). That is,

$$A_{i,j} = X_{i,j} + Z_{i,j}, \text{ with } Z_{i,j} \text{ i.i.d. } \sim N(0, \sigma^2). \quad (1)$$

Since the rating matrix in the real world is incomplete and sparse, Srebro and Jaakkola [7] proposed an Expectation-Maximization (EM) algorithm to maximize the log-likelihood of all observed ratings A^o , that is $\log \Pr(A^o|X)$. In this paper, we use this SVD-based algorithm in which the EM algorithm is incorporated. The details of the derivation of this EM algorithm can be found in [7, 9]; an overview is given below.

In the Expectation step of the t th iteration, a filled-in matrix $A^{(t)}$ is formed where unobserved entries $A_{i,j}^{(t)}$ are equal to the corresponding values of the computed linear model in the previous iteration ($X_{i,j}^{(t-1)}$) and observed entries are unchanged from A . That is,

$$A_{i,j}^{(t)} = \begin{cases} A_{i,j} & \text{if } A_{i,j} \text{ is rated} \\ X_{i,j}^{(t-1)} & \text{otherwise.} \end{cases}$$

In the following Maximization step, we perform SVD on this filled-in matrix $A^{(t)}$ to get $A^{(t)} = USV^T$. The updated linear model $X^{(t)}$ is computed as

$$X^{(t)} = U_k S_k V_k^T,$$

where U_k , S_k , and V_k are matrices composed of the top k left singular vectors, singular values, and right singular vectors, respectively.

The above EM procedure is ensured to converge, which means that the log-likelihood of all observed ratings given the current model estimate is always nondecreasing. After the EM procedure finishes, a prediction is computed as the corresponding entry in the final computed model, *i.e.*,

$$P_{i,j} = X_{i,j}.$$

5 Implementation

The SCENS recommendation component contains two main functions: *Dataset Evaluation* and *Dataset Recommendation*.

Dataset evaluation enables users to leave feedback to data resources. The feedback information include text-based comments and discrete ratings ranging from 1 to 5. Such information will be used by collaborative filtering algorithms

Data Detail: zhengyi's TA code

Data Details	
ID	000000013
Name	zhengyi's TA code
Owner	zhengyi
Data Type	code
Registration Time	2005-01-18 17:08:27
Language	Java
Size	less than 20K
AuthorShip	yes
Category	Algorithms,Others,
Use Restriction	Non-disclosure (cannot disseminate or disclose details)
Rating	0

Rating

Rate: 1 2 3 4 5

Comment:

Fig. 3. The dataset evaluation interface of the recommendation component. Users can leave text-based comments and provides numerical ratings.

to provide recommendation results. Figure 3 shows the implemented dataset evaluation interface.

Dataset recommendation interface provides four collaborative filtering algorithms. The users are allowed to get recommendations by selecting their preferred algorithm. For each recommendation item, the basic information including data ID, data name, data owner, data type, and update time, are provided to help the user get a brief idea about this data. If the user is interested, she can further access the details of this data. We also provide the prediction value of each data resource, which indicates how much (as the recommendation component believes) the current user will rate this data resource. The range of the prediction value is consistent with the range of Dataset Evaluation. Figure 4 shows the web-based dataset recommendation interface.

6 Summary and Future Work

In this paper, we describe our work in building a recommendation component in SCENS to help users find valuable information they are potentially interested in. Our recommendation component has four implemented recommendation algorithms. The naive item average algorithm provides recommendations by their rating averages. The user-based algorithm (item-based algorithm) computes predictions by exploiting information from those users (items) similar to the current user (item). The SVD-based algorithm computes a low-dimensional linear model from all observed ratings and uses this model for predictions.

Welcome to SCENS,rong

Home | My Account | Pending Negotiation | Negotiation History | Register new data

Search By In [Advance Search >>](#) [All data >>](#) [Your data>>](#)

Select Recommendation Algorithm Item Average Cosine Correlation (item-based) Pearson Correlation (user-based) Singular Value Decomposition

There are 6 Recommendation Items for you:

ID	Name	Owner	Type	Date	Prediction
0000000015	yans data	yan	code	2005-01-19 16:38:21	5.0
0000000013	zhengyi's TA code	zhengyi	code	2005-01-18 17:08:27	4.25
0000000004	auto registration	bob	code	2004-08-25 16:58:45	4.0
0000000005	auto segmentation	bob	code	2004-08-25 16:58:45	3.0
0000000012	Thesis materials	jford	code	2005-01-18 16:58:00	3.0
0000000001	Fast FT	alice	code	2004-06-18 00:00:00	1.8

Fig. 4. The dataset recommendation interface provides a user with a list of recommendation results in decreasing order of their prediction values. The user can also choose her preferred collaborative filtering algorithm from a list of candidate algorithms.

One of our future work is to evaluate the implemented recommendation component using the following popular metrics in recommender systems.

- Predictive accuracy metrics (*e.g.*, Mean Absolute Error) measure how close a recommender system’s predicted ratings are to true user ratings.
- Classification accuracy metrics (*e.g.*, Precision and Recall) measure the frequency with which a recommender system makes correct or incorrect decisions about whether an item is relevant (interesting) to a user.
- Receiver Operating Curve (ROC) plots the percentage of relevant items selected and shown versus the percentage on non-relevant items selected and shown. The area underneath this curve is known as the ROC area.
- The correlation between a recommender system’s predicted ratings and corresponding true user ratings.

7 Acknowledgment

This material is based in part upon work supported by the National Science Foundation under award number IDM 0308229 (Data Management of Protected Information for Data Sharing and Collaboration).

Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

1. Daniel Billsus and Michael J. Pazzani. Learning collaborative information filters. In *Proc. of the 15th Int. Conf. on Machine Learning*, pages 46–54, 1998.

2. Jonathan L. Herlocker, Joseph A. Konstan, Al Borchers, and John Riedl. An algorithmic framework for performing collaborative filtering. In *Proc. of the 22nd ACM SIGIR*, pages 230–237, 1999.
3. James T. McClave and Frank H. Dietrich II. *Statistics*. Dellen Publishing Company, 1988.
4. Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. GroupLens: An Open Architecture for Collaborative Filtering of Netnews. In *Proc. of the ACM Conf. on Computer Supported Cooperative Work*, pages 175–186, 1994.
5. Badrul M. Sarwar, George Karypis, Joseph A. Konstan, and John Reidl. Item-based collaborative filtering recommendation algorithms. In *Proc. of the 10th WWW*, pages 285–295, 2001.
6. Badrul M. Sarwar, George Karypis, Joseph A. Konstan, and John Riedl. Application of dimensionality reduction in recommender systems—a case study. In *Proc. of the ACM WebKDD Workshop*, 2000.
7. Nathan Srebro and Tommi Jaakkola. Weighted low-rank approximation. In *Proc. of the 20th Int. Conf. on Machine Learning*, pages 720–727, 2003.
8. Song Ye, Fillia Makedon, Tilmann Steinberg, Li Shen, James Ford, Yuhang Wang, Yan Zhao, and Sarantos Kapadakis. SCENS: A system for the mediated sharing of sensitive data. In *Proc. of the 3rd ACM/IEEE Joint Conf. on Digital Libraries (JC DL)*, pages 263–265, 2003.
9. Sheng Zhang, Weihong Wang, James Ford, Fillia Makedon, and Justin Pearlman. Using singular value decomposition approximation for collaborative filtering. In *Proc. of the 7th IEEE Conf. on E-commerce*, pages 257–264, 2005.