INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH & DEVELOPMENT

ISSN 2278 – 0211 (Online)

Hybrid Recommender System for IT Governance Tasks

Dr. Carsten Mueller
Department of Information Technologies, University of Economics, Prague, Czech Republic

Abstract:
The concept "Business Informatics" became popular about fifteen years ago when the usual concepts had lost their generality, abstractness and actuality. Business Informatics is the scientific discipline targeting information processes and related phenomena in their socio-economical business context, including companies, organizations, administrations and society in general. Business Informatics is a fertile ground for research with the potential for immense and tangible impact. As a field of study, it endeavors to take a systematic and analytic approach in aligning core concepts from management science, organizational science, economics information science, and informatics into an integrated engineering science. This research paper is connected with recommender systems that are used for management of economic efficiency of business informatics with the support of Business Intelligence and Data Mining. Current recommender approaches are analyzed and a flexible model is developed. This model based on recommender systems is used to evaluate methods regarding requirements in the context of IT Governance.

Keywords*: business management systems, recommendation*

1. Introduction

Management of business informatics has been the subject of interest of researchers and IT practitioners for a number of decades. Recent increase in the complexity and heterogeneity of the technological infrastructure and enterprise applications, as well as increased number of information services provider options make the effective management of business informatics challenging [1].
Traditional ‘soft sciences’, such as sociology or economics, have their fast-growing branches, relying on the study of these newly available massive data sets [2], [3]. Information Filtering has become a necessary technology to attack the information overload’ problem. The capacity of computers to provide recommendations was recognized fairly early in the history of computing.

1.1. Model for Management of Business Informatics

The aim of the Model for Management of Business Informatics (MBI) [4] is to offer IT professionals who are responsible for the management of information technology in organizations a comprehensive methodological support based on industry best practices [1]. Application of this model has the potential to increase overall IT effectiveness, improve IT governance and efficiency of IT services, and result in better business performance. MBI considers important relationships and attributes that are relevant to information systems management and provides a suitable methodological basis [1]. Task is a key MBI component that represents a basic enterprise IT management unit. Task describes how to proceed in solving a particular IT management problem [1].

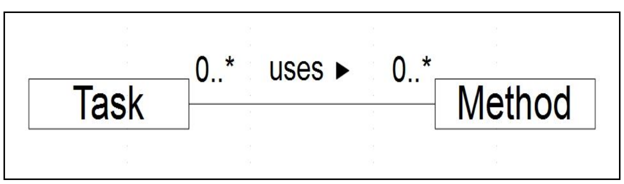

Figure 1: MBI - Relationship task and method

The MBI model defines a large number of tasks found in IT practice and in other IT management frameworks. MBI presents these tasks as a three-level hierarchy that corresponds to domains of IT management as illustrated

1.2. Recommender System

Since the appearance of the first papers on recommender systems in the mid-1990s a lot of research has been conducted in this area [5]. Recommendation systems [5], [6], [7], [8] provide suggestions for items that are of potential interest for a user.

The goal of a Recommender System is to generate meaningful recommendations to a collection of users for items or products that might interest them. Systems predict how a user would rate a specific item, based upon this information, and then recommend the items with the highest ratings. In order to predict these ratings, different prediction techniques are used. These prediction techniques come in three varieties: content-based, collaborative (or social) based and hybrid forms [5], [9].

Two principal paradigms for computing recommendations have emerged, namely content-based and collaborative filtering. Content-based filtering computes similarities between the active user a_i 's basket of appreciated products, and products from the product universe that is still unknown to a_i. It exploits only information derived from documents or item features (e.g. terms or attributes) [10], [11]. Product-product similarities are based on features and selected attributes. Collaborative filtering computes similarities between users, based upon their rating profile.

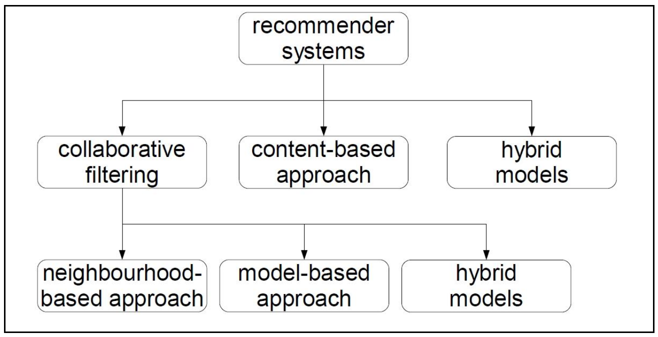

Figure 2: Recommender Systems – Hierarchy

The input to a Recommender System depends on the type of the employed filtering algorithm and belongs to one of the following categories:

Ratings, which express the opinion of users on items, are.

- Explicit ratings. Users are required to explicitly specify their preference for any particular item, usually by indicating their extent of appreciation on 5-point or 7-point likert scales. These scales are then mapped to numeric values, for instance continuous ranges $r_i(b_k) \in [-1,+1]$. Negative values indicate dislike, while positive values express the user's liking.
- Implicit ratings impose additional efforts on users. Consequently, users often tend to avoid the burden of explicitly stating their preferences and either leave the system or rely upon free-riding.

$$R_{mn} = \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ r_{21} & r_{22} & \cdots & r_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ r_{m1} & r_{m2} & \cdots & r_{mn} \end{bmatrix} \qquad (1)$$

User-item matrix is described as a $m \times n$ ratings matrix R_{mn}, where row represents m users and column represents n items. The element of matrix r_{ij} means the score rated to the user i on the item j.

	$method_1$	**$method_2$**	**$Method_3$**	**$method_4$**
$task_1$	3		4	2
$task_2$	4	5		4
$task_3$		4	2	3
$task_4$	3	5	2	

Table 1: Rating Matrix

Output of a Recommender System is either a prediction or a recommendation.

- Prediction (individual scoring) is expressed as a numerical value, $p_{a,j}$, which represents the anticipated opinion of active user u_a for item i_j.
- Recommendation is expressed as a list of x items, where $x \leq n$. This list includes only items that user has not purchased or rated.

2. Similarity
Two algorithms are applied to calculate the correlation between users: Pearson Correlation (PCC) and Vector Space Similarity (VSS).

2.1. Pearson Correlation
This method computes the statistical correlation (Pearson's r) between two user's common ratings to determine their similarity. The correlation is computed by the following:

$$PCC(u,v) = \frac{\sum_{i\in R_u \cap R_v}\left(r_{u,i}-\overline{r_u}\right)\cdot\left(r_{v,i}-\overline{r_v}\right)}{\sqrt{\sum_{i\in R_u \cap R_v}\left(r_{u,i}-r_u\right)^2\cdot\sum_{i\in R_u \cap R_v}\left(r_{v,i}-r_v\right)^2}} \qquad (2)$$

Where $r_{u,i}$ is the rating given to item i by user u . $R_u \cap R_v$ is the list of items rated by both users, u and v .
It measures the strength of linear dependency between two lines and is independent to changes in location and scale.

2.2. Vector Space Similarity
Vector Space Similarity metric is used for estimate the similarity between two instances u and v in information retrieval that the objects are in the shape of vector r_u and vector r_v .

$$VSS(u,v) = \frac{\sum_{i\in R_u \cap R_v} r_{u,i}\cdot r_{v,i}}{\sqrt{\sum_{i\in R_u \cap R_v} r_{u,i}^2}\cdot\sqrt{\sum_{i\in R_u \cap R_v} r_{v,i}^2}} \qquad (3)$$

Where $r_{u,i}$ is the rating given to item i by user $R_u \cap R_v$ is the list of items rated by both users, u and v .

3. Collaborative Filtering
Collaborative Filtering (CF) systems work by collecting user feedback in the form of ratings for items in a given domain and exploiting similarities in rating behavior amongst several users in determining how to recommend an item. The fundamental assumption behind this method is that other users' opinions are selected and aggregated in such a way as to provide a reasonable prediction of the active user's preference.
CF techniques depend on several concepts to describe the problem domain and the particular requirements placed on the system. The information domain for a collaborative filtering system consists of users which have expressed preferences for various items. A preference expressed by a user for an item is called a rating and is frequently represented as a (user, item, rating) triple.
CF methods are sub-divided into neighborhood-based and model-based approaches. Neighborhood-based methods are commonly referred to as memory-based approaches [12].

3.1. Neighborhood-based Collaborative Filtering
In neighborhood-based techniques, a subset of users is chosen based on their similarity to the active user, and a weighted combination of their ratings is used to produce predictions for this user.
The algorithm summarized in the following steps:
- *Assign a weight to all users with respect to similarity with the active user.*
- *Select k users that have the highest similarity with the active user - called the neighborhood.*
- *Compute a prediction from a weighted combination of the selected neighbor ratings..*

In step 1, the weight $w_{a,u}$ is a measure of similarity between the user u and the active user a . The most commonly used measure of similarity is the Pearson correlation coefficient between the ratings of the two users:

$$w_{a,u} = \frac{\sum_{i\in I}\left(r_{a,i}-\overline{r_a}\right)\left(r_{u,i}-\overline{r_u}\right)}{\sqrt{\sum_{i\in I}\left(r_{a,i}-\overline{r_a}\right)^2\sum_{i\in I}\left(r_{u,i}-\overline{r_u}\right)^2}} \qquad (4)$$

Where I is the set of items rated by both users, $r_{u,i}$ is the rating given to item i by user u , and $\overline{r_u}$ is the mean rating given by user u .
In step 3, predictions are computed as the weighted average of deviations from the neighbors mean:

$$p_{a,i} = \overline{r_a} + \frac{\sum_{u\in K}\left(r_{u,i}-\overline{r_u}\right)\times w_{a,u}}{\sum_{u\in K} w_{a,u}} \qquad (5)$$

Where $p_{a,i}$ is the prediction for the active user a for item i , $w_{a,u}$ is the similarity between users a and u , and K is the neighborhood or set of most similar users.
Similarity based on Pearson correlation measures the extent to which there is a linear dependence between two variables.

Alternatively the ratings of two users are treated as a vector in an m-dimensional space, and compute similarity based on the cosine of the angle between them:

$$w_{a,u} = \cos(\vec{r}_a, \vec{r}_u) = \frac{\vec{r}_a \cdot \vec{r}_u}{\|\vec{r}_a\|_2 \times \|\vec{r}_u\|_2} = \frac{\sum_{i=1}^{m} r_{a,i} r_{u,i}}{\sqrt{\sum_{i=1}^{m} r_{a,i}^2 \sum_{i=1}^{m} r_{u,i}^2}} \quad (6)$$

When computing cosine similarity, one cannot have negative ratings, and unrated items are treated as having a rating of zero.

4. User-Based Collaborative Filtering

User-based algorithms work on the assumption that each user belongs to a group of similar behaving users. The basis for the recommendation is composed by items that are liked by users. Items are recommended based on users' tastes (in term of their preference on items). The algorithm considers that users who are similar (have similar attributes) will be interested on same items.

The similarity between user U_a (the current user) and another user U_x is determined, for example, based on the Pearson correlation coefficient [13] where m_c is the set of items that have been rated by both users, r_{a,m_i} is the rating of user a for item m_i, and $\overline{r_a}$ is the average rating of user a. Similarity values resulting from the application of the formula above take values on a scale of $[-1,\ldots,+1]$.

4.1. Similarity

Cosine-based Similarity: Two users are thought of as two vectors in the n dimensional user-space. The similarity between them is measured by computing the cosine of the angle between these two vectors. In the $m \times n$ ratings matrix, similarity between users u and v, denoted by $sim(u,v)$ is given by

$$sim(u,v) = \cos(\vec{u}, \vec{v}) = \frac{\vec{u} \cdot \vec{v}}{\|\vec{u}\| \times \|\vec{v}\|} = \frac{\sum_{i=1}^{n} r_{ui} \cdot r_{vi}}{\sqrt{\sum_{i=1}^{n} r_{ui}^2} \sqrt{\sum_{i=1}^{n} r_{vi}^2}} \quad (7)$$

Correlation-based Similarity: Similarity between two users u and v is measured by computing the Pearson-r correlation $corr(u,v)$. For accurate computation the co-rated cases (cases where items rated by u and v) are isolated. Set of item, which both rated by u and v are denoted by I_{uv} the correlation similarity is given by

$$sim(u,v) = \frac{\sum_{i \in luv} (r_{ui} - \overline{r_u})(r_{vi} - \overline{r_v})}{\sqrt{\sum_{i \in luv} (r_{ui} - \overline{r_u})^2} \sqrt{\sum_{i \in luv} (r_{vi} - \overline{r_v})^2}} \quad (8)$$

$\overline{r_u}$ / $\overline{r_v}$ is the average of the u-th user / v-th user

$$\overline{r_u} = \frac{1}{|I_{uv}|} \sum_{i \in I_{uv}} r_{ui}, \ \overline{r_v} = \frac{1}{|I_{uv}|} \sum_{i \in I_{uv}} r_{vi}$$

4.2. Prediction

Prediction (individual scoring) is expressed as a numerical value, $p_{a,i}$, which represents the anticipated opinion of active user a for item i; $P_{a,u}$ is the similarity between users a and u; n is the number of users in the neighborhood.

$$p_{a,i} = \overline{r_a} + \frac{\sum_{u=1}^{n} (r_{u,i} - \overline{r_u}) \times P_{a,u}}{\sum_{u=1}^{n} P_{a,u}} \quad (10)$$

5. Item-Based Collaborative Filtering

Sarwar et al. proposed an item-based collaborative filtering algorithm [14]. One important step is to compute a similarity among items. The purpose of item-based collaborative filtering algorithm is to address data sparsity problem and model expansion of user-based algorithm. Item-based algorithms avoid this bottleneck by exploring the relationships between items first, rather than the relationships between users. Recommendations for users are computed by finding items that are similar to other items the user has preferred.

5.1. Similarity

Cosine-based Similarity: Two items i_p and i_q are considered as two column vectors in the user ratings matrix R. The similarity between items is measured by computing the cosine of these two vectors.

$$sim(i_p, i_q) = \cos(i_p, i_q) = \frac{i_p \cdot i_q}{\sqrt{\|i_p\| * \|i_q\|}} \quad (11)$$

Where $\cdot$ denotes the dot-product of two vectors.
Correlation-based Similarity:

$$sim(i_p, i_q) = \frac{\sum_{k=1}^{m} \left(r_{k,p} - \overline{r_p}\right) \cdot \left(r_{k,p} - \overline{r_q}\right)}{\sqrt{\sum_{k=1}^{m} \left(r_{k,p} - \overline{r_p}\right)^2 \cdot \sum_{k=1}^{m} \left(r_{k,q} - \overline{r_q}\right)^2}} \quad (12)$$

Where $r_{k,p}$ denotes the rating of user k on item p. $\overline{r_p}$ is the average rating of the item p.

5.2. Prediction
After computing the similarity between items, a set of most similar items to the target item is selected.

$$r_{a,k} = \frac{\sum_{k=1}^{K} \left(r_{a,t} \cdot sim(i_k, i_t)\right)}{\sum_{t=1}^{K} sim(i_k, i_t)} \quad (13)$$

Where $r_{a,k}$ denotes the prediction rating of target user a on item k. Only the K most similar items (K nearest neighbors of item k) are used to generate the prediction.

6. Singular Value Decomposition
Singular Value Decomposition [15] (SVD) is a powerful technique for dimensionality reduction. It is a particular realization of the Matrix Factorization approach and it is related to Principal Component Analysis (PCA). PCA is a classical statistical method to find patterns in high dimensionality data sets. The key issue in SVD decomposition is to find a lower dimensional feature space where the new features represent concepts and the strength of each concept in the context of the collection is computable. Reduction solves the Synonymy Problem, capturing latent relationships among users [16]. SVD allows to automatically deriving semantic concepts in a low dimensional space, it is used as the basis of latent-semantic analysis [17], a technique for text classification in Information Retrieval.
After generating a reduced dimensionality matrix, a vector similarity metric is applied to compute the proximity between users and to form the neighborhood. Neighborhood in the SVD-reduced space is created with the execution of the following steps:

- Initial user-item matrix R of size $m \times n$ is preprocessed in order to eliminate all missing data values to obtain normalized matrix R_{norm}.
- SVD of R_{norm} is computed to obtain matrices U, S and V, of size $m \times m$, $m \times n$, and $n \times n$. Relationship is expressed by: $R_{norm} = U \cdot S \cdot V^T$. Matrices U and V are orthogonal and span the column space and the row space. Matrix S is a diagonal matrix, called the singular matrix.
- Dimensionality reduction step is performed by keeping only k diagonal entries of the matrix S to obtain the $k \times k$ matrix S_k. Similarly, matrices U_k and V_k of size $m \times k$ and $k \times n$ are generated.
- $\sqrt{S_k}$, $U_k \cdot \sqrt{S_k^T}$ and $\sqrt{S_k} \cdot V_k^T$ are computed. These two matrix products represent m pseudo-users and n pseudo-items in the k dimensional feature space.
- Neighborhood formation is performed.

The core of the SVD algorithm lies in the following theorem:
It is always possible to decompose a given matrix A into $A = U\lambda V^T$. Given the $n \times m$ matrix data A (n items, m features), an $n \times r$ matrix U (n items, r concepts), an $r \times r$ diagonal matrix λ (strength of each concept), and an $m \times r$ matrix V (m features, r concepts) are obtained.
SVD methods are a direct consequence of a theorem in linear algebra. The theorem states that a matrix can be composed into three components: $U(m \times m)$, $S(m \times n)$ and $V(n \times n)$.

6.1. Definition VI.1 (Singular Value Decomposition)
Any $m \times n$ matrix A whose number of rows m is greater than or equal to its number of columns n, can be written as the product of a $m \times m$ column-orthogonal matrix U, and $m \times n$ diagonal matrix W with positive or zero elements (singular values), and the transpose of an $n \times n$ orthogonal matrix V.

	method$_1$	**method$_2$**	**method$_3$**	**method$_4$**
task$_1$	5	5	0	5
task$_2$	5	0	3	4
task$_3$	3	4	0	3
task$_4$	0	0	5	3
task$_5$	5	4	4	5
task$_6$	5	4	5	5

Table 2: Singular Value Decomposition m tasks and n methods

6.2. Dimensionality Reduction

Applying SVD to the matrix shown in Table II yields three different components: matrix $U(6\times 6)$, matrix $S(6\times 6)$ and matrix $V(4\times 4)$. This matrix is collapsed from a (6×4) space into a 2-Dimensional one by the first two columns of U, S and V.

$$U = \begin{pmatrix} -0.4472 & 0.5373 \\ -0.3586 & -0.2461 \\ -0.2925 & 0.4033 \\ -0.2078 & -0.6700 \\ -0.5099 & -0.0597 \\ -0.5316 & -0.1887 \end{pmatrix} \tag{14}$$

$$S = \begin{pmatrix} 17.7139 & 0.0000 \\ 0.0000 & 6.3917 \end{pmatrix} \tag{15}$$

$$V^T = \begin{pmatrix} -0.5710 & 0.2228 \\ -0.4275 & 0.5172 \\ -0.3846 & -0.8246 \\ -0.5859 & -0.0532 \end{pmatrix} \tag{16}$$

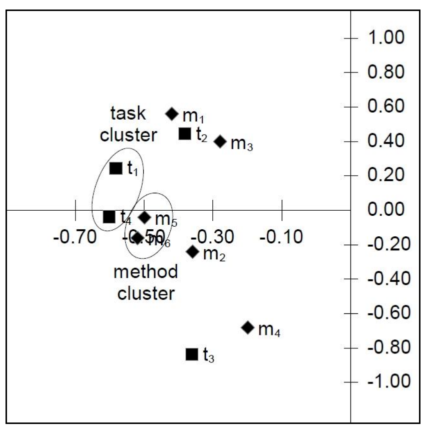

Figure 3: SVD - 2-Dimensional Projection

Next, $user_5$ joins the site and shared a few ratings $(5 \quad 5 \quad 0 \quad 0 \quad 0 \quad 5)$ for items 1-6. $user_5 = user_5^T \times U \times S^{-1}$ (17)

$$user_5 = \begin{pmatrix} 5 \\ 5 \\ 0 \\ 0 \\ 0 \\ 5 \end{pmatrix} \times \begin{pmatrix} -0.4472 & 0.5373 \\ -0.3586 & -0.2461 \\ -0.2925 & 0.4033 \\ -0.2078 & -0.6700 \\ -0.5099 & -0.0597 \\ -0.5316 & -0.1887 \end{pmatrix} \times \begin{pmatrix} 17.7139 & 0.0000 \\ 0.0000 & 6.3917 \end{pmatrix}^{-1} \tag{18}$$

$$user_5 = \begin{pmatrix} -0.3775 & 0.0802 \end{pmatrix} \tag{19}$$

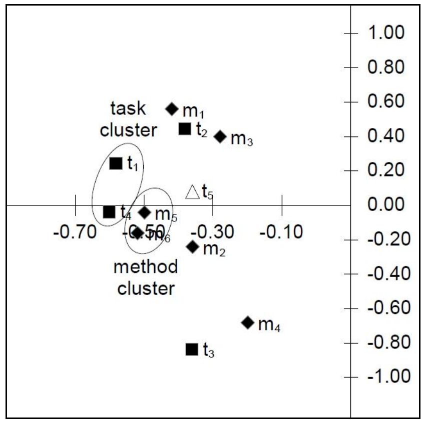

Figure 4: SVD - Finding similar IT Governance tasks I

Similarity is judged by the cosine similarity. All tasks are discarded whose similarity is less than 0.90 (outside the shaded region). Tasks t_1 and t_2 are selected. Task t_1 rated every method except m_4, and Task t_5 rated methods m_1, m_2 and m_6. Set of difference $\left([1,2,3,4,5,6]-[1,2,6]=[3,5]\right)$ is determined which are the methods Task t_1 rated but not used by the Task t_5. This set is returned as a recommendation in the decreasing order of Task t_1 ratings: Method m_5 (5 stars) and Method m_3 (3 stars).

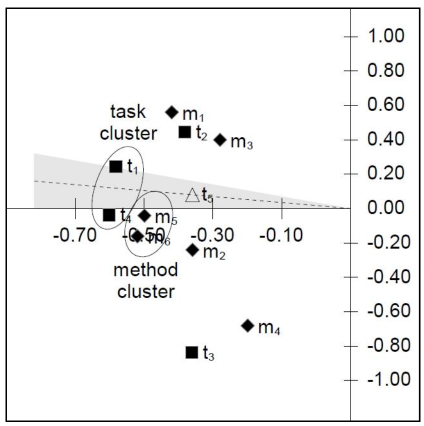

Figure 5: SVD - Finding similar IT Governance tasks II

7. Conclusion

Business Informatics is a fertile ground for research with the potential for immense and tangible impact. This research paper is connected with recommender systems that are used for management of economic efficiency of business informatics with the support of Business Intelligence and Data Mining. Current recommender approaches were analyzed and a flexible model was developed.

User based methods rely on the opinion of likeminded users to predict a rating, and generate recommendations. Item based approaches look at ratings given to similar items and generate recommendations. The computational complexity of user based methods scales up as the number of users increases whereas that of item based methods goes up as the number of items grows. This approach exploits the item based neighborhood. The choice of this method resides in the fact that in production environments users are added more often than items are.

8. Acknowledgment

This research paper is part of my habilitation project. I would like to take this opportunity to express my profound gratitude and deep regard to my both colleagues Prof. Dr. Vorisek and Dr. Pour, for their exemplary guidance, valuable feedback and constant encouragement throughout the duration of the project. Their valuable research on the MBI framework and suggestions were of immense help throughout this research work.

9. References

1. J. Vorisek and J. Pour, "Model for management of business informatics - Principles and practices," Journal of Systems Integration, vol. 4(1), pp. 3–12, 2014.
2. Vespignani, " Predicting the Behavior of Techno-Social Systems," Science, vol. 325, pp. 425-428, 2009.
3. D. J. Watts, "A twenty-first century science," Nature, vol. 445, p. 489, 2007.
4. J. Vorisek and J. Pour, "Management podnikove informatiky," Professional Publishing, Tech. Rep., 2012.

5. G. Adomavicius and A. Tuzhilin, "Towards the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions," IEEE Transactions on Knowledge and Data Engineering, vol. 17(6), pp. 734–749, 2005.
6. R. Burke, A. Felfernig, and M. Goeker, "Recommender systems: An overview," AI Magazine, vol. 32(3), pp. 13–18, 2011.
7. J. A. Konstan, "Introduction to Recommender Systems: Algorithms and Evaluation," ACM Transactions on Information Systems, vol. 22(1), pp. 1–4, 2004.
8. P. Resnick and H. R. Varian, "Recommender Systems," Communications of the ACM, vol. 40(3), pp. 56–58, 1997.
9. G. Adomavicius and Y. Kwon, "New recommendation techniques for multicriteria rating systems," IEEE Intelligent Systems, vol. 22(3), pp. 48–55, 2007.
10. R. Burke, "Hybrid recommender systems: survey and experiments," User Modeling and User-Adapted Interaction, vol. 12(4), pp. 331–370, 2002.
11. M. Pazzani and Billsus, "Adaptive web site agents," Journal of Agents and Multiagent systems, vol. 5(2), pp. 205–218, 2002.
12. J. S. Breese, D. Heckerman, and C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering," in Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, 1998.
13. D. Jannach, M. Zanker, A. Felfernig, and G. Friedrich, "Recommender Systems - An Introduction," Cambridge University Press, 2010.
14. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Itembased collaborative filtering recommendation algorithms," Proceedings of the 10th international conference on World Wide Web, pp. 285–295, 2001.
15. G. Golub and C. Reinsch, "Singular value decomposition and least squares solutions," Numerical Mathematics, vol. 14(5), pp. 403–420, 1970.
16. M. W. Berry, S. T. Dumais, and G. W. O'Brien, "Using linear algebra for intelligent information retrieval," SIAM Review, vol. 37, pp. 573–595, 1995.
17. S. Deerwester, S. T. Dumais, G. W. Furnas, and R. Harshman, "Indexing by latent semantic analysis," Journal of the American Society for Information Science, vol. 41, 1990