CONTENT-BOOSTED COLLABORATIVE FILTERING USING SEMANTIC SIMILARITY MEASURE

Ugur Ceylan
Department of Computer Engineering, Baskent University, Ankara, Turkey

Aysenur Birturk
Department of Computer Engineering, METU, Ankara, Turkey

Keywords: Content-boosted collaborative filtering, Semantic similarity, Ontology, Sparsity, Item cold-start.

Abstract: Collaborative filtering is one of the most used recommendation approaches in recommender systems. However, collaborative filtering systems have some major problems such as sparsity, scalability and cold-start problems. In this paper we focus on the sparsity and item cold-start problems in collaborative filtering in order to improve the quality of recommendations. We propose an approach that uses semantic similarities between items based on a priori defined ontology-based metadata in the movie domain. According to the semantic similarities between items and past user preferences, recommendations are made. The results of the evaluation phase show that our approach improves the quality of recommendations.

1 INTRODUCTION AND RELATED WORK

The aim of recommender systems is to predict the valuable information/items for a user and recommend these items. Some examples of items that are recommended by recommender systems are web pages, movies, music, books, restaurants, etc.

One of the most commonly used recommendation approaches in recommender systems is collaborative filtering. The idea behind collaborative filtering is that similar users almost have the same opinion about an item. Collaborative filtering systems try to find the similarities between active user and other users in the system, and then recommend items to the active user by taking into account these similarities. But, collaborative filtering systems suffer from some problems such as sparsity and item cold-start problems (Melville et al., 2002; Claypool et al., 1999).

Content-based filtering is another recommendation approach that is used widely. In content-based filtering, the system tries to recommend items which have similar contents with the items that are liked by the users. But also, content-based filtering systems have some major problems (Balabanovíc and Shoham, 1997). For some domains in which extracting content of items is difficult and content of items are insufficient to express items, content-based filtering is not a suitable recommendation approach. Another problem in content-based approach is that it tends to recommend items that are similar to those already highly rated. This problem is called over-specialization problem.

Some techniques are used in order to cope with sparsity and item cold-start problems. The simplest technique, which is used to overcome the sparsity problem, is called default voting (Breese et al., 1998). In this technique, a default rating is inserted for items which don't have rating values given by either one of them or the other. Thus, the number of overlapping rated items by both users is increased. The other technique for dealing with the sparsity problem is using dimensionality reduction techniques such as Singular Value Decomposition (SVD) (Sarwar et al., 2000). By applying SVD, user-item rating matrix may become less sparse.

Hybrid recommendation approach is generally implemented by combining collaborative and content-based filtering approaches to cope with the drawbacks of these two filtering methods (Balabanovíc and Shoham, 1997). Some hybrid approaches are as follows. One approach to use both content-based and collaborative filtering approaches is combining them. In this approach, system generates recommendations by using content-based and collaborative filtering approaches and then combines these

Ceylan U. and Birturk A..
CONTENT-BOOSTED COLLABORATIVE FILTERING USING SEMANTIC SIMILARITY MEASURE.
DOI: 10.5220/0003402403660371
In *Proceedings of the 7th International Conference on Web Information Systems and Technologies* (WEBIST-2011), pages 366-371
ISBN: 978-989-8425-51-5

independent recommendations (Pazzani, 1999). Another approach gives some weight values to collaborative and content-based predictions. Then hybrid approach takes the weighted sum of the predicted ratings. And finally items that are recommended are selected based on the calculated weighted sum (Claypool et al., 1999). Also content-based filtering is used to complete the missing data in the user-item rating matrix. Then collaborative filtering approach is used to recommend items to users. This hybrid approach is also called content-boosted collaborative filtering (Melville et al., 2002).

In this paper, we propose a hybrid approach based on content-boosted collaborative filtering presented in (Melville et al., 2002). The purpose of our approach is to cope with sparsity and item cold-start problems of collaborative filtering. Our approach utilizes both content-based and collaborative filtering. The crucial point in this study is that the content-based filtering in our approach uses semantic similarity measures on ontology-based metadata, based on the studies in (Maedche and Zacharias, 2002) and (Lula and Paliwoda-Pekosz, 2008), instead of naive Bayesian classifier (Mitchell, 1997) which is mentioned in (Melville et al., 2002). After that, collaborative filtering is performed using enhanced data to overcome over-specialization problem in content-based filtering.

2 PROPOSED APPROACH

The flow diagram of our approach, called *SEMCBCF*, is shown in the Figure 1. *SEMCBCF* consists of three phases:

1. Generating ontology-based metadata
2. Finding enhanced user-item rating matrix by using content-based filtering
3. Using collaborative filtering on enhanced user-item matrix

2.1 Generating Ontology-based Metadata

In order to find semantic similarities between items, ontology model and metadata model have to be defined. Ontology and metadata models are defined as follows (Maedche and Zacharias, 2002):

$$O := \{\tilde{C}, \tilde{P}, \tilde{A}, H^c, prop, att\} \quad (1)$$

$$MD := \{O, \tilde{I}, \tilde{L}, inst, instl, instr\} \quad (2)$$

For ontology model, $\tilde{C}$,$\tilde{P}$ and $\tilde{A}$ are sets which consist of concepts, relations and attributes' identifiers

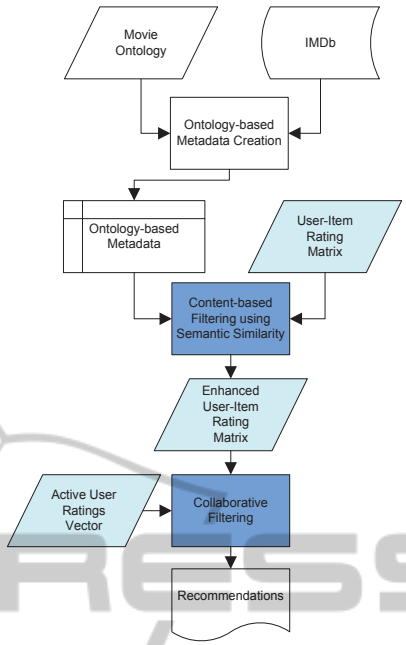

Figure 1: System Overview.

respectively. H^c is called concept taxonomy which defines the hierarchical relations between concepts. *prop* and *att* are functions that define non-taxonomical relations. For metadata model, $\tilde{I}$ and $\tilde{L}$ are sets which consist of instances and literal values respectively. *inst*, *instr*, *instl* are functions that define concept instantiation, relation instantiation and attribute instantiation. Predicates and their meanings that are used in the following sections are shown in the Table 1.

Table 1: Predicates and Meanings.

Predicate	Meaning
$H^c(C_1, C_2)$	C_1 is a subconcept of C_2
$P(C_1, C_2)$	P is a relation with domain C_1 and range C_2
$A(C_1)$	A is an attribute of C_1
$C(I)$	I is a instance of concept C
$P(I_1, I_2)$	Instance I_1 has a P relation to instance I_2
$A(I_1, L)$	Instance I_1 has an A attribute with value of L

To define movie ontology, we use a free, open source ontology editor and knowledge-base framework called Protege (*http://protege.stanford.edu*). By using Protege, we create a number of different movie ontologies manually. Then, the metadata are generated based on the defined movie ontology and the content of movies extracted from IMDb (The Internet Movie Database, $http://www.imdb.com$). Con-

tent of a movie is represented as 10 dimensions which consist of cast, director, writer, language, genre, runtime, release date, country, color and average rating given by IMDb users. A feature of a dimension can be an instance or a concept or a literal in ontology.

In order to evaluate our approach, four different ontologies are used. Ontology1 is the basic ontology used in our approach. All features of genre dimension are sub-concept of *Movie* concept. For each remaining dimensions, a concept exists in Ontology1. And the features of a dimension are instances of its corresponding concept. Ontology2 is similar to Ontology1. Only difference between them is that concepts represent dimensions except genre has more sub-concepts in Ontology2 than it has in Ontology1. For example the concept that represents runtime dimension has a number of sub-concepts that defines the runtime intervals. The only difference between Ontology1 and Ontology3 is that runtime, release date and average rating dimensions are represented as attributes in Ontology3. In Ontology4, the features of genre dimensions are grouped into six sets and a set represents a concept. A feature of genre dimension is a sub-concept of its corresponding concept.

2.2 SEMCBF:Content-based Filtering using Semantic Similarity

In order to recommend items, a similarity measure between items has to be defined first. And then, by using the active user model, which is a vector that consists of user's ratings, and similarities between items, we predict the ratings of unrated items which will be given by the active user. In *SEMCBF*, to calculate similarities between items which are described by ontology-based metadata, we use three types of similarity measures(Maedche and Zacharias, 2002): taxonomy similarity (TS), relation similarity (RS), and attribute similarity (AS).

Taxonomy similarity between two instances (TS) is based on their corresponding concepts' positions in concept taxonomy (H^c) which is defined in ontology model. Basically, the idea behind taxonomy similarity is that closer concepts in taxonomy are more similar.

An instance can be instance-of two different concepts in ontology. So, to find taxonomy similarities between instances (TS), first, taxonomy similarities between concepts (TSC) have to be defined.

In order to calculate TSC, four different methods, TSC_{CM}, $TSC_{Wu\&Palmer}$, TSC_{Lin} and TSC_{Mclean} in the studies (Maedche and Zacharias, 2002), (Wu and Palmer, 1994), (Lin, 1998), (Li et al., 2003) respectively, are used.

After finding the taxonomy similarity between concepts, calculating taxonomy similarity between instances is reduced to calculating the similarity of two sets. So TS is defined as follows:

$$TS(I_1,I_2) = SSIM(CSET(I_1),CSET(I_2)) \quad (3)$$

where $CSET(I) = \{C \in \tilde{C} | C(I)\}$.

Similarity between two sets can be found using the similarities between their elements, in this case TSC of concepts, and using different methods. These methods are mentioned later in this section.

The second type of similarity measure using ontology-based metadata is relation similarity. Relation similarity (RS) between two instances is based on their relations to other instances in ontology-based metadata. For relation similarity measure, we use a modified version of relation similarity measure in (Maedche and Zacharias, 2002). RS between I_1 and I_2 can be calculated as follows:

$$RS(I_1,I_2) = \frac{\sum_{p \in P_{co-I}} OR(I_1,I_2,p,IN)}{|P_{co-I}| + |P_{co-O}|} + \frac{\sum_{p \in P_{co-O}} OR(I_1,I_2,p,OUT)}{|P_{co-I}| + |P_{co-O}|} \quad (4)$$

P_{co-I} stands for 'incoming relations' and is the set of relations that allows $UC(C(I_1),H^c)$ and $UC(C(I_2),H^c)$ as range where $UC(C_i,H^c) = \{C_j \in \tilde{C} | H^c(C_i,C_j) \vee C_i = C_j\}$. P_{co-O} stands for 'outgoing relations' and is the set of relations that allows $UC(C(I_1),H^c)$ and $UC(C(I_2),H^c)$ as domain.

$OR(I_1,I_2,p,DIR)$ stands for the similarity for relation p and direction DIR between instances I_1 and I_2 where $DIR \in \{IN,OUT\}$. $OR(I_1,I_2,p,DIR)$ can be calculated by considering associated instances of I_1 and I_2 with respect to relation P and direction DIR. Associated instances (A_s) of instance I_i with respect to relation P and direction DIR is as follows:

$$A_s(P,I_i,DIR) = \begin{cases} \{I_j : I_j \in \tilde{I} \wedge (P(I_j,I_i)\}, & \text{if DIR = IN} \\ \{I_j : I_j \in \tilde{I} \wedge (P(I_i,I_j)\}, & \text{if DIR = OUT} \end{cases} \quad (5)$$

After defining A_s, calculating $OR(I_1,I_2,p,DIR)$ is reduced to calculating the similarity of two sets that contains associated instances. So OR is defined as follows:

$$OR(I_1,I_2,p,DIR) = SSIM(A_S(P,I_1,DIR),A_S(P,I_2,DIR)) \quad (6)$$

If $A_S(P,I_1,DIR) = \emptyset$ or $A_S(P,I_2,DIR)) = \emptyset$ then $OR(I_1,I_2,p,DIR)$ is 0. The point is that to calculate SSs of instances, RS is used and to calculate RSs of instances, SSs of associated instances are used. In order to avoid infinite cycles, a maximum depth of recursion has to be defined.

The advantage of using relation similarity is that the similarities of features are taken into account. Suppose that, in a system, movies have only one feature which is an actor played in movie and we try to find the similarity between two movies, MovieX and MovieY. MovieX has a feature ActorA and MovieY has a feature ActorB. If a user only rated movies in which only ActorA played, it is unable to predict the rating of MovieY by using naive Bayesian classifier. But in *SEMCBF*, similarity of MovieX and MovieY depends of the similarity between ActorA and ActorB. In a recursive manner, the similarity of ActorA and ActorB depends on similarity of other instances which have relations with ActorA and ActorB. Thus, we can calculate a similarity value between these two movies and a prediction can be made.

Attribute similarity is the third similarity measure that is used to calculate semantic similarities in ontology-based metadata. *AS* between instances I_1 and I_2 is as follows:

$$AS(I_1,I_2) = \frac{\sum_{a \in P_A} OA(I_1,I_2,a)}{|P_A|} \quad (7)$$

where P_A represents the set of attributes that are attributes of both $UC(C(I_1),H^c)$ and $UC(C(I_2),H^c)$. $OA(I_1,I_2,a)$ is the similarity for attribute a between instances I_1 and I_2. Like calculation of *OR*, $OA(I_1,I_2,a)$ is calculated by considering associated literals of I_1 and I_2 with respect to the attribute a. Associated literal (A_l) of I_i with respect to the attribute A is the following:

$$A_l(A,I_i) = \begin{cases} L_x, & \text{if } L_x \in \tilde{L} \wedge A(I_i,L_x) \\ \emptyset, & \text{otherwise} \end{cases} \quad (8)$$

The difference between A_s and A_l is that A_l can contain at most one literal unlike A_s. Thus, rather than calculating similarity of two sets, similarity between attribute values is focused in order to calculate *OA* which is as follows:

$$OA(I_1,I_2,a) = LSIM(L_1,L_2,a) \quad (9)$$

where $L_1 = A_l(a,I_1)$ and $L_2 = A_l(a,I_2)$. If $L_1 = \emptyset$ or $L_2 = \emptyset$ then $OA(I_1,I_2,a)$ is 0. In our approach, all of the attributes used are numeric features of the instances like release date (as a year), runtime etc. So, the similarity between two numeric values (L_1 and L_2) of an attribute (a) is as follows:

$$LSIM(L_1,L_2,a) = 1 - \frac{(L_1 - L_2)}{MDIF(a)} \quad (10)$$

where

$$MDIF(A) = max\{(L_i - L_j) : A(I_1,L_i) \wedge A(I_2,L_j) \wedge I_1,I_2 \in \tilde{I}\} \quad (11)$$

In order to calculate *TS* and *RS*, we have to define the similarity between sets of elements. Elements of these sets are, *concepts* for *TS* calculation and *instances* for *RS* calculation. Similarities of elements are *TSC*s for *TS* and *SS*s for *RS*. The first three methods $SSIM_1$, $SSIM_2$ and $SSIM_3$ used for calculating the similarity between sets are the methods in the studies (Maedche and Zacharias, 2002), (Tintarev and Masthoff, 2006), (Bach and Kuntz, 2005) respectively.

The other methods used for calculating the similarity between sets are based on the methods used for calculating the distance of pair of clusters in hierarchical clustering algorithms. These methods are single-link ($SSIM_S$), complete-link ($SSIM_C$) and average-link ($SSIM_A$) (Maimon, 2005).

Up to now, the taxonomy, relation and attribute similarities between instances are defined. Now, we can combine these measures by giving them some weight values. Semantic similarity (*SS*) between two instances is defined as follows:

$$SS(I_1,I_2) = \frac{aTS(I_1,I_2) + bRS(I_1,I_2) + cAS(I_1,I_2)}{a+b+c} \quad (12)$$

The last step of *SEMCBF* is prediction of the unknown ratings of the items given by users. In order to predict the unknown ratings, our approach uses the calculated similarities between items and user model which consists of ratings given by the user on a neighborhood-based method (Herlocker et al., 1999)(Sarwar et al., 2001). To compute a prediction, two prediction functions can be used after selecting the k most similar items. First function calculates the predicted rating of user u on item i by taking the average of ratings given by the user u on k most similar items to i. Second function calculates the predicted rating of user u on item i by taking the weighted average of the ratings given by the user u on k most similar items to i. Weights of the ratings are set according to the similarities between items.

Using user-item rating matrix, *SEMCBF* creates enhanced user-item rating matrix. In other words, the sparsity of user-item rating matrix is reduced. And also, even if an item has no explicit rating given by any user in the system, by using *SEMCBF*, our approach predicts a rating given by every user for that item.

2.3 Collaborative Filtering

In the third phase of our approach, a neighborhood-based (Herlocker et al., 1999) collaborative filtering algorithm is used on enhanced user-item matrix and active user ratings vector which consists of only actual ratings given by the active user. The algorithm

first computes the similarity between the active user and other users in enhanced user-item matrix using Pearson correlation. After computing similarities between active user and other users, *n* number of most similar users is selected. An unknown rating is predicted by calculating the adjusted weighted sum of the nearest neighbors' ratings of active users.

At the end of a recommendation process, the system recommends a number of unrated items which have the highest predicted rating to the active user.

3 EXPERIMENTAL EVALUATION

The performance of proposed approach was evaluated in the movie domain by using the MovieLens 100k dataset (*http://www.grouplens.org*) which is publicly available. We apply 5-fold cross-validation on the disjoint test sets (20% of rating data) and their corresponding training sets (80% of rating data) that are also provided in MovieLens 100k dataset. In experimental evaluation, we use mean absolute error (MAE), precision, recall and F-measure performance metrics which are commonly used to evaluate the performance of recommender systems (Herlocker et al., 1999). The evaluation of our approach consists of two phases. In the first phase of the evaluation we try to find the most appropriate values for *SEMCBF* parameters. In the second phase, the results of *SEMCBF* and *SEMCBCF* is compared with some other approaches.

SEMCBF consists of some parameters as mentioned in section 2. These parameters and their possible values are shown in Table 2. It is obvious that the performance of *SEMCBF* depends on the values of these parameters. To find the most appropriate value of a parameter, performance of *SEMCBF* is evaluated using all possible combinations of this parameter, *O* and *k* while the values of other parameters remain constant. In each evaluation, the determined value as the most appropriate value for a parameter is used in later evaluations.

At the beginning of the evaluation, parameters *rd*, *a*, *b*, *c*, TSC, $SSIM_{TS}$, $SSIM_{RS}$ are set to 1, 0.4, 0.3, 0.3, TSC_{CM}, $SSIM_{SSIM_1}$, $SSIM_{SSIM_1}$ respectively. The parameters are analyzed in the following order; *PF*, TSC, $SSIM_{TS}$, $SSIM_{RS}$, *a*, *b*, *c*, *rd*. *SEMCBCF* using the values Ontology4 for *O*, 10 for *k*, *pred*2 for *PF*, TSC_{Lin} for TSC, $SSIM_S$ for $SSIM_{TS}$ and $SSIM_{RS}$, 0.1 for *a*, 0.1 for *b*, 0.8 for *c*, 2 for *rd* gives the best result.

SEMCBF is used for enhancing user-item matrix in *SEMCBCF*. So the performance of *SEMCBCF* is dependent to the performance of *SEMCBF*. Because of that, in this evaluation phase, both the performance of *SEMCBF* and *SEMCBCF* are compared with some other approaches. Table 3 gives precision, recall and F-measure results of *SEMCBF*, *SEMCBCF* and some approaches obtained from (Karaman, 2010). And also *CBCF* (Melville et al., 2002) is implemented using the same dataset to make a fair comparison. It can be seen from Table 3, *SEMCBF* and *SEMCBCF* outperform the other approaches in recommending high-quality items.

Table 2: Parameters and Possible Values of *SEMCBF*.

Parameter	Values
Ontology (*O*)	Ontology1 Ontology2 Ontology3 Ontology4
Max. Recursive Depth (*rd*)	0,1,2,3,4, 5,6,7,8
Weight of TS (*a*)	$(a+b+c)=1$
Weight of *RS* (*b*)	
Weight of *AS* (*c*)	
Measure for Taxonomy Similarity Between Concepts (TSC)	TSC_{CM} $TSC_{Wu\&Palmer}$ TSC_{Lin} TSC_{Mclean}
SSIM Method for TS ($SSIM_{TS}$)	$SSIM_1$, $SSIM_2$, $SSIM_3$, $SSIM_S$
SSIM Method for *RS* ($SSIM_{RS}$)	$SSIM_C$, $SSIM_A$
Number of Nearest Neighbors (*k*)	5,10,15,20, 30,50,100,200
Prediction Function (*PF*)	*pred*1, *pred*2

Table 3: Comparison of *SEMCBF* with Other Approaches.

Approach	Prec. (%)	Rec. (%)	F-Measure (%)
MovieLens	66	74	69,8
MovieMagician Feature-Based	61	75	67,3
MovieMagician Clique-Based	74	73	73,5
MovieMagician Hybrid	73	56	63,4
OPENMORE	75,2	73,7	74,4
ReMovender	72	78	74,9
CBCF	60	95,2	73,6
SEMCBF	63,4	92,3	75,2
SEMCBCF	63,7	93,1	75,6

4 CONCLUSIONS

This paper presents a hybrid approach, which uses both content-based and collaborative filtering, in order to overcome the sparsity and item cold-start prob-

lems of collaborative filtering. The presented approach is based on content-boosted collaborative filtering presented in (Melville et al., 2002). Our hybrid approach (*SEMCBCF*) first uses content-based filtering (*SEMCBF*) to enhance the user-item similarity matrix, then performs collaborative filtering using this enhanced user-item matrix. The contribution of our approach is that it uses semantic similarity measures on ontology-based metadata to calculate the similarities of items in content-based filtering. Our hypothesis was that using semantic similarity measures rather than naive Bayesian classifier (Mitchell, 1997) which is used in (Melville et al., 2002) will improve the quality of recommendations.

In the evaluation phase, first, *SEMCBF* was fine-tuned by determining the values of its parameters. Then, using the determined values, *SEMCBF* and *SEMCBCF* was evaluated. The results showed that *SEMCBF* and *SEMCBCF* outperforms content-boosted collaborative filtering presented in (Melville et al., 2002) and some other approaches.

The characteristics of the ontology, such as the taxonomy of concepts and representation of features significantly effect performance of *SEMCBF*. For further research, ontology refinement will be focused to improve *SEMCBF*. And also, *SEMCBF* will be improved by assigning some weight to relations and attributes in the ontology.

REFERENCES

Bach, T. L. and Kuntz, R. D. (2005). Measuring similarity of elements in owl dl ontologies. In *Proceedings of the AAAI'05 Workshop on Contexts and Ontologies: Theory, Practice and Applications*, pages 96–99.

Balabanovíc, M. and Shoham, Y. (1997). Fab: Content-based, collaborative recommendation. *Communications of the ACM*, 40(3):66–72.

Breese, J. S., Heckerman, D., and Kadie, C. (1998). Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence (UAI-98)*, pages 43–52, San Francisco. Morgan Kaufmann.

Claypool, M., Gokhale, A., Miranda, T., Murnikov, P., Netes, D., and Sartin, M. (1999). Combining content-based and collaborative filters in an online newspaper. In *Proceedings of ACM SIGIR Workshop on Recommender Systems*.

Herlocker, J. L., Konstan, J. A., Borchers, A., and Riedl, J. (1999). An algorithmic framework for performing collaborative filtering. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 230–237, New York, NY, USA. ACM Press.

Karaman, H. (2010). Content based movie recommendation system empowered by collaborative missing data prediction. M.Sc. Thesis in Computer Engineering Department of Middle East Technical University.

Li, Y., Bandar, Z. A., and McLean, D. (2003). An approach for measuring semantic similarity between words using multiple information sources. *IEEE Transactions on Knowledge and Data Engineering*, 15:871–882.

Lin, D. (1998). An information-theoretic definition of similarity. In *ICML '98: Proceedings of the Fifteenth International Conference on Machine Learning*, pages 296–304, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Lula, P. and Paliwoda-Pekosz, G. (2008). An ontology-based cluster analysis framework. In *7th International Semantic Web Conference (ISWC2008)*.

Maedche, A. and Zacharias, V. (2002). Clustering ontology-based metadata in the semantic web. In Elomaa, T., Mannila, H., and Toivonen, H., editors, *Proceedings of the 6th European Conference on Principles of Data Mining and Knowledge Discovery (PKDD 2002), August 19-23, 2002, Helsinki, Finland*, volume 2431 of *Lecture Notes in Computer Science*, pages 383–408. Springer, Berlin–Heidelberg, Germany.

Maimon, O. (2005). *Decomposition Methodology For Knowledge Discovery And Data Mining: Theory And Applications (Machine Perception and Artificial Intelligence)*. World Scientific Publishing Co., Inc., River Edge, NJ, USA.

Melville, P., Mooney, R. J., and Nagarajan, R. (2002). Content-boosted collaborative filtering for improved recommendations. In *Proceedings of the Eighteenth National Conference on Artificial Intelligence*, pages 187–192.

Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill International Edit.

Pazzani, M. J. (1999). A framework for collaborative, content-based and demographic filtering. *Artificial Intelligence Review*, 13:393–408.

Sarwar, B., Karypis, G., Konstan, J., and Reidl, J. (2001). Item-based collaborative filtering recommendation algorithms. In *WWW '01: Proceedings of the 10th international conference on World Wide Web*, pages 285–295, New York, NY, USA. ACM.

Sarwar, B. M., Karypis, G., Konstan, J. A., and Riedl, J. T. (2000). Application of dimensionality reduction in recommender systems: A case study. In *WebKDD Workshop at the ACM SIGKKD*.

Tintarev, N. and Masthoff, J. (2006). Similarity for news recommender systems. In *Proceedings of the AH06 Workshop on Recommender Systems and Intelligent User Interfaces*.

Wu, Z. and Palmer, M. (1994). Verb semantics and lexical selection. In *Proc. of the 32nd annual meeting on Association for Computational Linguistics*, pages 133–138.