Inferring User's Preferences using Ontologies

Vincent Schickel-Zuber and Boi Faltings
Ecole Polytechnique Fédérale de Lausanne (EPFL)
Artificial Intelligence Laboratory
CH-1015 Lausanne, Switzerland
{vincent.schickel-zuber, boi.faltings}@epfl.ch

Abstract

We consider recommender systems that filter information and only show the most preferred items. Good recommendations can be provided only when an accurate model of the user's preferences is available. We propose a novel technique for filling in missing elements of a user's preference model using the knowledge captured in an ontology. Furthermore, we show through experiments on the MovieLens data set that our model achieves a high prediction accuracy and personalization level when little about the user's preferences is known.

Introduction

Recommendation systems (RS) have been devised as tools to help people find items on the internet. Two kinds of techniques are widely used in e-commerce sites today.

The first technique is item-to-item collaborative filtering (CF, (Sarwar *et al.* 2001)), which recommends products to users based on the experience of like-minded groups of users. CF assumes that similar users like similar objects, which means that its ability to recommend items depends on the capability to successfully identify the set of similar users, known as the target user's neighbourhood. Furthermore, it does not build an explicit model of the user's preferences. Instead, preferences remain implicit in the ratings that the user gives to some subset of products. In practice, CF is the most popular recommendation technique, and this is due to three main reasons. First, studies have shown it to have satisfactory performance when sufficient data is available. Second, it can compare items without modeling them and thus can theoretically deal with any kind of item, as long as they have been rated by other people. Finally, the cognitive requirement on the user is very low. However, it has been argued by many authors that CF suffers from profound handicaps such as the *cold-start*, *first-rater,* and *scalability* problems (Li *et al.* 2005), (Mobasher, Jin, & Zhou 2004), and (Sullivan *et al.* 2004).

The other widely used technique is preference-based recommendation. Here, a user is asked to express explicit preferences for certain attributes of the product. If preferences are accurately stated, multi-attribute decision theory (MAUT, (Keeney & Raiffa 1993)) provides methods to find the preferred product even when the set of alternatives is extremely large and/or volatile. This technique does not suffer from cold start, latency or scalability problems, since recommendations are based only on the individual user's data. However, the big drawback of preference-based methods is that the user needs to express a potentially quite complex preference model. This may require a large number of interactions, and places a higher cognitive load on the user since he has to reason about the attributes that model the product.

At the same time, the use and benefit of ontologies in recommendation systems has been widely accepted. (Bradley, Rafter, & Smyth 2000) have used a simple ontology called Concept Tree to build a personalized search engine that increased classification accuracy by more than 60%. (Mobasher, Jin, & Zhou 2004) have reduced data sparsity in CF by combining semantic and item similarities together. (Middelton, Shadbolt, & De Roure 2004) have used ontological relationships between topics of interest to infer other topics of interest, which might not have been browsed explicitly. More recently, it has been shown that topic diversity in recommendation via the use of an ontology can increase recommendation usefulness (Ziegler *et al.* 2005).

In this paper, we define a novel similarity measure called *Ontology Structure based Similarity* (OSS). It is based on assigning concepts in the ontology an *a-priori score* (APS), and computing the relations between the scores assigned to different concepts. These similarities are then used to propagate scores for a specific user. We use this in a novel preference based technique that solves the recommendation problem even when very little data about the user is known. As in collaborative filtering, user's preferences are expressed implicitly via the ratings of some items. The novelty of our work is to infer missing preferences using the OSS approach, thus avoiding the need for complex preference elicitation.

Definitions & Assumptions

In this work, an ontology λ is defined as a directed acyclic graph (DAG) where a node represents a primitive concept, and an edge models the binary specialization relation (isa) between two concepts. Thus, the ontology establishes a hierarchy where each concept can have a set of sub-concepts known as the *descendants*, but not all instances of a concept must belong to a sub-concept.

Copyright © 2006, American Association for Artificial Intelligence (www.aaai.org). All rights reserved.

An example of a popular ontology is WordNet (Miller *et al.* 1993), where concepts represent groups of similar words (synonyms), and edges are hypernyms (is-subset-of) and hyponyms (part-of) relations. E-commerce sites like Amazon.com also use simple taxonomies to classify their items.

This work assumes the existence of an ontology, where all the items of our catalog are instances of a concept. Our model allows an item to be instance of any concept in the ontology, not just a leaf concept.

In the recommender system context, a concept represents a group of items with the same *features*. Consequently, items in the different sub-concepts are distinguished by differences in certain features. However, these are usually not made explicit in the ontology. Concretely, we see a feature as a restriction on a property or a combination of properties that differentiates a concept from its parent. For example, the subclasses of red and white wines are distinguished by a combination of features which include color and also certain aspects of taste.

The recommendation problem can be seen as the problem of predicting a *score* S assigned to an item. For example, the score could be a preference score or popularity rating. We assume that the score is a real-valued function that satisfies the following assumptions:

- A1: the score depends on features of the item. Thus, all instances of the same concept will have the same score.
- A2: each feature contributes independently to the score, i.e. the score can be modeled as the sum of the scores assigned to each feature. Each feature makes a non-negative contribution to the score.
- A3: features that are unknown make no contribution to the score. Thus, the score attached to a concept can be seen as a lower bound of the score that items belonging to that concept might have.

The third assumption may appear counterintuitive, but it is important because it eliminates any dependence of the score on the probability distribution of items within the concepts. If the score models the price that a user is willing to pay for an item, it is rational for users to adopt this pessimistic view, since one would not normally be willing to pay for features that have not been explicitly provided.

More generally, some analogy can be made between the score function and the lower prevision (Walley 1996). The lower prevision of a gamble X is a real number, which is interpreted as the highest price a user is willing to pay for X. In fact, the score of a concept c corresponds to a strict lower bound of the prevision for selecting any instance of c. Preferences can be reasonably modeled as scores, but scores could also model other properties.

We show in experiments that such an assumption about scores yields the best known model of concept similarity.

Computing an A-Priori Score

A major ingredient of OSS is to compute the *a-priori score* of a concept c, $APS(c)$, based on its location in the ontology. The APS models the expected score of each concept for an average user, but without using any user information. It is used to estimate constants that determine how actual user scores propagate through the ontology.

As we have no information about the user, we assume that all the items have an a-priori score that is uniformly distributed between 0 and 1. The score represents how much an item is liked, where 0 means maximally disliking the concept and, conversely, 1 means maximally liking it. Following assumption A3, the score of a concept is the greatest lower bound of the scores of its items. The probability that the score of a concept c is superior to the threshold x, ($S(c) > x$), is equal to $1 - x$. However, this probability ignores the fact that concepts can have descendants. Our model is by definition pessimistic (A3), which means that the score of a concept is a computed as a lower bound of its score and the score of its descendants. Therefore, the probability that the score of any concept c is superior to a threshold x is equal to $(1-x)^{n+1}$, where n is the number of descendants of c. Note that we count all descendants, not just the leaves, to account for the fact that each concept has instances that do not belong to any sub-concept.

The probability distribution of the score for a concept c is $P(S(c) \leq x) = 1 - (1-x)^{n+1}$, with the following density function:

$$f_c(x) = \frac{d}{dx}\left(1 - (1-x)^{n+1}\right) = (n+1) \,.\, (1-x)^n \quad (1)$$

To compute the expected score of the concept c, $E(c)$, equation (1) is integrated as shown in equation 2.

$$E(c) = \int_0^1 x f_c(x) dx = (n+1) \int_0^1 x(1-x)^n dx = \frac{1}{n+2} \quad (2)$$

The expected score tells us that the expected score of a concept c will be inversely proportional to the number of its descendants + 2. Following equation (2), the a-priori score of a concept c with n_c descendants is defined as:

$$APS(c) = \frac{1}{n_c + 2} \quad (3)$$

The a-priori score defined in equation (3) implies that the leaves of the ontology will have an APS equal to $1/2$, which is equal to the mean of a uniform distribution between 0 and 1. Conversely, the lowest values will be found on the root. This means that when we travel up the ontology, the concept becomes more generalized, and therefore the APS decreases. From an economic point of view, it means that a user is willing to pay less for a general concepts as there is more chance that it subsumes an items that the user dislikes. Another important aspect of this APS is the fact that the difference in score between concepts decreases when we travel up the ontology, due to the increasing number of descendants.

(Resnik 1998) also uses the topology to compute the *information content* of a concept. He extended the definition of the entropy and defined the information carried by a concept c as $-log(P(c))$, where $P(c)$ is the probability that the concept c or one of its descendants occur. The APS share some similarities with the information content approach. First, the difference in both the score and information content decreases when we travel up the ontology. Second, Resnik also uses the number of descendants to compute the probability of occurring of a concept. However, some profound differences exist. The APS is a bottom-up approach that considers the differences between the concepts, while Resnik's is a top-down approach and considers the similarities. Second, we use the $1/x$ function to compute our score, while Resnik uses the logarithm to base 2. In the validation section, we show that the APS brings better results that the information content approach.

Propagating Score in an Ontology

The a-priori score represents an estimator without considering a particular user. When a user's scores for certain concepts are known more precisely, we can derive a personalized score for the other concepts by propagation.

For example, imagine a situation with two concepts x and y, but where only $S(x)$ is known. To propagate the score from concept x to y, a link between these two concepts must be found. There are three cases to consider: when y is a parent of x ($x \subset y$, Fig. 1a), x is a parent of y ($y \subset x$, Fig. 1b), or when x is neither a parent nor a child of y ($x \not\subset y \wedge y \not\subset x$, Fig. 1c).

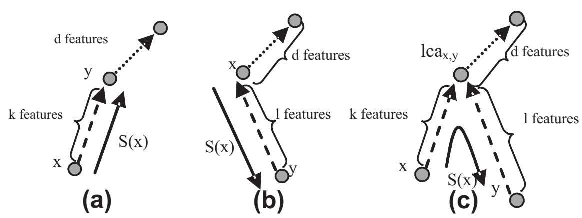

Figure 1: Possible chains between the concepts x and y: (a) x is a child of y, (b) x is a parent of y, and x is neither a parent nor a child of y (c)

Thus, the first task in the propagation is to identify the chain $C(x, y)$ that contains both concepts. To minimize the amount of propagation, we construct the chain through a lowest common ancestor, LCA. In a tree graph, a lowest common ancestor is defined as the closest upward reachable node shared by x and y (Knappe, Bulskov, & Andreasen 2003).

Note that in a DAG, there can be several LCA nodes; in fact, the number of LCA nodes can grow exponentially with the size of the graph. Fortunately, this number tends to be small in reality, as most concepts have only a few parents. For example, a concept in the WordNet ontology has on average 1.03 parents.

We use the following heuristic method to select which LCA to use to propagate the score. For each LCA node n, we compute the following values:

- its depth d(n), given as the distance of the longest path between the root and n, and
- its reinforcement r(n), given as the number of different paths leading to n from x and y.

We pick the LCA as the node with the highest value of $r(n) * 2^{d(n)}$. This heuristic is based on the idea that while we generally should limit the amount of propagation, if a node appears in many different connections between concepts x and y it can become more meaningful as a connection.

Upward Inference

This situation arises when there is a path going from concept x to its k^{th} parent y ($x \subset_k y$). From the tree construction, both concepts have d features in common but the concept x has an extra k features that differentiate it from its ancestor. By definition of the model, we know that the score of a concept depends on the features defining that concept (A1). Informally, it means that the score of y can be estimated knowing the score of x, $S(y|x)$, by looking at the ratio of features they have in common. Formally, $S(y|x)$ is defined as follows.

$$S(y|x) = \alpha S(x) \tag{4}$$

where α is the coefficient of generalization that contains the ratio of features in common which are liked according to their respective distribution. Obviously, α is unknown in our case. We estimate α by using the a-priori score captured by the concepts in the ontology. Thus, the coefficient of generalization can be estimated as the ratio of a-priori scores:

$$\alpha = \frac{APS(y)}{APS(x)} \tag{5}$$

Downward Inference

Inversely, we have the case when y is the l^{th}descendant of x ($y \subset_l x$). From the previous result, it is very tempting to assume that $S(y|x) = \beta S(x)$, where β is a coefficient of specialization that contains the ratio of features in common. However, this reasoning is not compatible with our second assumption – features contribute to the score independently. To understand this assumption, imagine that the score of the object is equal to the maximum price a user is willing to pay. Consider two concepts x and y, where y has one more feature than x. Now consider two users A and B such that A values x more than B does. This does not automatically mean that A will also attaches a higher value to the extra feature that distinguishes y from x. Notice also that when we were traveling upwards, we were considering super concept, which means we were removing known features whose contribution to the score is likely to be proportional to 1. However, when traveling downwards, we are adding new (unknown) features to the concept. Therefore, we need to consider the score of each new feature independently. Formally, it means that $S(y|x)$ must be defined as follows.

$$S(y|x) = S(x) + \beta \tag{6}$$

where β is the coefficient of specialization that contains the score of the features contained in concept y but not in x. Again, β can be estimated using the a-priori score:

$$\beta = APS(y) - APS(x) \tag{7}$$

Upward & Downward Inference

Finally, we consider the case when there is no direct path between concepts $x$and y. Figure 1.c reveals that in order to transfer the preference, we need to carry it up to the lowest common ancestor $LCA_{x,y}$, and then down to the concept y. Furthermore, and because the chain between concept x and y is not a path, we assume independence between x and y (the same reasoning is done on Bayesian Networks if no hard information is known about $LCA_{x,y}$). Thus, and using the result contained in equations (4) and (6), the score can be decomposed as follows.

$$S(y|x) = \alpha S(x) + \beta \tag{8}$$

Validation of the Model

To validate the approach, we used it to derive a similarity metric for the WordNet ontology.

There exist two main approaches for estimating similarity between concepts in a hierarchical ontology: the edge based approach and the node based approach. The edge based approach is the traditional, most intuitive, and simplest similarity measure. It computes the distance between two concepts based on the number of edges found on the path between them. One of the biggest problems of the edge based approach is that it considers the distance uniform on all edges, which is rarely the case in reality. (Resnik 1998) proposed a new approach based on the information content of a concept. This node-based approach measures the similarity based on the amount of information shared. More recently, (Jiang & Conrath 1998) proposed a hybrid approach that inherits the edge based approach of the edge counting scheme, which is then enhanced by the information count calculation.

We propose a novel approach to compute the similarity between concepts that is based on the following idea: the more features are propagated from one concept to another, the more similar these concepts will be. Following this, we define the *propagation of score* from a concept x to y, $\theta(x, y)$, as the amount of score being propagated from x to y. $\theta(x, y)$ is computed as follows. First, the score of a concept is transformed in such a way that $\theta(x, y) = 1$ iff $x = y$; this is achieved by setting the score $S(x)$ to 1. Second, we make sure that $\theta(x, y)$ is monotonically decreasing as the distance (in term of features) between concepts x and y increases. As a result, $\theta(x, y)$ is equal to α when traveling upwards, and inversely $1/(1 + \beta)$ when traveling downwards.

However, a distance function between two concepts x and y, $D(x, y)$, should be monotonically increasing when the number of edges separating x and y increases, and equal to 0 iff $x = y$ (Ming *et al.* 2004). There exist many function that satisfies the properties stated above, but our experiments have shown that it is $-log(\theta(x, y))$ that yields the best results. Using the same reasoning as previously, we define the distance between two concepts x and y as follows.

$$D(x, y) = -\log(\alpha) + \log(1 + \beta) \tag{9}$$

When Resnik introduced the node-based approach, he also established an evaluation procedure that has become widely used ever since. He evaluated his similarity metric by computing the similarity of word pairs using the WordNet ontology, and then looked at how well it correlated with human ratings of the same pairs. These word pairs were selected in such a way that they covered high, intermediate, and low levels of similarity.

WordNet is the most widely used and one of the biggest ontologies in the world ($\sim$80000 concepts), which makes experiments credible. Thus, we reproduced Resnik's experiment with the WordNet ontology version 2.0 on the original 30 word pairs. The correlation between various metrics and the human ratings are displayed in table 1.

	EdgeBased	Resnik	Jiang	OSS
Correlation	0.603	0.793	0.859	0.908

Table 1: Correlation with various similarity metrics

Our approach using the a-priori score achieves over 90% correlation with real user ratings, and clearly demonstrates significant benefit over earlier approaches (t-obs = 2.54 and p-value < 0.03). These results validate the inferring model and the a-priori score which were used to build our similarity metric.

As expected, the hybrid approach performed better than both existing techniques, but the improvement over the information based approach was not statistically significant (t-obs = 1.46 and p-value $\simeq$ 0.08). The edge based approach is the worst performing metric as it supposes that links in the ontology represent uniform distances, which is obviously not true in WordNet.

Finally, we tried different combinations of the coefficients α and β in order to test the upward and downward propagation. The experiment has shown that the best correlation is obtained when using α going up and β going down.

Application to Recommendation Systems

The *recommendation problem* is the problem of finding the items that best match the user's preferences. In this scenario, the score S can be seen as the user's preference value.

As in collaborative filtering, users express their preferences by rating a given number of items. These ratings are then used as a user's preference value on the representative concepts. Then, using our model, we infer the missing user's preference value of each concept. Finally, to recommend the best N items to the user (also known as the top-N strategy), we simply select N items from the concepts that have the highest preference value.

The standard metric for measuring the predictive accuracy of a recommendation is the mean absolute error (MAE, (Sarwar *et al.* 2001)), which computes the mean deviation between the predictions and the user's true ratings. Over the years, it has been argued that this metric may be less appropriate for the top-N task, as the granularity in the rating is usually small. However, the data is very sparse in our situation. Thus, the deviation in the ratings becomes very significant, which makes this metric relevant. Furthermore, (Herlocker *et al.* 2004) as argued that the MAE as many advantages such as the simplicity of understanding the results, and well studied statistical properties when comparing two approaches.

We also acknowledge the fact that the accuracy of a prediction is usually not enough to build a good recommendation system. For example, it is less interesting to recommend very popular items that everybody likes, and such recommendations bring nearly no information to the user. Thus, a new dimension for analyzing predictions that considers non-obvious predictions is required.

Novelty is a metric that measures the degree to which a recommendation is non-obvious. We will use the novelty metric defined by equation (10), which measures the number of correct recommendations made by algorithm a that are not present in the recommendations made by a reference algorithm b.

$$Novelty(r_a|r_b) = (|cr_a| - |cr_a \cap cr_b|)/N \qquad (10)$$

where r_a are the top-N recommendations made by the algorithm a, and cr_a are the correct recommendations contained in r_a, i.e. liked by the user.

To test our approach, we implemented a movie recommendation system using the famous MovieLens[1]; a data set containing the ratings of 943 real users on at least 20 movies. There are 1682 movies in total described by 19 themes: drama, action, and so forth. To increase the description of the movies, we wrote a wrapper that extracted the year, MPPA rating, and duration from the IMDb[2] website. As there is no common ontology modeling the movie domain, we created one using common sense and definitions found in dictionaries.

The experiment was as follows. First, users with less than 65 ratings were removed. For each remaining user, 15 ratings were inserted into a test set, TS, while the rest were inserted into an intermediate set, IS. Then, we transferred a given number of ratings from the IS into the learning set, LS, and built the preference model as follows. First, based on the rated item in LS, we set the user's preference value on the concepts that the items in LS are instances of. Then, we estimated the missing values using our propagation model. Finally, we predicted the grade of each movie in the TS, and the selected the Top-5. The experiment was run 5 times, and our technique (Heterogeneous Attribute Preference Propagation Model - HAPPL) was benchmarked against the following:

- *Popularity* is a simple but very effective strategy that ranks the movies based on their popularity. The popularity of each movie was computed using the users that were removed in the first phase of our experiment.
- *Hybrid* combines the Popularity and HAPPL approaches based on the averaged predicted ratings of each approach.
- *CF* is the adjusted cosine collaborative filtering. We set the neighbors to 90 as (Mobasher, Jin, & Zhou 2004) and (Sarwar *et al.* 2001) have shown that the optimum for MovieLens is very close to this value. CF was chosen as benchmark over classical content filtering as it is today's best performing filtering and most widely used RS. Furthermore, content approach requires a lot of information to compute an accurate model of the user, which is not available in our situation.

First, we measured the predictive accuracy of each method using various size of the learning set LS. Figure 2 clearly shows the weakness of CF when only a few ratings are used to learn the model. This is known as the cold-start problem, where a minimum number of ratings need to be known in order to find the right neighborhood of similar users, i.e. at least 20 in this case. However, our approach does not suffer from this problem and shows significant improvement over CF (p-value<0.01), when we have less than 30 ratings in the learning set. Surprisingly, the popularity metric performs well, even better than CF when the number of ratings in LS < 50, which shows that users tend to like popular items. As expected, the best accuracy is obtained using when we combine our approach with the popularity one. The combination of the grade allows the system to better discriminate between good and bad items with a higher confidence.

[1] http://www.cs.umn.edu/Research/GroupLens/data/

[2] http://www.imdb.com

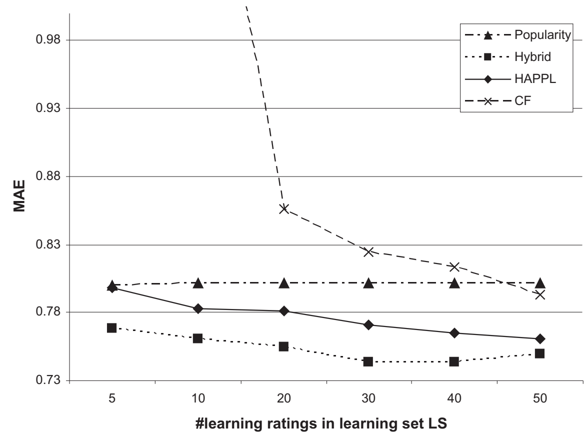

Figure 2: Prediction Accuracy of various techniques

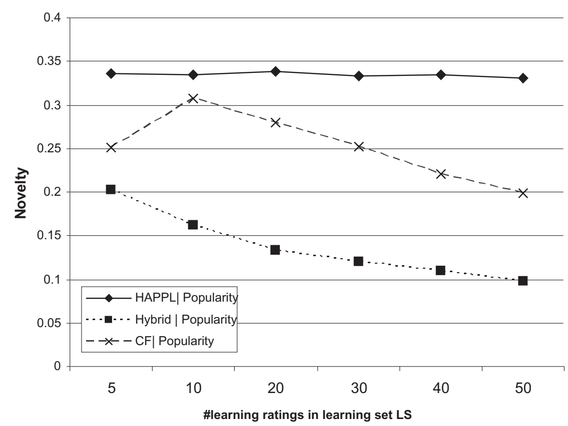

Figure 3: Novelty metric of various techniques against the Popularity approach

Second, we tested the novelty of each approach compared to the Popularity one. Again, the results (Figure 3) are very interesting in two points. First, it shows that it is our model that produces the best non-obvious recommendations whatever the size of the learning set, which has novelty value greater than 33%. Second, CF's novelty seems to improve when we have less than 10 ratings, and then decreases steadily down to the 20% threshold. This behavior can be explained if we superpose this result with the MAE. When we have less than 20 ratings, CF's accuracy is very low, which tends to indicate that items were selected from many diverse neighborhoods.

Finally, the Hybrid approach tells us that the use of collaborative data can improve the overall recommendation accuracy over our HAPPL approach, but this gain is then lost in recommendation novelty.

Conclusion

This paper makes two major contributions. First, we introduced a new technique called Ontology Structure Similarity to derive a similarity metric based on the structure of an ontology. The similarity metric exploits the implicit knowledge of the person who wrote the ontology and gave it a certain structure. Experimental evaluation has shown that it outperforms existing technique. Then, we used OSS to define the novel recommendation system HAAPL that is able to predict items with a very high accuracy and novelty, whatever the size of the learning set.

The results have unquestionably shown that the score is a suitable to model user's preference and concept similarity, and that it can be propagated as long as the domain is modeled by ontologies. It has been most surprising that a technique based only on the structure of an ontology and an individual user's own preferences can produce more novel recommendation than CF, which makes available other user's data.

In future work, we will study more complex ontologies that can contain more than one kind of relationships, and also try to see how the a-priori score contained in the ontology can be learned to better fit the user's preferences.

Acknowledgments

The authors would like to thank Prof. Martin Golumbic for his helpful insights in graph theory

References

Bradley, K.; Rafter, R.; and Smyth, B. 2000. *Case-Based User Profiling for Content Personalization (AH2000)*. Springer-Verlag. 133–143.

Herlocker, J. L.; Konstan, J. A.; Terven, L. G.; and Riedl, J. T. 2004. Evaluating Collaborative Filtering Recommender Systems. ACM Transaction on Information Systems. In *ACM Transactions on Information Systems*, volume 22, 5 –53.

Jiang, J., and Conrath, D. 1998. Semantic Similarity based on corpus and lexical taxonomy. In *Proc. of 10th Int. Conf. on Research in Computational Linguistics*.

Keeney, R., and Raiffa, H. 1993. *Decisions with Multiple Objectives: Preference and Value Tradeoffs*. Cambridge University Press.

Knappe, R.; Bulskov, H.; and Andreasen, T. 2003. Similarity graphs. In *Proceedings of ISMIS'03*, volume LNAI 2871, 668 – 672. Springer-Verlag.

Li, Q.; Kin, B. M.; Myaeng, S.; and H. 2005. Clustering For Probabilistic Model Estimation for CF. In *Proc. of 14th Int. WWW Conf*, 1104 – 1005.

Middelton, S. E.; Shadbolt, N. R.; and De Roure, D. C. 2004. Ontological User Profiling in Recommender Sytems. *ACM Transactions of Information Systems* 22(1):54 – 88.

Miller, G.; Beckwith, R.; Fellbaum, C.; Gross, D.; and Miller, K. 1993. Introduction to WordNet: An On-line Lexical Database. Technical report, Cognitive Science Laboratory, Princeton University.

Ming, L.; Xin, C.; Xin, L.; Bin, M.; and Vitanyi, P. 2004. The similarity metric. *IEEE Transactions on Information Theory* 50(12):3250 – 3264.

Mobasher, B.; Jin, X.; and Zhou, Y. 2004. Semantically Enhanced Collaborative Filtering on the Web. In *Proceedings of EWMF03*, volume LNAI 3209, 57 – 76. Springer-Verlag.

Resnik, P. 1998. Semantic Similarity in a Taxonomy: An Information-Based Measure and its Application to Problems of Ambiguity in Natural Language. *Journal of Artificial Intelligence Research* 11:95 – 130.

Sarwar, B.; Karypis, G.; Konstan, J.; and Riedl, J. 2001. Item-based Collaborative Filtering Recommendation Algorithms. In *Proc. of 10th Int. WWW Conf*, 285 – 295.

Sullivan, D.; Smyth, B.; Wilson, D.; McDonald, K.; and Smeaton, A. 2004. Improving the Quality of the Personalized Electronic Program Guide. *User Modeling and User-Adapted Interaction* 14(1):5 – 36.

Walley, P. 1996. Measures of uncertainty in expert systems. *Journal of Artificial Intelligence* 83:1 – 58.

Ziegler, C.-N.; McNee, S.; Konstan, J.; and Lausen, G. 2005. Improving Recommendation List Through Topic Diversification. In *Proc. of 14th Int. WWW Conf*, 22 – 32.