Collaborative Filtering by Personality Diagnosis: A Hybrid Memory- and Model-Based Approach

David M. Pennock

University of Michigan
Artificial Intelligence Lab
1101 Beal Ave
Ann Arbor, MI 48109-2110
dpennock@umich.edu

Eric Horvitz

Microsoft Research
One Microsoft Way
Redmond, WA 98052-6399
horvitz@microsoft.com

Abstract

The growth of Internet commerce has stimulated the use of collaborative filtering (CF) algorithms as recommender systems. Such systems leverage knowledge about the known preferences of multiple users to recommend items of interest to other users. CF methods have been harnessed to make recommendations about such items as web pages, movies, books, and toys. Researchers have proposed and evaluated many approaches for generating recommendations. We describe and evaluate a new method called personality diagnosis (PD). Given a user's preferences for some items, we compute the probability that he or she is of the same "personality type" as other users, and, in turn, the probability that he or she will like new items. PD retains some of the advantages of traditional similarity-weighting CF approaches in that all data is brought to bear on each prediction and new data can be added easily and incrementally. Additionally, PD has a meaningful probabilistic interpretation, which may be leveraged to justify, explain, and augment results. We show empirically that PD provides better predictions than all four of the algorithms tested by Breese et al. [1998] on the EachMovie database of movie ratings. The probabilistic framework naturally supports a variety of descriptive measurements—in particular, we consider the applicability of a value of information (VOI) computation.

1 Introduction

The goal of ***collaborative filtering*** (CF) is to predict the preferences of one user, referred to as the ***active user***, based on the preferences of a group of users. For example, given the active user's ratings for several movies and a database of other users' ratings, the system predicts how the active user would rate unseen movies. The key idea is that the active user will prefer those items that like-minded people prefer, or even that dissimilar people don't prefer. The effectiveness of any CF algorithm is ultimately predicated on the underlying assumption that human preferences are correlated—if they were not, then informed prediction would not be possible. There does not seem to be a single, obvious way to predict preferences, nor to evaluate effectiveness, and many different algorithms and evaluation criteria have been proposed and tested. Most comparisons to date have been empirical or qualitative in nature [Billsus and Pazzani, 1998; Breese *et al*., 1998; Konstan and Herlocker, 1997; Resnick and Varian, 1997; Resnick *et al*., 1994; Shardanand and Maes, 1995], though some worst-case performance bounds have been derived [Freund *et al*., 1998; Nakamura and Abe, 1998] and some general principles have been advocated [Freund *et al*., 1998]. Initial methods were statistical, though several researchers have recently cast CF as a machine learning problem [Basu *et al*., 1998; Billsus and Pazzani, 1998; Freund *et al*., 1998; Nakamura and Abe 1998].

Breese *et al.* [1998] identify two major classes of prediction algorithms. ***Memory-based*** algorithms maintain a database of all users' known preferences for all items, and, for each prediction, perform some computation across the entire database. On the other hand, ***model-based*** algorithms first compile the users' preferences into a descriptive model of users, items, and/or ratings; rec-

ommendations are then generated by appealing to the model. Memory-based methods are simpler, seem to work reasonably well in practice, and new data can be added easily and incrementally. However, this approach can become computationally expensive, in terms of both time and space complexity, as the size of the database grows. Additionally, these methods generally cannot provide explanations of predictions or further insights into the data. For model-based algorithms, the model itself may offer added value beyond its predictive capabilities by highlighting certain correlations in the data, offering an intuitive rationale for recommendations, or simply making assumptions more explicit. Memory requirements for the model are generally less than for storing the full database. Predictions can be calculated quickly once the model is generated, though the time complexity to compile the data into a model may be prohibitive, and adding one new data point may require a full recompilation.

In this paper, we propose and evaluate a CF method called ***personality diagnosis*** (PD) that can be seen as a hybrid between memory- and model-based approaches. All data is maintained throughout the process, new data can be added incrementally, and predictions have a meaningful probabilistic semantics. Each user's reported preferences are interpreted as a manifestation of their underlying "personality type." It is assumed that users report ratings for an item with Gaussian error. Given the active user's known ratings of items, we compute the probability that he or she has the same personality type as every other user, and then compute the probability that he or she will like some new item. The full details of the algorithm are given in Section 3.

PD retains some of the advantages of both memory- and model-based algorithms, namely simplicity, extensibility, normative grounding, and explanatory power. In Section 4, we show that PD empirically outperforms all four of the algorithms evaluated by Breese *et. al.* [1998] on a movie ratings data set, according to average absolute deviation. For large amounts of data, a straightforward application of PD suffers from the same time and space complexity concerns as memory-based methods. In Section 5, we describe how the probabilistic formalism naturally supports an ***expected value of information*** (VOI) computation. An interactive recommender could use VOI to favorably order queries for ratings, thereby mollifying what could otherwise be a tedious and frustrating process. VOI could also serve as a guide for pruning entries from the database with minimal loss of accuracy.

2 Background and Notation

Subsection 2.1 discusses previous research on collaborative filtering and recommender systems. Subsection 2.2 describes a general mathematical formulation of the CF problem and introduces any necessary notation.

2.1 Collaborative Filtering Approaches

A variety of collaborative filters or recommender systems have been designed and deployed. The Tapestry system relied on each user to identify like-minded users manually [Goldberg *et al.*, 1992]. GroupLens [Resnick *et al.*, 1994] and Ringo [Shardanand and Maes, 1995], developed independently, were the first CF algorithms to automate prediction. Both are examples of a more general class called ***memory-based*** approaches, where for each prediction, some measure is calculated over the entire database of users' ratings. Typically, a similarity score between the active user and every other user is calculated. Predictions are generated by weighting each user's ratings proportionally to his or her similarity to the active user. A variety of similarity metrics are possible. Resnick *et al.* [1994] employ the ***Pearson correlation coefficient***. Shardanand and Maes [1995] test a few metrics, including correlation and mean squared difference. Breese *et al.* [1998] propose the use of ***vector similarity***, based on the vector cosine measure often employed in information retrieval. All of the memory-based algorithms cited predict the active user's rating as a similarity-weighted sum of the others users' ratings, though other combination methods, such as a weighted product, are equally plausible. Basu et al. [1998] explore the use of additional sources of information (for example, the age or sex of users or the genre of movies) to aid prediction.

Breese *et al.* [1998] identify a second general class of CF algorithms called ***model-based*** algorithms. In this approach, an underlying model of user preferences is first constructed, from which predictions are inferred. The authors describe and evaluate two probabilistic models, which they term the ***Bayesian clustering*** and ***Bayesian network*** models. In the first model, like-minded users are clustered together into classes. Given his or her class membership, a user's ratings are assumed to be independent (i.e., the model structure is that of a naïve Bayesian network). The number of classes and the parameters of the model are learned from the data. The second model also employs a Bayesian network, but of a different form. Variables in the network are titles and their values are the allowable ratings. Both the structure of the network, which encodes the dependencies between titles, and the conditional probabilities are learned from the data. See [Breese *et al.*, 1998] for the full description of these two models. Ungar and Foster [1998] also suggest clustering as a natural preprocessing step for CF. Both users and titles are classified into groups; for each category of users, the probability that they like each category of titles is estimated. The authors compare the results of several statistical techniques for clustering and model estimation, using both synthetic and real data.

CF technology is in current use in several Internet commerce applications. For example, firefly (`http://www.firefly.com`), originally a recommender much like GroupLens and Ringo, now offers more general personalized services based on individual

and community preferences. Alexa (`http://www.alexa.com`) is a web browser plug-in that recommends related links based in part on other people's web surfing habits. Online retailer `Amazon.com` employs CF methods to recommend books to its customers.

2.2 Formal Framework and Notation

A CF algorithm recommends items or ***titles*** to the active user based on the ratings of n users. Denote the set of all m titles as T and the rating of user i for title j as $r_i(j)$. The function $r_i: T \rightarrow \Re \cup \{\perp\}$ maps titles to real numbers or to $\perp$, the symbol for "no rating." Denote the vector of all of user i's ratings for all titles as $\boldsymbol{r_i}(T)$, and the vector of all of the active user's ratings as $\boldsymbol{r_a}(T)$. Define $NR \subset T$ to be the subset of titles that the active user has not rated, and thus for which we would like to provide predictions. That is, title j is in the set NR if and only if $r_a(j) = \perp$.

In general terms, a collaborative filter is a function f that takes as input all ratings for all users, and outputs the predicted ratings for the active user:

$$\boldsymbol{r_a}(NR) = f(\boldsymbol{r_1}(T), \boldsymbol{r_2}(T), \ldots, \boldsymbol{r_n}(T)) \quad (1)$$

where the $\boldsymbol{r_i}(T)$'s include the ratings of the active user.

3 Collaborative Filtering by Personality Diagnosis

Traditional memory-based CF algorithms (e.g., similarity-weighted summations like GroupLens and Ringo) work reasonably well in practice, especially when the active user has rated a significant number of titles [Breese *et al.* 1998]. These algorithms are designed for, and evaluated on, predictive accuracy. Little else can be gleaned from their results, and the outcome of comparative experiments can depend to an unquantifiable extent on the chosen data set and/or evaluation criteria. In an effort to explore more semantically meaningful approaches, we propose a simple model of how people rate titles, and describe an associated ***personality diagnosis*** (PD) algorithm to generate predictions. One benefit of this approach is that the modeling assumptions are made explicit and are thus amenable to scrutiny, modification, and even empirical validation.

Our model posits that user i's ***personality type*** can be described as a vector of "true" ratings $\boldsymbol{r_i}^{\mathbf{true}}(T)$ for all seen titles. These encode his or her underlying, internal preferences for titles. We assume that all users report ratings for titles they've seen with Gaussian noise. That is, user i's actual rating for title j is assumed to be drawn from an independent normal distribution with mean $r_i^{\text{true}}(j)$. Specifically,

$$\Pr\left(r_i(j) = x \middle| r_i^{\text{true}}(j) = y\right) \propto e^{-(x-y)^2/2\sigma^2}, \quad (2)$$

where σ is a free parameter. Thus the same user may report different ratings on different occasions, perhaps depending on the context of any other titles rated in the same session, or on his or her mood, or on other external factors. All factors are summarized here as Gaussian noise. Given the user's personality type, his or her ratings are assumed independent. (If $y = \perp$ in Equation 2, then we assign a uniform distribution over ratings.)

We further assume that the distribution of personality types or ratings vectors in the database is representative of the distribution of personalities in the target population of users. That is, the prior probability $\Pr(\boldsymbol{r_a}^{\mathbf{true}}(T)=\boldsymbol{v})$ that the active user rates items according to a vector $\boldsymbol{v}$ is given by the frequency that other users rate according to $\boldsymbol{v}$. Instead of explicitly counting occurrences, we simply define $\boldsymbol{r_a}^{\mathbf{true}}(T)$ to be a random variable that can take on one of n values—$\boldsymbol{r_1}(T)$, $\boldsymbol{r_2}(T)$, …$\boldsymbol{r_n}(T)$ —each with probability $1/n$.

$$\Pr\left(r_a^{\text{true}}(T) = r_i(T)\right) = \frac{1}{n} \quad (3)$$

From Equations (2) and (3), and given the active user's ratings, we can compute the probability that the active user is of the same personality type as any other user, by applying Bayes' rule.

$$\begin{aligned}\Pr\left(r_a^{\text{true}}(T) = r_i(T) \middle| r_a(1) = x_1, \ldots, r_a(m) = x_m\right) \propto \\ \Pr\left(r_a(1) = x_1 \middle| r_a^{\text{true}}(1) = r_i(1)\right) \cdot \cdot \\ \Pr\left(r_a(m) = x_m \middle| r_a^{\text{true}}(m) = r_i(m)\right) \cdot \Pr\left(r_a^{\text{true}}(T) = r_i(T)\right)\end{aligned} \quad (4)$$

Once we compute this quantity for each user i, we can compute a probability distribution for the active user's rating of an unseen title j.

$$\begin{aligned}\Pr\left(r_a(j) = x_j \middle| r_a(1) = x_1, \ldots, r_a(m) = x_m\right) = \\ \sum_{i=1}^{n} \Pr\left(r_a(j) = x_j \middle| r_a^{\text{true}}(T) = r_i(T)\right) \cdot \\ \Pr\left(r_a^{\text{true}}(T) = r_i(T) \middle| r_a(1) = x_1, \ldots, r_a(m) = x_m\right)\end{aligned} \quad (5)$$

where $j \in NR$. The algorithm has time and space complexity $O(n \cdot m)$, as do the memory-based methods described in Section 2.1. The model is depicted as a naïve Bayesian network in Figure 1. It has the same structure as a classical diagnostic model, and indeed the analogy is apt. We observe ratings ("symptoms") and compute the probability that each personality type ("disease") is the

cause using Equation 4. We then can compute the probability of rating values for an unseen title *j* using Equation 5. We return the ***most probable*** rating as our prediction.

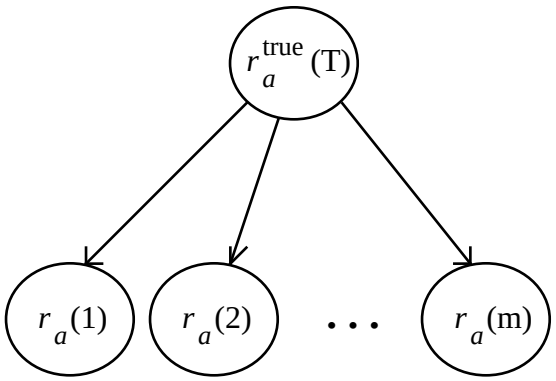

Figure 1. Naïve Bayesian network semantics for the PD model. Actual ratings are independent and normally distributed given the underlying "true" personality type.

An alternative but equivalent interpretation of this model is as follows. The active user is assumed to be "generated" by choosing one of the other users uniformly at random and adding Gaussian noise to his or her ratings. Given the active user's known ratings, we infer the probability that he or she is actually one of the other users, and then compute the probabilities for ratings of other items. PD can also be thought of as a "clustering" method [Breese *et al.*, 1998; Ungar and Foster, 1998] with exactly one user per cluster. The general approach of casting CF as a classification problem has been advocated and examined previously [Basu *et al.*, 1998; Billsus and Pazzani, 1998; Freund *et al.*, 1998; Nakamura and Abe 1998]. Note that in the PD model, the only free parameter is σ.

4 Empirical Results

We have evaluated the PD algorithm on a subset of the EachMovie database, available from the Digital Equipment Research Center.[1] This data contains many thousands of users' ratings for various movies, elicited on a scale from 0 to 5. We used the same subset of the data as Breese *et al.* [1998], consisting of 1623 titles, 5000 users in the training set and 4119 users in the test set. On average, each user rated about 46 movie titles. To carry out testing, we withhold some of the ratings of users in the test set and attempt to predict them using the PD algorithm. Again following the methodology of Breese *et al.* [1998], we employ four different protocols. Under the first protocol, called ***all but one***, we withhold for prediction only one rating for each user in the test set; all other ratings are used as input for the PD algorithm. In the other three protocols, ***given ten***, ***given five***, and ***given two***, we retain the given number of ratings for each user for input to the algorithm, and try to predict the rest. Each protocol admits less information than the previous, and we should expect a corresponding decrease in accuracy. If a user does rate enough movies to satisfy a particular protocol, then he or she is dropped from that experiment. We set σ to 2.5, though results did not seem to be particularly sensitive to this parameter.

Breese et al. [1998] propose two evaluation criteria to measure accuracy: rank scoring and average absolute deviation. We consider here only the latter. Let $r_a^{\text{pred}}(j)$ be our predicted rating for title *j*, and let *num_p* be the total number of predictions made for all users in the test set. Then the average absolute deviation is simply $1/num_p \; \Sigma|r_a^{\text{pred}}(j) - r_a(j)|$.

The results are summarized in Table 1. Scores for all algorithms except PD are transcribed directly from Breese et al. [1998]. We did *not* replicate their experiments, though we used the same data. Due to randomization, we almost certainly did not withhold exactly the same titles for prediction. PD performed statistically significantly better than each of the other four algorithms under all four protocols. In fact, PD under the given-ten protocol outperformed correlation under the all-but-one protocol, which was the previous best score. Note that, among the other four algorithms, none was a strict winner.

Algorithm	All But 1	Given 10	Given 5	Given 2
PD	0.951	0.981	1.015	1.034
Correl.	0.994	1.069	1.139	1.257
V. Sim.	2.136	2.235	2.177	2.113
B. Clust.	1.103	1.138	1.144	1.127
B. Net.	1.066	1.139	1.154	1.143

Table 1. Average absolute deviation scores for PD and for the four algorithms tested in Breese et al. [1998]. Correlation and vector similarity are memory-based algorithms; Bayesian clustering and Bayesian network are model-based. PD performed best under all conditions.

5 Harnessing Value of Information in Recommender Systems

Formulating collaborative filtering as the diagnosis of personality under uncertainty provides opportunities for leveraging information- and decision-theoretic methods to provide functionalities beyond the core prediction service. We have been exploring the use of the ***expected value of information*** (VOI) in conjunction with CF. VOI computation identifies, via a cost–benefit analysis, the most valuable new information to acquire in the context

[1] `www.research.digital.com/SRC/EachMovie`

of a current probability distribution over states of interest [Howard, 1968]. In the current context, a VOI analysis can be used to drive a hypothetico-deductive cycle [Horvitz *et al.*, 1988] that identifies at each step the most valuable ratings information to seek next from a user, so as to maximize the quality of recommendations.

Recommender systems in real-world applications have been designed to acquire information by explicitly asking users to rate a set of titles or by implicitly watching the browsing or purchasing behavior of users. Employing a VOI analysis makes feasible an optional service that could be used in an initial phase of information gathering or in an ongoing manner as an adjutant to implicit observation of a user's interests. VOI-based queries can minimize the number of explicit ratings asked of users while maximizing the accuracy of the personality diagnosis. The use of general formulations of expected value of information as well as simpler information-theoretic approximations to VOI hold opportunity for endowing recommender systems with intelligence about evidence gathering. Information-theoretic approximations employ measures of the expected change in the information content with observation, such as relative entropy [Bassat, 1978]. Such methods have been used with success in several Bayesian diagnostic systems [Heckerman *et al.*, 1992].

Building a VOI service requires the added specification of utility functions that captures the cost of querying a user for his or her ratings. A reasonable class of utility models includes functions that cast cost as a monotonic function of the number of items that a user has been asked to evaluate. Such models reflect the increasing frustration that users may have with each additional rating task. In an explicit service guided by such a cost function, users are queried about titles in decreasing VOI order, until the expected cost of additional requests outweigh the expected benefit of improved accuracy.

Beyond the use of VOI to guide the gathering of preference information, we are pursuing the offline use of VOI to compress the amount of data required to produce good recommendations. We can compute the average information gain of titles and/or users in the data set and eliminate those of low value accordingly. Such an approach can provide the means for both alleviating memory requirements and improving the running time of recommender systems with as little impact on accuracy as possible.

6 Conclusion

We have described a new algorithm for collaborative filtering (CF) called personality diagnosis (PD), which can be thought of as a hybrid between existing memory- and model-based algorithms. Like memory-based methods, PD is fairly straightforward, maintains all data, and does not require a compilation step to incorporate new data. Most memory-based algorithms operate as a "black box": efficacy is evaluated by examining only the accuracy of the output. Since results do not have a meaningful interpretation, the reason for success or failure is often hard to explain, and the search for improvements becomes largely a trial-and-error process. The PD algorithm is based on a simple and reasonable probabilistic model of how people rate titles. Like other model-based approaches, its assumptions are explicit, and its results have a meaningful probabilistic interpretation. According to absolute deviation, PD makes better predictions than four other algorithms—two memory-based and two model-based—under four conditions of varying information about the active user. We also discussed how value of information might be used in the context of an interactive CF algorithm or a data compression scheme.

Acknowledgments

Thanks to Jack Breese, Carl Kadie, and the anonymous reviewers for insightful comments and pointers to related work.

References

[Bassat, 1978] M. Ben-Bassat. Myopic policies in sequential classification. *IEEE Transactions on Computers*, 27: 170–178, 1978.

[Basu *et al.*, 1998] Chumki Basu, Haym Hirsh, and William Cohen. Recommendation as classification: Using social and content-based information in recommendation. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, pages 714–720, July 1998.

[Billsus and Pazzani, 1998] Daniel Billsus and Michael J. Pazzani. Learning collaborative information filters. In *Proceedings of the Fifteenth International Conference on Machine Learning*, pages 46–54, July 1998.

[Breese *et al.*, 1998] John S. Breese, David Heckerman and Carl Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth Annual Conference on Uncertainty in Artificial Intelligence*, pages 43–52, July 1998.

[Freund *et al.*, 1998] Yoav Freund, Raj Iyer, Robert E. Schapire, and Yoram Singer. An Efficient boosting algorithm for combining preferences. In *Proceedings of the Fifteenth International Conference on Machine Learning*, pages 170–178, July 1998.

[Goldberg *et al.*, 1992] David Goldberg, David Nichols, Brian M. Oki, and Douglas Terry. Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 35(12): 61–70, December 1992.

[Heckerman *et al*., 1992] D. Heckerman, E. Horvitz, and B. Nathwani. Toward normative expert systems: Part I. The Pathfinder project. *Methods of Information in Medicine*, 31: 90–105, 1992.

[Horvitz *et al*., 1988] E. J. Horvitz, J. S. Breese, and M. Henrion. Decision theory in expert systems and artificial intelligence. *International Journal of Approximate Reasoning*, Special Issue on Uncertain Reasoning, 2:247–302, 1988.

[Howard, 1968] R. A. Howard. The foundations of decision analysis. *IEEE Transactions on Systems, Science, and Cybernetics*, 4: 211–219, 1968.

[Konstan and Herlocker, 1997] Joseph A. Konstan, Bradley N. Miller, David Maltz, Jonathan L. Herlocker, Lee R. Gordon, and John Riedl. GroupLens: Applying collaborative filtering to Usenet news. *Communications of the ACM*, 40(3): 77–87, March 1997.

[Nakamura and Abe, 1998] Atsuyoshi Nakamura and Naoki Abe. Collaborative filtering using weighted majority prediction algorithms. In *Proceedings of the Fifteenth International Conference on Machine Learning*, pages 395–403, July 1998.

[Resnick and Varian, 1997] Paul Resnick and Hal R. Varian. Recommender systems. *Communications of the ACM*, 40(3): 56–58, March 1997.

[Resnick *et al*., 1994] Paul Resnick, Neophyts Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. GroupLens: An Open architecture for collaborative filtering of netnews. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work*, pages 175–186, 1994.

[Shardanand and Maes, 1995] Upendra Shardanand and Pattie Maes. Social information filtering: Algorithms for automating "word of mouth." *In Proceedings of Computer Human Interaction*, pages 210–217, May 1995.

[Ungar and Foster, 1998] Lyle H. Ungar and Dean P. Foster. Clustering methods for collaborative filtering. In *Workshop on Recommendation Systems at the Fifteenth National Conference on Artificial Intelligence*, July 1998.