5-11-2017

Information-Based Neighborhood Modeling

Marcelo Matus Nicodemos
mam9054@rit.edu

Information-Based Neighborhood Modeling

by

Marcelo Matus Nicodemos

<u>Committee Members</u>

Steve Zilora

Qi Yu

Jai Kang

Thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science in Information Technology

Rochester Institute of Technology

B. Thomas Golisano College

of

Computing and Information Sciences

Department of Information Sciences and Technologies

05/11/2017

Information-Based Neighborhood Modeling

Marcelo Matus Nicodemos

Rochester Institute of Technology

Table of Contents

Abstract

Since the inception of the World Wide Web, the amount of data present on websites and internet infrastructure has grown exponentially that researchers continuously develop new and more efficient ways of sorting and presenting information to end-users. Particular websites, such as e-commerce websites, filter data with the help of recommender systems. Over the years, methods have been developed to improve recommender accuracy, yet developers face a problem when new items or users enter the system. With little to no information on user or item preferences, recommender systems struggle generating accurate predictions. This is the cold-start problem. Ackoff defines information as data structured around answers to the question words: what, where, when, who and how many. This paper explores how Ackoff's definition of information might improve accuracy and alleviate cold-start conditions when applied to the neighborhood model of collaborative filtering (Ackoff, 1989, p. 3).

Keywords: recommender systems, collaborative filtering, neighborhood model, latent factor model, matrix factorization, data, information, DIKW hierarchy

Part I – Introduction, Problem Statement, Hypothesis

Introduction

The abundance of data on the internet presents developers with a search and retrieval problem similar to the proverbial needle in the haystack. The needle represents relevant information, while the hay represents every possible bit of retrievable information. This retrieval problem is known as the Information Overload problem, a problem that can strain systems and developers when dealing with larger, and sometimes sparser, amounts of data (Himma & Tavani, 2008, p. 497-498). E-commerce websites utilize recommender systems, a series of services and applications which help alleviate information overload problems by sorting through data and presenting what's relevant to end-users and consumers. Recommender systems exist within three categories: content-based, collaborative filtering, and hybrid systems. Content-based recommender systems sort items and information according to their category, while collaborative filtering relies on algorithms and computational models to generate recommendations. The increase in computer storage capacity and processing power has proven beneficial for systems that rely on model-based approaches, with much of the research focusing on accuracy and efficiency improvement (Liu, Zhao, Xiang & Yang, 2010, p. 102; Ding & Li, 2005, p. 490; O'Donovan & Smyth, 2005, p. 172; Lai, Liu & Lin, 2013, p. 44-47; Jeong, Lee & Cho, 2008, p. 7312; Bobadilla, Hernando, Ortega & Gutierrez, 2011, p. 15). The lack of information on new items or users in a system presents a different challenge: the cold-start problem, which researchers have been tackling by incorporating demographic information into their algorithms and calculations (Chekkai, Chikhi & Kheddouci, 2012, p. 760; Eckhardt, 2012, p. 11511-11512; Moreno et al., 2011, p. 256-257; Zhang, Liu, Zhang & Zhou, 2010, p. 1-2). This paper explores a new method for dealing with: scalability, information overload, and data sparsity and cold-start condition issues as they apply to the neighborhood model of collaborative filtering (Su & Khoshgoftaar, 2009, p. 3; Lu et al, 2012, p. 3-5).

Information Overload, Recommender Systems, and the Cold-Start Problem

The growth of the World Wide Web introduced the Information Overload problem. As a result, e-commerce websites like Amazon and Netflix face challenges when presenting relevant merchandise to their end-users. Amazon's catalog numbers in the hundreds of thousands if not millions of items. If Amazon presented its entire inventory to end-users, they risk losing a customer and potential sale. According to Lu et al., statistics show that relevant information leads to returning customers and increase in sales, with Amazon reporting that twenty to forty percent of its sales are attributable to recommender system performance. Likewise, Netflix attributes sixty percent of its user activity to its personalization and recommendation system (Lu et al., 2012, p. 3). The ability of recommender systems to filter and present relevant data, and the number of sales increase Amazon and Netflix attributes to its systems demonstrates that such technology has inherent value. Focus now falls on improving recommendation performance, particularly in two areas: prediction accuracy and alleviation of cold-start problems (Lu et al., 2012, p. 3-5).

Recommender systems lie within three categories: content-based, collaborative filtering, and hybrid systems. Hybrid systems combine approaches found in content-based and collaborative filtering systems. Content-based systems rely on juxtaposition of item categories, while collaborative filtering relies on algorithms that examine the behavior and preferences of elements within the system, namely users and items. Researchers and industry experts refer to two types of collaborative filtering approaches: memory-based and model-based. Memory-based examines user preferences in relation to items, and vice-versa. Users provide ratings for items and the memory-based system runs algorithms that examine the similarity of users or items within a neighborhood of other users or items, namely those with the highest number of similar ratings. Another name for the memory-based approach is the neighborhood model, and recommendations can be generated either via user-orientation or item-orientation (Su & Khoshgoftaar, 2009, p. 5-8; Lu et al., 2012, p. 15).

The other collaborative filtering approach is the model-based approach. Two major models are associated with model-based collaborative filtering: latent factoring and matrix factorization. While there are several implementations, with various optimization and regularization techniques for different implementations of matrix factorization, latent factor modeling and matrix factorization rely on a matrix model similar to that produced by the neighborhood model. Unlike the neighborhood model, both latent factor modeling and matrix factorization go a step further in retrieving information that assists in the general rating prediction process. One such method of extracting latent information within the original user-to-item rating matrix is singular value decomposition, which breaks the matrix into factors of itself (Barbieri, Manco & Ritacco, 2014, p. 17). Although Koren, Bell and Volinsky mention that matrix factorization results demonstrate improved accuracy over traditional neighborhood models, researchers continue to search for ways of improving memory-based and model-based approaches (Koren, Bell, & Volinsky, 2009, p. 35-36). Another area of concern is the cold-start problem, which occurs when not enough information is available to generate accurate recommendations (Su & Khoshgoftaar, 2009, p. 2; Lu et al., 2012, p. 4). Researchers have tackled several approaches aimed at alleviating the cold-start problem, with several work focusing on demographic information as a viable way to account for the lack of any information pertaining to a user’s preferences (Chekkai et al., 2012, p. 760; Eckhardt, 2012, p. 11511-11512;

Moreno et al., 2011, p. 256-257; Zhang et al., 2010, p. 1-2). While demographic information may be an excellent substitute for traditional rating scales, as those found in neighborhood modeling approaches, this paper examines a specific definition of information in an attempt to provide a foundation for alleviating the cold-start and recommendation accuracy problems. This paper examines Ackoff's definition of information, which structures information around its relationship to data (Ackoff, 1989, p. 3).

Ackoff's Definition of Data and Information

Recommender systems deal with the processing, filtering, sorting and presentation of information. However, the definition of information varies according to discipline. Disciplines such as communications theory, library and information sciences, cognitive sciences, and management sciences have their own respective definitions. Ackoff (1989) presented the notion that information resides within a hierarchy known as the data, information, knowledge and wisdom (DIKW) hierarchy. He was a researcher and developer who examined conceptual and organizational hierarchy for use in corporate, business, and managerial organizations and positions. Of interest to him was the flow of data in business settings. Through his examinations, he defined information through its relationship to data. He provides the following remarks:

> Data are symbols that represent properties of objects, events and their environment. They are productions of observation. Data, like metallic ores, are of no value until they are processed into useable form. Information is contained in descriptions, answers to questions that begin with such words as who, what, where, when, and how many (p. 3).

According to Ackoff, data is nothing more than what is observable and identifiable in physical environments, in particular the corporate and institutional environment where he conducted his work. Data becomes information when it is processed into useable forms. Data that becomes useable information, according to Ackoff, answers questions that being with the question words he mentioned above. Although such processing of observable bits of data, either in software applications or in physical environments, can occur in differing computational devices and programs, this thesis project explores how the structuring of data following Ackoff's definition of information can benefit recommender systems in both predication relevancy and accuracy improvement and in alleviating cold-start problems (Ackoff, 1989, p. 3).

Hypothesis

A collaborative filtering (CF) neighborhood model structured around Ackoff's definition of information will both generate more accurate predictions and alleviate cold-start conditions when compared to the traditional neighborhood model.

<u>Part II – Literature Review</u>

Background Information

Since the early 1990s, recommender systems have followed three basic types of configurations or approaches (Balabanovic & Shoham, 1997, p. 66-67). These include content-based, collaborative filtering, and hybrid approaches to recommendations.

<u>Content-Based Recommender Systems</u>

Content-Based recommender systems provide recommendations based on labels and ontologies that categorize and facilitate the identification of item-to-item relationships, meaning, items within a categorical subset are more likely to share similarities than items outside their sub-categorical groups. An example of this type of categorization exists within physical bookstores across the United States and many other countries. For example, books carry labels such as fiction or nonfiction, genre fiction versus literary or mainstream fiction, or several types of nonfiction based on topic. Content-based recommender systems sort and label items according to category and descriptive labels, after which recommendations are generated via the degree of similarity between items in relation to the labels and categories associated with them.

The type of process that occurs during a typical content-based recommendation can best be exemplified by analyzing how people recommend or suggest new books for book lovers to read. If a person identifies themselves as an avid reader of science fiction, his or her friends might suggest they read popular books in the science fiction category. If that person goes on to explain that they like romance in their science fiction, then people might suggest science fiction books that feature romantic relations between the characters. Content-based recommender systems follow this type of logic, but instead of relying on word-of-mouth or interactions between the users, content-based systems apply the labels via several different methods, one of them being manual cataloguing of items (Lu et al., 2012, p. 9).

Unlike content-cased recommender systems, collaborative filtering systems do take into account user preferences and feedback. The previous example utilized a scenario where actual people interacted with each other and provided recommendations based on what they read. Recommender systems following a purely content-based approach would have a difficult time generating recommendations based on particular tastes and peer-to-peer interactions. In realistic, contemporary world scenarios, many users receive instantaneous recommendations when they load a homepage for a website like Amazon. They do not have to seek out peers and ask for recommendations. But computer-generated recommendations have improved significantly since the inception of the first recommender systems; several follow models that simulate interactions between people that lead to recommendations. The approach that takes into consideration the stated and unstated preferences of a user, or the explicit or implicit choices a user makes when visiting a website like Amazon, is known as the collaborative filtering technique (Su & Khoshgoftaar, 2009, p. 2).

Collaborative Filtering

Collaborative filtering (CF) looks at the similarities between users or items in terms of preferences, provided or deduced, rather than similarities of items in relation to categories, labels, and content-type. A common approach to collaborative filtering is the memory-based approach, sometimes referred to as the neighborhood model, which examines user ratings across several users or items to generate recommendations. Continuing with the previous example of science fiction fans recommending books to each other, collaborative filtering mimics a group of science fiction fans exploring each other's tastes before coming to a decision about which book a specific reader might like. A science fiction fan might mention that he or she read a book which featured social commentary and aliens. A subset of people in a gathering of science fiction fans might respond by saying they read books A, B, and C that all feature social commentary and aliens. They might come to an agreement about which of the three books is the better book, which they then suggest to their friend. Collaborative filtering systems, to a certain degree, mimic this kind of behavior. However, users are not required to interact with each other directly to suggest recommendations. A number of algorithms and computational models have been developed to determine similarities between users. Within the neighborhood model, recommendations are generated by evaluating similarity between groups of users or items. This is accomplished by examining three key sources of information: a list of users, a list of items, and user-to-item ratings. A user (User A) in an e-commerce website or service, such as Netflix, would purchase and rate a series of movies. This forms User A's user-to-item ratings list. Their list of ratings is then compared to the list of ratings from other users. If there's a strong correlation between User A's list and another user's list (User B), then items in User B's list that received no rating from User A might filter in to the recommendation list provided by the CF system. This user-centric approach is known as a user-oriented neighborhood model. An item-oriented approach would focus on one item (Item A) and its similarities to other items based on user ratings (Lu et al., 2012, p.10).

Barbieri, Manco and Ritacco describe the neighborhood model algorithm, the K-Nearest Neighbor (K-NN). The K-NN algorithm provides the main functionality for memory-based collaborative filtering systems. It defines the rating prediction $\hat{r}_i^u$ as a similarity function that finds the K neighbors most similar to user *u*, and averages out the ratings for items in that neighborhood. The average is then weighted against a similarity coefficient. Equation (1) represents the user-oriented algorithm presented in Barbieri, Manco and Ritacco's paper:

$$\hat{r}_i^u = \frac{\sum_{v \in N^K(u)} S_{u,v} \cdot r_i^v}{\sum_{v \in N^K(u)} S_{u,v} \cdot r_i^v} \tag{1}$$

v stands for a member of the set of nearest *K* neighbors *N* for user *u*, while *S* stands for the similarity function for user *u* given the rating *r* from *v* for item *i*.

Equation (2) represents the item-oriented version of the K-NN algorithm, where *j* represents an item in the K-Nearest Neighbor set and *i;u* the item under consideration:

$$\hat{r}_i^u = \frac{\sum_{j \in N^K(i;u)} S_{u,j} \cdot r_j^v}{\sum_{j \in N^K(i;u)} S_{u,j}} \tag{2}$$

Similarity coefficients play a major role in the K-Nearest Neighbor algorithms. They help generate the nearest neighbors, and they act as a weight for the prediction phase (Barbieri, Manco & Ritacco, 2014, p. 15). Su and Khoshgoftaar (2009), and Barbieri, Manco and Ritacco (2014) name the Pearson Correlation, the Vector Cosine Similarity, and the Adjusted Vector Cosine as the similarity coefficients most usually applied to the neighborhood model (Su & Khoshgoftaar, 2009, p. 5-6; Barbieri, Manco & Ritacco, 2014, p. 15). The Pearson Correlation applied to user-oriented K-NN algorithms with two users labeled *u* and *v* is represented in equation (3):

$$w_{u,v} = \frac{\sum_{i \in I}(r_{u,i} - \bar{r}_u)(r_{v,i} - \bar{r}_v)}{\sqrt{\sum_{i \in I}(r_{u,i} - \bar{r}_u)^2}\sqrt{\sum_{i \in I}(r_{v,i} - \bar{r}_v)^2}} \quad (3)$$

$w_{u,v}$ is the similarity based on the $i \in I$ summations over items both users have rated against an average rating of co-rated items $\bar{r}_u$ for user u. Table 1 provides a matrix with rating data from users for items:

Table 1

User to Item Rating Matrix with Sample Data

	Item 1	Item 2	Item 3	Item 4
User 1	4		5	5
User 2	4	2	1	
User 3	3		2	4
User 4	4	4		
User 5	2	1	3	5

The similarity between users 1 and 5 ($w_{1,5}$) is 0.756.

Aside from comparing user similarities, the Pearson Correlation calculates similarities between items according to equation (4):

$$w_{i,j} = \frac{\sum_{u \in U}(r_{u,i} - \bar{r}_i)(r_{u,j} - \bar{r}_j)}{\sqrt{\sum_{u \in U}(r_{u,i} - \bar{r}_i)^2}\sqrt{\sum_{u \in U}(r_{u,j} - \bar{r}_j)^2}} \quad (4)$$

with $r_{u,i}$ standing for the rating of user *u* on item *i*, and $\bar{r}_i$ being the average rating for the given item *i*, calculation would proceed similarly as with user-based Pearson Correlation but with an item-specific orientation (Su & Khoshgoftaar, 2009, p. 5-6).

The Vector Cosine Similarity examines the similarity between two documents. In word documents, the Vector Cosine Similarity establishes word frequency via vectors containing the occurrences of each word in a document. When applied to CF systems, users and items replace documents and term frequency. As seen in the matrix above, user and item lists generate rows and columns which contain the ratings a user gives to a specific item. The matrix can be defined as R, with m X n rows and columns. Given items (or users) *i* and *j*, the Vector Cosine Similarity between the two is defined as the cosine of the n dimensional vector corresponding to the i^{th} and j^{th} column of matrix R:

$$w_{i,j} = \cos(\vec{i}, \vec{j}) = \frac{\vec{i} \cdot \vec{j}}{||\vec{i}|| * ||\vec{j}||} \tag{5}$$

Barbieri, Manco and Ritacco mention the prevalence of the Adjusted Cosine in CF systems operating on K-NN approaches. The Adjusted Cosine is similar to regular Vector Cosine similarity, but with a significant adjustment: the rating function present in the Pearson Correlation is applied to the dot product and the multiplication of the absolute values of the two vectors:

$$w_{i,j} = \text{Adjusted } \cos(\vec{i}, \vec{j}) = \frac{\sum_{i \in I_R} \vec{i} \cdot \vec{j}}{\sum_{i \in I_R} ||\vec{i}|| * ||\vec{j}||} \tag{6}$$

The R subscript is used to denote the current item rating compared to the average ratings of similar item from other users (Barbieri, Manco & Ritacco, 2014, p. 15-16; Su & Khoshgoftaar, 2009, p. 5-6).

Su and Khoshgoftaar note that correlation-based similarity coefficients are used to weight and generate the nearest neighbors. These include the Spearman Rank correlation, Kendall’s Tau, and conditional probability-based similarity metrics. Kendall’s Tau and Spearman Rank are similar to Pearson Correlation, but they define ratings as ranks, and while the Spearman Rank examines numerous ranks to determine similarity, Kendall’s Tau examines only those that are relevant (Su & Khoshgoftaar, 2009, p. 6).

Memory-based, neighborhood model CF is a simple and effective approach to providing recommendations, but it has its limitations. Primarily, if the amount of data is either too large or too sparse, the reliability of memory-based CF systems begins to falter. In this regard, model-based approaches prove ideal for dealing with sparsity and scalability issues. Barbieri, Manco and Ritacco note that model-based approaches outperform memory-based systems in the areas of accuracy and overall predication performance. While there are a number of model-based approaches, such as Bayesian Belief Net CF, Clustering-based CF, Regression-based CF and MDP-based CF, this paper limits its examination to two oft-mentioned models in the recommender system literature: latent factor modeling and matrix factorization (Su & Khoshgoftaar, 2009, p. 8-10; Barbieri, Manco & Ritacco, 2014, p. 16-17).

Latent factor modeling excels in a specific area within the recommendation generation process where neighborhood models falter. Take, for example, two people discussing their preferences of movies. They might come to an agreement that the Matrix movies exist within a shared universe or setting, but computers are unable to identify this simple correlation without a number of programmed steps and instructions. While content-based recommender systems can and do sort movies according to item content, neighborhood modeling cannot identify what makes one movie similar to another based on ratings alone. Latent factoring modeling addresses this issue without having to explicitly rely on identifying labels (Koren & Bell, 2007, p. 146).

The latent factor model searches for information that might be latent in the user-to-item rating data. Singular value decomposition (SVD) is an approach that divides the user-to-item rating matrix into feature matrices that provide more information about user and item preferences. SVD stems from linear algebra operations on matrices. Matrices represent rows and columns of data. A matrix is a visual representation of data stored in a two-dimensional array, where rows represent the first dimension of the array, and the columns the second. Data in memory-based methods fit into matrices, as illustrated in the examples above. SVD factors the data from one matrix to establish multiple matrices of latent data. Similar to real number factorization, matrix factorization produce factors of the original matrix, which, if multiplied together, generate the original matrix (Sarwar, Karypis, Konstan & Riedl, 2000, p. 4-5):

$$SVD(A) = U \times \Sigma \times V^T \tag{7}$$

Matrix A equals the matrices U, Σ and V^T multiplied together. If a list of users and items generates data stored in matrix A, then the matrices U, Σ and V^T represent latent features associated with the users and items in matrix A (Sarwar et al., 2000, p. 4-5). Barbieri, Manco and Ritacco illustrate the process via four matrices. Table 2 represents a user-to-item ratings matrix.

Table 2

SVD Latent Factor Example: Original User-to-Item Rating Matrix

	Item 1	Item 2	Item 3
User 1	3	4	5
User 2	4	2	5
User 3	3	2	4
User 4	5	4	1
User 5	5	5	1

The application of SVD to the original matrix factors it into three feature matrices: the user-to-features matrix, the feature relevancy matrix, and the item-to-features matrix. Tables 3, 4, and 5 represent these three matrices:

Table 3

SVD Latent Factor Example: User-to-Features Matrix

	Comedy	Action	Romance
User 1	0.48	0.34	-0.72
User 2	0.45	0.45	0.56
User 3	0.37	0.34	0.19
User 4	0.42	-0.58	0.24
User 5	0.50	-0.49	-0.19

Table 4

SVD Latent Factor Example: Feature Relevancy Matrix

Comedy	Action	Love
14.06	0	0
0	4.41	0
0	0	1.66

Table 5

SVD Latent Factor Example: Item-to-Features Matrix

	Comedy	Action	Romance
Item 1	0.64	0.54	0.54
Item 2	-0.35	-0.42	0.84
Item 3	0.69	-0.72	-0.07

Over the years developers and researchers have expanded on latent factor and matrix factorization models with optimization and regularization techniques. Several techniques exist, including measures that force data values to assume non-negativity, the nonnegative matrix factorization (Wang & Zhang, 2013, p. 1337), and the Stochastic Gradient Descent, which adjusts values at given intervals (Koren, Bell & Volinsky, 2009, p. 33). Barbieri, Manco, and Ritacco illustrate an optimization approach with equation (8):

$$(U, V) = \underset{U,V}{\operatorname{argmin}}[\sum_{(u,i)\in T}(r_i^u - \sum_{k-1}^{K} U_{u,k}V_{k,i})^2] \tag{8}$$

with U and V representing low-ranking approximations of the original user-to-item ratings matrix.

Equation (8) represents an optimized method of extracting feature matrices, and the equation can be further optimized with a regularization technique such as Maximum Margin Matrix Factorization (MMMF). MMMF allows several factors to exist within the system, but limits the number of factors that are considered important (Barbieri, Manco & Ritacco, 2014, p. 20). Barbieri, Manco and Ritacco illustrate an adjustment equation (8) with regularization in the form of coefficients λ_U and λ_V:

$$(U, V) = \underset{U,V}{\operatorname{argmin}}[\sum_{(u,i)\in T}(r_i^u - \sum_{k-1}^{K} U_{u,k}V_{k,i})^2 + \lambda_U tr(U^T U) + \lambda_V tr(V^T V)] \tag{9}$$

where the function $tr(A)$ stands in for the squared matrix of A (Barbieri, Manco & Ritacco, 2014, p. 20). Koren, Bell and Volinsky reiterate a similar statement while discussing optimization with Stochastic Gradient descent and Alternating least squares (Koren, Bell & Volinsky, 2009, p. 32-33).

<u>Hybrid Systems</u>

It’s important to note the existence of hybrid systems. Hybrid systems take many forms, from combinations of content-based features and collaborative filtering features, to combinations of different methods within memory-based and model-based CF (Su & Khoshgoftaar, 2009, p. 3).

Latent factor modeling and matrix factorization are two widely used model-based approaches to CF, yet research continues to explore novel ways of improving both the traditional memory-based technique and model-based methods. Below are examples of time-based, trust-based, and extra model approaches.

Time-Based Collaborative Filtering

Temporal collaboration filtering focuses on the application of time-based metrics to either memory-based or model-based CF. Liu, Zhao, Xiang and Yang's 2010 research explored differences between movie ratings over an extended time period. If two users enjoyed similar movies at one point, but later their similarity diverged when new ratings entered the system, Liu et al. calculated the newer and more general similarity to account for changes over time. Liu and his team compared the results of their Evolutionary Nearest Neighbor Method (ENN) to a time-aware matrix factorization model and a probabilistic latent semantic analysis (PLSA) model. The matrix factorization model and the PLSA model outperformed their ENN model in terms of prediction accuracy. However, they also examined the run time of calculations. Their ENN model surpassed both the time-aware matrix factorization and PLSA models, since their algorithms and methods required more time to generate results, and the incremental updates during each step of those models resulted in reduced runtime speeds. This result prompted Liu and his team to deduce that their method, overall, was optimal when consider the temporal nature of the recommendations they were generating (Liu et al., 2010, p. 97-102).

Ding and Li offered a different take on time-weighted collaborative filtering. Instead of adjusting general rating scores over a span of time, Ding and Li's research applied a temporal weighting metric over a traditional, item-oriented neighborhood model which emphasized recently rated items over previously rated items. Ding and Li found that their model demonstrated improved rating predictions over the traditional item-oriented model (Ding & Li, 2005, p. 487-490).

Trust-Based Collaborative Filtering

O'Donovan and Smyth's research explored how the idea of a trust might impact prediction accuracy. They labeled trust as user-defined scores which identified the overall level of rating and review helpfulness. They provided examples of this type of interaction in their discussion of the Epinions website, where users rate ratings and reviews based on overall helpfulness. If they found a review did not help them consider a particular item, the trust score decreased. If ratings and reviews received high scores on helpfulness, then trust levels for that item went up. This feature is also available on Amazon's website, where every user-submitted review carries with it a helpful or not rating associated with the number of people who found the review helpful out of a total number of user-submitted yes or no scores. O'Donovan and Smyth applied trust-based weighting mechanism that relied on a modified Resnick prediction formula. They also associated each modified formula to account for different levels of trust. They examined scenarios where their algorithm applied a weighting metric over the prediction formula versus an algorithm that filtered results according to profile- or item-level trust. Their overall results showed that the standard Resnick model outperformed predictions that utilized profile-level and item-level trust weighting metrics. However, on a number of occasions their filtering methods surpassed the baseline numbers of the standard Resnick formula (O'Donovan & Smyth, 2005, p. 167-172).

While O'Donovan and Smyth proposed using the concept of trust in filtering and CF weighting contexts (p. 170), Lai, Liu and Lin (2013) looked at global and personal reputation models for improving recommendation accuracy, basing their system on personal- and group-levels of trustworthiness. For personal-level trust, they examined a given user's rated items to similar items rated by other users. If their algorithm detected numerous co-rated items, personalized levels of trust received priority in the weighting mechanism. If there were not enough co-rated items, group-level trust gained precedence in the prediction model (Lai, Liu & Lin, 2013, p.35). This approach proved superior to trust-based CF methods utilizing Resnick's prediction formula alongside the Pearson coefficient for filtering and generating predictions. They also examined the performance of models that emphasized profile- and item-level trust, rating-based trust, and relationship trust as generated between users with similar or top-to-bottom, senior-to-junior, relationships within the group (p. 36-45).

Jeong, Lee, and Cho examined the application of trust in CF systems. They called their approach a credit-based collaborative filtering method, but the idea is similar to trust-based methods mentioned above. While the systems described in O'Donovan and Smyth's approach relied on feedback from the user, Jeong, Lee and Cho examined similarity between users against a representative majority rating. Users sharing numerous similar ratings with the target user gained priority over the general consensus score. Traditional CF methods then calculated rating recommendations. But the results of Jeong, Lee and Cho's experiments illustrated that traditional memory-based CF systems outperformed their proposed credit-based system on numerous trial runs, with one variant of their system performing better than the traditional neighborhood model (Jeong, Lee & Cho, 2008, p. 7310-7312).

Finally, Bobadilla, Hernando, Ortega and Gutierrez looked at method similar to trust-based collaborative, but which they termed collaborative filtering based on significances. Instead of viewing trustworthiness in terms of users and their ratings, they viewed rating significance as the relevance of one user's ratings in relation to the items already rated by the user awaiting a rating prediction. Their results showed that when they compared to traditional neighborhood similarity metrics, the significance-based system outperformed in areas of precision and recall. However, the coverage area, which is the number of items or users that can be used to generate a rating prediction, proved smaller when compared to traditional memory-based results (Bobadilla, Hernando, Ortega & Gutierrez, 2011, p. 5-15).

Vector Space Modeling, Fuzzy Linguistic Modeling, and Other Novel Approaches to CF

The aforementioned studies examined time- and trust-based additions to CF techniques, but researchers have explored non-temporal and non-credit-based approaches in their efforts to improve prediction accuracy. These include fuzzy linguistic modeling and vector space modeling (Bellogin, Wang & Castells, 2011, p. 2260; Porcel, Lopez-Herrera & Herrera-Viedma, 2008, p. 5175-5176; Wang, Su, Gao & Ma, 2012, p. 1491). Wang, Su, Gao and Ma investigated vector space modeling as a method of storing data for collaborative filtering use. Traditional memory-based CF uses two lists, one for items and users, with users providing ratings for certain items. Vector space modeling stores variables within a vector, and while vector space modeling is used in information retrieval as a way of storing term frequency, or the number of times a term appears in a text document, Wang's team stored user information into a document of text which later relied on vector space modeling to generate recommendations. The user documents stored item preferences such as ratings, information applied to vectors in the term frequency scheme of vector space modeling. Their experimental tests showed that their vector space model CF system outperformed traditional memory-based systems, with improvement results ranging from approximately 4% to 7% on different trial runs (Wang, Su, Gao & Ma, 2012, p. 1488-1491).

Bellogin, Wang and Castells applied the vector space model to their CF work as well, but with some variations. They viewed users as a query. The information retrieval version of vector space modeling treats queries as a vector that is then applied to a document. Bellogin and his team treated users as a query in a similar fashion, but instead of search terms, the user query vector consisted of ratings for specific items. Bellogin's CF system showed significant improvement over traditional memory-based approaches (Bellogin, Wang & Castells, 2011, p. 2257-2260).

Porcel, Lopez-Herrera and Herrera-Viedma experimented with the Fuzzy Linguistic Model (FLM). Fuzzy linguistic modeling follows the principles of Fuzzy Set Theory, which looks at values on a scale of degrees rather than Boolean ones and zeroes. Porcel et al. looked into fuzzy linguistic modeling to build a recommender system for research featuring qualitative data. Their system utilized a content-based approach, which proved more effective due the qualitative nature of the research documents they had to sort through. Although their approach does not directly correlate with common techniques seen throughout collaborative filtering research, their examination of a multi-granular method of data examination and retrieval correlates with aspects of the proposed information-based neighborhood model (Porcel, Lopez-Herrera & Herrera-Viedma, 2008, p. 5175-5176).

Others have used different models in their approach to improving accuracy of collaborative filtering recommender systems. Chen and Chiang (2009) approached the issue using a system for constructing personal ontologies and methods for sorting and filtering through them to deliver recommendations (p. 323). Wang, Xie, and Fang (2011) applied the cloud model to item-based similarity metrics. The cloud model maps qualitative data to quantitative variables, and attempts to address the fuzziness and randomness of qualitative information (p. 18). Finally, Kamishima and Akaho (2010) approached the problem of improving accuracy by allowing users to sort through displayed recommendations based on their preferences. They named their CF approach the Nantonac model (p. 274).

The Cold-Start Problem

Researcher's attempts to improve accuracy have proven fruitful on numerous occasions, specifically in research examining vector space modeling as a viable addition or substitute to traditional CF methods (Bellogin, Wang & Castells, 2011, p. 2260; Wang, Su, Gao & Ma, 2012, p. 1491). However, they fail to address the cold-start problem, which is the condition met by the introduction of a new item or user in the system. This problem occurs because not enough data is available for traditional memory-based systems to generate accurate recommendations. A new customer might join Amazon, but since he or she hasn't purchased any items, and hasn't rated any items, there's not much data which a rating-based recommender system might generate recommendations (Lika, Kolomvatsos & Hadjiefthymiades, 2014, p. 2065-2066).

Research that addresses cold-start conditions vary in approach, but the trend seems to focus on extracting information and data from sources other than explicit user ratings. These other sources of data include demographic information, which then become variables for algorithms to calculate user similarity to generate recommendation predictions (Chekkai et al., 2012, p. 760; Lika et al., 2013, p. 2067-2072; Moreno et al., 2011, p. 256-257; Zhang et al., 2010, p. 1-2). Eckhardt (2012) proposed a system that combines content-based filtering with collaborative filtering techniques. Content-based filtering differs from collaborative filtering in which products and users are compared to one another based on taxonomies and ontologies concerning users and items. Eckhardt combined a user preference model with collaborative filtering in order to alleviate the cold-start problem. Data in user preference models can be entered either explicitly or implicitly via input from the user or user activity. Input forms provided the bulk of explicit data entry, while item browsing and purchasing activity consisted of much of the implicit data that entered the preference models utilized by Eckhardt's system (p. 11511-11512).

Others used certain features of the World Wide Web to provide data about users. Moreno et al. (2011) applied data mining algorithms to mine through semantic web information, gathering data concerning users to generate similarities for comparisons in recommendations (p. 256-257). Zhang et al. (2010) gathered data on users via social tagging, which allowed them to generate three lists: a list of users, a list of tags, and a list of items. These formed the basis of their calculations and experiments (p. 1-2).

Lika et al. (2013) applied demographic information to algorithms that generated similarity scores between a set of new users. Their methods relied on classification algorithms, which helped establish the characteristics of each new user. They surmised that users with similar backgrounds would share similar preferences. They tested their system using three classification algorithms: the C4.5 algorithm, the Naïve Bayes, and a random classification algorithm (RCA). Random classification formed their baseline for results, and both the C4.5 and Naïve Bayes outperformed their baseline. However, they adopted two variations to the C4.5 algorithm. The first limited the algorithm to two classes, the $C^2$4.5. The second allowed multiple classes, the C^M4.5. The $C^2$4.5 classifier outperformed the C^M4.5, while the Naïve Bayes presented mixed results, yet outperformed their baseline (p. 2067-2072).

The aforementioned studies relied on demographic data. Chekkai, Chikhi, and Kheddouci (2012) approached the problem in a different manner. They utilized social graphs to structure the data and its relationships, not focusing on data extraction methods and classification. In their graph model, users or items formed nodes of a graph. The graph's edges illustrated the degree of similarity between users and items. To circumvent cold-start conditions, Chekkai et al. identified critical nodes, that is, nodes whose removal would fragment the graph. These critical nodes, they claim, served as mediators, which helped identify user preferences based on sparse data (p. 760-762).

Part III: Information-Based Neighborhood Modeling

Returning to the previous example of people providing recommendations in the context of social situations, memory-based CF resembles groups of people, or neighborhoods, who share similar interests. Recommendations stem from the most similar neighbors. While this may seem like a simple approach to discovering which movie to see next, it is not always accurate, since movies differ from each other, even within the same genre. But considering the amount of work a person would have to do in order to find the next best thing to see, given all the steps associated with model-based approaches, one can easily conclude that a leeway between memory-based and model-based methods would prove best in increasing recommendation accuracy while alleviating cold-start problems.

Memory-based approaches are easy to implement. New data can be easily added and incremented, and there is no need to examine or factor in the content of items being recommended in memory-based systems. However, memory-based approaches are dependent on human ratings of items, and performance decreases when data is sparse, meaning that new users or items receive faulty or no recommendations until users rate items or items receive ratings. Memory-based approaches also suffer from scalability issues. They do not scale well when moving to larger datasets. Model-based CF addresses sparsity and scalability issues, and they show improved prediction performance over typical neighborhood models. But model-based CF requires expensive model-building, and a trade-off exists between predication performance and scalability (Su & Khoshgoftaar, 2009, p. 3).

Hybrid approaches attempt to remedy the flaws of memory- and model-based approaches while adhering to their advantages. There are two types of hybrid recommender systems. The first merges content-based and collaborative filtering approaches. This technique in solving or reducing the limitations found in content-based or CF systems, but of particular interest are the hybrid systems that merge memory-based CF techniques with those that are model-based. Such systems benefit from improved prediction performance and reduced sparsity problems (p. 3).

The Information-Based Neighborhood Model relies on a memory-based approach. It can be considered a hybrid system in the sense that it applies an additional model for generating the data that goes into the K-Nearest Neighbor algorithm. Whereas the K-NN algorithm focuses on lists of users, items and their ratings, the Information-Based Neighborhood model supplements, if not overtly substitutes, the K-NN model with data other than users, items and their ratings. Instead of users, items and their ratings, the Information-Based Neighborhood model would generate neighborhoods and examine their similarities via other criteria, such age ranges, release date of items, production location of items, prices ranges, and user location, among other sources of data. Although the I-BN model adds an extra layer of complexity to the K-NN model, it does so without complex sets of algorithms. Theoretically, it would preserve the faster runtimes and the ability to easily add to users or items to the system, benefits found in memory-based models (Su & Khoshgoftaar, 2009, p. 3). It would also address scalability issues with its ability to focus on specific pieces of information, which resembles the Maximum Margin Matrix Factorization technique of allowing expanding dataset sizes by focusing on key data for its calculations (Barbieri, Manco, Ritacco, 2014, p. 20). The I-BN model furthermore has the potential to alleviate cold-start problems, since, as seen in the literature review, researchers who worked with demographic data helped solve sparsity issues apparent in cold-start conditions (Moreno et al., 2011, p. 256-257; Zhang et al., 2010, p. 1; Lika et al., 2013, p. 2067). The difference between the I-BN model over the K-NN model is that it'll rely on Ackoff's definition of information to generate the data required for the comparison matrices.

In the mid-twentieth century a new field of study emerged. The field was Information Science, a discipline used to evaluate data and information for management purposes. One of the field's proponents and scholars was Ackoff. In 1988 he proposed a hierarchy which he used to describe the flow and transition of data into information, knowledge and wisdom. This hierarchy is called the Data, Information, Knowledge and Wisdom (DIKW) hierarchy. Ackoff defines each level of the hierarchy, starting with data. He labels data as the raw buildings blocks of information. Information is data structured around the five key question words, "what, where, who, when, and how many (1989, p. 3)."

Traditional recommender systems rely on lists of items and users. Each user rates or gives a score to a purchased or viewed item. This generates a comparison matrix for use in predicting similarity amongst users or items. However, it is also possible to use data other than user ratings for certain items. Demographic information, along with item information and ratings, has the potential of augmenting the K-NN model's capabilities while limiting its drawbacks. Ackoff's definition of information serves as a viable way for generating alternative data for the K-NN model (Nilashi et al., 2012, p. 4169-4175).

Table 6 is a sample matrix filled with data pertaining to users, items and their ratings in a traditional approach to the neighborhood model:

Table 6

Traditional Neighborhood Model Sample Matrix Data

	Item A	Item B	Item C
User A	5	2	2
User B	4	2	1

User C	1	5	5

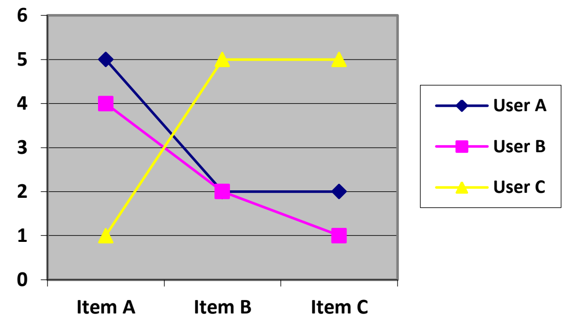

Figure 1. Line Graph of Table 6 Data

Figure 1 shows a visual representation of the similarity between three users based on ratings for items A, B, and C. According to the data, users A and B are more similar. Products A rates highly would most likely be recommended to user B, and vice-versa. Data for the matrix, however, does not have to limit itself to users, items and their ratings. Ackoff's definition of information, which classifies data according to the question words "what, where, when, who, and how many," provides a wide range of data possibilities (Ackoff, 1989, p. 3). To illustrate this point, consider which question words the terms "users, items, ratings" answer when establishing the data in the comparison matrix. Users represent people, while the question word "who" asks about an unknown person. Items represent a tangible product, while the question word "what" asks about an unknown object. Finally, ratings represent a number, usually between 1 and 5, while the question word "how many" asks about an unknown quantity. Therefore, "what" answers questions about objects, "who" answers questions about people, "where" answers questions about location, "when" answers questions about time, and "how many" answers questions about quantity.

Table 7 illustrates a representation of the user-to-item rating matrix according to the placement of users, items, and ratings within the five key question word categories associated with Ackoff's definition of information:

Table 7

Data Possibilities in User-to-Item Rating Matrix Using Information-Based Terms

	(What?) Item A	(What?) Item B	(What?) Item C
(Who?) User A	(How Many?) 1-5/5	(How Many?) 1-5/5	(How Many?) 1-5/5
(Who?) User B	(How Many?) 1-5/5	(How Many?) 1-5/5	(How Many?) 1-5/5
(Who?) User C	(How Many?) 1-5/5	(How Many?) 1-5/5	(How Many?) 1-5/5

With this representation it becomes possible to generate neighborhood comparisons between different types of data, not just users, items, and the ratings between them. As illustrated in the works of previous researchers investigating the cold-start problem, demographic information helped solve data sparsity issues in recommender systems (Moreno et al., 2011, p. 256-257; Zhang et al., 2010, p. 1; Lika et al., 2013, p. 2067). Demographic information provides details on a user of the system. Consider a customer on Amazon or Netflix. Each customer has a real name and a username. Associated with each user is an address. Addresses answer the "where" questions tied to location. Users lies within certain age ranges and two physical sexes or gender, with age providing a number that answers a "how many" question tied to quantity, and physical sex answering a "what" question tied to physical aspect of a human being (Ackoff, 1989, p. 3).

Items also carry identifying information. Time of purchase, place of production, and national origin of product provide details that may be tied to sales numbers which traditional neighborhood models might fail to associate with increased profit. Certain seasons may see higher sales, such as Christmas, the New Year, and national holidays. Products from certain nations may be popular with certain people, and products from certain time periods may likewise be popular with people in certain age groups. The possibilities may be endless, and though this paper limits itself in examining the viability of the I-BN model, machine learning and data mining techniques would help in generating lists and data for inclusion in the comparison matrices.

Table 8 represents a matrix following the I-BN labeling approaching, where data for age groups answers the how many question, data for products answers the what question, and data from ratings answers the how many question.

Table 8

Information-Based Representation of Data in a User Age Range to Item Rating Matrix

	(What?) Item A	(What?) Item B	(What?) Item C
(How Many?) Age Range A	(How Many?) Rating 1-5/5	(How Many?) Rating 1-5/5	(How Many?) Rating 1-5/5
(How Many?) Age Range B	(How Many?) Rating 1-5/5	(How Many?) Rating 1-5/5	(How Many?) Rating 1-5/5
(How Many?) Age Range C	(How Many?) Rating 1-5/5	(How Many?) Rating 1-5/5	(How Many?) Rating 1-5/5

Table 9 and Table 10 serve as two additional examples of comparison matrices following the I-BN labeling approach:

Table 9

User Ratings based on Location

	(Where?) Location A	(Where?) Location B	(Where?) Location C
(Who?) User A	(How Many?) 1-5/5	(How Many?) 1-5/5	(How Many?) 1-5/5
(Who?) User B	(How Many?) 1-5/5	(How Many?) 1-5/5	(How Many?) 1-5/5
(Who?) User C	(How Many?) 1-5/5	(How Many?) 1-5/5	(How Many?) 1-5/5

Table 10

Number of Item Purchases based on Month

	(When?) Month 1-12	(When?) Month 1-12	(When?) Month 1-12
(What?) Item A	(How Many?) >= 0	(How Many?) >= 0	(How Many?) >= 0
(What?) Item B	(How Many?) >= 0	(How Many?) >= 0	(How Many?) >= 0
(What?) Item C	(How Many?) >= 0	(How Many?) >= 0	(How Many?) >= 0

Table 11 illustrates a matrix with rating data for three items from three different age groups:

Table 11

Item Rating Based on Age Range with Sample Data

	Item A	Item B	Item C
Teens	5	3	1
Adults	4	2	1
Seniors	3	1	5

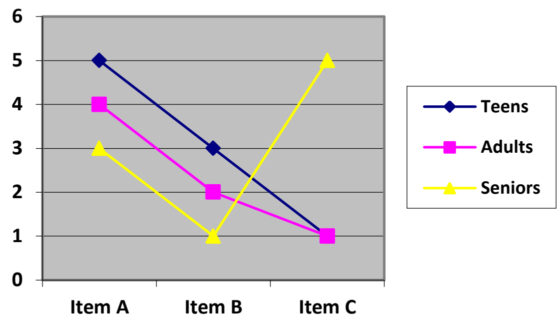

Figure 2. Line Graph of Table 11 Data

Comparison graphs, such as Figure 2, help establish relationships between the various different users, which helps alleviate the cold-start problem. New teenaged users receive recommendations based on purchase and rating acting from other teenagers. The same goes for adults and senior adults.

Another topic of importance is the level of granularity concerning the number of variables and the specificity of information when it comes to making recommendations. Porcel et al. examined Fuzzy Linguistic Modeling and its application to recommendations relying on qualitative research data. He examined tuple sets and multi-granular FLM when considering his FLM-based recommendation approach (2009, p. 5175-5176). Likewise, the Information-Based Neighborhood model may benefit from lists of tuples and multi-granular items. The terms teenaged boys and teenaged girls specifies certain users within a range of users. The term teenaged identifies the user as someone within the age range of 13-19, and boy and girl identifies the user as someone who is either male or female. The question words (how many) and (what) would form the descriptive tuple that identifies the users that form either the column or the rows of the comparison matrix.

Table 12 is a sample matrix example illustrating preferences among teenage boys and girls:

Table 12

Item Ratings to Age Range and Gender Matrix

	Item A	Item B	Item C
(how many years, what physical sex) Teenage Boys	5	3	1
(how many years, what physical sex) Teenage Girls	1	3	5

Only the rows of the matrix consists of tuple items, but theoretically rows and columns, as well as cells of the matrix, could consist of items with varying degrees of granularity. It is entirely possible that the higher the level of specificity, the more accurate the predictions in either normal data conditions or in cold-start conditions. High levels of specificity might also be problematic when not enough data is present in the system. In order to determine the effectiveness of the I-BN model, in terms of improving accuracy and alleviating cold-start problems, an experiment consisting of a recommender system running on libraries with machine-learning and memory-based collaborative filtering algorithms, as well filtering functions that sort through data following the concepts behind the I-BN model, follows in the section below (Ackoff, 1989, p.3).

Part IV: Experimental Analysis

Three testing scenarios evaluate the performance of recommender system methods and their engines. These include offline testing, online testing, and use-case testing (Shani & Gunawardana, 2011, pp. 261-267).

Offline Testing

Offline testing implements metrics meant to evaluate the performance of recommender systems along with machine learning algorithms and procedures. An offline test examines data from a given dataset, and may not necessarily need participants to provide experience and usage information. This is beneficial when there is no need to examine the human component of the system. But if human-computer interaction is necessary to determine the effectiveness of the system, one of the two testing procedures below, either on-line testing or use-case scenario testing, would prove more effective than off-line testing (p. 261).

Online Testing

Online testing studies real-time usage of a system, with users providing feedback in the form of questionnaires, usage habits, and numbers that represent increased or decreased sales, time spent on interface, among other quantitative data. Online testing allows users to come and go, making recruitment and implementation easy. It's an easy and cost-effective way of gathering quantitative data, but it is limited in its ability to get qualitative data, specific information about a specific user's experience with the system. For a more qualitative examination of a user's interaction with the system, a use-case scenario test would be more helpful (p. 266-267).

Use-Case Scenario Testing

Use-Case testing explores in-depth the experience of select users of an interface or system. Use-Case studies provide researchers with qualitative data to can explore the likes and dislikes of a user and examine why they found certain recommendations valuable or not valuable, as well as reasons for continuing usage of a website, or other application, utilizing the recommender system (p. 263-264).

Evaluating the Information-Based Neighborhood Model

The proposed test of the Information-Based Neighborhood model examines the efficiency of the system in normal and cold-start conditions when compared to the traditional K-Nearest Neighbor model. Therefore, the test does not require an interface and user interaction. Since online and use-case testing serve best the scenarios where user interaction is present, the test of the I-BN model runs offline following offline testing procedures. Below is the procedure for the I-BN model test.

Procedure

Offline testing requires a data source, and the I-BN test utilizes Grouplens' Movielens datasets. The 100k dataset and the 1m dataset allows for an examination of the scalability of the I-BN model, while a number of test scenarios examine the functionality of the system in two key settings: a normal data distribution setting, for examination of accuracy differences between the I-BN model and the K-NN model, and a distribution of data resembling cold-start conditions for examination of cold-start problem alleviation. The test examines accuracy differences under normal data conditions, and starts with lists containing single variable elements. The tests run on both the 100k and 100m datasets. The single variable element variable test precedes multi-granular tests with the same dataset and under the same data conditions. In total, offline evaluation tests run six times for both normal data distribution and cold-start condition cases, totaling twelve test runs ("MovieLens," n.d.).

Below are the setups for each test, along with a description of the Movielens datasets and a description of the system setup and evaluation metrics used, and the observed limitations of the system and testing conditions:

A few conditions must be observed prior to analyzing the performance of a traditional neighborhood model in comparison with an information-based model. The first is that neighborhood models follow one of two orientations, user-based or item-based. The second is the level of granularity. The number of demographic attributes can affect the results of the information-based neighborhood model system, and testing varying numbers of attributes can account for level of granularity and system performance. The final factor is the size of the dataset, which helps determine consistency of results, or whether larger or smaller datasets affect overall performance (Lu et al., 2012, p. 10; Nilashi et al., 2012, p. 4169-4175).

Lists of Accuracy and Performance Tests

Single Variable Element Test

The single variable element test follows K-NN user-orientation, and examines the performance of the proposed I-BN model in regards to single variables forming the rows of the comparison matrix. Traditional neighborhood models structure the comparison matrix with lists of users and items. For the I-BN model, ratings for items remain the same, but user information becomes much more specific with each subsequent test. Rather than examine every user in the system, the single variable element test limits the list of users to users matching certain criteria. For each of the tests, in both normal and cold-start conditions, three scenarios examine the flexibility of the I-BN system. And each of the three scenarios adheres to some proponent of Ackoff's definition of information as it relates to data, meaning one of the five key question words addresses a different aspect of the users forming the list of users. Each test generates a rating and recommendation for one user, and the table below contains the age, gender, and occupation data for users with ID 1 from the 100k and 1m datasets:

Table 13

Age, Gender, and Occupation Data for User with ID 1 in 100k and 1M Datasets

Dataset	User ID	Age	Gender	Occupation
100k	1	24	Male	Technician
1M	1	Under 18	Female	Student

Age, gender and occupation provide the I-BN model with the three test cases for the single variable element test. One test generates a recommendation for User 1 based on age range, the other based on gender, and the last based on occupation. In essence, the system searches for ratings from users with the same age range, gender, or occupation as User 1 in the 100k and 1m datasets. A traditional K-NN model runs alongside these tests, and below the results for the three test cases stands the label “Generic,” which accounts for the performance of the traditional K-NN model (“MovieLens,” n.d.).

<u>Two Variable Element Tests</u>

Following the single variable element test is the two variable element test. Whereas the single variable element test populated the list of users with one piece of identifying information from User 1, the two variable element test produces a list of users matching two identifying criteria from User 1. The three test variations are as follows: age and gender, gender and occupation, age and occupation. This follows allows tuples to populate the user list, and examines how granularity affects the overall performance of the I-BN model (“MovieLens,” n.d.).

<u>Three Variable Element Test</u>

A single test case examines the performance of the three variable element test, as opposed to three in the single and two variable element tests. The user list for the test case consists of users who share the same age group, gender, and occupation as User 1 from the 100k and 1m datasets (“MovieLens,” n.d.).

<u>Details of User Information</u>

Since the 100k dataset contains around 100,000 entries, it'll be impossible to list every piece of information from that dataset in this paper. More so for the 1m dataset, which contains around 1,000,000 entries. Of importance to this study is the user and item data. Both datasets contain demographic information for each user. User occupation for the 100k dataset includes: administrator, artist, doctor, educator, engineer, entertainment, executive, healthcare, homemaker, lawyer, librarian, marketing, none, other, programmer, retired, salesman, scientist, student, technician, writer. For the 1m dataset, user occupations include: other or not specified, academic/educator, artist, clerical/admin, college/grad student, customer service, doctor/health care, executive/managerial, farmer, homemaker, K-12 student, lawyer, programmer, retired, sales/marketing, scientist, self-employed, technician/engineer, tradesman/craftsman, unemployed, writer. The 100k dataset provides a fixed age for each user, but the 1m dataset affixes an age range for each user. To allow the system to account for age ranges on the 100k dataset, an algorithm establishes age ranges similar to those found in the 1m dataset, which includes: Under 18, 18-24, 25-34, 35-44, 45-49, 50-55, 56 and over. The human species consists of two physical sexes, which both datasets label as the male or female gender. Appendix D contains a table with data for the occupations and age ranges from the MovieLens datasets ("MovieLens," n.d.).

Testing Under Cold-Start Conditions

Testing under cold-start conditions follows the same setup as the I-BN accuracy tests, but with one major exception: to account for limited available data, an extra user was created and inserted into copies of the 100k and 1m MovieLens datasets. Similar to the accuracy tests, three test cases examine the performance of the I-BN model under reduced availability of information for the new users. Table 14 contains age, gender, and occupation data for the new users, User 944 for the 100k dataset, and User 6041 for the 1m dataset:

Table 14

Age, Gender, and Occupation Data for Users 944 (100k Dataset) and 6041 (1M Dataset)

Dataset	User ID	Age	Gender	Occupation
100k	944	32	Male	Student
1M	6041	25-34	Male	Graduate Student

The three test cases examine cold-start conditions where data is limited to five available movie ratings, then three, and finally two. Table 15 showcases the ratings, the movies, and the dataset associated users 944 and 6041:

Table 15

Ratings from Users 944 (100k Dataset) and 6041 (1M Dataset) for Five Different Movies

Dataset	User ID	Movie Name	Rating
100k	944	Jumanji	5
100k	944	Seven	2
100k	944	Toy Story	1
100k	944	Star Wars	4
100k	944	Pulp Fiction	3
1M	6041	Jumanji	5
1M	6041	Saving Private Ryan	3
1M	6041	Seven	2
1M	6041	Dumb & Dumber	5
1M	6041	Species	2

System Setup and Limitations

The I-BN test system was coded in Java using libraries from the Apache Mahout machine learning and data mining framework. In particular, the collaborative filtering neighborhood model and comparison metrics constituted the bulk of the recommendation and evaluation process. Aside from running a generic K-NN model, a series of algorithms ran a setup similar to the proposed I-BN model. The algorithms read the data from the 100k and 1m datasets and filtered the user lists according to the criteria established above. In the end, the system ran memory-based tests where the nearest neighbor lists in the I-BN configuration consisted of users with matching age ranges, gender, and occupations, or combinations and variations thereof, of

User 1 in the accuracy tests, and Users 944 and 6041 in the cold-start tests ("Apache Mahout," n.d.).

Mahout is not without its limitations though. The CF and K-NN algorithms available for item-oriented recommendation proved ineffective during test runs of the system. This could be due to the provided algorithms conflicting with the I-BN setup. It could also be that the developers at the Apache Foundation, responsible for the creation of the Mahout framework, chose to provide a limited number of item-oriented memory-based CF algorithms. As a result, the tests were limited to user-orientation, and the whole range of possible test scenarios, with multi-variable elements forming both the lists of users and items, remains to be tested ("Apache Mahout," n.d.).

Due to the limitations of Apache Mahout, and its reluctance to interface with the setup of the I-BN system, the offline testing procedure ran in a closed testing environment, where both the I-BN setup and the generic K-NN model ran under similar conditions and with similar evaluation metrics ("Apache Mahout," n.d.).

Part V: Results

The Apache Mahout framework provided two evaluation metrics that helped determine the accuracy of rating and predictions: the Absolute Average Difference Evaluator, and the Root Square Means Evaluator. These two metrics provided prediction scores for each test. Prediction scores consist of a number between 0.0 and 1.0, with numbers closer to 0.0 reflecting a higher accuracy over those closest or beyond 1.0 (Owen, Anil, Dunning & Friedman, 2012, p. 20-21; Bejoy, 2011, para. 4-5).

Precision and recall scores stem from the field of information extraction and retrieval, and evaluate accuracy in terms of relevancy of retrieved elements. Precision is the fraction of the subset of relevant elements from the retrieved elements over all retrieved elements, while recall is the fraction of relevant elements from the retrieved set over all relevant elements. If even numbers from 0-200 reflect all relevant numbers, and the numbers 50-150 reflect all retrieved numbers, then precision would equal the number of even numbers in the set of numbers 50-150 over the number of numbers in the set of 50-150. There are 50 even numbers in the set of numbers 50-150, and a total of 100 numbers in the set of numbers 50-150, making precision equal to 50/100, which is 0.5. Recall would be 0.5, since the top portion of the fraction matches the precision equation, while the number of even numbers in the set 0-200 is 100, making the fraction 50/100, which is 0.5. High precision and recall score means more relevant results, which means a higher likelihood of the recommendation being accurate (Owen et al., 2012, p. 21).

<u>Accuracy Test Results</u>

Table 16

Results of Accuracy Test 1a: Single Element User-Based 100k

RS Type	Recommendation	Prediction Score	Precision	Recall
Info-Based Male Users	Item ID: 748 (The Saint)	0.9123232979928291	0.008670520231213865	0.006802721088435376
Info-Based 20-30 Year Old Users	Item ID 313 (Titanic)	0.8622032926323706	0.018099547511312222	0.014344262295081971
Info-Based Technician Users	N/A	1.1100707203149798	0.0	0.0
Generic User-Based	Item ID: 748 (The Saint)	0.9468946771418792	0.008620689655172415	0.006345177664974617

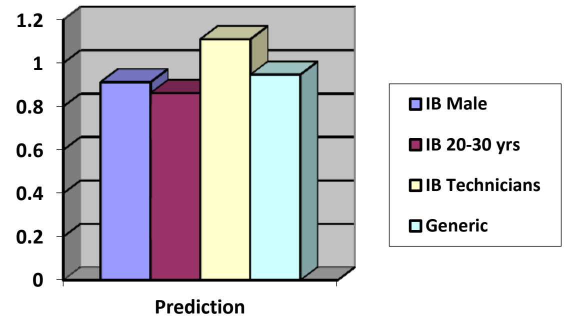

Figure 3. Acc. Test 1a Prediction Scores

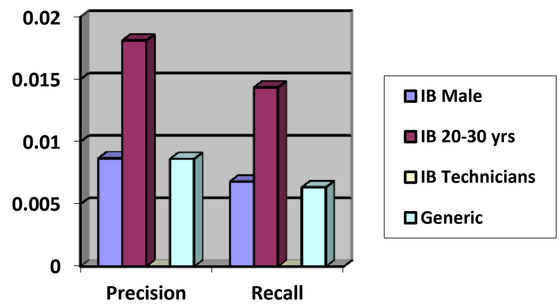

Figure 4. Acc. Test 1a Precision Recall

For this first test the Information-Based test of age-range from twenty to thirty years performed better in both prediction scores and precision and recall scores, followed by Information-Based gender scores, which happened to evenly match the results of the generic neighborhood model. The Information-Based occupation score perhaps underperformed due to the low list of results, meaning that few users were technicians and this resulted in few user comparisons, causing difficulty in processing predictions and evaluations. This is apparent in the precision and recall scores for the Information-Based occupation test run.

Table 17

Results of Accuracy Test 1b: Single Element User-Based 1m

RS Type	Recommendation	Prediction Score	Precision	Recall
Info-	Item ID: 2581	0.95070422535211	0.0117004680187207	0.0086633663366336

Based Female Users	(Never Been Kissed)	2	37	64
Info-Based Under 18 Users	Item ID: 110 (Braveheart)	1.1133586502075195	0.010040160642570281	0.004411764705882353
Info-Based K-12 Student Users	Item ID: 2571 (The Matrix)	0.8935115744427932	0.009389671361502342	0.005136986301369862
Generic User-Based	Item ID: 2581 (Never Been Kissed)	0.8839017169343092	0.01017847183491356	0.00705128205128205 1

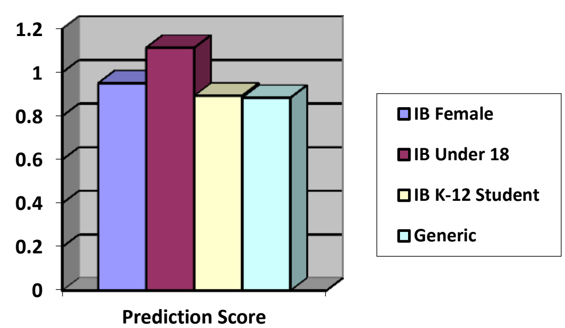

Figure 5. Acc. Test 1b Prediction Scores

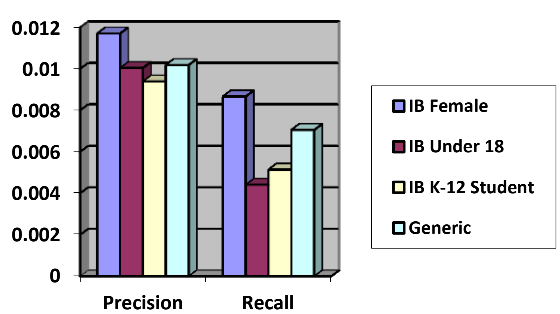

Figure 6. Acc. Test 1b Precision Recall

As opposed to the 100k ratings dataset, the one million ratings dataset shows that in precision and recall the Information-Based test for female users outperformed the generic user-based neighborhood model. This suggests that there might be a slight improvement as the system scales upward, but not enough to formulate a definitive conclusion on the matter.

Table 18

Results of Accuracy Test 2a: Two Elements User-Based 100k

RS Type	Recommendation	Prediction Score	Precision	Recall
Info-Based 20-30 Year Old Male Users	Item ID: 313 (Titanic)	1.006771409511566	0.020467836257309944	0.01804123711340206
Info-Based Male, Technician Users	Item ID 294 (Liar Liar)	0.8121258517106373	0.0	0.0
Info-Based 20-30 Year Old Technician Users	Item ID 294 (Liar Liar)	0.3930600711277552	0.0	0.0
Generic User-Based	Item ID: 748 (The Saint)	0.9468946771418792	0.008620689655172415	0.006345177664974617

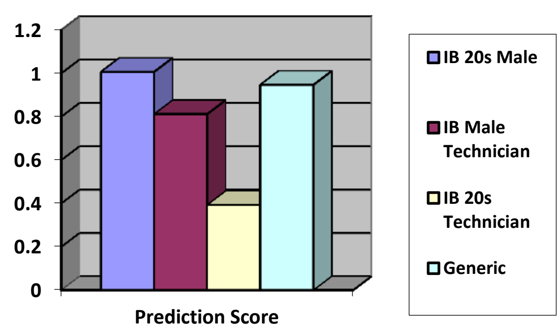

Figure 7. Acc. Test 2a Prediction Scores

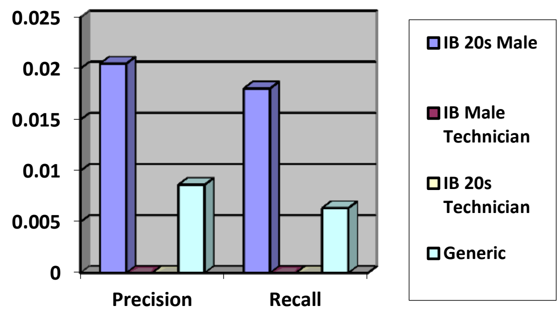

Figure 8. Acc. Test 2a Precision Recall

For the one hundred thousand ratings dataset male technician users and technician users in their twenties outperformed in prediction score results while precision and recall scores, though males in their twenties scored high points, results were inconclusive for the other Information-Based variables due to the smaller dataset size.

Table 19

Results of Accuracy Test 2b: Two Elements User-Based 1m

RS Type	Recommendation	Prediction Score	Precision	Recall
Info-Based Female, Under 18 Users	Item ID: 110 (Braveheart)	0.9088282151655718	0.0142857142857142 84	0.0107142857142857 14
Info-Based Female, K-12 Student Users	Item ID: 110 (Braveheart)	1.0248437523841858	0.0057471264367816 07	0.00423728813559322
Info-Based Under 18, K-12 Student Users	Item ID: 110 (Braveheart)	0.8518518518518516	0.0084745762711864 39	0.0039370078740157 445

Generic User-Based	Item ID: 2581 (Never Been Kissed)	0.8839017169343092	0.01017847183491356	0.007051282051282051

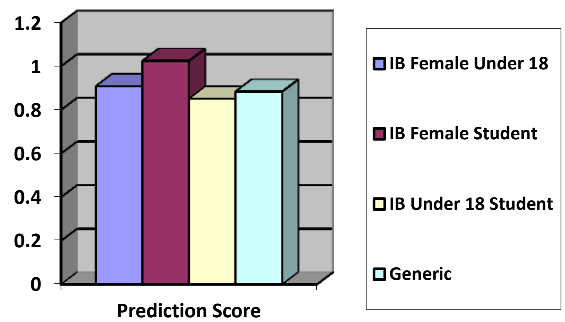

Figure 9. Acc. Test 2b Prediction Scores

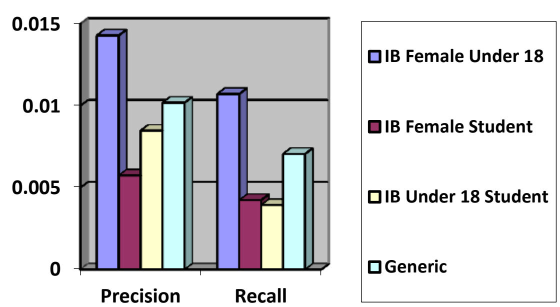

Figure 10. Acc. Test 2b Precision Recall

The only conclusive result for this test was the high precision and recall scores for female users under eighteen years of age. In the prediction score range the results were either too close, or higher than the generic user-based neighborhood model to deem that the Information-Based approach was more accurate in its predictions.

Table 20

Results of Accuracy Test 3a: Three Elements User-Based 100k

RS Type	Recommendation	Prediction Score	Precision	Recall
Info-Based Male, 20s,	Item ID: 294 (Liar Liar)	0.3930600711277553	0.0	0.0

Technician User				
Generic User-Based	Item ID: 748 (The Saint)	0.9468946771418792	0.008620689655172415	0.006345177664974617

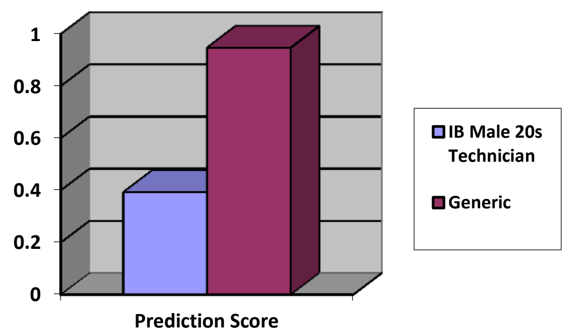

Figure 11. Acc. Test 3a Prediction Scores

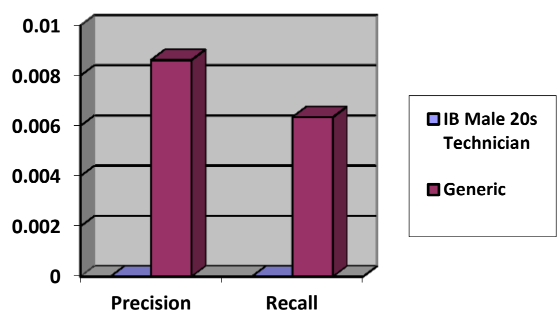

Figure 12. Acc. Test 3a Precision Recall

The Information-Based prediction score outperformed the generic user-based neighborhood approach, but as before, the smaller dataset size struggled providing results for precision and recall scores.

Table 21

Results of Accuracy Test 3b: Three Elements User-Based 1m

RS Type	Recommendation	Prediction Score	Precision	Recall
Info-Based Female, Under	Item ID: 110 (Braveheart)	1.2352941176470587	0.011627906976744182	0.008695652173913044

18, K-12 Student User				
Generic User-Based	Item ID: 2581 (Never Been Kissed)	0.8839017169343092	0.01017847183491356	0.0070512820512820 51

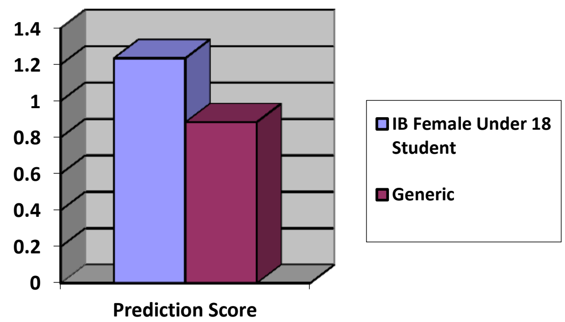

Figure 13. Acc. Test 3b Prediction Scores

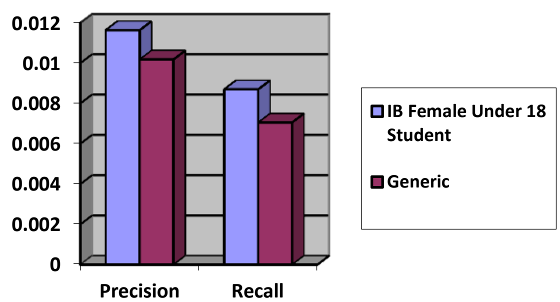

Figure 14. Acc. Test 3b Precision Recall

The larger dataset showed that precision and recall scores in the Information-Based approach outperformed the generic user-based model.

Cold-Start Test Results

Table 22

Results of Cold-Start Test 1a: Single Element User-Based 100k

RS Type	Recommendation	Prediction Score	Precision	Recall
Info-Based Male	Item ID: 127 (The Godfather)	0.9150190626809354	0.0089641434262948 11	0.0067911714770797 98

User Five Ratings				
Info-Based 30s User Five Ratings	Item ID: 96 (Terminator 2: Judgment Day)	0.9738020673394205	0.012727272727272733	0.009404388714733545
Info-Based Student User Five Ratings	Item ID: 258 (Contact)	0.7688842146020187	0.013812154696132596	0.011061946902654864
Info-Based Male User Three Ratings	Item ID: 9 (Dead Man Walking)	0.9123232979928291	0.008806262230919768	0.006802721088435376
Info-Based 30s User Three Ratings	Item ID: 9 (Dead Man Walking)	0.9689175099441684	0.012820512820512827	0.009433962264150945
Info-Based Student Three Ratings	Item ID: 9 (Dead Man Walking)	0.7637486533114782	0.013888888888888888	0.011111111111111106
Info-Based Male User Two Ratings	Item ID: 50 (Star Wars)	0.9123232979928297	0.0087040618955512 5	0.006802721088435376
Info-	Item ID: 50	0.96891750994416	0.0124113475177305	0.0094339622641509

Based 30s User Two Ratings	(Star Wars)	88	02	45
Info-Based Student User Two Ratings	Item ID: 50 (Star Wars)	0.7637486533114785	0.013440860215053764	0.0111111111111111106
Generic User-Based	Item ID: 127 (The Godfather)	0.946894677141879	0.0087463556851131192	0.006337135614702153

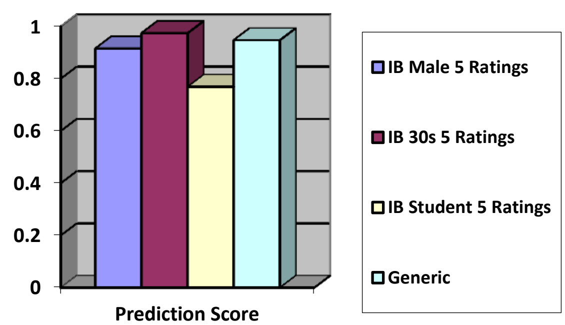

Figure 15. CS Test 1a Prediction Scores (Five Ratings)

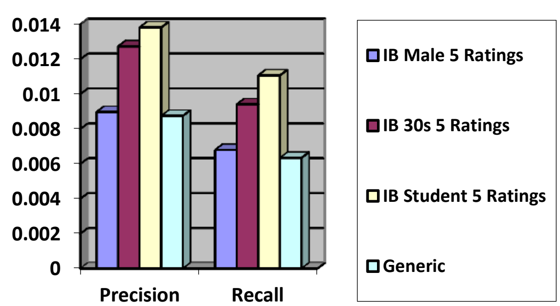

Figure 16. CS Test 1a Precision Recall (Five Ratings)

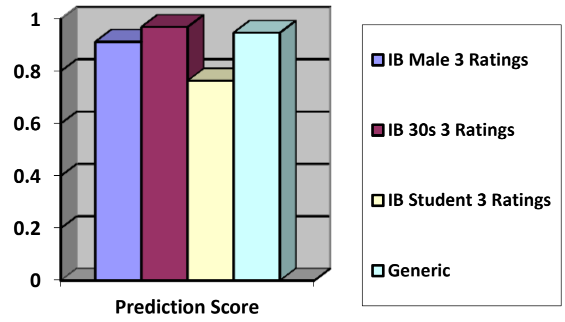

Figure 17. CS Test 1a Prediction Scores (Three Ratings)

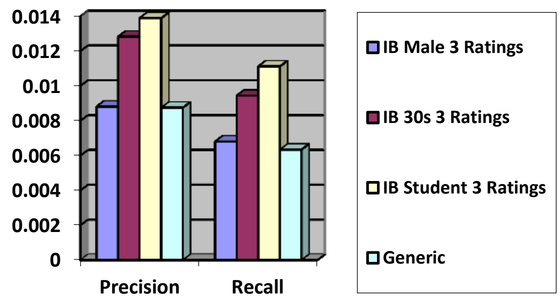

Figure 18. CS Test 1a Precision Recall (Three Ratings)

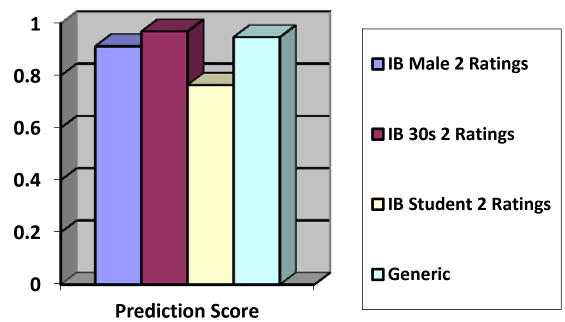

Figure 19. CS Test 1a Prediction Scores (Two Ratings)

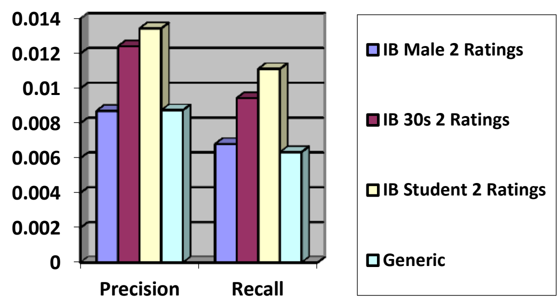

Figure 20. CS Test 1a Precision Recall (Two Ratings)

30 year old users, regardless of the number of ratings, and precision and recall scores across the board, outperformed the generic user-based models. The other results did not significantly outperform.

Table 23

Results of Cold-Start Test 1b: Single Element User-Based 1m

RS Type	Recommendation	Prediction Score	Precision	Recall
Info-Based Male User Five Ratings	Item ID: 1266 (Unforgiven)	0.8813922488800819	0.009900990099009927	0.006843455945252349
Info-Based 25-34 Years User Five Ratings	Item ID: 1266 (Unforgiven)	0.8993963782696178	0.009097035040431271	0.006545820745216518
Info-Based College/Graduate Student User Five Ratings	Item ID: 2571 (The Matrix)	0.9802226777625296	0.013687600644122389	0.008581235697940502
Info-Based Male User Three Ratings	Item ID: 1968 (The Breakfast Club)	0.8803637865126619	0.00991609458428683	0.006845407872219049
Info-Based 25-34 Years User Three Ratings	Item ID: 1968 (The Breakfast Club)	0.901803607214429	0.009121621621621634	0.0065491183879093215
Info-Based College/Graduate Student	Item ID: 2918 (Ferris Bueller's Day	0.9756118954552537	0.013709677419354844	0.008591065292096219

User Three Ratings	Off)			
Info-Based Male User Two Ratings	Item ID: 110 (Braveheart)	0.8803637865126609	0.009885931558935385	0.006845407872219049
Info-Based 25-34 Years User Two Ratings	Item ID: 110 (Braveheart)	0.901803607214429	0.009042196918955126	0.0065491183879093215
Info-Based College/Graduate Student User Two Ratings	Item ID: 110 (Braveheart)	0.9845007843441431	0.013535031847133755	0.008591065292096219
Generic User-Based	Item ID: 1266 (Unforgiven)	0.8839017169343101	0.010195530726256999	0.007049775688955351

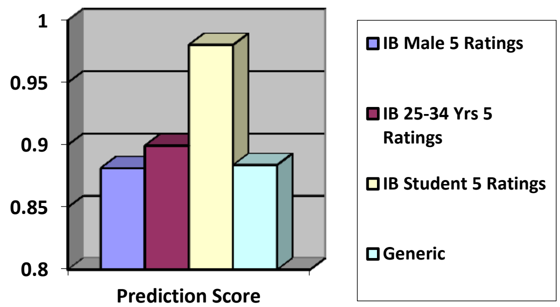

Figure 21. CS Test 1b Prediction Scores (Five Ratings)

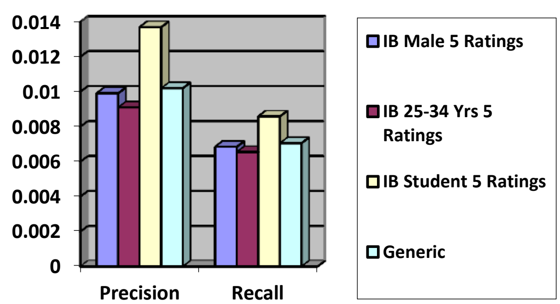

Figure 22. CS Test 1b Precision Recall (Five Ratings)

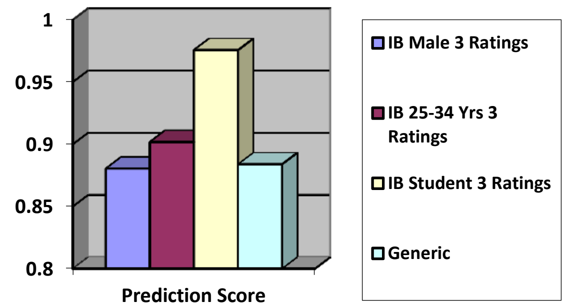

Figure 23. CS Test 1b Prediction Scores (Three Ratings)

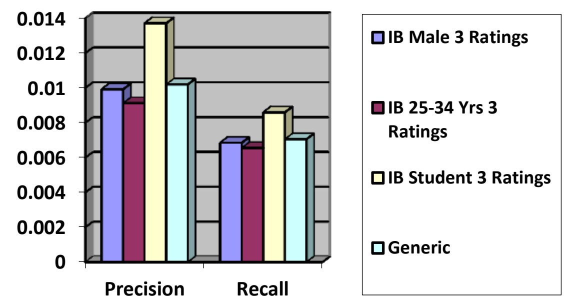

Figure 24. CS Test 1b Precision Recall (Three Ratings)

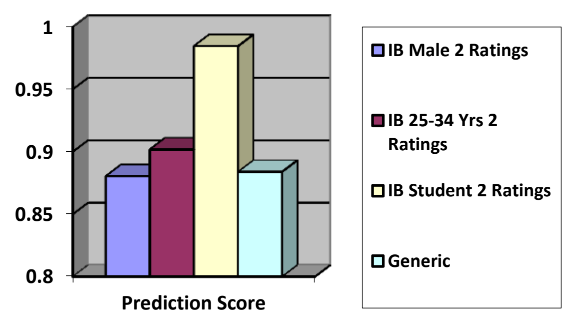

Figure 25. CS Test 1b Prediction Scores (Two Ratings)

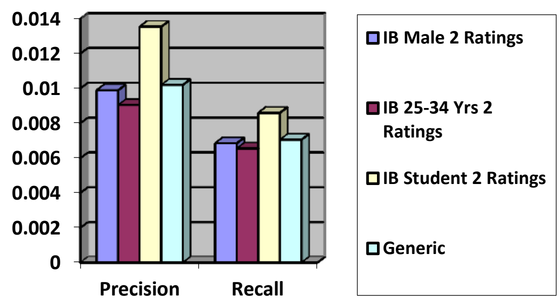

Figure 26. CS Test 1b Precision Recall (Two Ratings)

In the larger dataset, precision and recall scores for student users outperformed the generic user-based model. Other scores provided no overall variation to suggest that the I-BN model outperformed the K-NN model, illustrating that perhaps more data does not translate into improved or reduced performance.

Table 24

Results of Cold-Start Test 2a: Two Elements User-Based 100k

RS Type	Recommendation	Prediction Score	Precision	Recall
Info-Based Male, 30s User Five Ratings	Item ID: 96 (Terminator 2: Judgment Day)	1.0445976094766098	0.0164319248826291 23	0.0119047619047619 1
Info-Based Male, Student User Five Ratings	Item ID: 258 (Contact)	0.9053800166407716	0.0250000000000000 05	0.0202312138728323 74
Info-Based 30s, Student User	Item ID: 258 (Contact)	1.0203447937965393	0.0169491525423728 74	0.0136986301369862 99

Five Ratings				
Info-Based Male, 30s User Three Ratings	Item ID: 9 (Dead Man Walking)	1.0445976094766098	0.016129032258064533	0.011952191235059766
Info-Based Male, Student User Three Ratings	Item ID: 9 (Dead Man Walking)	0.9138891520323575	0.028985507246376805	0.023255813953488372
Info-Based 30s, Student User Three Ratings	Item ID: 9 (Dead Man Walking)	1.1295083502064582	0.03076923076923077	0.0208333333333333
Info-Based Male, 30s User Two Ratings	Item ID: 50 (Star Wars)	1.0445976094766098	0.015555555555555566	0.011952191235059766
Info-Based Male, Student User Two Ratings	Item ID: 100 (Fargo)	0.9138891520323579	0.028169014084507036	0.023255813953488372
Info-Based 30s,	Item ID: 69 (Forrest Gump)	1.017678794406709	0.030303030303030304	0.0208333333333333

Student User Two Ratings				
Generic User-Based	Item ID: 127 (The Godfather)	0.946894677141879	0.008746355685131192	0.006337135614702153

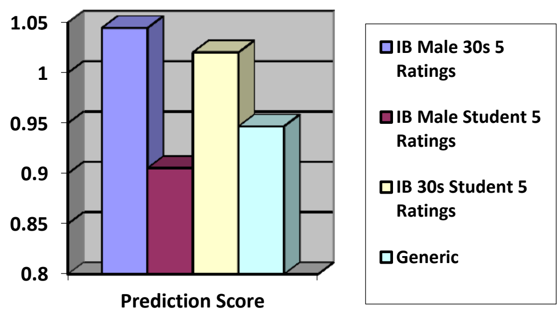

Figure 27. CS Test 2a Prediction Scores (Five Ratings)

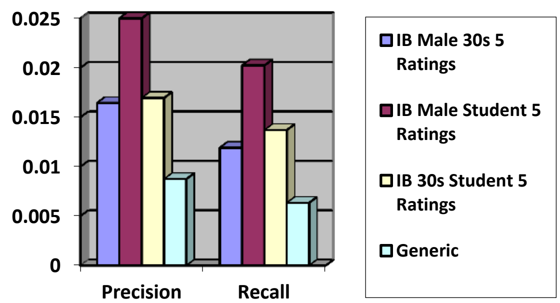

Figure 28. CS Test 2a Precision Recall (Five Ratings)

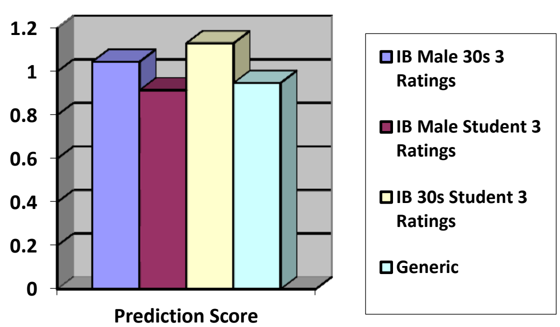

Figure 29. CS Test 2a Prediction Scores (Three Ratings)

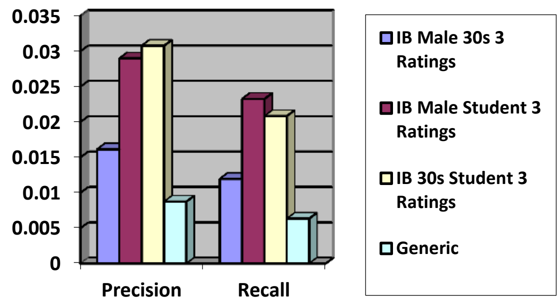

Figure 30. CS Test 2a Precision Recall (Three Ratings)

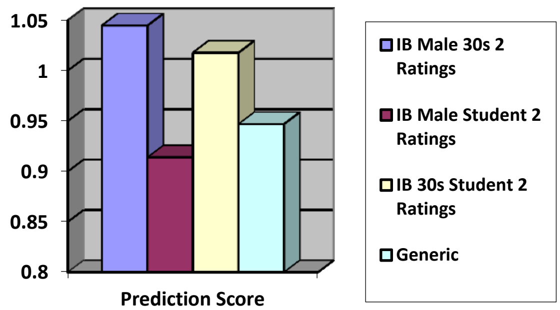

Figure 31. CS Test 2a Prediction Scores (Two Ratings)

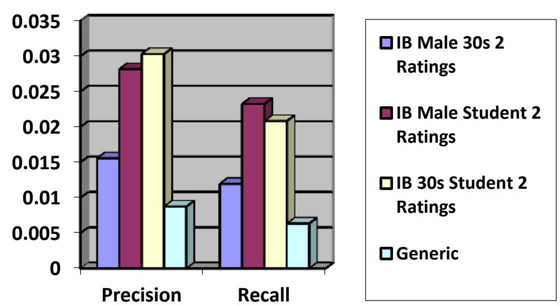

Figure 32. CS Test 2a Precision Recall (Two Ratings)

All Information-Based test cases managed to outperform in precision and recall tests, with male student users outperforming in prediction scores throughout the varying number of provided ratings. This suggests a possibility that higher granularity of items results in better performance of the I-BN model under cold-start conditions.

Table 25

Results of Cold-Start Test 2b: Two Elements User-Based 1m

RS Type	Recommendation	Prediction Score	Precision	Recall
Info-Based Male, 25-34 Years User Five Ratings	Item ID: 2571 (The Matrix)	0.8926991150442477	0.008703220191470837	0.0064020486555697855
Info-Based Male, College/Graduate Student User Five Ratings	Item ID: 2571 (The Matrix)	0.7775723807477714	0.020697167755991286	0.01295731707317072
Info-Based 25-34 Years, College/Graduate Student User Five Ratings	Item ID: 1029 (Dumbo)	0.8647663941544093	0.024999999999999999	0.012658227848101266
Info-Based Male, 25-34 Years User Three Ratings	Item ID: 1968 (The Breakfast Club)	0.8946784922394677	0.009532062391681105	0.006726457399103139
Info-Based Male, College/Graduate Student User Three Ratings	Item ID: 2918 (Ferris Bueller’s Day Off)	0.7775723807477714	0.020879120879120867	0.01297709923664121
Info-Based 25-34, College/Graduate Student User Three Ratings	Item ID: 3578 (Gladiator)	0.8816203267386791	0.025345622119815656	0.012698412698412698
Info-Based Male, 25-34 Years User Two Ratings	Item ID: 1200 (Aliens)	0.8946784922394678	0.009458297506448837	0.006726457399103139
Info-Based Male, College/Graduate Student User Two Ratings	Item ID: 110 (Braveheart)	0.7775723807477714	0.02047413793103449	0.01297709923664121
Info-Based	Item ID: 110	0.881620326738	0.02262443438914	0.012698412698412

25-34 Years, College/Graduate Student User Two Ratings	(Braveheart)	6791	0264	693
Generic User-Based	Item ID: 1266 (Unforgiven)	0.8839017169343101	0.010195530726256999	0.007049775688955351

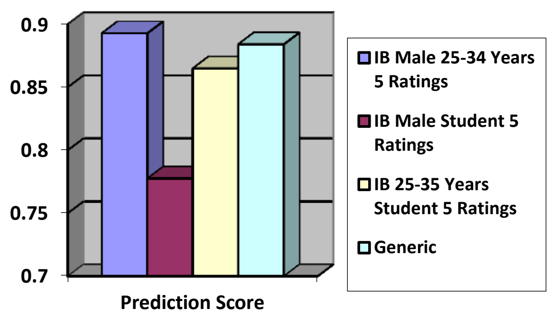

Figure 33. CS Test 2b Prediction Scores (Five Ratings)

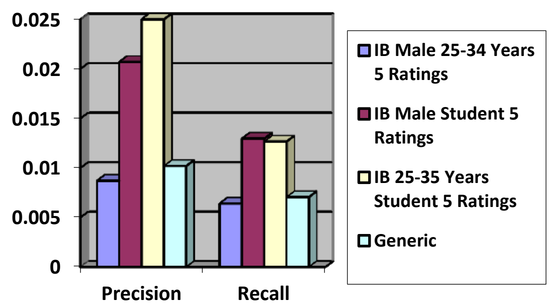

Figure 34. CS Test 2b Precision Recall (Five Ratings)

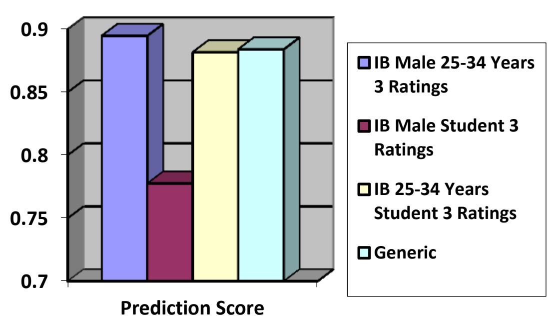

Figure 35. CS Test 2b Prediction Scores (Three Ratings)

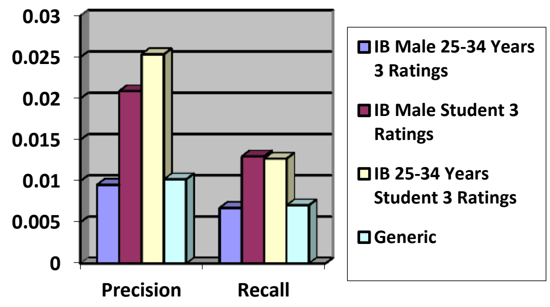

Figure 36. CS Test 2b Precision Recall (Three Ratings)

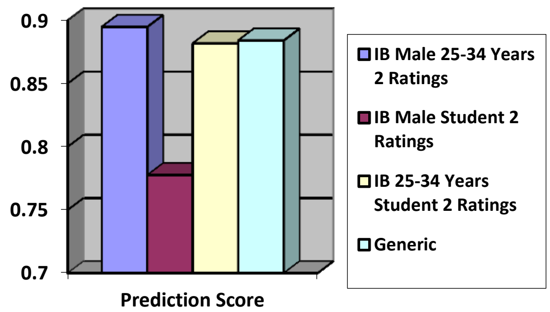

Figure 37. CS Test 2b Prediction Scores (Two Ratings)

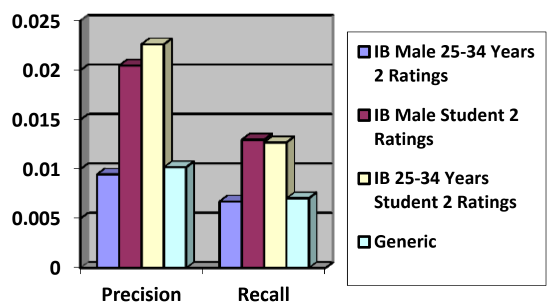

Figure 38. CS Test 2b Precision Recall (Two Ratings)

The results of this larger dataset nearly resemble the performance of the smaller 100k dataset, suggesting that dataset size is not a factor when it comes to the I-BN model. Performance scores in precision and recall varied from the 100k dataset with male users between the ages of twenty-five to thirty-four not significantly outperforming the generic model.

Table 26

Results of Cold-Start Test 3a: Three Elements User-Based 100k

RS Type	Recommendation	Prediction Score	Precision	Recall
Info-Based Male, 30s, Student User Five Ratings	Item ID: 258 (Contact)	0.9794142908520168	0.0306122448979591 8	0.0245901639344262 26
Info-Based Male, 30s, Student User Three Ratings	Item ID: 9 (Dead Man Walking)	0.860556307960959	0.0480769230769230 8	0.0333333333333333 2
Info-Based Male, 30s, Student User Two Ratings	Item ID: 69 (Forrest Gump)	0.8605563079609589	0.0471698113207547 2	0.0333333333333333 2
Generic User-Based	Item ID: 127 (The Godfather)	0.9468946771418791	0.0087463556851311 92	0.0063371356147021 53

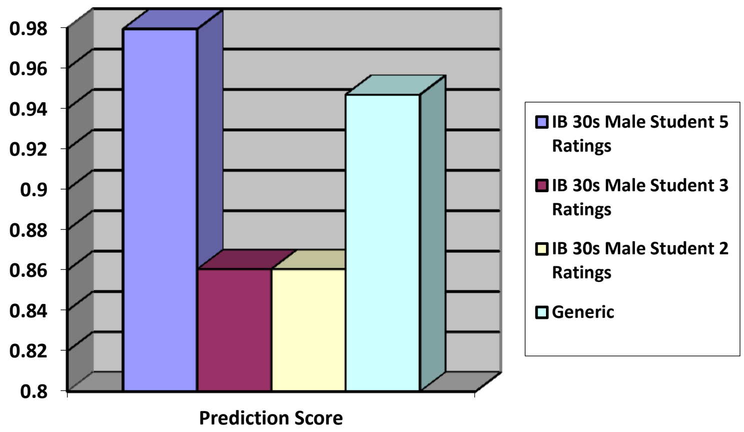

Figure 39. CS Test 3a Prediction Scores

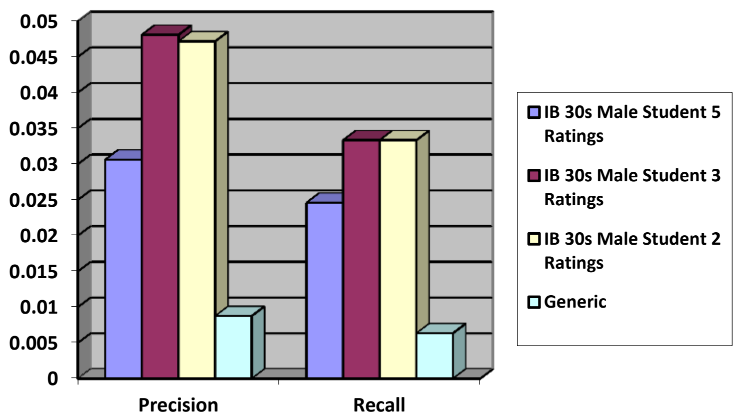

Figure 40. CS Test 3a Precision Recall

Similar to the two variable element test under cold-start conditions, with only male students in their 30s not outperforming the generic model in prediction scores, test 3a begins to illustrate that the I-BN model might serve as a viable solution to the cold-start problem when lists consists of multi-granular elements.

Table 27

Results of Cold-Start Test 3b: Three Elements User-Based 1m

RS Type	Recommendation	Prediction Score	Precision	Recall

Info-Based Male, 25-34, College/Graduate Student User Five Ratings	Item ID: 1029 (Dumbo)	1.1629993471048645	0.027108433734939 76	0.012658227848101264
Info-Based Male, 25-34, College/Graduate Student User Three Ratings	Item ID: 3578 (Gladiator)	1.1522839864095054	0.027607361963190184	0.012711864406779658
Info-Based Male, 25-34, College/Graduate Student User Two Ratings	Item ID: 110 (Braveheart)	1.1522839864095056	0.026627218934911243	0.012711864406779658
Generic User-Based	Item ID: 1266 (Unforgiven)	0.8839017169343101	0.010195530726256999	0.007049775688955351

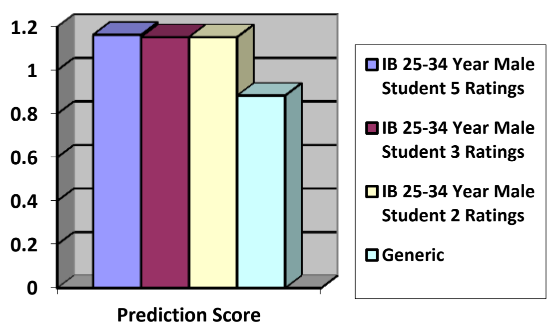

Figure 41. CS Test 3b Prediction Scores

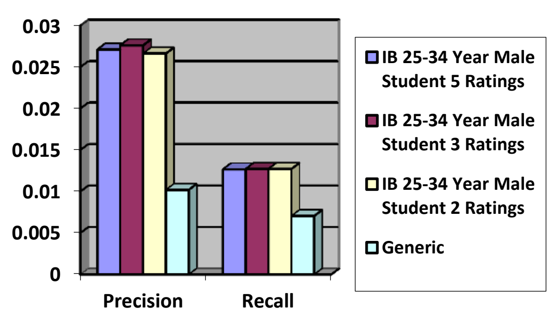

Figure 42. CS Test 3b Precision Recall

Although the generic model outperformed in prediction scores, the high precision and recall scores for the I-BN model continues to support the idea that the I-BN model, utilizing lists of multi-granular elements, might serve as a solution to the cold-start problem.

Part VI: Discussion

The experiment ran a total of twelve major tests, six for accuracy improvement testing, and six for cold-start alleviation testing. The accuracy tests contained three scenarios for tests 1 and 2, while test three contained only one test case scenario. The same setup applies for the cold-start tests, with the exception that cold-start tests ran under three new conditions testing against levels of data sparsity. The test ran with data where the new users had rated five, three, and two movies in total. This resulted in three times the number of results when compared to the accuracy tests, for a total of six test results for tests 1a, 1b, 2a, and 2b. Tests 3a and 3b offered two charts of results. The letters (a) and (b) after each test number refers to the dataset, 100k and 1m respectively. The three test case scenarios for all accuracy and cold-start tests are as follows:

Accuracy Single Variable Element Test 1a:

- I-BN Male Users
- I-BN 20-30 Year Old Users
- I-BN Technician Users.

Accuracy Single Variable Element Test 1b:

- I-BN Female Users
- I-BN Under 18 Users
- I-BN K-12 Student Users

Accuracy Two Variable Element Test 2a:

- I-BN 20-30 Year Old Male Users

- I-BN Male Technician Users
- I-BN 20-30 Year Old Technician Users

Accuracy Two Variable Element Test 2b:

- I-BN Under 18 Female Users
- I-BN Female K-12 Student Users
- I-BN Under 18 K-12 Student Users

Accuracy Three Variable Element Test 3a:

- I-BN 20-30 Year Old Male Technician User

Accuracy Three Variable Element Test 3b:

- I-BN Under 18 Female K-12 Student User

Cold-Start Single Variable Element Test 1a:

- I-BN 30-40 Year Old Users
- I-BN Male Users
- I-BN Student Users

Cold-Start Single Variable Element Test 1b:

- I-BN 25-34 Year Old Users
- I-BN Male Users
- I-BN Student Users

Cold-Start Two Variable Element Test 2a:

- I-BN 30-40 Year Old Male Users
- I-BN Male Student Users
- I-BN 30-40 Year Old Student Users

Cold-Start Two Variable Element Test 2b:

- I-BN 25-34 Year Old Male Users
- I-BN Male Student Users
- I-BN 25-34 Year Old Student Users

Cold-Start Three Variable Element Test 3a:

- I-BN 30-40 Year Old Male Student User

Cold-Start Three Variable Element Test 3b:

- I-BN 25-34 Year Old Male Student User

Table 28 illustrates the number of cases which outperformed the generic K-NN model:

Table 28

Number of Scenarios in Each Test that Outperformed the Traditional Neighborhood Model

Test #	Prediction	Precision	Recall	% Outperform
Accuracy 1a	1/3	1/3	1/3	30%
Accuracy 1b	1/3	1/3	1/3	30%
Accuracy 2a	2/3	1/3	1/3	40%
Accuracy 2b	1/3	1/3	1/3	30%
Accuracy 3a	1/1	N/A	N/A	30-100%
Accuracy 3b	0/1	1/1	1/1	67%
Cold-Start 1a	1/3	2/3	2/3	56%
Cold-Start 1b	0/3	1/3	1/3	20%
Cold-Start 2a	1/3	3/3	3/3	78%
Cold-Start 2b	1/3	2/3	2/3	56%
Cold-Start 3a	2/3	3/3	3/3	89%
Cold-Start 3b	0/3	3/3	3/3	67%

Though the cold-start tests examined cases where User 944, from the 100k dataset, and User 6041, from the 1m dataset, rated 5, 3 and 2 movies, the results showed little to no difference between rating 5, 3, or 2 movies. Therefore, the number of total test cases went from 9 to 3 in tests 1a, 1b, 2a, and 2b, and since the results for 5, 3 or 2 ratings matched, resulting in numbers like 0/9, 3/9 and 6/9 and 9/9 tests which outperformed the generic K-NN model, the fractions were reduced to 0/3, 1/3, 2/3, and 3/3 respectively.

Out of the twelve tests, six outperformed generic neighborhood methods over 50% of the time, given results from the prediction, precision and recall scores. When examining accuracy results alone, one out of six tests outperformed over 50% of the cases. In cold-start conditions, only one of the six tests did not outperform over 50% of the time. This illustrates that while the Information-Based Neighborhood model does not outperform regular K-NN models in situations of normal data distribution, under cold-start conditions the I-BN Model surpassed the K-NN model in several test case scenarios, suggesting it as a viable solution to the cold-start problem.

The multi-granular aspect of the I-BN model must also be mentioned. For the tests where lists consisted of two and three variable elements, for both the accuracy and cold-start tests, the results illustrated a potential for performance improvement. A drawback may lie when the amount of data is minimal. This can be observed in cases where the test runs returned no predicted rating or recommendation, as in the precision and recall test results for test 3a. There are also spots in the test result data which returned no information about recommendation evaluation. It might be possible that at higher multi-granular levels, the nearest neighborhood size becomes too small for the system to generate an accurate prediction. Although higher levels of multi-granularity might zoom in on specific individuals who might share similar tastes to the target user, the lack of results for some test cases suggests that a work around might be necessary when the amount of user or item data results in a small nearest neighborhood size.

The results for the 100k and 1m datasets showed little variation, which implies that the I-BN model might work as well, or perhaps better, with larger datasets. The reason for better performance with a larger dataset stems from the possibility that more data on users and items might be available, allowing for expanded levels of multi-granularity. Since this paper and its accompanying experimentation did not produce results for two lists of multi-granular elements, no definitive conclusion can be given on the viabilities of multi-granular performance on larger datasets.

Part VII: Conclusion

The experiment results suggest that under normal data distribution levels the I-BN model did not significantly outperform the K-NN model, but under cold-start conditions the I-BN model displayed better performance over the traditional K-NN approach, illustrating a potential in handling data sparsity issues. It is also important to note that higher levels of granularity, particularly in larger datasets and under cold-start conditions, demonstrated better performance overall.

The literature mentions that latent factor and matrix factorization models outperform traditional neighborhood models by around 8% when they meet certain optimization criteria, such as a combining K-NN and latent factor techniques with information about movies (Bell & Koren, 2007, p. 75), and cold-start conditions meliorate when demographic information enters the system (Chekkai et al., 2012, p. 760; Eckhardt, 2012, p. 11511-11512; Moreno et al., 2011, p. 256-257; Zhang et al., 2010, p. 1-2). The result of the tests performed on the Information-Based Neighborhood model seems to reiterate much of the previous research. Where accuracy is concerned, the proposed system would benefit from either some form of modification, or further testing, to determine levels of improvement. However, under cold-start conditions, the results of the test show that the Information-Based Neighborhood model might be a viable way of dealing with sparsity issues surrounding data on new users or items.

Part VIII: Future Work

This study leaves open future possibilities to improve or extend upon the Information-Based Collaborative Filtering model. In particular, a few ideas stem from the work performed.

First is the possibility of looking into other methods that reduce the search space, therefore limiting the size of the set of nearest neighbors. Maximum Margin Matrix Factorization was mentioned earlier in this paper. It attempts to limit the number of relevant factors within a larger set of factors (Barbieri, Manco & Ritacco, 2014, p. 20), but MapReduce is another technique that returns a proportionately smaller set of data from a viably larger one. Perhaps future studies could explore MapReduce as a technique that can improve accuracy or cold-start conditions (Dean & Ghemawat, 2008, pp. 107-108).

Another area of exploration pertains to classification methods, in particular classification of items set so that they more readily fall into the five categories that Ackoff mentions describe information, the question words "What, Who, Where, When, and How Many (1989, p.9)." Li and Roth (2003) explored and developed a method of classification stemming from questions and the types of answers they aim to provide. This method of classification is part of the field of study known as natural language processing, and their approach attempts to establish a way in which questions and answers serve as a way of identifying words and their types. Li and Roth's question categories could help identify which bits of data in a database or other storage system would correspond to one of the five descriptor words from Ackoff's definition of information (p. 232-233).

Finally, more exploration needs to be done in other areas of Collaborative Filtering, and even other types of recommender systems. A few candidates for studies include the incorporation of Ackoff's definition of information into methods that relate to content-based recommender systems, tagging using Ackoff's definition of information for such content-based systems, and classification algorithms that might help apply Ackoff's definition of information within a variety of useful scenarios (Lu et al., 2012, p. 9)

References:

Apache Mahout: Scalable machine Learning and data mining. Retrieved March 12, 2015, from http://mahout.apache.org/

Balabanovic, M., & Shoham, Y. (1997). Fab: Content-Based, collaborative recommendation. *Communications of the ACM*, *40*(3), 66-72.

Barbieri, N., Manco, G., & Ritacco, E. (2014). Probabilistic approaches to recommendations. *Synthesis Lectures on Data Mining and Knowledge Discovery*, *9*, 1-181. doi:10.2200/S00574ED1V01Y201403DMK009

Bejoy, KS. (2011). Evaluating Mahout based recommender implementations. Retrieved from http://kickstarthadoop.blogspot.com/2011/05/evaluating-mahout-based-recommender.html

Bobadilla, J., Hernando, A., Ortega, F., & Gutierrez, A. (2012). Collaborative filtering based on significances. *Information Sciences*, *185*(1), 1.

Chekkai, N., Chikhi, S., & Kheddouci, H. (2012). A weighted-graph based approach for solving the cold start problem in collaborative recommender systems. Proceedings from: *The 2012 IEEE Symposium on Computers and Communication*.

Chen, D., & Chiang, Y. (2009). Combining personal ontology and collaborative filtering to design a document recommendation system. *Journal of Service Science and Management*, *2*(4), 322-328.

Dean, J., & Ghemawat, S. (2008). MapReduce: simplified data processing on large clusters *Communications of the ACM*, *51*, 107-113.

Ding, Y., & Li, X. (2005). Time weight collaborative filtering. Proceedings from CIKM ‘05: *The 14th ACM international conference on Information and knowledge management*. Bremen, Germany.

Eckhardt, A. (2012). Similarity of Users' (Content-Based) Preference models for collaborative filtering in few ratings scenario. *Expert Systems with Applications*, *39*(14), 11511.

Himma, K. E., & Tavani, H. T. (2008) *The handbook of information and computer ethics*. Hoboken, NJ: John Wiley & Sons, Inc.

Jeong, B., Lee, J., & Cho, H. (2009). User credit-based collaborative filtering. *Expert Systems with Applications*, *36*(3), 7309-7312.

Kamishima, T., & Akaho, S. (2010). Nantonac collaborative filtering: a model-based approach. Proceedings from RecSys '10: *The fourth ACM Conference on Recommender Systems*. Barcelona, Spain.

Koren, Y., & Bell, R. (2007). Lessons from the Netflix Prize Challenge. *SIGKDD Explorations, 9*(2), 75-79.

Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer, 42*(8), 30-37.

Lai, C., Liu, D., & Lin, C. (2013). Novel personal and group-based trust models in collaborative filtering for document recommendation. *Information Sciences, 239*, 31.

Li, X., & Roth, D. (2006). Learning question classifiers: the role of semantic information. *Natural Language Engineering, 12*(3), 229-249. doi: 10.1017/S1351324905003955

Lika, B., Kolomvatsos, K., & Hadjiefthymiades, S. (2014). Facing the cold start problem in recommender systems. *Expert Systems with Applications, 41*(4), 2065.

Liu, N., Zhao, M., Xiang, E., & Yang, Q. (2010). *Online evolutionary collaborative filtering*. Proceedings from RecSys '10: *The fourth ACM conference on Recommender systems*. Barcelona, Spain.

Lu, L., Medo, M., Yeung, C. H., Zhang, Y., Zhang, Z., & Zhou, T. (2012). Recommender systems. *Physics Reports, 519*(2), 1.

Moreno, M., Segrera, S., López, V., Muñoz, M., & Sánchez, Á. (2011). Mining semantic data for solving first-rater and cold-start problems in recommender systems. Proceedings from IDEAS ‘11: *The 15th Symposium on International Database Engineering & Applications*. Lisbon, Portugal.

MovieLens. Retrieved March 12, 2015, from http://grouplens.org/datasets/movielens/

Nilashi, M., Bagherifard, K., Ibrahim, O., Alizadeh, H., Nojeem, L.A., & Roozegar, N. (2013). Collaborative filtering recommender systems. *Research Journal of Applied Sciences, Engineering and Technology, 5*(16), 4168-4182.

O'Donovan, J., & Smyth, B. (2005). Trust in recommender systems. Proceedings from IUI ‘05: *The 10th International Conference on Intelligent User Interfaces*. San Diego, CA.

Owen, S., Anil, R., Dunning, T., & Friedman, E. (2012). *Mahout in Action*. Shelter Island, NY: Manning.

Porcel, C., López-Herrera, A. G., & Herrera-Viedma, E. (2009). A recommender system for research resources based on fuzzy linguistic modeling. *Expert Systems with Applications, 36*(3), 5173-5183.

Su, X., & Khoshgoftaar, T. M. (2009). A Survey of Collaborative Filtering Techniques. *Advances in Artificial Intelligence,* 1-19. doi:10.1155/2009/421425

Wang, S., Sun, J., Gao, B., & Ma, J. (2012). Adapting vector space model to ranking-based collaborative filtering. Proceedings from CIKM ’12: *The 21st ACM international conference on information and knowledge management*. Maui, Hawaii, USA

Wang, S., Xie, Y., & Fang, M. (2011). A collaborative filtering recommendation algorithm based on item and cloud model. *Wuhan University Journal of Natural Sciences, 16*(1), 16-20.

Wang, Y. & Zhang, Y. (2013). Nonnegative matrix factorization: A comprehensive review. *IEEE Transactions on Knowledge and Data Engineering*, *25*(6), 1336-1353.

Xin, L. I., & Dan, R. (2006). Learning question classifiers: The role of semantic information. *Natural Language Engineering, 12*(3), 229.

Zhang, Z., Liu, C., Zhang, Y., & Zhou, T. (2010). Solving the cold-start problem in recommender systems with social tags. *EPL Europhysics Letters, 92*, 28002.

Appendix A

Equations

(1) K-Nearest Neighbor with User Orientation (Barbieri, Manco & Ritacco, 2014, p. 15)

$$\hat{r}_i^u = \frac{\sum_{v \in N^K(\boldsymbol{u})} \boldsymbol{S}_{\boldsymbol{u},\boldsymbol{v}} \cdot r_i^v}{\sum_{v \in N^K(\boldsymbol{u})} \boldsymbol{S}_{\boldsymbol{u},\boldsymbol{v}} \cdot r_i^v}$$

(2) K Nearest Neighbor with Item Orientation (Barbieri, Manco & Ritacco, 2014, p. 15)

$$\hat{r}_i^u = \frac{\sum_{j \in N^K(i;u)} S_{u,j} \cdot r_j^v}{\sum_{j \in N^K(i;u)} S_{u,j}}$$

(3) Pearson Correlation with User Orientation (Barbieri, Manco & Ritacco, 2014, p. 15)

$$w_{u,v} = \frac{\sum_{i \in I}(r_{u,i} - \bar{r}_u)(r_{v,i} - \bar{r}_v)}{\sqrt{\sum_{i \in I}(r_{u,i} - \bar{r}_u)^2}\sqrt{\sum_{i \in I}(r_{v,i} - \bar{r}_v)^2}}$$

(4) Pearson Correlation with Item Orientation (Barbieri, Manco & Ritacco, 2014, p. 16)

$$w_{i,j} = \frac{\sum_{u \in U}(r_{u,i} - \bar{r}_i)(r_{u,j} - \bar{r}_j)}{\sqrt{\sum_{u \in U}(r_{u,i} - \bar{r}_i)^2}\sqrt{\sum_{u \in U}(r_{u,j} - \bar{r}_j)^2}}$$

(5) Vector Cosine Similarity (Su & Khoshgoftaar, 2009, p. 6)

$$w_{i,j} = \cos(\vec{\imath}, \vec{\jmath}) = \frac{\vec{\imath} \cdot \vec{\jmath}}{\|\vec{\imath}\| * \|\vec{\jmath}\|}$$

(6) Adjusted Cosine Similarity (Su & Khoshgoftaar, 2009, p. 6)

$$w_{i,j} = \text{Adjusted } \cos(\vec{\imath}, \vec{\jmath}) = \frac{\sum_{i \in I_R} \vec{\imath} \cdot \vec{\jmath}}{\sum_{i \in I_R} \|\vec{\imath}\| * \|\vec{\jmath}\|}$$

(7) Singular Value Decomposition (Barbieri, Manco & Ritacco, 2014, p. 17)

$$\text{SVD(A)} = \text{U x } \Sigma \text{ x V}^{\text{T}}$$

(8) Matrix Factorization Optimization Equation (Barbieri, Manco & Ritacco, 2014, p. 19)

$$(U, V) = \underset{U,V}{\operatorname{argmin}}[\sum_{(u,i) \in T} (r_i^u - \sum_{k-1}^{K} U_{u,k} V_{k,i})^2]$$

(9) Regularized Matrix Factorization Optimization Equation (Barbieri, Manco & Ritacco, 2014, p. 20)

$$(U,V) = \underset{U,V}{\operatorname{argmin}}[\sum_{(u,i)\in T}(r_i^u - \sum_{k-1}^{K} U_{u,k}V_{k,i})^2 + \lambda_U tr(U^T U) + \lambda_V tr(V^T V)]$$

Appendix B

Tables

Table 1

User to Item Rating Matrix with Sample Data

	Item 1	Item 2	Item 3	Item 4
User 1	4		5	5
User 2	4	2	1	
User 3	3		2	4
User 4	4	4		
User 5	2	1	3	5

Table 2

SVD Latent Factor Example: Original User-to-Item Rating Matrix

	Item 1	Item 2	Item 3
User 1	3	4	5
User 2	4	2	5
User 3	3	2	4
User 4	5	4	1
User 5	5	5	1

Table 3

SVD Latent Factor Example: User-to-Features Matrix

	Comedy	Action	Romance
User 1	0.48	0.34	-0.72
User 2	0.45	0.45	0.56

User 3	0.37	0.34	0.19
User 4	0.42	-0.58	0.24
User 5	0.50	-0.49	-0.19

Table 4

SVD Latent Factor Example: Feature Relevancy Matrix

Comedy	Action	Love
14.06	0	0
0	4.41	0
0	0	1.66

Table 5

SVD Latent Factor Example: Item-to-Features Matrix

	Comedy	Action	Romance
Item 1	0.64	0.54	0.54
Item 2	-0.35	-0.42	0.84
Item 3	0.69	-0.72	-0.07

Table 6

Traditional Neighborhood Model Sample Matrix Data

	Item A	Item B	Item C
User A	5	2	2
User B	4	2	1
User C	1	5	5

Table 7

Data Possibilities in User-to-Item Rating Matrix Using Information-Based Terms

	(What?) Item A	(What?) Item B	(What?) Item C
(Who?) User A	(How Many?) 1-5/5	(How Many?) 1-5/5	(How Many?) 1-5/5
(Who?) User B	(How Many?) 1-5/5	(How Many?) 1-5/5	(How Many?) 1-5/5
(Who?) User C	(How Many?) 1-5/5	(How Many?) 1-5/5	(How Many?) 1-5/5

Table 8

Information-Based Representation of Data in a User Age Range to Item Rating Matrix

	(What?) Item A	(What?) Item B	(What?) Item C
(How Many?) Age Range A	(How Many?) Rating 1-5/5	(How Many?) Rating 1-5/5	(How Many?) Rating 1-5/5
(How Many?) Age Range B	(How Many?) Rating 1-5/5	(How Many?) Rating 1-5/5	(How Many?) Rating 1-5/5
(How Many?) Age Range C	(How Many?) Rating 1-5/5	(How Many?) Rating 1-5/5	(How Many?) Rating 1-5/5

Table 9

User Ratings based on Location

	(Where?) Location A	(Where?) Location B	(Where?) Location C
(Who?) User A	(How Many?) 1-5/5	(How Many?) 1-5/5	(How Many?) 1-5/5
(Who?) User B	(How Many?) 1-5/5	(How Many?) 1-5/5	(How Many?) 1-5/5
(Who?) User C	(How Many?) 1-5/5	(How Many?) 1-5/5	(How Many?) 1-5/5

Table 10

Number of Item Purchases based on Month

	(When?) Month 1-12	(When?) Month 1-12	(When?) Month 1-12
(What?) Item A	(How Many?) >= 0	(How Many?) >= 0	(How Many?) >= 0
(What?) Item B	(How Many?) >= 0	(How Many?) >= 0	(How Many?) >= 0
(What?) Item C	(How Many?) >= 0	(How Many?) >= 0	(How Many?) >= 0

Table 11

Item Rating Based on Age Range with Sample Data

	Item A	Item B	Item C
Teens	5	3	1
Adults	4	2	1
Seniors	3	1	5

Table 12

Item Ratings to Age Range and Gender Matrix

	Item A	Item B	Item C
Teenage Boys	5	3	1
Teenage Girls	1	3	5

Table 13

Age, Gender, and Occupation Data for User with ID 1 in 100k and 1M Datasets

Dataset	User ID	Age	Gender	Occupation
100k	1	24	Male	Technician
1M	1	Under 18	Female	Student

Table 14

Age, Gender, and Occupation Data for Users 944 (100k Dataset) and 6041 (1M Dataset)

Dataset	User ID	Age	Gender	Occupation
100k	944	32	Male	Student
1M	6041	25-34	Male	Graduate Student

Table 15

Ratings from Users 944 (100k Dataset) and 6041 (1M Dataset) for Five Different Movies

Dataset	User ID	Movie Name	Rating
100k	944	Jumanji	5
100k	944	Seven	2
100k	944	Toy Story	1
100k	944	Star Wars	4
100k	944	Pulp Fiction	3
1M	6041	Jumanji	5
1M	6041	Saving Private Ryan	3
1M	6041	Seven	2
1M	6041	Dumb & Dumber	5
1M	6041	Species	2

Table 16

Results of Accuracy Test 1a: Single Element User-Based 100k

RS Type	Recommendation	Prediction Score	Precision	Recall
Info-Based Male Users	Item ID: 748 (The Saint)	0.9123232979928291	0.008670520231213865	0.006802721088435376
Info-Based 20-30 Year Old Users	Item ID 313 (Titanic)	0.8622032926323706	0.018099547511312222	0.014344262295081971
Info-Based Technician Users	N/A	1.1100707203149798	0.0	0.0
Generic User-	Item ID: 748 (The Saint)	0.9468946771418792	0.008620689655172415	0.006345177664974617

Based				

Table 17

Results of Accuracy Test 1b: Single Element User-Based 1m

RS Type	Recommendation	Prediction Score	Precision	Recall
Info-Based Female Users	Item ID: 2581 (Never Been Kissed)	0.950704225352112	0.011700468018720737	0.008663366336633664
Info-Based Under 18 Users	Item ID: 110 (Braveheart)	1.1133586502075195	0.010040160642570281	0.004411764705882353
Info-Based K-12 Student Users	Item ID: 2571 (The Matrix)	0.8935115744427932	0.009389671361502342	0.005136986301369862
Generic User-Based	Item ID: 2581 (Never Been Kissed)	0.8839017169343092	0.0101784718349135 6	0.007051282051282051

Table 18

Results of Accuracy Test 2a: Two Elements User-Based 100k

RS Type	Recommendation	Prediction Score	Precision	Recall
Info-Based 20-30 Year Old Male Users	Item ID: 313 (Titanic)	1.006771409511566	0.020467836257309944	0.01804123711340206
Info-Based Male, Technician Users	Item ID 294 (Liar Liar)	0.8121258517106373	0.0	0.0
Info-Based 20-30 Year Old	Item ID 294 (Liar Liar)	0.3930600711277552	0.0	0.0

Technician Users				
Generic User-Based	Item ID: 748 (The Saint)	0.9468946771418792	0.008620689655172415	0.006345177664974617

Table 19

Results of Accuracy Test 2b: Two Elements User-Based 1m

RS Type	Recommendation	Prediction Score	Precision	Recall
Info-Based Female, Under 18 Users	Item ID: 110 (Braveheart)	0.9088282151655718	0.014285714285714284	0.0107142857142857l4
Info-Based Female, K-12 Student Users	Item ID: 110 (Braveheart)	1.0248437523841858	0.005747126436781607	0.00423728813559322
Info-Based Under 18, K-12 Student Users	Item ID: 110 (Braveheart)	0.8518518518518516	0.008474576271186439	0.0039370078740157445
Generic User-Based	Item ID: 2581 (Never Been Kissed)	0.8839017169343092	0.01017847183491356	0.00705128205128205l

Table 20

Results of Accuracy Test 3a: Three Elements User-Based 100k

RS Type	Recommendation	Prediction Score	Precision	Recall
Info-Based Male, 20s,	Item ID: 294 (Liar Liar)	0.3930600711277553	0.0	0.0

Technician User				
Generic User-Based	Item ID: 748 (The Saint)	0.9468946771418792	0.008620689655172415	0.006345177664974617

Table 21

Results of Accuracy Test 3b: Three Elements User-Based 1m

RS Type	Recommendation	Prediction Score	Precision	Recall
Info-Based Female, Under 18, K-12 Student User	Item ID: 110 (Braveheart)	1.2352941176470587	0.011627906976744182	0.008695652173913044
Generic User-Based	Item ID: 2581 (Never Been Kissed)	0.8839017169343092	0.0101784718349135 6	0.0070512820512820 51

Table 22

Results of Cold-Start Test 1a: Single Element User-Based 100k

RS Type	Recommendation	Prediction Score	Precision	Recall
Info-Based Male User Five Ratings	Item ID: 127 (The Godfather)	0.9150190626809354	0.008964143426294811	0.006791171477079798
Info-Based 30s User Five Ratings	Item ID: 96 (Terminator 2: Judgment Day)	0.9738020673394205	0.012727272727272733	0.009404388714733545
Info-Based Studen	Item ID: 258 (Contact)	0.7688842146020187	0.013812154696132596	0.011061946902654864

t User Five Ratings				
Info-Based Male User Three Ratings	Item ID: 9 (Dead Man Walking)	0.9123232979928291	0.008806262230919768	0.006802721088435376
Info-Based 30s User Three Ratings	Item ID: 9 (Dead Man Walking)	0.9689175099441684	0.012820512820512827	0.009433962264150945
Info-Based Student Three Ratings	Item ID: 9 (Dead Man Walking)	0.7637486533114782	0.013888888888888888	0.011111111111111106
Info-Based Male User Two Ratings	Item ID: 50 (Star Wars)	0.9123232979928297	0.0087040618955125	0.006802721088435376
Info-Based 30s User Two Ratings	Item ID: 50 (Star Wars)	0.9689175099441688	0.012411347517730502	0.009433962264150945
Info-Based Student User Two Ratings	Item ID: 50 (Star Wars)	0.7637486533114785	0.013440860215053764	0.011111111111111106
Generi	Item ID: 127	0.94689467714187	0.0087463556851311	0.0063371356147021

c User-Based	(The Godfather)	9	92	53

Table 23

Results of Cold-Start Test 1b: Single Element User-Based 1m

RS Type	Recommendation	Prediction Score	Precision	Recall
Info-Based Male User Five Ratings	Item ID: 1266 (Unforgiven)	0.8813922488800819	0.009900990099009927	0.006843455945252349
Info-Based 25-34 Years User Five Ratings	Item ID: 1266 (Unforgiven)	0.8993963782696178	0.009097035040431271	0.006545820745216518
Info-Based College/Graduate Student User Five Ratings	Item ID: 2571 (The Matrix)	0.9802226777625296	0.013687600644122389	0.008581235697940502
Info-Based Male User Three Ratings	Item ID: 1968 (The Breakfast Club)	0.8803637865126619	0.00991609458428683	0.006845407872219049
Info-Based 25-34 Years User Three Ratings	Item ID: 1968 (The Breakfast Club)	0.901803607214429	0.009121621621621634	0.0065491183879093215
Info-Based College/Graduate Student User Three Ratings	Item ID: 2918 (Ferris Bueller's Day Off)	0.9756118954552537	0.013709677419354844	0.008591065292096219
Info-Based Male User Two Ratings	Item ID: 110 (Braveheart)	0.8803637865126609	0.009885931558935385	0.006845407872219049
Info-Based 25-34 Years User Two Ratings	Item ID: 110 (Braveheart)	0.901803607214429	0.009042196918955126	0.0065491183879093215
Info-Based College/Graduate Student User Two Ratings	Item ID: 110 (Braveheart)	0.9845007843441431	0.013535031847133755	0.008591065292096219
Generic User-	Item ID: 1266	0.883901716934	0.01019553072625	0.007049775688955

Based	(Unforgiven)	3101	6999	351

Table 24

Results of Cold-Start Test 2a: Two Elements User-Based 100k

RS Type	Recommendation	Prediction Score	Precision	Recall
Info-Based Male, 30s User Five Ratings	Item ID: 96 (Terminator 2: Judgment Day)	1.0445976094766098	0.016431924882629123	0.0119047619047619 1
Info-Based Male, Student User Five Ratings	Item ID: 258 (Contact)	0.9053800166407716	0.025000000000000005	0.020231213872832374
Info-Based 30s, Student User Five Ratings	Item ID: 258 (Contact)	1.0203447937965393	0.016949152542372874	0.013698630136986299
Info-Based Male, 30s User Three Ratings	Item ID: 9 (Dead Man Walking)	1.0445976094766098	0.016129032258064533	0.011952191235059766
Info-Based Male, Student User Three Rating	Item ID: 9 (Dead Man Walking)	0.9138891520323575	0.028985507246376805	0.023255813953488372

s				
Info-Based 30s, Student User Three Ratings	Item ID: 9 (Dead Man Walking)	1.1295083502064582	0.03076923076923077	0.0208333333333333
Info-Based Male, 30s User Two Ratings	Item ID: 50 (Star Wars)	1.0445976094766098	0.015555555555555566	0.011952191235059766
Info-Based Male, Student User Two Ratings	Item ID: 100 (Fargo)	0.9138891520323579	0.028169014084507036	0.023255813953488372
Info-Based 30s, Student User Two Ratings	Item ID: 69 (Forrest Gump)	1.017678794406709	0.030303030303030304	0.0208333333333333
Generic User-Based	Item ID: 127 (The Godfather)	0.946894677141879	0.008746355685131192	0.006337135614702153

Table 25

Results of Cold-Start Test 2b: Two Elements User-Based 1m

RS Type	Recommendation	Prediction Score	Precision	Recall
Info-Based Male, 25-34 Years User Five Ratings	Item ID: 2571 (The Matrix)	0.8926991150442477	0.008703220191470837	0.0064020486555697855

Info-Based Male, College/Graduate Student User Five Ratings	Item ID: 2571 (The Matrix)	0.7775723807477714	0.020697167755991286	0.01295731707317072
Info-Based 25-34 Years, College/Graduate Student User Five Ratings	Item ID: 1029 (Dumbo)	0.8647663941544093	0.024999999999999999	0.012658227848101266
Info-Based Male, 25-34 Years User Three Ratings	Item ID: 1968 (The Breakfast Club)	0.8946784922394677	0.009532062391681105	0.006726457399103139
Info-Based Male, College/Graduate Student User Three Ratings	Item ID: 2918 (Ferris Bueller's Day Off)	0.7775723807477714	0.020879120879120867	0.01297709923664121
Info-Based 25-34, College/Graduate Student User Three Ratings	Item ID: 3578 (Gladiator)	0.8816203267386791	0.025345622119815656	0.012698412698412698
Info-Based Male, 25-34 Years User Two Ratings	Item ID: 1200 (Aliens)	0.8946784922394678	0.009458297506448837	0.006726457399103139
Info-Based Male, College/Graduate Student User Two Ratings	Item ID: 110 (Braveheart)	0.7775723807477714	0.02047413793103449	0.01297709923664121
Info-Based 25-34 Years, College/Graduate Student User Two Ratings	Item ID: 110 (Braveheart)	0.8816203267386791	0.022624434389140264	0.012698412698412693
Generic User-Based	Item ID: 1266 (Unforgiven)	0.8839017169343101	0.010195530726256999	0.007049775688955351

Table 26

Results of Cold-Start Test 3a: Three Elements User-Based 100k

RS Type	Recommendation	Prediction Score	Precision	Recall
Info-Based Male, 30s, Student User Five Ratings	Item ID: 258 (Contact)	0.9794142908520168	0.0306122448979591 8	0.0245901639344262 26
Info-Based Male, 30s, Student User Three Ratings	Item ID: 9 (Dead Man Walking)	0.860556307960959	0.0480769230769230 8	0.0333333333333333 2
Info-Based Male, 30s, Student User Two Ratings	Item ID: 69 (Forrest Gump)	0.8605563079609589	0.0471698113207547 2	0.0333333333333333 2
Generic User-Based	Item ID: 127 (The Godfather)	0.9468946771418791	0.0087463556851311 92	0.0063371356147021 53

Table 27

Results of Cold-Start Test 3b: Three Elements User-Based 1m

RS Type	Recommendation	Prediction Score	Precision	Recall
Info-Based Male, 25-34, College/Graduate Student	Item ID: 1029 (Dumbo)	1.1629993471048645	0.02710843373493976	0.012658227848101264

User Five Ratings				
Info-Based Male, 25-34, College/Graduate Student User Three Ratings	Item ID: 3578 (Gladiator)	1.1522839864095054	0.027607361963190184	0.012711864406779658
Info-Based Male, 25-34, College/Graduate Student User Two Ratings	Item ID: 110 (Braveheart)	1.1522839864095056	0.026627218934911243	0.012711864406779658
Generic User-Based	Item ID: 1266 (Unforgiven)	0.8839017169343101	0.010195530726256999	0.007049775688955351

Table 28

Number of Scenarios in Each Test that Outperformed the Traditional Neighborhood Model

Test #	Prediction	Precision	Recall	% Outperform
Accuracy 1a	1/3	1/3	1/3	30%
Accuracy 1b	1/3	1/3	1/3	30%
Accuracy 2a	2/3	1/3	1/3	40%
Accuracy 2b	1/3	1/3	1/3	30%
Accuracy 3a	1/1	N/A	N/A	30-100%
Accuracy 3b	0/1	1/1	1/1	67%
Cold-Start 1a	1/3	2/3	2/3	56%
Cold-Start 1b	0/3	1/3	1/3	20%
Cold-Start 2a	1/3	3/3	3/3	78%
Cold-Start 2b	1/3	2/3	2/3	56%
Cold-Start 3a	2/3	3/3	3/3	89%
Cold-Start 3b	0/3	3/3	3/3	67%

Appendix C

Figures

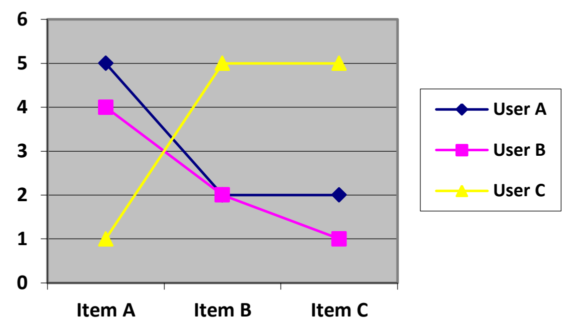

Figure 1. Line Graph of Table 6 Data

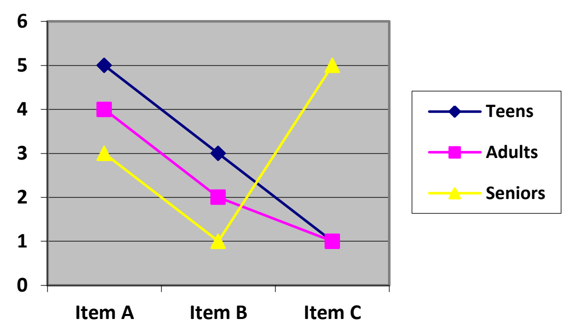

Figure 2. Line Graph of Table 11 Data

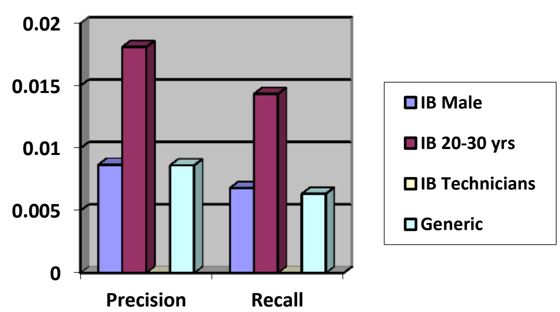

Figure 3. Acc. Test 1a Precision Recall

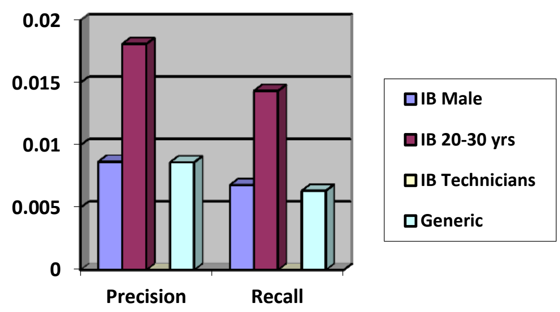

Figure 4. Acc. Test 1a Precision Recall

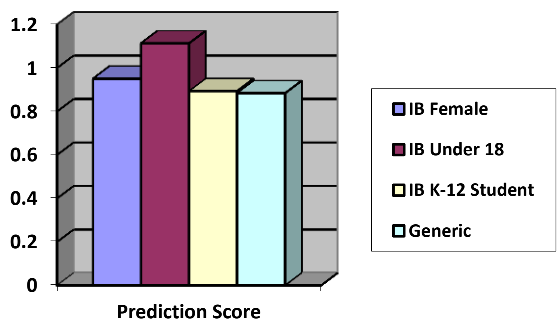

Figure 5. Acc. Test 1b Prediction Scores

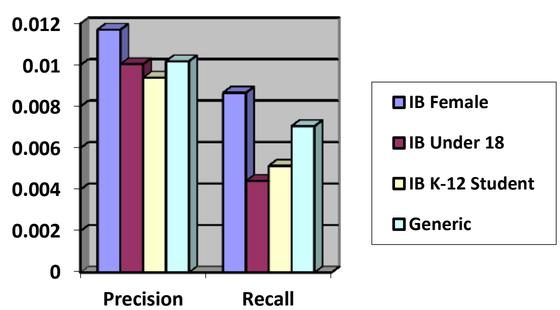

Figure 6. Acc. Test 1b Precision Recall

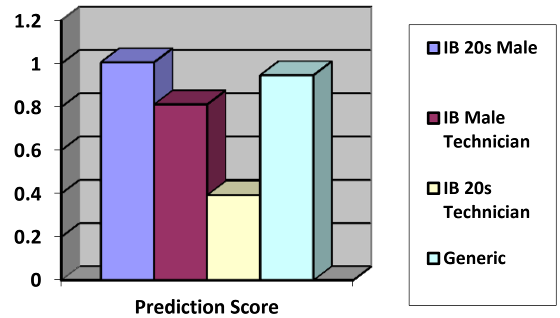

Figure 7. Acc. Test 2a Prediction Scores

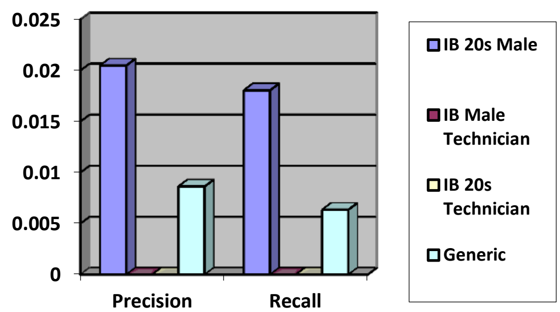

Figure 8. Acc. Test 2a Precision Recall

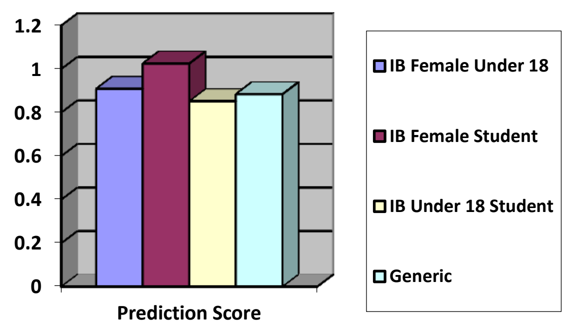

Figure 9. Acc. Test 2b Prediction Scores

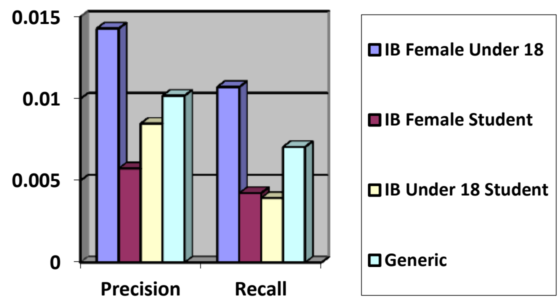

Figure 10. Acc. Test 2b Precision Recall

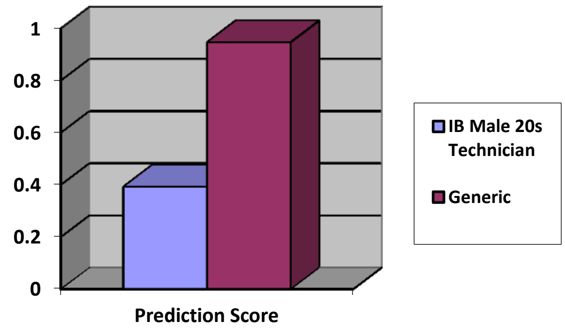

Figure 11. Acc. Test 3a Prediction Scores

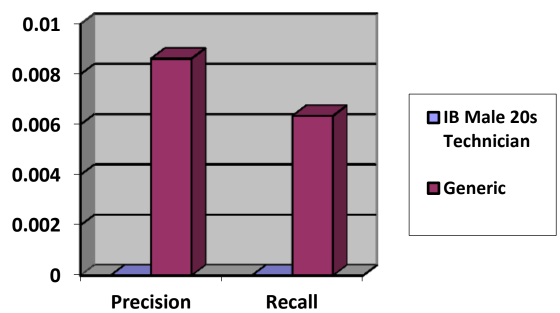

Figure 12. Acc. Test 3a Precision Recall

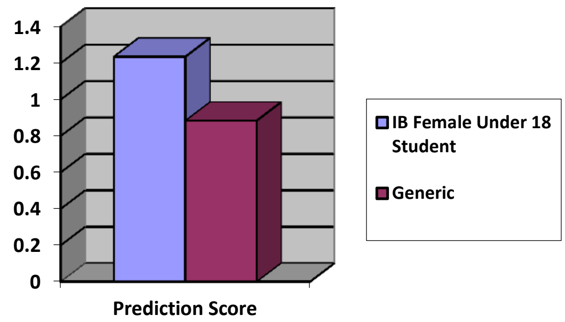

Figure 13. Acc. Test 3b Prediction Scores

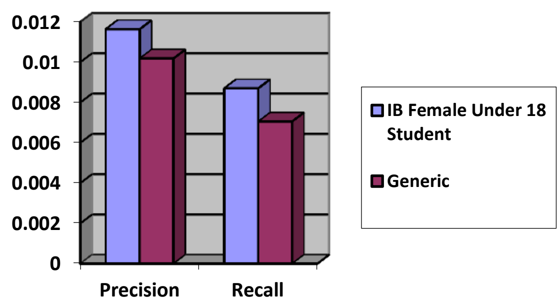

Figure 14. Acc. Test 3b Precision Recall

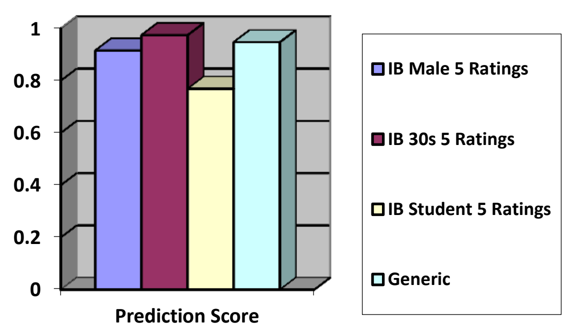

Figure 15. CS Test 1a Prediction Scores (Five Ratings)

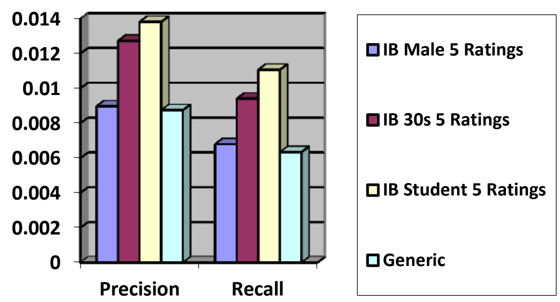

Figure 16. CS Test 1a Precision Recall (Five Ratings)

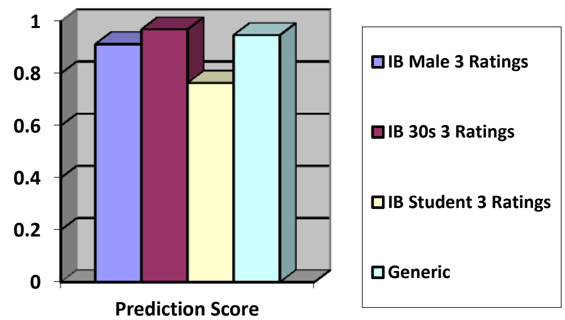

Figure 17. CS Test 1a Prediction Scores (Three Ratings)

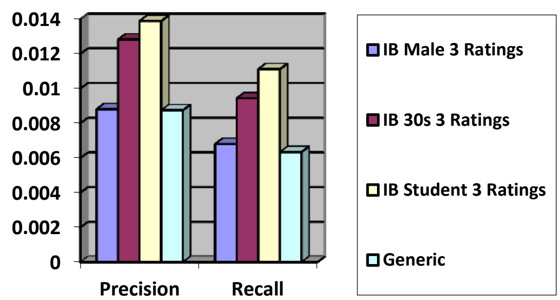

Figure 18. CS Test 1a Precision Recall (Three Ratings)

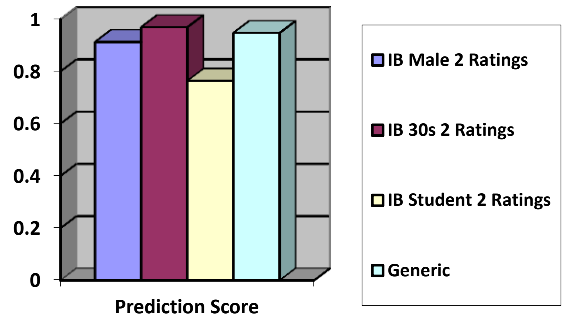

Figure 19. CS Test 1a Prediction Scores (Two Ratings)

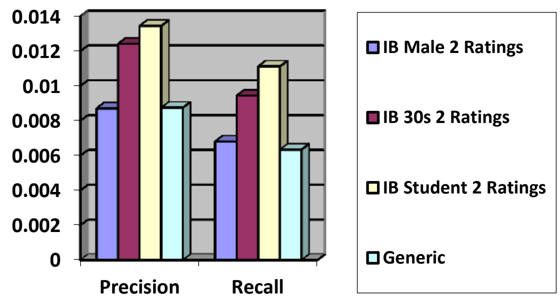

Figure 20. CS Test 1a Precision Recall (Two Ratings)

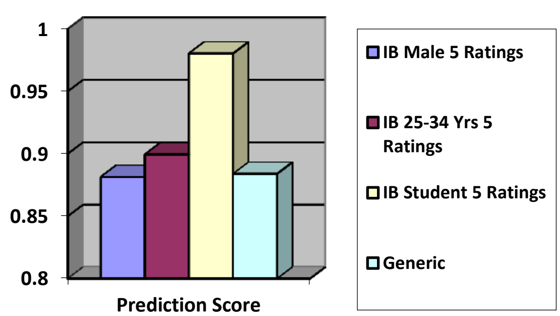

Figure 21. CS Test 1b Prediction Scores (Five Ratings)

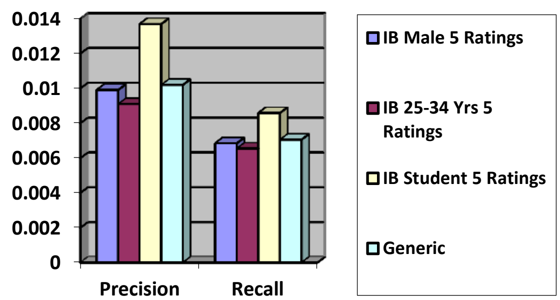

Figure 22. CS Test 1b Precision Recall (Five Ratings)

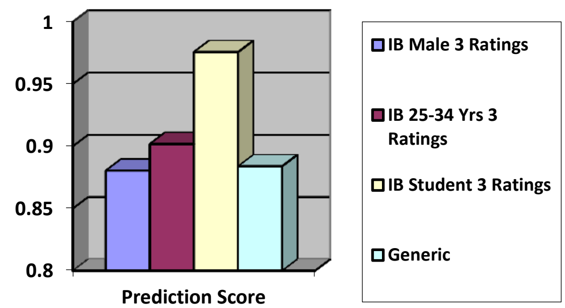

Figure 23. CS Test 1b Prediction Scores (Three Ratings)

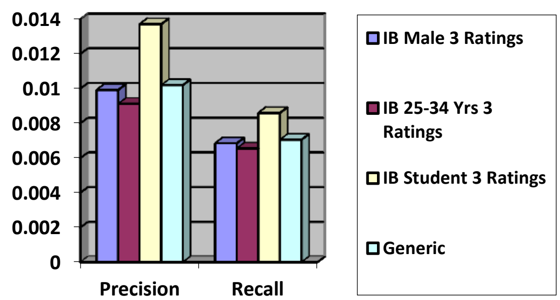

Figure 24. CS Test 1b Precision Recall (Three Ratings)

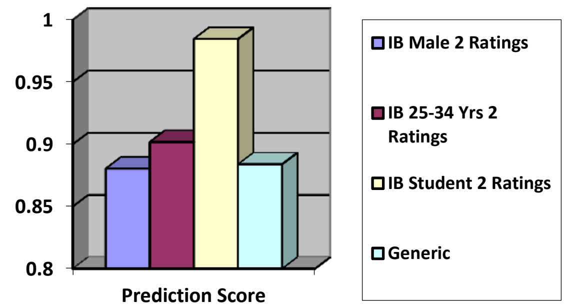

Figure 25. CS Test 1b Prediction Scores (Two Ratings)

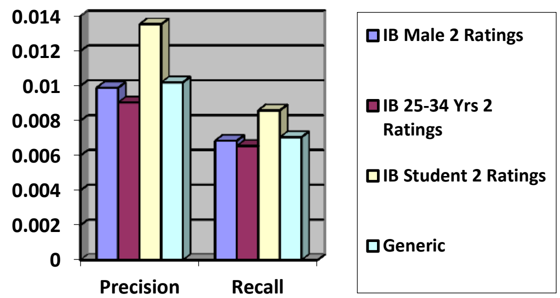

Figure 26. CS Test 1b Precision Recall (Two Ratings)

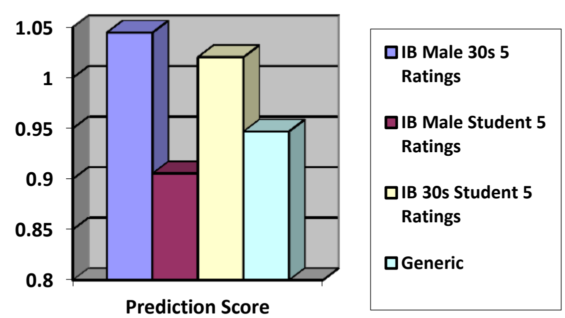

Figure 27. CS Test 2a Prediction Scores (Five Ratings)

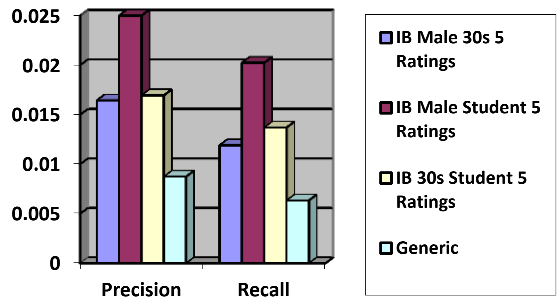

Figure 28. CS Test 2a Precision Recall (Five Ratings)

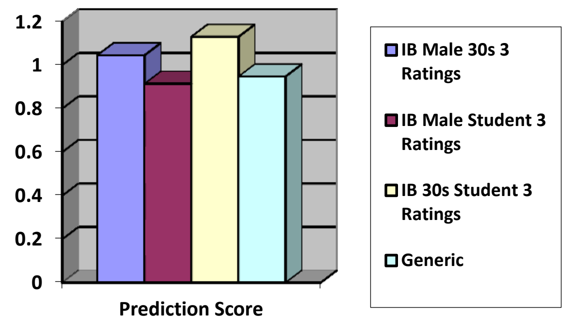

Figure 29. CS Test 2a Prediction Scores (Three Ratings)

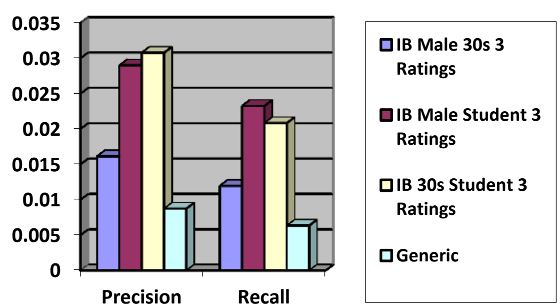

Figure 30. CS Test 2a Precision Recall (Three Ratings)

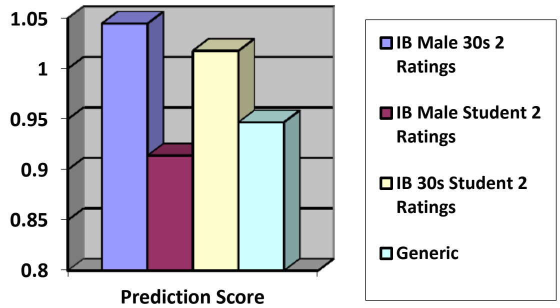

Figure 31. CS Test 2a Prediction Scores (Two Ratings)

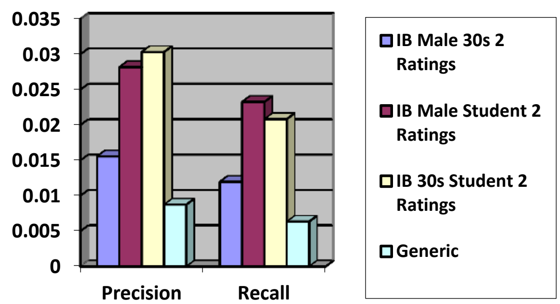

Figure 32. CS Test 2a Precision Recall (Two Ratings)

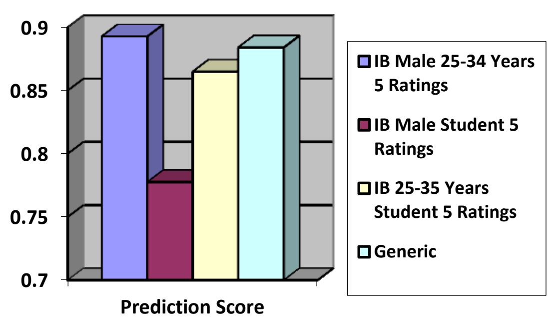

Figure 33. CS Test 2b Prediction Scores (Five Ratings)

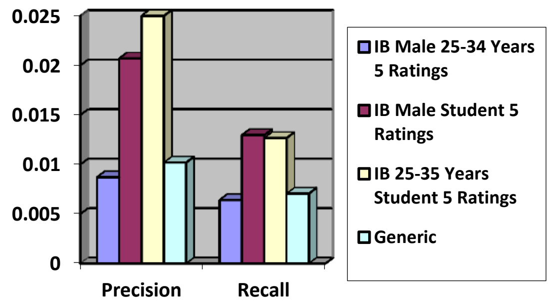

Figure 34. CS Test 2b Precision Recall (Five Ratings)

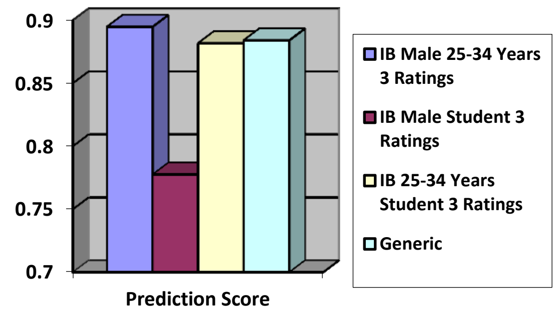

Figure 35. CS Test 2b Prediction Scores (Three Ratings)

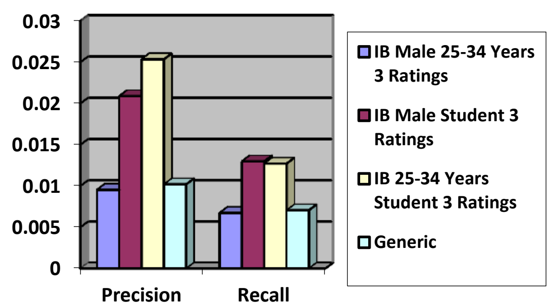

Figure 36. CS Test 2b Precision Recall (Three Ratings)

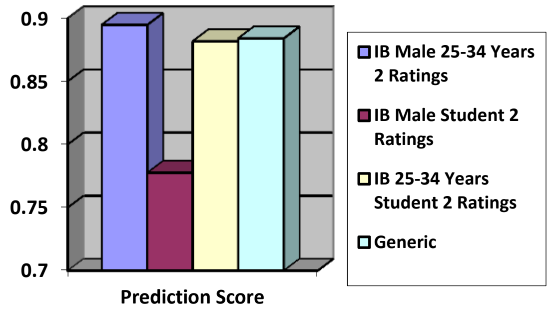

Figure 37. CS Test 2b Prediction Scores (Two Ratings)

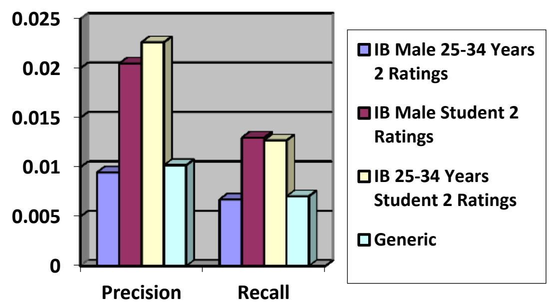

Figure 38. CS Test 2b Precision Recall (Two Ratings)

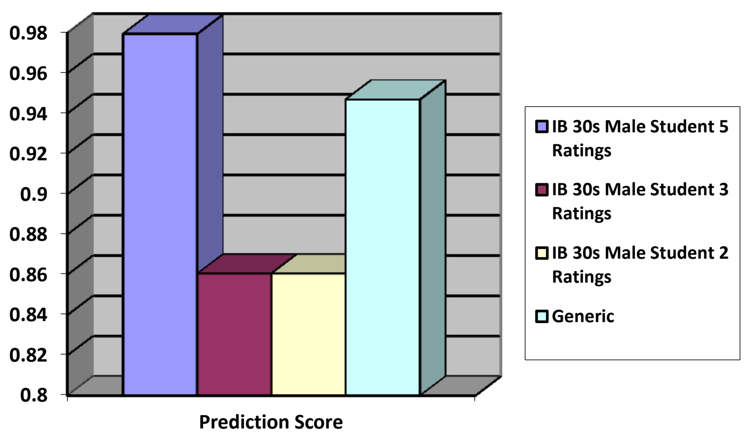

Figure 39. CS Test 3a Prediction Scores

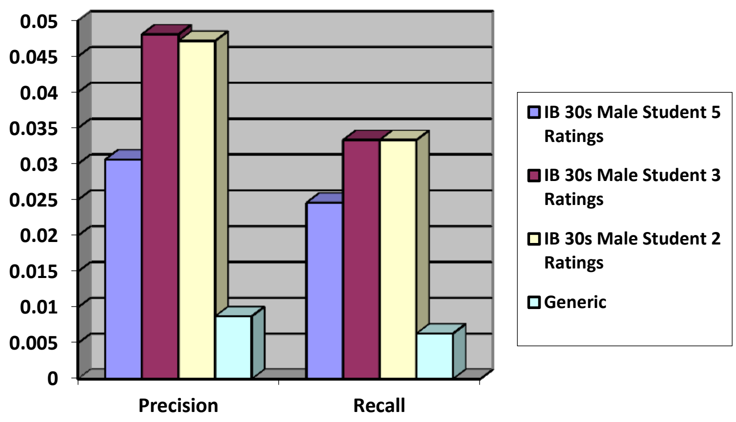

Figure 40. CS Test 3a Precision Recall

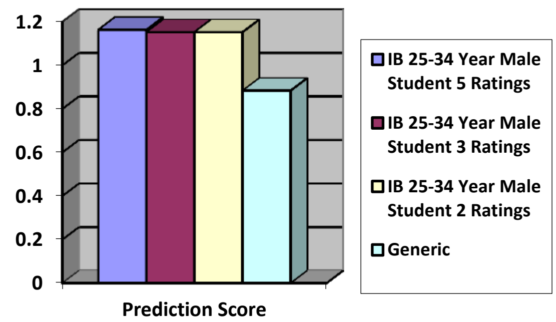

Figure 41. CS Test 3b Prediction Scores

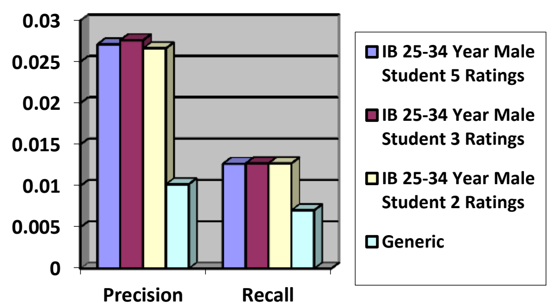

Figure 42. CS Test 3b Precision Recall

Appendix D

Movielens Age and Occupation Data

Note: Age ranges for 100k dataset are approximations. The 100k dataset did not include age ranges, but to match the data in the 1m dataset, an algorithm developed within the I-BN model test system arranged the age numbers for the 100k dataset in a manner similar to that seen in the chart below.

Category	Data	Dataset
Age	0-9	100k
Age	10-19	100k
Age	20-29	100k
Age	30-39	100k
Age	40-49	100k
Age	50-59	100k
Age	Under 18	1m
Age	18-24	1m
Age	25-34	1m
Age	35-44	1m
Age	45-49	1m
Age	50-55	1m
Age	56+	1m
Occupation	administrator	100k
Occupation	artist	100k
Occupation	doctor	100k
Occupation	educator	100k
Occupation	engineer	100k
Occupation	entertainment	100k
Occupation	executive	100k
Occupation	healthcare	100k
Occupation	homemaker	100k
Occupation	lawyer	100k
Occupation	librarian	100k
Occupation	marketing	100k
Occupation	none	100k
Occupation	other	100k
Occupation	programmer	100k
Occupation	retired	100k
Occupation	salesman	100k
Occupation	scientist	100k
Occupation	student	100k
Occupation	technician	100k
Occupation	writer	100k
Occupation	other/not specified	1m

Occupation	academic/educator	1m
Occupation	artist	1m
Occupation	clerical/admin	1m
Occupation	college/grad student	1m
Occupation	customer service	1m
Occupation	doctor/health care	1m
Occupation	executive/managerial	1m
Occupation	farmer	1m
Occupation	homemaker	1m
Occupation	K-12 student	1m
Occupation	lawyer	1m
Occupation	programmer	1m
Occupation	retired	1m
Occupation	sales/marketing	1m
Occupation	scientist	1m
Occupation	self-employed	1m
Occupation	technician/engineer	1m
Occupation	tradesman/craftsman	1m
Occupation	unemployed	1m
Occupation	writer	1m