Dynamically structured holographic memory for recommendation

Matthew Rutledge-Taylor, André Vellino, & Robert L. West

Carleton University Cognitive Science Technical Report 2010-01

cogscitechreports@carleton.ca

Institute of Cognitive Science
2201 Dunton Tower
Carleton University
1125 Colonel By Drive
Ottawa, Ontario, K1S 5B6
Canada

Dynamically Structured Holographic Memory for Recommendation

Matthew F. Rutledge-Taylor
Institute of Cognitive Science
Carleton University
Ottawa, Ontario, K1S 5B6
mrtaylo2@connect.carleton.ca

André Vellino
CISTI Research
National Research Council
Ottawa, Ontario K1A 0R6
andre.vellino@nrc.ca

Robert L. West
Institute of Cognitive Science
Carleton University
Ottawa, Ontario, K1S 5B6
robert_west@carleton.ca

Abstract

Dynamically Structured Holographic Memory (DSHM) is a cognitive model of associative memory that can be applied to the problem of recommendation. DSHM uses holographically reduced representations to encode the associations between objects that it learns about to generate recommendations. We compare the recommendations from this holographic recommender to a user-based collaborative filtering algorithm on several dataset, including MovieLens, and two bibliographic datasets from a scientific digital library.

Off-line experiments show that the DSHM recommender predicts movie ratings as well as collaborative filtering and much better than collaborative filtering on very sparse bibliographic data sets. DSHM also has a unified underlying model that makes multi-dimensional recommendations and their explanations easier to develop. However, DSHM requires significant amounts of computational resources to generate recommendations and it may require a distributed implementation for it to be practical as a recommender for large data sets.

1 Introduction

The function of a recommender system is to recommend items (such as songs, books, movies or merchandise) that are likely to be of interest to a user given both the preferences of the user and the collective preferences of the user community. Recommender systems have been used not only to enhance personalized e-commerce web sites [12] but also to offer a richer information retrieval experience in digital library portals [6].

Most conventional recommender systems operate by clustering similar items according to some characteristic of the item (content-based recommendation) [11], by measuring the similarity among ratings that users have given to items (collaborative filtering – either memory-based or model-based) [3] or by combining the two in some manner (hybrid recommenders) [5]. Hybrid recommenders have been used as a strategy in situations where pure collaborative filtering suffers from well-understood limitations, for instance in situations where usage or rating data is sparse.

Data sparsity is especially problematic in the context of recommending journal articles in a digital library, where a relatively small number of scholars (users) need recommendations from among a relatively large number of articles (items). Extremely small user-item ratios demand more than collaborative filtering alone can provide [18]

In addition, recommender systems need to provide explanations to the user about how the recommendation was made. This allows the user to ascertain the relevance of recommendations, assume greater control over how the recommender behaves and have greater confidence in the validity of its results. However, recommenders that use multiple sources of information and integrate results from multiple algorithms generate relatively ad-hoc explanations. Hybrid recommendation algorithms are typically more difficult to generate explanations for than ones that have a unified prediction model [5].

This paper describes an approach to recommendation based on a cognitive model of associative memory – Dynamically Structured Holographic Memory (DSHM) (section 1.2). This approach is motivated by the intuition that applying a cognitive model of memory could enhance the effectiveness of an information retrieval system by making it behave more like a human expert. In [8] Michael Hugget et. al. assert that "to make information management systems more useful to a wider range of people, it seems reasonable to apply functional cognitive principles to data storage and retrieval".

Thus, the motivation for the experiments described below is grounded in the question of whether a recommender system can be made to perform more like a human expert. Often the best way to get a movie recommendation is to ask the video store clerk to recommend a movie based on what you have enjoyed viewing recently. Similarly, your best bet for finding relevant journal articles is to ask an expert in the field what to read next given a set of articles that you have found useful. Our objective was not so much to discover a recommendation technique that was more effective or efficient for typical recommender tasks in commercial applications as to verify the intuition that a cognitive model of memory was a viable alternative to purely statistical or probabilistic approaches.

We believe that a holographic cognitive model of memory has two noteworthy features that make it suitable for addressing the recommendation problem. The first is adaptability. A holographic memory model can be given any item of information - even properly encoded visual cues - and can potentially make use of it. The second, which follows from the first, is novelty. Human beings, with their wide variety of knowledge sources are often able to integrate information in a way that produces a novel result. For a recommender system, this could mean generating serendipitous, but potentially useful and otherwise unlikely recommendations in ways that could extend beyond the serendipity provided by collaborative filtering systems.

Measuring the accuracy of a recommender on bibliographic data is difficult because there are no standard off-line benchmarks for testing recommender quality. Furthermore, for recommending scholarly articles, end-user satisfaction is a better overall measure of a recommender's success than are the relevance or precision of its underlying algorithms [17]. Hence we chose first to benchmark DSHM with off-line experiments on the MovieLens data, which is often used to benchmark CF recommenders (2.1).

We then performed the off-line experiments described in section 2.2 to compare CF and DSHM on two very sparse bibliographic datasets taken from a digital library. These experiments do not directly evaluate the serendipity of the recommendations provided by DSHM - this is a characteristic that we intend to study in future work. Here we are solely concerning ourselves with demonstrating that, as a starting point, a DSHM based recommender can perform with competitive accuracy to existing CF systems.

1.1 Collaborative Filtering

Recommender systems typically operate on three kinds of entities: users, items and the preference ratings that users have assigned to items. Given a set of ratings for certain items – whether they are obtained from users explicitly or implicitly from, for example, browsing patterns – a user-based collaborative filtering (CF) system will attempt to predict the rating of a previously unrated item for the active user based on how other (similar) users previously rated the same item. In contrast with collaborative filtering methods, which use algorithmic statistics to generate recommendations, DSHM encodes the co-occurrence of a set of items in the representation of the items themselves and uses the memory model of the items' history of associations to generate recommendations.

As noted in the introduction, recommending journal articles in a digital library is more problematic than recommending other kinds of items because the usage data is sparse relative to the number of items in the collection [7]. One remedy for this problem is to use bibliographic citations as a proxy for user ratings [17]. This was the technique we used for the experiments described in this paper.

In the experiments with CF we used a user-based CF recommender that implements k-nearest neighbour and cosine correlation in the Taste framework (now part of the Apache-Lucene machine learning library Mahout [1]). Note that, in this instance, where user preferences are equivalent to article citations, a user-based approach is equivalent to an item-based one.

1.2 DSHM - Dynamically Structured Holographic Memory

DSHM is a cognitive model of human long-term memory [14, 15]. It is designed as a tool for understanding how the human mind organises knowledge, e.g., how it stores, confuses, forgets and accurately retieves information. The implementation of DSHM used for the experiments described in this paper is a self-contained Python program that does not depend on any other modeling software. DSHM has also been reimplemented in Java. The Java version includes improvements in caching vector computations and in system state persistence.

DSHM makes use of Holographic Reduced Representations (HRRs) to encode associations between concepts [13]. It is based on Jones and Mewhort's BEAGLE model of the lexicon [10]. DSHM generalizes BEAGLE to apply to any memory type. BEAGLE can be considered a special case of DSHM. According to the DSHM model, memory is composed of holographic items, which we will refer to here as H-items. H-items represent entities to be remembered by the DSHM system. We refer them as "H-items" to avoid confusion with the generic term "item" used to refer to rated entities in a recommender.

Each H-item consists, primarily, of two large vectors - an "environmental vector" and a "memory vector" - of floating point numbers, each with a Euclidean length of 1.0. The numbers that make up the vectors are generated at random, and adhering to a gaussian distribution. The environmental vector is static (i.e., it does not change) after its creation. It is used as the system's internal representation of the mental entity corresponding to the H-item. In contrast, the memory vector is dynamic (i.e., it changes over time). The memory vector of one H-item is used to store all the associations between that H-item and other H-items in the system. The

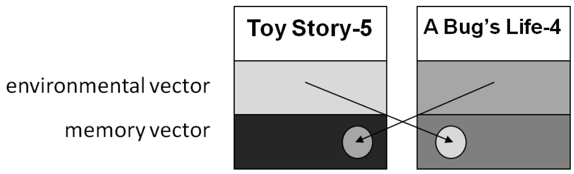

Figure 1. Learning Associations in DSHM

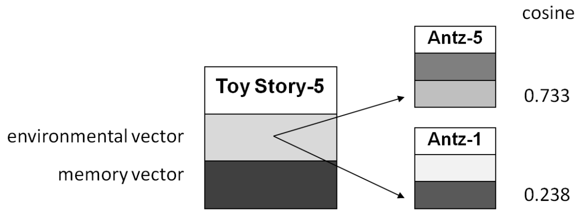

Figure 2. Item Similarity in DSHM

number of elements in the vectors determines the memory capacity of the system. This is because the greater numbers of elements results in less significant collisions between unrelated vectors, on average. Simple models of memory phenomena often use vectors with 128 or 256 elements. The experiments described in this paper used very large data sets. Thus, the number of elements in each vector was set to 2048, providing the models with adequate memory capacity to perform the recommendation task. The drawback to using large vectors is that they take more computational resources to manipulate mathematically. The relationship between vector size, n, and the computational cost is $n \ln(n)$.

Associations between H-items are formed when a set of H-items is given to the system as input. From a cognitive perspective this can be interpreted as the H-items co-occurring in a thought, a verbal utterance or a perception. The system distinguishes between sets for which the order of the elements is essential to the content of the set as a whole (e.g., the words in a sentence) from those that are not (e.g., the things scattered about my work desk that I am perceiving right now). If the set of H-items is unordered, every H-item is associated with every subset of the other H-items in the set, up to a predefined maximum number of elements. In the experiments presented in this paper, this maximum was set to the lowest permissible value (namely one); i.e., each element of a set is only associated with every other element of the set, but not with any combinations of pairs and other n-tuples of elements. The effect of increasing this maximum is to improve the context sensitivity of the system at the cost of additional computational resources. A typical DSHM model of memory[15], applied to smaller data-sets, would use a value of two or three.

These associations between elements are recorded by binding the environmental vectors of the H-items in each subset together, and adding the resulting vector to the memory vectors of the other H-items in the set (see Figure 1). A binding is formed by recursively computing the circular convolution of the environmental vector of an H-item from the given subset and the binding of the remainder of the subset. The circular convolution of vectors is commutative, and thus the order in which the H-items of an unordered set are bound together does not affect the resulting aggregate binding. However, if the set is an ordered list, the neighbours of every H-item up to a system-defined maximum distance are associated with the given H-item in a manner that preserves the order of the H-items [14]. The result of these methods of associating H-items together is that the memory vector of each H-item encodes information about all of the other H-items with which it has co-occurred. The strength of the association between two items can be determined by calculating the cosine similarity between the memory vector of one item and the environmental vector of the other (see Figure 2).

Conceptually, DSHM shares properties in common with, but distinct from computationally inspired cognitive architectures such as ACT-R and neural networks that make use of unsupervised learning. We have argued elsewhere that DSHM may help bridge incompatibilities between these two general frameworks [15].

1.2.1 The structure of DSHM

A collection of H-items in a DSHM system defines a multi-dimensional state-space where each environmental and memory vector is a point on the surface of a hypersphere with radius 1.0. This state-space implements a complex semantic network occupied by the items represented in the system . One property of this organisation of H-items is that given an incomplete pattern, the provided, known, H-items from the pattern can be used to predict the most likely candidates for completing the pattern. This is done first by generating a set of "probe" vectors based on the memory vectors of the known H-items in the pattern. Each probe is computed by reversing the binding process described above and predicts an environmental vector that approximates the vector of the item that best completes the pattern [14]. We refer to this method of prediction as "decoding".

These probes are compared to the environmental vectors of the H-items that might be completions of the pattern. Each candidate H-item in the system is then ranked according to a score based on the sum of the cosines of the candidate H-items' environmental vector and each of the probes. The H-items with the greatest combined scores are those proposed by the system as the most likely completions of the pattern.

By leveraging this pattern-completion property, DSHM has been successfully applied to modeling human memory recall and recognition [15].

Another interesting property of the organisation of H-items in the system is the relationships between H-items' memory vectors. When a large set of patterns has been entered into the DSHM system, and the associations between H-items have been computed, the memory vectors of the H-items in the system will cluster. These clusters are usually open to a meaningful interpretation relating to the content represented by the H-items in the sets of patterns originally presented to the system. For example, as is the case with BEAGLE – where the patterns are sentences and the items are words – the H-items will cluster according to semantic similarity, or synonymy [10]. The reason for this is that items, which are in some way equivalent and can be interchanged for one another, will tend to have the same sets of neighbours, and will therefore develop similar memory vectors. Thus, given an H-item, its memory vector can be used as a probe to be compared to other H-items' memory vectors. The matches found will be the H-items that are similar to the one providing the probe. We refer to this method of prediction as "clustering". It is important to emphasize that the similarity between two H-items discussed here is not based on any content about the items provided to the DSHM system. Rather, the system is inducing the similarity of H-items based only on the patterns in which the items occur. This ability is, of course, part of what make DSHM an interesting cognitive model of memory.

The following section describes how DSHM can exhibit collaborative filtering effects by making use of these properties of DSHM systems.

1.2.2 Recommendation in DSHM

Given a set of items (e.g., books, movies, or journal articles), and a set of users who are defined by what subset of the items they have rated, the purpose of a CF system is to accurately predict what rating a user is likely to assign an item that he or she has not yet rated. Given a test item and a test user, CF assigns to the item a value based on two factors. The first (in user-based CF) is the similarity of the test user to a neighbourhood of other users who have also rated the item and the second is the ratings assigned to the item by the other users. There are various functions that can be used to compute this value. What such a function provides is a metric for how consistent the test item is with the items rated by the test user. In the absence of any other considerations, the concept of "user" is inessential and serves only as a container for his or her rated items.

In contrast, CF methods which use algorithmic statistics to generate recommendations, the most natural way to recommend items in DSHM, in a manner equivalent to user-based CF, is to treat a user as simply a set of unordered ratings. The ratings are considered to be unordered because

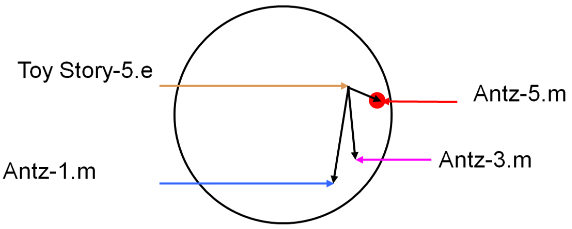

Figure 3. DSHM State Space

we assume that the value given to an item is highly, although not completely, independent of any considerations that may impose an order on the ratings, such as the dates the ratings were provided. We define a rating as the combination of an item and a preference value. When imported into the DSHM system, the ratings are converted into H-items, which we will refer to as H-ratings, and are associated with one another by the binding process described above. Thus, for each H-rating, information about other H-ratings with which it has co-occurred is stored in the memory vector of the H-rating. Once all of the users' ratings have been imported into the system, the state-space defined by the H-ratings' memory vectors will cluster as described above.

Hence, H-ratings that occur in similar sets of ratings will be located together. This does not only mean that ratings by the same user will be located together in the state-space (this may or may not be the case). It is also the case that two ratings that have never been rated by a common user can have very similar memory vectors if the users who have rated them have other ratings in common. Thus, given a rating, the memory vector of the corresponding H-rating can be used to located other H-ratings that are consistent with the given rating (see Figure 3).

Therefore, to predict a new rating, the combined influence of all of the known ratings by the test user can be used to converge on a set of probes that point to a location in the state space where DSHM calculates the new rating ought to exist. H-ratings occupying this location are then returned by the system as its predictions.

It should be noted that, strictly speaking, in DSHM, only the decoding method corresponds to traditional user-based CF. A simple example illustrates the difference between the decoding and clustering methods (see Figure 4). Consider an object A that co-occurs with another object B in one instance, and with a third object C in another instance. The decoding method, when applied to B, will predict A. This is because only A co-occurs with B. The clustering method, when applied to B will predict C because only C has the same neighbours (i.e., A) as B. We have included reference to the clustering method in this paper because it makes use of the exact same data as does user-based CF and produces noteworthy results.

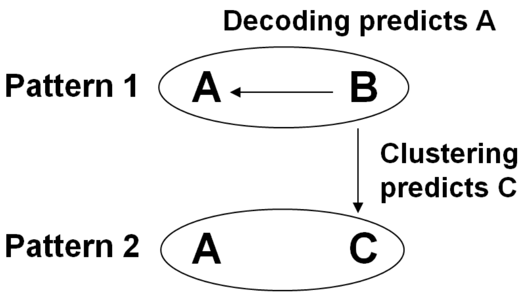

Figure 4. Decoding vs. Clustering in DSHM

2 Experiments

Which methodology to choose for evaluating a recommender's performance depends a great deal on the kind of user task for which the recommender is applied, the datasets being used to perform the evaluation as well as what characteristics (e.g., accuracy, usefulness, serendipity) of the recommender are being evaluated [9]. For this study, our objective was to compare DSHM with a conventional CF recommender to understand both whether DSHM is an applicable technique and how it compares in accuracy with CF on sparse bibliographic datasets. We first establish a baseline for DSHM on the MovieLens dataset and then extend the comparison between these two approaches to two different datasets obtained from a repository of biomedical journal articles.

2.1 MovieLens

Our first experiment was to establish a baseline of predictive accuracy comparisons between DSHM and CF on the MovieLens dataset. This test was done using the usual 10%-90% cross-validation methodology. Previous studies of CF on this data show a Mean Absolute Error (MAE) of approximately 0.73, depending on parameters for the neighbourhood size [4].

The greatest challenge to predicting ratings using DSHM, is the fact that the current implementation has no innate ability to represent magnitude. Given that the goal is to predict a numerical rating, some means of accommodating magnitudes needed to be incorporated into the representation of ratings in the system. We decided that for every movie in the MovieLens dataset, five distinct H-items would be created, one for each possible rating, e.g., "Toy Story (1995)" with a rating value of 4 would be a single atomic entity in the system, as would "Toy Story (1995)" with a rating value of 5.

The cognitive interpretation of this is that the mental representations that correspond to liking a given movie very much, and liking it only somewhat, differ in many dimensions, and not just on a single numerical scale. Hence, representing each rating for each movie as entirely different H-items presumes nothing about how the ratings for the same movie ought to be related to one another. Thus, prior to the learning phase, the various H-ratings representing the ratings of a movie did not bear and a priori relationship to one another. That is, the H-rating representing a movie X, with a preference value of 4 was no more related to the H-rating representing X with a value of 5 than the H-rating representing X with a value of 1. Given this method of representing ratings, a rating prediction then becomes the task of determining which of the five H-ratings for a given movie is most highly associated with the H-ratings corresponding to a test user's other ratings.

Of the 6040 MovieLens users, approximately 10% (614) were randomly removed from the sample and used as test users. The DSHM recommender was trained on the ratings provided by the remaining 5426 users. For each of the test users, the DSHM recommender made predictions for ten of the test user's ratings. These predictions were done one at a time, and used all of the user's other ratings as sources from which to base the predictions.

2.1.1 Results

As mentioned above, there are two distinct, but related, ways in which DSHM can predict a rating. In the case of the decoding method, DSHM is being asked to find ratings that are likely to have co-occurred with the known ratings. Here, DSHM produced a MAE of 1.23. This poor result was initially surprising. The decoding method had been employed with a great deal of success in a DSHM model of human memory data [15]. We hypothesised that the poor performance of the model was due to our choice of how to represent ratings. The drawback of not presuming any relationship between the pair of ratings corresponding to a given movie with values of 4 and 5, which CF can leverage, is obvious. By creating five H-ratings for each movie, there are too many items relative to the number of users for DSHM to discover reliable co-occurrence patterns of preferences for the test users.

In contrast of the performance using the decoding method, the clustering method produced a competitive MAE of 0.71. In this latter case, DSHM is being asked to find ratings that are similar to the known ratings. The reader is reminded that the similarity discussed here is based only on what is induced by the DSHM system, and not based on any content explicitly provided about the movies. This task is more resilient to noise because the value of an H-rating's memory vector is the accumulated influence of many associations, which, on average, represents a reliable location in the state-space occupied by relevantly similar ratings. Note that this method of measuring H-rating similarity would allow for clusters to represent movie preference profiles. For example, a romance movie rated 1 could cluster with an action movie rated 5.

The authors are confident that if the provided with a sufficiently large set of training data the decoding method ought to produce better results. This is because the decoding method is potentially more powerful than the clustering method, but requires either less noisy data or data with more complete coverage of the data space. The potential of the decoding method pertains to its ability to identify more serendipitous patterns in data.

2.2 Journal Articles

Our second experiment is modeled after the off-line experiments in TechLens+, which evaluated the effectiveness of different strategies for recommending journal articles in a digital library context [17]. As with TechLens+, we treated articles as "users" and articles' lists of references as lists of boolean "ratings" for other articles (although we note that while bibliographic references in an article are an indicator of relevance they are not necessarily an indication of *favourable* relevance in the mind of the author).

Our experiment compares DSHM and CF on Top-N results on two bibliographic datasets: one was extracted from a collection of 31,000 articles from 39 Medicine journals and the other from a collection of 114,000 articles from 107 Biology journals. The Medicine collection was reduced to 7495 articles by eliminating articles for which references were unavailable. In addition, the references we used to populate the preferences matrix were only the references that were made to articles in that collection. Overall, the total number of references in the collection was over 273,000, but only 4100 of them were references to articles in the collection, for an average of only 0.55 references per article. In other words, the collection was both very sparse and very loosely connected. The Biology collection was reduced to 38,667 and also had a small average number of references (1.15 per article) to articles in the collection. The connectivity of the article collections — measured as the number of references in an article plus the number of articles that cite it — was slightly above 1 for the Medicine collection and slightly above 2.3 for the Biology collection, as compared to 14 in the CiteSeer collection used in Techlens+ [17].

Our experimental method also differs from the TechLens+ study in some respects. One is that the cross-validation was not 10-fold and not random. Instead, we chose to perform leave-one-out evaluations exhaustively on a sample of the articles biased towards those with the most references to items in the bibliographic collections. One reason for using this strategy rather than the random selection strategy was that the likelihood of picking a random

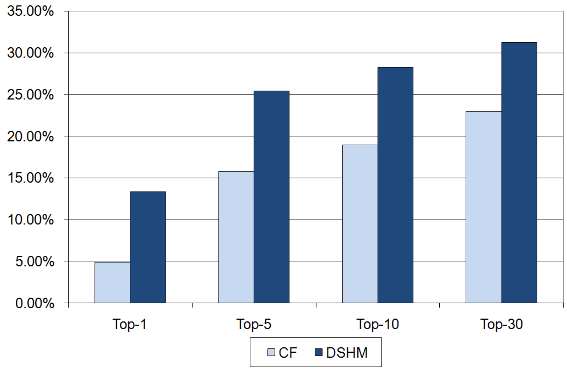

Figure 5. Top-N results for CF and DSHM on Medicine Collection.

article with only one or fewer references to articles in the collection was quite high. Leaving one reference out for each of the articles with the most references seemed more likely to produce a recommendation that was correct. Thus we chose a subset of 95 test articles in each collection which had between 17 and 5 references per article in the Medicine collection, for a total of 570 prediction attempts and between 32 and 13 references in the Biology collection, for a total of 1491 prediction attempts.

For this experiment, the DSHM implementation was essentially the same as for the MovieLens experiment, except that there needed to be only one H-item representation for each unique article. For each of the non-test articles, the H-items representing the article's references were associated together. To make a prediction, the memory vectors of the H-items corresponding to the remaining references were used to generate probes used to rank all of the other articles in the collection according to how highly they were associated with the probes. Again, the two distinct methods of generating these probes, as described above, were used. The DSHM recommender was asked either to recommend articles that were most likely to have co-occurred with the provided references, or to find references that were similar to the provided references.

2.2.1 Results

The results of these experiments are summarized in Figures 5 and 6. In the case of the journal article recommendations, the DSHM recommender produced very good results via the decode method, presented here. Unlike for the MovieLens data, the clustering method did not produce better results for journal article recommendation.

In the case of the Medicine data set, DSHM correctly predicted the first item (Top-1) 76 times out of 570; made a Top-5 recommendation 145 times; Top-10 161 times; and,

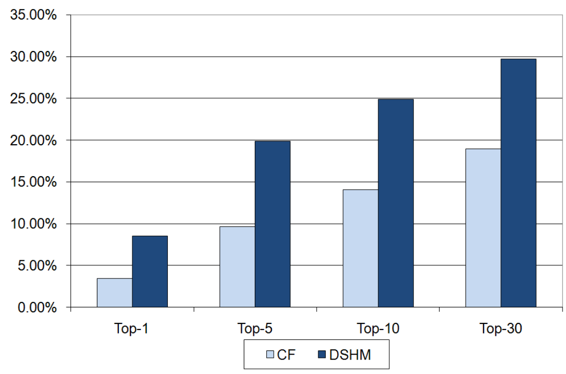

Figure 6. Top-N results for CF and DSHM on Biology Collection.

top-30 178 times. In the case of the Biology data set, the totals were: 127 Top-1; 296 Top-5; 371 Top-10; 443 Top-30, out of 1491 total recommendations. In contrast, our traditional CF system's performance was somewhat poorer. In the medicine case the totals were: 28 Top-1; 90 Top-5; 108 Top-10; and 131 Top-30. In the biology case the totals were 51 Top-1; 144 Top-5; 210 Top-10 and 340 Top-30. Comparing these results, it appears that DSHM does considerably better than CF at making Top-1 and Top-5 recommendations, but that the performance of the two systems converges at the tail end. DSHM also seems to be more sensitive to the reference structure of the two article collections and performs proportionately better with the Medicine collection.

In other words, DSHM correctly predicted the Top-1 references in the Medicine collection almost 3 times as often as CF (76 versus 28), and converged to a 36% improvement over CF for the Top-30. For the Biology collection the accuracy of DSHM is a little less dramatic but exhibits the same trends; e.g., 127 Top-1 predictions for DSHM versus 51 for CF, and a 30% improvement for Top-30.

3 Conclusion

Predicting bibliographic citations from a holographically reduced representation of bibliographic information shows that recommending Top-N items offers better accuracy than CF on very sparse datasets. We interpret this superiority of DSHM as resulting from its ability to self-organise based on the information extracted from the data.

In addition to being more accurate for sparse datasets, the flexibility of holographic recommenders offers promising possibilities for recommending items that have ratings in multiple dimensions as well as item correlations that are content-based. DSHM provides a unified mechanism for implementing what would otherwise be considered a "hybrid" recommender.

The benefits of using DSHM as a recommender for large datasets are mitigated by the considerable computational cost of producing them. In applications where recommendations must be provided quickly, DSHM may not be able to respond fast enough. In some respects, DSHM extracts too much information from the available data, but at too great a computational cost.

In cases where this information is largely composed of noise, the computational resources required to process the information produces too small a return to justify the expense, especially in on-line applications. This is due to the cost in space and time of performing thousands of matrix computations to produce each recommendation. Hence, variants on collaborative filtering techniques are still more practical for digital library recommender systems under most foreseeable circumstances. Nevertheless, in situations where significant pre-compiling of information is feasible, e.g., for relatively static or very small datasets, or when few unique queries are made to the system (see 3), DSHM may be useful for maximally digesting the available data in advance.

4 Future Work

Our future work includes both a research and an applications development effort.

4.1 Research

Our future research on DSHM as a recommender system will focus on examining exactly how information is exploited differently in DSHM compared to CF. This will include cluster analysis of the vector state-space of DHSM recommenders, as well as a detailed examination of how learning in DSHM differs from model building CF. We would also like to compare the accuracy of a DSHM system to which content information (e.g., movie genres or article abstracts) has been added, against the accuracy of typical hybrids of CF and content-based filtering.

In addition, we intend to compare the serendipity characteristics of DSHM recommendations. We believe DSHM may distinguish itself my mimicking the serendipity of recommendations that human experts provide and differ significantly from the serendipity of collaborative filtering.

Finally we plan to investigate the explanatory capabilities of a DSHM system and compare those explanations with ones that are derived from a hybrid CF and Content-Based Filtering algorithm. We believe that DSHM's unified representation of H-item associations provides an opportunity to generate recommendation explanations that improve upon ad-hoc hybrid explanations.

All these experiments will be developed on an open-source, Java implementation of DSHM and their performance evaluated on a Hadoop cluster [2].

4.2 Applications

DSHM is currently being used in the development of an application that provides enhanced metadata to information management systems (IMS). One of the features of this IMS is a dynamic, faceted classification system that uses metadata fields which are customized for each deployment. Facets are distinct, mutually exclusive, and collectively exhaustive perspectives used to describe information. Faceted classification systems are commonly used in the field of Library and Information Sciences as comprehensive set of categories that can be arranged in multiple ways rather than only in a rigid hierarchy [16].

One objective of this application is to reduce the information management burden placed on the end-user. Thus, unlike standard faceted classification systems, the logical relationships between the facets are built into the system. These relationships constrain the possible combinations of metadata values, thus making it easy to autocomplete the metadata fields that follow as a logical consequence. For example, once the end-user has selected his or her name in the Name field and a project name in the Project field, the system will determine that other fields, such as Function and Country, have only one possible value and these fields will be filled in automatically.

The DSHM recommender enables the application to also predict the likely, though not logically necessary, metadata values. The backend server maintains a DSHM model which takes collections of metadata field and value pairs as input. Whenever a user saves a document, the metadata assigned to it is sent to the server and the model is updated. When the user prompts the system for suggested metadata values the DSHM model is presented with a query that contains the metadata fields and values that the user has provided so far, and the remaining empty fields are queried for possible values. The DSHM model then produces rank ordered lists of possible metadata values for each of the empty fields. Each possible metadata value is associated with a real number that represents how strongly it satisfies the empty field's relationship to the completed fields in the DSHM model.

The current application-centric research focuses on determining which metadata values for empty fields are appropriate enough to recommend to the user. DSHM will provide candidates for filling each empty field, even if they are not strong candidates.The current approach is to provide a single metadata value as a recommendation for each empty field only if the internal numerical value associated with the metadata value exceeds a certain threshold. The current best solution is to evaluate whether the value associated with the highest rated metadata value is significantly greater than the value associated with the second highest rated metadata value. If this is the case, there is sufficient certainty that the highest rated metadata value distinguishes itself from the other possible values, and that the second highest rated metadata value would not make a recommendation that is as good as or better than the highest rated metadata value.

Once the system has determined which metadata values for the empty fields make appropriate recommendations, the values are returned and the previously empty fields are filled with recommended values. Recommendations are distinguished from user-selected values by annotating the recommended values with a user interface marker such as a highlight.

The general problem of the computational cost of DSHM, mentioned in section 1.2 is not a problem provided the server has adequate computational resources, and concurrent recommendation requests are infrequent. This is because individual metadata recommendations in this system are less complex than the digital library and MovieLens recommendations presented in this paper, and thus take much less time to compute. Additionally, recommendation is not a necessary feature of metadata assignment. Thus, the number of recommendation requests per user per day is low. Thus, the likelihood that the server will be able to respond to all recommendation requests in a timely manner is good for most anticipated deployments of the system.

References

[1] http://lucene.apache.org/mahout.

[2] http://hadoop.apache.org/.

[3] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *Knowledge and Data Engineering, IEEE Transactions on*, 17(6):734–749, 2005.

[4] B. Sarwar, G. Karypis, J. Konstan and J. Reidl. Item-Based Collaborative Filtering Recommendation Algorithms. In *WWW '01: Proceedings of the 10th International Conference on World Wide Web*, pages 285–295, New York, NY, USA, 2001. ACM Press.

[5] R. Burke. Hybrid Recommender Systems: Survey and Experiments. *User Modeling and User-Adapted Interaction*, 12(4):331–370, 2002.

[6] E. Frias-Martinez, G. Magoulas, S. Chen, and R. Macredie. Automated User Modeling for Personalized Digital Libraries. *International Journal of Information Management*, 26(3):234–248, June 2006.

[7] Z. Huang, H. Chen, and D. Zeng. Applying Associative Retrieval Techniques to Alleviate the Sparsity Problem in Collaborative Filtering. *ACM Trans. Inf. Syst.*, 22(1):116–142, January 2004.

[8] M. Huggett, H. Hoos, and R. Rensink. Cognitive principles for information management: The principles of mnemonic associative knowledge (p-mak). *Minds Mach.*, 17(4):445–485, 2007.

[9] J.L. Herlocker, J.A. Konstan, L.G. Terveen and J.T. Riedl. Evaluating Collaborative Filtering Recommender Systems. *ACM Trans. Inf. Syst.*, 22(1):5–53, 2004.

[10] M. Jones and D. Mewhort. Representing word meaning and order information in a composite holographic lexicon. *Psychological Review*, 114(1):1–37, 2007.

[11] M. Pazzani and D. Billsus. Content-based recommendation systems. pages 325–341. 2007.

[12] D. Pierrakos, G. Paliouras, C. Papatheodorou, and C. D. Spyropoulos. Web Usage Mining as a Tool for Personalization: A Survey. *User Modeling and User-Adapted Interaction*, 13(4):311–372, November 2003.

[13] T. Plate. Holographic reduced representations. *IEEE Transactions on Neural Networks*, 6(3):623–641, May 1995.

[14] M. Rutledge-Taylor and R. West. MALTA: Enhancing ACT-R with a holographic persistent knowledge store. *Proceedings of the XXIX Annual Conference of the Cognitive Science Society, Nashville, TN.*, pages 1433–1438, 2007.

[15] M. Rutledge-Taylor and R. West. Modeling the fan effect using dynamically structured holographic memory. *Proceedings of the XXX Annual Conference of the Cognitive Science Society, Washington, DC.*, 2008.

[16] A. G. Taylor, D. P. Miller, and A. G. Taylor. *Introduction to cataloging and classification / Arlene G. Taylor ; with the assistance of David P. Miller*. Libraries Unlimited, Westport, Conn. :, 10th ed. edition, 2006.

[17] R. Torres, S. McNee, M. Abel, J. Konstan, and J. Riedl. Enhancing Digital Libraries with TechLens+. *Proceedings of the 2004 Joint ACM/IEEE Conference on Digital Libraries*, pages 228–236, 2004.

[18] A. Vellino and D. Zeber. A Hybrid, Multi-dimensional Recommender for Journal Articles in a Scientific Digital Library. *WI-IATW '07: Proceedings of the 2007 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology - Workshops*, pages 111–114, 2007.