

A Belief Propagation Based Recommender System for Online Services

Erman Ayday
Department of Electrical and Computer
Engineering
Georgia Institute of Technology
Atlanta, GA, 30332
erman@ece.gatech.edu

Faramarz Fekri
Department of Electrical and Computer
Engineering
Georgia Institute of Technology
Atlanta, GA, 30332
fekri@ece.gatech.edu

ABSTRACT

In this paper we report our progress in the first application of iterative probabilistic algorithms in the design and evaluation of recommender systems. The proposed iterative recommender system (referred to as BPRS) is based on the belief propagation, a powerful decoding algorithm for turbo codes and Low-Density Parity-Check (LDPC) codes. The belief propagation algorithm relies on a graph-based representation of an appropriately chosen factor graph for the recommender systems. The factor graph representation of the recommender systems turned out to be a bipartite graph, where the users and products are arranged as two sets of variable and factor nodes that are connected via some edges. Recommendations (predicted ratings) for each particular user can be computed by probabilistic message passing between nodes in the graph. We provide an evaluation of BPRS via computer simulations using the MovieLens dataset. We observed that BPRS iteratively reduces the error in the predicted ratings of the users until it converges. Further, our initial results indicate an improvement in the Mean Average Error (MAE) and Root Mean Square Error (RMSE) over the Item Averaging. Therefore, we are confident that the belief propagation is a new promising approach which will offer robustness and accuracy for the recommender systems.

Categories and Subject Descriptors

H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval—*Information filtering*; H.3.5 [Information Storage and Retrieval]: On-line Information Services—*Web-based services*

General Terms

Algorithms, Design

Keywords

Recommender systems, Iterative algorithms, Belief propagation, Online services, E-commerce

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

RecSys2010, September 26–30, 2010, Barcelona, Spain.

Copyright 2010 ACM 978-1-60558-906-0/10/09 ...\$10.00.

1. INTRODUCTION

In most online systems, a user has no choice but to rely on the systems's foresight about his taste on a product based on the existing feedbacks from different users. Recommender systems form a specific type of information filtering technique that aim to support users in their decision-making while interacting with large information spaces. A recommender system is a promising mechanism to help the user by forming the users's predicted taste on the products before using them. A recommender system compares the user's profile to some reference characteristics and recommends items of interest to the user based on preferences he has expressed. Recommender systems are evaluated based on the accuracy of their recommendations (how closely predicted ratings of items match subsequent actual ratings). They are widely used in many websites and are believed to lead the social web in near future.

Most well-known commercial websites such as eBay, Amazon, Netflix and Google use some types of recommender systems. Further, recommender systems have applications in advertisements; which is a successful source of income for Google and social networking websites. By finding similarity among people's choices, recommender systems can be used for the customer directed advertising in which users are directed toward those items that meet their needs and preferences the best. Hence, it is foreseeable that the social web is going to be driven by these recommender systems. Despite recent advances in recommender systems, there is yet a need to develop reliable, scalable and dependable schemes that would also be resilient to various ways a recommender system may be attacked. Moreover, new and untested applications open up new vulnerabilities, and hence, requiring specific solutions for recommender systems.

In this paper we introduce the first application of iterative probabilistic algorithms in the design and evaluation of recommender systems. In our previous work, inspired by our earlier work on iterative decoding of error-control codes in the presence of stopping sets [9], we proposed an algebraic iterative algorithm [1] for reputation systems and showed the benefit of using iterative algorithms for trust and reputation management. Here, instead of the algebraic approach, we expand the idea and introduce a fully probabilistic approach for recommender systems based on the belief propagation. Belief propagation has been successfully applied to the decoding of turbo codes and Low-Density Parity-Check (LDPC) codes [7]. In iterative decoding of LDPC, every check-vertex (in the graph representation of the code) has some opinion of what the value of each bit-vertex should be. The iterative decoding algorithm would then analyze

the collection of these opinions to decide, in each iteration, what value to assign for the bit-vertex under examination. Once the values of the bit-vertices are estimated, in the next iteration, those values are used to determine the satisfaction of the check-vertex values. The contribution of our research stems from the observation that a similar approach can be adapted to provide recommendations (i.e., compute the predicted ratings for the users). Further, to some extent, the analysis of recommender systems resembles that of the code design problem.

We introduce the ‘‘Belief Propagation Based Iterative Recommender System’’ (BPRS). Belief propagation [7] is a message passing algorithm for performing inference on graphical models such as Bayesian networks or Markov random fields. It is used for computing marginal distributions of the unobserved nodes conditioned on the observed ones. The belief propagation algorithm is usually described in terms of operations on factor graphs. The factor graph representation of the recommender systems turned out to be a bipartite graph, where the products and users are arranged as two sets of variable and factor nodes that are connected via some edges. The predicted ratings of a user can be computed by message passing between nodes in the graph. In each iteration of the algorithm, all the variable vertices (products), and subsequently all the factor vertices (users), pass new messages to their neighbors along with their edges until the recommendations converge. Our preliminary results indicate that belief propagation based recommender systems are promising both in terms of performance and efficiency. Therefore, we are very optimistic that the outcome of this research will result in a new class of the recommender system which will be robust and accurate.

The rest of this paper is organized as follows. In the rest of this section, we summarize the related work. In Section 2, we describe the proposed BPRS in detail. Next, in Section 3, we evaluate BPRS via computer simulations using the MovieLens dataset. Finally, in Section 4, we conclude our paper.

1.1 Related Work

In this section, we briefly review the main works in the context. The list of references is not exhaustive due to the page limit. Techniques to build recommender systems can be classified into two main categories: i) content-based filtering [2] in which the system uses behavioral data about a user to recommend items similar to those previously consumed by the user, and ii) collaborative filtering in which the system compares one user’s behavior against the other users’ behaviors and identifies items which were preferred by similar users. Collaborative filtering algorithms also fall into two general classes: i) memory-based algorithms [3, 5] in which the value of an unknown rating is computed by aggregating the ratings of some other users for the same item, and ii) model-based algorithms [6, 8] in which the system uses the collection of the ratings to learn a model that is then used to make rating predictions. Some well known memory-based collaborative filtering algorithms are k-nearest neighbors [5], cosine similarity (vector space similarity) and Pearson’s correlation coefficient based schemes [3]. On the other hand, model-based algorithms include clustering methods [8], probabilistic methods [6] and methods exploiting Singular Value Decomposition (SVD), Principal Component Analysis (PCA), or Maximum Margin Matrix Factorization (MMMF) techniques [4, 10]. Different from the existing schemes, the proposed BPRS algorithm is a graph-based iterative probabilistic algorithm motivated by

previous success of message passing techniques and belief propagation algorithms in decoding error correcting codes as well as the reputation systems [1, 7].

2. BELIEF PROPAGATION FOR RECOMMENDER SYSTEMS

As in every recommender system, our main goal is to provide accurate recommendations for each user by computing the predicted ratings of each user for the items that it did not rate before. We assume two different sets in the system: i) the set of users, and ii) the set of items (products) which are rated by the users and recommended to the users by the system. Users provide feedbacks, in the form of ratings, about the items for which they have an opinion. We let \widehat{R}_{ij} be the predicted rating of user i for the item j . Further, R_{ij} represents the rating previously provided by user i about the item j .

We apply the proposed BPRS algorithm for each user to compute the predicted ratings of that user for the items that it did not rate before. Here, we consider the particular user f and compute the predicted ratings for user f to describe the algorithm in detail. The first step in applying belief propagation is to represent the recommender system by a graphical model. For the factor graph representation, we arrange the collection of the users and the items together with their associated relations (i.e., the ratings of the items by the users) as a bipartite graph in Fig. 1. In this representation, each user corresponds to a *check vertex* in the graph, shown as a square. On the other hand, each item is represented by a *bit vertex* shown as a hexagon in the graph. Each rating (which is directly provided by a user) is represented by an edge from the check-vertex to the bit-vertex. Hence, if a user i has a rating about the j^{th} item, we place an edge with value R_{ij} from the i^{th} check-vertex to the bit-vertex representing the j^{th} item. Since we consider the particular user f , the bipartite graph is reduced such that only the users that connected via a path to f are kept and all the others are removed from the graph together with their edges. In each iteration, the probabilities are exchanged over the edges of the bipartite graph in Fig. 1, estimating probabilistically the predicted ratings of the items which were not previously rated by user f . We assume that the rating values are integers from the set $\Gamma = \{1, 2, 3, 4, 5\}$. The iterative algorithm is executed using the R_{ij} values to obtain the predicted ratings for user f (i.e., \widehat{R}_{fj}).

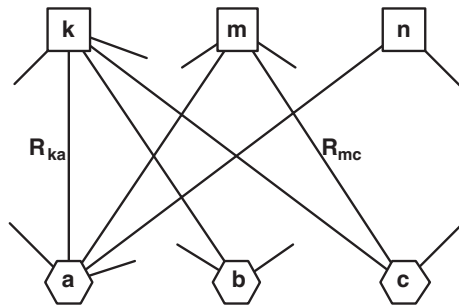


Figure 1: Graphical representation of the scheme from user f ’s point of view.

Details of BPRS may be described by the following procedure. Let $\widehat{R}_{fj}^{(\nu)}$ be the predicted value of the j^{th} bit-vertex at the iteration ν of the BPRS algorithm. For simplicity,

we consider the network in Fig. 1 and describe message exchange between user k and item a . We represent the set of neighbors of item a and users k and f (for whom we are finding the predicted ratings) as \mathbf{N}_a , \mathbf{N}_k , and \mathbf{N}_f , respectively. Note that neighbors of an item are the users who rated it and neighbors of a user are the items whom it rated. Further, let $\Xi = \mathbf{N}_a \setminus \{k\}$ and $\Delta = \mathbf{N}_k \setminus \{a\}$, where $\Xi = \{\xi_1, \xi_2, \dots, \xi_{|\Xi|}\}$ and $\Delta = \{\delta_1, \delta_2, \dots, \delta_{|\Delta|}\}$.

We denote the messages from items to users as μ and from users to items as λ . The message vector $\mu_{a \rightarrow k}^{(\nu)} = [\mu_{a \rightarrow k}^{(\nu)}(1), \mu_{a \rightarrow k}^{(\nu)}(2), \dots, \mu_{a \rightarrow k}^{(\nu)}(5)]$ denotes the probability of $\widehat{R}_{fa} = \ell$ ($\ell \in \Gamma$) at the ν^{th} iteration. In other words, $\mu_{a \rightarrow k}^{(\nu)}(\ell)$ is the probability that $\widehat{R}_{fa} = \ell$ at the ν^{th} iteration. On the other hand, $\lambda_{k \rightarrow a}^{(\nu)}$ denotes the belief (confidence) of user k that the \widehat{R}_{fa} value is equal to R_{ka} at the ν^{th} iteration. This resembles the belief/plausibility concept of the Dempster-Shafer Theory [11]. Given $R_{ka} = 1$, we denote $\lambda_{k \rightarrow a}^{(\nu)}$ as the belief of the k^{th} user that the \widehat{R}_{fa} must be one (at the ν^{th} iteration). Further, since there is no evidence contrary to the hypothesis $\widehat{R}_{fa} = 1$, the plausibility that $\widehat{R}_{fa} = 1$ is equal to one. In other words, from user k 's point of view, the \widehat{R}_{fa} value is equal to one with probability $\lambda_{k \rightarrow a}^{(\nu)}$, and it is equal to one, two, three, four or five with probability $(1 - \lambda_{k \rightarrow a}^{(\nu)})$. Thus, $(1 - \lambda_{k \rightarrow a}^{(\nu)})$ corresponds to the uncertainty in the belief of user k . In order to remove this uncertainty and express $\lambda_{k \rightarrow a}^{(\nu)}$ as a probability of \widehat{R}_{fa} being ℓ ($\ell \in \Gamma$), we distribute the uncertainty among the possible outcomes (one to five) in proportion to the average rating of user f , $A_f = \frac{\sum_{i \in \mathbf{N}_f} R_{fi}}{|\mathbf{N}_f|}$. Therefore, from user k 's point of view, \widehat{R}_{fa} value is equal to one with probability $\lambda_{k \rightarrow a}^{(\nu)} + (1 - \lambda_{k \rightarrow a}^{(\nu)}) \times \frac{1/|A_f - 1|}{\sum_{\gamma \in \Gamma} 1/|A_f - \gamma|}$. Further, it is equivalent

to ℓ ($\ell \neq 1$) with probability $(1 - \lambda_{k \rightarrow a}^{(\nu)}) \times \frac{1/|A_f - \ell|}{\sum_{\gamma \in \Gamma} 1/|A_f - \gamma|}$. The above discussion assumed $R_{ka} = 1$. We note that a similar statement holds for the case when $R_{ka} = 2, 3, 4, 5$.

For item a , we calculate the probability of \widehat{R}_{fa} being ℓ (i.e., the message $\mu_{a \rightarrow k}^{(\nu)}(\ell)$, where $\ell \in \Gamma$) by multiplying all probabilities it received from its neighbors excluding the one from user k , as shown in Fig. 2. We note that the previous ratings of user f for whom the recommendations are computed are the base for the algorithm. Hence, the \widehat{R}_{fj} values of the items which are rated by user f are set to the values of edges R_{fj} (i.e., $\widehat{R}_{fj} = R_{fj}$ when $j \in \mathbf{N}_f$). Therefore, if $a \in \mathbf{N}_f$, the messages generated from item a are constant (do not change from one iteration to the next) since the \widehat{R}_{fa} value of this items is fixed based on the ratings of user f . Hence, the message $\mu_{a \rightarrow k}$ from item a to user k at the ν^{th} iteration is given by

$$\mu_{a \rightarrow k}^{(\nu)}(\ell) = \begin{cases} \frac{\prod_{i \in \Xi} \Pr(\widehat{R}_{fa} = \ell | R_{ia}, \lambda_{i \rightarrow a}^{(\nu-1)})}{\sum_{\gamma \in \Gamma} \prod_{i \in \Xi} \Pr(\widehat{R}_{fa} = \gamma | R_{ia}, \lambda_{i \rightarrow a}^{(\nu-1)})} & \text{if } a \notin \mathbf{N}_f \\ 1 & \text{if } a \in \mathbf{N}_f \text{ and } R_{fa} = \ell \\ 0 & \text{if } a \in \mathbf{N}_f \text{ and } R_{fa} \neq \ell, \end{cases} \quad (1)$$

where $\Pr(\widehat{R}_{fa} = \ell | R_{ia}, \lambda_{i \rightarrow a}^{(\nu-1)})$ is given by

$$\Pr(\widehat{R}_{fa} = \ell | R_{ia}, \lambda_{i \rightarrow a}^{(\nu-1)}) = \begin{cases} \lambda_{i \rightarrow a}^{(\nu-1)} + (1 - \lambda_{i \rightarrow a}^{(\nu-1)}) \times \frac{1/|A_f - \ell|}{\sum_{\gamma \in \Gamma} 1/|A_f - \gamma|} & \text{if } R_{ia} = \ell \\ (1 - \lambda_{i \rightarrow a}^{(\nu-1)}) \times \frac{1/|A_f - R_{ia}|}{\sum_{\gamma \in \Gamma} 1/|A_f - \gamma|} & \text{if } R_{ia} \neq \ell. \end{cases} \quad (2)$$

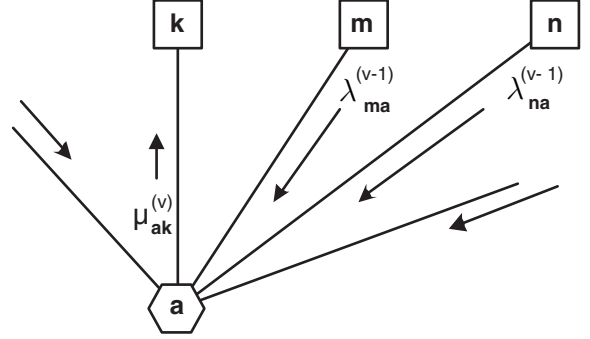


Figure 2: Message from item a to user k at the ν^{th} iteration.

This computation is repeated for every neighbors of each item. This finishes the first half of the ν^{th} iteration. During the second half, for each user, we generate the messages (λ) and pass to its neighbors. For user k , we calculate its confidence on its ratings by computing the deviation in its ratings R_{ki} ($i \in \Delta$) based on the messages $\mu_{i \rightarrow k}^{(\nu)}$ ($i \in \Delta$) it received from its neighbors. Thus, the message from user k to item a at the ν^{th} iteration is formed using the principles of the belief propagation as

$$\lambda_{k \rightarrow a}^{(\nu)} = 1 - \frac{1}{\rho |\Delta|} \left\{ \sum_{\alpha_1 \in \Gamma} \dots \sum_{\alpha_{|\Delta|} \in \Gamma} [|R_{k\delta_1} - \alpha_1| + \dots + |R_{k\delta_{|\Delta|}} - \alpha_{|\Delta|}|] \prod_{x \in \Delta} \mu_{x \rightarrow k}^{(\nu)}(h(x)) \right\}, \quad (3)$$

where $h(x)$ is a mapping defined as $h(x) = \alpha_i$ if $x = \delta_i$. Further, ρ , which is the highest possible deviation of a user is set to 4 in this particular rating system, where the rating values are integers from the set Γ . The above equation can be interpreted as one minus the average inconsistency of user k calculated by the messages received from its neighbors (excluding item a). In other words, reliability of users (in their ratings) is measured by their relative similarity to user f (for whom we compute the recommendations). Hence, the most reliable user in the system is the user with the most similar rating pattern to the user f . This message transfer is illustrated in Fig. 3. Further, it can be shown that (3) can be simplified to

$$\lambda_{k \rightarrow a}^{(\nu)} = 1 - \frac{1}{\rho |\Delta|} \left(\sum_{i \in \Delta} \sum_{\gamma \in \Gamma} (\gamma - R_{ki}) \mu_{i \rightarrow k}^{(\nu)}(\gamma) \right). \quad (4)$$

The algorithm proceeds to the next iteration in the same way as the ν^{th} iteration. We clarify that the iterative algorithm starts by computing $\mu_{a \rightarrow k}^{(1)}$ in (1) by using $\lambda_{i \rightarrow a}^{(0)} = \epsilon$ in (1)

and (2)) as initial values for all items, where $0 < \epsilon < 1$. We used $\epsilon = 0.8$ throughout the evaluation of BPRS.

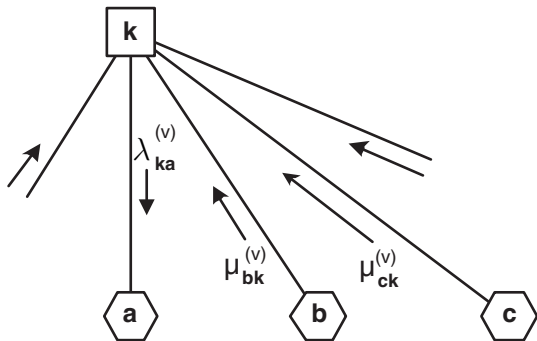


Figure 3: Message from user k to item a at the ν^{th} iteration.

At the end of each iteration the predicted ratings of user f are calculated using modified (1). That is, we use the set \mathbf{N}_a instead of Ξ in (1) to compute $\mu_a^{(\nu)}(\ell)$ ($\ell \in \Gamma$). Then we set $\widehat{R}_{fa}^{(\nu)} = \sum_{i=1}^5 i\mu_a^{(\nu)}(i)$. We repeat this to compute the predicted rating of every item for user f (for which it did not have any rating). The iterations stop when the \widehat{R}_{fj} values converge for every item.

3. EVALUATION OF BPRS

We evaluated the performance of BPRS using the 100K MovieLens dataset¹ via MATLAB. The dataset contains 100,000 ratings from 943 users on 1682 items (movies) in which each user has rated at least 20 items. Further, the rating values are integers from 1 to 5. We used the Mean Average Error (MAE) and Root Mean Square Error (RMSE) metrics over the predicted ratings (the ones in the test dataset) to measure the performance of BPRS. Furthermore, we compared the performance of BPRS with the Item Averaging technique, in which the recommendations are calculated by averaging over all ratings of each item. We illustrated the results in Table 3. Our initial results indicate an improvement of 11% in MAE and 6% in RMSE over the Item Averaging. Therefore, we believe that BPRS is a promising method that will result in a new class of recommender system which will be robust and accurate. Finally, assuming N_U users and N_I items in the system, we obtained the computational complexity of BPRS as $\max(\mathcal{O}(cN_UN_I), \mathcal{O}(cN_U^2))$ in the number of multiplications, where c is the average number of nonzero elements in each row of the user-item matrix. We note that due to the sparseness of the user-item matrix, the coefficient c is a small number.

	MAE	RMSE
BPRS	0.7514	0.9867
Item Averaging	0.8357	1.0417

Table 1: Performance of BPRS and Item Averaging in MAE and RMSE.

¹Available at: <http://www.grouplens.org/node/73>.

4. CONCLUSION

In this paper, we introduced the Belief Propagation Based Iterative Recommender System (BPRS). The scheme is an iterative probabilistic algorithm motivated by the prior success of message passing techniques and belief propagation algorithms on decoding of turbo codes and low-density parity-check codes. BPRS relies on a graph-based representation of an appropriately chosen factor graph for recommender systems. The predicted ratings of users on the products are computed by message passing between nodes in the factor graph until the convergence. We showed that BPRS is a promising method via computer simulations using the MovieLens dataset. Further, our initial results indicated an improvement in the Mean Average Error (MAE) and Root Mean Square Error (RMSE) over the Item Averaging. Therefore, we believe that the outcome of this research will result in a new class of the recommender system which will be robust and accurate.

5. REFERENCES

- [1] E. Ayday, H. Lee, and F. Fekri. An iterative algorithm for trust and reputation management. *ISIT '09: Proceedings of IEEE International Symposium on Information Theory*, 2009.
- [2] M. Balabanovic and Y. Shoham. Fab: Content-based, collaborative recommendation. *Communications of the ACM*, 40:66–72, 1997.
- [3] J. S. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence*, pages 43–52, 1998.
- [4] K. Goldberg, T. Roeder, D. Gupta, and C. Perkins. Eigentaste: A constant time collaborative filtering algorithm. *Information Retrieval*, 4(2):133–151, 2001.
- [5] J. Herlocker, J. A. Konstan, and J. Riedl. An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms. *Information Retrieval*, 5(4):287–310, 2002.
- [6] T. Hofmann. Latent class models for collaborative filtering. *In Proceedings of the 16th International Joint Conference on Artificial Intelligence*, pages 688–693, 1999.
- [7] F. Kschischang, B. Frey, and H. A. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 47:498–519, Feb. 2001.
- [8] B. Mobasher, H. Dai, T. Luo, and M. Nakagawa. Discovery and evaluation of aggregate usage profiles for web personalization. *Data Mining and Knowledge Discovery*, 6(1):61–82, 2002.
- [9] H. Pishro-Nik and F. Fekri. On decoding of low-density parity-check codes on the binary erasure channel. *IEEE Transactions on Information Theory*, 50:439–454, March 2004.
- [10] B. M. Sarwar, G. Karypis, J. A. Konstan, and J. T. Riedl. Application of dimensionality reduction in recommender system - a case study. 2000.
- [11] G. Shafer. The Dempster-Shafer theory. *Encyclopedia of Artificial Intelligence*, 1992.