BPRS: Belief Propagation Based Iterative Recommender System

Erman Ayday, Arash Einolghozati, Faramarz Fekri
School of Electrical and Computer Engineering
Georgia Institute of Technology, Atlanta, GA 30332
Email:{eaydayi, einolghozati, fekri}@ece.gatech.edu

***Abstract*—In this paper we introduce the first application of the Belief Propagation (BP) algorithm in the design of recommender systems. We formulate the recommendation problem as an inference problem and aim to compute the marginal probability distributions of the variables which represent the ratings to be predicted. However, computing these marginal probability functions is computationally prohibitive for large-scale systems. Therefore, we utilize the BP algorithm to efficiently compute these functions. Recommendations for each active user are then iteratively computed by probabilistic message passing. As opposed to the previous recommender algorithms, BPRS does not require solving the recommendation problem for all the users if it wishes to update the recommendations for only a single active. Further, BPRS computes the recommendations for each user with linear complexity and without requiring a training period. Via computer simulations (using the 100K MovieLens dataset), we verify that BPRS iteratively reduces the error in the predicted ratings of the users until it converges. Finally, we confirm that BPRS is comparable to the state of art methods such as Correlation-based neighborhood model (CorNgbr) and Singular Value Decomposition (SVD) in terms of rating and precision accuracy. Therefore, we believe that the BP-based recommendation algorithm is a new promising approach which offers a significant advantage on scalability while providing competitive accuracy for the recommender systems.**

I. Introduction

Today, the quantity of available information grows rapidly, overwhelming consumers to discover useful information and filter out the irrelevant items. Thus, the user is confronted with a big challenge of finding the most relevant information or item in the short amount of time. Recommender systems are aimed at addressing this overload problem, suggesting to the users those items that meet their interests and preferences. More generally, recommender systems can learn about user preferences and profile over time, based on data mining algorithms, and automatically suggest products (from a large space of possible options) that fit the users' needs. Hence, it is foreseeable that the social web is going to be driven by these recommender systems.

However, there are certain challenges to design scalable, accurate and dependable recommender systems. The available data for the recommender systems is incomplete, uncertain, inconsistent and/or intentionally-contaminated. Further, since new data (ratings) becomes available continuously, recommendations need to be updated in frequent intervals causing computational limitations for large-scale systems. Latent factor models (such as Matrix Factorization) have proven to be the most accurate method in the Root Mean Square Error (RMSE) sense. However, most existing and highly popular Matrix Factorization-based recommender algorithms are shown to be prone to malicious behavior [1] and they have scalability issues. In other words, they fall short of incorporating the attack profiles and the extra noise generated by the malicious users. Further, each new update (using the most recent data or ratings) for a particular active user requires to solve the entire problem for every user in the system. Hence, new research needed to focus on algorithms which meet these challenges and provide scalable, accurate and dependable recommender systems.

In this paper we introduce the first application of Belief Propagation (BP), an iterative probabilistic algorithm, to solve the recommendation problem. We have applied BP to trust and reputation systems in our previous work [2]. In such systems, BP is used to solve the inference problem for finding the global reputation of service providers in a network based on the previous ratings of the users. The main difference between trust and reputation systems and recommender systems is that in the former one the inference problem has to be solved globally but in the latter one, the inferences are local and specific for each user.

The key observation we make is that recommender systems deal with complicated global functions of many variables (e.g., users and items). By using a factor graph, we can obtain a qualitative representation of how the users and items are related on a graphical structure. Therefore, we propose to model the recommender system on a factor graph using which our goal is to compute the marginal probability distribution functions of the variables representing the ratings to be predicted for the users. However, we observe that computing the marginal probability functions is computationally prohibitive for large-scale recommender systems. Therefore, we utilize the BP algorithm to efficiently compute these marginal probability distributions. The key role of the BP algorithm is that we can use it to compute the marginal distributions in a complexity that grows linearly with the number of nodes (i.e, users/items).

Hereafter, we refer to our scheme as the "Belief Propagation Based Iterative Recommender System" (BPRS). BPRS has several prominent features. First, it does not require to solve the problem for all users if it wishes to update the predictions for only a single active user and it does not require a training period to utilize the most recent data (ratings).

Second, its complexity remains linear per single user, making it very attractive for large-scale systems. Therefore, it can update the recommendations for each active (online) user instantaneously using the most recent data (ratings). Further, we show that BPRS provides comparable usage prediction and rating prediction accuracy to other popular methods such as the Correlation-based neighborhood model (CorNgbr) and Singular Value Decomposition (SVD). Therefore, we are very optimistic that this work promises a new direction for the recommender systems which will be scalable, accurate, and resilient to attacks.

The rest of this paper is organized as follows. In the rest of this section, we summarize the related work. In Section II, we describe the proposed BPRS in detail. Next, in Section III, we evaluate BPRS via computer simulations using the MovieLens dataset. Finally, Section IV concludes the paper.

A. Related Work

Recommender systems [3] can be classified into two main categories: i) content-based filtering [4] 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 [5] 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 fall further into two general classes: memory-based [6] and model-based algorithms [7], [8]. Model-based algorithms include methods exploiting Singular Value Decomposition (SVD), Principal Component Analysis (PCA) and Maximum Margin Matrix Factorization (MMMF) techniques [9], [10].

The application of Bayesian networks and message passing algorithms for recommender systems is also studied in the past [11], [12]. In [11], the message passing technique is used to determine the latent factors of the users and items (as an alternative to SVD). In [12], because of the fuzziness associated with the ambiguity in the description of the ratings, a (non-iterative) inference is proposed among the users to remove this ambiguity. The key difference between our approach and the other message passing-based methods is that, we describe the recommendation problem as computing marginal likelihood distributions from complicated global functions of many variables and to use Belief Propagation (BP) to find them. This is inspired by successful applications of BP algorithms in various fields such as decoding of error correcting codes [13], Artificial Intelligence [14], and reputation systems [2].

II. Belief Propagation for Recommender Systems

Belief Propagation (BP) [13], [14] is a message passing algorithm for performing interface on graphical models (e.g., factor graphs, Bayesian networks, Markov random fields). It has demonstrated empirical success in numerous applications including LDPC codes, turbo codes, free energy approximation, and satisfiability. BP is a method for computing marginal distributions of the unobserved nodes conditioned on the observed ones.

Our objective is to formulate the recommendation problem as making statistical inference about the ratings of users for unseen items based on observations. That is, given the past data evidence, what would be the likelihood (probability) that the rating takes a particular value? Here, the probability is the degree of belief to which the prediction of the rating is supported by the available evidence. This requires finding the marginal probability distributions of the variables representing the ratings of the items to be predicted conditioned on some observed preferences.

We assume two different sets in the system: i) the set of users $\mathbb{U}$ and ii) the set of items (products) $\mathbb{I}$. Users provide feedbacks, in the form of ratings, about the items for which they have an opinion. The main goal is to provide accurate recommendations for every user by predicting the ratings of the user for the items that he/she has not rated before (unseen item). Here, we consider an arbitrary user z (referred as the active user) and compute the prediction of ratings for user z for unseen items. We assume u users and s items in the system (i.e., $|\mathbb{U}| = u$ and $|\mathbb{I}| = s$). Let $\mathbb{G}_z = \{G_{zj} : j \in \mathbb{I}\}$ be the collection of variables representing the ratings of the items to be predicted for the active user z. Note that a subset of these variables are already known as the corresponding items were rated by user z. Hence, they do not require any prediction. Let also $\mathbb{R}_z = \{R_{zi} : i \in \mathbb{U}\}$ be the confidence of the system on the users for their ratings' reliability, given the active user is z. Further, we let T_{ij} represent the rating provided previously by user i about the item j. We denote $\mathbb{T}$ as the $s \times u$ item-user matrix that stores these ratings, and $\mathbb{T}_i$ as the set of ratings provided by the user i. We note that some rating entries could be missing (attributed to unseen items). To be consistent with the most of existing recommender systems, we assume that the rating values are integers from the set $\Upsilon = \{1, 2, 3, 4, 5\}$.

The recommendation problem can be viewed as finding the marginal probability distributions of each variable in $\mathbb{G}_z$, given the observed data (i.e., existing ratings and the confidence of the system for the user's ratings). There are s marginal probability functions, $p(G_{zj}|\mathbb{T}, \mathbb{R}_z)$, each of which is associated with a variable G_{zj}; the predicted rating of item j for user z. We formulate the problem by considering the global function $p(\mathbb{G}_z|\mathbb{T}, \mathbb{R}_z)$, which is the joint probability distribution function of the variables in $\mathbb{G}_z$ given the rating matrix and the confidence of the system for the user's ratings. Then, clearly, each marginal probability function $p(G_{zj}|\mathbb{T}, \mathbb{R}_z)$ may be obtained as follows:

$$p(G_{zj}|\mathbb{T}, \mathbb{R}_z) = \sum_{\mathbb{G}_z \setminus \{G_{zj}\}} p(\mathbb{G}_z|\mathbb{T}, \mathbb{R}_z), \tag{1}$$

where the notation $\mathbb{G}_z \setminus \{G_{zj}\}$ implies all variables in $\mathbb{G}_z$ except G_{zj}.

Unfortunately, the number of terms in (1) grows exponentially with the number of variables, making the direct computation infeasible for large-scale systems. However, we propose to factorize (1) to local functions f_i using a factor graph and utilize the BP algorithm to calculate the marginal

probability distributions in linear complexity. A factor graph is a bipartite graph containing two sets of nodes (corresponding to variables and factors) and edges incident between two sets. Following [13], we form a factor graph by setting a variable node for each variable G_{zj}, a factor node for each function f_i, and an edge connecting variable node j to the factor node i if and only if G_{zj} is an argument of f_i.

We arrange the collection of the users and items together with the ratings provided by the users as a factor graph $g(\mathbb{U}, \mathbb{I})$. Then, since we consider the particular active user z, the factor graph is reduced to $g(\hat{\mathbb{U}}, \mathbb{I})$ (as in Fig. 1) by only keeping the users that are connected to z via a path of length at most two in $g(\mathbb{U}, \mathbb{I})$ (i.e., the users who rated at least one item that is also rated by z) and removing all the other user nodes from the graph together with their edges. In this representation, each user corresponds to a factor node in the graph, shown as a square and each item is represented by a variable node shown as a hexagon. Further, each rating is represented by an edge from the factor node to the variable node. Hence, if a user i ($i \in \hat{\mathbb{U}}$) has a rating about item j ($j \in \mathbb{S}$), we place an edge with value T_{ij} from the factor node i to the variable node representing item j. Eventually, the $g(\hat{\mathbb{U}}, \mathbb{I})$ graph has $|\hat{\mathbb{U}}| = \hat{u}$ users and $|\mathbb{I}| = s$ items.

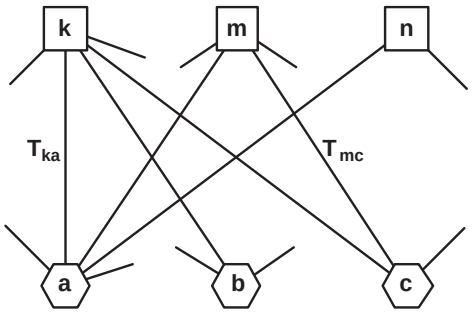

Fig. 1. Graphical representation of the scheme from user z's point of view.

Next, we suppose that the global function $p(\mathbb{G}_z|\mathbb{T}, \mathbb{R}_z)$ factors into products of several local functions, each having a subset of variables from $\mathbb{G}_z$ as arguments as follows:

$$p(\mathbb{G}_z|\mathbb{T}, \mathbb{R}_z) = \frac{1}{Z} \prod_{i \in \hat{\mathbb{U}}} f_i(\mathcal{G}_{zi}, \mathbb{T}_i, R_{zi}), \tag{2}$$

where Z is the normalization constant and $\mathcal{G}_{zi}$ is a subset of $\mathbb{G}_z$. Hence, in the graph representation of Fig. 1, each factor node is associated with a local function and each local function f_i represents the probability distributions of its arguments given the confidence of the system for the associated user and the existing ratings of the associated user.

We now describe the message exchange between a user k and an item a (in Fig. 1) provided that the active user is z in BPRS. We clarify that all the messages are formed by the algorithm that is ran in the central authority. We represent the set of neighbors of the variable node a and the factor nodes k and z (in $g(\hat{\mathbb{U}}, \mathbb{I})$) as $\mathbf{N_a}$, $\mathbf{N_k}$, and $\mathbf{N_z}$, respectively (neighbors of an item are the set of users who rated the item while neighbors of a user are the items which it rated). Further, let $\Xi = \mathbf{N_a} \backslash \{k\}$ and $\Delta = \mathbf{N_k} \backslash \{a\}$. Let $G_{zj}^{(\nu)}$ and $R_{zi}^{(\nu)}$ be the value of variable G_{zj} and system's confidence on user i at the iteration ν of the algorithm, respectively. The message $\lambda_{k \to a}^{(\nu)}(G_{za}^{(\nu)})$ (from factor node k to the variable node a) denotes the relative probabilities that $G_{za}^{(\nu)} = \ell$ ($\ell \in \Upsilon$) at the ν^{th} iteration, given T_{ka} and $R_{zk}^{(\nu-1)}$. On the other hand, $\mu_{a \to k}^{(\nu)}(G_{za}^{(\nu)})$ (from variable node a to the factor node k) denotes the probability that $G_{za}^{(\nu)} = \ell$ ($\ell \in \Upsilon$) at the ν^{th} iteration.

The message from the factor node k to the variable node a at the ν^{th} iteration is formed using the principles of the BP as

$$\lambda_{k \to a}^{(\nu)}(G_{za}^{(\nu)}) = \sum_{\mathcal{G}_{zk}^{(\nu)} \backslash \{G_{za}^{(\nu)}\}} f_k(\mathcal{G}_{zk}^{(\nu)}, \mathbb{T}_k, R_{zk}^{(\nu-1)}) \prod_{x \in \Delta} \mu_{x \to k}^{(\nu-1)}(G_{zx}^{(\nu)}), \tag{3}$$

where $\mathcal{G}_{zk}$ is the set of variable nodes which are the arguments of the local function f_k at the factor node k. This message transfer is illustrated in the right half of Fig. 2. Further, $R_{zk}^{(\nu-1)}$ is a value between zero and one and can be calculated as follows:

$$R_{zk}^{(\nu-1)} = 1 - \frac{1}{\rho |\mathbf{N_k}|} \sum_{i \in \mathbf{N_k}} \sum_{x \in \Upsilon} |T_{ki} - x| \mu_{i \to k}^{(\nu-1)}(x). \tag{4}$$

The above equation can be interpreted as one minus the average inconsistency of user k calculated by using the messages it received from all its neighbors. 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 Υ. Thus, the reliability of users (in their ratings) is measured based on the messages formed by the algorithm. Using (3) and assuming that the predicted ratings in set $\mathcal{G}_{zk}$ are independent from each other at each intermediate step (to reduce the computational complexity), it can be shown that

$$f_k(\mathcal{G}_{zk}^{(\nu)}, \mathbb{T}_k, R_{zk}^{(\nu-1)}) = \prod_{i \in \mathbf{N_k}} f_k(G_{zi}^{(\nu)}, \mathbb{T}_k, R_{zk}^{(\nu-1)}). \tag{5}$$

Thus, the message in (3) becomes

$$\lambda_{k \to a}^{(\nu)}(G_{za}^{(\nu)}) = f_k(G_{za}^{(\nu)}, \mathbb{T}_k, R_{zk}^{(\nu-1)}) \times \Big\{ \sum_{\mathcal{G}_{zk}^{(\nu)} \backslash \{G_{za}^{(\nu)}\}} \Big[\prod_{i \in \mathbf{N_k} \backslash \{a\}} f_k(G_{zi}^{(\nu)}, \mathbb{T}_k, R_{zk}^{(\nu-1)}) \prod_{x \in \Delta} \mu_{x \to k}^{(\nu-1)}(G_{zx}^{(\nu)}) \Big] \Big\}. \tag{6}$$

Since the second part of (6) is a constant, $\lambda_{k \to a}^{(\nu)}(G_{za}^{(\nu)}) \propto f_k(G_{za}^{(\nu)}, \mathbb{T}_k, R_{zk}^{(\nu-1)})$, and hence, $\lambda_{k \to a}^{(\nu)}(G_{za}^{(\nu)}) \propto p(G_{za}^{(\nu)}|T_{ka}, R_{zk}^{(\nu-1)})$, where

$$p(G_{za}^{(\nu)} = \ell | T_{ka}, R_{zk}^{(\nu-1)}) =$$

$$\begin{cases} \left[R_{zk}^{(\nu-1)} + (1 - R_{zk}^{(\nu-1)}) \times \frac{|\kappa_a^z(\ell)|+1}{\sum_{h \in \Upsilon} [|\kappa_a^z(h)|+1]} \right] & \text{if } T_{ka} = \ell \\ \\ \left[(1 - R_{zk}^{(\nu-1)}) \times \frac{|\kappa_a^z(T_{ka})|+1}{\sum_{h \in \Upsilon} [|\kappa_a^z(h)|+1]} \right] & \text{if } T_{ka} \neq \ell. \end{cases} \tag{7}$$

Here, κ_a denotes the genre (i.e., type) or the set of genres of item a. Further, $|\kappa_a^z(h)|$ is the number of items in the same genre as κ_a which are previously rated as h by the active user z. The way we compute the probabilities in (7) resembles the belief/plausibility concept of the Dempster-Shafer Theory [15]. Given $T_{ka} = 1$, $R_{zk}^{(\nu-1)}$ can be viewed as the belief of user k that $G_{za}^{(\nu)}$ is one (at the ν^{th} iteration). In other words, in the eyes of user k, $G_{za}^{(\nu)}$ is equal to one with probability $R_{zk}^{(\nu-1)}$. Thus, $(1 - R_{zk}^{(\nu-1)})$ corresponds to the uncertainty in the belief of user k. In order to remove this uncertainty and express $p(G_{za}^{(\nu)} | T_{ka}, R_{zk}^{(\nu-1)})$ as the probabilities that $G_{za}^{(\nu)}$ is ℓ ($\ell \in \Upsilon$), we distribute the uncertainty among the possible outcomes (one to five) in proportion to the histogram of the ratings provided by the active user z for the items in the same genre as κ_a. That is, if the active user previously provided high ratings for the items in the same genre as κ_a, then we distribute most of the uncertainty to the higher ratings in proportion to the rating histogram of the active user for the items in the same genre as κ_a. Similarly, if the active user previously provided low ratings for the items in the same genre as κ_a, we distribute most of the uncertainty to the lower ratings. Therefore, from user k's point of view, G_{za} is equal to one with probability $R_{zk}^{(\nu-1)} + (1 - R_{zk}^{(\nu-1)}) \times \frac{|\kappa_a^z(1)|+1}{\sum_{h \in \Upsilon} [|\kappa_a^z(h)|+1]}$. On the other hand, it is equal to ℓ ($\ell \neq 1$) with probability $(1 - R_k^{(\nu-1)}) \times \frac{|\kappa_a^z(\ell)|+1}{\sum_{h \in \Upsilon} [|\kappa_a^z(h)|+1]}$. We note that the above discussion assumed $T_{ka} = 1$ and similar statements hold for the cases when $T_{ka} = 2, 3, 4, 5$. It is worth clarifying that, as opposed to the Dempster-Shafer Theory, we do not combine the beliefs of the users. Instead, we consider the belief of each user individually and calculate probabilities that $G_{za}^{(\nu)}$ being ℓ ($\ell \in \Upsilon$) in the eyes of each user as in (7). We note that if the active user z did not rate any items from this particular genre (κ_a), we distribute the uncertainty in proportion to the average rating of user z (for the items it previously rated) ($A_z = \frac{\sum_{i \in \mathbf{N_z}} T_{zi}}{|\mathbf{N_z}|}$). The above computation in (7) must be performed for every neighbors of each factor node. This finishes the first half of the ν^{th} iteration.

In the second half of the ν^{th} iteration, we calculate the message $\mu_{a \to k}^{(\nu)}(G_{za}^{(\nu)})$ by multiplying all probabilities the variable node a received from its neighbors excluding the one from the factor node k, as shown in the left half of Fig. 2. We note that the previous ratings of the active user play a key role in the algorithm. Hence, the values of those variables in $\mathbb{G}_z$ which are associated with the items already rated by the active user z are set to the corresponding ratings (i.e., $G_{zj} = T_{zj}$ if $j \in \mathbf{N_z}$). Thus, if $a \in \mathbf{N_z}$, the messages generated from the variable node a do not vary with iterations since the value of this variable node (G_{za}) is fixed based on the ratings of the active user. Therefore, the message from the variable node a to the factor node k at the ν^{th} iteration is given by

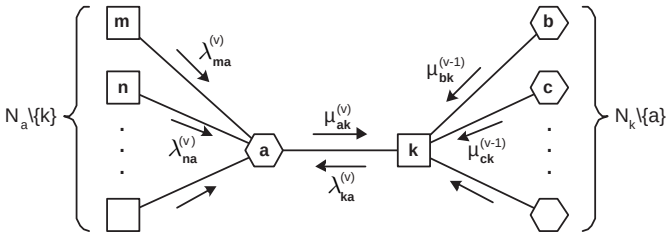

Fig. 2. Message exchange between the factor node k and variable node a.

$$\mu_{a \to k}^{(\nu)}(G_{za}^{(\nu)} = \ell) =$$

$$\begin{cases} \frac{1}{\sum_{h \in \Upsilon} \prod_{i \in \Xi} \lambda_{i \to a}^{(\nu)}(h)} \times \prod_{i \in \Xi} \lambda_{i \to a}^{(\nu)}(G_{za}^{(\nu)}) & \text{if } a \notin \mathbf{N_z} \\ \\ 1 & \text{if } a \in \mathbf{N_z} \text{ and } T_{za} = \ell \\ 0 & \text{if } a \in \mathbf{N_z} \text{ and } T_{za} \neq \ell. \end{cases} \tag{8}$$

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 $\lambda_{k \to a}^{(1)}$ by using $R_{zk}^{(0)} = \varrho$, where ϱ ($0 < \varrho < 1$) is the system's present confidence on the users for the reliability of their ratings computed at the previous execution of the algorithm. At the end of each iteration, the upper equation in (8), after following modification, is used to compute the prediction of ratings of the active user z. That is, we use the set $\mathbf{N_a}$ instead of Ξ in (8) to compute $\mu_a^{(\nu)}(G_{za}^{(\nu)})$ for every item a for which the active user z did not have any rating. Then, we set $G_{za}^{(\nu)} = \sum_{i=1}^{5} i \mu_a^{(\nu)}(i)$. The iterations stop when G_{zj} values converge for every item j.

III. Evaluation of BPRS

We evaluate the performance of BPRS using the 100K MovieLens dataset. 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 note that based on our simulations, we observed that BPRS converges, on the average, in 10 iterations. Therefore, for the remaining of this section, we either show our results during the first 10 iterations or after the 10^{th} iteration.

A. Prediction Accuracy

We evaluate the rating prediction accuracy of BPRS in terms of Root Mean Square Error (RMSE) metrics over the predicted ratings. We note that each test dataset is created by $80\%/20\%$ split of the full data into training and test data.Then, we used the training data (80% of the whole dataset) to predict the ratings in the test dataset. We computed the RMSE as below:

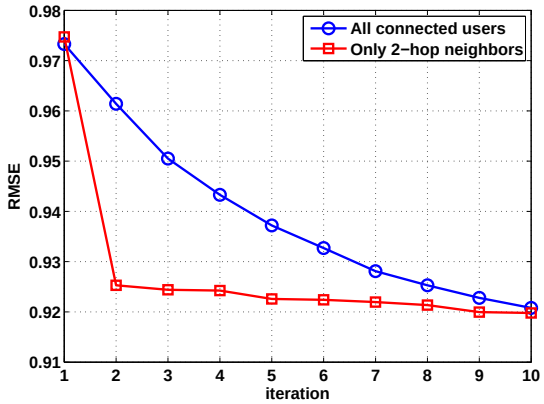

Fig. 3. Performance of BPRS in RMSE vs. number of iterations when: (i) all users and (ii) only the 2-hop neighbors are used.

$$\text{RMSE} = \sqrt{\frac{1}{|K|} \sum_{i \in \mathbb{U}, j \in \mathbb{I}} (G_{ij} - \hat{G}_{ij})^2} \qquad (9)$$

where $|K|$ is the number of ratings (to be predicted) in the test dataset, $\hat{G}_{ij}$ is the actual value of the rating provided by user i for the item j in the test dataset, and G_{ij} is the predicted rating value by the algorithm.

In Figs. 3, we show the RMSE provided by BPRS for two different scenarios: when all users connected to each active user via a path are used and when only the 2-hop neighbors of each active user are used in the algorithm. We observe that keeping only the 2-hop neighbors of each active user provides better performance in terms of RMSE. It also reduces the computational complexity as will be discussed later.

Finally, we evaluated BPRS against some popular recommendation algorithms such as: 1. MovieAvg (which computes the predicting ratings for the movies by averaging all the received ratings for each movie) with an RMSE of 1.053, 2. Correlation-based neighborhood model (CorNgbr), with an RMSE of 0.9406 [8], and 3. SVD latent factor model, with 50 factors and RMSE of 0.9046 [8]. We conclude that BPRS is comparable to existing methods such as CorNgbr and SVD in terms of rating prediction accuracy. On the other hand, BPRS generates recommendations in linear complexity for each active user and updates the recommendations for each active user instantaneously using the most recent data.

B. Computational Complexity

Assuming u users and s items in the system, we obtained the computational complexity of BPRS (in the number of multiplications) as $\mathbf{max}(\mathbb{O}(cs), \mathbb{O}(cu))$ per each active user, 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. Further, as we discussed before, BPRS converges, on the average, in 10 iterations. Hence, we did not include the number of iterations in the complexity measure as it only introduces a small constant in front of the total complexity. This result indicates that BPRS can compute the recommendations for each active user very efficiently using the most recent data (ratings). Therefore, we claim that the BP-based approach toward the recommendation problem is very promising and can result in a new class of accurate and scalable recommender systems.

IV. Conclusion

In this paper, we introduced the Belief Propagation Based Iterative Recommender System (BPRS). BPRS formulates the recommendation problem as making statistical inference about the ratings of users for unseen items based on observations. BPRS provides a complexity that remains linear per single active user, making it very attractive for large-scale systems. Further, it can update the recommendations for each active user instantaneously using the most recent data (ratings) and without solving the recommendation problem for all users. While providing these significant scalability advantages over the existing methods, we showed that BPRS also provides comparable usage prediction and rating prediction accuracy to other popular methods such as Correlation-based neighborhood model (CorNgbr) and Singular Value Decomposition (SVD).

References

[1] Z. Cheng and N. Hurley, "Effective diverse and obfuscated attacks on model-based recommender systems," *In Proceedings of the third ACM conference on Recommender systems*, pp. 141–148, 2009.

[2] E. Ayday and F. Fekri, "Application of belief propagation to trust and reputation management," in *Information Theory Proceedings (ISIT), 2011 IEEE International Symposium on*, 31 2011-aug. 5 2011, pp. 2173 –2177.

[3] P. Resnick and H. R. Varian, "Recommender systems," *Communications of the ACM*, vol. 40, pp. 56–58, March 1997.

[4] M. Balabanovic and Y. Shoham, "Fab: Content-based, collaborative recommendation," *Communications of the ACM*, vol. 40, pp. 66–72, 1997.

[5] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "GroupLens: an open architecture for collaborative filtering of netnews," *CSCW '94: Proceedings of the 1994 ACM conference on Computer supported cooperative work*, pp. 175–186, 1994.

[6] J. Herlocker, J. A. Konstan, and J. Riedl, "An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms," *Information Retrieval*, vol. 5, no. 4, pp. 287–310, 2002.

[7] T. Hofmann, "Latent class models for collaborative filtering," *In Proceedings of the 16th International Joint Conference on Artificial Intelligence*, pp. 688–693, 1999.

[8] Y. Koren, "Factorization meets the neighborhood: a multifaceted collaborative filtering model," *In Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 426–434, 2008.

[9] B. M. Sarwar, G. Karypis, J. A. Konstan, and J. T. Riedl, "Application of dimensionality reduction in recommender system - a case study," *ACM WebKDD Workshop*, 2000.

[10] N. Srebro, J. Rennie, and T. Jaakkola, "Maximum margin matrix factorizations," *In Proceedings of Advances in Neural Information Processing Systems (NIPS)*, 2005.

[11] D. H. Stern, R. Herbrich, and T. Graepel, "Matchbox: large scale online bayesian recommendations," *In Proceedings of the 18th international conference on World wide web*, pp. 111–120, 2009.

[12] L. M. de Campos, J. M. Fernández-Luna, and J. F. Huete, "A collaborative recommender system based on probabilistic inference from fuzzy observations," *Fuzzy Sets Syst.*, vol. 159, pp. 1554–1576, June 2008.

[13] F. Kschischang, B. Frey, and H. A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Transactions on Information Theory*, vol. 47, pp. 498–519, Feb. 2001.

[14] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers, Inc., 1988.

[15] G. Shafer, *A Mathematical Theory of Evidence*. Princeton University Press, Princeton, N.J., 1976.