Distributed Elastic Net Regularized Blind Compressive Sensing for Recommender System Design

Anupriya Gogna
IIIT-Delhi
Delhi, INDIA
anupriyag@iiitd.ac.in

Angshul Majumdar
IIIT-Delhi
Delhi, INDIA
angshul@iiitd.ac.in

ABSTRACT

Design of recommender system following the latent factor model is widely cast as a matrix factorization problem yielding a rating matrix, which is a product of a dense user and a dense item factor matrices. A dense user factor matrix is a credible assumption as all users are expected to have some degree of affinity towards all the latent factors. However, for items it's not a reasonable supposition as no item is expected to possess all the traits (factors). In this work, we propose a matrix factorization model which yields a dense user but a sparse item factor matrix; having equivalence to Blind Compressive Sensing (BCS) formulation. Basic BCS framework is augmented with an added elastic net regularization term. The addition helps in capturing correlation between different item latent factors. Despite the efficiency of matrix factorization approach, it's not feasible to apply the techniques for very large datasets (rating matrices). For this purpose, we employ Divide and Combine (DnC) approach – wherein proposed method is applied to distinct subsets of the rating matrix simultaneously and resulting estimates combined to yield the final result. The (randomized) DnC approach retains the convergence guarantees of matrix factorization. Experiments were conducted on real world Movielens dataset and our technique was compared against popular matrix factorization methods. The results indicate the superiority of our method in terms of both accuracy and speed.

Keywords

Blind compressive sensing, collaborative filtering, elastic net regularization, latent factor model.

1. INTRODUCTION

Information overload on the internet regards to the number of items, services, service providers and even reviews makes the job of finding the desired cumbersome for any customer. With the advent of Recommender Systems (RS), in 1990's [30], [36], this task is considerably eased. An efficient Recommender System helps both the customers – by providing relevant suggestions, as well as e-commerce portals (like Amazon, Flipkart) – by increasing their popularity and hence revenue. Task of a RS is to predict a user's choice based on his/her past history and make relevant recommendation of products and services for future.

Methods for design of RS can be classified on the basis of information utilized and technique adopted for rating prediction into – Content based, Collaborative filtering and hybrid techniques [1]. Content based methods [32] are based on finding similarity/match between user's choice profile and the content description of items. Collaborative filtering (CF) techniques [18], [14] rely on either implicit ratings – inferred from users past behavior such as browsing history, or on explicit ratings – ratings given by users on a small subset of items. They are the most widely used and efficient means of design of recommender systems. Unlike content based methods, they do not require any explicit characterization of items or users; which might not be always feasible. Hybrid schemes employ a combination of both [28].

CF methods can be further subdivided into memory based and model based approaches. Memory based methods [35], [36] are primarily neighborhood based strategies. They scan the entire rating matrix to find users with high similarity (measured based on ratings on commonly rated items) to the target user. Predicted rating for an item is just a linear combination of ratings given by similar users on the concerned item. The approach can be extended to work on item similarity rather than user similarity [22]. These methods are more intuitive, but lack the desired speed of computation; due to large size of the rating matrix. Also, the sparsity of rating matrix makes finding similar users difficult at times, which prohibits predicting ratings especially for a new user – the cold start problem [1].

Model based methods [39], [42] on the other hand, construct a model from existing dataset and subsequently use it for rating prediction. The lower dimension of the model viz-a-viz original database makes them suitable for faster online computations. Also, they are able to provide better coverage and prediction accuracy than their memory-based counterparts [1]. Several model based approaches have been studied such as Bayesian probabilistic modelling [37], cluster based methods [42], and latent factor models [19], [39].

Latent factor models have gained tremendous popularity over the past decade. These models rest on the premise that a user's choice of an item is governed by the traits possessed by the item and the user's affinity towards those characteristics. Every user can be profiled as vector of his/her affinity to certain characteristics or latent factors. Similarly, every item can also be profiled by a vector describing the extent to which it possesses those latent factors. User's rating on an item are a result of interaction between these latent factor vectors.

The 20th International Conference on Management of Data (COMAD),
17th-19th Dec, 2014 at Hyderabad, India.

Latent factor based approach has been conventionally cast as a matrix factorization problem – representing the rating matrix as a product of user and item latent factor matrices [19]. Recently, in some works [20], [38], the latent factor model has been cast as a (low rank) matrix completion task. It's a convex formulation unlike matrix factorization which is bilinear and hence non-convex. But, use of singular value decomposition makes these algorithms computationally too intensive to allow wide spread application in RS design.

Our proposed approach is based on matrix factorization formulation, which recovers the latent factor matrices for users and items. Existing works on matrix factorization aims to recover a dense user and item latent factor matrices [34]. It's realistic to expect a dense user latent factor matrix, as all users will have some degree of affinity towards all latent factors. However, such a scenario is not correct with respect to items. For example, consider the case of a Music recommender system. A user will have certain degree of interest towards all forms, be it Bollywood, Ghazals or Rap. Similarly, he/she may have a favourite singer, but will not be averse to listening to others. All this will translate into dense latent factor vectors for the users. However, a song cannot simultaneously belong to all genres or be sung by all singers. Thus the latent factor vector for any song, will have very few non-zero values – indicating possession of a few of the entire list of latent factors. Following this proposition, we formulate a matrix factorization approach that promotes recovery of the rating matrix as a product of a dense user latent factor matrix and a sparse item latent factor matrix. Our formulation shows equivalence to Blind Compressive Sensing (BCS) framework [11] in signal processing.

In addition to being sparse, item latent factors also exhibit some correlation amongst themselves. For example, an album by Lady Gaga will invariably be Pop. This correlation can be captured by an elastic net [45] type penalty term added to our base matrix factorization formulation. Thus our framework resembles a BCS formulation with an additional elastic net penalty.

The application of any matrix factorization algorithm to huge datasets (with rating matrix dimension exceeding tens of thousands) is not a feasible scenario. In [26] authors proposed a Divide and Conquer Approach which can be used to apply MF algorithms to huge datasets. We extend the approach to our formulation in order to generate predictions for very big datasets.

The results obtained using our algorithm are compared against those obtained using existing state of the art formulations. Our method yields better results than the techniques compared against with regards to both recovery accuracy and execution time – important aspects of RS design.

The rest of the paper is organized as follows. Section 2 provides brief description of work done in related areas. Our proposed formulation is elaborated in section 3. Section 4 includes the experimental setup and results. Conclusion and future work are presented in section 5.

2. LITERATURE SURVERY AND PRELIMINARIES

2.1 Latent Factor Model

Actual ratings available in the database are influenced by not just the liking of a user towards the traits possessed by an item, but are also impacted by certain biases embedded in both users and items. If we consider a user who is a movie enthusiast, he will probably be fervent about all movies and rate all of them generously. Such a user ends up having a positive user bias. Similarly, a movie which is a big Oscar Awardee will tend to get higher ratings by almost all users, inflicting it with a positive item bias.

The bias terms constitute the baseline estimate which can be modeled as [19]

$$baseline(u_a, i_b) = m_g + b_u(u_a) + b_i(i_b) \quad (1)$$

where, m_g is the global mean, $b_u(u_a)$ is the user bias for user 'a' and $b_i(i_b)$ is the item bias for item 'b'.

The interaction component of the actual ratings, i.e. excluding the baseline estimates, is what can be modelled as in terms of user's affinity to traits possessed by the item. The challenge in RS design is modelling the interaction component; baseline estimation is easy. Modelling this component using latent factor based design rests on the suggestion that that a users' rating of an item is a function of his/her affinity towards the traits (latent factors) possessed by the target item. For example, consider the case of a book recommendation system. Each user's choice of a book can be defined in terms of choice of category (fiction/nonfiction etc.), author and certain other related features. Similarly a book will have these characteristics (latent factors) to varying extent. The interaction between the user and item can be modelled as the interaction between these feature vectors for books and users as in (2)

$$interaction(u_a, i_b) = \left\langle f_{u_a}, f_{i_b} \right\rangle \quad (2)$$

where, f denotes the latent factor vector. Actual ratings can be considered as a combination of interaction and baseline measures.

$$R(u_a, i_b) = baseline(u_a, i_b) + interaction(u_a, i_b) \quad (3)$$

Extending (2) and (3) to entire rating matrix, we can model the rating matrix, R, as

$$R = m_g I + B_u + B_i + F_U \times F_I \quad (4)$$

where, B_u / B_i *and* F_U / F_I are the matrix counterpart of the user/item bias and latent factor vectors respectively.

The observed rating matrix Y given by

$$Y = M(R) \quad (5)$$

where, M is the masking/subsampling operator. Only a small percentage of total ratings are available in the database, i.e. Y is extremely sparse. The task in Collaborative filtering is to predict the missing ratings and fill in the rating matrix.

Most frequently used method of rating prediction using (5) is Matrix Factorization (MF) [19] - involving solving an optimization problem of the form

$$\min_{B_u, B_i, F_U, F_I} \left\| Y - M(m_g I + B_u + B_i + F_U \times F_I) \right\|_F^2 + \lambda_b \left(\|B_i\|_F^2 + \|B_u\|_F^2 \right) + \lambda_i \left(\|F_U\|_F^2 + \|F_I\|_F^2 \right) \quad (6)$$

where, λ_b and λ_i are the regularization parameters which aid in preventing over fitting of model to observed data. Equation (6) is

a non-convex formulation, but with separable variables. This enables minimizing over each of the variables alternately using alternating least squares [3] or stochastic gradient descent algorithm [47].

Because of efficiency of MF approach, it has received lots of attention and several works [20], [29], [39] have proposed algorithms to solve the same.

A recent addition to solving latent factor model based formulations is Matrix Completion (MC) approach. MC formulation aims to recover the interaction model (W) directly, instead of its factored version $F_U \times F_I$ by solving expression of the form given below

$$Q = M(W) \tag{7}$$

where, Q is the interaction component of observed set of ratings. Given the subsampling nature of masking operator M, (7) is an underdetermined linear system of equation. However, we can aim for a unique solution if we place a constraint on W [12]. In case of RS, the overall interaction component is affected by only the latent factors (which are the independent variables). As the number of latent factors (generally around 40-50), is far less than dimension of rating matrix (even reaching hundreds of thousands), W has a significantly low rank structure. Thus, we can look for the lowest rank solution. Hence, MC problem can be cast as

$$\min_W \left\| Q - M(W) \right\|_F^2 + \lambda_n rank(W) \tag{8}$$

where, λ_n is the regularizing term penalizing any deviation in W from the low rank nature. However, rank minimization is a NP-hard problem [2http], and thus any algorithm for the same has complexity of a brute force algorithm. Hence, the rank constraint can be replaced by its convex hull, nuclear norm – sum of singular values – constraint as in (9) [5].

$$\min_W \left\| Q - M(W) \right\|_F^2 + \lambda_n \left\| W \right\|_* \tag{9}$$

Nuclear norm regularization term in (9), while maintaining convexity of the formulation, promotes recovery of a low rank solution [33]. Several solvers exist for (9) [4], [25], [41]. Although MC being convex has convergence guarantees, it is not widely used in RS design. This is so because most existing solvers use Singular value Decomposition (SVD) for solving (9), which because of its high complexity, is inefficient for very large rating datasets.

2.2 Compressed Sensing

Theory of Compressed Sensing (CS) focusses on recovery of a sparse signal from its lower dimensional projections [8]. If y is the observation vector, and x is the original signal, given the (linear) projection operator A, the three are related as

$$y = Ax \tag{10}$$

If A has a dimension of $m \times n; m \ll n$, then (10) being an under determined system of equation can have infinite solutions. According to CS theory, if the signal 'x' is sparse or adequately compressible, a unique solution to (10) can be obtained [2] by looking for the sparest solution (minimizing l_0 norm), i.e. solving problem of the form

$$\min_x \left\| x \right\|_0 \ subject\ to\ y = Ax \tag{11}$$

Most signals are often not themselves sparse, but sparse in some transform domain (for example images are sparse in wavelet domain). In such a case, we can modify (11) as follows.

$$\begin{aligned} &\min_x \left\| \beta \right\|_0 \ subject\ to\ y = AD^T\beta \\ &where,\ x = D^T\beta \end{aligned} \tag{12}$$

where, D is the sparsifying dictionary.

However, (12) is NP-hard [31], with all algorithms to solve the same having complexity equal to brute force algorithms. But, non-convex l_0 norm can be approximated by its convex surrogate, the l_1 norm (13), which yields the same solution as (12) if certain conditions are met [6].

$$\begin{aligned} &\min_x \left\| \beta \right\|_1 \ subject\ to\ y = AD^T\beta \\ &where,\ x = D^T\beta \end{aligned} \tag{13}$$

Equation (13) can be put as an unconstrained convex formulation

$$\min_\beta \left\| y - AD^T\beta \right\|_2^2 + \lambda \left\| \beta \right\|_1 \tag{14}$$

where, λ is the regularization parameter. It has been shown and if A and D are incoherent and x is sufficiently sparse, solution of (12) and (14) match.

2.3 Blind Compressed Sensing

CS theory assumes the either the signal is sparse as it is, or sparse in some known transform domain i.e. sparsifying dictionary D is known a priori. But, there could be cases where that is not the case. Such problems fall under the framework of Blind Compressed Sensing (BCS) [11].

BCS formulation attempts to simultaneously recover the sparse signal and the sparsifying basis from the under sampled signal measurements. However, a robust solution to this is possible only in case of Multiple Measurement vector (MMV) setting depicted in (15). Consider multiple observation vectors stacked in columns of Y and B being the sparse coefficients matrix, then

$$Y = AD^T B \tag{15}$$

where, $Y = [y_1 \mid y_2 \mid .... \mid y_N]$ and $B = [\beta_1 \mid \beta_2 \mid .... \mid \beta_N]$

Even though, BCS shows similarity to dictionary learning, latter is an offline technique, i.e. it cannot be used for signal recovery/ reconstruction.

To obtain a unique solution, it's necessary to impose some constraint on the sparsifying basis also, alongside the sparsity constraint on the transformed signal coefficients [11]. One of the possible formulation for BCS, used in [23] for dynamic MRI reconstruction, is given in (16). It imposes a constraint on the Frobenius norm of the dictionary.

$$\min_{\beta, D}\left[\sum_{i=1}^{l}\left\|A_i(\beta D)-y_i\right\|_2^2\right]+\lambda\|\beta\|_1 \quad (16)$$

subject to $\|D\|_F^2 \le const$

2.4 Elastic-net Regularization

Classical regression model can be written as

$$y = Ax+\eta,\ \eta \sim N(0,\sigma^2) \quad (17)$$

where, y is a observed data, A is a matrix of explanatory variables and x is the unknown weight vector which explains the observation in terms of explanatory variables.

The most basic and straight forward regularization is the ridge regression [15] which solves the problem of the form

$$x_r = \min_x \|y-Ax\|_2^2+\lambda\|x\|_2^2 \quad (18)$$

Where, λ is the regularization parameter. Solving (18) promotes recovery of a dense solution because of l_2 norm penalty term.

However, in cases where only a few explanatory variables explain the entire formulation, ridge regression fails to capture the structure correctly. In such a case, we want a sparse solution (weight vector) x so that only a few explanatory variables participate in the describing the observed variable. Here comes the LASSO (Least Angle Shrinkage and Selection Operator) regularization [40].

LASSO solves (17) with the l_2 norm penalty term replaced by l_1 regularization (19)

$$x_l = \min_x \|y-Ax\|_2^2+\lambda\|x\|_1 \quad (19)$$

Use of l_1 penalty promotes recovery of a sparse weight vector.

But, in certain cases even with the desired weight vector being sparse, LASSO fails to yield the correct structure. For example, consider a case where the explanatory variables are interdependent or highly correlated. In this scenario, the correlated explanatory variables should occur together, i.e. if one of them is selected, others in the group should also be a part of selection. However, LASSO fails to capture or promote this group structure.

This is where, the role of elastic net regularization [27], [45] comes into play. It includes an additional penalty term (l_2 norm) on the weight vector into the LASSO framework. This quadratic penalty aims at selecting all the correlated variables together. Equation (20) shows elastic net regularization.

$$x_{enet} = \min_x \|y-Ax\|_2^2+\lambda_1\|x\|_1+\lambda_2\|x\|_2^2 \quad (20)$$

2.5 Divide and Conquer

Most e-commerce sites have very large database of users and items; generating a huge but extremely sparse rating matrix. Applying any algorithm to such huge matrices is almost computationally impossible because of its enormous time and space complexity.

Research has been undertaken to solve this scalability problem and several approaches have been suggested. In [46] authors proposed a parallel ALS algorithm. In this, various "labs" in parallel MATLAB work on a subset of columns of the rating matrix and the resulting estimates are shared with other "labs". [24] proposed an approach for performing nonnegative matrix factorization as a series of map and reduce steps.

Authors in [26] proposed a distributed approach for performing MF on large datasets on a distributed architecture. It follows three steps

1. Divide: Divide the observed rating matrix (Y) into sub matrices - $M_1, M_2, ..., M_N$, by splitting Y along the longer dimension.
2. Factor: Perform independent MF on each of the sub matrices, to yield partial estimates corresponding to each sub matrix - $\hat{M}_1, \hat{M}_2, ...\hat{M}_N$.
3. Combine: A technique using randomized column projection method suggested in [10] is used. One of the estimated sub matrix is selected at random, and all sub matrix estimates are projected on to its column space to yield a (low rank) estimate for the entire rating matrix. Same procedure is implemented for all the sub matrices and an average of all resulting estimates yields the final matrix factors.

This methodology is implemented on a distributed platform, with split and factorization algorithms running simultaneously on all sub matrices in parallel. It achieves a huge decrease in run time and also reduces net computational complexity.

3. PROPOSED APPROACH

3.1 Proposed Formulation

In this section, we present our novel proposition for latent factor model based design of an efficient recommender system.

For our model, we first estimate the baseline offline. Baseline estimation is done using stochastic gradient descent algorithm for solving the formulation in (21).

$$\min_{b_u, b_i} \sum_{u,i\in\Omega}\left(y_{u,i}-\left(b_u+b_i+m_g\right)\right)^2+\lambda_b\left(b_u{}^2+b_i{}^2\right) \quad (21)$$

where, Ω is the set of observed ratings.

For our design, offline baseline estimation not only reduces the online computation burden substantially, but also gives better recovery accuracy than online baseline estimation for our model. Once the baseline is computed by solving (21), the interaction part is segregated from the actual ratings. Our model is applied to this 'interaction component'. After the interaction estimate for the entire matrix is computed, baseline terms are added back to get final rating values for making relevant prediction.

Modelling of the interaction part is done following the latent factor based design approach. In line with the conventional latent factor model, we also propose to factorize the rating matrix into two sub matrices – user latent factor and item latent factor. Existing latent factor matrix factorization models solve the problem of the form

$$\min_{F_U, F_I}\|Q-M(F_U\times F_I)\|_F^2+\lambda\left(\|F_U\|_F^2+\|F_I\|_F^2\right) \quad (22)$$

where, $Q = Y - B_u - B_i - m_g I$ is the observed interaction part, F_U and F_I are the matrix composed of user and item latent factor vectors, respectively and *M* is the subsampling operator. Above optimization problem promotes recovery of a dense item and a dense user latent factor matrix i.e. latent factor vectors for both users and items are dense, with non-zero values for all latent factors.

A user latent factor vector can be reasonably assumed to be dense, but the same cannot be applied to the item's latent factors. Let us consider the case of a Restaurant recommender system. In this case the relevant latent factors (defining characteristics) include those related to cuisine, location, price, ambience, service and alike. A user might have a liking for continental cuisine but will be completely against Indian food. Similarly a user having an affinity for fine dining restaurants, might not be opposed to going to a self-service café. Hence, it can be safely presumed that a user's affinity to almost all factors, to varying degree, will translate into a dense user factor matrix. On the other hand, if a restaurant is fine dining, it cannot have self-service. Similarly a bakery won't serve Indian cuisine. Hence, if we construct a restaurant's latent factor vector, it will have a large number of zeros, as no restaurant can possess all latent factors concurrently. Hence, the dense latent factor assumption does not hold true for items.

In contrast to previous works, we propose to factorize the rating (interaction) matrix into a dense user factor matrix and a sparse item latent factor matrix. The problem can be mathematically formulated as

$$\min_{F_U, F_I} \left\| Q - M(F_U \times F_I) \right\|_F^2 + \lambda_u \left\| F_U \right\|_F^2 + \lambda_i \left\| vec(F_I) \right\|_1 \quad (23)$$

Where, $vec(F_1)$ is the vectorized (column concatenated) form of item factor matrix. Equation (23) has equivalence to blind compressed sensing formulation (16) discussed in the previous section. The Frobenius norm penalty on the user matrix is in accordance with the constraint on dictionary in BCS framework. Together with the sparsity constraint on item factor matrix it provides conditions for recovery of a unique solution.

Our above formulation (23) captures the sparse nature of item latent factor vectors but fails to capture the dependence of these factors on each other. Carrying on with the case of a restaurant RS, a fine dining restaurant will inevitably be expensive, as also will be a restaurant in a star property. It can be observed that certain latent factors are linked together, i.e. they will usually occur together. Hence, item latent factor vectors follows a group sparse structure. Equation (23) may select certain factors, but keep related latent factors as zero, as it fails to exploit their correlation. But, the lack of knowledge about which factor (position of latent factor in the entire vector) corresponds to which trait prevents us from imposing a strict group sparse penalty.

Elastic net regularization allows us to embed this group nature and inter factor dependence into the matrix factorization model. Incorporating elastic net type penalty term into our previous formulation we get

$$\min_{F_U, F_I} \left\| Q - M(F_U \times F_I) \right\|_F^2 + \lambda_u \left\| F_U \right\|_F^2 + \lambda_i \left\| vec(F)_I \right\|_1 + \lambda_{enet} \left\| F_I \right\|_F^2 \quad (24)$$

Inclusion of both l_1 and l_2 norm penalty on the item latent factor matrix promotes recovery of a sparse solution with correlated factors being chosen together.

To enable efficient implementation of our approach to very big datasets, we place it in the structure of Divide and Combine methodology [26]. The observed interaction (component) matrix (*Q*) is split into several disjoint sub parts - smaller dimension matrices - by splitting randomly along the longer matrix dimension. Our latent factor based approach is then applied to each of the sub matrices. After all the partial estimates are obtained, they are combined in accordance with the procedure outlined in preceding sections. This approach helps us in efficient parallel implementation of our algorithm on a distributed platform.

3.2 Algorithm Design

In this section we present the design of an algorithm with low computational complexity to solve (24). Our algorithm is based on the principle of Majorization Minimization (MM) [9].

Majorization Minimization scheme proposes to map computationally intensive optimization problems into much simpler and effective iterative procedures. We briefly discuss the MM approach before using it for our algorithm design.

In several applications, we are required to solve least square optimization of the form

$$\min_x \left\| y - Ax \right\|_2^2 \quad (25)$$

The solution to above is given by $x = \left(A^T A\right)^{-1} A^T y$.

If the size of signal *x*, is very large, computation of pseudo inverse of *A* is computationally very intensive – forming the algorithm's bottleneck. MM technique eliminates this bottleneck. It involves replacing the existing function $h(x) = \min_x \left\| y - Ax \right\|_2^2$, by another function $g(x)$ which is much simpler to minimize. It is essentially a majorizer of $h(x)$ and the two are related as follows

- $g(x_k) = h(x_k)$
- $g(x) \geq h(x) \forall x$

As shown in fig. 1, new function is defined such that it touches the existing function at the point of definition, and lies above it otherwise.

For (25), at an initial (guess of minima) point x_k , $g(x_k)$ can be defined as

$$g(x_k) = \left\| y - Ax \right\|_2^2 + (x - x_k)^T \left(\gamma I - A^T A\right)(x - x_k) \quad (26)$$

Under the constraint that $\gamma \geq \max eig\left(A^T A\right)$, (26) will satisfy conditions for majorizer. Now, instead of minimizing our original function, (26) is minimized. Its minima forms the new point of definition $x_{k+1} = \min_x g(x)$.

After some mathematical manipulations, minimization of $g(x)$ can be converted into two iterative steps

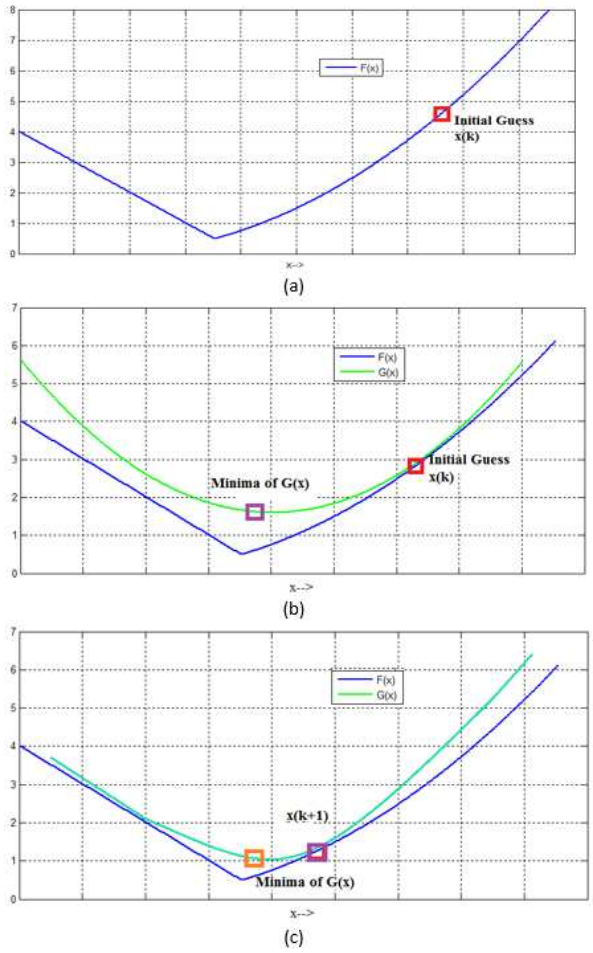

Figure 1. Majorization – Minimization Approach

$$\textit{Step 1: } b = x_k + \frac{1}{\gamma} A^T (y - Ax_k) \tag{27a}$$

$$\textit{Step 2: } \min_x \|b - x\|_2^2 \tag{27b}$$

Hence, the solution to (25) no longer requires pseudo inverse computations.

Before extending the same methodology to our formulation (24), we apply Alternating Direction Method of Multipliers (ADMM) to (24), to split this bilinear formulation into two convex sub problems (as both variables are separable) – one optimizing over F_U and other over F_I .

Sub problem 1:

$$\min_{F_U} \|Q - M(F_U \times F_I)\|_F^2 + \lambda_u \|F_U\|_F^2 \tag{28}$$

Sub problem 2:

$$\min_{F_I} \|Q - M(F_U \times F_I)\|_F^2 + \lambda_i \|vec(F)_I\|_1 + \lambda_{enet} \|F_I\|_F^2 \tag{29}$$

Sub problem 1 can be simplified using MM approach discussed above into following steps

Initialization: *No. of partitions for DnC - n*
Split rating matrix into n subparts
for $i = 1:n$
  % *Perform MF on each submatrix*
    Initialize variables, $F_{U_0}, F_{I_0} = rand$
    Set maximum no. of iterations, T
    while $k < T$ *or* $obj(k) - obj(k-1) \le 1e-7$

$$D = F_{Uk} F_{Ik} + \frac{1}{\nu} M^T \left(y - M\left(F_{Uk} F_{Ik}\right)\right)$$

$$F_{Uk+1} \leftarrow \min_{F_U} \left\| \begin{pmatrix} D \\ 0 \end{pmatrix} - F_U \begin{pmatrix} F_I \\ \sqrt{\lambda_u} I \end{pmatrix} \right\|_F^2$$

$$W = F_{Uk+1} F_{Ik} + \frac{1}{\nu} M^T \left(y - M\left(F_{Uk+1} F_{Ik}\right)\right)$$

$$F_{Ik+1} \leftarrow \min_{F_I} \left\| \begin{pmatrix} W \\ 0 \end{pmatrix} - \begin{pmatrix} F_{Uk+1} \\ \sqrt{\lambda_{enet}} I \end{pmatrix} F_I \right\|_F^2 + \lambda_i \|vec(F_I)\|_1$$

    end while
end for
for $i = 1:n$
  $C\_estimate \leftarrow$ *Project estimates on column space of* $C(n)$
end for
$Net\_Estimate \leftarrow Mean\ (C\_estimate)$

Figure 2. Algorithm

$$\textit{Step 1: } \mathrm{D} = F_{Uk} F_{Ik} + \frac{1}{\nu} M^T \left(y - M\left(F_{Uk} F_{Ik}\right)\right); \nu \ge \max eig\left(M^T M\right) \tag{30}$$
$$\textit{Step 2: } \min_{F_U} \|D - F_U \times F_I\|_F^2 + \lambda_u \|F_U\|_F^2$$

Step 2 can be recast as a simple least squares optimization as

$$\min_{F_U} \left\| \begin{pmatrix} D \\ 0 \end{pmatrix} - F_U \begin{pmatrix} F_I \\ \sqrt{\lambda_u} I \end{pmatrix} \right\|_F^2 \tag{31}$$

Which can be efficiently solved using any least square solver like gradient descent.

Similarly, sub problem 2 can also benefit from MM approach and written as following iterates

$$\textit{Step 1: } \mathrm{W} = F_{Uk} F_{Ik} + \frac{1}{\nu} M^T \left(y - M\left(F_{Uk} F_{Ik}\right)\right); \nu \ge \max eig\left(M^T M\right) \tag{32}$$
$$\textit{Step 2: } \min_{F_U} \|W - F_U \times F_I\|_F^2 + \lambda_i \|vec(F)_I\|_1 + \lambda_{enet} \|F_I\|_F^2$$

Step 2 can put reformulated as in (31)

$$\min_{F_I} \left\| \begin{pmatrix} W \\ 0 \end{pmatrix} - \begin{pmatrix} F_U \\ \sqrt{\lambda_{enet}} I \end{pmatrix} F_I \right\|_F^2 + \lambda_i \|vec(F_I)\|_1 \tag{33}$$

Equation (33) can be solved using iterative soft thresholding [7].

Both the sub problems are iteratively solved till convergence criteria is satisfied, i.e. maximum number of iterations reached or objective function variation between consecutive iterations falls below the threshold (1e-7).

The complete algorithm for implementation of our formulation on big datasets is given in fig. 2.

4. EXPERIMENTAL SETUP AND RESULTS

This section describes our experimental setup for testing our novel proposition. Also, comparison with various standard matrix factorization algorithms is given in terms of mean absolute error (MAE) and execution times.

4.1 Experimental Setup

We conducted experiments on Movielens 10M dataset [48] which has been used extensively for benchmarking collaborative filtering algorithms. The rating matrix has a dimension of 71567 users and 10667 items with 10 million ratings valued 1-5. The size of rating matrix is justifies the use of divide and combine approach. We performed fivefold cross validation, the available ratings are split into five parts. Four of the parts (80% of available ratings) form the training data and the last set (20% of available data) constitutes the test set. For each of the test-train pair, 50 independent runs of the algorithms were carried out.

Baseline estimation was done offline, using (21) and the interaction part fed into our model framework. The value of regularization parameter for the same was kept at 0.001 and 250 iterations were carried out using stochastic gradient algorithm.

For our experimentation, the complete interaction (user-item rating) matrix was split into four disjoint sub matrices of almost equal dimensions – by splitting along the column. Our matrix factorization algorithm (EBCS-BD) was applied to each of the sub matrices in parallel. The value of regularization parameters were selected using L-curve method [13]. The optimum values were found to be $\lambda_i = 1e-3$, $\lambda_{enet} = 1e-2$, $\lambda_u = 1e5$. The dimensionality of the model – number of latent factors – were experimentally selected to be 50. The resulting predicted estimates were combined as per the design procedure underlined in fig. 2. After the interaction component of the ratings are predicted, baseline data computed offline is added back to recover completely filled rating matrix.

4.2 Results

Our model and design algorithm was compared against the traditional and state of the art methods for matrix factorization, namely Accelerated Proximal Gradient (APG) [41], Singular Value Thresholding (SVT) [4] and optSpace [17].

Table 1 shows the comparison between all techniques on the basis of recovery accuracy measured in terms of MAE (34).

$$MAE = \frac{\sum_{m,n} \Re_{m,n} - \hat{\Re}_{m,n}}{|\Re|} \quad (34)$$

Where, $\Re_{m,n}$ and $\hat{\Re}_{m,n}$ are the actual and predicted ratings and $|\Re|$ is the cardinality of the rating matrix $\Re$. It's the standard measure for benchmarking algorithm for recommender system design.

It can be observed that our algorithm gives better recovery accuracy than the other standard algorithms compared against. Our algorithm performs around 4% better than optSpace and around 7.5% improvement is shown with respect to SVT in terms of mean absolute error. APG algorithm shows erratic behavior and does not give 100% coverage, i.e. not all ratings can be predicted in all the cases. The values shown in table 1 are the best case values. On the other hand, our algorithm ensures 100% coverage and consistently perform well for various test sets and multiple runs. Even for best case results, our algorithm is able to achieve a decrease of 2% in MAE values over APG. By RS standards, this improvement is substantially relevant.

Table 1. Mean Absolute Error for Various Algorithms

Algorithm	Mean Absolute Error
EBCS-BD (proposed)	0.6185
APG	0.6307
OptSpace	0.6437
SVT	0.6645

Mean absolute error is a measure of overall accuracy of the algorithm. However, for each user what's important is how close the predictions to his /her actual choice are. Hence, in table 2 we show the spread of prediction error of various algorithms. Prediction error, *PE=n* indicates that the error between the actual and predicted ratings is *n*. The values shown in the table are the percentage of ratings having the stated prediction error. On this measure also, our algorithm performs the best, with most of ratings having an error of less than 2.

Table 2. Spread of Prediction error

Algorithm	PE=0	PE=1	PE=2	PE=3	PE=4
EBCS-BD	38.47	54.71	6.60	0.18	0.02
APG	37.67	52.34	8.82	1.04	0.12
OptSpace	36.67	53.10	7.99	1.99	0.14
SVT	35.40	53.46	8.71	1.27	0.16

It's important for a RS design algorithm to not just be accurate but be sufficiently fast. A faster algorithm ensures that the model can be updates more frequently and also online computation of rating can also be carried out more efficiently.

Table 3. Run Times for Various Algorithms

Algorithm	Run Times (seconds)
EBCS-BD (proposed)	170.61
APG	276.05
OptSpace	1159.89
SVT	265.74

The run times for various algorithms is shown in table 3. The times are for all three phases combined i.e., split, perform matrix factorization and combine to yield final estimate. It's evident that our algorithm is considerably faster than other algorithms. Our algorithm (because of use of MM approach) is almost 1.5 times faster than APG and SVT, nearest to it in terms of time requirement. This aids in design of an efficient recommender system.

5. CONCLUSION

In this work, we propose a novel recommender system design approach based on the latent factor model implemented on a distributed platform. Existing latent factor based models aim to

recover the rating matrix as a product of a dense item and a dense user latent factor matrix. We also propose to recover the rating matrix as a product of user and item factor matrices, but do not impose the dense structure constraint. We claim that the user latent factor matrix is dense but for the item latent factor matrix the same doesn't hold true. This is because every user will demonstrate certain degree of affinity towards all traits but, no item can concurrently possess all the traits. Hence, we promote recovery of a dense user and a sparse item latent factor matrix.

Along with this, we also argue that the various traits defining the items are not independent. This correlation and interdependence between the items is captured by use of an elastic-net regularization based penalty term. This addition promotes a group sparsity effect in the sparse item factor vector - correlated factors are selected together. We show that our proposed formulation naturally fits into the blind compressive sensing framework with an add-on elastic net penalty term.

We also derive and efficient algorithm for solving our problem formulation using Majorization minimization approach. Use of MM technique helps in breaking complex and computationally intensive optimization problem into simple iterative procedure. Thus, use of MM method greatly reduces the computational burden and run times.

Also, we employ divide and combine approach to employ our formulation efficiently on a distributed platform to very large rating matrices.

In this work we have experimented on the movielens dataset. It is shown that our algorithm outperforms other collaborative filtering techniques compared against. Our algorithm is able to achieve improved quality of prediction and a reduction in mean absolute error. Also, our algorithm using MM approach ensures that the run times for our design is much smaller than for other algorithms. Hence, our design incorporates two basic requirements of recommender systems – high accuracy and smaller execution.

6. REFERENCES

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

[2] Baraniuk, R. G. Compressive sensing. IEEE signal processing magazine, 24(4), 2007

[3] Bell, R. M., and Koren, Y. Scalable collaborative filtering with jointly derived neighborhood interpolation weights. In Seventh IEEE International Conference Data Mining, 2007. ICDM 2007, pp. 43-52

[4] Cai, J. F., Candès, E. J., and Shen, Z. A singular value thresholding algorithm for matrix completion. SIAM Journal on Optimization, 20(4), 2010, 1956-1982.

[5] Candès, E. J.,and Recht, B. Exact matrix completion via convex optimization. Foundations of Computational mathematics, 9(6), 2009, 717-772.

[6] Candes, E. J. The restricted isometry property and its implications for compressed sensing. Comptes Rendus Mathematique, 346(9), 2008, 589-592.

[7] Daubechies, I., Defrise, M., and De Mol, C. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. Communications on pure and applied mathematics, 57(11), 2004, 1413-1457.

[8] Donoho, D. L. Compressed sensing. IEEE Transactions on Information Theory, 52(4), 2006, 1289-1306.

[9] Figueiredo, M. A., Bioucas-Dias, J. M., and Nowak, R. D. Majorization–minimization algorithms for wavelet-based image restoration. IEEE Transactions on Image Processing, 16(12), 2007, 2980-2991.

[10] Frieze, A., Kannan, R., and Vempala, S. Fast Monte-Carlo algorithms for finding low-rank approximations. Journal of the ACM (JACM), 51(6), 2004, 1025-1041.

[11] Gleichman, S., & Eldar, Y. C. Blind compressed sensing. IEEE Transactions on Information Theory, 57(10), 2011, 6958-6975.

[12] Gross, D. Recovering low-rank matrices from few coefficients in any basis. Information Theory, IEEE Transactions on, 57(3), 2011, 1548-1566.

[13] Hansen, P. C., and O'Leary, D. P. The use of the L-curve in the regularization of discrete ill-posed problems. SIAM Journal on Scientific Computing, 14(6), 1983, 1487-1503.

[14] Herlocker, J.L., Konstan, J.A., Terveen, L.G., and Riedl, J.T. Evaluating collaborative filtering recommender systems. ACM Transactions on Information Systems (TOIS), 22(1), 2004, 5-53

[15] Hoerl, A. E., and Kennard, R. W. Ridge regression: Biased estimation for nonorthogonal problems. Technometrics, 12(1), 1970, 55-67.

[16] Hofmann, T. Latent semantic models for collaborative filtering. ACM Transactions on Information Systems (TOIS), 22(1), 2004, 89-115.

[17] Keshavan, R. H., and Oh, S. A gradient descent algorithm on the grassman manifold for matrix completion. arXiv preprint arXiv:0910.5260.

[18] Koren, Y., and Bell, R. Advances in collaborative filtering. In Recommender Systems Handbook, Springer US, 145-186

[19] Koren, Y., Bell, R., and Volinsky, C. Matrix factorization techniques for recommender systems. Computer, 42(8), 2009, 30-37.

[20] Lee, D. D., and Seung, H. S. Algorithms for non-negative matrix factorization. In Advances in neural information processing systems, 2001, 556-562

[21] Lee, J., Recht, B., Srebro, N., Tropp, J., & Salakhutdinov, R. (2010). Practical large-scale optimization for max-norm regularization. In Advances in Neural Information Processing Systems , 1297-1305, 2010J.D

[22] Linden, G., Smith, B., and York, J. Amazon. com recommendations: Item-to-item collaborative filtering. IEEE Internet Computing, 7(1), 2003, 76-80.

[23] Lingala, S. G., and Jacob, M. Blind compressive sensing dynamic MRI.Medical Imaging, IEEE Transactions on, 32(6), 2013, 1132-1145.

[24] Liu, C., Yang, H. C., Fan, J., He, L. W., and Wang, Y. M. Distributed nonnegative matrix factorization for web-scale dyadic data analysis on mapreduce. In Proceedings of the 19th international conference on World wide web , April 2010, 681-690

[25] Ma, S., Goldfarb, D., & Chen, L. Fixed point and Bregman iterative methods for matrix rank minimization. Mathematical Programming, 128(1-2), 2011, 321-353.

[26] Mackey, L. W., Jordan, M. I., and Talwalkar, A. Divide-and-conquer matrix factorization. In Advances in Neural Information Processing Systems, 2011, 1134-1142

[27] Majumdar, A., and Ward, R. K. Classification via group sparsity promoting regularization. In IEEE International Conference on Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. 861-864

[28] Melville, P., Mooney, R.J., and Nagarajan, R. Content-boosted collaborative filtering for improved recommendations. In AAAI/IAAI, July 2002, 187-192

[29] Mnih, A., and Salakhutdinov, R. Probabilistic matrix factorization. InAdvances in neural information processing systems, 2007, 1257-1264

[30] Mooney, R. J., Bennett, P. N., and Roy, L. Book recommending using text categorization with extracted information. In Proc. Recommender Systems Papers from 1998 Workshop, Technical Report WS-98-08, 1998

[31] Natarajan, B. K. Sparse approximate solutions to linear systems. SIAM journal on computing, 24(2), 1995, 227-234.

[32] Pazzani, M.J., and Billsus, D. Content-based recommendation systems. In The adaptive web. Springer Berlin Heidelberg, 2007, 325-341

[33] Recht, B., Fazel, M., and Parrilo, P. A. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. SIAM review,52(3), 2010, 471-501.

[34] Rennie, J. D., & Srebro, N. Fast maximum margin matrix factorization for collaborative prediction. In Proceedings of the 22nd international conference on Machine learning, August 2005, 713-719

[35] Sarwar, B., Karypis, G., Konstan, J., and Riedl, J. Item-based collaborative filtering recommendation algorithms. In Proceedings of the 10th international conference on World Wide Web, April 2001, 285-295

[36] Schafer, J. B., Konstan, J., and Riedl, J. Recommender systems in e-commerce. In Proceedings of the 1st ACM conference on Electronic commerce, New York, NY, USA, 1999, 158-169

[37] Salakhutdinov, R., and Mnih, A. Bayesian probabilistic matrix factorization using Markov chain Monte Carlo. In Proceedings of the 25th international conference on Machine learning, July 2008, 880-887

[38] Shamir, O., and Shalev-Shwartz, S. Collaborative filtering with the trace norm: Learning, bounding, and transducing, COLT, JMLR Proceedings, 19, 2011, 661-678

[39] Srebro, N., Rennie, J., and Jaakkola, T. S. Maximum-margin matrix factorization. In Advances in neural information processing systems, 2004, 1329-1336

[40] Tibshirani, R. Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society. Series B (Methodological), 1996, 267-288.

[41] Toh, K. C., and Yun, S. An accelerated proximal gradient algorithm for nuclear norm regularized linear least squares problems. Pacific Journal of Optimization, 6(15), 2010, 615-640)

[42] Yu, K., Schwaighofer, A., Tresp, V., Xu, X., and Kriegel, H. P. Probabilistic memory-based collaborative filtering. IEEE Transactions on Knowledge and Data Engineering, 16(1), 2004, 56-69.

[43] Xiong, L., Chen, X., Huang, T. K., Schneider, J. G., and Carbonell, J. G. Temporal Collaborative Filtering with Bayesian Probabilistic Tensor Factorization. In *SDM* , 10, 2010, 211-222

[44] Xue, G. R., Lin, C., Yang, Q., Xi, W., Zeng, H. J., Yu, Y., and Chen, Z. Scalable collaborative filtering using cluster-based smoothing. In Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval , August 2005, 114-121

[45] Zou, H., and Hastie, T. Regularization and variable selection via the elastic net. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 67(2), 2005, 301-320.

[46] Zhou, Y., Wilkinson, D., Schreiber, R., and Pan, R. Large-scale parallel collaborative filtering for the netflix prize. In Algorithmic Aspects in Information and Management, Springer Berlin Heidelberg, 2008, 337-348

[47] http://sifter.org/~simon/journal/20061211.html

[48] http://grouplens.org/datasets/movielens/