LATENT FACTOR MODELS FOR COLLABORATIVE FILTERING

by

ANUPRRIYA GOGNA

Under the Supervision of Dr. Angshul Majumdar

Indraprastha Institute of Information Technology Delhi
April, 2017

LATENT FACTOR MODELS FOR COLLABORATIVE FILTERING

by

ANUPRRIYA GOGNA

Submitted
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

to the
Indraprastha Institute of Information Technology Delhi
April, 2017

Certificate

This is to certify that the thesis titled "LATENT FACTOR MODELS FOR COLLABORATIVE FILTERING" being submitted by ANUPRRIYA GOGNA to the Indraprastha Institute of Information Technology Delhi, for the award of the degree of Doctor of Philosophy, is an original research work carried out by her under my supervision. In my opinion, the thesis has reached the standards fulfilling the requirements of the regulations relating to the degree.

The results contained in this thesis have not been submitted in part or full to any other university or institute for the award of any degree/diploma.

April, 2017
Dr. Angshul Majumdar

Indraprastha Institute of Information Technology Delhi
New Delhi 110 020

Acknowledgements

This thesis is a combined effort of several people who have contributed one way or the other towards its completion. The following lines are a humble effort to express my gratitude to all of them.

First thanks go to my advisor, Dr. Angshul Majumdar, who has been an embodiment of everything I expected out of my advisor. He has been a source of constant support, motivation and encouragement and I cannot thank him enough for the belief he showed in me. This thesis and my growth in the past four years are credits to his technical prowess and constant engagement in my work. He has guided me in all technical matters, while forcing me at all the right times to chart my own course.

This thesis and my PhD itself would not have seen culmination if not for my family. My parents have given me the freedom to choose, the confidence to decide and a faith that no matter what, I have them watching my back. My gratitude to my family is incomplete without a very special mention for my brother. He has been a friend, a confidante and also my thesis reviewer. He has been there for me through all stressful times and also a part of all the joyous moments. I want to take this opportunity to express my heartfelt gratitude to my family for making me what I am and being proud of it.

Another person who bore the highs and lows of this journey almost as much as I did is Shwet. He has borne all my mood swings, stress and lack of time and yet stood strong by my side through it all. His pride in even the smallest of my accomplishments made me work harder in greed of that reward.

I believe no PhD journey can ever be successful without the support of some great friends and for me that support came from Alvika, Dhananjay and Harsha. Alvika has been the closest of my friends; my one stop destination for sharing it all – happiness and sorrows. She is also the one who taught me that efforts towards excelling should know no limits. Dhananjay is my buddy, my critic and my motivator. It is the endless cups of coffee and night long conversations with him that gave me a new perspective to look at everything from work to life. Harsha, my genius friend, has been the one who was there in

everything ranging from technical discussions and review of papers/presentations to planning of mini excursion during work related travels and parties celebrating it all. And, not to forget they all reviewed my thesis at very short notice!! The bond that I have developed with them will continue to enrich my life in all the years to come.

My PhD journey and dive into the field of optimization would have been a lot more difficult had it not been for Ankita who was not only my first friend in IIIT but also my first tuition teacher. I would also like to thank my colleagues, Hemant and Naushad, who provided valuable technical inputs and an opportunity to know things beyond my immediate work.

I would also like to express my gratitude to Admin Staff, especially Priti and Sheetu. They have not only helped in fast resolution of all admin related matters but also been my friends. Their office feels less like an Admin office and more like a place to catch a break from all the work and meet some ever smiling and helpful people.

Lastly, a special mention of Starbucks, Nehru place, a place that felt like home away from home when IIIT didn't!!

Addendum:

Here I am doing the final edits on the draft just days before the D-day. Now, I am very far from the place and people that made this happen, I have moved for job to another city. But one thing that didn't change is the way those who made this thesis possible, made me possible, stood by me, helped me, cared for me and feel happy for me that I have reached this landmark. And also what didn't change is the way I feel for them, the way I owe a lot of me to them, Thank you ☺

Abstract

The enormous growth in online availability of information content has made Recommender Systems (RS) an integral part of most online portals and e-commerce sites. Most websites and service portals, be it movie rental services, online shopping or travel package providers, offer some form of recommendations to users. These recommendations provide the users more clarity, that too expeditiously and accurately in limiting (shortlisting) the items/information they need to search through, thereby improving the customer's experience. The direct link between customer's satisfaction and revenue of e-commerce sites induce widespread interest of both, academia and industry, in the design of efficient recommender systems.

The current de-facto approach for RS design is Collaborative Filtering (CF). CF techniques use the ratings provided by users, to a subset of the items in the repository, to make future recommendations. However, the rating information is hard to acquire; often a user has rated less than 5% of the items. Thus, the biggest challenge in recommender system design is to infer users' preference from this extremely limited predilection information. The lack of adequate (explicit) preference information has motivated several works to augment the rating data with auxiliary information such as user's demographics, trust networks, and item tags. Further, the scale of the problem, i.e. the amount of the data to be processed (selecting few items out of hundreds and thousands of items for an equally large number of users) adds another dimension to the concerns surrounding the design of a good RS. There have been several developments in the field of RS design over the past decades. However, the difficulty in achieving the desired accuracy and effectiveness in recommendations leaves considerable scope for improvement.

In this work, we model effective recommendation strategies, using optimization centric frameworks, by exploiting reliable and readily available information, to address several pertinent issues concerning RS design. Our proposed recommendation strategies are built on the principals of latent factor models (LFM). LFM are constructed on the belief that a user's choice for an item is governed by a handful of factors – the latent factors. For

example, in the case of movies, these factors may be genre, director, language while for hotels it can be price and location.

Our first contribution targets improvement in prediction accuracy as well the speed of processing by suggesting modifications to the standard LFM frameworks. We develop a more intuitive model, supported by effective algorithm design, which better captures the underlying structure of the rating database while ensuring a reduction in run time compared to standard CF techniques. In the next step, we build upon these proposed frameworks to address the problem of lack of collaborative data, especially for cold start (new) users and items, by making use of readily available user and item metadata - item category and user demographics. Our suggested frameworks make use of available metadata to add additional constraints in the standard models; thereby presenting a comprehensive strategy to improve prediction accuracy in both warm (existing users/items for which rating data is available) and cold start scenario.

Although, high recommendation accuracy is the hallmark of a good RS, over-emphasis on accuracy compromises on variety and leads to monotony. Our next set of models aims to address this concern and promote diversity and novelty in recommendations. Most existing works, targeting diversity, build ad-hoc exploratory models relying heavily on heuristic formulations. In the proposed work, we modify the latent factor model to formulate a joint optimization strategy to establish accuracy-diversity balance; our models yield superior results than existing works.

The last contribution of this work is to explore the use of another representation learning tool for collaborative filtering – Autoencoder (AE). Conventional AE based designs, use only the rating information; lack of adequate data hampers the performance of these structures, thus, they do not perform as well as conventional LFM based designs. In this work, we propose a modification of the standard autoencoder – the Supervised Autoencoder – which can jointly accommodate information from multiple sources resulting in better performance than existing architectures.

List of Figures

List of Tables

List of Abbreviations

AD	Aggregate Diversity
ADMM	Alternating Direction Method of Multipliers
AE	Autoencoder
APG	Accelerated Proximal Gradient (Algorithm)
BCD-NMF	Block Co-ordinate Descent for Non-negative Matrix Factorization
BCS	Blind Compressive Sensing
BCS_AD	Blind Compressive Sensing Framework balancing Accuracy and Diversity
BCS_M_Item	Blind Compressive Sensing Framework incorporating Item Metadata
BCS_M_User	Blind Compressive Sensing Framework incorporating User Metadata
BCS_Neigh	Blind Compressive Sensing Framework with Neighborhood Information
CB	Content based methods
CF	Collaborative Filtering
eNet_BCS	Elastic Net Regularized Blind Compressive Sensing
FISM	Factored Item Similarity Model
FPC	Fixed point Continuation (Technique)
GR_M	Non-Negative Matrix Factorization with Graph Regularization
HC	Hierarchal Clustering
IA	Item (Rating) Average
ID	Individual Diversity
KNN_M	Nearest Neighbor Model using Metadata
LC_BCS	Label Consistent Blind Compressive Sensing Framework
LC_MC	Label Consistent Matrix Completion Framework
LFM	Latent Factor Models
MAE	Mean Absolute Error
MA_C	Matrix Approximation with Clustering

List of Abbreviations (contd.)

MC_AD	Matrix Completion Model balancing Accuracy and Diversity
MC_Neigh	Matrix Completion Framework with Neighborhood Information
MC_SB	Matrix Completion using Split Bregman (Technique)
MC_User	Matrix Completion Framework incorporating User Metadata
MF	Matrix Factorization
MM	Majorization-Minimization (Technique)
PMF	Probabilistic Matrix Factorization
QoP	Quality of Prediction
RBM	Restricted Boltzman Machine
RMSE	Root Mean Square Error
RPRV	Reverse Predicted Rating Value
RS	Recommender System
SGD	Stochastic Gradient Descent (Technique)
SSNMF	Semi-Supervised Non-negative Matrix Factorization
SupervisedAE	Supervised Autoencoder

Table of Contents

Chapter 1
INTRODUCTION

Netflix boasts of a catalog of more than 6000 movies and TV shows in the US alone[1]

Amazon has in excess of 350 million products on sale worldwide[2]

Flipkart offers over 80 million products to its customers[3]

The rapid expansion of the internet and the associated growth in online retail and e-commerce has brought varied resources – media, travel plans, conference papers, consumer products and more – right to our homes. However, this ease of access comes laden with the penalty of information overload.

In the words of eminent economist, Herbert A. Simon "…*information consumes the attention of its recipients. Hence a wealth of information creates a poverty of attention.*" These words aptly capture the demerits of information overload from both the user's as well as the system's perspective – first, the user has to undertake the daunting task of surfing through a large product repository which severely hampers his/her experience and second, e-commerce portals lose revenue owing to lack of customer engagement.

The necessity to address these issues has provided both industry and academia the impetus to develop effective Recommender Systems (RSs) [1, 2, 3]. Today RSs are the workhorse behind all Business-to-Client (B2C) e-commerce portals like Netflix, Amazon, Flipkart, and iTunes. The prime objective of a RS is to assist users in item selection, by suggesting a handful of products from the repository, such that the suggestions are aligned with the users' preference. To ascertain a user's preference pattern, RS uses either explicit (actively provided by the user in the past) or implicit information (garnered by observing the user's interaction with the online portal). Explicit

1. http://360pi.com/press_release/many-products-amazon-actually-carry-categories/
2. http://time.com/4272360/the-number-of-movies-on-netflix-is-dropping-fast/
3. http://www.flipkart.com/about-us?otracker=hp_footer_navlinks

information is often in the form of ratings (say on a scale of 1-10) given by users to items purchased by them in the past; for example, ratings given to items on Amazon or hotels on booking.com. Ratings can also be in the binary form; for example, like/dislike option on YouTube. On the other hand, implicit information can be construed by observing a user's online behavior such as his/her browsing history or buying pattern; for example, the number of views of a video on YouTube or the number of clicks on an advertisement. Implicit information is ambiguous and thus, not as reliable as the explicit ratings. However, explicit information is more difficult to acquire as it mandates a user's active participation and is thus, extremely limited. To overcome the challenge of data scarcity (limited availability), several RS design schemes use information from secondary sources such as content/tag description of items or social network profile, friend circle and group membership of users, in addition to explicit rating data. However, the availability of such information to online portals is restricted. For example, for a large number of small or niche product/service providers, restricted access to additional information is a major bottleneck in the design of effective recommender systems. Further, use of secondary information such as a user's social network profile is afflicted with privacy concerns arising out of disbursement and subsequent use of a user's personal/private information.

In this work, we model effective recommendation strategies, using optimization centric frameworks, by exploiting reliable and readily available information. Use of explicit (rating) information, instead of implicitly gathered data, improves the robustness and reliability of our designs. Also, our proposed frameworks use only limited and readily available add-on information (to augment the rating data) which make our designs suitable for use across a large number of e-commerce platforms (including those with limited access to multiple data sources). Despite a large number of works on the design of effective RSs, there is considerable scope for improvement and this work is an attempt towards reducing the gap between the desired and the current state of the art.

1.1 Overview of Recommender Systems

Interest in RS design started in 1990's when the initial works targeted towards improving a user's experience were proposed [4, 5]. Over the past decade, RSs have become an

indispensable component of online platforms, and are widely employed in areas ranging from movie/music recommendations [6] to travel package suggestions [7] and dating advice [8] to conference recommendations [9].

1.1.1 Desired Characteristics of a Recommender System

Recommender systems are software tools used to generate relevant purchase suggestions for the customers. Although the idea of relevance and effectiveness of suggestions is subjective and specific to the domain under consideration and the actual requirements of the business model, we briefly discuss some of the general properties of a good RS.

1. **Quality of Prediction**:

   One of the key requirements of a good recommender system is to generate meaningful recommendations so that a customer's interest in the portal is maintained. This need translates into generating recommendations that are aligned with a user's preference. Deciphering a user's preference from the extremely limited (available) information is a major challenge for RS design. However, the accuracy of recommendation from a user's perspective is not the sole measure of the effectiveness of a RS. A recommender system's capability to enhance the visibility of items, especially niche products is also advantageous from both the customer as well as retailer's perspective.

2. **Speed of Computation**:

   Most of the big online retailers like Amazon have more than a million registered customers and an equally large number of items. Thus, an effective recommender system should be equipped to handle this information overload and generate relevant suggestions in reasonable time. Further, continuously new ratings, users, and items are added to the portal, and a slow recommendation strategy will hinder frequent updates to accommodate the same. Thus, the speed of processing is a major criterion for any RS design algorithm.

3. **Applicability of Design**:

   The design of recommender systems that effectively use available information entails considerable effort. A generic recommendation framework which can work across domains and with limited information can enjoy wider applicability; this is an added benefit over target (portal) centric designs.

1.1.2 Design Methodologies for Recommender Systems

As stated before, RS essentially provides a means to infer a user's preference and make useful and personalized suggestions by exploiting available sources of information. The techniques for recommender systems design - the manner in which the available information is exploited to make future recommendations - can be broadly classified into three categories [2]:

1. Content-based Methods
2. Collaborative Filtering Techniques
3. Hybrid Recommender Systems

Content-based (CB) methods [10, 11] attempt to find and suggest those items (from the repository) to a user which are highly similar to the ones liked by them in the past. The similarity is inferred on the basis of attributes or descriptors associated with each item. For example, in the case of movies, genre, director and star cast can be the relevant factors; similarly for research papers, authors, the area of research or publication venue can be the defining characteristics. A major limitation of CB approaches is that they require explicit profiling of all items in the repository, which might not always be possible; missing information becomes a major limitation. Also, the basis for similarity computation is limited by the domain knowledge and also to the characteristics that can be explicitly listed such as genre or director of a movie rather than more abstract (higher order) factors or interactions amongst the listed attributes.

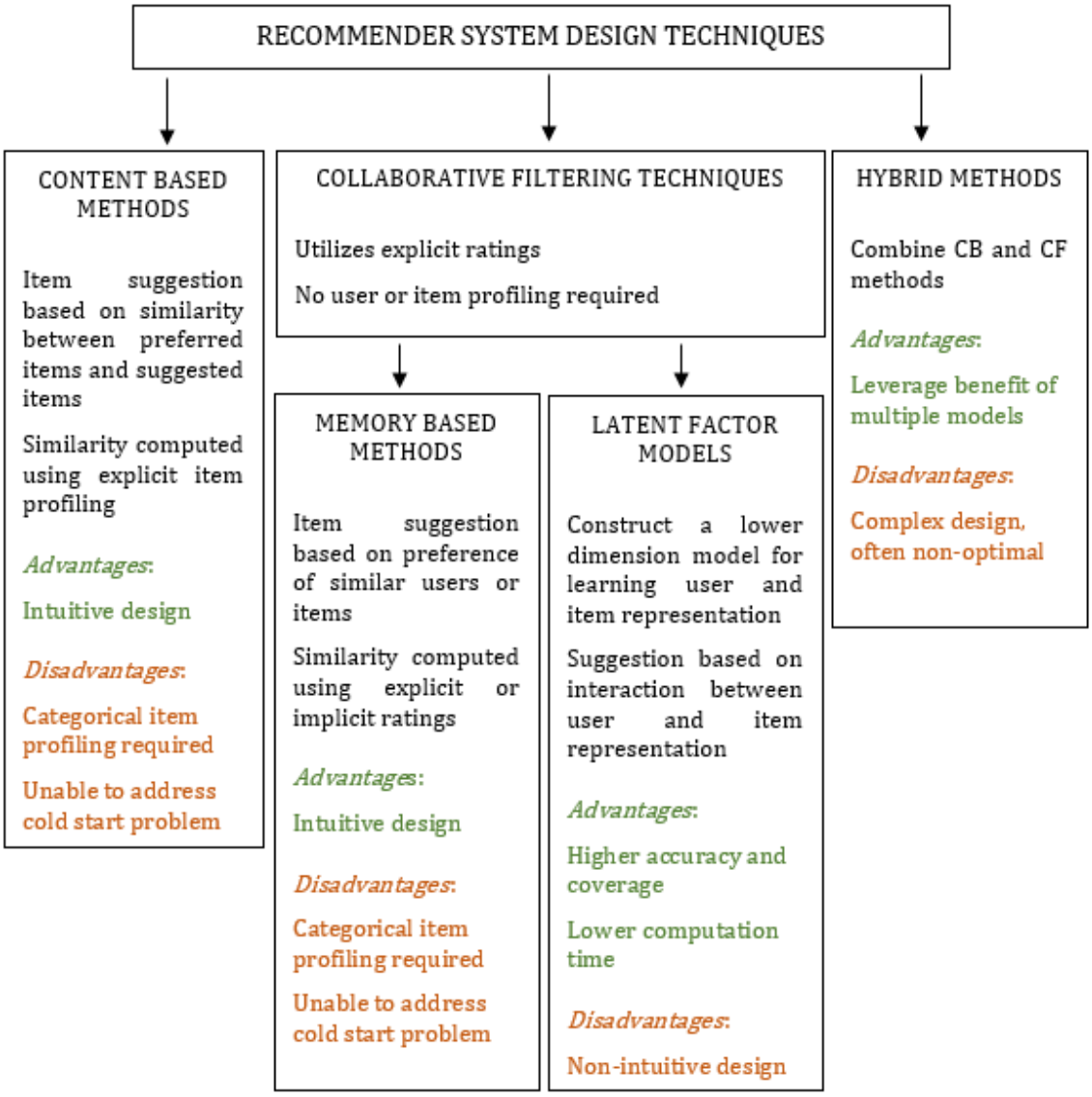


Figure 1.1 Classification of Recommender System Design Techniques

The limitations of the content-based models are alleviated to a large extent by Collaborative Filtering (CF) techniques [12, 13]. Unlike CB methods, CF techniques rely on explicit rating information provided by the users in the past, to ascertain their preference; thus, eliminating the need to acquire and store content descriptors for each item. CF algorithms can be further classified as memory based and latent factor model based formulations. Memory based methods [14, 15] are primarily heuristic formulations that work on the entire available rating (raw) dataset to define a user's neighborhood

(similar users) and use the ratings provided by these neighbors (similar users) for future predictions. On the contrary, latent factor model based strategies [16] build a lower dimension model from the existing rating database and use it for making recommendations. CF techniques, especially the latent factor model based frameworks are the current de-facto approach for the design of recommender systems. The recommendation frameworks proposed in this work also belong to the class of CF techniques and thus, the same are discussed in further detail in the next section.

In the recent past, hybrid RS have also been proposed that use a combination of CB and CF methods to leverage the advantages of both and build improved frameworks [17, 18]. A summary of the various techniques for RS design is given in figure 1.1.

1.2 Collaborative Filtering Techniques

Collaborative filtering techniques [12] use the known interactions between users and items to predict relevant interactions/preferences for the future. The interaction between users and items is captured via the rating matrix (having users as rows and items as columns) – ratings being either explicit or implicit. However, the difficulty in acquiring the rating information makes the rating matrix highly sparse (very few available values), thereby posing a challenge for RS design.

The information captured in the rating matrix can be exploited via two strategies – memory based methods and the latent factor model based designs. We briefly discuss the two.

1.2.1 Memory based Methods

Memory or neighborhood based methods [15] are heuristic design strategies that process the entire rating matrix to generate relevant suggestions for the concerned (active) user. These methods scan the rating matrix to compute similarities between users (user-user approach) or items (item-item approach), based on the available rating data, and thus, define a neighborhood (of similar entities) for the target user or item. Similarity

computation is done using measures such as Pearson correlation coefficient and cosine similarity.

In a user-centric design [19], to predict the rating of the target user on a yet unrated item, a neighborhood of target/active user is searched for, and the estimated rating is computed as a linear weighted combination of ratings given by its neighbors (similar users). Similarly, in an item-item approach [14] neighborhood (similar items) of the target item is defined, and a weighted average of the ratings given by target user to neighboring items is used for rating prediction. The user and the item based recommendation systems can also be combined [20].

Memory (neighborhood) based models are intuitive designs and thus, easy to interpret. However, data scarcity adversely affects the (local) neighborhood approach which leads to poor coverage, i.e. rating prediction is not possible for all items in the dataset. Also, the need to scan the entire (huge) rating matrix (more than several thousand users and items) for the task of rating prediction makes memory based models computationally slower.

1.2.2 Latent Factor Model based Methods

Unfortunately, neighborhood strategies do not always yield the best of results; Latent factor models [16, 21], though lacking the ease of interpretability, can perform better than their neighborhood based counterpart [2].

Model based techniques use the available rating information to construct a reduced dimensionality model which is used for further processing and rating prediction. Latent factor models (LFMs) are based on the assumption that a user's choice for an item is governed by a handful of factors – the latent factors. For example, in the case of books they can capture features like the genre, author, and language while for tourism destinations they encapsulate defining characteristic like category (beach, mountain and so on) or budgetary constraints. The LFM based techniques attempt to represent both users and items by vectors capturing their association/affinity with these latent factors and the rating given by a user to an item is the quantification of the interaction between the two vectors.

The advantage of LFMs over content-based methods is that the former class of methods does not require explicit profiling of items/users. Instead, LFMs attempt to capture an abstraction of the deciding features (latent factors) by exploiting the known interaction (ratings) between users and items. Also, working with the lower dimension model improves the processing speed compared to memory based methods. Further, LFMs are shown to outperform the latter in terms of both Quality of Prediction (QoP) as well as coverage attained [2].

In this thesis, we propose latent factor model based frameworks for design of effective recommender systems, owing to the advantages offered by the same over other modeling strategies.

1.3 Research Contributions

The main aim of this thesis is to propose collaborative filtering strategies for RS design, that address several pertinent issues concerning the recommendation process, using reliable and widely available information. We cast the recommender system design problem in an optimization framework which ensures mathematically/theoretically sound solution.

The research contributions of this thesis are summarized below.

1. **Design of improved frameworks and algorithms for latent factor model based collaborative filtering techniques.**

   The primary focus of collaborative filtering techniques is to improve the accuracy of prediction, i.e. generate recommendations highly aligned with a user's past preference. The paucity of reliable (explicit rating) data and the scale of the problem (hundreds of thousands of items and equally large number of registered users) make the design of accurate and fast RSs a challenge. In this work, we propose modifications to conventional latent factor framework and develop more intuitive models to improve the accuracy of prediction. We also design effective algorithms using sophisticated optimization techniques to achieve lower run time for our proposed models compared to standard collaborative filtering techniques.

2. **Design of supervised frameworks for latent factor model**

As stated before, rating information is highly limited. Thus, any additional source of information that can assist in deciphering users' preference pattern is a desirable addition. In this work, we propose RS design schemes that are adept at effectively utilizing readily available secondary information – item categories and user demographics – along with the rating data. Item categorization and user's demographic profile (such as age, location, gender) are widely available on almost all online portals, and thus, our proposed models have wide applicability. We utilize user and item metadata to impose additional constraints (such as similarity in preference pattern for items belonging to same/similar categories) which restrict the search space and thus yield improved solution. Such metadata is also available for cold start users (those with no rating data associated with them) and can be exploited to compensate for lack of rating data. In this work, we propose frameworks that can address the problem of rating prediction for both warm start (users with existing collaborative information) as well as cold start users. Our models are shown to outperform existing works in both cold and warm start scenarios.

3. **Design of comprehensive recommender system balancing accuracy with diversity**

The primary task of a recommender system is to improve users' experience by recommending relevant and interesting items to them. The conventional approach of using the accuracy of prediction as the sole evaluation criteria to judge the effectiveness of a RS is highly restrictive. Several works have suggested a need for a more comprehensive view of recommendations, which along with focussing on due coherence with a user's (past) choice, also places emphasis on providing variety and novelty in suggestions. Most existing works for achieving accuracy-diversity balance are ad-hoc exploratory models which entail use of certain heuristic measures. In this work, we propose to modify the standard latent factor model to jointly optimize diversity and accuracy. Although, the two characteristics are contradictory to each other, a joint optimization strategy helps

achieve the desired balance while ensuring optimality of the solution. The experimental results validate our claim that our models are more proficient at establishing the required accuracy-diversity balance than existing designs.

4. **Design of recommender system from a machine learning perspective**

   In this work, we extend the use of optimization techniques to the domain of machine learning. We propose the design of a supervised autoencoder module which is used for rating prediction. Conventional autoencoders attempt to reconstruct the input (rating data in case of RS design) from the latent or hidden layer representation of the same. We modify the standard autoencoder framework to accommodate information from multiple sources (rating data as well as user/item metadata). This modification assists in achieving satisfactory performance, despite the lack of adequate amount of (rating) data as required by conventional machine learning architectures.

1.4 Publications

The work done in this thesis has resulted in several publications. The following section lists the publications categorized by type of publication venue.

Journal Papers

1. Anupriya Gogna and Angshul Majumdar, "DiABlO: Optimization based design for improving diversity in recommender system," Information Sciences, Volume 378, Pages 59-74, 1 February 2017
2. Anupriya Gogna, and Angshul Majumdar, "A Comprehensive Recommender System Model: Improving Accuracy for both Warm and Cold Start Users," IEEE Access, vol.3, no., pp.2803-2813, 2015
3. Anupriya Gogna and Angshul Majumdar, "Blind Compressive Sensing Formulation Incorporating Metadata for Recommender System Design," APSIPA Transactions on Signal and Information Processing, vol. 4, e2, 2015

4. Anupriya Gogna and Angshul Majumdar, “Matrix Completion Incorporating Auxiliary Information for Recommender System Design,” Expert Systems with Applications, Volume 42, Issue 14, 15 August 2015, Pages 5789-5799
5. Anupriya Gogna and Angshul Majumdar, “Balancing Accuracy and Diversity in Recommendations using Matrix Completion Framework,” Knowledge based Systems, Volume 125, June 2017, Pages 83-95
6. Anupriya Gogna, Janki Mehta, Kavya Gupta and Angshul Majumdar, “Enhancing Latent Factor Model with Neighborhood Information for Recommender System Design,” IEEE Transaction on Knowledge and Data Engineering (Submitted)

International Conferences

7. Anupriya Gogna, and Angshul Majumdar, “Supervised Learning in Matrix Completion Framework for Recommender System Design,” IEEE International Conference on Management of Data, COMAD 2016, March 2016, Pune, India
8. Anupriya Gogna and Angshul Majumdar, “Distributed Elastic Net Regularized Blind Compressive Sensing for Recommender System Design,” 20th IEEE International Conference on Management of Data, COMAD 2014, pp. 29-37, December 2014, Hyderabad, India
9. Anupriya Gogna, Ankita Shukla and Angshul Majumdar, “Matrix recovery using Split Bregman,” 22nd International Conference on Pattern Recognition, ICPR 2014, pp. 1031-1036, August 2014, Stockholm, Sweden
10. Anupriya Gogna, and Angshul Majumdar, “Semi Supervised Autoencoder,” 23rd International Conference on Neural Information Processing, ICONIP 2016, pp. 82-89. Springer International Publishing, 2016

1.5 Outline of Thesis

The thesis is structured as follows

In Chapter 1, we introduce the problem of recommender system design and discuss various modeling techniques for the same. We present a brief review of collaborative filtering techniques, highlighting the requirements of a good recommendation algorithm followed by our contributions. The chapter also provides a list of the publications resulting out of the proposed works and the outline of the document.

In Chapter 2, the focus is on the design of latent factor model based collaborative filtering techniques for RS design. A discussion of the existing frameworks for LFM is followed by the description of our proposed models and algorithm design. The emphasis of our proposed work is on improving prediction accuracy while ensuring a reduction in the run-time (cost) of the recommendation strategies using modified latent factor model based designs and better algorithmic frameworks. A description of the benchmarks datasets and evaluation metrics for comparison of proposed approaches with existing frameworks is also given.

In Chapter 3, we concentrate on works that effectively exploit readily available secondary sources of information (user and item metadata) to augment the standard CF frameworks; latter using only the explicit rating information. The chapter presents a review of existing works using both ratings and metadata, followed by a description of our proposed models. We also discuss the cold start problem and showcase the ability of our models to generate relevant predictions in cold start scenario as well.

In Chapter 4, the emphasis of proposed work is on generating well-rounded and comprehensive recommendations which aim to balance accuracy of recommendation with diversity and novelty. The need for diversification, as argued by previous studies, is presented along with a survey of existing models for the same. The literature survey is followed by details of our proposed strategies for establishing accuracy-diversity balance; our designs are shown to outperform existing works.

In Chapter 5, we address the problem of rating prediction from a machine learning perspective. Conventional machine learning architectures require a large amount of data.

Lack of adequate, reliable information, as in the case of RS design, hampers the effectiveness of these architectures. In this chapter, we present our design for a supervised autoencoder module which can harness information from multiple sources to improve the accuracy of rating prediction compared to the standard autoencoder based designs.

In Chapter 6, we summarize the contributions of this thesis and highlight the future direction.

Chapter 2
LATENT FACTOR MODEL BASED COLLABORATIVE FILTERING TECHNIQUES

Modern day service/product oriented online business-to-client (B2C) portals rely significantly on recommender systems to offer an enhanced user experience; Collaborative Filtering (CF) techniques being the de facto approach for the same.

Standard collaborative filtering methods use the feedback provided by users, to a subset of products/services on offer, to make relevant predictions on the remaining; unlike content-based methods, that rely on domain knowledge and explicit characterization of users and items. The feedback is usually captured in the form of a rating matrix – a matrix of partially filled values wherein each available entry (rating value) is a measure of the affinity of a user (along rows) for an item (along columns). Feedback or the ratings can be of two types - the first one is implicit, i.e. the prediction needs to be made based on implicitly gathered data such as browsing history or buying pattern of users; the user does not explicitly specify if he/she likes the item or not. The second type of rating is explicit, for example, movie ratings on a 5-point scale, or the 'like' option in YouTube. Figure 2.1 gives examples of the explicit rating matrix. Implicit ratings are easier to obtain but are not very informative; especially for negative feedbacks. For example, if a person does not buy a product one does not know if he/she does not like it or does not know about it. On the other hand, explicit ratings are more dependable and thus, more widely used in recommender system (RS) design. However, as they involve the active participation of the user, they are hard to capture and thus, extremely limited. Despite the associated (data) scarcity, most recommender system design schemes use the explicit rating information to ensure more reliable predictions.

INTEGER RATINGS

	I1	I2	I3	I4
U1	5	-	4	-
U2	-	4	2	-
U3	1	-	-	-
U4	-	-	3	5

BINARY RATINGS

	I1	I2	I3	I4
U1	↑	-	↓	-
U2	-	↓	↑	-
U3	-	-	↑	↓
U4	↑	↓	-	-

Figure 2.1 Examples of (Explicit) Rating Matrix

CF models [12] are built on the assumption that if two users rate a few items similarly, they will most probably rate others also in similar fashion. This belief can be exploited in two ways – neighbourhood based approach and latent factor model based approach.

Memory based methods [15, 20] are simple heuristic strategies, which though more intuitive, perform poorer than their (latent factor) model based counterpart in terms of prediction accuracy and speed of computation [2]. Latent factor models (LFMs) perform better than memory based counterparts because they are based on theoretically sound frameworks and modeling strategies instead of ad-hoc heuristic rules employed by the latter. Thus, LFMs are the most widely adopted and researched technique for RS design.

There have been several works in the past decade which propose LFM based techniques to exploit the limited amount of rating information and address several pertinent concerns such as speed, scalability, and prediction quality. However, there is still considerable scope for improvement across varied evaluation criteria, dominantly owing to two facts – first, the scarcity of data places a practical limit on achievable accuracy and second, the accuracy of recommendation significantly impacts the sales profit of online portals. This can be gauged from the fact that in 2009, Netflix[4] awarded a grand prize of $1,000,000 for a 10% improvement in prediction accuracy. Thus, even a (relatively) small improvement in prediction accuracy is both significantly relevant as well as a major

4. http://www.netflixprize.com/

challenge. Motivated by the necessity and relevance of effective recommendation strategies, we propose latent factor model based formulations utilizing the (available) explicit rating information. The emphasis of our designs is on improving both the prediction accuracy as well as the computation cost (run time) via the design of novel and improved frameworks supported by an effective algorithmic implementation.

2.1 Review of Latent Factor Models for Collaborative Filtering

LFMs are parameterized modeling techniques that derive a lower dimension model from the rating data, and use it for subsequent processing. The model is built on the belief that a user's choice (rating) for an item is governed by only a handful of features – the latent factors. For example, in the case of movie recommendations, these can be genre, director, language, and alike. Although, the abstraction captured by the latent factors are domain specific, LFMs do not require an explicit knowledge of the area of recommendation or the associated users and items. LFMs attempt to recover the user's preference prototype and the items' characterization in terms of the latent factors from the given rating data itself.

LFM based formulations [21, 22] models/represents both users and items as vectors characterizing their association with each of the latent factors. Consider the example of movie recommendations; a movie can be described in terms of features such as genre (like thriller/comedy/drama), cast and language and a user's liking for the movie is a function of his/her affinities to these features. Thus, an item can be profiled as a vector enumerating the extent of possession of various features, and a user can be modeled as a vector describing his/her affinity towards the corresponding features. In such a scenario, a user's rating for an item, considered as an indicator of the degree of alignment of their respective latent factor representations, is modeled as an inner product between their corresponding latent factor vectors as shown in (2.1).

$$r_{c,i} = \langle u_c, v_i \rangle = u_c^T v_i \qquad \text{…..……} \ (2.1)$$

where, $r_{c,i} \in \mathbb{R}$ is the rating given by the user c to item i ; $u_c \in \mathbb{R}^{F\times 1}$ and $v_i \in \mathbb{R}^{F\times 1}$ are the latent factor vectors for user c and item i respectively and F is the number of latent factors.

Ideally, the ratings are expected to be purely a function of the interaction between users' and items' latent factor vectors; however, in practice, they are a noisy version of the same [22]. In addition to the pure interaction component, the actual ratings are also inflicted by certain biases which constitute the baseline component as in (2.2)

$$r_{c,i} = interaction(c,i) + baseline(c,i) = \langle u_c, v_i \rangle + baseline \quad \text{..... (2.2)}$$

2.1.1 Baseline Estimation

Baseline component captures the inherent biases associated with both users and items which contribute to the subjective evaluation (rating) of an item by a user.

The baseline component (2.3), consist of user bias, item bias and the global mean.

$$baseline(c,i) = b_{user_c} + b_{item_i} + mean_g \quad \text{..... (2.3)}$$

In (2.3), $b_{user_c} \in \mathbb{R}$ denotes the bias of user c; $b_{item_i} \in \mathbb{R}$ is the bias of item i; $mean_g = \frac{1}{|R|}\sum_{m,n\in\Omega} r_{m,n}$ is the global mean or mean of the available rating data. These biases capture the general rating pattern of the concerned user or item and are independent of their interaction or affinity for each other. For example, a liberal user will rate almost everything above average while a critical user will give (in general) lower ratings to items. Thus, for a critical customer, user bias is a negative quantity whereas it has a positive value for a liberal one. Likewise, items may also be (in general) rated high or low across users. For example, a movie like Titanic or Lord of the Rings III will always be rated high owing to the 10+ academy awards they won.

Thus, considering both the baseline and the interaction component, the rating by user c on item i can be modeled as follows

$$r_{c,i} = u_c^T v_i + b_{user_c} + b_{item_i} + mean_g \quad \text{..... (2.4)}$$

Baseline estimation (estimating user and item biases) is often done offline; this reduces the cost of computation during online processing, while not compromising on the prediction accuracy. User and item biases are computed by solving the following optimization problem (2.5) using stochastic gradient descent (SGD) technique [22].

$$\min_{b_{user^*}, b_{item^*}} \sum_{m,n \in \Omega} \left(r_{m,n} - b_{user_m} - b_{item_n} - mean_g \right)^2 + \delta \left(b_{user_m}^2 + b_{item_n}^2 \right) \quad \text{…..……} \ (2.5)$$

In (2.5), Ω denotes the set of user and item index pairs (m,n) for which the rating information is available and δ is the regularization parameter.

Once baseline estimation is done, the bias component is subtracted from the available rating values to recover the actual (clean) interaction component as in (2.6), where $y_{c,i}$ is the interaction part of the ratings capturing the affinity of user c for item i.

$$y_{c,i} = r_{c,i} - b_{user_c} - b_{item_i} - mean_g \quad \text{…..……} \ (2.6)$$

Rating prediction primarily involves the recovery of the interaction component of the (missing) ratings via LFM based formulations. Once the interaction component is recovered, the baseline estimates are added back to compute the net rating value. As all the formulations proposed in this work are with baseline correction, the term interaction component and rating are used interchangeably henceforth, unless specified.

Baseline estimation is relatively simple, the challenge in RS design is to use the interaction component of the available ratings to characterize the preference pattern of users. The most commonly used formulation to estimate the same is matrix factorization, which is discussed below.

2.1.2 Matrix Factorization Framework for Latent Factor Model

Matrix Factorization (MF) [22], is one of the most widely adopted means of recovering the latent factor representation of users and items owing to its performance and scalability. MF considers the task of estimation of the latent factor vectors of both users and items as a regularized optimization problem (2.7).

$$\min_{U,V} \left\| Y - A(UV) \right\|_F^2 + \lambda_u \left\| U \right\|_F^2 + \lambda_v \left\| V \right\|_F^2 \quad \text{…..…… (2.7)}$$

where, $Y \in \mathbb{R}^{M \times N}$ is the matrix consisting of the interaction component of ratings wherever available, rest being zero, M being the total number of users and N the total number of items; $U \in \mathbb{R}^{M \times F}$ contains the latent factor vectors of users stacked as rows of the matrix; $V \in \mathbb{R}^{F \times N}$ is the matrix of items' latent factor vectors; A is a binary mask having 1's in place of available ratings and 0's elsewhere; λ_u, λ_v are the regularization parameters.

In (2.7), the data consistency term (first term) ensures that the latent factor vectors are derived such that their interaction (inner product) is consistent with the available rating values while the Frobenius norm regularization, on the derived latent factor vectors, prevents over-fitting to the available dataset.

Once, the latent factor vectors for all users and items are recovered, the filled matrix of interaction values can be computed as $Z = U \times V$. Although (2.7) is bilinear, the two variables (U,V) are separable and hence, each of them can be efficiently solved for. Commonly used approaches for solving the MF formulation are Stochastic Gradient Descent (SGD) and Alternating Least Squares (ALS) [22].

SGD algorithm, loops through all the available ratings in the dataset and updates the latent factor vectors as in (2.8), where, γ is the step size for the descent algorithm.

$$\begin{aligned} &\text{for } m,n \in \Omega \\ &\quad e_{m,n} = r_{m,n} - u_m^T v_n \\ &\quad u_m \leftarrow u_m + \gamma \left(e_{m,n} v_n - \lambda_u u_m \right) \\ &\quad v_n \leftarrow v_n + \gamma \left(e_{m,n} u_m - \lambda_v v_n \right) \end{aligned} \quad \text{…..…… (2.8)}$$

In the case of ALS algorithm, one variable is held constant, while the other is updated using a simple least square solver as in (2.9) where the superscript k denotes the iteration number.

$$
\begin{aligned}
&//\ \textit{Solve for } U \\
&U_{k+1} \leftarrow \min_{U} \left\| Y - A\left(UV_k\right) \right\|_F^2 + \lambda_u \left\| U \right\|_F^2 \\
&//\ \textit{Solve for } V \\
&V_{k+1} \leftarrow \min_{V} \left\| Y - A\left(U_{k+1}V\right) \right\|_F^2 + \lambda_v \left\| V \right\|_F^2
\end{aligned}
\qquad \text{…..…… (2.9)}
$$

In several works, where bias correction is not considered, the net rating matrix has only non-negative values. For such frameworks, weighted non-negative matrix factorization (WNMF) algorithms [23] have been proposed which use ALS kind of a scheme but with an additional projection step as in (2.10), to ensure that the latent factor representation also has non-negative entries.

$$
\begin{aligned}
&//\ \textit{Solve for } U \\
&U_{k+1} \leftarrow \min_{U} \left\| Y - A\left(UV_k\right) \right\|_F^2 + \lambda_u \left\| U \right\|_F^2 \\
&U_{k+1} \leftarrow \left(U_{k+1}\right)_+ \\
&//\ \textit{Solve for } V \\
&V_{k+1} \leftarrow \min_{V} \left\| Y - A\left(U_{k+1}V\right) \right\|_F^2 + \lambda_v \left\| V \right\|_F^2 \\
&V_{k+1} \leftarrow \left(V_{k+1}\right)_+
\end{aligned}
\qquad \text{…..…… (2.10)}
$$

The scalability and efficiency offered by matrix factorization has made it an active area of research. Some of them are mentioned here. For WNMF, several algorithms have been proposed that use multiplicative update rules, rather than additive updates, to yield non-negative factors [24, 25]. Authors in [26] have modified the SGD algorithm to use it efficiently (for WNMF) for cases with very small percentage of available observations. In [27], a distributed stochastic gradient algorithm has been proposed to perform MF efficiently over large datasets. Conventional SGD proves to be impractical for huge datasets, due to a large number of iterations required; a distributed model enables parallelization for efficient application of SGD. In [28, 29], probabilistic modeling has been used to solve for user and item latent factor matrices. [28] modeled the latent factor vectors of items and users as Gaussian priors, with diagonal covariance matrices, implying independence of latent factors. They used logarithmic posterior probability distribution (MAP estimate) as the optimizing function; however, MAP estimate is prone to over-fitting. [29] also followed Gaussian distribution priors but uses a full covariance

matrix, doing away with "independence" assumption, and also learns the distribution parameters in the process of finding the factors.

Although MF is very popular, as it is a fast and scalable model, it suffers from the disadvantage that the solution does not carry the same theoretical guarantees (convergence to global minima) as offered by convex formulations owing to its bi-linearity and thus, non-convexity. To address this issue, researchers have recently proposed the use of matrix completion framework [30, 31] as a convex counterpart of the MF formulation for LFMs. We discuss the same in the next section.

2.1.3 Matrix Completion Framework for Latent Factor Model

According to the postulates of LFM, users' affinity for items, in effect the entries of the rating matrix, are a function of a small number of parameters or the latent factors. These latent factors are the variables that govern the underlying structure of the data captured in the rating matrix. As the number of latent factors is quite small (approximately ~50-100) in comparison to the dimension of the rating matrix (running into hundreds of thousands), rating matrix is expected to possess a highly low-rank structure.

Hence, in recent works researchers have argued the use of Low-Rank Matrix Completion/recovery (LRMC) frameworks in the domain of collaborative filtering [30, 31, 32]. According to LRMC framework, missing values in a matrix can be satisfactorily recovered, by looking for the minimum rank solution, provided the matrix be (adequately) low rank. Thus, LRMC finds direct application in the domain of RS design.

Low-rank matrix completion models aim to recover the missing values in the (low-rank) matrix; it translates into directly predicting the interaction component of the missing ratings in the case of collaborative filtering problem. This is in contrast to the MF model which recovers the rating matrix as a product of two matrices (user and item latent factor matrices). LRMC recovers the missing values (ratings) in the matrix by searching for the lowest rank solution consistent with the given observations as in (2.11).

$$\min_{Z} \left\| Y - A(Z) \right\|_F^2 + \lambda Rank(Z) \qquad \text{…..……} \ (2.11)$$

where, $Z \in \mathbb{R}^{M \times N}$ is the full rating (interaction) matrix, A is a binary mask (1's in place of available values and 0's elsewhere); Y is the matrix of available ratings; λ is the regularization parameter. However, rank minimization is an NP-hard problem [33, 34], with doubly exponential complexity. Therefore, instead of minimizing the rank as in (2.11), its convex surrogate the nuclear norm (sum of singular values) is minimized as follows

$$\min_{Z} \left\| Y - A(Z) \right\|_F^2 + \lambda \left\| Z \right\|_* \quad \text{…..……} \quad (2.12)$$

It has been shown that if conditions outlined below are met, the solution to (2.12) is similar to that of (2.11) [35].

1. Left and right singular vectors of the low-rank matrix are uncorrelated with the canonical basis.
2. The matrix must be uniformly sampled, with an entry available from each row and column.
3. The number of measurements (m_s) for a matrix of size $S \times S$ should satisfy $m_s \geq C \times rank_{\max} \times S^{1.2} \log(S)$, where $rank_{\max}$ is the maximum possible rank of the matrix.

Broadly speaking there are two existing approaches to address the problem of LRMC. The first approach is based on thresholding of the singular values [36] while the second approach is the iterative reweighted least squares technique [37]. Although LRMC provides a convex framework for LFM and comes with associated guarantees on convergence to global minima, it is not widely adopted in RS literature. This is because most LRMC algorithms use singular value decomposition (SVD) which has a computational complexity of $O(n^3)$, raising scalability issues. However, for relatively smaller problem sizes, it is a superior alternative to MF model especially from the perspective of prediction accuracy.

2.2 Research Contributions

In this chapter, we propose frameworks and associated algorithms to exploit the explicit rating information such that our models outperform existing LFM based formulations both in terms of prediction accuracy as well as run time. Our main contributions in the area are summarized below.

1. We propose a Blind Compressive Sensing (BCS) based formulation in the realm of matrix factorization framework for latent factor models. Conventional MF models promote recovery of a dense latent factor representation for both users and items via the use of the Frobenius norm regularization as in (2.7). Our modified framework is based on the argument that although users might show some degree of affinity towards all latent factors, an item being an inanimate and inherently categorized object will never simultaneously show alignment to all the latent factors. Thus, an item's latent factor vector should have a sparse structure. Further, we also postulate that not all latent factors are necessarily independent and thus, we attempt to include this intrinsic dependency in our proposed framework. We model our belief as a formulation akin to the BCS framework augmented by an elastic net regularizer. We also design an algorithm for our proposed formulation which helps our model outperform existing MF models in terms of run time cost.
2. Our next work focuses on the LRMC framework wherein we propose an algorithm for matrix completion formulation for latent factor models using the split Bregman technique. The use of split Bregman technique helps improve the convergence speed as well as prediction accuracy of the matrix completion model compared to existing LRMC algorithms.

2.3 Elastic Net Regularized Blind Compressive Sensing Framework for Latent Factor Model

Most of the recent works, built on the principles of latent factor models, aim to improve the prediction accuracy by incorporating additional constraints or information into the standard framework. However, the basic model remains the same; involving factorization of the rating matrix into two matrices - one representing users and other items as in (2.7), repeated below for convenience.

$$\min_{U,V} \left\| Y - A(UV) \right\|_F^2 + \lambda_u \left\| U \right\|_F^2 + \lambda_v \left\| V \right\|_F^2$$

This work focuses on changing the basic structure of the latent factor model itself while retaining the primary concept that a limited number of (latent) factors affect the overall structure of the rating matrix.

Previous studies assume that both the factor matrices are dense (have non-zero values corresponding to all the latent factors), as promoted by the Frobenius norm (regularization) penalty on them. A dense structure is logical for the user latent factor matrix (U) but not for the item matrix (V). We argue the same using the example of a book recommendation problem. As a user, we have a certain level of curiosity towards all authors, genres, and publishers, i.e. even though someone may like Garica Marquez and Neruda more, he/she may not be averse to reading Pamuk or Naipaul. Similarly, even though one may like books on history, the user does not leave aside books on geography or travelogues. Thus, for users, it is natural to assume that he/she will have an affinity for all possible latent factors which translates into a dense user latent factor matrix i.e. non-zero values for all factors. However, the same argument does not hold for the item latent factor matrix. A book on geography will not have aspects of history in it; therefore, if the latent factor corresponding to geography is high in the book, the factor corresponding to history or drama will be zero. In effect, in most cases, the latent factor vectors corresponding to the items will be sparse. Therefore, in this work we propose to factor the user-item choice (rating) matrix into a dense user latent factor matrix and a sparse item latent factor matrix.

Our proposed formulation is new. We will show how this problem can be cast into the BCS [38] framework. However, instead of using the exact BCS formulation, we modify it by incorporating elastic-net regularization [39] to model the dependencies that might exist between various latent factors. To solve the resulting problem, we design an algorithm using the Majorization-Minimization strategy [40].

2.3.1 Proposed Formulation

In this section, we discuss our novel proposition for latent factor model based formulation for the design of an effectual recommender system.

Latent factor model forms the foundation of our approach i.e. akin to conventional MF model, we also express the rating matrix as a product of the users' and the items' latent factor matrices. However, our formulation stems from the multi-task regression framework. To begin with, we assume that the users' latent factor matrix is provided by an oracle, i.e. we know the affinity of all users towards the different latent factors. For a single item (i), the ratings by all users on the item under consideration are therefore modeled as in (2.13) where u_* denote the latent factor vector of users; v_i is the latent factor vector of the concerned item; $Y(1:M,i)$ denotes the M elements (M is the total number of users) of the i^{th} column of the interaction matrix Y.

$$Y(1:M,i) = \begin{pmatrix} u^T{}_1 \\ u^T{}_2 \\ \cdots \\ u^T{}_M \end{pmatrix} (v_i) \qquad \ldots\ldots\ldots (2.13)$$

Equation (2.13) is a single task regression model i.e. for a single item. We postulate that the latent factor vector v_i will be sparse. For example, consider the problem of movie recommendations. If we assume a simple model that a movie is only defined by its genre (this is not an overtly simplifying assumption and is true in most cases), then we will have a latent factor corresponding to each of the different genres. It is not possible for one movie to belong to ALL genres - it can be a romantic comedy, an action thriller, sci-fi, but it cannot be all at the same time. This means that the latent factor vector v_i will be

sparse. Based on this argument (2.13) turns out to be a sparse single task regression problem - we know the user's latent factors vector (u_*) and the task is to learn the sparse latent factor vector for the item i.

If we consider all the N items, we get the corresponding multi-task regression model as in (2.14)

$$Y = \begin{pmatrix} u_1 \\ u_2 \\ ... \\ u_M \end{pmatrix} \left(v_1 \; v_2 .....v_N\right) = U \times V \qquad \text{…..…… (2.14)}$$

where, U is a dense matrix (assumed to be known by an oracle) while V is a sparse matrix. As U reflects the user's affinities towards different factors; it is realistic to assume that most users have non-zero affinities towards all possible factors. However, an item cannot possess all the factors - hence V is sparse.

Since we assume that U is given, the objective is to recover V given the sampled entries in the rating matrix Y. This is a typical sparse multi-task regression problem [41, 42] which is solved as follows

$$\min_V \left\|Y - A(UV)\right\|_F^2 + \lambda_v \left\|vec(V)\right\|_1 \qquad \text{…..…… (2.15)}$$

Unfortunately, we never have the oracle latent factor matrix for the users'; it needs to be estimated from the data. Blind Compressed Sensing [38] facilitates us to achieve this goal. BCS framework enables us to jointly recover a sparsifying dictionary along with the sparse representation of the input/observation under the recovered dictionary. In our formulation (2.16), the matrix U is akin to the dictionary and matrix V is akin to the sparse representation of the observation Y. According to our assumption, the matrix U is dense. Therefore, a Frobenius norm penalty is appropriate while V has a sparse penalty. Further details of BCS framework are given in Appendix A (A.1).

$$\min_{U,V} \left\|Y - A(UV)\right\|_F^2 + \lambda_v \left\|vec(V)\right\|_1 + \lambda_u \left\|U\right\|_F^2 \qquad \text{…..…… (2.16)}$$

Next, let us take a closer look at the item latent factor matrix V. There are some factors which are linked to each other. For example, if Tarantino makes a film, it will be a thriller; if Vin Diesel is in a movie, it is likely to be an action/thriller. For factors that are linked to each other and are always 'happening together,' all of them should have non-zero values if even one is non-zero. The l_1-norm penalty does not guarantee that all the related factors (explanatory variables) will have non-zero values; this was pointed out in the paper on elastic net [39]. For more details on elastic net regularization, refer to Appendix A (A.2).

Latent factor model is an abstract model; we do not know which of the factors correspond to what in real life, i.e. we do not know if factor 1 corresponds to the director, if factors 2-4 correspond to the actors, or if factors 5-10 correspond to the genre. There is no way we can say which of these factors will always happen together. Thus, we cannot pose a strong group sparse structure on the item latent factor vectors. We can only encourage a grouping effect by imposing an additional l_2-norm penalty following the elastic net formulation [39]. Thus, we propose to reformulate our previous framework (2.16) as follows, where λ is the regularization parameter.

$$\min_{U,V} \left\| Y - A(UV) \right\|_F^2 + \lambda_v \left\| vec(V) \right\|_1 + \lambda_u \left\| U \right\|_F^2 + \lambda \left\| V \right\|_F^2 \qquad \text{…..……} \ (2.17)$$

Equation (2.17) gives our proposed LFM based design – Elastic Net Regularized Blind Compressive Sensing Framework (eNet_BCS). This is a slightly novel formulation, but it is easy to solve. In the next section, we derive an algorithm for our proposed formulation.

2.3.2 Algorithm Design

We make use of Majorization-Minimization (MM) technique [40] to derive an algorithm for our proposed formulation (2.17). The advantage of MM scheme is that it breaks a complex optimization problem into simpler steps which are easier to solve than the original problem.

We briefly review the MM approach before proceeding with the algorithm design. Consider an underdetermined linear system of equation, $y = Az$ where, $A \in \mathbb{R}^{p \times q}; p < q$.

Least square minimization (2.18) yields the solution (2.19) to the above linear system of equations.

$$\min_z f(z) = \min_z \|y - Az\|_2^2 \quad \text{…..…… (2.18)}$$

$$z = \left(A^T A\right)^{-1} A^T y \quad \text{…..…… (2.19)}$$

Equation (2.19) involves computing the inverse of $A^T A$. For cases with even moderately sized A, this is a computationally intensive task. However, it becomes almost impossible to efficiently solve (2.18) for very large sized data, involved in applications such as recommender systems. The need for such resource intensive computation is eliminated by the use of MM approach.

MM technique [40] involves replacing the minimization of a complex optimization problem by the minimization of its majorizer (which is simpler to solve). A function $g_k(z)$ is a majorizer of $f(z)$ *iff* $g_k(z) \geq f(z) \forall z$ and $g_k(z) = f(z)$ *at* $z = z_k$ where k denotes the current iteration number. Consider the optimization problem in (2.18), its majorizer at the current iteration k can be defined as

$$g_k(z) = \|y - Az\|_2^2 + (z - z_k)^T \left(\alpha I - A^T A\right)(z - z_k)$$
$$\textit{where}, \alpha \geq \max\left(\textit{eigen\_value}\left(A^T A\right)\right) \quad \text{…..…… (2.20)}$$

The constraint on α ensures that the second term in (2.20) is always positive and thus, (2.20) is a majorizer of $f(z)$ as defined in (2.18). At each iteration (k), instead of $f(z)$, $g_k(z)$ is minimized.

Minimization of $g_k(z)$ can be recast as the following minimization problem; for further details refer to Appendix A (A.3).

$$\min_z \|w - z\|_2^2 \text{ where, } w = z_k + \frac{1}{\alpha} A^T (y - Az_k) \quad \text{…..…… (2.21)}$$

As is evident, the solution to (2.21) does not involve computing the inverse of big matrices; replacing it by simple matrix-vector multiplication, thereby reducing

computation cost. MM algorithm (2.21) can be understood as a gradient descent algorithm on the least squares problem, using a constant step size of $1/\alpha$.

Returning to our proposed framework, the solution to (2.17) involves computing the inverse of the matrix $A^T A$ which given the large size of the rating matrix (dimension of A is same as that of Y) is computationally cumbersome. Thus, we employ MM technique, as discussed above, to decompose the minimization of (2.17) into simpler and easier to solve expression as in (2.22).

$$\min_{U,V} \left\|W-(UV)\right\|_F^2 + \lambda_v \left\|vec(V)\right\|_1 + \lambda_u \left\|U\right\|_F^2 + \lambda \left\|V\right\|_F^2$$

$$\text{where, } W = (U_k V_k) + \frac{1}{\alpha} A^T \left(Y - A(U_k V_k)\right) \quad \text{…..…… (2.22)}$$

$$\text{and, } \quad \alpha \geq \max\left(eigen\_value\left(A^T A\right)\right)$$

Further, (2.22) is a bi-linear expression involving minimization over two (optimization) variables. However, the variables are separable which allows us to split the objective function into simpler sub-problems, each minimizing over a single variable while holding the other constant, as follows

Sub-problem 1

$$\min_U \left\|W-(UV)\right\|_F^2 + \lambda_u \left\|U\right\|_F^2 \quad \text{…..…… (2.23)}$$

Sub-problem 2

$$\min_V \left\|W-(UV)\right\|_F^2 + \lambda_v \left\|vec(V)\right\|_1 + \lambda \left\|V\right\|_F^2 \quad \text{…..…… (2.24)}$$

The sub-problem in (2.23) can be framed as a simple least square expression (2.25) and solved using any gradient descent approach (like lsqr [43]).

$$\min_U \left\| \begin{pmatrix} W \\ \mathbf{0} \end{pmatrix} - U \begin{pmatrix} V \\ \sqrt{\lambda_u} I \end{pmatrix} \right\|_F^2 ; \mathbf{0} \text{ is a matrix of all zeros} \quad \text{…..…… (2.25)}$$

Similarly, sub-problem 2 can be recast as follows

$$\min_V \left\|P - QV\right\|_F^2 + \lambda_v \left\|vec(V)\right\|_1$$

$$\text{where, } P = \begin{pmatrix} W \\ \mathbf{0} \end{pmatrix} \text{ and } Q = \begin{pmatrix} U \\ \sqrt{\lambda} I \end{pmatrix} \qquad \text{…..…… (2.26)}$$

Equation (2.26) can be solved using iterative soft thresholding [44] as given in (2.27).

$$V \leftarrow soft\left(\Upsilon, \frac{\lambda_v}{2\beta}\right)$$

$$\text{where, } \Upsilon = V + \frac{1}{\beta} Q^T \left(P - QV\right), \qquad \text{…..…… (2.27)}$$

$$\beta \geq \max\left(eigen\_value\left(Q^T Q\right)\right)$$

$$\text{and} \quad soft\left(t, \tau\right) = sign\left(t\right)\max\left(0, \left|t\right| - \tau\right)$$

Input : U_0, V_0 are randomly initialized, maximum iterations $m\_iter, \lambda_u, \lambda_v, \lambda$

Output : U, V

while $k \leq m\_iter$ *or* $obj\_func(k) - obj\_func(k-1) \leq 1e-7$

$$W = \left(U_{k-1} V_{k-1}\right) + \frac{1}{\alpha} A^T \left(Y - A\left(U_{k-1} V_{k-1}\right)\right)$$

Solve for U

$$U_k \leftarrow \arg\min_U \left\| \begin{pmatrix} W \\ 0 \end{pmatrix} - U \begin{pmatrix} V_{k-1} \\ \sqrt{\lambda_u} I \end{pmatrix} \right\|_F^2$$

Solve for V

$$V_k \leftarrow soft\left(V_{k-1} + \frac{1}{\beta} Q^T \left(P - QV_{k-1}\right), \frac{\lambda_v}{2\beta}\right);$$

where, $\beta \geq \max\left(eigen\_value\left(Q^T Q\right)\right)$

and $\quad P = \begin{pmatrix} W \\ 0 \end{pmatrix}$ *and* $Q = \begin{pmatrix} U_k \\ \sqrt{\lambda} I \end{pmatrix}$

end while

Figure 2.2 Algorithm for Elastic Net Regularized Blind Compressive Sensing Framework (eNet_BCS)

Sub-problem 1 and 2 are alternately solved until convergence; convergence implying either the difference between objective function value in consecutive iterations falls below a threshold or a maximum number of iterations are reached. The complete algorithm is given in figure 2.2.

2.4 Matrix Completion using Split Bregman

Recently, low-rank matrix completion models [45] have been used for RS design because of their superior performance, compared to MF models, primarily in terms of prediction accuracy, owing to proven convergence guarantees. The principal focus of algorithms for recommender system design is to enable faster processing for efficient online implementation of the design formulation. However, the computational complexity of LRMC algorithms is high because of the need for singular value decomposition at every iteration.

In this work, we propose an algorithm for matrix completion (2.28), given the subsampled rating matrix Y, using split Bregman technique [46]

$$\min_{Z} \left\| Y - A(Z) \right\|_F^2 + \lambda_n \left\| Z \right\|_* \qquad \ldots\ldots\ldots (2.28)$$

where, $Z \in \mathbb{R}^{M \times N}$ is the (filled rating) matrix to be recovered, A is a linear projection (subsampling) operator, λ_n is the regularization parameter.

The use of split Bregman technique helps achieve improvement in both accuracy and convergence speed, thereby making our approach better suited to RS design than existing LRMC algorithms.

2.4.1 Algorithm Design

Before considering the LRMC framework, we briefly discuss the merits of split Bregman technique [46]. Split Bregman algorithm is an adaptation of Alternating Direction Method of Multipliers (ADMM) [47] particularly suited to solving l_1 minimization

problem. As nuclear norm minimization is an extension of sparse (vector) recovery, split Bregman technique can be well adapted to solving problems with low-rank constraints.

The idea behind split Bregman technique is to enable separation of multiple norm terms so that each can be efficiently (and separately) solved. Also, unlike conventional approaches (like iterative soft thresholding [44]) the regularization parameters need not be cooled and can be maintained at optimum value for better convergence and accuracy of recovery. Further details of split Bregman algorithm can be found in Appendix A (A.4).

Returning to the matrix completion model (2.28), we discuss the proposed algorithm design, using split Bregman technique.

Equation (2.28) imposes two constraints on the optimization variable - data consistency imposed by Frobenius norm term and the low-rank nature imposed by the nuclear norm term. To enable splitting of multiple norm terms, we introduce a proxy variable X ; X being a proxy of original variable Z .

$$\min_{Z} \left\| Y - A(Z) \right\|_F^2 + \lambda_n \left\| X \right\|_* \text{ such that } X = Z \qquad \ldots\ldots\ldots (2.29)$$

Usually, following a Lagrangian based approach, the constraint in (2.29) is incorporated into the optimization framework as in (2.30), where λ is the Lagrangian variable [110].

$$\min_{Z,X} \left\| Y - A(Z) \right\|_F^2 + \lambda_n \left\| X \right\|_* + \lambda \left\| X - Z \right\|_F^2 \qquad \ldots\ldots\ldots (2.30)$$

However, the exact Lagrangian would enforce equality between the proxy and the variable at every iteration. This is not required; for practical purposes, we only need the proxy and the variables to converge at the solution. Therefore, following the principles of Bregman iterations, we can relax the Lagrangian to the form given in (2.31)

$$\min_{Z,X} \left\| Y - A(Z) \right\|_F^2 + \lambda_n \left\| X \right\|_* + \mu \left\| X - Z - B \right\|_F^2 \qquad \ldots\ldots\ldots (2.31)$$

where, B is the Bregman relaxation variable; μ is the regularization parameter. The Bregman variable is updated via simple gradient descent in every iteration as in (2.32), to

ensure that at convergence the equality constraint between the original and proxy variable is satisfied.

$$B \leftarrow X - Z - B \qquad \ldots\ldots\ldots (2.32)$$

Further, aided by the use of a proxy variable, we split (2.31) into two sub-problems, each minimizing over a single variable.

Sub-problem 1

$$\min_{Z} \left\| Y - A(Z) \right\|_F^2 + \mu \left\| X - Z - B \right\|_F^2 \qquad \ldots\ldots\ldots (2.33)$$

Sub-problem 2

$$\min_{X} \lambda_n \left\| X \right\|_* + \mu \left\| X - Z - B \right\|_F^2 \qquad \ldots\ldots\ldots (2.34)$$

Sub-problem 1 (2.33), can be recast as a simple least square minimization problem as in (2.35) and solved using any gradient based solver.

$$Z \leftarrow \arg\min_{Z} \left\| \begin{pmatrix} Y \\ \sqrt{\mu}(X - B) \end{pmatrix} - \begin{pmatrix} A \\ \sqrt{\mu} I \end{pmatrix} Z \right\|_F^2 \qquad \ldots\ldots\ldots (2.35)$$

Sub-problem 2 (2.34) can be solved via soft thresholding of singular values [33, 45] – standard approach for nuclear norm minimization – as follows

$$X \leftarrow S_{\frac{2\lambda_n}{\mu}}(Z + B);$$

$$\text{where, } S_\alpha(H) = soft\left(singular\_value(H), \alpha\right) \qquad \ldots\ldots\ldots (2.36)$$

$$\text{and} \quad soft(t, \tau) = sign(t) \max\left(0, |t| - \tau\right)$$

In every iteration, the two sub-problems are alternately solved along with an update of the Bregman variable. Iterations continue until convergence; convergence implying either the difference between objective function value in consecutive iterations falls below a threshold, or a maximum number of iterations are reached. Our complete algorithm, Matrix Completion using Split Bregman (MC_SB) is given in figure 2.3.

Initially, for a few iterations, the algorithm returns over-regularized results. However, continuous update of Bregman variable makes sure that any information that is not

captured is added back. The internal updates (integrated into the algorithmic framework) of the Bregman variable improves the robustness of the algorithm along with achieving faster convergence, mainly due to two reasons. First, the error in the result (current value of optimization variable) is added back which ensures improved accuracy and second, the value of regularization parameters is held constant and thereby optimal values for each of the penalty (norm) terms can be selected. This scheme makes sure that the algorithm achieves higher accurcay and lower run time compared to existing LRMC algorithms.

$Input: Z_0, X_0$ are randomly initialized, maximum iterations m_iter, λ_n

$Output: Z$

while $k \le m\_iter$ *or* $obj\_func(k) - obj\_func(k-1) \le 1e-7$

Solve for Z (original variable)

$$Z \leftarrow \arg\min_{Z} \left\| \begin{pmatrix} Y \\ \sqrt{\mu}(X - B) \end{pmatrix} - \begin{pmatrix} A \\ \sqrt{\mu} I \end{pmatrix} Z \right\|_F^2$$

Solve for X (proxy variable)

$$X \leftarrow soft\left(singular\_value(Z + B), \frac{2\lambda_n}{\mu} \right)$$

Update Bregman Variable

$$B \leftarrow X - Z - B$$

end while

Figure 2.3 Algorithm for Matrix Completion using Split Bregman (MC_SB)

2.5 Experiment and Evaluation

In this section, we discuss the performance of our proposed frameworks and compare the results with those obtained using existing state of the art methods in terms of several relevant evaluation criteria.

2.5.1 Description of Dataset and Evaluation Setup

We evaluate the performance of various recommendation algorithms on the benchmark Movielens dataset [48]. Movielens dataset is a rating database for movies from

movielens.org, compiled and provided by Grouplens, thereby representing a real world (movie) rating database. It is a very popular, publically available dataset used widely for benchmarking and evaluation of recommender system algorithms [49].

We work on two variants of the Movielens dataset – the 100K and the 1M dataset. Both the datasets contain ratings on a scale of 1-5 given by users to movies. Testing on a dataset with integer rating values, instead of binary like or dislike option, helps in more detailed evaluation of the recommendation strategies. The two datasets – 100K and 1M differ in the number of users, items and the sparsity of the rating matrix (i.e. the percentage of ratings available). Details of the two datasets are given in Table 2.1.

Table 2.1 Description of Movielens Datasets

Dataset	Number of Users	Number of Items	Number of Available Ratings	Comments
Movielens 100K	943	1682	100,000	Data cleaned - Minimum 20 movies rated per user
Movielens 1M	6040	3900	1 Million	Data not cleaned

We conduct 5-fold cross-validation, using 80% of the rating data for training and rest 20% forms the test set. The train (rating) data is used to predict the ratings for cases in the testset. Once the ratings are predicted, the predicted values are ranked in descending order to generate top- N_R recommendations for each user. To showcase the robustness of our models, for each test case (fold), 100 simulations are carried out. The standard deviation in each case is less than *1e-3*; however, individual values are not reported. The simulations are carried out on a system with i7-3770S CPU @3.10GHz processor with 8GB RAM.

2.5.2 Evaluation Metrics

The efficiency of various recommender system design algorithms is evaluated on two broad evaluation measures – the recommendation accuracy and the computation cost. The prediction (recommendation) accuracy, an evaluation of the quality of prediction, is

measured in terms of two sets of metrics – rating based metrics and ranking based metrics.

The ranking based metrics – precision and recall - measure the effectiveness of a recommendation algorithm in suggesting relevant items to the user. We report the precision and recall measures [50] for varying number of recommendations $\left(N_R\right)$ for each of the RS design methodologies. Precision (2.37) is a measure of the percentage of suggested items that are relevant and recall (2.38) measures the percentage of the relevant items that are eventually suggested. In the following equations, t_p denotes true positive (item relevant and recommended), f_p is false positive (item irrelevant and recommended) and f_n is false negative (item relevant and not recommended).

$$Precision = \frac{\#t_p}{\#t_p + \# f_p} \quad \text{…..…… (2.37)}$$

$$Recall = \frac{\#t_p}{\#t_p + \# f_n} \quad \text{…..…… (2.38)}$$

We define the relevance of an item based on its rating value; movies rated 4 or higher are marked as relevant, and the rest are classified as irrelevant. This is in line with the conventional nomenclature for Movielens dataset.

Although, ranking based measures are effective at capturing the user's experience, the limited amount of test data renders these measures inadequate in accurately comparing various algorithms, especially if their performance is similar. Thus, we also use rating based measures and conduct a more detailed comparison/evaluation of various RS design models. We compare the algorithms in terms of two rating based metrics – Mean Absolute Error (MAE) (2.39) and Root Mean Square Error (RMSE) (2.40).

$$MAE = \frac{\sum_{m,n\in\Omega} \left| r_{m,n} - \bar{r}_{m,n} \right|}{\left|\Omega\right|} \quad \text{…..…… (2.39)}$$

$$RMSE = \sqrt{\frac{\sum_{m,n \in \Omega} \left(r_{m,n} - \bar{r}_{m,n} \right)^2}{|\Omega|}} \quad \text{…..……} \ (2.40)$$

In the above equations, $r_{m,n}$ is the actual ratings by user m on movie n and $\bar{r}_{m,n}$ is the corresponding predicted rating; Ω is the set of indices of available ratings and $|\Omega|$ is the cardinality of the rating dataset i.e. number of available ratings.

As the speed of prediction is also an important characteristic of a good RS, we also report the run time of various algorithms as a measure of their computation cost.

2.5.3 Results and Discussion

2.5.3.1 Parameter Setting for Proposed Models

We conduct empirical studies to determine the model parameters for our proposed frameworks – eNet_BCS (2.17) and MC_SB (2.28). To determine the number of latent factors for eNet_BCS, we compute the error measure of the proposed model over a range of values (of the number of latent factors). The matrix completion based model (MC-SB), on account of recovering the rating matrix instead of factor matrices, does not require any such input. Figure 2.4 gives variation in error (MAE) as a function of the number of latent factors for the eNet_BCS model for both the datasets.

It is evident from the given figure that a value (of the number of latent factors) in the range of 40-60 is appropriate for both 100K and 1M datasets. Similar results for both the datasets imply that ~50 is the appropriate number of latent factors for the movie database, irrespective of the dimension of the rating matrix i.e. number of users and items. This result also validates the theory of LFM that only a small number of parameters affect the net rating matrix and those are independent of the dimensions of the matrix. Based on the above evaluation, we keep the number of latent factors as 50 for our model.

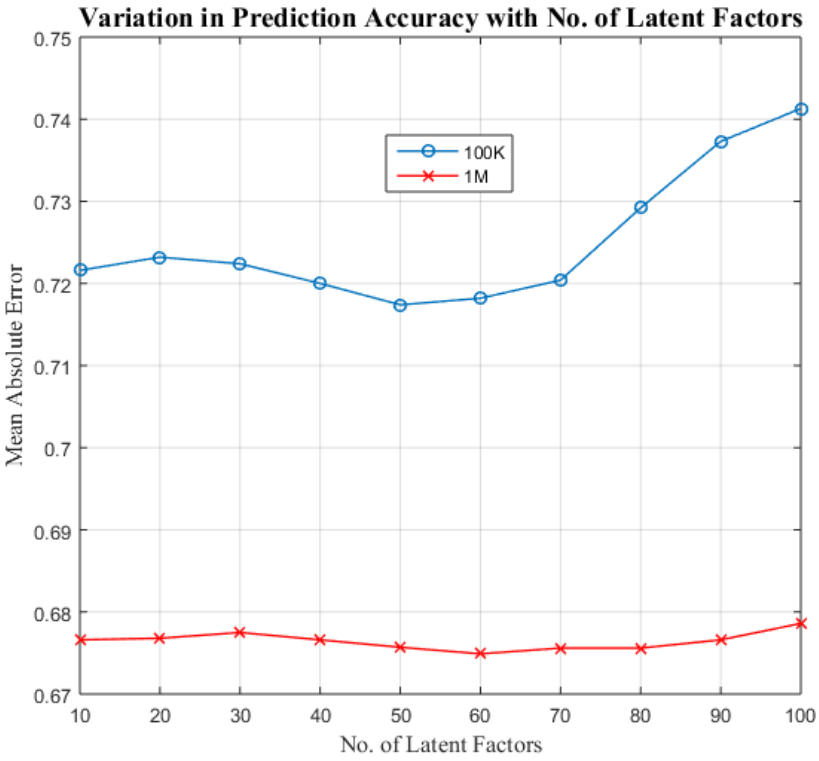


Figure 2.4 Prediction Error as a function of Number of Latent Factors for eNet_BCS

The value of regularization parameters for our proposed frameworks is computed using *l*-curve technique [51]. The values for various regularization parameters are given in Table 2.2.

Table 2.2 Value of Regularization Parameters for Proposed Latent Factor Models

Algorithm	Dataset	λ_u	λ_v	λ	λ_n
MC_SB	100K	-	-	-	$1e-2$
	1M	-	-	-	$1e-2$
eNet_BCS	100K	$1e+3$	$1e-3$	$1e-1$	-
	1M	$1e+4$	$1e-2$	$1e-4$	-

2.5.3.2 Comparison of Proposed Models with Existing Techniques

We compare the performance of our models against the following state-of-the-art algorithms in the field of collaborative filtering and matrix completion:

1. Blind Compressive Sensing Formulation for Collaborative filtering (BCS_CF): In addition to the eNet_BCS model, we also showcase the results of the BCS model (2.16). This highlights the improvement obtained via elastic net regularizer. For the BCS framework, we use the code provided in [52].
2. Accelerated Proximal Gradient (APG) [36]: It is a LRMC algorithm which attempts to improve the speed of computation by partial computation of SVD.
3. Stochastic Gradient Descent (SGD) [22]: It is a conventional approach for solving the standard MF framework.
4. Probabilistic Matrix Factorization (PMF) [28]: It is one the most widely adopted means of MF based RS design wherein the user and item latent factor vectors are modeled as Gaussian priors with a diagonal co-variance matrix.
5. Block Co-ordinate Descent Non-negative Matrix Factorization (BCD_NMF) [53]: It considers regularized block multi-convex optimization problem which is shown to achieve superior solution quality and reduced run time for MF problem compared to state of the art methods.
6. Fixed Point Continuation (FPC) [54]: It is a matrix completion algorithm employing operator splitting and soft thresholding of singular values.

For all the above listed (existing) techniques, codes provided by the authors are used.

Table 2.3 to 2.6 show the comparison of our proposed models with the above-listed frameworks in terms of various accuracy centric evaluation metrics as well as run time.

Table 2.3 Rating based Evaluation Metrics for 100K Movielens Dataset

Algorithm	MAE	RMSE	Run Time (seconds)
BCS_CF	0.7356	0.9409	8.56
eNet_BCS	0.7273	0.9255	2.94
MC_SB	0.7351	0.9319	57.22
APG	0.8847	3.7076	15.01
FPC	0.7527	0.9616	694.52
SGD	0.7432	0.9421	150.34
PMF	0.7564	0.9639	9.02
BCD_NMF	0.7582	0.9816	8.83

Table 2.4 Rating based Evaluation Metrics for 1M Movielens Dataset

Algorithm	MAE	RMSE	Run Time (seconds)
BCS_CF	0.6917	0.8789	171.21
eNet_BCS	0.6899	0.8655	74.19
MC_SB	0.6813	0.8711	979.23
APG	0.9782	3.8109	228.5
FPC	Does not run in reasonable time		
SGD	0.6936	0.8763	1262.5
PMF	0.7240	0.9127	197.3
BCD_NMF	0.6953	0.8890	190.03

Table 2.5 Ranking based Evaluation Metrics for 100K Movielens Dataset

Algo.	Precision					Recall				
	@10	@20	@30	@40	@50	@10	@20	@30	@40	@50
BCS_CF	51.33	38.05	30.14	24.82	21.31	64.16	77.57	82.89	85.47	86.86
eNet_BCS	52.57	38.79	30.69	25.20	21.31	65.22	78.62	84.08	86.64	88.07
MC_SB	51.42	38.41	30.47	25.12	21.21	64.43	78.41	83.86	86.53	87.97
APG	47.43	36.34	27.63	23.19	20.98	60.87	73.11	79.73	83.18	84.96
FPC	50.51	37.91	30.34	24.99	21.87	63.70	76.81	82.79	85.55	87.03
SGD	50.72	37.95	30.28	24.97	21.17	63.76	76.95	82.63	85.25	86.80
PMF	50.52	37.74	30.06	24.71	21.09	63.56	76.76	82.43	85.06	86.56
BCD_NMF	51.33	37.22	29.2	23.88	20.04	64.13	76.77	82.04	84.7	86.10

Table 2.6 Ranking based Evaluation Metrics for 1M Movielens Dataset

Algo.	Precision					Recall				
	@10	@20	@30	@40	@50	@10	@20	@30	@40	@50
BCS_CF	67.19	52.36	42.94	36.16	31.17	62.53	79.91	87.57	91.47	93.82
eNet_BCS	68.64	53.64	43.83	36.91	31.81	63.61	80.61	88.06	92.02	94.39
MC_SB	67.64	52.79	43.33	36.54	31.52	63.33	80.20	87.93	91.96	94.39
APG	60.31	49.22	40.02	33.78	30.10	58.75	75.99	85.22	86.65	90.08
SGD	62.46	49.85	41.3	35.1	30.39	60.35	77.18	84.8	88.81	91.17
FPC	Does not run in reasonable time									
PMF	63.06	50.45	41.9	35.7	30.79	60.32	78.41	86.65	91.17	93.82
BCD_NMF	66.95	52.64	42.57	35.82	30.79	62.51	79.18	86.72	90.69	93.03

Based on the results shown above, following observations can be made:

1. BCS_CF model, i.e. the BCS framework for collaborative filtering outperforms all the existing MF based models (like PMF and BCD_NMF) in terms of

prediction accuracy. The MAE obtained with BCS-CF is at least 1% lower than the other models for the 100K dataset. This validates our intuition behind a sparse item latent factor and its ability to better capture the underlying structure of the rating matrix.

2. Our MC based design, MC_SB outperforms all the standard MF models as well as our proposed BCS model in terms of prediction accuracy, indicating the superiority of MC models owing to the associated convexity. MC_SB shows an improvement of ~1% in terms of MAE over even the proposed BCS model. However, it is less effective than eNet_BCS model as the latter is able to capture the dependencies amongst latent factors as well. Also, because of the need for SVD, the run time of our MC model, MC_SB, is higher than most MF schemes.

3. Our algorithm for matrix completion, MC-SB is compared against two existing LRMC algorithm – APG and FPC. FPC is based on operator splitting and does full SVD computation whereas APG is based on proximal gradient and does partial SVD computation. The impact of partial SVD computation on the recovery accuracy is clearly visible from the results reported for APG; the MAE with APG being 20% higher than our model, MC-SB. Thus, even though the run time for APG is lower than ours, it is not of practical use due to very poor recovery accuracy. FPC algorithm performs much better than APG in terms of error measures but is still poorer than our design. Moreover, the run time for FPC for even the 100K dataset is very large (almost 11 times that of MC-SB); the algorithm is not able to run in reasonable time for the 1M dataset.

   Our design (MC_SB) owing to the use of split Bregman technique, which reduces error while simultaneously improving the convergence behavior, performs better than FPC on both the evaluation criteria – accuracy as well as computation cost. Thus, giving due relevance to both the evaluation metrics, our proposed design is better suited for application to RS design than existing LRMC methods.

4. Our proposed MF based design, eNet_BCS is superior to all schemes compared against in terms of both prediction accuracy as well as run time. The dependencies

amongst latent factor models captured by the use of elastic net regularization yield a better abstraction of the rating data, thereby improving accuracy.

5. In terms of run time, MF methods, except SGD, are much faster than MC based designs for reasons cited above. SGD has very high run time, as it is not suited for problems of such large scale. Further, our Majorization minimization based algorithm helps achieve much lower run time for eNet_BCS model compared to other MF or MC frameworks; run time are lower by almost three times compared to other MF models.

6. Similar results can be inferred from the ranking based measures shown in Table 2.5 and 2.6. Here also, our proposed schemes perform better or at par with existing formulations. However, it can be seen that the ranking based measures do not provide as large a demarcation, as given by rating based metrics, amongst various algorithms.

From the above observations, it can be safely concluded that our proposed frameworks perform on expected lines. eNet_BCS is the best performing model, in terms of both the prediction accuracy as well as the speed of processing.

2.6 Summary

In this chapter, we discussed the latent factor model based formulations for RS design using only the explicit rating information. First, we reviewed the existing techniques for the same. Next, we discussed our two proposed models – the MF fomrulation eNet_BCS and the split Bregman based algorithm for LRMC formulation (MC_SB). The modified MF framework helps capture the essence of rating matrix as well as interaction amongst various latent factors more accurately compared to standard MF models; a claim validated by the comparison of results of the proposed models with existing works. Also, our proposed algorithm for matrix completion framework provides much-improved accuracy within reasonable run times compared to existing LRMC algorithms.

Chapter 3
SUPERVISED FRAMEWORKS FOR LATENT FACTOR MODELS

In the previous chapter, we discussed the recent advances and our proposed formulations in the domain of latent factor models for RS design. However, an improved latent factor model alone does not suffice in significantly improving the prediction accuracy. This is because of the highly sparse nature of the rating dataset, which puts a limit on achievable accuracy. Usually, the rating matrix has less than 10% of the entries available which makes predicting the remaining 90% of the ratings a formidable task.

The problem of limited preference information becomes even more significant in the case of cold start users (new users registering on the system) for whom there is a complete lack of collaborative information. Standard latent factor models, using only the rating information, are not capable of handling the cold start scenario. The inadequacy of any RS in providing meaningful suggestions to new users can cause potential loss of customers and hence a solution to the cold start problem is of substantial relevance.

Fortunately, along with the explicit ratings, certain other pertinent information is also available in RS database; same can be utilized to compensate for the insufficiency of collaborative (rating) data [55, 56]. Such information includes but not restricted to users' demographic profile, their social circle data, and item descriptors or tags. This secondary information can be used to reduce the under-determinacy of the (rating prediction) problem thereby improving the prediction accuracy.

The advantage offered by the use of secondary data has motivated several studies towards building frameworks that harness information from multiple sources in either the memory based [57, 58] or model based set up [56, 59]. Existing works exploit various kinds of metadata – social profile, trust relations or group association of users, tagging information of items and age-gender of users. A few works like [60], build an interview

based design to garner sufficient amount of rating data, wherein they collect user's ratings on a few selected items. However, such a procedure might not be convenient in all scenarios outside the academia. For example, e-retailers such as Amazon or Alibaba does not gather such information from new users and sites garnering such information are becoming increasingly rare.

In this work, we propose models that utilize readily available auxiliary information (metadata) to augment the rating data. We supplement the rating information with users' demographic profile and item categories for improving the quality of prediction. Item categories are easily available on all online portals. For example, the information about movie categories is usually maintained by all online movie portals (like IMDB). Also, during the process of sign-up, a user's demographic information (like age and gender) is often acquired. Hence, this data is easily available along with explicit ratings at no extra cost; making our designs widely applicable. This provides an advantage to our proposed scheme over the frameworks that utilize information such as user's social circle or trust network data, as the latter are laden with concerns of restricted availability and privacy intrusion. Our designs, in addition to improving the prediction accuracy for existing users, are also capable of alleviating the cold start problem. There are no works in our knowledge that (jointly) address the problem of improving prediction accuracy in both the warm start as well the as cold start conditions.

3.1 Review of Existing Models using User and Item Metadata

In this section, we review existing works targeted at accommodating user/item metadata in the collaborative filtering (CF) framework. We also discuss existing approaches for solving the cold start problem.

3.1.1 Incorporating Metadata in Collaborative Filtering Framework

The major bottleneck in rating estimation is the extreme sparsity of the available data. Numerous works have been proposed which augment the CF frameworks, to include

supplementary data, to improve the accuracy of estimation. We review some of them in this section.

Several works include secondary information in the neighborhood based setup. Authors in [57] have proposed a modified similarity measure, to determine the nearest neighbors, combining both the rating data as well as users' demographic information (3.1).

$$similarity_{modified} = similarity_{demographic} \times similarity_{rating} + similarity_{rating} \quad \ldots\ldots\ldots (3.1)$$

where, $similarity_{demographic}$ is computed based on the demographic profile of users and $similarity_{rating}$ is computed using explicit rating data. In [61], a photograph recommendation system has been designed which makes use of geo-spatial information, in addition to rating values. It uses geographical tag data to group photographs into clusters and propagate ratings amongst the members of the same cluster. Thus, a dense rating matrix is obtained which is used as input to a neighborhood-based CF algorithm. Although strategies which incorporate side information in memory based models, help improve accuracy and coverage to some extent, they still suffer from slow computation speed. Thus, the focus of most of the recent works is on exploiting metadata using Latent Factor Model (LFM) based formulations.

In [62] authors have proposed a modified matrix factorization (MF) model (3.2), which includes an additional regularization term, penalizing the deviation of a user's latent factor vector from other users in his/her trust network.

$$\min_{U,V} \left\| Y - A(UV) \right\|_F^2 + \lambda \left(\left\| U \right\|_F^2 + \left\| V \right\|_F^2 \right) + \beta \sum_i \sum_{f \in F_i^+} sim(i, f) \left\| u_i - u_f \right\|_F^2 \quad \ldots\ldots\ldots (3.2)$$

In (3.2), $sim(i, f)$ – similarity amongst users (based on rating pattern) – is used to weigh individual members of trust network differently; u_i is the latent factor vector of user i and u_f denotes the latent factor vector of its neighbors (F_i^+ being the neighborhood) as defined by the trust relationship. The model ensures that users in a common trust network share similar latent factor vectors. Gradient based method has been used to solve the formulation. In [56], authors have used graph regularization to include auxiliary data

about user's demography, social profile, and item categorization in a non-negative matrix factorization framework (3.3).

$$\min_{U,V} \left\| Y - A(UV) \right\|_F^2 + \lambda \left(Tr\left(U^T L_u U \right) \right) + \mu \left(Tr\left(V^T L_v V \right) \right) \quad \text{…..……} \ (3.3)$$

where, L_u and L_v are the graph Laplacians; λ, μ are the regularization parameters. The edges in the user/item graphs are weighted; weights defined by metadata based similarity between users/items. The resulting optimization problem is formulated as a low-rank semi-definite program. A similar graph based strategy has been adopted in [63], where graphical representation is used to populate the sparse rating database using domain information. In [59], social network information has been integrated into the probabilistic matrix factorization (PMF) framework. They constructed a probabilistic framework for modeling latent factor vectors as Gaussian distributed priors whose correlation function captures interdependencies between latent factor vector; users' social network information is used to construct kernels capturing the social/trust relations. Stochastic gradient descent is employed for solving the proposed formulation.

3.1.2 Solving the Cold Start Problem

The problem of providing effective suggestions to new users or recommending new items to existing users – the cold start problem, is a big challenge in RS design. In this section, we review some of the prior art in the area.

Most works in the area assume that there is, albeit highly limited, some rating information available for even the cold start users/items. Thus, these works do not solve the pure cold start problem and use auxiliary data to only supplement the rating information. Authors in [64] have used the available user metadata (demographic details) to model an alpha-community space model. Once a new user's community is defined, one recommendation list per community is generated based on ad-hoc level of agreement recommendation process. Several works like [65] have proposed to modify the similarity measure to make it more suitable for cold start users. The new similarity measure, in addition to the actual ratings, also considers the frequency and count of co-rated items to remove the disparity between users with highly varied rating patterns. Authors in [66]

have utilized the available rating data alone and relied on imputation to reduce data sparsity. They used auto-adaptive imputation method to fill in the missing ratings before applying neighborhood-based scheme for rating prediction. Authors in [67] have proposed a trust based measure to compute similarity and determine neighbors of a new user. Their model is based on the reasoning that in the case of cold start users if rating data is used, the available set of neighbors will be very few. On the other hand, as trust propagates, it provides a measure to include a wider number of users for selection of neighborhood. Authors in [68] have used social tags as a means of relating users to items. The predictions are based on the frequency of tags and the semantic relationships between tags and items.

Unlike most of existing literature, we in this work, propose a solution for the pure cold start problem (no rating information available) for new users as well as new items. Also, our proposed scheme has a cohesive structure targeting effective recommendations for both warm and cold start users.

3.2 Research Contributions

In this chapter, we discuss our proposed frameworks, built to jointly exploit the explicit rating information and user/item metadata, to improve the prediction accuracy. We build our designs using both the MF as well as MC designs as the base formulations. This provides an opportunity to leverage the benefits of each of the base formulation and choose a suitable design as per requirement. We augment our proposed frameworks – the Blind Compressive Sensing (BCS) model for MF and the matrix completion model, discussed in the previous chapter, with the metadata derived constraints to enhance the QoP.

Our models are based on the belief that in the absence of sufficient explicit predilection information, the rating prediction for users (both old and new) can be based on available secondary data like user's demographics and item category. This follows the argument presented in several existing works that users' demographic details such as age and gender influence their likes and dislikes [56, 57]. On similar lines, metadata for items

(such as their category information) can be used to gauge user's interest in them. For example, a user who liked comedies in the past will most likely enjoy comic recommendations. Based on the above understanding, we propose modifications to standard latent factor model to exploit auxiliary information. For all our frameworks, secondary information is in the form of item category (movie genre for Movielens dataset) and user's age, gender, and occupation.

Our main contributions in the area can be summarized as follows:

Models to improve prediction accuracy for warm start scenario

1. We propose modifications to the basic LFM based framework so as to incorporate secondary information along with the rating data to improve prediction accuracy. Our designs are based on the belief that users sharing similar demographic profile tend to display similar affinity pattern towards various latent factors. Under this belief, we define user groups (based on their demographics) and penalize intra-group variability amongst user's rating/affinity pattern. This notion is harnessed to propose modifications to both MF based BCS model as well as matrix completion design as an additional intra-group variance penalty term appended to the base model. Both our frameworks are generic designs which can be customized to include multiple information sources (like age and gender) as per available data. Similarly, for items, we argue that items belonging to same category (genre in the case of movies) will display similar sparsity pattern. To incorporate this belief, we modify the BCS model from being a sparse recovery framework to a joint sparse recovery problem.

Models for improving prediction accuracy for both warm and cold start scenario

2. In this work, our objective is to utilize the available auxiliary data in a latent factor framework using concepts from supervised learning. The novelty of our approach lies in presenting a model that targets improvement in recommendation accuracy for both warm and cold start scenarios. We propose two variants of the model; one based on the BCS framework and other on the MC formulation. We use the available metadata to classify users and items into relevant classes, and the

latent factor matrices for (warm start) users and items are recovered such that in addition to satisfying the (rating) data consistency they are also consistent with the class label information. During optimization, we learn a mapping between the metadata and the rating (latent factor) domain which is used to target the cold start problem, thereby handling both warm start and cold start scenarios in a common framework.

Models combining latent factor and neighborhood formulations

3. In this work, we extend the use of supervised models to jointly harness the information from the neighborhood based as well as model based designs. Neighborhood-based designs are good at extracting local relationships (close associations or pairwise similairty between users or items), and latent factor frameworks capture the global picture (they consider the structure of the complete rating matrix rather than pairwise similarity amongst users/items); thus, a joint model can better predict the ratings by virtue of looking at the rating data from multiple perspectives.

3.3 Latent Factor Models Incorporating Metadata for Improving Prediction Accuracy in Warm Start Scenario

To enhance the prediction quality of the latent factor models, we aim to incorporate user and item metadata into the conventional latent factor formulation. The additional information helps in reducing the underdetermined nature of the problem of rating prediction.

Our design is based on the belief that users sharing similar demographic profile have similar preferences. For example, children in age group of 1-10 will most likely have an affinity for animation movies; similarly, young adults (say 20-30 years) can have an affinity for action/thriller and thus, age can be used to provide additional information about a user's preference. Similarly, women may have, in general, an affinity for rom-com or family genres whereas men might be more inclined towards action; providing a

reason to incorporate gender into the LFMs. We model this understanding as an additional regularization term appended to both the MC as well as the BCS framework.

We put forward a similar strategy for items as well; grouping them based on genre (category) information. We suggest an extension to the BCS framework, wherein the items which belong to a common genre are expected to display similar sparsity patterns. For example; it is more than likely that if two movies belong to genre "comedy," they will most probably not have "action" content. Thus, both will rate high for features corresponding to "comedy" but will have zeros in the positions corresponding to factors related to "action" – thereby imposing similar sparsity pattern on both movies. We exploit this belief in our suggested framework, an improvement over the proposed BCS model, which in turn leads to a joint-sparse representation within the groups.

We also design algorithms to support our proposed frameworks.

3.3.1 Proposed Formulation

3.3.1.1 Blind Compressive Sensing Framework incorporating User Metadata

In this section, we discuss our proposed framework, for the movie recommendation problem, where user's age, gender and occupational information is used to support rating database and improve prediction quality. However, our framework is a generic model which can utilize any available source of metadata to augment the basic (BCS) model.

As stated above, our model rests on the understanding that a user's demographics influence his/her choice; demographically similar users will show similarity in their affinity pattern. This will reflect as similar latent factor representations for demographically similar users. To model our assumption of similar latent factor vectors, we impose the constraint of minimizing the variance amongst the latent factor vectors of users belonging to the same group; groups are defined based on various demographic features. This constraint is included in the BCS formulation as an additional penalty term capturing intra-group variance (3.4).

$$\min_{U,V} \left\| Y - A(UV) \right\|_F^2 + \lambda_u \left\| U \right\|_F^2 + \lambda_v \left\| vec(V) \right\|_1 + \sum_{d=1}^{|D|} \left(\lambda_d \sum_k \text{variance}\left(U_{g^k} \right) \right) \qquad \text{…….} (3.4)$$

where, D denotes the groups formed using a particular demographic trait (like occupation, age or gender); $|D|$ is the cardinality of D i.e. number of different demographic features considered; k represents the indices of groups formed as per a particular demographic trait (like age can be used to form groups $\left(g^1:1-10, g^2:11-20 \text{ and so on}\right); U_{g^k}=\left[u_{g_1^k}|u_{g_2^k}|u_{g_3^k}|..\right]$ is the set of latent factor vectors of users belonging to group g^k (like age group 1-10) and λ_d is the regularization parameter for each group structure; λ_d controls the relative importance given to metadata based measures compared to rating data consistency (1st term).

Variance amongst user's latent factor vectors within a group g^k can be defined as $\sum_{l=1}^{L}\left\|u_{g_l^k}-mean\left(U_{g^k}\right)\right\|_2^2$ where $u_{g_l^k}$ is the latent factor vector for a user l belonging to the group g^k; L being the total number of users in the group g^k. Term representing the summation of all intra-group variances amongst latent factor vectors can be recast in matrix form as in (3.5) which is similar to a laplacian regularizer.

$$\left\|\begin{bmatrix}u_1\\u_2\\.\\u_M\end{bmatrix}-\begin{bmatrix}S_{1,1}&S_{1,2}&.&S_{1,M}\\S_{2,1}&.&.&.\\.&.&.&.\\S_{M,1}&.&.&S_{M,M}\end{bmatrix}\begin{bmatrix}u_1\\u_2\\.\\u_M\end{bmatrix}\right\|_F^2=\left\|U-S_dU\right\|_F^2 \qquad \text{…..…… (3.5)}$$

where, u_* represents latent factor vector of users; M is the number of users; S is the similarity matrix constructed such that $S_{i,j}=1/|g^k|$ if users i and j belong to the same group $\left(g^k\right)$ and 0 otherwise; $|g^k|$ being the number of members (users) in the group g^k.

Using (3.5) we can rewrite (3.4) as

$$\min_{U,V}\left\|Y-A(UV)\right\|_F^2+\lambda_u\left\|U\right\|_F^2+\lambda_v\left\|vec(V)\right\|_1+\sum_{d=1}^{|D|}\left(\lambda_d\left\|U-S_dU\right\|_F^2\right) \qquad \text{…..…… (3.6)}$$

Representing $I_M - S_d$ as G_d; I_M being an $M \times M$ Identity matrix, we can rewrite (3.6) as

$$\min_{U,V} \left\| Y - A(UV) \right\|_F^2 + \lambda_u \left\| U \right\|_F^2 + \lambda_v \left\| vec(V) \right\|_1 + \sum_{d=1}^{|D|} \left(\lambda_d \left\| G_d U \right\|_F^2 \right) \quad \text{…..……} \ (3.7)$$

Equation (3.7) models our notion that users sharing a demographic trait can be characterized by similar latent factor vectors. Use of additional information, for augmenting the rating data, helps improve the robustness and accuracy of our design. The value of regularization parameter λ_d dictates the relevance given to a particular demographic trait while recovery of the latent factor representations.

Although designed for the specific case of exploiting a user's demographic details, our design, Blind Compressive Sensing Framework incorporating User Metadata (BCS_M_User) (3.7) is a generic framework which can incorporate any available metadata (which can be exploited to define groups of similar users) into the BCS model.

3.3.1.2 Matrix Completion Framework incorporating User Metadata

In this section, we discuss our MC based design using user metadata along with the rating information. We adopt the understanding developed in the previous section only, that demographically similar users have similar affinities. But, instead of minimizing the variability amongst the latent factor representation of similar users, we minimize the variability in the rating pattern (predicted rating values) of similar users. In effect, we promote recovery of similar ratings by demographically similar users.

$$\min_{Z} \left\| Y - A(Z) \right\|_F^2 + \lambda_n \left\| Z \right\|_* + \sum_{d=1}^{|D|} \left(\eta_d \sum_k \text{variance}\left(Z_{g^k} \right) \right) \quad \text{…..……} \ (3.8)$$

In (3.8), in addition to quantities defined in (3.4), Z_{g^k} is the set of ratings (across all items) given by users belonging to the group g^k; η_d is the regularization parameter controlling the impact of user demographics.

Following a procedure similar to that outlined above for the BCS_M_User framework, we can recast (3.8) as

$$\min_{Z} \left\| Y - A(Z) \right\|_F^2 + \lambda_n \left\| Z \right\|_* + \sum_{d=1}^{|D|} \left(\eta_d \left\| T_d Z \right\|_F^2 \right) \quad \text{…..…… (3.9)}$$

where, $T_d = I_M - S_d$; S_d defined as in (3.5)

Formulation (3.9), showing our formulation – Matrix Completion Framework incorporating User Metadata (MC_User) is an adaptation of (3.7) to the matrix completion framework.

3.3.1.3 Blind Compressive Sensing Framework incorporating Item Metadata

Next, we discuss the proposed modification to the BCS formulation to incorporate item metadata.

In this work, we use item category (specifically, movie genre) as the additional data source. Most portals maintain a database of such information, like a movie renting portal or an online book website, have information about the categories to which the items belong. However, we cannot extend the scheme used for incorporating user's metadata to items. This is because each item might belong to multiple genres, for example, a movie may belong to two genres – drama and romance. In such a case, each item is a part of multiple groups making the intra-group variance based formulation highly complex with large interdependencies. Our proposed approach to include item metadata into the BCS framework is discussed below.

It was argued in Chapter 2 that an item's latent factor vector is sparse (BCS Framework), i.e. only a few latent factors have non-zero values – corresponding to traits possessed by the item. Building upon this notion, we exploit the belief that items that share a genre (category) tend to display similar sparsity pattern. To understand this better, consider two books authored by Arthur C. Doyle. They will have suspense or action but highly unlikely to possess features such as comedy. Thus, the latent factor vectors defining the two books will rate highly on factors corresponding to suspense, action, and adventure but will have zeros for those corresponding to comedy. Hence, they will display similar sparsity pattern. To capture this understanding, we modify the BCS framework by replacing the sparsity constraint $(l_1 \text{ norm})$ by a group sparse penalty term $(l_{21} \text{ norm})$ as in (3.10).

$$\min_{U,V} \left\|Y - A(UV)\right\|_F^2 + \lambda_u \left\|U\right\|_F^2 + \lambda_v \sum_{s=1}^{|C|} \left\|V_{C^s}\right\|_{2,1}$$
$$where, \left\|X\right\|_{2,1} = \sum_i \left\|X^i\right\|_2 ; X^i \text{ is the } i^{th} \text{ row of matrix } X \qquad \text{…..……} (3.10)$$

where, C defines the item groups, one for each category; $|C|$ is the cardinality of C i.e. number of different categories; V_{C^s} is the set of latent factor vectors of items belonging to the category C^s. Our proposed formulation, Blind Compressive Sensing Framework incorporating Item Metadata (BCS_M_Item) minimizes the l_{21} norm across latent factor vectors of items belonging to each (same) category. Use of group sparsity constraint promotes recovery of latent factor vectors having similar sparsity pattern.

3.3.2 Algorithm Design

In this section, we discuss algorithm design for our three formulation – one incorporating user metadata in BCS model (3.7), second for user metadata in MC design (3.9) and third, integrating item metadata in BCS framework (3.10).

3.3.2.1 Blind Compressive Sensing Framework incorporating User Metadata

First, we discuss the user metadata based BCS design, repeated here for convenience

$$\min_{U,V} \left\|Y - A(UV)\right\|_F^2 + \lambda_u \left\|U\right\|_F^2 + \lambda_v \left\|vec(V)\right\|_1 + \sum_{d=1}^{|D|} \left(\lambda_d \left\|G_d U\right\|_F^2\right)$$

Similar to the procedure outlined in section 2.3, we use Majorization minimization technique [40] to recast our formulation as

$$\min_{U,V} \left\|W - (UV)\right\|_F^2 + \lambda_u \left\|U\right\|_F^2 + \lambda_v \left\|vec(V)\right\|_1 + \sum_{d=1}^{|D|} \left(\lambda_d \left\|G_d U\right\|_F^2\right)$$
$$\text{where, } W = (U_k V_k) + \frac{1}{\alpha} A^T \left(Y - A(U_k V_k)\right) \qquad \text{…..……} (3.11)$$
$$\text{and, } \quad \alpha \geq \max\left(eigen\_value\left(A^T A\right)\right)$$

Equation (3.11), though bilinear can be split into two sub-problems, each minimizing over a single variable while maintaining the other constant as follows

Sub-problem 1

$$\min_{U} \left\| W - (UV) \right\|_F^2 + \lambda_u \left\| U \right\|_F^2 + \sum_{d=1}^{|D|} \left(\lambda_d \left\| G_d U \right\|_F^2 \right) \quad \text{…..…… (3.12)}$$

Sub-problem 2

$$\min_{V} \left\| W - (UV) \right\|_F^2 + \lambda_v \left\| vec(V) \right\|_1 \quad \text{…..…… (3.13)}$$

Equation (3.12) can be recast as a simple least squares problem as in (3.14) and solved using any gradient descent method.

$$\min_{U} \left\| \begin{pmatrix} vec(W) \\ 0 \\ 0 \\ .. \\ 0 \end{pmatrix} - \begin{pmatrix} V^T \otimes I \\ \sqrt{\lambda_u}(I \otimes I) \\ \sqrt{\lambda_1}(I \otimes G_1) \\ .. \\ \sqrt{\lambda_{|D|}}\left(I \otimes G_{|D|}\right) \end{pmatrix} vec(U) \right\|_2^2 \quad \text{…..…… (3.14)}$$

Sub-problem 2 (3.13) can be solved using iterative soft thresholding [44] as follows

$$\begin{aligned} & V \leftarrow Soft\left(B, \frac{\lambda_v}{2\beta}\right) \\ & \text{where, } B = V + {}^{1}\!/\!_{\beta}\left(U^T (W - UV)\right) \\ & \qquad \beta \geq \max\left(eigen\_value\left(U^T U\right)\right) \\ & \text{and} \quad Soft(t,u) = sign(t)\ \max(0, |t| - u) \end{aligned} \quad \text{…..…… (3.15)}$$

Both the sub-problems (3.14, 3.15) are alternately solved until convergence i.e. either a maximum number of iterations are reached or reduction in objective function falls below the threshold over consecutive iterations. The complete algorithm (BCS_M_User) is given in figure 3.1.

Input : U_0, V_0 are randomly initialized, maximum iterations $m\_iter, \lambda_u, \lambda_v, \lambda_d$

Output : U, V

while $k \le m\_iter$ *or* $obj\_func(k) - obj\_func(k-1) \le 1e-7$

$$W = \left(U_{k-1}V_{k-1}\right) + \frac{1}{\alpha} A^T \left(Y - A\left(U_{k-1}V_{k-1}\right)\right)$$

Solve for U

$$U_k \leftarrow \arg\min_U \left\| \begin{pmatrix} vec(W) \\ 0 \\ 0 \\ .. \\ 0 \end{pmatrix} - \begin{pmatrix} {V_{k-1}}^T \otimes I \\ \sqrt{\lambda_u}\left(I \otimes I\right) \\ \sqrt{\lambda_1}\left(I \otimes G_1\right) \\ .. \\ \sqrt{\lambda_{|D|}}\left(I \otimes G_{|D|}\right) \end{pmatrix} vec(U) \right\|_2^2$$

Solve for V

$$V_k \leftarrow Soft\left(B, \frac{\lambda_v}{2\beta}\right)$$

where, $B = V + {}^{1}\!/\!_{\beta}\left({U_k}^T \left(W - U_k V\right)\right)$

$$\beta \ge \max\left(eigen\_value\left({U_k}^T U_k\right)\right)$$

end while

Figure 3.1 Algorithm for Blind Compressive Sensing Framework incorporating User Metadata (BCS_M_User)

3.3.2.2 Blind Compressive Sensing Framework incorporating Item Metadata

In this section, we discuss the algorithm design using MM technique [40] for our formulation proposed using item metadata (3.10), repeated below for convenience

$$\min_{U,V} \left\|Y - A(UV)\right\|_F^2 + \lambda_u \left\|U\right\|_F^2 + \lambda_v \sum_{s=1}^{|C|} \left\|V_{C^s}\right\|_{2,1}$$

Applying MM technique and splitting the above problem into two simpler sub-problems, we get (3.16) and (3.17), where $W = \left(U_k V_k\right) + \frac{1}{\alpha} A^T \left(Y - A\left(U_k V_k\right)\right)$

Input : U_0, V_0 are randomly initialized, maximum iterations $m\_iter, \lambda_u, \lambda_v$

Output : U, V

while $k \le m\_iter$ *or* $obj\_func(k) - obj\_func(k-1) \le 1e-7$

$$W = \left(U_{k-1}V_{k-1}\right) + \frac{1}{\alpha} A^T \left(Y - A\left(U_{k-1}V_{k-1}\right)\right)$$

Solve for U

$$U_k \leftarrow \arg\min_U \left\| \begin{pmatrix} W \\ \mathbf{0} \end{pmatrix} - U \begin{pmatrix} V_{k-1} \\ I \end{pmatrix} \right\|_2^2 ; \mathbf{0} \text{ is a matrix of all zeros}$$

Solve for V

$$V_{C^s} \leftarrow Soft\_2\left(B, \frac{\lambda_v}{2\beta}\right)$$

where, $B = V_{C^s} + {}^{1}\!/_{\beta}\left(U_k{}^T \left(W_{C^s} - U_k V_{C^s}\right)\right)$

$$\beta \ge \max\left(eigen\_value\left(U_k{}^T U_k\right)\right)$$

end while

Figure 3.2 Algorithm for Blind Compressive Sensing Framework incorporating Item Metadata (BCS_M_Item)

Sub-problem 1

$$\min_U \|W - UV\|_F^2 + \lambda_u \|U\|_F^2 \quad \text{…..…… (3.16)}$$

Sub-problem 2

$$\min_V \|W - UV\|_F^2 + \lambda_v \sum_s \left\|V_{C^s}\right\|_{2,1} \quad \text{…..…… (3.17)}$$

Sub-problem 1 is a simple least square expression solvable using any gradient descent scheme. For sub-problem 2, modified soft thresholding [69] is employed as in (3.18), to solve for each item group separately, here, W_{C^s} is the set of columns of W corresponding to those in V_{C^s} ,

$$V_{C^s} \leftarrow Soft\_2\left(B, \frac{\lambda_v}{2\beta}\right)$$

$$\text{where, } B = V_{C^s} + \frac{1}{\beta}\left(U^T\left(W_{C^s} - UV_{C^s}\right)\right) \qquad \text{…..…… (3.18)}$$

$$\beta \geq \max\left(eigen\_value\left(U^T U\right)\right)$$

$$\text{and} \quad Soft\_2(\mathbf{t}, u) = \frac{\mathbf{t}}{\|\mathbf{t}\|_2} \max\left(0, \|\mathbf{t}\|_2 - u\right)$$

An item's latent factor vector is computed as an average of latent factor vectors obtained from each group data. The two sub-problems (3.16 and 3.17) are alternately solved until convergence. The complete algorithm (BCS_M_Item) is given in figure 3.2.

3.3.2.3 Matrix Completion Framework incorporating User Metadata

We discuss the algorithm design for MC based design for incorporating user metadata using split Bregman technique [46]. The formulation is repeated here for convenience.

$$\min_Z \|Y - A(Z)\|_F^2 + \lambda_n \|Z\|_* + \sum_{d=1}^{|D|}\left(\eta_d \|T_d Z\|_F^2\right)$$

Following the procedure discussed in chapter 2 (section 2.4), we introduce proxy variables $\left(P_g\right)$ in the above expression to enable split Bregman type splitting of norm terms (3.19)

$$\min_Z \|Y - A(Z)\|_F^2 + \lambda_n \|Z\|_* + \sum_{d=1}^{|D|}\left(\eta_d \|T_d P_d\|_F^2\right) + \sum_{d=1}^{|D|}\left(\mu_d \|P_d - Z - B_d\|_F^2\right) \qquad \text{…..…… (3.19)}$$

where, B_d denote the Bregman relaxation variables used to enforce equality at convergence between original and proxy variables; μ_d's are the regularization parameters. Equation (3.19) can be split into two (simpler) sub-problems by variable splitting as follows

Sub-problem 1

$$\min_Z \|Y - A(Z)\|_F^2 + \lambda_n \|Z\|_* + \sum_{d=1}^{|D|}\left(\mu_d \|P_d - Z - B_d\|_F^2\right) \qquad \text{…..…… (3.20)}$$

Sub-problem 2

$$\min_{P_d} \sum_{d=1}^{|D|} \left(\eta_d \left\| T_d P_d \right\|_F^2 \right) + \sum_{d=1}^{|D|} \left(\mu_d \left\| P_d - Z - B_d \right\|_F^2 \right) \quad \text{…..……} \quad (3.21)$$

Sub-problem 2 involves minimization over each of the proxy variables individually.

The first sub-problem (3.20) can be recast as follows

$$\min_{Z} \left\| \begin{pmatrix} Y \\ \sqrt{\mu_1}\left(P_1 - B_1\right) \\ \sqrt{\mu_2}\left(P_2 - B_2\right) \\ \cdots \\ \sqrt{\mu_{|D|}}\left(P_{|D|} - B_{|D|}\right) \end{pmatrix} - \begin{pmatrix} A \\ \sqrt{\mu_1} I \\ \sqrt{\mu_2} I \\ .. \\ \sqrt{\mu_{|D|}} I \end{pmatrix} Z \right\|_F^2 + \lambda_n \left\| Z \right\|_* \quad \text{…..……} \quad (3.22)$$

Equation (3.22) can be solved by soft thresholding of singular values [33, 34] as follows

$$Z \leftarrow Soft\left(Singular\_value(T), \frac{\lambda_n}{2\alpha} \right)$$

$$\text{where, } T = Z + {}^{1}\!/_{\alpha} \left(\begin{pmatrix} A \\ \sqrt{\mu_1} I \\ \sqrt{\mu_2} I \\ .. \\ \sqrt{\mu_{|D|}} I \end{pmatrix}^T \left(\begin{pmatrix} Y \\ \sqrt{\mu_1}\left(P_1 - B_1\right) \\ \sqrt{\mu_2}\left(P_2 - B_2\right) \\ \cdots \\ \sqrt{\mu_{|D|}}\left(P_{|D|} - B_{|D|}\right) \end{pmatrix} - \begin{pmatrix} A \\ \sqrt{\mu_1} I \\ \sqrt{\mu_2} I \\ .. \\ \sqrt{\mu_{|D|}} I \end{pmatrix} Z \right) \right)$$

$$\alpha \geq \max \left(eigen\_value \begin{pmatrix} A \\ \sqrt{\mu_1} I \\ \sqrt{\mu_2} I \\ .. \\ \sqrt{\mu_{|D|}} I \end{pmatrix}^T \begin{pmatrix} A \\ \sqrt{\mu_1} I \\ \sqrt{\mu_2} I \\ .. \\ \sqrt{\mu_{|D|}} I \end{pmatrix} \right)$$

$$\text{and} \quad Soft(t,u) = sign(t)\ \max\left(0, |t| - u\right) \quad \text{…..……} \quad (3.23)$$

Sub-problem 2 (3.21) is a simple least squares formulation (for each proxy variable) which can be efficiently solved using a gradient based solver.

Input : Z_0, X_0 is randomly initialized, maximum iterations $m\_iter, \lambda_n, \eta_d$

Output : Z

while $k \le m\_iter$ *or* $obj\_func(k) - obj\_func(k-1) \le 1e-7$

Solve for Z (original variable)

$$Z \leftarrow Soft\left(Singular\_value(T), \frac{\lambda_n}{2\alpha} \right)$$

where, $T = Z + 1/\alpha \left(\begin{pmatrix} A \\ \sqrt{\mu_1} I \\ \sqrt{\mu_2} I \\ .. \\ \sqrt{\mu_{|D|}} I \end{pmatrix}^T \left(\begin{pmatrix} Y \\ \sqrt{\mu_1}(P_1 - B_1) \\ \sqrt{\mu_2}(P_2 - B_2) \\ ... \\ \sqrt{\mu_{|D|}}(P_{|D|} - B_{|D|}) \end{pmatrix} - \begin{pmatrix} A \\ \sqrt{\mu_1} I \\ \sqrt{\mu_2} I \\ .. \\ \sqrt{\mu_{|D|}} I \end{pmatrix} Z \right) \right)$

$$\alpha \ge \max \left(eigen\_value \begin{pmatrix} A \\ \sqrt{\mu_1} I \\ \sqrt{\mu_2} I \\ .. \\ \sqrt{\mu_{|D|}} I \end{pmatrix}^T \begin{pmatrix} A \\ \sqrt{\mu_1} I \\ \sqrt{\mu_2} I \\ .. \\ \sqrt{\mu_{|D|}} I \end{pmatrix} \right)$$

Solve for P_d (proxy variable)

$$P_d \leftarrow \arg\min_{P_d} \sum_{d=1}^{|D|} \left(\eta_d \left\| T_d P_d \right\|_F^2 \right) + \sum_{d=1}^{|D|} \left(\mu_d \left\| P_d - Z - B_d \right\|_F^2 \right)$$

Update Bregman Variable

$$B_d \leftarrow B_d + Z - P_d$$

end while

Figure 3.3 Algorithm for Matrix Completion Framework Incorporating User Metadata (MC_User)

Each iteration for solving the above sub-problems is followed by an update of all the Bregman variables as follows

$$B_d = B_d + Z - P_d \qquad \text{…..……} \quad (3.24)$$

Iterations continue till convergence i.e. either the decrease in objective function reaches a threshold or maximum numbers of iterations are reached. The complete algorithm for Matrix completion using user metadata (MC_User) is given in figure 3.3.

3.4 Recommender System Models for Improving Accuracy in both Warm and Cold Start Scenario

The works discussed in the previous section are directed towards improving the accuracy of recommendation in the warm start scenario, i.e. consider only those users and items for whom collaborative data is available. However, the problem of inadequate data is even more pronounced for new users and items, being introduced in the system, for whom there are no available ratings.

Fortunately, the RS does have access to the metadata for even the cold start users and items. A new user registering on the portal is often required to provide basic details at sign-up such as age and gender. Similarly, for any new item being added to the repository, category information is invariably available. This metadata can be used to solve the cold start problem.

In this section, we discuss our proposed formulations that use rating information along with user demographics and item categories to target the problem of rating prediction for both old and new users and items. We introduce required modification in both the matrix factorization (BCS) and matrix completion (MC) models. To the best of our knowledge, such a comprehensive framework has not been proposed before.

Our model involves classifying users and items into several overlapping clusters, based on the information derived from available metadata. For example, users may be classified based on their demographic profile; Similarly, items can be classified based on their categories (say genre for movies). The class labeling is not restrictive, and a user/item can simultaneously belong to multiple classes. Our primary model recovers the latent factor matrices for warm start users and items such that in addition to satisfying the (rating) data consistency they are also consistent with the given class label information. A simple extension of our principal formulation presents a solution to the pure (item and user) cold

start problem. To this effect, we learn a label consistency map – relating user or item label data to their respective latent factor vectors – almost on the side-lines, while solving our primary formulation. This label consistency map along with the auxiliary data of new users/items can be used to make predictions for them. Our proposition is based on the belief that users belonging to the same class (say same demographic profile) will tend to display similar preferences. Similarly, items belonging to a common genre will possess similar (characterizing) latent factor vectors. Thus, label consistency map learned for existing users or items is equally applicable to new users and items as well. Our formulation helps solve the pure cold start problem for both users and items, unlike most existing works capable of solving only the partial cold start problem.

3.4.1 Proposed Formulation

3.4.1.1 Incorporating Auxiliary Data in Latent Factor Framework

In this section, we discuss our design for a latent factor model based RS which uses explicit ratings, user demography and item genre information. We introduce our model considering BCS framework has the base formulation. However, the design can be extended to MC model as well, as shown later.

We borrow ideas from supervised learning [70] to incorporate user and item metadata into our base - BCS model (3.25)

$$\min_{U,V} \left\| Y - A(UV) \right\|_F^2 + \lambda_v \left\| vec(V) \right\|_1 + \lambda_u \left\| U \right\|_F^2 \qquad \text{…..……} \ (3.25)$$

Our model is constructed on the premise that users can be clubbed or categorized into several classes (of similar users) based on the available secondary information. We define multiple classes based on gender, age brackets, and different occupational profiles; the user can simultaneously belong to multiple classes. Using this label information a user-class label matrix (L_u) is defined, such that $L_u(m,c)=1$ if the user m belongs to class c else 0. Let us consider an example; wherein we form 2 distinct gender groups – male and female, *P* distinct non-overlapping age groups (say 1-17, 18-24 and so on) and *Q* distinct occupational categories. The label matrix (L_u) will have a row corresponding to each user

and columns corresponding to (*2+P+Q*) classes. Let us consider a user (*User 1*), who is a male in the age group of 18-24 and a lawyer by profession. The classification information of this user can be used to fill up the first row of L_u as shown in figure 3.4. Similarly, for a female in the age group of 60+ and an artist by profession, the corresponding row will be as shown in row 2 and so on.

	Gender 1 (M)	Gender 2 (F)	Age 1 (1-17)	Age 2 (18-24)	....	Age P (60+)	Occ. 1 (Tech.)	Occ. 2 (Artist)	....	Occ. Q (Lawyer)
User 1	1	0	0	1	0	0	0	0	0	1
User 2	0	1	0	0	0	1	0	1	0	0
..	..	..	..	..	..	..	..	..	..	..
User M	1	0	0	1	0	0	1	0	0	0

Figure 3.4 Example of the User Label Matrix

This class label information is used to learn the latent factor vectors of users in a supervised environment, such that they are consistent with both the class information as well as the available rating data. The use of secondary information (class label) provides additional constraints (thereby restricting the search space) which in effect reduces the under determinacy of the problem.

We model this idea as an additional regularization term appended to the base BCS formulation (3.25) which penalizes any deviation from class label consistency as in (3.26)

$$\min_{U,V,C} \left\|Y - A(UV)\right\|_F^2 + \lambda_v \left\|vec(V)\right\|_1 + \lambda_u \left\|U\right\|_F^2 + \mu_u \left\|L_u - UC\right\|_F^2 \qquad \text{…….} (3.26)$$

where, $C \in \mathbb{R}^{F \times C_u}$, C_u is the number of classes constructed based on user metadata; μ_u is the regularization parameter which controls the relative importance given to the label consistency constraint in comparison to data consistency.

The matrix *C*, which is learned as part of the optimization process, is essentially a linear map from the user latent factor space to the classification domain of users. User latent

factor matrix (U) is learned so as to be consistent with the class label information via the mapping defined by *C*.

The idea is extended to items as well. All items are categorized based on their genre; each item belonging to one or more genres. We append this item label information to (3.26) as follows

$$\min_{U,V,C,D} \left\|Y - A(U \times V)\right\|_F^2 + \lambda_v \left\|vec(V)\right\|_1 + \lambda_u \left\|U\right\|_F^2 + \mu_u \left\|L_u - UC\right\|_F^2 + \mu_v \left\|L_v - DV\right\|_F^2 \quad \text{..... (3.27)}$$

where, item class information matrix (L_v) is constructed on similar lines as (L_u) i.e. $L_v(p,n) = 1$ if item n belongs to class p else 0 and $D \in \mathbb{R}^{C_v \times F}$ is the linear map from item latent factors to the respective class domain (C_v is the number of classes constructed based on item metadata). Item latent factor matrix (V) is learned so as to be consistent with the available class label (genre) information. Equation (3.27) gives our final formulation – Label consistent Blind Compressive Sensing Framework (LC_BCS).

Introducing supervised learning into the standard latent factor framework helps improve prediction accuracy by alleviating the problem of data scarcity. Using (3.27) for rating prediction helps improves prediction accuracy for warm start users.

We can extend the model discussed above to the MC framework as well as in (3.28)

$$\min_{Z,S_u,S_v} \left\|Y - A(Z)\right\|_F^2 + \lambda_n \left\|Z\right\|_* + \eta_u \left\|L_u - ZS_u\right\|_F^2 + \eta_v \left\|L_v - S_v Z\right\|_F^2 \quad \text{..... (3.28)}$$

where, $S_u \in \mathbb{R}^{F \times C_u}, S_v \in \mathbb{R}^{C_v \times F}$ are the linear mappings between the ratings and the user and item classes respectively and η_v, η_u are the regularization parameters. Equation (3.28) – Label Consistent Matrix Completion Framework (LC_MC) presents the MC version of our supervised model.

In our formulation we have used user demographics and item category; however, the model can easily accommodate any other pertinent classification criteria (such as social

network information or item descriptors like movie cast/director) as per the available information.

3.4.1.2 Addressing the Cold Start Problem

Our proposed model targets both the partial as well as pure cold start problem. For the partial cold start problem i.e. for users or items with limited collaborative information models presented in (3.27) and (3.28) can be applied. The lack of adequate amount of rating data is compensated by use of metadata and helps improve (rating) prediction accuracy.

More importantly, a simple extension to our label consistent formulation generates relevant recommendations in pure cold start conditions as well. The category information of new items, added to the repository, is invariably available in a RS database. Also, when a new user registers on the system, he/she is more often than not required to enter his/her age/gender-related information. Thus for a new user or a new item, even though no rating data is available, associated metadata is readily available.

Focussing on the BCS based formulation (3.27), we use the linear map (C and D), generated almost on the side-lines while solving the proposed model, along with the available metadata to predict ratings for new users or (on) new items.

First, let us consider the new user cold start problem. Our design is based on the idea that users' demographics have an impact on their preference. Thus, users having similar demographic profiles can be defined by similar latent factor vectors. This theory can be exploited by using the mapping (C) between users' classification domain and their latent factor vectors to generate latent factors for new users if classification data is available.

When a new user registers on the system, his/her relevant demographic information is captured. We use this information along with the linear (information) map C to estimate the latent factor vector for the new user. As discussed above, matrix C is a map from latent factor space to user class domain. It primarily establishes a generic relation between a user's demographic profile and its latent factor vector. Hence, it gives

information about the user's preference for features given his/her age, gender and occupational profile which can be exploited for cold start users.

Let $l_{new\_user} \in \mathbb{R}^{1\times C_u}$ be the class label vector for the new user. The class label vector and the linear map C are related as follows

$$l_{new\_user} = u_{new\_user} \times C \quad \text{…..…… (3.29)}$$

where, $u_{new\_user} \in R^{1\times F}$ is the latent factor vector of the new user.

Equation (3.29) can be solved efficiently using any conjugate gradient based solver to estimate new user's latent factor vector. Once u_{new_user} is recovered, (interaction component) rating by the user can be computed as $z_{new\_user} = u_{new\_user} \times V$, where V is the latent factor matrix for existing items.

A similar model can be constructed for new items as well. Here also, we argue that items belonging to the same genre will share similarities in their latent factor vectors. Thus, the mapping from item classification to item latent factor space, derived using existing items' auxiliary data, can be used for new items as well.

A new item can be characterized in terms of its genre as a vector $l_{new\_item} \in R^{C_v\times 1}$. For item cold start case, we use the information matrix D which relates latent factor space with the item label space. Matrix D establishes a relation between the latent factor vector of an item and its category (genre). For a new item, we have the genre information and hence, similar to the case for new user, its latent factor vector $\left(v_{new\_item}\right)$ can be estimated using (3.30)

$$l_{new\_item} = D \times v_{new\_item} \quad \text{…..…… (3.30)}$$

Using previously estimated user latent factor matrix (U), users' rating (preference) for the new item can be computed as $z_{new\_item} = U \times v_{new\_item}$.

Similar processing can be carried out for the MC framework (3.28) as well, wherein the ratings by new users or on new items can be estimated using their metadata by solving (3.31)

$$\begin{aligned} l_{new\_user} &= z_{new\_user} S_u \\ l_{new\_item} &= S_v z_{new\_item} \end{aligned} \qquad \text{…..……} \quad (3.31)$$

Our design procedure can thus be used to solve both new user and new item pure cold start problem, without significant add-on computations; requiring only the solution to a linear system of equations.

3.4.2 Algorithm Design

3.4.2.1 Label Consistent Blind Compressive Sensing Framework

In this section, we discuss the design of our algorithm for BCS based formulation, LC_BCS, repeated here for convenience.

$$\min_{U,V,C,D} \left\| Y - A(UV) \right\|_F^2 + \lambda_v \left\| vec(V) \right\|_1 + \lambda_u \left\| U \right\|_F^2 + \mu_u \left\| L_u - UC \right\|_F^2 + \mu_v \left\| L_v - DV \right\|_F^2$$

First, we employ Majorization minimization [40] and variable splitting to split the above equation into simpler sub-problems, each minimizing over a single variable; with $W = (UV) + \frac{1}{\alpha} A^T \left(Y - A(UV) \right)$ and $\alpha \geq \max\left(eigen\_value\left(A^T A \right) \right)$

Sub-problem 1

$$\min_U \left\| W - (UV) \right\|_F^2 + \lambda_u \left\| U \right\|_F^2 + \mu_u \left\| L_u - UC \right\|_F^2 \qquad \text{…..……} \quad (3.32)$$

Sub-problem 2

$$\min_V \left\| W - (UV) \right\|_F^2 + \lambda_v \left\| vec(V) \right\|_1 + \mu_v \left\| L_v - DV \right\|_F^2 \qquad \text{…..……} \quad (3.33)$$

Sub Problem 3

$$\min_C \left\| L_u - UC \right\|_F^2 \qquad \text{…..……} \quad (3.34)$$

Sub Problem 4

$$\min_{D} \left\| L_v - DV \right\|_F^2 \qquad \text{…..…… (3.35)}$$

Input : U_0, V_0 are randomly initialized, maximum iterations $m\_iter, \lambda_u, \lambda_v, \mu_u, \mu_v$

Output : U, V

while $k \le m\_iter$ *or* $obj\_func(k) - obj\_func(k-1) \le 1e-7$

$$W = \left(U_{k-1}V_{k-1}\right) + \frac{1}{\alpha} A^T \left(Y - A\left(U_{k-1}V_{k-1}\right)\right)$$

Solve for U

$$U_k \leftarrow \arg\min_{U} \left\| \begin{pmatrix} W \\ \mathbf{0} \\ \sqrt{\mu_u} L_u \end{pmatrix} - U \begin{pmatrix} V_{k-1} \\ I \\ \sqrt{\mu_u} C \end{pmatrix} \right\|_F^2$$

Solve for V

$$V_k \leftarrow Soft\left(B, \frac{\lambda_v}{2\alpha}\right)$$

$$\text{where, } B = V + {}^{1}\!/_{\beta} \left(\begin{pmatrix} U_k \\ \sqrt{\mu_v} D \end{pmatrix}^T \left(\begin{pmatrix} W \\ \sqrt{\mu_v} Q \end{pmatrix} - \begin{pmatrix} U_k \\ \sqrt{\mu_v} D \end{pmatrix} V \right) \right)$$

$$\beta \ge \max \left(eigen\_value \begin{pmatrix} U_k \\ \sqrt{\mu_v} D \end{pmatrix}^T \begin{pmatrix} U_k \\ \sqrt{\mu_v} D \end{pmatrix} \right)$$

Solve for Linear Maps

$$C \leftarrow \min_{C} \left\| L_u - UC \right\|_F^2$$

$$D \leftarrow \min_{D} \left\| L_v - DV \right\|_F^2$$

end while

Figure 3.5 Algorithm for Label Consistent Blind Compressive Sensing Framework (LC_BCS)

Sub-problem 1 can be cast as a least square problem (3.36) and thus has a closed form solution.

$$\min_{U} \left\| \begin{pmatrix} W \\ \mathbf{0} \\ \sqrt{\mu_u} L_u \end{pmatrix} - U \begin{pmatrix} V \\ I \\ \sqrt{\mu_u} C \end{pmatrix} \right\|_F^2 ; \mathbf{0} \text{ is a matrix of all zeros} \quad \text{…..……} \quad (3.36)$$

Sub-problem 2 can be solved by iterative soft thresholding [44] as follows

$$V \leftarrow Soft\left(B, \frac{\lambda_v}{2\alpha} \right)$$

$$\text{where, } B = V + {}^{1}\!/_{\beta} \left(\left(\begin{pmatrix} U \\ \sqrt{\mu_v} D \end{pmatrix}^T \left(\begin{pmatrix} W \\ \sqrt{\mu_v} Q \end{pmatrix} - \begin{pmatrix} U \\ \sqrt{\mu_v} D \end{pmatrix} V \right) \right) \quad \text{…..……} \quad (3.37)$$

$$\beta \geq \max \left(eigen\_value \left(\begin{pmatrix} U \\ \sqrt{\mu_v} D \end{pmatrix}^T \begin{pmatrix} U \\ \sqrt{\mu_v} D \end{pmatrix} \right) \right)$$

$$\text{and} \quad Soft(t,u) = sign(t) \ \max(0, |t| - u)$$

Sub-problems 3 and 4 are simple least squares; efficiently solvable by any conjugate gradient method. For cold start problem, we can again solve two least square expressions, (3.29) and (3.30) using conjugate gradient. The complete algorithm (LC_BCS) is given in figure 3.5.

3.4.2.2 Label Consistent Matrix Completion Framework

In this section, we present the algorithm for our MC based label consistent formulation (3.28), repeated here for convenience.

$$\min_{Z, S_u, S_v} \left\| Y - A(Z) \right\|_F^2 + \lambda_n \left\| Z \right\|_* + \eta_u \left\| L_u - Z S_u \right\|_F^2 + \eta_v \left\| L_v - S_v Z \right\|_F^2$$

To use split Bregman technique [46], we introduce proxy variables (P and Q) in the above formulation as in (3.38), where, B_u, B_v are the Bregman variables and δ_u, δ_v are the regularization parameters.

$$\min_{Z,S_u,S_v,P,Q} \left\|Y - A(Z)\right\|_F^2 + \lambda_n \left\|Z\right\|_* + \eta_u \left\|L_u - QS_u\right\|_F^2 + \eta_v \left\|L_v - S_v P\right\|_F^2 + \delta_v \left\|P - Z - B_v\right\|_F^2 + \delta_u \left\|Q - Z - B_u\right\|_F^2 \quad \text{…..…… (3.38)}$$

Next, we split our formulation into simpler sub-problems, minimizing over each variable separately.

Sub-problem 1

$$\min_{Z} \left\|Y - A(Z)\right\|_F^2 + \lambda_n \left\|Z\right\|_* + \delta_v \left\|P - Z - B_v\right\|_F^2 + \delta_u \left\|Q - Z - B_u\right\|_F^2 \quad \text{…..…… (3.39)}$$

Sub-problem 2

$$\min_{P} \eta_v \left\|L_v - S_v P\right\|_F^2 + \delta_v \left\|P - Z - B_v\right\|_F^2 \quad \text{…..…… (3.40)}$$

Sub-problem 3

$$\min_{Q} \eta_u \left\|L_u - QS_u\right\|_F^2 + \delta_u \left\|Q - Z - B_u\right\|_F^2 \quad \text{…..…… (3.41)}$$

Sub-problem 4

$$\min_{S_u} \left\|L_u - QS_u\right\|_F^2 \quad \text{…..…… (3.42)}$$

Sub-problem 5

$$\min_{S_v} \left\|L_v - S_v P\right\|_F^2 \quad \text{…..…… (3.43)}$$

Sub-problems 2 and 3 can be cast as least square expression as in (3.44) and (3.45) respectively.

$$\min_{P} \left\| \begin{pmatrix} \sqrt{\eta_v} L_v \\ \sqrt{\delta_v}\left(Z + B_v\right) \end{pmatrix} - \begin{pmatrix} \sqrt{\eta_v} S_v \\ \sqrt{\delta_v} I \end{pmatrix} P \right\|_F^2 \quad \text{…..…… (3.44)}$$

$$\min_{Q} \left\| \begin{pmatrix} \sqrt{\eta_u} L_u \\ \sqrt{\delta_u}\left(Z + B_u\right) \end{pmatrix} - Q \begin{pmatrix} \sqrt{\eta_u} S_u \\ \sqrt{\delta_u} I \end{pmatrix} \right\|_F^2 \qquad \text{…..…… (3.45)}$$

Sub-problem 1, can be solved by soft thresholding of singular values [33] as follows

$$Z \leftarrow Soft\left(Singular\_value\left(T\right), \frac{\lambda_n}{2\alpha} \right)$$

$$\text{where, } T = Z + \frac{1}{\alpha} \left(\begin{pmatrix} A \\ \sqrt{\delta_v} I \\ \sqrt{\delta_u} I \end{pmatrix}^T \left(\begin{pmatrix} Y \\ \sqrt{\delta_v}\left(P - B_v\right) \\ \sqrt{\delta_u}\left(Q - B_u\right) \end{pmatrix} - \begin{pmatrix} A \\ \sqrt{\delta_v} I \\ \sqrt{\delta_u} I \end{pmatrix} Z \right) \right) \qquad \text{…..…… (3.46)}$$

$$\alpha \geq \max \left(eigen\_value \begin{pmatrix} A \\ \sqrt{\mu_v} I \\ \sqrt{\mu_u} I \end{pmatrix}^T \begin{pmatrix} A \\ \sqrt{\mu_v} I \\ \sqrt{\mu_u} I \end{pmatrix} \right)$$

$$\text{and} \quad Soft\left(t, u\right) = sign\left(t\right) \ \max\left(0, \left|t\right| - u\right)$$

Sub-problems 4 and 5 are simple least squares, solvable using any conjugate gradient type solver. In each iteration, Bregman variables are updated as follows

$$\begin{aligned} B_u &\leftarrow B_u + Z - Q \\ B_v &\leftarrow B_v + Z - P \end{aligned} \qquad \text{…..…… (3.47)}$$

The iterations continue till convergence. The complete algorithm (LC_MC) is given in figure 3.6.

Input : Z_0, X_0 is randomly initialized, maximum iterations $m\_iter, \lambda_n, \eta_u, \eta_v$

Output : Z

while $k \le m\_iter$ *or* $obj\_func(k) - obj\_func(k-1) \le 1e-7$

Solve for Z (original variable)

$$Z \leftarrow Soft\left(Singular\_value(T), \frac{\lambda_n}{2\alpha} \right)$$

$$\text{where, } T = Z + \frac{1}{\alpha}\left(\begin{pmatrix} A \\ \sqrt{\delta_v} I \\ \sqrt{\delta_u} I \end{pmatrix}^T \left(\begin{pmatrix} Y \\ \sqrt{\delta_v}(P - B_v) \\ \sqrt{\delta_u}(Q - B_u) \end{pmatrix} - \begin{pmatrix} A \\ \sqrt{\delta_v} I \\ \sqrt{\delta_u} I \end{pmatrix} Z \right) \right)$$

$$\alpha \ge \max\left(eig \begin{pmatrix} A \\ \sqrt{\mu_v} I \\ \sqrt{\mu_u} I \end{pmatrix}^T \begin{pmatrix} A \\ \sqrt{\mu_v} I \\ \sqrt{\mu_u} I \end{pmatrix} \right)$$

Solve for P and Q (proxy variables)

$$P \leftarrow \arg\min_P \left\| \begin{pmatrix} \sqrt{\eta_v} L_v \\ \sqrt{\delta_v}(Z + B_v) \end{pmatrix} - \begin{pmatrix} \sqrt{\eta_v} S_v \\ \sqrt{\delta_v} I \end{pmatrix} P \right\|_F^2$$

$$\mathrm{Q} \leftarrow \arg\min_Q \left\| \begin{pmatrix} \sqrt{\eta_u} L_u \\ \sqrt{\delta_u}(Z + B_u) \end{pmatrix} - Q \begin{pmatrix} \sqrt{\eta_u} S_u \\ \sqrt{\delta_u} I \end{pmatrix} \right\|_F^2$$

Solve for Linear Maps

$$S_u \leftarrow \min_{S_u} \left\| L_u - Q S_u \right\|_F^2$$

$$S_v \leftarrow \min_{S_v} \left\| L_v - S_v P \right\|_F^2$$

Update Bregman Variable

$$B_u \leftarrow B_u + Z - Q$$

$$B_v \leftarrow B_v + Z - P$$

end while

Figure 3.6 Algorithm for Label Consistent Matrix Completion Framework (LC_MC)

3.5 Combining Latent Factor Models with Neighbourhood Formulation

We use the label consistent formulations proposed in section 3.4 to accommodate the principles of latent factor model and the neighborhood-based setup for collaborative filtering in a joint framework. Our proposed model view the ratings from multiple perspectives – global (looking at rating matrix in its entirety) as well as local (extracting patterns/relationship between a small group of users and items).

In standard neighborhood schemes, the (local) neighborhood of target user/item is searched for, and thus strong local relationships are easily captured. On the contrary, LFM considers the entire rating dataset as a single entity and are thereby effective in capturing the global structure. Our multi-view approach helps extract greater information – both global and local - from the available ratings [71].

In our proposed approach, the LFM based formulation helps harness the information embedded in the structure of the entire rating matrix. We augment the LFM based frameworks, both MC and BCS, by incorporating the strong (local) correlation amongst similar users/items computed using neighborhood-based method. To capture the strong locality embedded in the available rating data, we employ the clustering technique. Clustering methods, though limited by their low accuracy and poor coverage, have been shown to capture strong local associations [2]. Several existing works use hard clustering i.e. a user or an item belongs to only one cluster. However, users/items can seldom be expected to be similar to a single set of users/items; they will demonstrate some degree of association/similarity with multiple sets. In light of the same, we employ fuzzy C-means clustering [72] which allows the user or item to belong to multiple clusters with varying degree of membership. The clusters hence formed are used to define a label vector for each user and item. Conventional LFM based designs provide improved coverage but lack the ability to capture strong user-user or item-item correlation; for capturing local patterns neighborhood models are more suited. Thus combining the two, via the use of regularization terms help build a better framework for rating prediction.

3.5.1 Proposed Formulation

3.5.1.1 Capturing Local Relations – Fuzzy Clustering Approach

Clustering is essentially an unsupervised learning technique which finds the relation amongst data points - data instances that are similar to (or near) each other are placed in one cluster, and distinct data instances are placed in separate clusters. It is a widely adopted technique for finding groups of similar users and items in RS design [73, 74].

Conventionally adopted K-means clustering allows a data point (user or item) to belong to just one cluster, ignoring the possible relationship it shares with members of other clusters. However, as stated above, such a clustering can be very restrictive (and not capture the true picture) in the case of recommender systems.

Hence, in our proposed formulation, we employ Fuzzy K-means clustering – which allows user/items to belong to several clusters with varying degree of membership [75]. Fuzzy clustering involves solving the following optimization problem

$$\arg\min_{c} \sum_{i=1}^{n}\sum_{j=1}^{k} w_{ij}^{m} d(s_i, \mu_j);\;\; w_{ij}{}^{m} = \left(\sum_{f=1}^{k} \left(\frac{d(s_i, \mu_j)}{d(s_i, \mu_f)} \right)^{\frac{2}{m-1}} \right)^{-1} \quad \text{…..……} \quad (3.48)$$

where, $d(s_i, \mu_j)$ denotes the distance between the data point (user/item) s_i and cluster centroid μ_j of the j^{th} cluster; n being the total number of data points and k is the total number of clusters; w_{ij}^{m} is the membership weight or degree to which point i belongs to cluster j; m is the fuzzifier s.t. $m \geq 1$.

We cluster both users and items using the approach highlighted above, with the rating data (vector) for each user/item acting as the input. The most commonly used distance metric for clustering is Euclidean distance. However, as the rating data (vector) representing each user/item is highly sparse (0's in places where a user has not rated, or an item has not been rated), Euclidean distance is not the best metric. Instead, in this work we employ cosine similarity metric (3.49) for clustering, which is inherently capable of taking care of zeros in the structure.

$$d(x,y) = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2 \sum_{i=1}^{n} y_i^2}} \quad \text{…..……} \ (3.49)$$

3.5.1.2 Combining Local Information with Global Structure

Next, we integrate the local associations into the latent factor model based designs.

To enable the same, we represent each user (and item) by a label vector l_u (l_v) of length equal to the number of clusters. Each element of the label vector i.e. $l_u(c)$ denotes the degree of association of *u*, with the cluster *c* – as found using the fuzzy clustering approach. This information is used to introduce additional constraints – as add-on penalty terms – to both the matrix factorization as well as matrix completion formulations for LFM, like metadata derived labels in section 3.4.

Our final formulations (3.50) and (3.51) are similar to LC_BCS and LC_MC with the exception that the label vectors are now defined using information from Fuzzy clustering step, instead of being based on metadata.

MC_Neigh: $\min_{U,V,C,D} \|Y - A(UV)\|_F^2 + \lambda_v \|vec(V)\|_1 + \lambda_u \|U\|_F^2 + \mu_u \|L_u - UC\|_F^2 + \mu_v \|L_v - DV\|_F^2$

…..…… (3.50)

BCS_Neigh: $\min_{Z,S_u,S_v} \|Y - A(Z)\|_F^2 + \lambda_n \|Z\|_* + \eta_u \|L_u - ZS_u\|_F^2 + \eta_v \|L_v - S_v Z\|_F^2$ …..…… (3.51)

3.6 Experiment and Evaluation

In this section, we discuss the performance of our proposed frameworks and compare the results with those obtained using existing state of the art methods in terms of relevant evaluation measures. We briefly summarise the method discussed in this chapter before proceeding further.

Table 3.1 Summary of Proposed Methods

Algorithm	Acronym	Objective Function
Blind Compressive Sensing with User Metadata	BCS_M_User	$\min_{U,V} \|Y - A(UV)\|_F^2 + \lambda_u \|U\|_F^2 + \lambda_v \|vec(V)\|_1 + \sum_{d=1}^{\|D\|} \left(\lambda_d \|G_d U\|_F^2 \right)$
Blind Compressive Sensing with Item Metadata	BCS_M_Item	$\min_{U,V} \|Y - A(UV)\|_F^2 + \lambda_u \|U\|_F^2 + \lambda_v \sum_{s=1}^{\|C\|} \|V_{C^s}\|_{2,1}$ $where, \|X\|_{2,1} = \sum_i \|X^i\|_2$; X^i is the i^{th} row of matrix X
Matrix completion with User Metadata	MC_User	$\min_Z \|Y - A(Z)\|_F^2 + \lambda_n \|Z\|_* + \sum_{d=1}^{\|D\|} \left(\eta_d \|T_d Z\|_F^2 \right)$ where, $T_d = I_M - S_d$; S_d defined as in (3.5)
Label Consistent Blind Compressive Sensing	LC_BCS	$\min_{U,V,C,D} \|Y - A(U \times V)\|_F^2 + \lambda_v \|vec(V)\|_1 + \lambda_u \|U\|_F^2 + \mu_u \|L_u - UC\|_F^2 + \mu_v \|L_v - DV\|_F^2$
Label Consistent Matrix completion	LC_MC	$\min_{Z,S_u,S_v} \|Y - A(Z)\|_F^2 + \lambda_n \|Z\|_* + \eta_u \|L_u - ZS_u\|_F^2 + \eta_v \|L_v - S_v Z\|_F^2$
Blind Compressive Sensing with Neighborhood data	BCS_Neigh	$\min_{U,V,C,D} \|Y - A(UV)\|_F^2 + \lambda_v \|vec(V)\|_1 + \lambda_u \|U\|_F^2 + \mu_u \|L_u - UC\|_F^2 + \mu_v \|L_v - DV\|_F^2$
Blind Compressive Sensing with Neighborhood data	MC_Neigh	$\min_{Z,S_u,S_v} \|Y - A(Z)\|_F^2 + \lambda_n \|Z\|_* + \eta_u \|L_u - ZS_u\|_F^2 + \eta_v \|L_v - S_v Z\|_F^2$

3.6.1 Description of Dataset and Evaluation Setup

We evaluate the performance of various recommendation algorithms on the two benchmark Movielens datasets – 100K and 1M datasets. We conduct 5-fold cross-validation, using 80% of rating data for training and rest 20% for test. The simulations are carried out on a system with i7-3770S CPU @3.10GHz processor with 8GB RAM.

We augment the rating data with user and item metadata provided in the database; we use users' age, gender, and occupation and movie genre for the purpose. The manner in which the metadata is employed in each of the proposed schemes is detailed below.

BCS_M_User, BCS_M_Item and MC_User

For 100K dataset, the groups formed are Male and female in age brackets of 1-10, 11-20, 21-30, 31-40, 41-50, 51-60, 61-70, 71-80. Thus we have a total of 18 groups of similar users as per their age-gender combination. We also grouped users by their occupation; wherein there are 21 different occupations and an equal number of groups.

In case of 1M dataset, there are 7 predefined age groups: 1-17, 18-24, 25-34, 35-44, 45-49, 50-55 and 56+. So, a total of 14 groups are made with age-gender (M/F) information. Similar to 100K dataset, users are divided into 21 groups based on their occupation.

LC_BCS, LC_MC

For both the 100K and 1M datasets following classes are formed. For classification of users (in both datasets) we consider 30 classes/categories - 7 age groups (1-17, 18-24, 25-34, 35-44, 45-49, 50-55, 56+), 2 gender groups (M/F) and 21 occupational categories. Items are assigned to one or more of 19 classes, each representing a single genre.

MC_Neigh, BCS_Neigh

For extracting neighborhood information, fuzzy clustering is employed. The degree of fuzzification or the fuzzifier (m in Eq. 3.49) is kept as 2 for all cases. The cosine similarity (distance) was used as the distance measure for clustering, as reasoned above.

For both 100K and 1M datasets, users are clustered into 10 clusters, and for items, 18 clusters are formed. These values are found using empirical evaluation.

3.6.2 Evaluation Metrics

The comparison of our proposed models with the existing works is carried out in terms of accuracy using ranking and rating based measures.

The Ranking based metrics used for evaluation are Precision (3.52) and recall (3.53)

$$Precision = \frac{\#t_p}{\#t_p + \#f_p} \quad \text{…..…… (3.52)}$$

$$Recall = \frac{\#t_p}{\#t_p + \#f_n} \quad \text{…..…… (3.53)}$$

where, t_p denotes true positive (item relevant and recommended), f_p is false positive (item irrelevant and recommended) and f_n is false negative (item relevant and not recommended). Items rated 4 or higher are marked relevant whereas those rated below that are considered irrelevant.

Rating based metrics that are employed for evaluation are MAE (3.54) and RMSE (3.55).

$$MAE = \frac{\sum_{m,n\in\Omega} \left| r_{m,n} - \bar{r}_{m,n} \right|}{\left|\Omega\right|} \quad \text{…..…… (3.54)}$$

$$RMSE = \sqrt{\frac{\sum_{m,n\in\Omega} \left(r_{m,n} - \bar{r}_{m,n} \right)^2}{\left|\Omega\right|}} \quad \text{…..…… (3.55)}$$

In the above equations, $r_{m,n}$ is the actual ratings by user m on movie n and $\bar{r}_{m,n}$ is the corresponding predicted ratings; Ω is the set of indices of available ratings and $\left|\Omega\right|$ is the cardinality of the rating dataset i.e. number of available ratings.

3.6.3 Results and Discussion

3.6.3.1 Parameter Setting for Proposed Models

The value of regularization parameters for our proposed frameworks is computed using *l*-curve technique [51]. The selected values of regularization parameters for all the proposed schemes are given below.

Table 3.2 Value of Regularization Parameters for Warm Start Models

Algorithm	Dataset	λ_u	λ_v	λ_d / η_d	λ_n
BCS_M_User	100K	$1e+3$	$1e-1$	$1e-1$, $1e-1$	-
	1M	$1e+4$	$1e-2$	-	-
BCS_M_Item	100K	$1e+3$	$1e-1$	$1e-1$	-
	1M	$1e+4$	$1e-1$	-	-
MC_User	100K & 1M	-	-	$1e-1$, $1e-2$	$1e+1$

Table 3.3 Value of Regularization Parameters for Label Consistent Models

Algorithm	Dataset	λ_u	λ_v	μ_u / η_u	μ_v / η_v	λ_n
LC_BCS	100K & 1M	$1e+2$	$1e-2$	$1e-1$	$1e+1$	-
LC_MC	100K & 1M	-	-	$1e-1$	$1e-1$	$1e+1$
MC_Neigh	100K & 1M	-	-	$1e-1$	$1e-1$	$1e+1$
BCS_Neigh	100K & 1M	$1e+1$	$1e+1$	$1e-1$	$1e-1$	-

3.6.3.2 Comparison with Existing Techniques (Warm Start Scenario)

In this section, we illustrate the impact of incorporating metadata into the standard LFM based formulations which use only the rating data.

As we have already shown (in Chapter 2) the superiority of our proposed models – BCS, eNet_BCS (elastic net regularized blind compressive sensing), and MC_SB over the existing latent factor frameworks, here we compare our proposed models only against these frameworks.

We also compare our proposed models existing works which use metadata to augment the rating information. The works compared against are:

1. Nearest Neighbor Model using Secondary Information (KNN_M) [57]: It is a neighborhood-based model wherein the similarity between users is computed based on two information sources – explicit rating and user metadata.
2. Nonnegative Matrix Factorization using Graph Regularization (GR_M) [56]: In this paper, authors suggested to augment the non-negative MF model with an

additional regularization terms derived from the graphical representation of the user/item model constructed using the available metadata.

3. Semi-supervised Nonnegative Matrix Factorization (SSNMF) [76]: In this work, a semi-supervised learning based factorization model is proposed for general matrix factorization problem. They augmented the basic non-negative MF framework to exploit information from multiple sources for generating the factor matrices.

Further, we also compare our models with two works that propose modifications to LFM but use only the rating data – both attempt to mitigate the sparsity of rating matrix.

4. Matrix Approximation via clustering (MA_C) [77]: In this work, authors borrow the idea from code-book transfer to take advantage of the implicit similarity between user and item vectors by deriving cluster level rating pattern. It has been shown to help reduce the impact of rating data sparsity.

5. Factored Item Similarity Model (FISM) [78]: In this work, authors present an item-based method for generating top-N recommendations that learn the item-item similarity matrix as the product of two low-dimensional latent factor matrices. These matrices are learned using a structural equation modeling approach, wherein the value being estimated is not used for its estimation.

For all the above frameworks, codes provided by the authors are used.

Table 3.4, 3.5 and 3.6 lists the results obtained for all the algorithms on the Movielens 100K and 1M datasets.

Table 3.4 Rating based Evaluation Metrics for Movielens Datasets

Algorithm	100K Dataset		1M Dataset	
	MAE	RMSE	MAE	RMSE
Standard Latent Factor Model Designs using Ratings alone				
BCS_CF	0.7356	0.9409	0.6917	0.8789
eNet_BCS	0.7273	0.9255	0.6899	0.8655
MC_SB	0.7351	0.9319	0.6813	0.8711
Proposed Supervised Models				
BCS_M_User	0.7200	0.9191	0.6744	0.8623
BCS_M_Item	0.7217	0.9229	0.6733	0.8622
MC_User	0.7206	0.9187	0.6749	0.8622
LC_BCS	0.7199	0.9146	0.6709	0.8567
LC_MC	0.7193	0.9145	0.6731	0.8559
BCS_Neigh	0.7224	0.9273	0.6836	0.8689
MC_Neigh	0.7221	0.9271	0.6800	0.8640
Existing Models using Metadata with Ratings				
KNN_M	0.8302	1.0146	0.8198	0.9989
GR_M	0.7577	0.9616	0.7233	0.9139
SSNMF	0.7723	1.0112	0.7285	0.9401
Existing Modification to Latent Factor Models using Rating Alone				
MA_C	0.8828	1.0572	1.0222	1.2502
FISM	0.7431	0.9439	0.7196	0.9102

Table 3.5 Ranking based Evaluation Metrics for 100K Movielens Dataset

Algo.	Precision					Recall				
	@10	@20	@30	@40	@50	@10	@20	@30	@40	@50
Standard Latent Factor Model Designs using Ratings alone										
BCS_CF	51.33	38.05	30.14	24.82	21.31	64.16	77.57	82.89	85.47	86.86
eNet_BCS	52.57	38.79	30.69	25.20	21.31	65.22	78.62	84.08	86.64	88.07
MC_SB	51.42	38.41	30.47	25.12	21.21	64.43	78.41	83.86	86.53	87.97
Proposed Supervised Models										
BCS_M_ User	52.49	38.88	30.5	25.30	21.45	65.11	78.73	84.16	86.76	88.15
BCS_M_ Item	52.41	38.87	30.48	25.21	21.42	65.10	78.67	84.14	86.73	88.11
MC_User	52.51	38.92	30.5	25.30	21.45	65.11	78.73	84.16	86.76	88.15
LC_BCS	52.48	38.91	30.82	25.38	21.47	65.14	78.75	84.15	86.77	88.18
LC_MC	52.55	38.99	31.01	25.42	21.51	65.17	78.77	84.19	86.80	88.21
BCS_Neigh	52.63	38.80	30.34	24.91	20.95	64.62	78.48	83.95	86.32	87.96
MC_Neigh	52.72	38.96	30.46	24.97	20.95	64.76	78.65	83.98	86.36	88.03

Table 3.5 Ranking based Evaluation Metrics for 100K Movielens Dataset (contd.)

Algo.	Precision					Recall				
	@10	@20	@30	@40	@50	@10	@20	@30	@40	@50
Existing Models using Metadata with Ratings										
KNN_M	35.65	23.09	15.22	9.97	6.14	49.21	62.52	68.11	70.92	72.37
GR_M	50.96	38.39	30.52	25.27	21.45	64.51	77.82	83.41	86.32	87.67
SSNMF	51.6	38.24	30.46	25.06	21.21	64.79	79.23	83.8	86.49	87.96
Existing Modification to Latent Factor Models using Rating Alone										
MA_C	49.30	36.92	29.37	24.36	20.59	67.95	82.09	87.65	90.57	91.98
FISM	43.95	34.41	38.14	23.79	20.46	63.20	75.15	80.39	83.27	84.07

Table 3.6 Ranking based Evaluation Metrics for 1M Movielens Dataset

Algo.	Precision					Recall				
	@10	@20	@30	@40	@50	@10	@20	@30	@40	@50
Standard Latent Factor Model Designs using Ratings alone										
BCS_CF	67.19	52.36	42.94	36.16	31.17	62.53	79.91	87.57	91.47	93.82
eNet_BCS	68.64	53.64	43.83	36.91	31.81	63.61	80.61	88.06	92.02	94.39
MC_SB	67.64	52.79	43.33	36.54	31.52	63.33	80.20	87.93	91.96	94.39
Proposed Supervised Models										
BCS_M_ User	67.99	53.01	43.11	36.2	35.98	63.54	80.23	87.76	91.98	94.31
BCS_M_ Item	67.95	52.99	43.09	36.17	35.97	63.50	80.21	87.73	91.97	94.28
MC_User	67.99	53.01	43.11	36.2	35.98	63.54	80.23	87.76	91.98	94.31
LC_BCS	68.22	53.17	43.59	36.72	35.64	63.67	80.47	88.15	92.15	94.51
LC_MC	68.23	53.21	43.62	36.77	35.71	63.68	80.52	88.19	92.16	94.53
BCS_Neigh	67.78	52.79	42.81	35.91	31.79	63.14	79.65	87.17	91.07	93.84
MC_Neigh	67.55	52.83	42.94	35.81	31.72	63.22	80.18	87.54	91.18	93.91
Existing Models using Metadata with Ratings										
KNN_M	45.23	30.91	21.74	15.15	10.25	41.73	58.63	66.48	70.63	73.12
GR_M	65.99	52.03	42.82	36.26	31.38	62.43	76.69	87.54	91.69	94.23
SSNMF	66.23	51.91	42.74	36.15	31.25	62.73	79.63	87.48	91.63	94.12
Existing Modification to Latent Factor Models using Rating Alone										
MA_C	58.03	47.18	39.61	34.06	29.80	58.64	76.55	84.90	89.63	92.57
FISM	65.69	51.42	42.26	35.67	30.75	62.2	79.12	87	91.15	93.63

Based on the results listed in the tables, following observations can be made

1. It is clearly evident that user/item metadata can be effectively used to improve the prediction accuracy. All our proposed schemes employing metadata show an increase in prediction accuracy over the base models – BCS_CF or MC_SB. Metadata based designs show an improvement over even the eNet_BCS framework as the latter relies on dependencies derived from implicit and intuitive modeling whereas the former are built on dependencies derived from actual data (user/item metadata).
2. Compared to the base BCS formulation, BCS frameworks including metadata (BCS_M_User and BCS_M_Item) show a reduction of ~2% in MAE for 100K dataset. The improvement is even more prominent in the case of 1M dataset (MAE is lower by ~2.5%) as the latter suffers from more severe rating data sparsity than the 100K dataset.
3. Our label consistent models which use both user and item metadata (LC_BCS and LC_MC), help achieve a further reduction in error measures over methods using only one of them (like BCS_M_User and MC_User). The improvement in MAE is ~1% for the 100K dataset and ~0.5% for 1M dataset. Further, with the use of metadata, the divide between MC and MF models reduces; both perform at par with each other.
4. Our models embedding LFM designs with neighborhood information (BCS_Neigh, MC_Neigh), are also able to improve upon the standard LFM based designs (like eNet_BCS, MC_SB). The reduction in MAE is around 2% for the 100K dataset and 0.2% for 1M dataset. The lower gain for 1M dataset stems from the fact that the highly sparse rating matrix does not facilitate adequate extraction/harnessing of the neighborhood information. However, these models, as expected, do not perform as well as models using metadata along with the rating information.
5. Existing models using user/item metadata show poorer performance than our designs, especially the neighborhood based scheme (KNN_M). This is owing to

the limitations of the basic neighborhood formulation which are not sufficiently mitigated by the use of metadata also. Further, the LFM based frameworks (GR_M and SSNMF) are non-negative matrix factorization models, i.e. they work with raw rating values and no baseline correction is done. Thus, the input to the models is noisy, adversely affecting their performance.

6. The existing modified frameworks using rating data alone (MA_C, FISM) also perform poor compared to our proposed models using only the rating data – BCS_Neigh and MC_Neigh.
7. The observations made above are supported by the ranking based measures as well. All our proposed methods perform better than existing frameworks in terms of precision and recall measures also.

Thus, it can be safely concluded that our designs offers an effective framework targeting substantial gain in recovery accuracy for warm start users/items.

3.6.3.3 Comparison with Existing Techniques (Cold Start Scenario)

In this section, we illustrate the efficiency of our formulation in solving pure user/item cold start problem.

Table 3.7 shows the performance of our models for user and item cold start problem

Table 3.7 Rating based Evaluation Metrics for Movielens Datasets (Cold Start)

Algorithm	100K Dataset		1M Dataset	
	MAE	RMSE	MAE	RMSE
LC_BCS (User)	0.7275	0.9224	0.7082	0.8984
LC_BCS (Item)	0.7273	0.9214	0.7176	0.9148
LC_MC (User)	0.7275	0.9217	0.7100	0.8984
LC_MC (Item)	0.7271	0.9214	0.7099	0.8983

The results reported in Table 3.6 clearly indicate that even for new users and items, our models generate significantly relevant suggestions. The results are poor compared to those reported for warm start scenario, which is understandable given the lack of any collaborative data. However, as the linear mapping (from the rating domain to

classification domain), used for cold start situation, is learned using both the rating data as well as metadata, it is robust and works well even in the case of cold start scenario. It can be seen that our designs yield better results for cold start condition than (few) other works for even warm start users. This is credited to our efficient use of available data compared to existing frameworks.

There are very limited works that target the pure cold start problem and almost all concentrate on the new user problem and not on the item cold start. We compare our models to two recent works [79, 80]. Both the works solve only the user cold start problem.

In [79] a hybrid scheme based on SCOAL algorithm is proposed to alleviate the user cold start problem. SCOAL is used to cluster together users and builds a separate prediction model for each cluster. For each new user, the closest cluster is identified (based on demographic details), and its preference is predicted based on the applicable model. They report an MAE of 0.93 for the 100K dataset, ~28% higher than MAE of our models. In [58] authors use a combination of existing classification (computed based on demographic information) techniques and similarity-based prediction mechanisms to retrieve recommendations. Their experiment on the 1M Movielens dataset gives an MAE of 0.75 and RMSE of 0.95. Our model shows an improvement of around 6% over the reported values.

Hence, our label consistent formulations are shown to yield substantial improvement in prediction accuracy in both warm start and cold start scenario.

3.7 Summary

In this chapter, we discussed our modifications of the standard latent factor models which are capable of exploiting both the rating data as well as the user and item metadata for improving prediction accuracy. A review of the existing methods for incorporating metadata into CF frameworks was followed by a detailed discussion of our proposed models. We propose two kinds of strategies - one using metadata to improve prediction accuracy for existing users and other focussed on the design of comprehensive RS

targeting the warm start and cold start scenario jointly. For both the design scheme, we proposed MF as well as MC based models so as to provide the freedom to select a suitable model as per design requirement. Our proposed models are shown to outperform existing frameworks designed for using rating information along with secondary information.

Chapter 4
ACCURACY-DIVERSITY BALANCE IN RECOMMENDER SYSTEMS

Most works in RS design [56, 67, 71], including those discussed so far, quantify the efficiency of these systems in terms of prediction accuracy. However, such an accuracy centric design, with emphasis on recommending only similar items may lead to dreariness, and over time customers may lose interest in the suggested item list. On the other hand, heterogeneity in the recommendations can assist in maintaining customers' interest in the RS [80].

The necessity for diversified recommendations and studies [81, 82] suggesting the need for a broader measure of recommendation quality have motivated several works in the recent past that concentrate on improving novelty, diversity, and visibility of long tail items [83, 84, 85]. However, the increase in diversity comes at the cost of accuracy. This stems from the fact that accuracy of RS focuses on recommending items highly similar to a user's past preference whereas, diversity promotes digression from the users' past choices. Thus, the challenge in the design of an effective recommendation process is to make novel and diverse suggestions while maintaining sufficient relevance to a user's past choice (i.e. ensure high accuracy).

Existing works on balancing accuracy and diversity in recommendations can be divided into two categories – two stage recommendation strategies and unified models. The former class of work [83, 86, 87] build a cascaded system wherein the first stage uses an existing Collaborative filtering (CF) technique to predict the missing ratings and the second level consists of a modified ranking strategy (unlike conventional ranking in order of decreasing rating value) which promotes desired diversification. The second step requires selection of heuristic ranking threshold; items rated above the threshold are considered as prospective candidates for recommendation. Although, these methods have

the advantage/freedom of using an off the shelf CF technique, the two-stage process, in addition to increasing the computation burden, does not guarantee optimality of solution. On the other hand, unified models [88, 89] present a framework consisting of a weighted combination of diversity and accuracy promoting functions. These models, on account of being primarily based on a joint optimization strategy, provide some guarantee on the optimalilty of the solution. Despite the theoretically sound framework presented by these models, there has been limited exploration of such architectures; our proposed designs belong to this class of work.

In this work, we present a modified latent factor model to provide diverse, yet sufficiently accurate, recommendations. Our model is based on utilizing two divergent concepts, one promoting accuracy and other diversity, in a unified optimization framework. Amalgamation of diversity-promoting criteria, into the optimization framework for rating prediction itself, eliminates the need for arbitrary selection of ranking thresholds and heuristic ranking strategies otherwise employed in several existing approaches. The combined framework is used to predict the missing ratings. The predicted ratings, hence obtained, are sorted in decreasing order (similar to conventional RS model) thus eliminating the need for empirically derived ranking strategies. In addition to this, our designs yield not only higher individual diversity (a diverse set of recommendations to individual customers) but higher aggregate diversity (visibility of a higher percentage of total items in the repository) and novelty (recommendation of less popular and niche items) in recommendations as well. This is unlike existing models [82, 86] where, there is a prominent focus on only one of the diversity measures.

4.1 Review of Existing Models for Diversifying Recommendations

Motivated by studies [81] suggesting measures such as diversity and unexpectedness in recommendations as a factor in enhancing user's experience, several works have been proposed that are directed towards increasing diversity with an acceptable loss in accuracy.

The focus of these works is either on diversifying an individual user's recommendations or on recommending a greater percentage of total items available in the repository; the latter being system centric. The diversification from a user's perspective is evaluated in terms of Individual Diversity (ID), which is quantified as the pairwise dissimilarity between items in a (given) user's recommendation list. The system centric notion of diversity is captured by Aggregate Diversity (AD), measured as the number of unique items recommended across users.

Works such as [86, 90, 91] have focused on the business side alone and aimed to increase the aggregate diversity or suggest long tail items. In [86] authors have proposed several ranking strategies based on the variation in item ranking with parameters such as item's net rating and its popularity. They proposed a new ranking measure as given in (4.1)

$$rank(i, T_R) = \begin{cases} rank_x(i) & \text{if } \mathrm{R}(u,i) \in [T_R, T_{\max}] \\ \alpha + rank(i) & \text{otherwise} \end{cases} \quad \ldots\ldots\ldots (4.1)$$

where, T_R is the ranking threshold and $rank_x$ is the new ranking strategy applied to items ranked above a specific (chosen) ranking threshold; $rank$ denotes the usual ranking strategy (highest rated ranked first). The heuristic selection of parameters and strategies makes the scheme in [86] less effective than pure optimization based schemes. In [90] a new approach has been proposed of recommending users to items rather than the traditional approach of recommending items to users; it gives fair opportunity to all items. They have also designed a probabilistic approach to address the same. Authors in [91] have borrowed concepts from association mining to create characterization vectors for long tail items and establish their correlation with more heavily rated items. In [92], the problem of liquidating long tail stock has been addressed using relevance models. Authors in [93] have used evolutionary algorithm to improve the visibility of long tail items, however, such a model cannot place any guarantees on the optimality of solution. Also, it is difficult to vary the model parameters to introduce varying degree of diversity. The explicit focus of above works is only on suggesting a larger number of items which may not result in improved diversity for users, thereby affecting their interest in RS.

Other works [94, 95, 96] have addressed the problem from user's perspective and suggested schemes which improve individual diversity. Authors in [94] have proposed a topic diversification based method to increase individual diversity. They used hierarchical classification of items combined with item-based CF to generate diverse yet reasonably accurate suggestions. In [95] the idea of *k*-furthest neighbors has been introduced. They have showed that suggesting items disliked by furthest neighbors leads to higher diversity with a small reduction in accuracy. Both these methods are based on heuristic measures and neighborhood based techniques which are not as effective as latent factor models. Only a few works [88, 96] have focused on pure optimization based designs. In [96], a probabilistic framework has been proposed, which aims at generating a recommendation list such that the probability of an item from the list being selected by an user is maximized. However, they proposed a greedy scheme for solving their optimization framework. Also, their model is suited for binarized ratings alone and suggests only one item at a time to the user. A more cohesive approach has been presented in [88] for accuracy-diversity tradeoff. They have formulated a binary optimization problem combining two optimizing measures; one promoting diversity and another accuracy as in (4.2).

$$\max_{y} (1-\theta)\alpha y^T D y + \theta\beta m^T y \quad s.t \quad 1^T y = p \qquad \text{…….…… (4.2)}$$

where, y is an indicator vector having value 1 at the i^{th} position if item i is recommended, D is a matrix defining pair-wise distance (dissimilarity) between items and m is the vector defining a user's affinity to each item. They also used a greedy scheme for solving their formulation owing to quadratic nature of the problem.

In this work, we suggest a single stage, modified CF, a scheme which predicts missing ratings such that both diversity and relevance are adequately represented in the final recommendation list. Our models are shown to achieve significant improvement in individual diversity, aggregate diversity and novelty with minimal compromise on accuracy.

4.2 Research Contributions

In this chapter, we discuss our proposed models for achieving accuracy-diversity balance in the recommendations. Our models are based on the notion that ideally, to provide maximum diversity, predicted rating values across diverse items (characterized by distinct features) should be uniformly distributed [88]. This ensures an equal probability of selection of items from each of the distinct item subsets, thereby enhancing diversity. However, on its own, such random selection can lead to considerable reduction in prediction accuracy. To ensure that there is minimum accuracy drop for a given diversity level, we formulate an optimization problem combining the uniform distribution constraint with the principles of latent factor model based CF design methods.

Our main contributions in the area can be summarized as follows:

1. We build a matrix completion model which in addition to exploiting the available rating values, makes use of item metadata (in our case movie genres) as well. In this work, we build upon the LRMC framework (targeted towards high accuracy) and impose additional constraint which ensures greater diversity in recommendations. This add-on constraint is incorporated into the LRMC framework as a regularization term promoting uniform distribution amongst average ratings across genres. Uniform distribution allows all the (varied) genre to have an equal chance of representation in the recommendation list; thereby enhancing diversity. We also design an algorithm to support our model.
2. Our next model is built on the BCS framework. To introduce diversity in the recommendations, we exploit the understanding that if a user displays a uniform affinity for all relevant features (for example, similar affinity for all genre or all directors in the case of movies), then he/she will exhibit a similar penchant for all items. In effect, it will reflect in an equal chance for diverse items, across multiple genres/feature sets, to be recommended to such a user. We incorporate this concept into our framework by promoting the latent factor vector of each user to display minimum variance i.e. user displays similar liking for all the latent

factors. We design an algorithm based of MM technique to support our formulation.

Our designs provide improved diversity from both the user's as well as the system's perspective while maintaining due coherence with a user's past preference (i.e. sufficiently high accuracy). We are also able to substantially augment the visibility of long tail (less popular) items. This is credited to our optimization centric design, which ensures optimality of the solution, and thus, shows improvement over existing methods.

4.3 Matrix Completion Framework Balancing Accuracy and Diversity in Recommendations

In this work, we propose a (convex) optimization framework which establishes a balance between the two desired (yet conflicting) characteristics – accuracy and diversity – while predicting ratings. The convexity inherent in our proposed formulation ensures optimality of solution; thereby yielding the desired level of diversity with minimal loss of accuracy – a claim supported by our experimental results. Our model is capable of introducing diversity as per an explicitly chosen criteria - in our case, we use movie genres to explicitly enforce genre based diversity. To enable the same, we promote average ratings across distinct genres to display a uniform distribution so as to enable a fair chance of representation to all genres.

4.3.1 Proposed Formulation

Matrix completion model (4.3) for CF is built on the assumption, that the rating given by a user to an item is a function of a small number of latent factors and thus, the rating matrix is inherently low-rank.

$$\min_{Z} \left\| Y - A(Z) \right\|_F^2 + \lambda_n \left\| Z \right\|_* \qquad \text{…..…… (4.3)}$$

Equation (4.3) ensures recovery of the rating matrix (Z) from the given observation matrix (Y), with high accuracy. However, it places no emphasis on obtaining a diverse set of recommendations.

We modify the formulation in (4.3) to predict ratings such that the top-N items, based on the traditional ranking approach (highest rated being highest ranked), provide a high degree of diversity with minimal reduction in accuracy. To obtain maximally diversified recommendations, each diverse set of items must have an equal probability of being recommended. In this work, we focus on the movie database wherein the movie genre defines the diversification criteria i.e. recommendation list should ideally have an equal probability of representation/selection from each genre. The proposed model can be easily tuned to accommodate other diversifying features as well. When combined with the base model in (4.3), this diversifying criterion ensures that diversity is introduced in recommendations while maintaining adequate relevance to a user's past preference (latter is credited to the base - accuracy centric - model).

To accommodate the diversification criterion, we include an additional regularization term in (4.3) which promotes minimum variability amongst average rating for each genre by a given user as in (4.4)

$$\min_{Z} \left\| Y - A(Z) \right\|_F^2 + \lambda_n \left\| Z \right\|_* + \lambda_d \sum_{m=1}^{M} \text{variance}_{genre}(m) \qquad \ldots\ldots\ldots (4.4)$$

where, $\text{variance}_{genre}(m) = \sum_{g=1}^{|G|} \left(Z^A{}_{m,\text{g}} - mean_{m,genre} \right)^2$ gives the variance of average ratings (over a genre) across all genre for a given user m; $Z^A{}_{m,g}$ is the average rating by user m to movies belonging to genre g ($|G|$ being the total number of genres) and $mean_{m,genre} = \frac{1}{|G|} \sum_{g=1}^{|G|} Z^A{}_{m,g}$ is the mean of ratings across all genres; λ_d is the regularization parameter controlling the contribution of diversity-promoting term.

The nuclear norm constraint in (4.4) ensures a low-rank structure, yielding high correlation with previously recorded rating pattern of a user, while the diversifying criteria - uniform distribution term - in (4.4) promotes deviation away from a set pattern by spreading the ratings uniformly across genres. The values of λ_d and λ_n determine the relative contribution of prediction accuracy (promoted by nuclear norm constraint) and diversity (promoted by variance minimization constraint) on the recommendation list. As

the model is based on a convex optimization framework, optimality of solution i.e. minimum loss in accuracy for a given degree of diversity is ensured.

The variance based regularization term in (4.4) can be represented in matrix form as in (4.5) where M is the number of users; N is the number of items; μ_g is the number of movies belonging to the genre g and $\mathbf{1}_{|G|\times|G|}$ is a matrix of dimension $|G|\times|G|$ consisting of all 1's. Matrix S is defined such that $S_{i,j}=1$ *iff* movie i belongs to genre classification *j*.

$$\left\| \begin{bmatrix} Z_{1,1} & Z_{1,2} & . & Z_{1,N} \\ Z_{2,1} & Z_{2,2} & . & . \\ . & . & . & . \\ Z_{M,1} & . & . & Z_{M,N} \end{bmatrix} \begin{bmatrix} \frac{S_{1,1}}{\mu_1} & \frac{S_{1,2}}{\mu_2} & . & \frac{S_{1,|G|}}{\mu_{|G|}} \\ \frac{S_{2,1}}{\mu_1} & \frac{S_{2,2}}{\mu_2} & . & . \\ . & . & . & . \\ \frac{S_{N,1}}{\mu_1} & . & . & \frac{S_{N,|G|}}{\mu_{|G|}} \end{bmatrix} - \frac{1}{|G|} \begin{bmatrix} Z_{1,1} & Z_{1,2} & . & Z_{1,N} \\ Z_{2,1} & Z_{2,2} & . & . \\ . & . & . & . \\ Z_{M,1} & . & . & Z_{M,N} \end{bmatrix} \begin{bmatrix} \frac{S_{1,1}}{\mu_1} & \frac{S_{1,2}}{\mu_2} & . & \frac{S_{1,|G|}}{\mu_{|G|}} \\ \frac{S_{2,1}}{\mu_1} & \frac{S_{2,2}}{\mu_2} & . & . \\ . & . & . & . \\ \frac{S_{N,1}}{\mu_1} & . & . & \frac{S_{N,|G|}}{\mu_{|G|}} \end{bmatrix} \begin{bmatrix} \mathbf{1}_{|G|\times|G|} \end{bmatrix} \right\|_F^2 \quad \text{…..…… (4.5)}$$

Using (4.5) in (4.4), we can formulate our problem as follows

$$\min_Z \left\| Y - A(Z) \right\|_F^2 + \lambda_n \left\| Z \right\|_* + \lambda_d \left\| ZS_\mu - ZS_\mu \bar{\mathbf{1}}_{|G|\times|G|} \right\|_F^2 \quad \text{…..…… (4.6)}$$

where, $S_\mu(i,j) = S_{i,j} \big/ \mu_j$ and $\bar{\mathbf{1}}_{|G|\times|G|}$ is a matrix of dimension $|G|\times|G|$ with each element as $1 \big/ |G|$. Equation (4.6) can be written concisely as

$$\min_Z \left\| Y - A(Z) \right\|_F^2 + \lambda_n \left\| Z \right\|_* + \lambda_d \left\| ZF \right\|_F^2 \quad \text{..... (4.7)}$$

where, $F = \mathrm{S}_\mu\left(I_{|G|\times|G|} - \bar{\mathbf{1}}_{|G|\times|G|}\right)$; $I_{|G|\times|G|}$: Identity matrix

Equation (4.7) represents our proposed formulation – Matrix Completion Framework balancing Accuracy and Diversity (MC_AD). The resultant matrix *Z* obtained on solving (4.7) contains ratings recovered in such a manner that both prediction accuracy and diversity are jointly optimized.

As the predicted ratings are themselves an outcome of balancing diversity and accuracy, there is no need for random/heuristic ranking strategy to be adopted. The ratings are ranked in order of decreasing predicted value to generate top-N recommendation list for each user. The diversity promoting term in our formulation yields high individual diversity. As, for each user, accuracy (recommending preferred genre) is also considered along with diversity (reasonably similar chance of recommendation to each genre), our model can achieve high aggregate diversity as well. Consider for example a user who likes comedy, he/she will be recommended more comedy oriented movies but other genres like action and drama will also find representation. Similarly, if another user likes drama, more emphasis will be on movies belonging to this genre. In the overall scenario, across all users, many dissimilar movies spanning multiple genres are recommended thereby improving sales diversity. Our claim is supported by the results shown in the following sections.

We design an algorithm, to support our proposed framework; same is discussed in the next section.

4.3.2 Algorithm Design

In this section, we present the design of an algorithm, for our proposed framework, using split Bregman technique [46].

We introduce a proxy variable (X) in our formulation as in (4.8) where B is the Bregman variable and η is the regularization parameter.

$$\min_{Z,X} \|Y - A(Z)\|_F^2 + \lambda_n \|Z\|_* + \lambda_d \|XF\|_F^2 + \eta \|X - Z - B\|_F^2 \quad \text{…..…… (4.8)}$$

Next, as the two variables X and Z are separable, we split the problem in (4.8) into two simpler sub-problems, each minimizing over a single variable as follows.

Sub-problem 1

$$\min_{Z} \|Y - A(Z)\|_F^2 + \lambda_n \|Z\|_* + \eta \|X - Z - B\|_F^2 \quad \text{…..…… (4.9)}$$

Sub-problem 2

$$\min_{X} \lambda_d \|XF\|_F^2 + \eta \|X - Z - B\|_F^2 \quad \text{…..…… (4.10)}$$

Sub-problem 1 can be solved by soft thresholding of singular values [33] as in (4.11)

$$Z \leftarrow soft\left(singular\_value(T), \frac{\lambda_n}{2\alpha} \right)$$

$$\text{where,} \quad T = Z + \frac{1}{\alpha}\left(\begin{pmatrix} A \\ \sqrt{\eta} I \end{pmatrix}^T \left(\begin{pmatrix} Y \\ \sqrt{\eta}(X - B) \end{pmatrix} - \begin{pmatrix} A \\ \sqrt{\eta} I \end{pmatrix} Z \right) \right) \quad \text{…..…… (4.11)}$$

$$\alpha \geq \max\left(eigen\_value \begin{pmatrix} A \\ \sqrt{\eta} I \end{pmatrix}^T \begin{pmatrix} A \\ \sqrt{\eta} I \end{pmatrix} \right)$$

$$\text{and} \quad Soft(t,u) = sign(t)\ \max(0, |t| - u)$$

Sub-problem 2 can be written as a simple least square minimization problem (4.12) and solved using any gradient based solver.

$$\min_{X} \left\| \begin{pmatrix} \mathbf{0} \\ \sqrt{\eta}(Z + B) \end{pmatrix} - X \begin{pmatrix} \sqrt{\lambda_d} F \\ \sqrt{\eta} I \end{pmatrix} \right\|_F^2 ; \mathbf{0} \text{ is a matrix of all zeros} \quad \text{…..…… (4.12)}$$

The two sub-problems, are alternately solved with consecutive iteration of both interlaced with an update of Bregman variable as in (4.13)

$$B = B + Z - X \qquad \text{…..……} \quad (4.13)$$

The iterations continue until convergence, i.e. the maximum number of iterations reached or reduction in objective function value drops below the threshold. The complete algorithm is summarized in figure 4.1.

Input : Z_0, X_0 is randomly initialized, maximum iterations $m\_iter, \lambda_n, \lambda_d$

Output : Z

while $k \le m\_iter$ *or* $obj\_func(k) - obj\_func(k-1) \le 1e-7$

Solve for Z (original variable)

$$Z \leftarrow Z \leftarrow soft\left(singular\_value(T), \frac{\lambda_n}{2\alpha} \right)$$

where, $$T = Z + \frac{1}{\alpha}\left(\begin{pmatrix} A \\ \sqrt{\eta} I \end{pmatrix}^T \left(\begin{pmatrix} Y \\ \sqrt{\eta}(X - B) \end{pmatrix} - \begin{pmatrix} A \\ \sqrt{\eta} I \end{pmatrix} Z \right) \right)$$

$$\alpha \ge \max\left(eigen\_value \begin{pmatrix} A \\ \sqrt{\eta} I \end{pmatrix}^T \begin{pmatrix} A \\ \sqrt{\eta} I \end{pmatrix} \right)$$

Solve for X (proxy variable)

$$X \leftarrow \arg\min_X \left\| \begin{pmatrix} \mathbf{0} \\ \sqrt{\eta}(Z + B) \end{pmatrix} - X \begin{pmatrix} \sqrt{\lambda_d} F \\ \sqrt{\eta} I \end{pmatrix} \right\|_F^2$$

Update Bregman Variable

$$B \leftarrow X - Z - B$$

end while

Figure 4.1 Algorithm for Matrix Completion Framework balancing Accuracy and Diversity (MC_AD)

4.4 Blind Compressive Sensing Framework Balancing Accuracy and Diversity in Recommendations

In this work, we present a modified latent factor model to provide diverse, yet sufficiently accurate, recommendations.

The accuracy promoting term in our model is derived from the theory of latent factor model and cast in a BCS framework proposed in Chapter 2. To introduce diversity in the recommendation, we build upon the understanding that if a user displays a uniform affinity for all relevant features i.e. all latent factors (for example similar affinity for all genre or all directors in case of movies) then he/she will exhibit a similar penchant for all items. In effect, it will reflect in equal chance for diverse items, across multiple feature sets, to be recommended to such a user. We incorporate this concept into the BCS framework as a regularization term promoting the latent factor vector of each user to display minimum variance i.e. user displays a similar liking for all latent factors.

4.4.1 Proposed Formulation

Our proposed formulation is a modification of the blind compressive sensing formulation (4.14), proposed in Chapter 2.

$$\min_{U,V} \left\| Y - A(UV) \right\|_F^2 + \lambda_u \left\| U \right\|_F^2 + \lambda_v \left\| V \right\|_1 \qquad \text{…..……} \ (4.14)$$

Equation (4.14) warrants that the recovered latent factor matrices U and V are consistent with the given rating values. Thus, latent factor vectors are recovered such that a user $c's$ latent factor vector (u_c) shares a high degree of correlation with item $i's$ latent factor vector (v_i) if the available rating $y_{c,i}$ is high. Such a model ensures that the predicted rating is high for items which are decidedly similar to the ones liked by the user in the past. Thus, a traditional ranking approach (predicted ratings sorted in decreasing order), applied to the predicted ratings, generates a recommendation list characterized by high accuracy but very limited diversity.

In an attempt to provide high diversity while maintaining necessary relevance to users' preference we introduce additional constraint in the Latent factor Model (LFM) based design (4.14). A user's choice is characterized by its latent factor vector. If a user displays a similar level of interest in all the relevant features (latent factors), then elements of his/her latent factor vector will be akin to those pulled out from a uniform distribution. In effect, the user will exhibit (nearly) equal affinity to distinct item genre/categories. This constraint can be accommodated into our base formulation (4.14)

as an additional regularization term, as shown in (4.15), which minimizes the variance amongst all the elements of a user's latent factor vector.

$$\min_{U,V} \left\| Y - A(UV) \right\|_F^2 + \lambda_u \left\| U \right\|_F^2 + \lambda_v \left\| V \right\|_1 + \lambda_d \sum_{m=1}^{M} \text{variance}(u_m) \quad \ldots\ldots\ldots (4.15)$$

where, $\text{variance}(u_m)$ is the variance of the latent factor vector of the user m (u_m) and is given by $\text{variance}(u_m) = \sum_{l=1}^{F} \left(u_m(l) - mean_m \right)^2$ where, F is the number of latent factors considered, $u_m(l)$ is the l^{th} element of u_m; $mean_m = \frac{1}{F} \sum_{l=1}^{F} u_m(l)$ is the mean of latent factor vector of the user m; λ_d is the regularization parameter.

The use of (add-on) variance minimization constraint prevents the predicted rating scores for each user from being biased towards a particular item group; as a near uniform (user) latent factor vector shows (similar) high correlation with items characterized by diverse feature set. For example, in the case of movie recommendation, a user's predicted ratings will show uniform distribution across distinct features resulting in a recommendation list with the presence of multiple genres or different director or cast.

If considered as a standalone term, such a condition will ensure maximum diversity but very poor accuracy. In our proposed formulation, a balance between the (accuracy promoting) data consistency term $\left(\left\| Y - A(UV) \right\|_F^2 \right)$ and the (diversity promoting) variance minimization regularization factor $\left(\sum_{m=1}^{M} \text{variance}(u_m) \right)$ is established using the regularization parameter λ_d. The increase in importance given to variance minimization term (via the high value of λ_d) makes diversity go higher at the cost of accuracy. The results shown in subsequent sections validate the same.

We illustrate our model and the underlying principle with a small (toy) example shown in figure 4.2. Let us consider a LFM based framework for movie recommendation where each movie/user is represented by a latent factor vector of length four (corresponding to four distinct genres: thriller, drama, rom-com, and animation). Given three users, one

	Thri.	Drama	Rom-Com	Anim.
U1	0.9	0.3	0.1	0.1
U2	0.1	0.2	0.9	0.05
U3	0.3	0.3	0.9	0.01

a. User vector for standard LFM

	Thri.	Drama	Rom-Com	Anim.
U1*	0.6	0.5	0.2	0.2
U2*	0.2	0.3	0.6	0.1
U3*	0.5	0.5	0.6	0.1

c. User vector for proposed model

	Thri.	Drama	Rom-Com	Anim.
M1	0.9	0	0	0.1
M2	0.8	0	0.3	0
M3	0	0	0.7	0
M4	0	0.5	0.7	0
M5	0.6	0.4	0	0
M6	0	0.5	0.5	0

b. Item vector for standard LFM

	U1	U2	U3	U1*	U2*	U3*
M1	**0.82**	0.10	0.27	**0.56**	0.19	0.46
M2	**0.75**	0.35	0.51	0.54	0.34	**0.58**
M3	0.07	**0.63**	**0.63**	0.14	0.42	0.42
M4	0.22	**0.73**	**0.78**	0.39	**0.57**	**0.67**
M5	0.66	0.14	0.30	**0.56**	0.24	0.50
M6	0.20	0.55	0.60	0.35	**0.45**	0.55

d. Predicted ratings for two cases

Un: Latent factor vectors (standard model); Un*: Latent factor vectors (proposed model)

Figure 4.2 Example to Illustrate the Working of Proposed Model Balancing Accuracy and Diversity

showing a higher preference for thriller other two for romance, figure 4.2 (a) illustrates their probable latent factor vectors respectively, derived using standard LFM based approach (which is purely focused on accuracy). Latent factor vectors for six movies of the diverse genre are given in figure 4.2 (b).

The impact of our formulation on the (recovered) user's latent factor vector is shown in figure 4.2 (c). It shows that compared to the latent factor representation in figure 4.2 (a), user's affinities derived using our model have a flatter distribution. Figure 4.2 (d) shows the interaction (correlation) between the users and items for both the standard formulation and our proposed design; the top-two recommended movies (highest correlation) in each case highlighted in bold. It can be observed that for the standard model, the user gets

recommended very similar movies (like all thrillers for user 1). Our proposed design helps generate recommendations which not only gives due attention to a user's preferred genre but also imbibes diversity. For example, user 1 gets recommended a thriller as well as a movie high on drama. Similarly, user 3 is recommended a romantic thriller alongside a movie belonging to only romantic genre. Also, our model is able to recommend overall five movies compared to four suggested by standard model improving aggregate diversity as well. The above toy example validates our claim that our design criteria is able to maintain alignment with a user's past preference (high accuracy) while promoting variety in recommendation (high diversity).

Coming back to our proposed model (4.15), the variance minimization regularization term can be recast in the matrix form as follows

$$\left\| \begin{bmatrix} u_1(1) & u_1(2) & . & u_1(F) \\ u_2(1) & u_2(2) & . & . \\ . & . & . & . \\ u_M(1) & . & . & u_M(F) \end{bmatrix} - \frac{1}{F} \begin{bmatrix} u_1(1) & u_1(2) & . & u_1(F) \\ u_2(1) & u_2(2) & . & . \\ . & . & . & . \\ u_M(1) & . & . & u_M(F) \end{bmatrix} \bar{\mathbf{1}}_{F\times F} \right\|_F^2$$

…..…… (4.16)

where, M is the number of users; $\bar{\mathbf{1}}_{F\times F}$ is a $F\times F$ matrix of all 1's.

Using (4.16) in (4.15), we can represent our proposed formulation as in (4.17) where, $D = \mathbf{I}_{F\times F} - \frac{1}{F}\bar{\mathbf{1}}_{F\times F}$ and $\mathbf{I}_{F\times F}$ is an identity matrix of dimension $F\times F$.

$$\min_{U,V} \left\| Y - A(UV) \right\|_F^2 + \lambda_u \left\| U \right\|_F^2 + \lambda_v \left\| V \right\|_1 + \lambda_d \left\| UD \right\|_F^2 \qquad \text{…..…… (4.17)}$$

Equation (4.17) defines our proposed model – Blind Compressive Sensing Framework balancing Accuracy and Diversity (BCS_AD) - which optimally combines the two opposing yet preferred features of a good recommender system, namely accuracy and diversity. The predicted rating values are high for items across distinct item groups, thus enabling a conventional ranking scheme to offer improved diversity to customers.

Our model does not require any additional information or pre-processing of ratings, like several existing methods [83, 86]. Also, we can achieve significant improvement across

multiple diversity measures: individual diversity, aggregate diversity, and novelty. As the latent factor vector of each user is promoted to display uniform distribution, each user is offered diverse items in its recommendation list improving individual diversity. The presence of accuracy promoting constraints necessitates that, along with diversity, user's choice is also given due prominence. Every user has greater affinity towards few items/features, like some may enjoy comedy more while others prefer drama; some may prefer a particular director over others and so on. The recommendation list of each user gives necessary credit to its inclination while also suggesting certain variety. Thus, considering recommendation lists across all users, a wide variety of distinct items gain ample visibility; improving aggregate diversity. Experimental results certify that our model is capable of improving novelty as well by reducing the bias towards highly rated or popular items.

4.4.2 Algorithm Design

In this section, we discuss the design of our algorithm based on Majorization-minimization (MM) technique [40].

Using MM technique, (4.17) can be recast as follows

$$\min_{U,V} \left\|W-(UV)\right\|_F^2 + \lambda_u \left\|U\right\|_F^2 + \lambda_v \left\|V\right\|_1 + \lambda_d \left\|UD\right\|_F^2$$

$$\text{where,} \quad \mathrm{W} = UV + \frac{1}{\alpha} A^T \left(Y - A(UV)\right) \qquad \ldots\ldots\ldots (4.18)$$

$$\alpha \geq \max\left(eigen\_value\left(A^T A\right)\right)$$

We split (4.18) into simpler sub-problems, each minimizing over a single variable.

Sub-problem 1

$$\min_U \left\|W-UV\right\|_F^2 + \lambda_u \left\|U\right\|_F^2 + \lambda_d \left\|UD\right\|_F^2 \qquad \ldots\ldots\ldots (4.19)$$

Sub-problem 2

$$\min_V \left\|W-UV\right\|_F^2 + \lambda_v \left\|V\right\|_1 \qquad \ldots\ldots\ldots (4.20)$$

Sub-problem 1 can be cast as a least square problem (4.21) and thus has a closed form solution.

$$\min_{U} \left\| \begin{pmatrix} W \\ \mathbf{0} \\ \mathbf{0} \end{pmatrix} - U \begin{pmatrix} V \\ \sqrt{\lambda_u} I \\ \sqrt{\lambda_d} D \end{pmatrix} \right\|_F^2 ; \mathbf{0} \text{ is a matrix of all zeros} \quad \text{…..……} \quad (4.21)$$

Sub-problem 2 can be solved by iterative soft thresholding [44] as follows

$$V \leftarrow Soft\left(V + {}^{1}\!/\!_{\beta}\left(U^T \left(B - UV \right) \right), \frac{\lambda_v}{2\beta} \right)$$

$$\text{where, } \beta = \max\left(eigen\_value\left(U^T U \right) \right) \quad \text{…..……} \quad (4.22)$$

$$\text{and } \quad Soft\left(t, u \right) = sign\left(t \right) \ \max\left(0, |t| - u \right)$$

Both the sub-problems are solved alternately until convergence. Complete algorithm for our formulation is given in figure 4.3.

Input : U_0, V_0 are randomly initialized, maximum iterations $m\_iter, \lambda_u, \lambda_v, \lambda_d$

Output : U, V

while $k \le m\_iter$ *or* $obj\_func(k) - obj\_func(k-1) \le 1e-7$

$$W = \left(U_{k-1} V_{k-1} \right) + \frac{1}{\alpha} A^T \left(Y - A\left(U_{k-1} V_{k-1} \right) \right)$$

Solve for U

$$U_k \leftarrow \arg\min_{U} \left\| \begin{pmatrix} W \\ \mathbf{0} \\ \mathbf{0} \end{pmatrix} - U \begin{pmatrix} V_{k-1} \\ \sqrt{\lambda_u} I \\ \sqrt{\lambda_d} D \end{pmatrix} \right\|_F^2$$

Solve for V

$$V_k \leftarrow Soft\left(V + {}^{1}\!/\!_{\beta}\left(U_k{}^T \left(B - U_k V \right) \right), \frac{\lambda_v}{2\beta} \right)$$

where, $\beta = \max\left(eigen\_value\left(U_k{}^T U_k \right) \right)$

end while

Figure 4.3 Algorithm for Blind Compressive Sensing Framework balancing Accuracy and Diversity (BCS_AD)

4.5 Experiment and Evaluation

In this section, we discuss the performance of our proposed diversity promoting frameworks and compare the results with those obtained using existing state of the art methods in terms of relevant evaluation measures.

4.5.1 Description of Dataset and Evaluation Setup

We evaluate the performance of various recommendation algorithms on the two benchmark Movielens datasets – the 100K and the 1M. We perform experiments on a subset of the two datasets, selecting only those users who have rated more than 100 movies. The improved density of available ratings helps us evaluate the algorithms better by providing sufficient amount of test data, which better captures accuracy-diversity balance. The statistics of the abridged datasets is shown in Table 1.

Table 4.1 Details of the Movielens Datasets

Dataset	# Users	# Items	# Ratings
Movielens 100K	364	1682	74522
Movielens 1M	2945	3670	847302

We conduct fivefold cross-validation on both the datasets with 80% data used for training and 20% reserved for testing. The simulations are carried out on a system with i7-3770S CPU @3.10GHz with 8GB RAM.

For the MC_AD framework, Item metadata (genre information) is used. There are 18 different genres into which movies are clubbed – Action, Adventure, Animation, Children's, Comedy, Crime, Documentary, Drama, Fantasy, Film-Noir, Horror, Musical, Mystery, Romance, Sci-Fi, Thriller, War and Western.

4.5.2 Evaluation Metrics

We evaluate each algorithm in terms of the trade-off between accuracy and diversity centric evaluation parameters.

Prediction accuracy is assessed in terms of two commonly adopted measures - precision and recall. Both the measures together capture the efficiency of a RS in suggesting relevant top-N recommendations to the user.

$$Precision = \frac{\#t_p}{\#t_p + \# f_p} \quad \text{…..……} \quad (4.23)$$

$$Recall = \frac{\#t_p}{\#t_p + \# f_n} \quad \text{…..……} \quad (4.24)$$

where, t_p denotes true positive i.e. items that are relevant to the user and are recommended by RS; f_p denotes false positives i.e. items that are irrelevant but are recommended to the user; f_n defines false negatives i.e. relevant items that are not recommended to the user.

The diversity in recommendations is evaluated using three criteria: Individual Diversity (ID) [87], Aggregate Diversity (AD) [86] and novelty (NV) [85].

Individual diversity measures diversity from an individual's (user's) perspective; greater is the dissimilarity between items in a user's recommendation list; higher is the ID. It is defined as in (4.25), wherein the diversity offered to all users is averaged to get a net measure of ID.

$$Individual\ Diversity = \frac{1}{M} \sum_{u \in Users} \frac{\sum_{i \in RL(u)} \sum_{j \in RL(u)} \left(1 - sim(i,j)\right)}{N_R \left(N_R - 1\right)} \quad \text{…..……} \quad (4.25)$$

where, M is the total number of users; $RL(u)$ is the recommendation list of user u; N_R is the length of recommendation list; $sim(i,j)$ is the similarity between item i and j both belonging to $RL(u)$. We measure similarity using cosine distance. A high value of ID indicates a more diverse recommendation list.

Aggregate diversity (AD) (4.26) measures diversity purely from the system's standpoint. It is measured as the total number of unique items which are recommended across users and helps quantify the net visibility of the database. A high AD ensures greater visibility for multiple items, raising their chances of being bought by the customers.

$$Aggregate\ Diversity = \left| \bigcup_{u \in Users} \left(RL(u) \right) \right| \quad \text{..... (4.26)}$$

where $|x|$: *cardinality of* x

Novelty offered by the system is computed as in (4.27).

$$Novelty = \frac{1}{M} \sum_{u \in Users} \frac{\sum_{i \in RL(u)} \log_2 \left(M / \#i \right)}{|RL(u)|} \quad \text{..... (4.27)}$$

where, in addition to previously defined terms, $\#i$ denotes the number of ratings for item *i* in the training data. Equation (4.27) is a measure of the newness of recommended items; the highly rated or visible items contributing to lower novelty. A higher value of novelty indicates the recommendation of long tail items, which increases profit margins for the business. Also, it serves the primary purpose of RS of suggesting those items to users which they would not have (probably) found out on their own.

4.5.3 Results and Discussion

In this section, we discuss the performance and behavior of our proposed models as well as show their comparison with existing techniques for achieving diversity-accuracy balance.

4.5.3.1 Diversity-Accuracy Trade-off as a function of Regularization Parameters for Proposed Models

Both our formulations – MC_AD and BCS_AD, attempt to predict the ratings such that desired diversity level is achieved with minimum loss in accuracy. The relevance given to accuracy in comparison to diversity is controlled by the relative magnitude of the two corresponding constraints in the formulation, i.e. the relative value of λ_d w.r.t that of λ_n (in the case of MC_AD) or λ_u (for BCS_AD).

In this section, we show the behavior of our models in terms of accuracy and diversity metrics as a function of these regularization parameters.

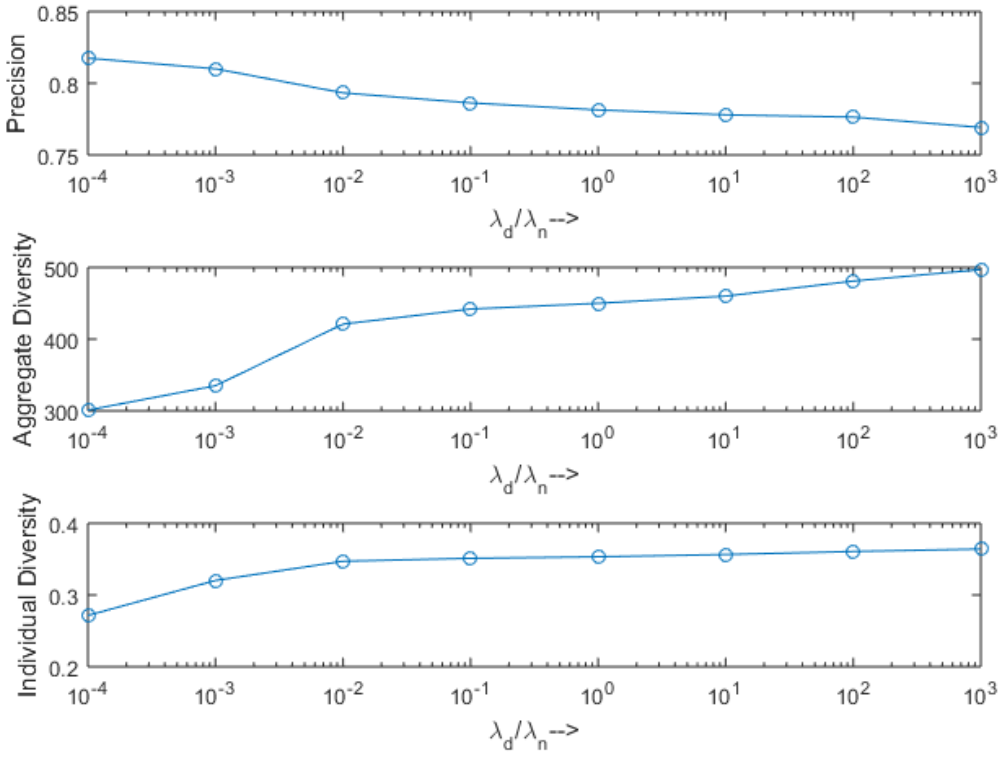


Figure 4.4 Variation in Evaluation Metrics with Regularization Parameter for MC_AD

Figure 4.4 shows the accuracy-diversity trade-off achieved by varying the relative importance given to accuracy and diversity promoting constraints in our formulations – MC_AD for the 100K Movielens dataset. The trade-off between the two is a function of the relative values of the two regularization parameters λ_n and λ_d . We show the change in precision, aggregate diversity, and individual diversity as a function of the ratio of two regularization parameters i.e. λ_d / λ_n .

Similar results for BCS_AD (4.17) model are shown in figure 4.5. Similar to the trend observed for MC_AD, here also it can be observed that with an increase in the ratio λ_d / λ_u , accuracy measure (namely precision) drops while the diversity measures (namely individual diversity, aggregate diversity, and novelty) improves.

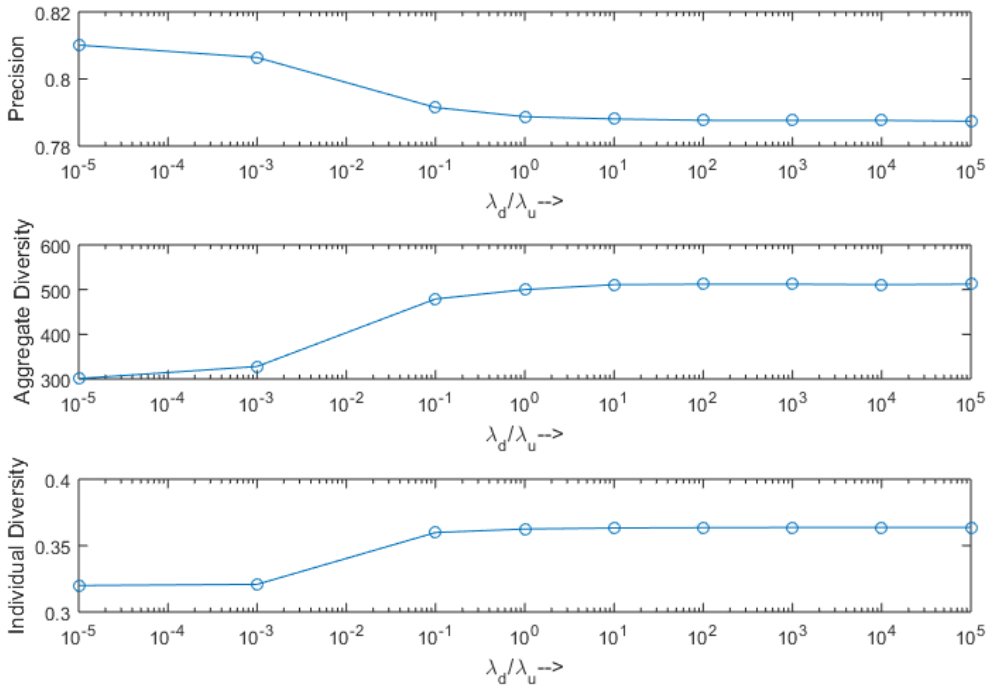


Figure 4.5 Variation in Evaluation Metrics with Regularization Parameter for BCS_AD

The behavior shown in the two figures is on expected lines as an increase in the relative value of λ_d places greater emphasis on diversity compared to accuracy which results in a rise in former. The precision drops till a certain point and then nearly saturates (diversity also rises and then saturates). This is because the data consistency term $\left(\left\|Y - M\left(UV\right)\right\|_F^2\right)$ in the formulation, which ensures consistency with observed values, prevents any further deviation from user's preference. By tuning the ratio λ_d/λ_u or λ_d/λ_n, the trade-off between diversity and accuracy of the model can be varied as per requirement.

A similar trend is observed for 1M dataset as well; however, it is not shown as it does not provide any additional information.

4.5.3.2 Comparison with Existing Techniques

In this section, we compare the performance of our models with the existing works aimed at diversifying recommendations.

We compare our model against the following schemes:

1. Reverse Predicted Rating Value (RPRV) [86]: This is a two-stage model whose first stage uses an existing CF method to predict missing ratings. We use either BCS (for comparison with BCS_AD) or MC (for comparison with MC_AD) for this stage. Next, the predicted values rated over a ranking threshold (T_{rank}) are ranked in ascending order, i.e. all items rated over T_{rank} are arranged such that lowest rated items are ranked higher in the order. This is opposite to the conventional scheme wherein items are ranked in descending order of (predicted) rating values. The selected value of T_{rank} dictates the amount of accuracy sacrificed to attain diversity. A high value of T_{rank} leads to higher accuracy and lower diversity. However, to ensure sufficient alignment with a user's past preference the minimum value of T_{rank} is set as 4; same is followed in [86].
2. Item Average Rating (IA) [86]: Similar to RPRV, it is also a two-stage model where in BCS or MC model constitutes the first stage. However, unlike RPRV, items rated above the ranking threshold are arranged such that items which have a higher average rating in the train set are ranked lower. The strategy is based on the argument that items with lower average rating are less popular and hence recommending those increases diversity. The rest of strategy and selection of ranking threshold is similar to RPRV.
3. Hierarchical Clustering (HC) [83]: It is also a two-stage design (using BCS or MC as the first stage) but the ranking is done using hierarchical clustering. They cluster items such that selection of items from different clusters helps improve recommendation diversity.

For the above-listed algorithms, codes provided by the authors are used.

Table 4.2 Comparison of Existing Algorithms with MC_AD for 100K Movielens Dataset

Precision Loss (% reduction)	MC_AD (% increase)			IA (% increase)			RPRV (% increase)			HC (% increase)		
	Aggregate Diversity	Individual Diversity	Novelty	Aggregate Diversity	Individual Diversity	Novelty	Aggregate Diversity	Individual Diversity	Novelty	Aggregate Diversity	Individual Diversity	Novelty
3	49.3	8.7	18.7	44.9	6.9	10.5	38.3	6.8	9.7	1.8	0.9	1.2
3.5	54.4	11.5	23.8	49.9	7.8	13.0	42.8	7.4	11.6	1.8	1.1	1.3
4	68.2	14.1	29.6	54.3	8.7	14.8	45.6	8.0	13.3	1.8	1.3	1.6
4.5	73.7	15.1	31.9	57.8	9.4	15.7	48.5	8.5	15.0	1.7	1.3	1.6
5	75.9	15.4	33.0	61.3	10.1	16.5	52.8	9.3	15.7	1.5	1.5	1.8
5.5	79.2	16.4	35.3	64.7	10.8	17.3	56.8	10.0	16.3	1.5	1.6	1.9
6	82.9	16.8	36.8	68.6	11.7	18.4	59.8	10.5	16.6	1.5	1.8	2.2
6.5	85.6	17.1	37.0	73.2	12.8	20.0	62.7	10.9	17.0	1.5	2.0	2.4
Base MC	Precision: 0.8148, AD: 274, ID: 0.3152, Novelty: 1.3011											

Table 4.3 Comparison of Existing Algorithms with MC_AD for 1M Movielens Dataset

Precision Loss (% reduction)	MC_AD (% increase)			IA (% increase)			RPRV (% increase)			HC (% increase)		
	Aggregate Diversity	Individual Diversity	Novelty	Aggregate Diversity	Individual Diversity	Novelty	Aggregate Diversity	Individual Diversity	Novelty	Aggregate Diversity	Individual Diversity	Novelty
1	117.1	14.9	27.6	33.8	4.4	7.7	46.8	5.4	9.5	2.0	0.1	0.4
1.5	136.8	16.1	29.7	42.8	5.6	9.6	56.1	6.4	11.4	2.6	0.3	0.8
2	157.2	17.7	33.6	52.7	6.8	11.9	65.3	7.6	13.7	3.1	0.6	1.3
2.5	174.2	19.2	37.8	60.6	7.9	13.9	74.0	9.0	16.2	3.4	0.8	1.8
3	187.1	20.5	41.3	66.5	8.8	15.5	82.8	10.3	18.4	3.2	1.1	2.3
3.5	198.4	21.7	43.1	73.6	9.9	17.6	90.4	11.2	20.2	3.4	1.3	2.7
Base MC	Precision: 0.8716, AD: 653, ID: 0.3459, Novelty: 1.5584											

Table 4.4 Comparison of Existing Algorithms with BCS_AD for 100K Movielens Dataset

Precision Loss (% reduction)	BCS_AD (% increase)			IA (% increase)			RPRV (% increase)			HC (% increase)		
	Aggregate Diversity	Individual Diversity	Novelty	Aggregate Diversity	Individual Diversity	Novelty	Aggregate Diversity	Individual Diversity	Novelty	Aggregate Diversity	Individual Diversity	Novelty
1	4.2	1.5	4.3	1.7	0.1	1.3	0.9	0.7	1.5	1.2	0.3	0.0
1.5	13.4	3.2	8.7	9.1	1.7	3.5	6.5	2.1	3.9	0.9	0.5	0.2
2	19.4	4.9	11.7	14.2	2.8	5.5	16.3	4.2	7.3	0.7	0.7	0.5
2.5	23.3	6.1	13.8	22.2	4.3	8.5	24.6	6.5	9.6	0.6	0.8	0.8
3	30.5	8.7	18.6	28.9	5.3	9.8	27.5	7.4	10.3	0.5	0.9	1.1
3.5	36.7	10.7	22.5	33.3	6.0	10.6	29.5	8.0	10.9	0.4	1.1	1.4
4	39.6	11.3	24.6	35.0	6.7	11.4	31.5	8.7	11.6	0.2	1.2	1.6
4.5	43.0	12.1	26.2	36.7	7.3	12.1	33.5	9.3	12.2	0.1	1.3	1.7
5	44.5	12.7	27.3	38.4	8.0	12.8	36.1	9.9	13.2	0.1	1.5	1.9
Base BCS	Precision: 0.8141, AD: 339, ID: 0.3196, Novelty: 1.3340											

Table 4.5 Comparison of Existing Algorithms with BCS_AD for 1M Movielens Dataset

Precision Loss (% reduction)	BCS_AD (% increase)			IA (% increase)			RPRV (% increase)			HC (% increase)		
	Aggregate Diversity	Individual Diversity	Novelty	Aggregate Diversity	Individual Diversity	Novelty	Aggregate Diversity	Individual Diversity	Novelty	Aggregate Diversity	Individual Diversity	Novelty
1	32.9	2.8	9.0	17.2	3.5	6.5	17.1	4.7	7.8	0.09	0.09	0.2
1.2	44.3	4.6	11.3	20.0	3.9	7.3	20.4	5.2	8.7	0.07	0.1	0.4
1.4	52.8	6.7	13.9	22.9	4.4	8.1	23.4	5.7	9.7	0.04	0.3	0.7
1.6	63.4	8.4	17.0	25.7	5.0	9.1	25.7	6.2	10.6	-.03	0.4	1.0
1.8	68.8	9.0	18.3	28.3	5.5	10.0	27.3	6.7	11.4	-.07	0.6	1.5
2	76.6	10.1	20.6	30.6	5.9	10.8	28.9	7.3	12.3	-0.1	0.8	2.0
Base BCS	Precision: 0.8808, AD: 820, ID: 0.3515, Novelty: 1.6046											

Table 4.2 and 4.3 shows the comparison of MC_ AD with other frameworks for 100K and 1M datasets respectively. The results shown are for a recommendation list of length 5; however similar results are obtained for a list of length 10 and 20 as well. The values show in the Table illustrate the percentage increase in novelty (NV), aggregate (AD) and individual diversity (ID) as a function of the reduction in precision, over the base – MC (4.3) - method.

Similar results for BCS_AD are shown in Table 4.4 and 4.5 where the base model is BCS-CF formulation (4.14).

The following observations summarize the results given in Tables 4.2-4.5

1. Amongst all the existing methods, RPRV and IA perform comparably, but HC gives very poor results. HC shows a drop in AD with a drop in precision in some cases. This is explainable as HC clusters items into diverse clusters and suggest one from each which can invariably be similar for several users, thereby reducing AD.

2. Our model achieves significantly higher diversity for a given drop in precision as compared to other designs. For example, referring to the last row of Table 4.4, for a 5% drop in precision w.r.t. BCS model, BCS_AD provides 44% increase in AD compared to around 37% increase offered by RPRV and IA. Similarly, we achieve a 28% increase in novelty whereas RPRV gives 13% and IA only 10% improvement. A similar trend is seen throughout the other Tables as well.

   This is credited to our design built on an optimization centric theoretically sound framework which helps give optimal performance. Also, accuracy and diversity are jointly optimized in a one stage process. Other works use an ad-hoc two-stage method, one stage focusing on accuracy, another on diversity. Combining the two stages by employing heuristic measures and criteria cannot yield optimal results in a practical scenario.

3. The superiority of our proposed scheme is even more evident for the 1M dataset (Table 6). Our model delivers almost twice the jump in diversity and novelty than IA or RPRV for the same accuracy loss.

4. Amongst the MC based design (MC_AD) and the BCS formulation (BCS_AD), the latter performs much better. There is a significant gain in diversity with a very small drop in accuracy for the MC_AD design as compared to BCS_AD. This can be credited to the convexity inherent in the MC based formulation and its capability to effectively model the hidden structure of rating matrix. Further, as the genre based diversity is explicitly enforced in the MC design, the same can generate higher ID (which is measured in terms of the genre only).

Next, we have a closer look at the results obtained with our proposed schemes

4.5.3.3 Behaviour of Proposed Schemes

In this section, we further analyze the behavior of our proposed frameworks and their influence of diversification of recommendations.

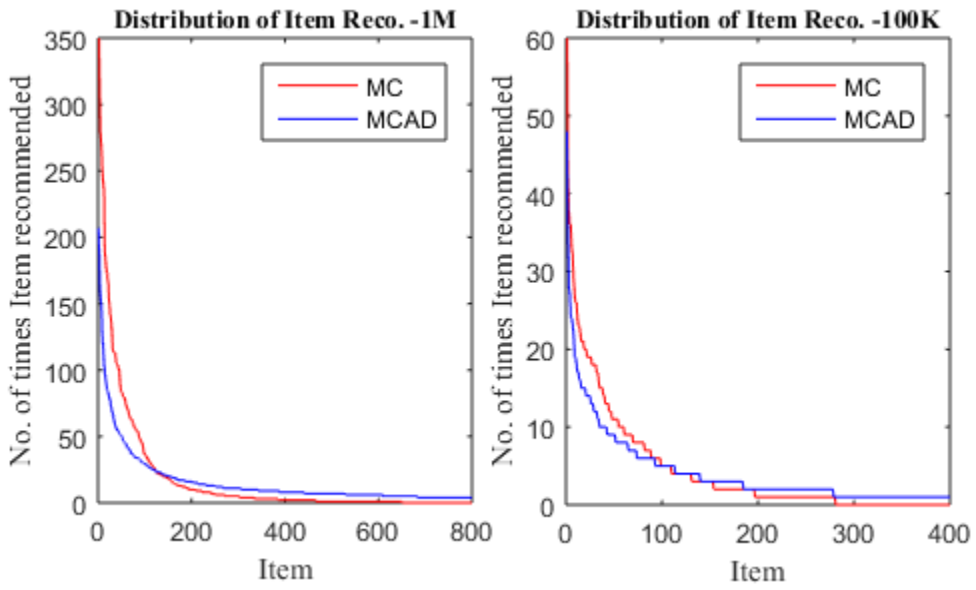


Figure 4.6 Distribution of Top-T Recommended Items

Most recommender system algorithms are biased towards heavily rated or popular items thereby recommending them across multiple users. This leads to build up of the long tail items in the repository.

Figure 4.6 shows the impact of our model MC_AD model in flattening the distribution of the number of times an item is recommended for 100K and 1M datasets. We show the number of recommendations of top-T recommended items (T is 400 for the 100K dataset and 800 for 1M dataset). It can be observed that our algorithm can flatten the distribution; reducing the maximum number of recommendations for any item as well as increasing the number of items recommended. Similar results are obtained for BCS_AD model as well.

The results shown in this section satisfactorily establishes the superiority of our proposed schemes in establishing an accuracy-diversity balance and achieving the desired diversification with much lower much reduction in accuracy as compared to existing formulations.

4.6 Summary

In this chapter, we discussed our proposed frameworks built to generate adequately accurate yet diverse and novel set of recommendations so as to enhance user's overall experience and keep them engaged on the portal. Unlike most of the existing techniques which either adopt heuristic two-stage modeling or greedy optimization schemes, we presented an optimization strategy which jointly optimizes accuracy and diversity, during the process of rating prediction itself and in effect provides improved results over existing models promoting diversity. Also, our designs were shown to substantially increase diversity from both the users (individual diversity) as well as system's perspective (aggregate diversity and novelty).

Chapter 5
COLLABORATIVE FILTERING WITH SUPERVISED AUTOENCODER

Matrix factorization frameworks for latent factor models have been the popular choice for RS design over the past decade. However, a seminal work [99] showed the possibility of using another representation learning / latent factor approach for collaborative filtering; it was the Restricted Boltzmann Machine (RBM). RBM based collaborative filtering has garnered very little interest since the publication of [99], as RBM based designs could not beat the advanced matrix factorization based techniques in terms of accuracy.

Over the last few years, researchers have started exploring the possibility of using another powerful representation learning technique for collaborative filtering – stacked autoencoder. Both stacked autoencoders and deep belief network (built from layers of RBM) are used to train deep neural networks. However, the lack of adequate rating information, afflicting the performance of latent factor models in general, poses a bigger challenge when viewing RS design from a machine learning perspective – latter relying on an abundance of data for effective performance. All of the existing autoencoder (AE) based RS design frameworks are minor variations of the standard AE and thus suffer from the curse of data sparsity.

In this work, we modify the basic autoencoder structure by incorporating (associated) item and user metadata into the learning framework in the form of a label consistency penalty - we propose a supervised version of autoencoder. In general, it has been found that better results are attained with supervised (discriminative) learning tools compared to unsupervised ones. The use of secondary information along with the rating data helps in mitigating the problem of data sparsity and improve the prediction accuracy.

5.1 Review of Autoencoder based Design

An autoencoder [100] is a self-supervised neural network, i.e. input and output are the same. It is unsupervised in the sense that the training does not require any class information.

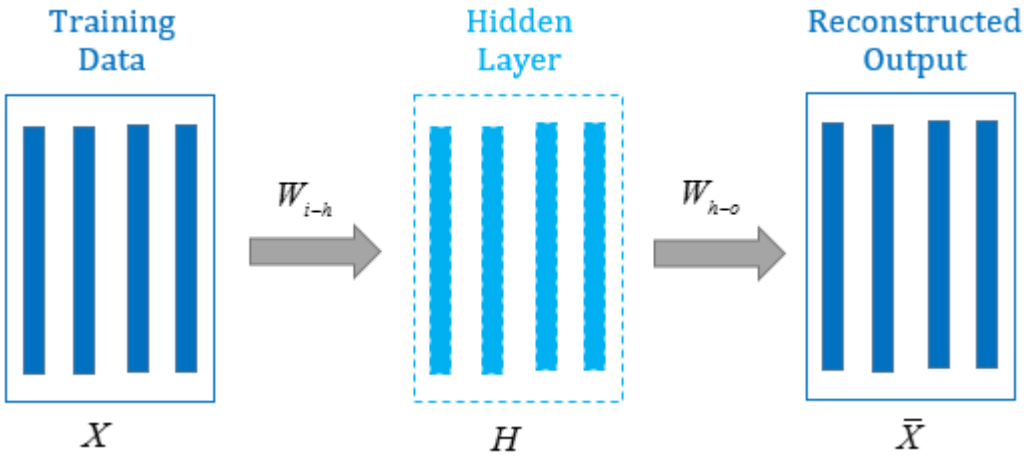


Figure 5.1 Standard Autoencoder

A standard AE (shown in figure 5.1) consists of two parts – the encoder maps the input $X = [x_1 | x_2 | ... | x_N]$ to a latent space $H = [h_1 | h_2 | ... | h_N]$, and the decoder maps the latent representation to the reconstruction of the (input) data $\bar{X} = [\bar{x}_1 | \bar{x}_2 | ... | \bar{x}_N]$ as in (5.1) and (5.2). Since the data space is assumed to be the space of real numbers, there is no sigmoidal function in the decoder section (5.2).

$$Encoder: \quad h_n = \phi(W_{i-h} x_n) \qquad \text{..... (5.1)}$$

$$Decoder: \quad \bar{x}_n = W_{h-o} h_n \qquad \text{..... (5.2)}$$

In (5.1) and (5.2), x_n is the n^{th} training sample (N being the total number of samples); h_n is the hidden layer representation of x_n and $\bar{x}_n$ is the reconstruction of the input; subscripts *i*, *h*, and *o* stand for input, hidden and output layers; W_{i-h} are the link weights from all the input nodes to the corresponding latent node and W_{h-o} are the weights for the

decoder end. The activation function $\phi(\bullet)$ is usually a non-linear function like sigmoid or tanh.

During training, the problem is to learn the encoding and decoding weights $-W_{i-h}$ and W_{h-o}. This is achieved by minimizing the error between the input and its reconstruction in the least square sense.

$$\min_{W_{i-h},W_{h-o}} \left\| X - W_{h-o}\left(\phi\left(W_{i-h}X\right)\right)\right\|_F^2 \qquad \text{…..……} \quad (5.3)$$

The problem in (5.3) is clearly non-convex. However, it is solved easily by gradient descent techniques since the activation function is smooth and continuously differentiable.

There have been several works that proposed modifications to the basic AE module. In [101], motivated by studies on neural activity [102], a l_1 penalty was imposed on the hidden layer representation. Stacked denoising autoencoders (SDAE) [103] are a variant of the basic autoencoder where the input consists of noisy samples and the output consists of clean samples; this is a stochastic regularization technique. Here the encoder and decoder are learned to denoise noisy input samples. The learned features appear to be more robust when learned by SDAE compared to standard stacked AE. In a recent work a marginalized denoising autoencoder has been proposed [104]; it does not learn a representation but learns the mapping from the input to the output. However, such an autoencoder cannot be used for representation learning and associated problems but can be used for domain adaptation.

Most of the studies in autoencoder based collaborative filtering are minor variations of each other. The basic autoencoder formulation has been used directly for RS design in [105, 106, 107]. In [108] baseline prediction has been used along with the ratings in the autoencoder framework; the baseline values were simply appended with the available ratings so that the autoencoder learns to reconstruct both the ratings and the baseline values. A combination of marginalized denoising autoencoder and probabilistic matrix factorization has been used in [109] for rating prediction.

5.2 Supervised Autoencoder for Recommender System Design

In this work, we present a novel supervised autoencoder (SupervisedAE) for rating prediction. We augment the standard l_2-norm loss function (5.3) with additional regularization constraints derived from the available class information. The class information is extracted from the available user or item metadata, i.e. users or items are grouped together based on the available metadata and assigned class labels.

Before discussing our proposed formulation, we briefly discuss the relationship between representation learning tools (specifically autoencoders) and the conventionally employed matrix factorization framework for collaborative filtering. Consider the rating matrix, with users along the rows and items along the columns; each item becomes a training sample and the corresponding column in the rating matrix is the input (raw data) to the autoencoder module. The encoder portion of the AE does not carry any significance in collaborative filtering (CF) framework; it is the decoder portion that captures the understanding of latent factor models. The connections between the representation (hidden layer) and the output (same as input) act as users' latent factor vectors and the representations at the hidden layer act as the item latent factors vectors. As the rating matrix is incomplete, the input is partially sampled. Thus, during the training phase, the connections corresponding to the missing inputs are not updated. This understanding has been exploited so far for existing collaborative filtering [106, 107] works using autoencoder.

Our design also follows similar interpretation; however, we modify the basic AE structure to enforce the computed feature vectors (latent factor representation) to be consistent with the corresponding class labels via a linear mapping. The use of additional information assists in improving the robustness and prediction accuracy of our design.

5.2.1 Proposed Formulation

Our proposed design for a supervisedAE is shown in figure 5.2. Our formulation is motivated by prior studies in discriminative [110] dictionary learning and discriminative restricted Boltzmann machine [111]. In these studies, a linear map is learned from the

representation layer to the class labels. Such a discriminative penalty has not been imposed on autoencoder learning before.

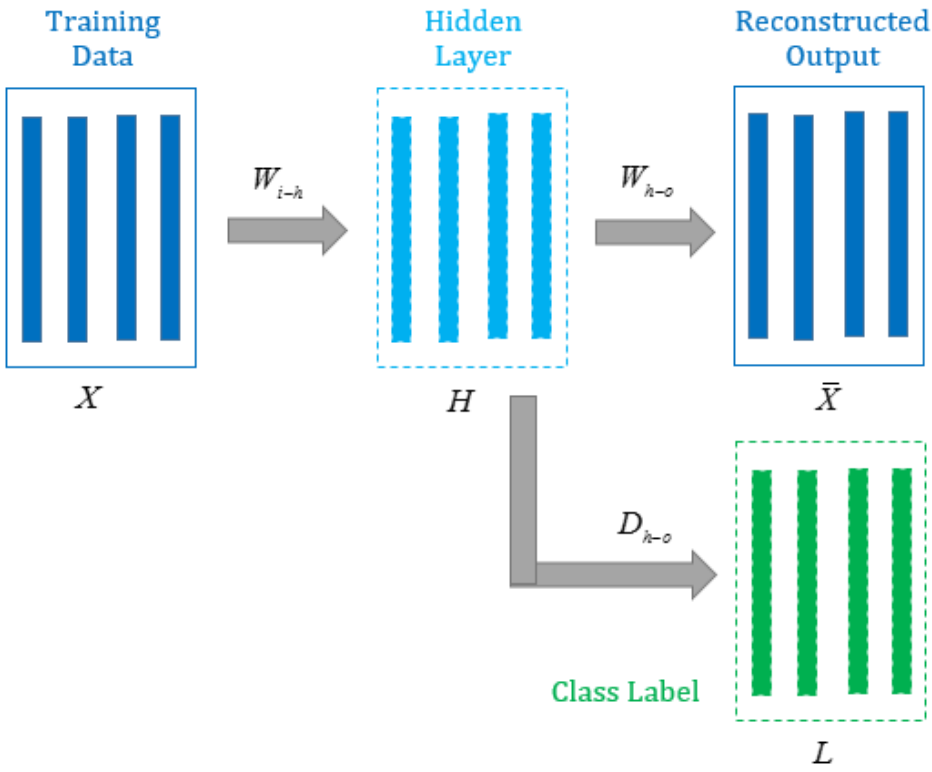


Figure 5.2 Design of Proposed Supervised Autoencoder

Our SupervisedAE module can incorporate either the user or item metadata depending on whether the users are considered as training sample instances or the items. We discuss the model using items as the samples; however, the same is applicable for users as well.

In this work, we are specifically interested in movie recommendations, so we take 'genre' as the auxiliary information. This information is used to assign class labels to each item, which is the used as an add-on source of information for the AE module. We generate a binary vector from the genre; the vector contains a 1 if the movie belongs to that genre or 0 otherwise. The vector is shown in Figure 5.3. Say a movie like Shawshank Redemption is tagged as 'crime' and 'drama' in IMDB. The corresponding feature vector is $[0,0,0,0,0,1,0,1,0,0,0,0,0,0,0,0,0,0]^T$; It has 1's corresponding to crime and drama and 0 everywhere else.

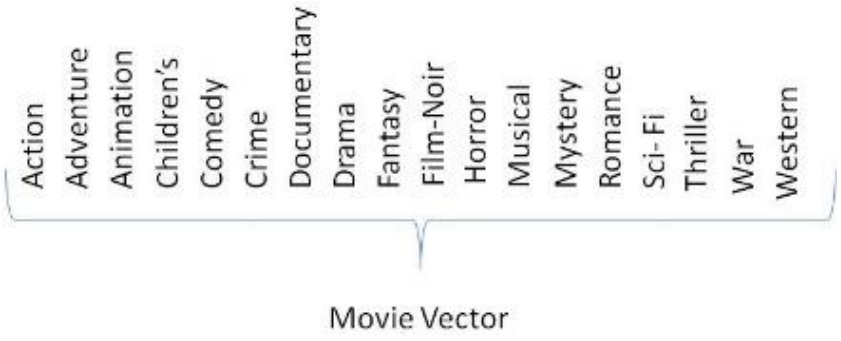


Figure 5.3 Item Genre-Label Vector

Our supervised autoencoder, on top of the regular autoencoder architecture, learns a linear map from the representation to binary targets or labels defined above. This makes the autoencoder supervised; it requires auxiliary information (class labels for the classification problem) for learning. Our design thus integrates information from multiple sources via a supervised framework.

The additional label consistency constraint is modeled as a regularization term (penalty) appended to the base formulation (5.3) as follows

$$\min_{W_{i-h},W_{h-o},D} \left\| X - W_{h-o}\left(\phi\left(W_{i-h}X\right)\right)\right\|_F^2 + \lambda \left\| L - D\left(\phi\left(W_{i-h}X\right)\right)\right\|_F^2 \quad \text{…..…… (5.4)}$$

Here $L = \left[l_1 \,|\, l_2 \,|\, ... \,|\, l_N\right]$ is the class label matrix and l_n is the class label vector of n^{th} item and D is a linear operator mapping the features to class labels vectors (also learned during the optimization process); $X = \left[\bar{r}_1 \,|\, \bar{r}_2 \,|\, ... \,|\, \bar{r}_N\right]$ where, $\bar{r}_n$ is the vector denoting the rating given by each user to the n^{th} item. This formulation is similar in concept to the label consistent models proposed in Chapter 3.

Further, to improve the robustness of design, we use concepts from denoising autoencoder and provide a noisy input to our autoencoder module. Considering, $\tilde{X}$ as the noisy version of X , our final formulation can be written as

$$\min_{W_{i-h},W_{h-o},D} \left\| X - W_{h-o}\left(\phi\left(W_{i-h}\tilde{X}\right)\right)\right\|_F^2 + \lambda \left\| L - D\left(\phi\left(W_{i-h}\tilde{X}\right)\right)\right\|_F^2 \quad \text{…..…… (5.5)}$$

In (5.5), both the information sources – the rating data as the item metadata (captured as class labels) influence the recovery of latent factor representation of users and items. Thus, our design is better placed compared to standard AE module in terms of handling the problem of data scarcity and prediction accuracy.

In this work, we have exploited item metadata. However, we can alternately encode the user metadata. In the latter case, each user (characterized by its rating vector) forms the training sample. In this supervised AE formulation, it is possible to encode either the user or the item metadata, but not both simultaneously.

In the following section, we discuss the algorithm design to support our SupervisedAE framework.

5.2.2 Algorithm Design

Our objective is to solve (5.5). We substitute, the latent representation as $Z = \phi\left(W_{i-h}\tilde{X}\right)$; this recasts (5.5) as follows,

$$\min_{W_{i-h},W_{h-o},D,Z} \left\|X - W_{h-o}Z\right\|_F^2 + \lambda\left\|L - DZ\right\|_F^2 \quad \text{such that } Z = \phi\left(W_{i-h}\tilde{X}\right) \qquad \text{…..…… (5.6)}$$

We can formulate the Lagrangian from (5.6); however, the Lagrangian will impose strict equality between the variable and the corresponding proxy in every iteration; this is not required in practice. We only want the two to be equal at convergence. Therefore, instead of the Lagrangian we form the augmented Lagrangian.

$$\min_{W_{i-h},W_{h-o},D,Z} \left\|X - W_{h-o}Z\right\|_F^2 + \lambda\left\|L - DZ\right\|_F^2 + \mu\left\|Z - \phi\left(W_{i-h}\tilde{X}\right)\right\|_F^2 \qquad \text{…..…… (5.7)}$$

For small values of μ, the equality constraint is relaxed and for large values, it is enforced. One heuristic way to solve the problem would be to start with a small value of μ, solve (5.7); increase the value of μ, solve (5.7) again and keep repeating. However, this is not an elegant solution. Also, one needs to rely on intuition for increasing the value of μ. A better approach is to introduce a Bregman variable between the proxy and the original variable [46]. The Bregman variable can be automatically updated, keeping the

value of μ fixed. The update of the Bregman variable would ensure that the proxy and the variable are equal during convergence.

This leads to our final formulation (5.8), where B is the Bregman variable.

$$\min_{W_{i-h},W_{h-o},D,Z} \|X - W_{h-o}Z\|_F^2 + \lambda\|L - DZ\|_F^2 + \mu\left\|Z - \phi\left(W_{i-h}\tilde{X}\right) - B\right\|_F^2 \quad \ldots\ldots\ldots (5.8)$$

We can segregate (5.8) into the following sub-problems.

Sub-problem 1

$$\min_{W_{h-o}} \|X - W_{h-o}Z\|_F^2 \quad \ldots\ldots\ldots (5.9)$$

Sub-problem 2

$$\min_{D} \|L - DZ\|_F^2 \quad \ldots\ldots\ldots (5.10)$$

Sub-problem 3*

$$\min_{W_{i-h}} \left\|Z - \phi\left(W_{i-h}\tilde{X}\right) - B\right\|_F^2 \equiv \left\|\phi^{-1}\left(Z - B\right) - W_{i-h}\tilde{X}\right\|_F^2 \quad \ldots\ldots\ldots (5.11)$$

Sub-problem 4

$$\min_{Z} \|X - W_{h-o}Z\|_F^2 + \lambda\|L - DZ\|_F^2 + \mu\left\|Z - \phi\left(W_{i-h}\tilde{X}\right) - B\right\|_F^2 \quad \ldots\ldots\ldots (5.12)$$

Sub-problems 1-3 are simple least square minimizations having closed form solution. Sub-problem 4 is also a least square minimization problem; it becomes apparent after re-arranging, as follows

$$\arg\min_{Z} \left\| \begin{pmatrix} X \\ \sqrt{\lambda}L \\ \sqrt{\mu}\left(\phi\left(W_{i-h}\tilde{X}\right) + B\right) \end{pmatrix} - \begin{pmatrix} W_{h-o} \\ \sqrt{\lambda}D \\ \sqrt{\mu}I \end{pmatrix} Z \right\|_F^2 \quad \ldots\ldots\ldots (5.13)$$

Subproblem 3 follows a recent work:
Gulcehre, C., Moczulski, M., Denil, M., & Bengio, Y. (2016). Noisy activation functions. *arXiv preprint arXiv:1603.00391*.

The final step in each iteration is the update the relaxation variable (by gradient descent).

$$B \leftarrow Z - \phi\left(W_{i-h}\tilde{X}\right) - B \qquad \text{…..……} \ (5.14)$$

The iterations continue till convergence, i.e. either a specified maximum number of iterations or until the difference between the objective function falls below a chosen threshold in successive iterations. The complete algorithm is given in figure 5.4.

Input : Z_0, X_0 is randomly initialized, maximum iterations m_iter, λ

Output : W_{i-h}, W_{h-o}

while $k \le m\_iter$ *or* $obj\_func(k) - obj\_func(k-1) \le 1e-7$

Solve for decoder weight W_{h-o}

$$W_{h-o} \leftarrow \min_{W_{h-o}} \left\| X - W_{h-o}Z \right\|_F^2$$

Solve for encoder weight W_{i-h}

$$W_{i-h} \leftarrow \arg\min_{W_{i-h}} \left\| \phi^{-1}\left(Z - B\right) - W_{i-h}\tilde{X} \right\|_F^2$$

Solve for Proxy Z

$$Z \leftarrow \arg\min_{Z} \left\| \begin{pmatrix} X \\ \sqrt{\lambda}L \\ \sqrt{\mu}\left(\phi\left(W_{i-h}\tilde{X}\right) + B\right) \end{pmatrix} - \begin{pmatrix} W_{h-o} \\ \sqrt{\lambda}D \\ \sqrt{\mu}I \end{pmatrix} Z \right\|_F^2$$

Solve for Linear Map

$$D \leftarrow \min_{D} \left\| L - DZ \right\|_F^2$$

Update Bregman Variable

$$B \leftarrow Z - \phi\left(W_{i-h}\tilde{X}\right) - B$$

end while

Figure 5.4 Algorithm for Supervised Autoencoder

5.3 Experiment and Evaluation

In this section, we discuss the performance of our proposed AE frameworks with conventional latent factor models.

5.3.1 Description of Dataset and Evaluation Setup

We evaluate the performance of various recommendation algorithms on the two benchmark Movielens datasets – 100K and 1M.

We conduct fivefold cross-validation on both the datasets. The simulations are carried out on a system with i7-3770S CPU @3.10GHz with 8GB RAM. The labels are derived using item metadata, as discussed above; there are 18 item genres. Further, the noisy version of the input is formed by randomly setting 30% of the available ratings to zero.

The regularization parameter values are found using the *l*-curve technique [51].

5.3.2 Evaluation Metrics

We compare the performance of our proposed AE` design with existing models using accuracy based measures.

Ranking based metrics - Precision (5.15) and recall (5.16)

$$Precision = \frac{\#t_p}{\#t_p + \#f_p} \quad \text{..... (5.15)}$$

$$Recall = \frac{\#t_p}{\#t_p + \#f_n} \quad \text{..... (5.16)}$$

where t_p denotes true positive (item relevant and recommended), f_p is false positive (item irrelevant and recommended) and f_n is false negative (item relevant and not recommended).

Rating based metrics - MAE (5.17) and RMSE (5.18).

$$MAE = \frac{\sum_{m,n \in \Omega} \left| r_{m,n} - \bar{r}_{m,n} \right|}{|\Omega|} \quad \text{..... (5.17)}$$

$$RMSE = \sqrt{\frac{\sum_{m,n \in \Omega} \left(r_{m,n} - \bar{r}_{m,n} \right)^2}{|\Omega|}} \quad \text{..... (5.18)}$$

In the above equations, $r_{m,n}$ is the actual rating by the user m on movie n and $\overline{r}_{m,n}$ is the corresponding predicted rating; Ω is the set of indices of available ratings and $|\Omega|$ is the cardinality of the rating dataset i.e. number of available ratings.

5.3.3 Results and Discussion

We compare our SupervisedAE module against the following techniques

1. Standard Latent Factor Models (LFM) using only ratings – BCD-NMF (Block Coordinate Descent Nonnegative Matrix Factorization), PMF (Probabilistic Matrix Factorization)
2. Proposed models using only ratings – eNet_BCS (Elastic net regularised Blind Compressive Sensing) and MC_SB (Matrix Completion using Split Bregman)
3. Proposed model using metadata – Label consistent Blind Compressive Sensing Framework (LC_BCS), Label consistent Matrix Completion Framework (LC_MC)
4. Existing techniques using Metadata – Nonnegative Matrix Factorization using Graph Regularization (GR_M), Semi-Supervised nonnegative Matrix Factorization (SSNMF)

The details of the above models have been discussed in previous chapters.

Table 5.1 gives the comparison of all methods on the rating centric measures for both 100K and 1M dataset. The performance of various algorithms on the ranking based metrics is given in Table 5.2 (for 100K dataset) and Table 5.3 (for 1M dataset)

Table 5.1 Rating based Evaluation Metrics for Movielens Datasets

Algorithm	100K Dataset		1M Dataset	
	MAE	RMSE	MAE	RMSE
Standard LFM using only Ratings				
PMF	0.7564	0.9639	0.7240	0.9127
BCD_NMF	0.7582	0.9816	0.6953	0.8890
Proposed Models using only ratings				
BCS_CF	0.7356	0.9409	0.6917	0.8789
eNet_BCS	0.7273	0.9255	0.6899	0.8655
MC_SB	0.7351	0.9319	0.6813	0.8711
Proposed Models using Metadata				
LC_BCS	0.7199	0.9146	0.6709	0.8567
LC_MC	0.7193	0.9145	0.6731	0.8559
Existing Models using Metadata				
GR_M	0.7577	0.9616	0.7233	0.9139
SSNMF	0.7723	1.0112	0.7285	0.9401
Proposed Autoencoder based Design				
SupervisedAE	0.7163	0.9141	0.6774	0.8640

Table 5.2 Ranking based Evaluation Metrics for 100K Movielens Dataset

Algo.	Precision					Recall				
	@10	@20	@30	@40	@50	@10	@20	@30	@40	@50
Standard LFM using only Ratings										
PMF	50.52	37.74	30.06	24.71	21.09	63.56	76.76	82.43	85.06	86.56
BCD_NMF	51.33	37.22	29.2	23.88	20.04	64.13	76.77	82.04	84.7	86.10
Proposed Models using only ratings										
BCS_CF	51.33	38.05	30.14	24.82	21.31	64.16	77.57	82.89	85.47	86.86
eNet_BCS	52.57	38.79	30.69	25.20	21.31	65.22	78.62	84.08	86.64	88.07
MC_SB	51.42	38.41	30.47	25.12	21.21	64.43	78.41	83.86	86.53	87.97
Proposed Models using Metadata										
LC_BCS	52.48	38.91	30.82	25.38	21.47	65.14	78.75	84.15	86.77	88.18
LC_MC	52.55	38.99	31.01	25.42	21.51	65.17	78.77	84.19	86.80	88.21
Existing Models using Metadata										
GR_M	50.96	38.39	30.52	25.27	21.45	64.51	77.82	83.41	86.32	87.67
SSNMF	51.6	38.24	30.46	25.06	21.21	64.79	79.23	83.8	86.49	87.96
Proposed Autoencoder based Design										
Supervis-edAE	52.92	38.24	30.81	25.11	21.40	64.98	78.51	84.02	86.34	87.88

Table 5.3 Ranking based Evaluation Metrics for 1M Movielens Dataset

Algo.	Precision					Recall				
	@10	@20	@30	@40	@50	@10	@20	@30	@40	@50
Standard LFM using only Ratings										
PMF	63.06	50.45	41.9	35.7	30.79	60.32	78.41	86.65	91.17	93.82
BCD_NMF	66.95	52.64	42.57	35.82	30.79	62.51	79.18	86.72	90.69	93.03
Proposed Models using only ratings										
BCS_CF	67.19	52.36	42.94	36.16	31.17	62.53	79.91	87.57	91.47	93.82
eNet_BCS	68.64	53.64	43.83	36.91	31.81	63.61	80.61	88.06	92.02	94.39
MC_SB	67.64	52.79	43.33	36.54	31.52	63.33	80.20	87.93	91.96	94.39
Proposed Models using Metadata										
LC_BCS	68.22	53.17	43.59	36.72	35.64	63.67	80.47	88.15	92.15	94.51
LC_MC	68.23	53.21	43.62	36.77	35.71	63.68	80.52	88.19	92.16	94.53
Existing Models using Metadata										
GR_M	65.99	52.03	42.82	36.26	31.38	62.43	76.69	87.54	91.69	94.23
SSNMF	66.23	51.91	42.74	36.15	31.25	62.73	79.63	87.48	91.63	94.12
Proposed Autoencoder based Design										
Supervis-edAE	67.99	53.18	43.13	36.57	35.12	63.52	80.41	87.84	91.78	94.01

Following observations can be made from the values reported in the tables above

1. It can be seen from the given results that our supervised autoencoder design performs better than existing techniques, including those that are designed to use metadata.
2. Our AE module, which utilizes only a single type of metadata (movie genre in our case), performs almost at par with our proposed label consistent designs – LC_MC and LC_BCS. The latter makes use of both user and item metadata; however, the non-linearity inherent in the autoencoder module is better able to capture the underlying structure of the rating matrix. Thus, even with the use of only the item metadata, along with the ratings, our AE module achieves almost similar recovery accuracy as the label consistent MF and MC designs.
3. The recovery accuracy is slightly poorer than designs using user and item metadata for the 1M dataset because it is sparser than the 100K dataset.

Further, we also compare our design against existing autoencoder based RS design techniques. We use the results given in the paper directly for comparison. The stacked (standard) autoencoder based design proposed in [106] for RS design yields an MAE of 0.728 and RMSE of 0.933 for the 100K dataset; corresponding values for 1M dataset are 0.684 and 0.89. The error values in the above work, are ~2% higher than that obtained using our proposed model validating the effectiveness of including label information in the formulation.

Thus, our SupervisedAE module performs comparably to standard latent factor formulations; further exploration and improvement in the design can produce better results.

5.4 Summary

In this chapter, we discussed the problem of RS design from the perspective of a machine learning task. We proposed a supervised Autoencoder module which can jointly use the rating information along with the available item or user metadata. It performs better than existing AE based works for CF which uses only the rating information. Our design also performs comparably to standard MF or MC based designs using user/item metadata along with rating information.

Chapter 6
CONCLUSION

In this work, we focused on developing recommendations strategies to address several pertinent issues, relevant to RS design, and improve upon the existing state of the art.

The salient features of our proposed RS design formulations are listed below

1. Our designs are built on the principles of latent factor models for collaborative filtering; LFMs are the current de-facto approach for RS design. We used explicit rating information as it is more reliable than implicit data.
2. We cast the problem of RS design in an optimization framework; use of theoretically sound framework ensures superior performance compared to heuristic formulations. Further, we developed algorithms, using sophisticated optimization techniques, to ensure that our designs have lower run time than the existing approaches.
3. To mitigate the problem of rating data sparsity, we used readily available metadata, namely user demographics and item categories. This information is invariably available to most online portals, making our designs widely applicable.

The main contributions of the thesis are summarized below

1. In the first part of the thesis, we proposed modifications to the conventional MF framework, a popular approach to model the concepts of LFMs. We supplement our design with an effective algorithm which enables lower computation time for our model compared to existing formulations. We also developed an algorithm for the matrix completion framework, a convex counterpart of MF formulation, which is shown to yield improved recovery accuracy within reasonable run time, compared to standard LRMC algorithms.
2. Next, we focused on alleviating data sparsity by making use of readily available user and item metadata. We designed several models which make use of user and

item metadata, in addition to rating information, in either the MF or MC framework. Our designs provide a comprehensive framework that targets improvement in prediction accuracy in both warm and cold start scenario.

3. Motivated by studies suggesting a need for diversity in recommendations, along with accuracy, as a means to enhance customer experience, we suggested models that establish the desired accuracy-diversity balance. Our models are built on a joint optimization framework that attempts to balance the two divergent measures. Use of mathematically sound models eliminates the need for heuristic measures adopted by several existing works. Thus, our models achieve a higher diversity value for the same precision drop, than existing techniques.
4. Lastly, we suggested another means for representation learning – the supervised autoencoder (SupevusedAE). Conventional machine learning architectures find limited utility in RS designs as their performance relies heavily on the amount of data available. In this work, we designed a supervised version of the standard autoencoder which can use data from multiple sources – rating information and item metadata. The joint model can effectively mitigate the problem of data sparsity and show results comparable to conventional latent factor frameworks using secondary information.

Some of the future research directions are highlighted below.

1. Develop an online update strategy for our proposed formulations: RS database is being continuously updated with new users and new items. Re-evaluating the entire model is computationally intensive, thereby hampering frequent updates. We propose to develop online-update strategies, which can aid in effectively updating the existing models, in much lesser time than the complete model re-evaluation.
2. Design an AE module which can jointly harness user and item metadata: Such a design would be able to improve the prediction accuracy further. We also propose to explore the application of other machine learning architectures, including deep learning, to the domain of RS design.

REFERENCES

[1] Cosley, D., Lam, S. K., Albert, I., Konstan, J. A., & Riedl, J. (2003, April). Is seeing believing?: how recommender system interfaces affect users' opinions. In Proceedings of the SIGCHI conference on Human factors in computing systems (pp. 585-592). ACM.

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

[3] Bobadilla, J., Ortega, F., Hernando, A., & Gutiérrez, A. (2013). Recommender systems survey. Knowledge-Based Systems, 46, 109-132.

[4] Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., & Riedl, J. (1994, October). GroupLens: an open architecture for collaborative filtering of netnews. In Proceedings of the 1994 ACM conference on Computer supported cooperative work (pp. 175-186). ACM.

[5] Shardanand, U., & Maes, P. (1995, May). Social information filtering: algorithms for automating “word of mouth”. In Proceedings of the SIGCHI conference on Human factors in computing systems (pp. 210-217). ACM Press/Addison-Wesley Publishing Co.

[6] Bennett, J., & Lanning, S. (2007, August). The netflix prize. In Proceedings of KDD cup and workshop (Vol. 2007, p. 35).

[7] Liu, Q., Ge, Y., Li, Z., Chen, E., & Xiong, H. (2011, December). Personalized travel package recommendation. In 2011 IEEE 11th International Conference on Data Mining (pp. 407-416). IEEE.

[8] Brozovsky, L., & Petricek, V. (2007). Recommender system for online dating service. arXiv preprint cs/0703042.

[9] Hornick, M., & Tamayo, P. (2012). Extending recommender systems for disjoint user/item sets: The conference recommendation problem. IEEE Transactions on Knowledge and Data Engineering, 24(8), 1478-1490.

[10] Mooney, R. J., & Roy, L. (2000, June). Content-based book recommending using learning for text categorization. In Proceedings of the fifth

ACM conference on Digital libraries (pp. 195-204). ACM.

[11] Pazzani, M., & Billsus, D. (1997). Learning and revising user profiles: The identification of interesting web sites. Machine learning, 27(3), 313-331.

[12] Su, X., & Khoshgoftaar, T. M. (2009). A survey of collaborative filtering techniques. Advances in artificial intelligence, 2009, 4.

[13] Ekstrand, M. D., Riedl, J. T., & Konstan, J. A. (2011). Collaborative filtering recommender systems. Foundations and Trends in Human-Computer Interaction, 4(2), 81-173.

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

[15] Desrosiers, C., & Karypis, G. (2011). A comprehensive survey of neighborhood-based recommendation methods. In Recommender systems handbook (pp. 107-144). Springer US.

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

[17] Melville, P., Mooney, R. J., & Nagarajan, R. (2002, July). Content-boosted collaborative filtering for improved recommendations. In Aaai/iaai (pp. 187-192).

[18] Basilico, J., & Hofmann, T. (2004, July). Unifying collaborative and content-based filtering. In Proceedings of the twenty-first international conference on Machine learning (p. 9). ACM.

[19] Breese, J. S., Heckerman, D., & Kadie, C. (1998, July). Empirical analysis of predictive algorithms for collaborative filtering. In Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence (pp. 43-52). Morgan Kaufmann Publishers Inc.

[20] Wang, J., De Vries, A. P., & Reinders, M. J. (2006, August). Unifying user-based and item-based collaborative filtering approaches by similarity fusion. In Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval (pp. 501-508). ACM.

[21] Koren, Y., & Bell, R. (2011). Advances in collaborative filtering. In

Recommender systems handbook (pp. 145-186). Springer US.

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

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

[24] Kim, Y. D., & Choi, S. (2009, April). Weighted nonnegative matrix factorization. In 2009 IEEE International Conference on Acoustics, Speech and Signal Processing (pp. 1541-1544). IEEE.

[25] Lee, D. D., & Seung, H. S. (1999). Learning the parts of objects by non-negative matrix factorization. Nature, 401(6755), 788-791.

[26] Luo, X., Zhou, M., Xia, Y., & Zhu, Q. (2014). An efficient non-negative matrix-factorization-based approach to collaborative filtering for recommender systems. IEEE Transactions on Industrial Informatics, 10(2), 1273-1284.

[27] Gemulla, R., Nijkamp, E., Haas, P. J., & Sismanis, Y. (2011, August). Large-scale matrix factorization with distributed stochastic gradient descent. In Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 69-77). ACM.

[28] Salakhutdinov, R., & Mnih, A. (2011, September). Probabilistic matrix factorization. In NIPS (Vol. 20, pp. 1-8).

[29] Shan, H., & Banerjee, A. (2010, December). Generalized probabilistic matrix factorizations for collaborative filtering. In 2010 IEEE International Conference on Data Mining (pp. 1025-1030). IEEE.

[30] Jaggi, M., & Sulovsk, M. (2010). A simple algorithm for nuclear norm regularized problems. In Proceedings of the 27th international conference on machine learning (ICML-10) (pp. 471-478).

[31] Shamir, O., & Shalev-Shwartz, S. (2011). Collaborative Filtering with the Trace Norm: Learning, Bounding, and Transducing. In COLT (pp. 661-678).

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

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

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

[35] Meka, R., Jain, P., Caramanis, C., & Dhillon, I. S. (2008, July). Rank minimization via online learning. In Proceedings of the 25th International Conference on Machine learning (pp. 656-663). ACM.

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

[37] Mohan, K., & Fazel, M. (2012). Iterative reweighted algorithms for matrix rank minimization. *Journal of Machine Learning Research*, *13*(Nov), 3441-3473.

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

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

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

[41] Wang, H., Nie, F., Huang, H., Risacher, S., Ding, C., Saykin, A. J., & Shen, L. (2011, November). Sparse multi-task regression and feature selection to identify brain imaging predictors for memory performance. In 2011 International Conference on Computer Vision (pp. 557-562). IEEE.

[42] Zhang, K., Gray, J. W., & Parvin, B. (2010). Sparse multitask regression for identifying common mechanism of response to therapeutic targets. Bioinformatics, 26(12), i97-i105.

[43] Paige, C. C., & Saunders, M. A. (1982). LSQR: An algorithm for sparse linear equations and sparse least squares. ACM transactions on mathematical software, 8(1), 43-71.

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

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

[46] Goldstein, T., & Osher, S. (2009). The split Bregman method for L1-regularized problems. SIAM journal on imaging sciences, 2(2), 323-343.

[47] Boyd, S., Parikh, N., Chu, E., Peleato, B., & Eckstein, J. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. Foundations and Trends® in Machine Learning, 3(1), 1-122.

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

[49] Harper, F. M., & Konstan, J. A. (2016). The movielens datasets: History and context. ACM Transactions on Interactive Intelligent Systems (TiiS), 5(4), 19.

[50] Shani, G., & Gunawardana, A. (2011). Evaluating recommendation systems. In Recommender systems handbook (pp. 257-297). Springer US.

[51] Lawson, C. L., & Hanson, R. J. (1995). Solving least squares problems (Vol. 15). Philadelphia: Siam.

[52] Lingala, S. G., & Jacob, M. (2013). Blind compressive sensing dynamic MRI. IEEE transactions on medical imaging, 32(6), 1132-1145.

[53] Xu, Y., & Yin, W. (2013). A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion. SIAM Journal on imaging sciences, 6(3), 1758-1789.

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

[55] Melville, P., Mooney, R. J., & Nagarajan, R. (2002, July). Content-boosted collaborative filtering for improved recommendations. In Aaai/iaai (pp.

187-192).

[56] Gu, Q., Zhou, J., & Ding, C. H. (2010, April). Collaborative Filtering: Weighted Nonnegative Matrix Factorization Incorporating User and Item Graphs. In SDM (pp. 199-210).

[57] Vozalis, M., & Margaritis, K. G. (2004, August). Collaborative filtering enhanced by demographic correlation. In AIAI symposium on professional practice in AI, of the 18th world computer congress.

[58] Lika, B., Kolomvatsos, K., & Hadjiefthymiades, S. (2014). Facing the cold start problem in recommender systems. Expert Systems with Applications, 41(4), 2065-2073.

[59] Zhou, T., Shan, H., Banerjee, A., & Sapiro, G. (2012, August). Kernelized Probabilistic Matrix Factorization: Exploiting Graphs and Side Information. In SDM (Vol. 12, pp. 403-414).

[60] Houlsby, N., Hernández-Lobato, J. M., & Ghahramani, Z. (2014, January). Cold-start Active Learning with Robust Ordinal Matrix Factorization. In *ICML* (pp. 766-774).

[61] Ostrikov, A., Rokach, L., & Shapira, B. (2013, October). Using geospatial metadata to boost collaborative filtering. In *Proceedings of the 7th ACM conference on Recommender systems* (pp. 423-426). ACM.

[62] Ma, H., Zhou, D., Liu, C., Lyu, M. R., & King, I. (2011, February). Recommender systems with social regularization. In *Proceedings of the fourth ACM international conference on Web search and data mining* (pp. 287-296). ACM.

[63] Tiroshi, A., Berkovsky, S., Kaafar, M. A., Vallet, D., Chen, T., & Kuflik, T. (2014, February). Improving business rating predictions using graph based features. In *Proceedings of the 19th international conference on Intelligent User Interfaces* (pp. 17-26). ACM.

[64] Nguyen, A. T., Denos, N., & Berrut, C. (2007, October). Improving new user recommendations with rule-based induction on cold user data. In *Proceedings of the 2007 ACM conference on Recommender systems* (pp. 121-128). ACM.

[65] Ahn, H. J. (2008). A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem. *Information Sciences*, *178*(1), 37-51.

[66] Ren, Y., Li, G., Zhang, J., & Zhou, W. (2013). Lazy collaborative filtering for data sets with missing values. *IEEE transactions on cybernetics*, *43*(6), 1822-1834.

[67] Massa, P., & Avesani, P. (2007, October). Trust-aware recommender systems. In *Proceedings of the 2007 ACM conference on Recommender systems* (pp. 17-24). ACM.

[68] Zou, H., Gong, Z., Zhang, N., Zhao, W., & Guo, J. (2015). TrustRank: a Cold-Start tolerant recommender system. *Enterprise Information Systems*, *9*(2), 117-138.

[69] Chartrand, R., & Wohlberg, B. (2013, May). A nonconvex ADMM algorithm for group sparsity with sparse groups. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing* (pp. 6009-6013). IEEE.

[70] Jiang, Z., Lin, Z., & Davis, L. S. (2013). Label consistent K-SVD: Learning a discriminative dictionary for recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *35*(11), 2651-2664.

[71] Koren, Y. (2008, August). Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 426-434). ACM.

[72] Jain, A. K. (2010). Data clustering: 50 years beyond K-means. *Pattern recognition letters*, *31*(8), 651-666.

[73] Guo, G., Zhang, J., & Yorke-Smith, N. (2015). Leveraging multiviews of trust and similarity to enhance clustering-based recommender systems. *Knowledge-Based Systems*, *74*, 14-27.

[74] Alam, S., Dobbie, G., Riddle, P., & Koh, Y. S. (2013). Analysis of web usage data for clustering based recommender system. In *Trends in Practical Applications of Agents and Multiagent Systems* (pp. 171-179). Springer International Publishing.

[75] Cacheda, F., Carneiro, V., Fernández, D., & Formoso, V. (2011).

Comparison of collaborative filtering algorithms: Limitations of current techniques and proposals for scalable, high-performance recommender systems. *ACM Transactions on the Web (TWEB)*, *5*(1), 2.

[76] Lee, H., Yoo, J., & Choi, S. (2010). Semi-supervised nonnegative matrix factorization. *IEEE Signal Processing Letters*, *17*(1), 4-7.

[77] Ji, K., Sun, R., Li, X., & Shu, W. (2016). Improving matrix approximation for recommendation via a clustering-based reconstructive method. *Neurocomputing*, *173*, 912-920.

[78] Kabbur, S., Ning, X., & Karypis, G. (2013, August). Fism: factored item similarity models for top-n recommender systems. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 659-667). ACM.

[79] Pereira, A. L. V., & Hruschka, E. R. (2015). Simultaneous co-clustering and learning to address the cold start problem in recommender systems. *Knowledge-Based Systems*, *82*, 11-19.

[80] Adamopoulos, P. (2013, October). Beyond rating prediction accuracy: on new perspectives in recommender systems. In *Proceedings of the 7th ACM conference on Recommender systems* (pp. 459-462). ACM.

[81] McNee, S. M., Riedl, J., & Konstan, J. A. (2006, April). Being accurate is not enough: how accuracy metrics have hurt recommender systems. In *CHI'06 extended abstracts on Human factors in computing systems* (pp. 1097-1101). ACM.

[82] Zhang, M., & Hurley, N. (2008, October). Avoiding monotony: improving the diversity of recommendation lists. In *Proceedings of the 2008 ACM conference on Recommender systems* (pp. 123-130). ACM.

[83] Pathak, A., & Patra, B. K. (2015, March). A knowledge reuse framework for improving novelty and diversity in recommendations. In *Proceedings of the Second ACM IKDD Conference on Data Sciences* (pp. 11-19). ACM.

[84] Santos, R. L., Castells, P., Altingövde, I. S., & Can, F. (2014, February). Diversity and novelty in web search, recommender systems and data streams. In *WSDM* (pp. 679-680).

[85] Vargas, S. (2014, July). Novelty and diversity enhancement and evaluation in recommender systems and information retrieval. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval* (pp. 1281-1281). ACM.

[86] Adomavicius, G., & Kwon, Y. (2012). Improving aggregate recommendation diversity using ranking-based techniques. *IEEE Transactions on Knowledge and Data Engineering*, *24*(5), 896-911.

[87] Zhang, M., & Hurley, N. (2009, September). Novel item recommendation by user profile partitioning. In *Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology-Volume 01* (pp. 508-515). IEEE Computer Society.

[88] Hurley, N., & Zhang, M. (2011). Novelty and diversity in top-n recommendation--analysis and evaluation. *ACM Transactions on Internet Technology (TOIT)*, *10*(4), 14.

[89] Oh, J., Park, S., Yu, H., Song, M., & Park, S. T. (2011, December). Novel recommendation based on personal popularity tendency. In *2011 IEEE 11th International Conference on Data Mining* (pp. 507-516). IEEE.

[90] Vargas, S., & Castells, P. (2014, October). Improving sales diversity by recommending users to items. In *Proceedings of the 8th ACM Conference on Recommender systems* (pp. 145-152). ACM.

[91] Niemann, K., & Wolpers, M. (2013, August). A new collaborative filtering approach for increasing the aggregate diversity of recommender systems. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 955-963). ACM.

[92] Wang, S., Gong, M., Li, H., & Yang, J. (2016). Multi-objective optimization for long tail recommendation. *Knowledge-Based Systems*, *104*, 145-155.

[93] Valcarce, D., Parapar, J., & Barreiro, Á. (2016). Item-based relevance modelling of recommendations for getting rid of long tail products. *Knowledge-Based Systems*, *103*, 41-51.

[94] Ziegler, C. N., McNee, S. M., Konstan, J. A., & Lausen, G. (2005, May).

Improving recommendation lists through topic diversification. In *Proceedings of the 14th international conference on World Wide Web* (pp. 22-32). ACM.

[95] Said, A., Kille, B., Jain, B. J., & Albayrak, S. (2012). Increasing diversity through furthest neighbor-based recommendation. *Proceedings of the WSDM*, *12*.

[96] Kabutoya, Y., Iwata, T., Toda, H., & Kitagawa, H. (2013, April). A probabilistic model for diversifying recommendation lists. In *Asia-Pacific Web Conference* (pp. 348-359). Springer Berlin Heidelberg.

[97] Salakhutdinov, R., Mnih, A., & Hinton, G. (2007, June). Restricted Boltzmann machines for collaborative filtering. In *Proceedings of the 24th international conference on Machine learning* (pp. 791-798). ACM.

[98] Bengio, Y., Lamblin, P., Popovici, D., & Larochelle, H. (2007). Greedy layer-wise training of deep networks. *Advances in neural information processing systems*, *19*, 153.

[99] Längkvist, M., & Loutfi, A. (2012). Learning Representations with a Dynamic Objective Sparse Autoencoder. In *Neural Information Processing Systems*.

[100] Olshausen, B. A. (1996). Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, *381*(6583), 607-609.

[101] Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., & Manzagol, P. A. (2010). Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, *11*(Dec), 3371-3408.

[102] Chen, M., Weinberger, K. Q., Sha, F., & Bengio, Y. (2014). Marginalized Denoising Auto-encoders for Nonlinear Representations. In *ICML* (pp. 1476-1484).

[103] Strub, F., & Mary, J. (2015). Collaborative Filtering with Stacked Denoising AutoEncoders and Sparse Inputs. In *NIPS Workshop on Machine Learning for eCommerce*.

[104] Ouyang, Y., Liu, W., Rong, W., & Xiong, Z. (2014, November). Autoencoder-based collaborative filtering. In *International Conference on Neural*

Information Processing (pp. 284-291). Springer International Publishing.

[105] Wu, Y., DuBois, C., Zheng, A. X., & Ester, M. (2016, February). Collaborative denoising auto-encoders for top-n recommender systems. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining* (pp. 153-162). ACM.

[106] Strub, F., Mary, J., & Gaudel, R. (2016). Hybrid Collaborative Filtering with Autoencoders.

[107] Li, S., Kawale, J., & Fu, Y. (2015, October). Deep collaborative filtering via marginalized denoising auto-encoder. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management* (pp. 811-820). ACM.

[108] Zhang, Q., & Li, B. (2010, June). Discriminative K-SVD for dictionary learning in face recognition. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on* (pp. 2691-2698). IEEE.

[109] Larochelle, H., & Bengio, Y. (2008, July). Classification using discriminative restricted Boltzmann machines. In *Proceedings of the 25th international conference on Machine learning* (pp. 536-543). ACM.

[110] Combettes, P. L., & Pesquet, J. C. (2011). Proximal splitting methods in signal processing. In *Fixed-point algorithms for inverse problems in science and engineering* (pp. 185-212). Springer New York.

[111] Komodakis, N., & Pesquet, J. C. (2015). Playing with duality: An overview of recent primal? dual approaches for solving large-scale optimization problems. IEEE Signal Processing Magazine, 32(6), 31-54.

APPENDIX

A.1 Blind Compressive Sensing

Compressive sensing (CS) paradigm[1] focuses on the recovery of sparse or compressible signals from a given set of observations, obtained at sub-Nyquist rates. A sparse signal x can be uniquely and exactly recovered by minimizing the l_1 norm (a convex surrogate of the l_0 norm) of the vector to be recovered under the constraint that recovery is consistent with the observations.

Real world signal are rarely sparse in spatial/time domain. However, most have a sparse/compressible transform domain representation. In such a case, we can model the recovery of signal x as follows

$$\min_{s} \|s\|_1 \; subject \; to \; y = A\Psi s \tag{1}$$

where, A is the sensing matrix; Ψ is the sparsifying transform; s is the sparse representation of x s.t. $x = \Psi s$. Above equation can be used to recover s and hence x accurately if the sensing and sparsifying matrices are mutually incoherent.[2,3]

In practice, the measurement process is corrupted by noise; therefore one needs to solve a noisy under-determined system, $y = Ax + \eta = A\Psi s + \eta$, where the noise η is assumed to be Normally distributed. For such a system, the equality constrained problem (1) is relaxed, and the following is solved instead;

1. Donoho, D. L. (2006). Compressed sensing. *IEEE Transactions on information theory*, *52*(4), 1289-1306.
2. Natarajan, B. K. (1995). Sparse approximate solutions to linear systems. *SIAM journal on computing*, *24*(2), 227-234.
3. Candes, E., & Romberg, J. (2007). Sparsity and incoherence in compressive sampling. *Inverse problems*, *23*(3), 969.

$$\min_{s} \|s\|_1 \; subject\ to \; \|y - A\Psi s\|_2^2 \le \varepsilon, \text{ where } \varepsilon = n\sigma^2 \tag{2}$$

Usually, the unconstrained counterpart (3) of (2) is solved[3], where, λ is the regularization parameter.

$$\min_{s} \|y - A\Psi s\|_2^2 + \lambda \|s\|_1 \tag{3}$$

In CS it is assumed that the signal is sparse in a known basis (say Ψ). However, proponents of dictionary learning (DL) claim that it is possible to get even better results if the sparsifying dictionary is empirically learned; i.e. given a training dataset $X = [x_1 | x_2 | ... | x_N]$, one learns a dictionary D such that the resulting coefficients (S) are sparse i.e.

$$X_{m \times N} = D_{m \times M} S_{M \times N} + \eta \tag{4}$$

The learning problem is formally it is expressed as:

$$\min_{D} \|X - DS\|_F^2 \text{ such that } \|S\|_0 \le \tau \tag{5}$$

Here, X is the signal, D is the dictionary to be learned and S is the sparse coefficient matrix.

Blind Compressed Sensing (BCS)[4], marries dictionary learning with Compressed Sensing. It should be noted that dictionary learning is an offline technique, i.e. it cannot be used for signal recovery/reconstruction. One should have some training data to learn the dictionary offline, after which the learned dictionary can be applied to other signal recovery problems.

In BCS, one is interested in simultaneously learning the dictionary and recovering the signal. However, this is not possible if there is only a single measurement vector - estimating a dictionary and signal from a single sample will not be robust. BCS is applicable for multiple measurement vectors (MMV) recovery problems of the following

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

form:

$$Y = AX + \eta \tag{6}$$

where, $X = [x_1 | x_2 | ... | x_N]$ and $Y = [y_1 | y_2 | ... | y_N]$

Instead of assuming X to be sparse in a known basis (Ψ), BCS assumes that it is sparse in a (yet to be) learned basis (D). Therefore BCS expresses (7) as:

$$Y = ADS + \eta \tag{7}$$

Both the dictionary as well as the sparse coefficients are learned simultaneously in BCS as in (9). This is unlike DL, where the emphasis was on learning only the sparsifying dictionary.

$$\min_{D,S} \|Y - ADS\|_F^2 + \lambda \|vec(S)\|_1 + \gamma \|D\|_F^2 \tag{8}$$

BCS imposes a simple Frobenius norm penalty on the dictionary, this acts mainly as a regularization term and yields a dense dictionary.

A.2 Elastic-Net Regularization

Consider the classical regression problem:

$$y = Ax + \eta,\ \eta \sim N(0, \sigma^2) \tag{9}$$

Where, y is a vector of the collected data; matrix A consists of the explanatory variables; x is the unknown weight vector which interprets the data in terms of the explanatory variables and η is the noise.

Since the noise is assumed to be Normally distributed, one needs to minimize the least squares cost function. Usually, the problem is not well conditioned and needs to be regularized.

The most straightforward regularization is the ridge regression, which is expressed as:

$$x_{ridge} = \min_x \|y - Ax\|_2^2 + \lambda \|x\|_2^2 \tag{10}$$

Unfortunately, ridge regression results in a dense solution, i.e. it explains the data in terms of all the explanatory variables; it lacks interpretability.

To overcome this issue, the LASSO[5] (least angle shrinkage and selection operator) was proposed.

LASSO replaces the l_2 norm constraint by an l_1 norm:

$$x_{lasso} = \min_x \|y - Ax\|_2^2 + \lambda \|x\|_1 \tag{11}$$

The l_1 norm penalty promotes selection of very few variables, i.e. the weight vector x is sparse. The selection of few variables improves interpretability. One can now analyze and interpret the data with only a few explanatory variables.

However, LASSO suffers from a serious shortcoming. In most cases the explanatory variables are not independent of each other, they are correlated. In such a situation, the LASSO selects only one variable from the group of correlated ones. This is the result of enforcing too much sparsity on the variable selection operation. One would like to know all the variables which have contributed to the outcome (y), but LASSO ignores the correlated variables and thus loses out on correct interpretability.

To promote the grouping of correlated variables, the elastic net regularization[6] was proposed. The optimization problem is framed as follows:

$$x_{net} = \min_x \|y - Ax\|_2^2 + \lambda_1 \|x\|_1 + \lambda_2 \|x\|_2^2 \tag{12}$$

Here the l_1 norm constraint promotes sparsity (as in LASSO), but the l_2 norm constraint promotes selection of correlated variables.

5. Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 267-288.
6. Zou, H., & Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *67*(2), 301-320.

A.3 Majorization - Minimization Technique

Majorization-Minimization (MM) approach enables us to break a complex optimization problem into simpler and easier to solve steps.

Consider an underdetermined system of equation, $y = Ax$ where, $A \in \mathbb{R}^{m \times n}, where\ m \ll n$ is a fat matrix. As A is not full rank, we can compute x by least square minimization as in (14), which involves computation of inverse of large matrices.

$$x = \left(A^T A\right)^{-1} A^T y \tag{13}$$

For cases where the variables to be recovered are very large, such as in recommender systems, the size of $\left(A^T A\right)$ becomes prohibitively large to efficiently compute its inverse within reasonable resource requirements. Use of MM approach eliminates the need to compute such inverses and significantly reduces the computation burden.

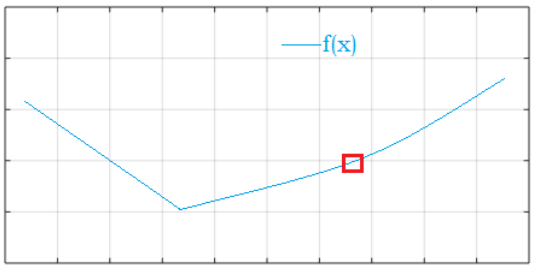
MM approach essentially involves replacing a complex or computationally intensive optimization problem by a series of simpler, easier to solve optimization steps.

Consider an original function $f(x)$ and an initial estimate for its minima (x_0) as shown in figure A.1 (a). MM technique replaces the minimization of $f(x)$ by minimization of a surrogate function $g_0(x)$ such that $g_0(x)$ is a majorizer of $f(x)$ i.e.

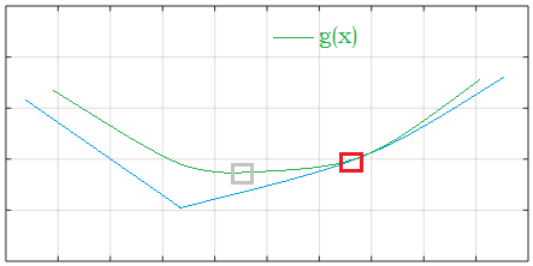
1. $g_0(x) \geq f(x) \forall x$
2. $g_0(x_0) = f(x_0)$

$g_0(x)$ is defined such that it is easier to optimize than $f(x)$.

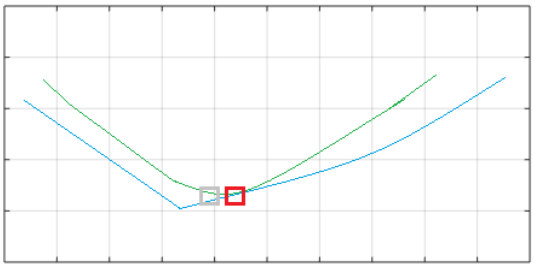
The minimum of $g_0(x), x_1$ hence obtained acts as the new estimate for the minima. Thus, at each iteration (k), a new majorizer function $g_k(x)$ is defined and its minimum, x_k forms the new estimate for the minima. The above procedure is repeated until convergence.



a. Original function $f(x)$ with initial guess x_0 for minima marked (red box)



b. $g_0(x)$ defined at x_0; optimized to obtain its minima (gray box)



c. Minima of $g_0(x)$ acts as next guess for minmia of $f(x)$

Figure 7.1 Majorization-Minimization Technique

Consider $f(x) = \|y - Ax\|_2^2$. Solving the same requires computing pseudoinverse of A, which can be problematic if A is a very large or ill-conditioned matrix. To eliminate need for such computation, we can use MM technique, defining the majorizer of $f(x)$ as

$$g_k(x) = \|y - Ax\|_2^2 + (x - x_k)^T (\alpha I - A^T A)(x - x_k) \quad (14)$$

Under the condition, $\alpha \geq \max\left(eigen\_value\left(A^T A\right)\right)$, the second term in (14), is always non-negative and thus, $g_k(x)$ is a majorizer of $f(x)$.

Equation (14) can be simplified as follows

$$\begin{aligned} g_k(x) &= (y-Ax)^T (y-Ax) + (x-x_k)^T \left(\alpha I - A^T A\right)(x-x_k) \\ &= y^T y + x_k^T \left(\alpha I - A^T A\right) x_k + \alpha x^T x - 2\left(y^T A + x_k^T \left(\alpha I - A^T A\right)\right) x \\ &= \alpha\left(-2 z_k{}^T x + x^T x\right) + y^T y + x_k^T \left(\alpha I - A^T A\right) x_k \end{aligned} \tag{15}$$

where, $z_k = \frac{1}{\alpha}\left(A^T y + \left(\alpha I - A^T A\right) x_k\right) = x_k + \frac{1}{\alpha} A^T (y - A x_k)$

Using z_k and updating $g_k(x)$ we get

$$g_k(x) = \|z_k - x\|_2^2 - \alpha\left(z_k{}^T z_k\right) + y^T y + x_k^T \left(\alpha I - A^T A\right) x_k \tag{16}$$

Considering the last three terms in $g_k(x)$ as constants, since they do not contain the optimization variable, the equation is updated to

$$\begin{aligned} &g_k(x) = \|z_k - x\|_2^2 + C \\ &\text{where, } C = -\alpha\left(z_k{}^T z_k\right) + y^T y + x_k^T \left(\alpha I - A^T A\right) x_k \end{aligned} \tag{17}$$

Minimizing (17), is relatively simpler than minimizing the original function, $f(x) = \|y - Ax\|_2^2$ as it does not involve only simple addition/multiplication operations.

A.4 Split Bregman Technique

In this section, we will review the split Bregman technique.

Bregman distance forms the basis for the formulation of these techniques. For a convex function $E: X \to \mathbb{R}$, where $u, v \in X$ and p belongs to the set of sub-gradient of the function, Bregman distance D_E^p is defined as follows:

$$D_E^p(u,v) = E(u) - E(v) - \langle p, u - v \rangle \tag{18}$$

Consider the objective function given in (19) where $\Phi(u)$ and $H(u)$ are convex and H is differentiable.

$$\min_{u} \ |\Phi(u)|_1 + H(u) \tag{19}$$

Split Bregman technique focuses on decomposing a complex optimization problem (like equation 19), with multiple norm terms, such that they form different sub-problems which can be solved easily compared to the original composite objective function.

Rewriting (19) by letting $d = \Phi(u)$ we get

$$\begin{aligned} &\min_{u} |d|_1 + H(u) \\ &\textit{Subject to } d = \Phi(u) \end{aligned} \tag{20}$$

The unconstrained equivalent of (20) is obtained by adding a penalization function to the problem as in (21).

$$\min_{u,d} |d|_1 + H(u) + \frac{\lambda}{2} \| d - \Phi(u) \|_2^2 \tag{21}$$

Considering $E(u,d) = |d|_1 + H(u)$ and using (18) we can write the iterative update of (21) as

$$\begin{aligned} (u^{k+1}, d^{k+1}) &= \min_{u,d} D_E^p \left(u, u^k, d\ , d^k\right) + \frac{\lambda}{2} \| d - \Phi(u) \|_2^2 \\ &= \min_{u,d} E(u,d) - \left\langle p_u^k, u - u_k \right\rangle - \left\langle p_d^k, d - d_k \right\rangle + \frac{\lambda}{2} \| d - \Phi(u) \|_2^2 \\ p_u{}^{k+1} &= p_u{}^k + \lambda (\nabla \Phi)^T \left(\Phi u^{k+1} - d^{k+1}\right) \\ p_d{}^{k+1} &= p_d{}^k + \lambda \left(d^{k+1} - \Phi u^{k+1}\right) \end{aligned} \tag{22}$$

The 1st update step can be solved using ADMM (alternating direction method of multipliers) [111] by alternatingly optimizing over each variable. A simplified form of (22) is given below.

$$u^{k+1} = \min_{u} H(u) + \frac{\lambda}{2} \| d - \Phi(u) - b^{k} \|_2^2$$

$$d^{k+1} = \min_{d} |d|_1 + \frac{\lambda}{2} \| d - \Phi(u) - b^{k} \|_2^2 \qquad (23)$$

$$b^{k+1} = b^{k} + \left(\Phi u^{k+1} - d^{k+1} \right)$$

Since $H(u)$ is differentiable everywhere, update for u can be solved analytically. The solution for d is nothing but the solution for synthesis prior formulation and is obtained directly by shrinkage (soft thresholding) operator. The last step is the update of Bregman variable.

Use of split Bregman technique aids in faster convergence and lower recovery errors, as no cooling of regularization parameter, is required and thus optimal values of regularization parameters for each of the sub-problem can be set.