Blind Regression: Nonparametric Regression for Latent Variable Models via Collaborative Filtering

by

Dogyoon Song

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Science in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2016

Author . . . .
Department of Electrical Engineering and Computer Science
May 20, 2016

Certified by . . . .
Devavrat Shah
Professor
Thesis Supervisor

Accepted by . . . .
Leslie A. Kolodziejski
Chair, Department Committee on Graduate Students

Blind Regression: Nonparametric Regression for Latent Variable Models via Collaborative Filtering

by

Dogyoon Song

Submitted to the Department of Electrical Engineering and Computer Science
on May 20, 2016, in partial fulfillment of the
requirements for the degree of
Master of Science in Electrical Engineering and Computer Science

Abstract

Recommender systems are tools that provide suggestions for items that are most likely to be of interest to a particular user; they are central to various decision making processes so that recommender systems have become ubiquitous. We introduce *blind regression*, a framework motivated by *matrix completion* for recommender systems: given m users, n items, and a subset of user-item ratings, the goal is to predict the unobserved ratings given the data, i.e., to complete the partially observed matrix. We posit that user u and movie i have features $x_1(u)$ and $x_2(i)$ respectively, and their corresponding rating $y(u,i)$ is a noisy measurement of $f(x_1(u), x_2(i))$ for some unknown function f. In contrast to classical regression, the features $x = (x_1(u), x_2(i))$ are not observed (latent), making it challenging to apply standard regression methods.

We suggest a two-step procedure to overcome this challenge: 1) estimate distance for latent variables, and then 2) apply nonparametric regression. Applying this framework to matrix completion, we provide a prediction algorithm that is consistent for all Lipschitz functions. In fact, the analysis naturally leads to a variant of collaborative filtering, shedding insight into the widespread success of collaborative filtering. Assuming each entry is revealed independently with $p = \max(m^{-1+\delta}, n^{-1/2+\delta})$ for $\delta > 0$, we prove that the expected fraction of our estimates with error greater than ϵ is less than γ^2/ϵ^2, plus a polynomially decaying term, where γ^2 is the variance of the noise.

Experiments with the MovieLens and Netflix datasets suggest that our algorithm provides principled improvements over basic collaborative filtering and is competitive with matrix factorization methods. The algorithm and analysis naturally extend to higher order tensor completion by simply flattening the tensor into a matrix. We show that our simple and principled approach is competitive with respect to state-of-art tensor completion algorithms when applied to image inpainting data. Lastly, we conclude this thesis by proposing various related directions for future research.

Thesis Supervisor: Devavrat Shah
Title: Professor

Acknowledgments

I would like to express my gratitude to my thesis supervisor, Devavrat Shah, for all his guidance and encouragement over the last two years.

I would also like to thank Christina Lee and Yihua Li for helpful discussions on related subjects.

Lastly, I would like to thank my parents, whom I owe everything for their unconditional love.

This research was supported in part by a Samsung Scholarship, an MIT EECS Advanced Television & Signal Processing Fellowship, and a Siebel Scholarship.

Contents

List of Figures

Chapter 1

Introduction

1.1 Background

Recommender systems have become ubiquitous in our lives. They help us filter a vast amount of information we encounter into smaller selections of likable items customized to the users' tastes. Amazon recommends items to customers; Netflix recommends movies to users; and LinkedIn recommends job positions to users or candidate profiles to recruiters.

There have been many studies on recommender systems, which were fueled by the Netflix Prize competition that began in October 2006 [5, 6, 7]. Because of the competition, the research community was able to gain access to large-scale data consisting of 100 million movie ratings, and a huge group of researchers were attracted to attack the problem. The competition has encouraged rapid development in techniques to improve prediction accuracy. As a result, much progress has been made in the field of collaborative filtering.

One natural approach for recommendation is to use auxiliary/exogenous content information about the users or items. For example, the information on the director, the lineup of actors, genre, or the language can help us figure out whether two movies are similar or not. Likewise, information including age, geographic location, and academic background reveals some characteristics of a user. Recommendations based on such content-specific data is called content filtering [4, 8, 44]. With quantitative content features, this setting becomes that of traditional regression problems.

However, in practice, recommendations are often made via a technique called *collaborative filtering* (CF), which provides recommendations in a content-agnostic way by exploiting patterns to determine similarity between users or items. The use of such techniques is partly because the exogenous content information (or its quantitative representation) is usually not available. Moreover, it is known that recommender systems purely based on content generally suffer from the problems of *limited content analysis* and *over-specialization* [48].

Instead of relying on content information, collaborative filtering approaches use the rating information of other users and items in the system. For example, if two users are revealed to have similar tastes, a CF algorithm might recommend the items the first user liked to the second user (user-user CF). On the other hand, if many users agree on two items, a CF algorithm might recommend the second item to a user who likes the first (item-item CF). Collaborative filtering has been successful and used extensively for decades including Amazon's recommendation system [32] and the Netflix Prize winning algorithm by BellKor's Pragmatic Chaos [29].

There are two primary approaches to relate two different entities: users and items, by utilizing such similarities. They are two main branches of CF: one is the *neighborhood-based mehod*, and the other is the *latent factor method* (=model-based method). Neighborhood-based methods concentrate on similarities (or relationships) between items or between items. For example, an item-item CF models the preference of a user for an item based on the past ratings of similar items by the user. Latent factor methods comprise an alternative approach by transforming both items and users to the same (low-dimensional) latent factor space. Matrix factorization methods, such as singular value decomposition (SVD) is an example of a latent factor method [30, 41].

Although latent factor models have gained popularity because of their relatively high accuracy and theoretical elegance, neighborhood-based approaches to CF are still widely used in practice. One reason for this is that good prediction accuracy is not the sole objective for recommender systems. Other factors, for instance, recommendation serendipity, can play an important role in the appreciation of users [21, 47].

The main advantages of neighborhood-based methods are as follows. They are intuitive and simple to implement (simplicity). They can provide intuitive explanations for the reasons recommendations work (justifiability). Unlike most model-based systems, they require less or no costs of training phases, which need to be carried at frequent intervals in large commercial applications; meanwhile, storing nearest neighbors requires very little memory (efficiency). In addition, a neighborhood-based approach is little affected by the constant addition of new data (stability). That is, once item similarities have been computed, an item-based system can readily provide immediate recommendations to a newly entered user based on her feedback. This property makes it desirable for an online recommendation setting [41].

1.2 Related Work

The term collaborative filtering was coined in [19]. Collaborative filtering approaches can be grouped into two general classes: *model* and *neighborhood*-based methods.

Model-based CF: Model-based approaches use the stored ratings to learn a predictive model. Principal characteristics of users and items are captured by a set of model parameters, learned from a training dataset, and used to predict ratings on new items. There have been numerous model-based approaches toward the task of recommendation, which include Bayesian Clustering [11], Latent Dirichlet Allocation [9], Maximum Entropy [54], Boltzmann Machines [46], Support Vector Machines [20], and Singular Value Decomposition [50, 43, 28, 51].

Low-rank Matrix Factorization: In the recent years, there has been exciting theoretical development in the context of matrix-factorization-based approaches. Since any matrix can be factorized, its entries can be described by a function with the form $f(x_1, x_2) = x_1^T diag(\sigma) x_2$, and the goal of factorization is to recover the latent features for each row and column. [50] was one of the earlier works to suggest the use of low-rank matrix approximation, observing that a low-rank matrix has a comparatively small number of free parameters. Subsequently, statistically efficient approaches were suggested using optimization based estimators, proving that matrix factorization can fill in the missing entries with sample complexity as low as $rn \log n$, where r is the rank of the matrix [15, 25, 45, 40, 23]. Also, there has been an exciting line of ongoing work to make the resulting algorithms faster and scalable [17, 14, 31, 49, 34, 38].

These approaches are based on the structural assumption that the underlying matrix is *low-rank* and the matrix entries are reasonably "incoherent". Unfortunately, the low-rank assumption may not hold in practice. The recent work [18] makes precisely this observation, showing that a simple non-linear, monotonic transformation of a low-rank matrix could easily produce an effectively high-rank matrix, despite few free model parameters. They provide an algorithm and analysis specific to the form of their model, which achieves sample complexity of $O((mn)^{2/3})$. However, their algorithm only applies to functions f which are a nonlinear monotonic transformation of the inner product of the latent features. The limitations of these approaches lie in the restrictive assumptions of the model.

Neighborhood-based CF: In neighborhood-based (also called memory-based) collaborative filtering, the stored user-item ratings are directly used to predict ratings for new pairs of user-item. This prediction can be done in two ways: *user-based* or *item-based*. User-based systems, such as Ringo [48], GroupLens [27], and Bellcore video [22], evaluate the preference of a target user for items by using the ratings for the items by other users (neighbors) who have similar preference patterns.

There are two main paradigms in neighborhood-based collaborative filtering: the user-user paradigm and the item-item paradigm. To recommend items to a user in the user-user paradigm, one first looks for similar users, and then recommends items liked by those similar users. In the item-item paradigm, in contrast, items similar to those liked by the user are found and subsequently recommended. Much empirical evidence exists that the item-item paradigm performs well in many cases [47, 32, 16, 41], however the theoretical understanding of the method has been limited. In recent works, Latent mixture models have been introduced to explain the collaborative filtering algorithm as well as the empirically observed superior performance of item-item paradigms, c.f. [12, 13].

However, these results assume a specific parametric model, such as a mixture distribution model for preferences across users and movies. We hope that by providing an analysis for collaborative filtering within our broader nonparametric model, we can provide a more complete understanding of the potentials and limitations of collaborative filtering.

Tensor completion: A tensor is the higher-dimensional analogue of a matrix (or a vector). Therefore, it is natural to consider extending the neighborhood-based approaches to the context of tensor completion; however, there is little known literature about this setting.

Tensor completion is known to be much harder than matrix completion. Tensors do not have a canonical decomposition such as the singular value decomposition (SVD) for a matrix, which simultaneously possesses two desirable properties: (i) it computes a rank-r decomposition, and (ii) it yields orthonormal row/column matrices. These properties makes obtaining a decomposition for a tensor challenging [26]. There have been recent developments in obtaining an efficient rank-1 tensor decomposition [1], which is effective in learning latent variable models and estimating missing data [24, 42]. In the context of learning latent variable models or mixture distributions, there have been developments in non-negative matrix/tensor factorizations [3, 2] which go beyond SVD.

Kernel regression: The algorithm we propose in this work is inspired by the local approximation of functions by the Taylor series expansion. We would first build local estimators with observed ratings, and then combine these with appropriately chosen weights. For this reason, there is a connection to the classical setting of kernel regression, which also relies on smoothed local approximations [35, 52]. However, both the power series expansion and the kernel regression require explicit knowledge of the geometry of feature space, which is not permitted in the setting of recommender systems. As a result, their analysis and proof techniques do not extend to our context of Blind regression, in which the features are latent; the analysis required is entirely different despite the similarity in the form of computing a convex combination of nearby datapoints.

1.3 Our Contribution

In contrast to numerous empirical attempts to obtain accurate prediction methods, we have few theoretical studies on neighborhood-based models. One objective of this thesis is to provide a general statistical framework for performing nonparametric regression over latent variable models, from which a neighborhood-based algorithm with provable performance bounds follows. We are initially motivated by the problem of matrix completion arising in the context of designing recommendation systems, but we additionally show that our framework allows for systematic extensions to higher order tensor completion as well.

In the popularized setting of Netflix, there are m users, indexed by $u \in [m]$, and n movies, indexed by $i \in [n]$. Each user u has a rating for each movie i, denoted as $y(u,i)$. The system observes ratings for only a small fraction of user-movie pairs. The goal is to predict ratings for the rest of the unknown user-movie pairs, i.e., to complete the partially observed $m \times n$ rating matrix. To be able to obtain meaningful predictions from the partially observed matrix, it is essential to impose a structure on the data.

We assume each user u and movie i is associated to features $x_1(u) \in \mathcal{X}_1$ and $x_2(i) \in \mathcal{X}_2$ for some compact metric spaces $\mathcal{X}_1, \mathcal{X}_2$. We assume that the latent features are drawn independently from an identical distribution (IID) with respect to some Borel probability measures on $\mathcal{X}_1, \mathcal{X}_2$. Following the philosophy of non-parametric statistics, we assume that there exists some function $f : \mathcal{X}_1 \times \mathcal{X}_2 \to \mathbb{R}$ such that the rating of user u for movie i is given by

$$y(u,i) = f(x_1(u), x_2(i)) + \eta_{ui}, \tag{1.1}$$

where η_{ui} is some independent bounded noise. However, we refrain from any specific modeling assumptions on f, requiring only mild regularity conditions following the traditions of non-parametric statistics. We observe ratings for a subset of the user-movie pairs, and the goal is to use the given data to predict $f(x_1(u), x_2(i))$ for all $(u,i) \in [m] \times [n]$ whose rating is unknown.

In classical nonparametric regression, we observe input features $x_1(u), x_2(i)$ along with the rating $y(u,i)$ for each datapoint, and thus we can approximate the function f well using local approximation techniques as long as f satisfies mild regularity conditions. However, in our setting, we do not observe the latent features $x_1(u), x_2(i)$, but instead we only observe the indices (u,i). Therefore, we use *blind regression* to refer to the challenge of performing regression with unobserved latent input variables. This paper addresses the question, does there exist a meaningful prediction algorithm for general nonparametric regression when the input features are unobserved?

Our answer is "yes." In spite of the minimal assumptions of our model, we provide a consistent matrix completion algorithm with finite sample error bounds as well. Furthermore, as a coincidental by-product, we find that our framework provides an explanation for the mystery of "why collaborative filtering algorithms work well in practice."

As the main technical result, we show that the user-user nearest neighbor variant of collaborative filtering method with our similarity metric yields a consistent estimator for any Lipschitz function as long as we observe $\max(m^{-1+\delta}, n^{-1/2+\delta})$ fraction of the matrix with $\delta > 0$. In the process, we obtain finite sample error bounds, whose details are stated in Theorem 1. We compared the Gaussian kernel variant of our algorithm to classic collaborative filtering algorithms and a matrix factorization based approach (softImpute) on predicting user-movie ratings for the Netflix and MovieLens datasets. Experiments suggest that our method improves over existing collaborative filtering methods, and sometimes outperforms matrix-factorization-based approaches depending on the dataset.

There are two conceptual parts to our algorithm. First, we derive an estimate of $f(x_1(u), x_2(i))$ for an unobserved index pair (u, i) by using linear approximation of f pivoted at $(x_1(u'), x_2(i'))$. By Taylor's theorem, this estimates the unknwon $f(x_1(u), x_2(i))$ quite well as long as $x_1(u')$ is close to $x_1(u)$ or $x_2(i')$ is close to $x_2(i)$. However, since the latent features are not observed, we need a method to approximate the distance in the latent space. The second observation we make is that under our mild Lipschitz conditions, the similarity metrics commonly used in collaborative filtering heuristics correspond to an estimate of distances in the latent space. In particular, we use the sample variance of the differences between observations between a pair of users to capture distances between two users in this thesis. Formally speaking, we cannot guarantee that if the sample variance is small, the distance in the latent space is small, yet we show in our analysis that there is a direct relation between the sample variance and the estimation error.

To analyze the performance of our algorithm, we make minimal model assumptions just like any such work in non-parametric statistics. Let the latent features be drawn independently from an identical distribution (IID) over a compact metric spaces; the function f is Lipschitz with respect to the latent space metrics; entries are observed independently with some probability p; and the additive noise in observations is bounded and independently distributed with zero mean.

In addition, there is no reason to limit ourselves to bivariate functions f in (1.1). The equivalent extension of the bivariate latent variable model to multivariate latent models is to extend from matrices to higher order tensors. The algorithm and analysis that we provide for matrix completion also extends to higher order tensor completion, due to the flexible and generic assumptions of our model.

The algorithm discussed above, as well as its analysis, naturally extends beyond matrices, to completing higher order tensors. In Section 4.2, we show that the tensor completion problem can be reduced to matrix completion, and thus we have a consistent estimator for tensor completion under similar model assumptions. We show in experiments that our method is competitive with respect to state of the art tensor completion methods when applied to the the image inpainting problem. Our estimator is naively simple to implement, and its analysis sidesteps the complications of non-unique tensor decompositions. The ability to seamlessly extend beyond matrices to higher order tensors suggests the general applicability and value of the blind regression framework.

1.4 Organization of Thesis

In Chapter 2 of this thesis, we propose a novel nonparametric framework to make predictions without knowledge on the latent function and feature representations. The framework comprises two steps: 1) estimating the distance between unknown feature representations, and 2) running nonparametric kernel regression. Unlike low-rank matrix factorization techniques, this framework does not require strict structural assumptions. Meanwhile, the framework may not need the explicit feature representations of the input, but only their identifiers. This property justifies the name *'blind' regression* and differentiates our framework from classical nonparametric regression.

In Chapter 3, we suggest a recommendation algorithm based on the suggested framework. It is essentially a neighborhood-based algorithm, motivated by the Taylor series expansion and the idea of boosting, which is a machine learning technique. The suggested algorithm is simple to implement, widely applicable due to its nonparametric nature, and fairly competitive in terms of its prediction accuracy.

In Chapter 4, we present our main technical results for the proposed algorithm. To the best of our knowledge, this analysis provides the first provable performance bounds on the sample complexity and prediction accuracy of neighborhood-based methods. Our algorithm is proven to be a consistent estimation algorithm as the size of the matrix grows infinitely large. In Section 4.2, the analysis extends to the tensor completion setting via flattening. In addition, the trade-off of averaging multiple estimators is briefly discussed in Section 4.3 under a set of simplifying assumptions.

Chapter 5 provides lemmas used in proving the main theorem (Theorem 1). Each section of this chapter contains a key lemma used in the proof of the theorem, and auxiliary lemmas if necessary. The proof of the lemmas are based on various concentration inequality techniques, whose details can be found in the Appendix.

In Chapter 6, we present the performance of our algorithm with experiments on real world datasets. First of all, we show our algorithm outperforms other neighbor-based collaborative filtering algorithms on MovieLens and Netflix datasets. Also, its performance is comparable to a matrix factorization method (SoftImpute). Next, we apply our algorithm for image reconstruction via tensor flattening. Despite its simplicity, our algorithm performs nearly as good as the best tensor completion algorithms reported.

In Chapter 7 we summarize all the pieces and discuss some directions for future work.

Chapter 2

Blind Regression

This chapter covers the introduction to our novel framework of *blind regression.* Regression is a statistical process for estimating relationships among variables, which help to understand how dependent variable varies as one or more independent variables changes. Regression includes many techniques, and is widely used for prediction. However, geometric information on the feature space is essential for all of these techniques, whereas it is not available for the class of problems we are interested in. In this chapter, we will briefly review regression, with an emphasis on kernel regression, a non-parametric technique. Then, we will describe the blind regression framework, pointing out its connection to the traditional regression as well as its unique features.

2.1 Traditional Regression

2.1.1 Regression Regime

Regression is a statistcal process for estimating relationships among variables. More specifically, the aim of regression analysis is to describe the value of a dependent variable in terms of other variables. This objective is achieved by estimating a target function of independent variables, called regression function. In many cases, it is also of interest to characterize the variation of dependent variable around the regression function, which can measure the descriptive power of the regression function.

Regression analysis is widely used for prediction and forecasting, and thus has connection to the field of machine learning. Consider the following situation, which is familiar in supervised learning context: we are given a traning dataset consisting of N labeled data points $\{(x_i, y_i)\}_{i=1}^{N}$. Our task is to estimate the relationship between the feature x_i and the response y_i, or to learn the structure concealed in the data, in order to make a prediction $\hat{y}$ when given a new input x.

There are many techniques developed for regression analysis. Some methods, such as linear regression and ordinary least squares estimation, are parametric, in which the regression function is defined in terms of a finite number of unknown parameters to be estimated from the data. On the other hand, nonparametric regression techniques allow the regression function to lie in a more general class of functions, which may possibly constitute an infinite-dimensional space of functions.

Parametric regression models assume a specific form of the underlying function $\mathbf{y} = f_\beta(\mathbf{x})$ where the model involves the following variables: the independent variables $\mathbf{x} \in \mathbb{R}^d$; the dependent variable $\mathbf{y} \in \mathbb{R}$; and the unknown parameters β. For example, a linear regression model can be written as

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} + \epsilon = X\beta + \epsilon,$$

and the objective is to estimate the parameter $\beta = (\beta_0, \beta_1)^T \in \mathbb{R}^{d+1}$ from the data. Sometimes, the matrix X, a stack of independent variable instances $\begin{bmatrix} 1 & x_i \end{bmatrix}$, is called a design matrix. This kind of structural assumption simplifies the problem and makes the parametric models easier to implement and analyze.

One of the most famous and standard approaches to parametric regression is the method of least squares. It attempts to minimize the mean squared error in the dependent variables, or the sum of squared residuals $\sum_{i=1}^{N} (y_i - x_i\beta)^2$, assuming there are zero or negligible errors in the independent variables. It has closed-form expressions for $\hat{\beta} = (X^TX)^{-1}X^TY$ and a nice geometric interpretation. The least square solution $\hat{\mathbf{y}}(\mathbf{x}) = \mathbf{x}\hat{\beta}$ is a Euclidean projection of y onto the subspace spanned by X.

Although parametric approaches are efficient, making incorrect structural assumptions may result in a poor estimator. For example, estimating a trigonometric function with polynomial bases will require an unnecessarily large number of parameters. Moreover, an attempt to estimate a quadratic function with a linear model will not be productive. Sometimes, machine learning practitioners deal with this problem with model selection and regularization techniques.

There is an alternative approach to overcome the rigidity of parametric regression, which is commonly referred to as nonparametric regression. This is a generic term for methods which do not make a priori structural assumptions for the underlying function. Typical examples include kernel regression and spline interpolation, which allow the data to decide which function fits them the best via local approximation. Abandoning such restrictions imposed by a parametric model allows more generality for this approach. However, nonparametric methods are computationally more expensive compared to parametric models.

2.1.2 Kernel Regression

There are several approaches to the nonparametric regression. Some of the most popular methods are based on local function smoothing, using kernel functions, spline functions, and wavelets. Each of these techniques has its own strengths and weaknesses. Among various nonparametric estimators, kernel estimators have the advantage of being intuitive, and are simple to analyze.

In any nonparametric regression, the objective is to find an estimator $\hat{f}(X)$ for the conditional expectation $\mathbb{E}\left[Y|X\right]$ of a dependent variable Y relative to a given independent variable X. Kernel regression is a technique to estimate this conditional expectation as a locally weighted average, using kernel as a weighting function.

In this thesis, we will consider the simple and traditional Nadaraya-Watson estimator:

$$\hat{f}_\lambda(x) = \frac{\sum_{i=1}^n K_\lambda(x, x_i) y_i}{\sum_{i=1}^n K_\lambda(x, x_i)},$$

where K is a kernel with a bandwidth h. A kernel determines the intensity of influence one point can exert on other points, while the bandwidth h controls how fast the effect decays in space.

This estimator is known to be consistent when certain technical conditions on the kernel K and the bandwidth λ are satisifed. For the rest of this thesis, we will exclusively consider the Gaussian kernel (also known as the heat kernel):

$$K_\lambda(x, x_i) = e^{-\lambda(x - x_i)^2}.$$

2.2 Blind Regression

We are interested in a certain class of problems, for which each instance of independent variable x_i is distinguishable by identities, but there is no meaningful feature representation. For example, Amazon can distinguish one customer from another from their customer IDs, but their IDs are arbitrary and do not represent their characteristics. In addition, they do not provide any clue on the distance between two users. This makes the traditional regression approach impossible because all the above approaches rely on the geometry of feature representations. In fact, having such a void representation is quite easily observed in real-world data applications.

This challenge mainly arises from the fact that we do not have a metric to measure distance between any two datapoints without having meaningful feature representations. Therefore, we suggest to learn geometry of a latent representation of data from the data themselves. In some applications, such as recommender systems, the notion of similarity between users or between items is already widely used. We propose to unify such heuristic approaches in language of a pseudo-metric in the latent space and kernel regression.

The term 'blind regression' refers to this two-step procedure: 1) estimate the distance between data points, using heuristics if applicable; and 2) make prediction based on a kernel regression estimator. One benefit of this framework is that we can adopt techniques for kernel regression to analyze estimators that can be parsed via lens of blind regression.

In the following chapters, we apply this framework to build and analyze neighbor-based algorithms for matrix completion.

Chapter 3

Application to Matrix Completion

In this chapter, we apply the blind regression framework to the matrix completion problem. We will describe the problem, and our modeling assumptions to obtain provable guarantees in the following chapters. Then we will describe our collaborative filtering algorithm, with two variants which will be analyzed in subsequent chapters. We will also provide some intuition behind the algorithm.

3.1 Model and Notation

3.1.1 Motivation

Our work is motivated by the problem of matrix completion arising in the context of designing recommendation systems. In the popularized setting of the Neflix Challenge, there are m users, indexed by $u \in [m]$, and n movies, indexed by $i \in [n]$. Each user u has a rating for each movie i, denoted as $R(u, i)$. The system observes ratings for only a small fraction of user-movie pairs. The goal is to predict ratings for the rest of the unknown user-movie pairs, i.e., to complete the partially observed $m \times n$ rating matrix. To be able to obtain meaningful predictions from the partially observed matrix, it is essential to impose a structure on the data.

3.1.2 Model and Assumptions

Model. We assume the following data generation process: each user u and movie i is associated to feature representations $x_1(u) \in \mathcal{X}_1$ and $x_2(i) \in \mathcal{X}_2$ for some metric spaces $\mathcal{X}_1, \mathcal{X}_2$. The rating of user u for movie i is given by

$$R(u,i) = f\left(x_1(u), x_2(i)\right),$$

for some function $f : \mathcal{X}_1 \times \mathcal{X}_2 \to \mathbb{R}$.

We observe ratings for $N \ll m \times n$ user-movie pairs, denoted as $\mathcal{D} = \left\{(u^k, i^k, R^k)\right\}_{k=1}^{N}$, where $(u^k, i^k, R^k) \in [m] \times [n] \times \mathbb{R}$. Also, we assume that the measurements are noisy:

$$A(u,i) = R(u,i) + \eta(u,i), \tag{3.1}$$

for $(u, i, R) \in \mathcal{D}$ with noise $\eta(u,i)$. The goal is to use the data $\mathcal{D}$ to predict $R(u,i)$ for all $(u,i) \in [m] \times [n]$ whose rating is unknown.

For brevity, this can be summarized as follows in terms of matrices: we are given an incomplete matrix $A \in \mathbb{R}^{m\times n}$ generated by

$$A = M \circ (R + E).$$

where $\circ$ is the Hadamard product (= entrywise multiplication) and

- The mask matrix M takes either 1 or ∞

$$M(u,i) = \begin{cases} 1 & \text{if } (u,i,R(u,i)) \in \mathcal{D}, \\ \infty, & \text{otherwise.} \end{cases}$$

- Each entry of the noise matrix E represents i.i.d. additive noise

$$E(u,i) = \eta(u,i).$$

Assumptions. We shall make the following assumptions on regularity of the latent spaces $\mathcal{X}_1, \mathcal{X}_2$ (assumptions 1 and 2), the latent function f (assumption 3), the noise η (assumption 4), and the dataset $\mathcal{D}$ (assumption 5).

1. $\mathcal{X}_1$ and $\mathcal{X}_2$ are compact metric (therefore totally bounded, and hence bounded) spaces endowed with metrics d_1 and d_2 respectively:

$$d_1(x_1, x_1') \leq B, \quad \forall x_1, x_1' \in \mathcal{X}_1,$$
$$d_2(x_2, x_2') \leq B, \quad \forall x_2, x_2' \in \mathcal{X}_2.$$

2. Let P_1 and P_2 be Borel probability measures on $(\mathcal{X}_1, T_1)$ and $(\mathcal{X}_2, T_2)$, respectively, where T_i denotes the Borel σ-algebra of $\mathcal{X}_i$ generated by the metric d_i above. We shall assume that the latent features of each user u and movie i, $x_1(u)$ and $x_2(i)$, are drawn i.i.d. from the distribution given by P_1 and P_2 respectively.

3. The latent function $f : \mathcal{X}_1 \times \mathcal{X}_2 \to \mathbb{R}$ is L-Lipschitz with respect to ∞-product metric (see Definition 3 in Appendix A.1) of d_1 and d_2:

$$|f(x_1, x_2) - f(x_1', x_2')| \leq L\left(d_1(x_1, x_1') \vee d_2(x_2, x_2')\right), \quad \forall x_1, x_1' \in \mathcal{X}_1, \ \forall x_2, x_2' \in \mathcal{X}_2.$$

4. The additive noise for all data points are independent and bounded with zero mean and variance γ^2: for all $u \in [n_1], i \in [n_2]$,

$$\eta(u,i) \in [-B_\eta, B_\eta], \quad \mathbb{E}[\eta(u,i)] = 0, \quad \mathrm{Var}[\eta(u,i)] = \gamma^2.$$

5. Rating of each entry is revealed (observed) with probability p, independently. In other words,

$$M(u,i) \sim Bernoulli(p).$$

3.1.3 Notations

We introduce some notations which will be used in later sections.

Index sets: We let $\mathcal{O}_u$ denote the set of column indices of observed entries in row u. Similarly, let $\mathcal{O}_i$ denote the set of row indices of the observed in column i, namely,

$$\begin{aligned} \mathcal{O}_u &:= \{i : M(u,i) = 1\}, \\ \mathcal{O}_i &:= \{u : M(u,i) = 1\}. \end{aligned}$$

For rows $v \neq u$, we define the *overlap* between two rows u and v as

$$\mathcal{O}_{uv} := \mathcal{O}_u \cap \mathcal{O}_v,$$

the set of column indices of commonly observed entries for the pair of rows (u, v). Similarly, the overlap between two columns i and j $(j \neq i)$ is defined as

$$\mathcal{O}_{ij} := \mathcal{O}_i \cap \mathcal{O}_j.$$

β-overlapping neighbors: Given a parameter $\beta \geq 2$, and (u, i), define β-overlapping neighbors of u and i respectively as

$$\begin{aligned} \mathcal{S}_u^\beta(i) &= \{v \ s.t. \ v \in \mathcal{O}_i, \ v \neq u, \ |\mathcal{O}_{uv}| \geq \beta\}, \\ \mathcal{S}_i^\beta(u) &= \{j \ s.t. \ j \in \mathcal{O}_u, \ j \neq i, \ |\mathcal{O}_{ij}| \geq \beta\}. \end{aligned}$$

Empirical statistics: For each $v \in \mathcal{S}_u^\beta(i)$, we can compute the empirical row variance between u and v,

$$s_{uv}^2 := \frac{1}{2\,|\mathcal{O}_{uv}|\,(|\mathcal{O}_{uv}| - 1)} \sum_{i,j \in \mathcal{O}_{uv}} \left((A(u,i) - A(v,i)) - (A(u,j) - A(v,j))\right)^2. \quad (3.2)$$

We can also compute empirical column variances between i and j, for all $j \in \mathcal{S}_i^\beta(u)$,

$$s_{ij}^2 := \frac{1}{2\left|\mathcal{O}_{ij}\right|\left(\left|\mathcal{O}_{ij}\right|-1\right)} \sum_{u,v \in \mathcal{O}_{ij}} \left((A(u,i)-A(u,j))-(A(v,i)-A(v,j))\right)^2. \tag{3.3}$$

In fact, these quantities can be computed in a more traditional manner:

$$\begin{aligned} m_{uv} &:= \frac{1}{\left|\mathcal{O}_{uv}\right|}\left(\sum_{i \in \mathcal{O}_{uv}} A(u,i)-A(v,i)\right), \\ m_{ij} &:= \frac{1}{\left|\mathcal{O}_{ij}\right|}\left(\sum_{u \in \mathcal{O}_{ij}} A(u,i)-A(u,j)\right), \\ s_{uv}^2 &:= \frac{1}{\left|\mathcal{O}_{uv}\right|-1} \sum_{i \in \mathcal{O}_{uv}} \left(A(u,i)-A(v,i)-m_{uv}\right)^2, \\ s_{ij}^2 &:= \frac{1}{\left|\mathcal{O}_{ij}\right|-1} \sum_{u \in \mathcal{O}_{ij}} \left(A(u,i)-A(u,j)-m_{ij}\right)^2. \end{aligned} \tag{3.4}$$

Here, m_{uv} is the row displacement from v to u, which is the sample mean of $A(u,j)-A(v,j)$ for $j \in \mathcal{O}_{uv}$; m_{ij} is the column displacement from j to i.

3.1.4 Performance Metrics

We will introduce three types of performance metrics. Root mean squared error (RMSE), or the Frobenius norm is most traditional. But our analysis based on Chebyshev's inequality doesn't allow a finite provable bound for RMSE.

We define a new metric, the ϵ-risk that we analyze to quantify the performance of our prediction algorithm. Let $E \subset [m] \times [n]$ be the evaluation set which is a subset of unobserved user-movie indices for which the algorithm predicts a rating. Specifically, let $\hat{R}(u,i)$ be the predicted rating while the true (unknown) rating is $R(u,i)$ for $(u,i) \in E$.

- **Root mean squared error (RMSE):** The root mean squared error (RMSE) is defined as follows:

$$RMSE = \sqrt{\frac{1}{|E|}\sum_{(u,i)\in E}\left(R(u,i) - \hat{R}(u,i)\right)^2}.$$

This is a risk with the squared loss, and converges to the L_2 distance between the estimated matrix and the true (unknown) matrix as E approaches to the whole index set. The RMSE is widely accepted as a standard performance metric.

- **Relative squared error (RSE):** Relative squared error is the ratio between the norms of the residual error and the true signal. We can also interpret this as the normalized version of RMSE:

$$RSE = \frac{\left\|Y - \hat{Y}\right\|_F}{\|Y\|_F} = \frac{\sqrt{\frac{1}{|E|}\sum_{(u,i)\in E}\left(R(u,i) - \hat{R}(u,i)\right)^2}}{\sqrt{\frac{1}{|E|}\sum_{(u,i)\in E}|R(u,i)|^2}}.$$

- **ϵ-risk:** For a given error threshold $\epsilon > 0$, we define ϵ-risk of the algorithm as the fraction of the entries for which our estimate has error greater than ϵ:

$$Risk_\epsilon = \frac{1}{|E|}\sum_{(u,i)\in E}\mathbb{I}\left(\left|R(u,i) - \hat{R}(u,i)\right| > \epsilon\right).$$

3.2 Description of the Algorithm

Let $B^\beta(u,i)$ denote the set of positions (v,j) such that the entries $A(v,j)$, $A(u,j)$ and $A(v,i)$ are observed, and the commonly observed ratings between (u,v) and between (i,j) are at least β.

$$B^\beta(u,i) = \left\{(v,j) \in \mathcal{S}_u^\beta(i) \times \mathcal{S}_i^\beta(u) \text{ s.t. } M(v,j) = 1\right\}.$$

Compute the final estimate as a convex combination of estimates derived in (3.6) for $(v, j) \in B^{\beta}(u, i)$,

$$\hat{R}(u,i) = \frac{\sum_{(v,j)\in B^{\beta}(u,i)} w_{ui}(v,j)\left(A(u,j) + A(v,i) - A(v,j)\right)}{\sum_{(v,j)\in B^{\beta}(u,i)} w_{ui}(v,j)}, \tag{3.5}$$

where the weights $w_{ui}(v, j)$ are defined as a function of (3.2) and (3.3). We proceed to discuss a few choices for the weight function, each of which results in a different algorithm.

3.2.1 User-User or Item-Item Nearest Neighbor Weights.

We can evenly distribute the weights only among entries in the nearest neighbor row, i.e., the row with minimal empirical variance,

$$w_{vj} = \mathbb{I}(v = u^*), \text{ for } u^* \in \arg\min_{v \in \mathcal{S}_u^{\beta}(i)} s_{uv}^2.$$

If we substitute these weights in (3.5), we recover an estimate which is asymptotically equivalent to the mean-adjusted variant of the classical user-user nearest neighbor (collaborative filtering) algorithm,

$$\hat{R}(u,i) = A(u^*, i) + m_{uu^*}.$$

Equivalently, we can evenly distribute the weights among entries in the nearest neighbor columns, i.e., the column with minimal empirical variance, recovering the classical mean-adjusted item-item nearest neighbor collaborative filtering algorithm. Theorem 1 proves that this simple algorithm produces a consistent estimator, and we provide the finite sample error analysis. Due to the similarities, our analysis also directly implies the proof of correctness and consistency for the classic user-user and item-item collaborative filtering method.

3.2.2 User-Item Gaussian Kernel Weights.

Inspired by kernel regression, we introduce a variant of the algorithm which computes the weights according to a Gaussian kernel function with bandwith parameter λ, substituting in the minimum row or column sample variance as a proxy for the distance,

$$w_{vj} = \exp(-\lambda \min\{s_{uv}^2, s_{ij}^2\}).$$

When $\lambda = \infty$, the estimate only depends on the basic estimates whose row or column has the minimum sample variance. When $\lambda = 0$, the algorithm equally averages all basic estimates. We applied this variant of our algorithm to both movie recommendation and image inpainting data, which show that our algorithm improves upon user-user and item-item classical collaborative filtering.

3.2.3 Some Intuition

Connection to Taylor Series Approximation

Our prediction algorithm for unknown ratings is inspired by the classical Taylor approximation of a function. Suppose $\mathcal{X}_1 \cong \mathcal{X}_2 \cong \mathbb{R}$, and we wish to predict unknown rating, $f(x_1(u), x_2(i))$, of user $u \in [m]$ for movie $i \in [n]$. Using the first order Taylor expansion of f around $(x_1(v), x_2(j))$ for some $u \neq v \in [m], i \neq j \in [n]$, it follows that

$$\begin{aligned} f(x_1(u), x_2(i)) \approx{}& f(x_1(v), x_2(j)) + \\ &+ \frac{\partial f(x_1(v), x_2(j))}{\partial x_1}(x_1(u) - x_1(v)) + \frac{\partial f(x_1(v), x_2(j))}{\partial x_2}(x_2(i) - x_2(j)). \end{aligned}$$

We are not able to directly compute this expression, as we do not know the latent features, the function f, or the partial derivatives of f. However, we can again apply Taylor series expansion for $f(x_1(v), x_2(i))$ and $f(x_1(u), x_2(j))$ around $(x_1(v), x_2(j))$, which results in a set of equations with the same unknown terms. It follows from rearranging terms and substitution that

$$f(x_1(u), x_2(i)) \approx f(x_1(v), x_2(i)) + f(x_1(u), x_2(j)) - f(x_1(v), x_2(j)),$$

as long as the first order approximation is accurate. Thus if the noise term in (3.1) is small, we can approximate $f(x_1(u), x_2(i))$ by using observed ratings $A(v, j)$, $A(u, j)$ and $A(v, i)$ according to

$$\hat{R}(u, i) = A(u, j) + A(v, i) - A(v, j). \tag{3.6}$$

Connection to Kernel Regression

Once basic estimates are obtained, our algorithm computes both the row and column sample variance, and uses the minimum of the two as the reliability of the estimate. We weight each estimate using a Gaussian kernel computed on the sample variance with the kernel bandwidth parameter λ , which is$\exp(-\lambda \min\{(s^{uv})^2, s_{ij}^2\})$. When $\lambda = \infty$, the estimate only depends on the basic estimates from the row or column which has the minimum sample variance. When $\lambda = 0$, the algorithm equally weights all basic estimates and takes simple average. The final estimate as a function of λ and β is computed to be a Nadaraya-Watson estimator with distance proxy.

Reliability of Local Estimates: We will show that the variance of the difference between two rows or columns upper bounds the estimation error. Therefore, in order to ensure the accuracy of the above estimate, we use empirical observations to estimate the variance of the difference between two rows or columns, which directly relates to an error bound. By expanding (3.6) according to (3.1), the error $R(u, i) - \hat{R}(u, i)$ is equal to

$$Error = (R(u, i) - R(v, i)) - (R(u, j) - R(v, j)) - \eta_{vi} + \eta_{vj} - \eta_{uj}.$$

If we condition on $x_1(u)$ and $x_1(v)$,

$$\mathbb{E}\left[(\text{Error})^2 \mid x_1(u), x_1(v)\right] = 2\ Var_{\mathbf{x}\sim P_2}\left[f(x_1(u),\mathbf{x}) - f(x_1(v),\mathbf{x}) \mid x_1(u), x_1(v)\right] + 3\gamma^2.$$

Similarly, if we condition on $x_2(i)$ and $x_2(j)$ it follows that the expected squared error is bounded by the variance of the difference between the ratings of columns i and j. This theoretically motivates weighting the estimates according to the variance of the difference between the rows or columns.

Connections to Cosine Similarity Weights: In our algorithm, we determine reliability of estimates as a function of the sample variance, which is equivalent to the squared distance of the mean-adjusted values. In classical collaborative filtering, cosine similarity is commonly used, which can be approximated as a different choice of the weight kernel over the squared difference. In other words, our blind regression framework subsumes collaborative filtering with cosine similarity as another variant for which $w_{ui}(v,j)$ is determined by the cosine kernel.

Chapter 4

Main Theorems

In this chapter, we state our main results for the algorithm presented in Chapter 3. Theorem 1 argues that the nearest neighbor algorithm is consistent when there is no noise. With presence of the noise, our error bound for the expected ϵ-risk will converge to the Chebyshev bound for the noise, which is the optimal achevable bound without imposing further structural assumptions. The proof of the main theorem depends on the lemmas in Chapter 5, whose proofs hinge on concentration inequalities and regularity assumptions on the latent feature space. We also discuss about extending our algorithm and analysis to tensor completion of higher order by simple flattening method. In Section 4.3, we provide a preliminary discussion for the effect of the parameter λ in more general Gaussian kernel variant of the proposed algorithm.

4.1 Consistency of the Nearest Neighbor Algorithm

Recall that for $\epsilon > 0$, we defined in Section 3.1.4 the overall ϵ-risk of the algorithm as the fraction of estimates whose error is larger than ϵ

$$Risk_\epsilon = \frac{1}{|E|} \sum_{(u,i)\in E} \mathbb{I}\left(\left|R(u,i) - \hat{R}(u,i)\right| > \epsilon\right),$$

where $E \subset [m]\times[n]$ denote the set of user-movie pairs for which the algorithm predicts a rating.

Theorem 1 provides an upper bound for the expected ϵ-risk of the nearest neighbor version of our algorithm. It proves the nearest neighbor estimator is consistent, in the presence of no noise, which means the estimates converge to the true values as $m, n \to \infty$. We may assume $m \leq n$ without loss of generality.

Theorem 1 (Consistency of the Nearest-neighbor version)**.** *For a fixed $\epsilon > 0$, as long as $p \geq \max\{m^{-1+\delta}, n^{-1/2+\delta}\}$ (where $\delta > 0$), for any $\rho > 0$, the user-user nearest-neighbor variant of our method with $\beta = np^2/2$ achieves*

$$\mathbb{E}\left[Risk_\epsilon\right] \leq \frac{3\rho + \gamma^2}{\epsilon^2}\left(1 + \frac{3 \cdot 2^{1/3}}{\epsilon} n^{-\frac{2}{3}\delta}\right) + O\left(\exp\left(-\frac{1}{4}Cm^\delta\right) + m^\delta \exp\left(-\frac{1}{5B^2}n^{\frac{2}{3}\delta}\right)\right),$$

where $B = 2(LB_{\mathcal{X}} + B_\eta)$, and $C = h\left(\sqrt{\frac{\rho}{L^2}}\right) \wedge \frac{1}{6}$ for $h(r) := ess\inf_{x_0 \in \mathcal{X}_1} \mathbb{P}_{\mathbf{x} \sim P_{\mathcal{X}_1}}\left(d(\mathbf{x}, x_0) \leq r\right)$.

For a generic β, we can also provide precise error bounds of a similar form, with modified rates of convergence. Choosing β to grow with np^2 ensures that as n goes to infinity, the required overlap between rows also goes to infinity, thus the empirical mean and variance computed in the algorithm converge precisely to the true mean and variance. The parameter ρ in Theorem 1 is introduced purely for the purpose of analysis, and is not used within the implementation of the the algorithm.

The function h behaves as the cumulative distribution function of $P_{\mathcal{X}_1}$, and it always exists under our assumptions that $\mathcal{X}_1$ is compact (see Section 5.2 for more detail). It is used to ensure that for any $u \in [m]$, with high probability, there exists another row $v \in \mathcal{S}_u^\beta(i)$ such that $d_{\mathcal{X}_1}(x_1(u), x_1(v))$ is small, implying that we can use the values of row v to approximate the values of row u well. For example, if $P_{\mathcal{X}_1}$ is a uniform distribution over a unit cube in q dimensional Euclidean space, then $h(r) = \min(1, r)^q$, and our error bound becomes meaningful for $n \geq (L^2/\rho)^{q/2\delta}$. On the other hand, if $P_{\mathcal{X}_1}$ is supported over finitely many points, then $h(r) = \min_{\mathbf{x} \in \text{supp}(P_{\mathcal{X}_1})} P_{\mathcal{X}_1}(\mathbf{x})$ is a positive constant, and the role of the latent dimension becomes irrelevant, allowing us to extend the theorem to $\rho = 0$. Intuitively, the "geometry" of $P_{\mathcal{X}_1}$ through h near 0 determines the impact of the latent space dimension on the sample complexity, and our results hold as long as the latent dimension $q = o(\log n)$.

4.1.1 Proof of Theorem 1

In this section, we will prove Theorem 1. From the definition of $Risk_\epsilon$, it follows that for any evaluation set of unobserved entries E, the expectation of ϵ-risk is

$$\begin{aligned}\mathbb{E}\left[Risk_\epsilon\right] &= \mathbb{E}\left[\frac{1}{|E|}\sum_{(u,i)\in E}\mathbb{I}\left(\left|R(u,i)-\hat{R}(u,i)\right|>\epsilon\right)\right]\\ &= \frac{1}{|E|}\sum_{(u,i)\in E}\mathbb{E}\left[\mathbb{I}\left(\left|R(u,i)-\hat{R}(u,i)\right|>\epsilon\right)\right]\\ &= \mathbb{P}\left(\left|R(u,i)-\hat{R}(u,i)\right|>\epsilon\right),\end{aligned}$$

because the indexing of the engries are exchangeable and identically distribued. Therefore, in order to bound the expected risk, it suffices to upper bound the probability $\mathbb{P}\left(\left|R(u,i)-\hat{R}(u,i)\right|>\epsilon\right)$ to prove the theorem.

Proof. For any fixed $a, b \in \mathcal{X}_1$, and random variable $\mathbf{x} \sim P_{\mathcal{X}_2}$, we denote the mean and variance of the difference $f(a,\mathbf{x}) - f(b,\mathbf{x})$ by

$$\begin{aligned}\mu_{ab} &\triangleq \mathbb{E}_{\mathbf{x}}[f(a,\mathbf{x}) - f(b,\mathbf{x})]\\ \sigma_{ab}^2 &\triangleq \mathrm{Var}_{\mathbf{x}}[f(a,\mathbf{x}) - f(b,\mathbf{x})].\end{aligned}$$

These are also equivalent to the expectation of the empirical means and variances computed by the algorithm when we condition on the latent representations of the users, i.e.

$$\mathbb{E}\left[m_{uv}|\mathbf{x}_1(u)=a,\mathbf{x}_1(v)=b\right]=\mu_{ab},\ \text{and}\ \mathbb{E}\left[s_{uv}^2|\mathbf{x}_1(u)=a,\mathbf{x}_1(v)=b\right]=\sigma_{ab}^2.$$

The computation of $\hat{R}(u,i)$ involves two steps: first the algorithm determines the neighboring row with the minimum sample variance, $u^* = \arg\min_{v\in\mathcal{S}_u^\beta(i)} s_{uv}^2$, and then it computes the estimate by adjusting according to the empirical mean, $\hat{R}(u,i) := A(u^*,i) + m_{uu^*}$.

The proof involves three key steps, each of which is stated as a separate Lemma.

Lemma 1 proves that with high probability the observations are dense enough such that there is sufficient number of rows with overlap of entries larger than β. Precisely, the number of the candidate rows, $|\mathcal{S}_u^\beta(i)|$, concentrates around $(m-1)p$. This relies on concentration of Binomial random variables via Chernoff's bound.

Lemma 2 proves that due to the assumption that the latent features are sampled iid from a bounded metric space, for any index pair (u,i), there exists "good" neighboring row $v \in \mathcal{S}_u^\beta(i)$, whose true variance $\sigma^2_{x_1(u)x_1(v)}$ is small. In the process, we use the function $h(\cdot)$ which satisfies

$$P_1\left(\mathbf{x} \in B(x_0, r)\right) \geq h(r), \quad \forall x_0 \in \mathcal{X}_1, r > 0,$$

where $B(x_0,r) \triangleq \{x \in \mathcal{X}_1 \ s.t. \ d_{\mathcal{X}_1}(x, x_0) \leq r\}$. Discussion about existence of such functions for essentially all probability distributions is discussed in Section 5.2.

Subsequently, conditioned on the event that $\left|\mathcal{S}_u^\beta(i)\right| \approx (m-1)p$, Lemmas 4 and 6 prove that the sample mean and sample variance of the differences between two rows concentrate around the true mean and true variance with high probability. This involves using the Lipschitz and bounded assumptions on f and $\mathcal{X}_1$, as well as the Bernstein and Maurer-Pontil inequalities.

Given that there exists a neighbor $v \in \mathcal{S}_u^\beta(i)$ whose true variance $\sigma^2_{x_1(u)x_1(v)}$ is small, and conditioned on the event that all the sample variances concentrate around the true variance, it follows that the true variance between u and its nearest neighbor u^* is small with high probability. Finally, conditioned on the event that $|\mathcal{S}_u^\beta(i)| \approx (m-1)p$ and the true variance between the target row and the nearest neighbor row is small, we provide a bound on the tail probability of the estimation error by using Chevyshev inequalities. The only term in the error probability which does not decay to zero is the error from Chebyshev's inequality, which dominates the final expression, thus leading to the desired result.

For readability, we define the following events: with $\beta = np^2/2$,

- Let A denote the event that $|\mathcal{S}_u^\beta(i)| \in [(m-1)p/2, 3(m-1)p/2]$.

- Let B denote the event that $\min_{v \in \mathcal{S}_u^\beta(i)} \sigma^2_{x_1(u)x_1(v)} < \rho$.

- Let C denote the event that $\left|\mu_{x_1(u)x_1(v)} - m_{uv}\right| < \alpha$ for all $v \in \mathcal{S}_u^\beta(i)$.

- Let D denote the event that $\left|s_{uv}^2 - (\sigma^2_{x_1(u)x_1(v)} + 2\gamma^2)\right| < \rho$ for all $v \in \mathcal{S}_u^\beta(i)$.

Consider the following:

$$\begin{aligned}
&\mathbb{P}\Big(\, |R(u,i) - \hat{R}(u,i)| > \epsilon \, \Big) \\
&\leq \mathbb{P}\Big(\, |R(u,i) - \hat{R}(u,i)| > \epsilon \,|\, A, B, C, D\Big) \\
&\qquad\qquad + \mathbb{P}\left(A^c\right) + \mathbb{P}\left(B^c|A\right) + \mathbb{P}\left(C^c|A,B\right) + \mathbb{P}\left(D^c|A,B,C\right). \qquad (4.1)
\end{aligned}$$

Now,

$$\begin{aligned}
\mathbb{P}\left(A^c\right) &= \mathbb{P}\left(|\mathcal{S}_u^\beta(i)| \notin \left[\frac{(m-1)p}{2}, \frac{3(m-1)p}{2}\right]\right) \\
&\leq 2\exp\left(-\frac{(m-1)p}{12}\right) + (m-1)\exp\left(-\frac{np^2}{8}\right), \qquad (4.2)
\end{aligned}$$

using Lemma 1. Similarly, using Lemma 2

$$\begin{aligned}
\mathbb{P}\left(B^c|A\right) &\leq \left(1 - h\left(\sqrt{\frac{\rho}{L^2}}\right)\right)^{\frac{(m-1)p}{2}} \\
&\leq \ \exp\left(-\frac{(m-1)p\ h\left(\sqrt{\frac{\rho}{L^2}}\right)}{2}\right). \qquad (4.3)
\end{aligned}$$

Given choice of parameters, i.e. choice of m and p large enough for a given ρ, as we shall argue, the right hand side of (4.3) will be going to 0, and hence definitely less than 1/2. That is, $\mathbb{P}\left(B|A\right) \geq 1/2$. Using this fact and Bayes formula, we have

$$\begin{aligned}
\mathbb{P}\left(C^c|A,B\right) &\leq 2\mathbb{P}\left(C^c|A\right) \\
&= 2\mathbb{P}\left(\bigcup_{v \in \mathcal{S}_u^\beta(i)} \left\{\left|\mu_{x_1(u)x_1(v)} - m_{uv}\right| > \alpha\right\} \,\middle|\, A\right) \\
&\leq 3(m-1)p\exp\left(-\frac{3np^2\alpha^2}{12B^2 + 4B\alpha}\right), \qquad (4.4)
\end{aligned}$$

where last inequality follows from union bound, Lemmas 4 and choice of $\beta = np^2/2$.

Again, choice of parameters, i.e. m, n, p and α will be such that we will have the right hand side of (4.4) going to 0 and definitely less than 1/8. Using this and arguments as used above based on Bayes' formula, we bound

$$\begin{aligned}
\mathbb{P}\left(D^c|A,B,C\right) &\leq \frac{\mathbb{P}\left(D^c|A\right)}{\mathbb{P}\left(B|A\right)\mathbb{P}\left(C|A,B\right)} \\
&\leq \ 4\mathbb{P}\left(D^c|A\right). \\
&= 4\mathbb{P}\left(\bigcup_{v\in\mathcal{S}_u^\beta(i)} \left\{ \left| s_{uv}^2 - \left(\sigma_{x_1(u)x_1(v)}^2 + 2\gamma^2\right)\right| > \rho \right\} \middle| \ A \right) \\
&\leq 12(m-1)p\exp\left(-\frac{\beta\rho^2}{4B^2(2LB_{\mathcal{X}}^2 + 4\gamma^2 + \rho)}\right), \qquad (4.5)
\end{aligned}$$

where last inequality follows from union bound and Lemma 6.

Finally, with the choice of $\alpha = \beta^{-1/3}$, which is $\left(\frac{np^2}{2}\right)^{-1/3}$ since $\beta = \frac{np^2}{2}$, using Lemma 7, we obtain that

$$\begin{aligned}
\mathbb{P}\left(\ |f(x_1(u), x_2(i)) - \hat{y}(u,i)| > \epsilon \,|\, A,B,C,D\right) &\leq \frac{3\rho+\gamma^2}{\epsilon^2}\left(1-\frac{\alpha}{\epsilon}\right)^{-2} \\
&\leq \frac{3\rho+\gamma^2}{\epsilon^2}\left(1+\frac{3\alpha}{\epsilon}\right). \qquad (4.6)
\end{aligned}$$

where we have used the fact that for given choice of α (since ϵ is fixed), as m increases, the term α/ϵ becomes less than 1/5; for $x \leq 1/5$, $(1-x)^{-2} \leq (1+3x)$.

If $p = \omega(m^{-1})$ and $p = \omega(n^{-1/2})$, all error terms from (4.2) to (4.5) diminish to 0 as $m, n \to \infty$. Specifically, if we choose $p = \max(m^{-1+\delta}, n^{-1/2+\delta})$, then putting everything together, we obtain (we assume that $m/2 \leq m-1 \leq m$)

$$\begin{aligned}
&\mathbb{P}\left(|R(u,i) - \hat{R}(u,i)| > \epsilon\right) \\
&\quad \leq \frac{3\rho+\gamma^2}{\epsilon^2}\left(1 + \frac{3\sqrt[3]{2}}{\epsilon}n^{-\frac{2}{3}\delta}\right) + 2\exp\left(-\frac{1}{24}m^\delta\right) + m\exp\left(-\frac{1}{8}n^{2\delta}\right) \\
&\qquad + \exp\left(-\frac{1}{4}h\left(\sqrt{\frac{\rho}{L^2}}\right)m^\delta\right) + 3m^\delta\exp\left(-\frac{1}{5B^2}n^{\frac{2}{3}\delta}\right) \\
&\qquad + 12m^\delta\exp\left(-\frac{\rho^2}{8B^2(2LB_{\mathcal{X}}^2 + 4\gamma^2+\rho)}n^{2\delta}\right).
\end{aligned}$$

The above bound holds for any $\rho > 0$, though as $\rho \to 0$, m, n also need to increase accordingly such that $h\left(\sqrt{\frac{\rho}{L^2}}\right)$ is not too small. When the support of $P_{\mathcal{X}}$ is finite, then

$$h\left(\sqrt{\frac{\rho}{L^2}}\right) \geq \min_{x \in \mathcal{X}} P_{\mathcal{X}}(x),$$

such that the above bound holds even when $\rho = 0$. □

4.2 Tensor Completion by Flattening

A tensor is a higher order analog of a matrix. As a matrix represents interactions between two entities (e.g. users and movies), higher order tensors represent interactions between more than two entries. As a result, tensors can yield more appropriate and flexible models for reality in some situations. For example, we may have time series data for each user-movie rating, such that the completion problem concerns predicting an unknown user-movie rating at a given time instance. Then the completion problem concerns predicting the unknown rating of a user for a movie at a given time instance. However, a tensor completion problem is known to be much harder than a matrix completion. Recently, specific tensor decomposition approaches have been suggested for tensor completion, but there is still little understanding on the problem.

One naïve approach toward tensor completion is to simply consider it as a matrix completion via flattening. Due to the mildness and simplicity of our assumptions, we can easily reduce a tensor to an appropriate matrix problem which our algorithm and analysis can solve. Let $T \in \mathbb{R}^{k_1 \times k_2 \times \dots k_t}$ denote a t-order tensor. Assume that the equivalent higher order assumptions presented in Section 3.1.2 hold, in particular that the indices are associated with latent features drawn according to a probability measure over a compact metric space, and that the observed values can be described by a Lipschitz function over the latent spaces with an independent bounded zero-mean additive noise.

We consider the higher order extension of the same assumptions as presented in Section 3.1.2. Assume that for each dimension $r \in [t]$, each index $i \in [k_t]$ is associated to a latent feature $z_r(i)$ which is sampled according to a probability measure $P_{\mathcal{Z}_r}$ over a compact metric space $\mathcal{Z}_r$. Assume that there exists some L-Lipschitz function $g : \prod_{r\in[t]} \mathcal{Z}_r \to \mathbb{R}$ which relates latent features to the observed values, such that for $\mathbf{i} = (i_1, i_2, \ldots i_t) \in [k_1] \times [k_2] \times \ldots [k_t]$, $T(\mathbf{i}) = g(z_1(i_1), z_1(i_2), \ldots z_t(i_t)) + \eta_{\mathbf{i}}$, where $\eta_{\mathbf{i}}$ is some independent bounded zero-mean noise.

Let $(\mathcal{T}_1, \mathcal{T}_2)$ be a disjoint partition of $[t]$ such that $\mathcal{T}_1 \cup \mathcal{T}_2 = [t]$. Then we can reduce the tensor to a matrix A_T by "flattening" or combining all dimensions in $\mathcal{T}_1$ as the rows of the matrix, and similarly combining dimensions in $\mathcal{T}_2$ as the columns of the matrix, such that the flattened matrix has dimensions $m \times n$ where $m = \prod_{r\in\mathcal{T}_1} k_r$ and $n = \prod_{r\in\mathcal{T}_2} k_r$. The latent spaces of the matrix are defined as the product spaces over the corresponding latent spaces of the tensor. Similarly the probability measures $P_{\mathcal{X}_1}$ and $P_{\mathcal{X}_2}$ are defined according to the appropriate product measures.

The matrix we constructed satisfies all assumptions required in Section 3.1.2, therefore we can proceed to apply our algorithm and analysis to predict the missing entries.When reducing a general t-order tensor to a matrix, it is desirable to balance the size of the two partitions so that $m \approx \sqrt{n}$ in order to achieve the best sample complexity. For a specific setting in which the dimensions of the tensor are equivalent (i.e. identical latent spaces, probability measures, and number of sampled indices), Theorem 2 presents error bounds for our tensor completion method, derived from Theorem 1.

Theorem 2. *For a t-order tensor $T \in \mathbb{R}^{k^t}$, given any partition $(\mathcal{T}_1, \mathcal{T}_2)$ of $[t]$ such that $|\mathcal{T}_1| = t/3$ and $|\mathcal{T}_2| = 2t/3$, let A_T denote the equivalent matrix which results from flattening the tensor according to the partitioning $(\mathcal{T}_1, \mathcal{T}_2)$. For a fixed $\epsilon > 0$, as long as $p \geq k^{-t/3+\delta}$ (where $\delta > 0$), for any $\rho > 0$, the user-user nearest-neighbor variant of our method applied to matrix A_T with parameter $\beta = k^{2t/3}p^2/2$ achieves*

$$\mathbb{E}\left[\mathrm{Risk}_\epsilon\right] \leq \frac{3\rho+\gamma^2}{\epsilon^2}\left(1+\frac{3\cdot 2^{1/3}}{\epsilon}k^{-\frac{2}{3}\delta}\right) + O\left(\exp\left(-\frac{1}{4}Ck^{\delta}\right) + k^{\delta}\exp\left(-\frac{1}{5B^2}k^{\frac{2}{3}\delta}\right)\right),$$

where $B = 2(LB_{\mathcal{Z}}+B_\eta)$, and $C = h\left(\sqrt{\frac{\rho}{2L^2}}\right)^{t/3}\wedge\frac{1}{6}$ for $h(r) := ess\inf_{z_0\in\mathcal{Z}} \mathbb{P}_{\mathbf{z}\sim P_{\mathcal{Z}}}\left(d_{\mathcal{Z}}(\mathbf{z}, z_0) \leq r\right)$.

4.3 The Effects of Averaging Estimates over Rows: Brief Discussion

Unlike the nearest neighbor algorithm, it is hard to obtain an error bound for the algorithm with general kernel weights. In this section, we will consider an intermediate between the nearest neighbor algorithm and the general kernel algorithm. Define row-collaborative algorithm by taking weights as

$$w_{vj} = \exp(-\lambda s_{uv}^2).$$

Alternatively, if we let the estimator for (u, i) based on the row v be

$$\hat{R}_v(u,i) = \frac{1}{|\mathcal{O}_{uv}|} \sum_{j \in \mathcal{O}_{uv}} \left[A(u,j) + A(v,i) - A(v,j) \right],$$

then the row-averaged estimator can be written as a weighted average of these:

$$\hat{R}(u,i) = \frac{\sum_{v \in \mathcal{S}_u^\beta(i)} c_v \hat{R}_v(u,i)}{\sum_{v \in \mathcal{S}_u^\beta(i)} c_v}, \tag{4.7}$$

where $c_v = \exp\left(-\lambda s_{uv}^2\right)$.

In fact, it is not easy to obtain an error bound even for the row-collaborative algorithm. Nevertheless, the analysis on it can help to understand trade-off between the increase in signal variance and the decrease in noise variance as λ changes. In this section, we briefly discuss the effect of averaging and the role of kernel parameter λ with some calculations for the row-collaborative algorithm, thereby gaining insights on the influence of averaging on the Chebyshev bound in Theorem 1. To simplify calculations, we will impose a set of strong assumptions on the latent space and the latent function throughout this section, which will help us appreciate the essential effects of averaging:

1. Consider $\mathbb{R}^d$ and the standard Gaussian measure $\gamma_{0,1}^d$ on it. For any $\epsilon > 0$, we can find $R < \infty$ such that $\gamma_{0,1}^d\left(B(0,R)\right) > 1 - \epsilon$.

2. Let $\mathcal{X}_1 = B(0, R) \subset \mathbb{R}^d$, and $P_{\mathcal{X}_1}$ is a uniform probability measure on $\mathcal{X}_1$.

3. Suppose that f and $(\mathcal{X}_2, P_{\mathcal{X}_2})$ satisfies that for any $a, b \in (\mathcal{X}_1, d_1)$,
$$d_1(a, b) = Var\left[f(a, \mathbf{x}_2) - f(b, \mathbf{x}_2)\right].$$

4. Consider $c_v := \exp\left(-\lambda s_{uv}^2\right) \to \exp\left(-\lambda(\sigma_{uv}^2 + 2\gamma^2)\right)$ for a fixed parameter λ.

The following analyses show that varying λ affects the signal variance and the noise variance in the opposite directions. As λ decreases, the algorithm becomes to count more on neighbors farther away, thereby the signal variance enlarges. On the other hand, as independent noises cancel each other, the noise variance diminishes from γ^2 to 0. Therefore, we can expect a trade-off between these two effects. We can interpret the parameter λ as the inverse temperature $\frac{1}{T}$: as temperature increases the thermal noises cancel out in expectation, however, when the system freezes as $T \to 0$, one single row dominates with noise γ^2.

4.3.1 Rough Signal Analysis

Since the noise is independent of the structured signal, we can analyze the variance of signals and noises separately. Although we do not have information on the covariance between row estimators, we can provide an upper bound from the following observation: $|Cov(X_1, X_2)| \leq \sqrt{Var(X_1)Var(X_2)}$ and hence,
$$Var(\sum_{i=1}^{n} c_i X_i) = \sum_{i=1}^{n}\sum_{j=1}^{n} c_i c_j Cov(X_i, X_j) \leq \left(\sum_{i=1}^{n} c_i \sqrt{Var(X_i)}\right)^2.$$

Under the simplifying assumptions, the combined variance
$$\frac{\left(\sum_v c_v \sigma_{uv}\right)^2}{\left(\sum_v c_v\right)^2} = \frac{\left(\sum_v e^{-\lambda s_{uv}^2} \sigma^{uv}\right)^2}{\left(\sum_v e^{-\lambda s_{uv}^2}\right)^2}. \tag{4.8}$$

Moreover, because $\sigma_{uv}^2 = d_1\left(x_1(u), x_1(v)\right)^2$ and $s_{uv}^2 \to \sigma_{uv}^2 + 2\gamma^2$,

$$\sum_v e^{-\lambda s_{uv}^2}\sigma_{uv} \to \frac{1}{Z}\int_0^\infty e^{-\lambda r^2} r\left(S_{d-1}(r)dr\right), \tag{4.9}$$

$$\sum_v e^{-\lambda s_{uv}^2} \to \frac{1}{Z}\int_0^\infty e^{-\lambda r^2}\left(S_{d-1}(r)dr\right), \tag{4.10}$$

as $m, n \to \infty$ and $\epsilon \to 0$. Here, Z is a normalization constant, and $S_{d-1}(r)$ is the surface area of $(d-1)$-sphere:

$$S_{d-1}(r) = \frac{2\pi^{d/2}}{\Gamma\left(\frac{d}{2}\right)} r^{d-1}.$$

It remains to compute the ratio of two integrals $F_d(\lambda)/F_{d-1}(\lambda)$, where

$$F_n(\lambda) = \frac{1}{Z}\frac{2\pi^{d/2}}{\Gamma\left(\frac{d}{2}\right)}\int_0^\infty e^{-\lambda r^2} r^n dr.$$

Then by Equations 4.8, 4.9, 4.10, it follows that

$$\frac{\left(\sum_v c_v \sigma_{uv}\right)^2}{\left(\sum_v c_v\right)^2} \lesssim \left(\frac{F_d(\lambda)}{F_{d-1}(\lambda)}\right)^2. \tag{4.11}$$

Integration by substitution yields

$$\int_0^\infty e^{-\lambda r^2} dr = \sqrt{\frac{\pi}{\lambda}},$$
$$\int_0^\infty e^{-\lambda r^2} r dr = \frac{1}{2\lambda}.$$

Now, we can obtain the following recursive formula from integration by parts:

$$\begin{aligned}\int_0^\infty e^{-\lambda r^2} r^d dr &= \frac{1}{d+1} e^{-\lambda r^2} r^{d+1}\Big|_0^\infty + \frac{2\lambda}{d+1}\int_0^\infty e^{-\lambda r^2} r^{d+2} dr \\ &= \frac{2\lambda}{d+1}\int_0^\infty e^{-\lambda r^2} r^{d+2} dr.\end{aligned}$$

We can conclude that

$$
\begin{aligned}
\frac{F_d(\lambda)}{F_{d-1}(\lambda)} &= \frac{d}{2\lambda}\frac{F_{d-2}(\lambda)}{F_{d-1}(\lambda)} = \frac{d}{2\lambda}\frac{2\lambda}{d-1}\frac{F_{d-2}(\lambda)}{F_{d-3}(\lambda)} = \dots \\
&= \begin{cases} \frac{(d-1)(d-3)\dots 2}{(d-2)(d-4)\dots 2}\frac{F_1(\lambda)}{F_0(\lambda)} & \textit{when } d \textit{ is odd,} \\ \frac{(d-1)(d-3)\dots 1}{(d-2)\dots 2}\frac{F_0(\lambda)}{2\lambda F_1(\lambda)} & \textit{when } d \textit{ is even.} \end{cases} \\
&= \begin{cases} \frac{(d-1)(d-3)\dots 2}{(d-2)(d-4)\dots 2}\frac{1}{2\sqrt{\pi\lambda}} & \textit{when } d \textit{ is odd,} \\ \frac{(d-1)(d-3)\dots 1}{(d-2)\dots 2}\sqrt{\frac{\pi}{\lambda}} & \textit{when } d \textit{ is even.} \end{cases} \\
&\leq \begin{cases} \sqrt{\frac{d}{4\pi\lambda}} & \textit{when } d \textit{ is odd,} \\ \sqrt{\frac{2d\pi}{\lambda}} & \textit{when } d \textit{ is even.} \end{cases}
\end{aligned}
$$

Plugging this into Equation 4.11 gives us

$$
\frac{\left(\sum_v c_v \sigma_{uv}\right)^2}{\left(\sum_v c_v\right)^2} \lesssim c\frac{d}{\lambda}.
$$

This asymptotic inequality suggests that this variance diminishes to 0 as $\lambda \to \infty$. It is an expected result because the algorithm itself shrinks to the nearest neighbor algorithm. Moreover, this bound also suggests that the signal variance is roughly less than or proportional to the dimension of the latent space.

4.3.2 Noise Analysis

The IID assumption on the noise makes the noise analysis much simpler than the signal analysis. In contrast to the signal variance, the noise variance decreases as $\lambda \to 0$:

$$
\lim_{\lambda \to 0} \frac{\sum_v c_v^2}{\left(\sum_v c_v\right)^2}\gamma^2 = \lim_{\lambda \to 0} \frac{e^{-2\lambda\gamma^2}\sum_v e^{-2\lambda\sigma_{uv}^2}}{e^{-2\lambda\gamma^2}\left(\sum_v e^{-\lambda\sigma_{uv}^2}\right)}\gamma^2 = \frac{\gamma^2}{\left|\mathcal{S}_u^\beta(i)\right|} \to 0,
$$

as $m, n \to \infty$. On the other hand, $\frac{\sum_v c_v^2}{\left(\sum_v c_v\right)^2}\gamma^2 \to \gamma^2$ as $\lambda \to \infty$.

Chapter 5

Useful Lemmas and their Proofs

In this chapter, we present five main lemmas (1, 2, 4, 6, 7) used in the proof of Theorem 1. Lemma 1 ensures that for each unknown entry (u, i), there is a sufficiently large number of rows and columns which have observed overlapped entries with high probability. Lemma 2 guarantees there exists a good neighbor with high probability whenever the size of the matrix is sufficiently large and the latent space is compact. Also, Lemmas 4 and 6 ascertain the sample statistics (mean and variance) concentrate to the population statistics; therefore, we can use the sample statistics in our estimators as surrogates for the population statistics. Lastly, we show an upper bound for the conditional error of our nearest-neighbor estimator. Combining all these results by the union bound proves Theorem 1; the detailed proof can be found in Section 4.1.1. The error bounds in each of these lemmas are obtained by applying various concentration inequalities.

5.1 Sufficient Overlap

In this section we show that there exists a a neighbor for a given target $(u, i) \in [m] \times [n]$ with high probability. A sufficiently many, yet still vanishing number of observations are required for that purpose. In fact, Lemma 1 implies an even stronger result. For a given (u, i), not only will there exist a feasible base row, but also the number of those rows concentrates to $(m-1)p$ with high probability. Moreover, we can show that every pair $(u, i) \in [m] \times [n]$ has roughly $(m-1)p$ neighbors (i.e. every estimate is defined) with high probability.

Lemma 1. *Given $p > 0$, $2 \leq \beta \leq \frac{np^2}{2}$, and $\alpha > 0$, for any $(u, i) \in [m] \times [n]$,*

$$\begin{aligned}\mathbb{P}\left(\left|\mathcal{S}_u^\beta(i)\right| \notin (1 \pm \alpha)(m-1)p\right) &\leq (m-1)\exp\left(-\frac{np^2}{2}\left(1 - \frac{\beta}{np^2}\right)^2\right) + 2\exp\left(-\frac{\alpha^2}{3}(m-1)p\right) \\ &\leq (m-1)\exp\left(-\frac{np^2}{8}\right) + 2\exp\left(-\frac{\alpha^2}{3}(m-1)p\right).\end{aligned}$$

Proof. First of all, we can observe that having 1) $|\mathcal{O}_{uv}| \geq \beta$ for every $v \in [m] \setminus u$, and 2) $(1-\alpha)(m-1)p \leq |\mathcal{O}_i| \leq (1+\alpha)(m-1)p$, is a sufficient condition for $(1-\alpha)(m-1)p \leq \left|\mathcal{S}_u^\beta(i)\right| \leq (1+\alpha)(m-1)p$. Our goal is to provide a lower bound on the probability of this event, by showing an upper bound on its complement (i.e., 'failure' probability). Given that the test 1) fails with probability at most ϵ_1, and 2) fails with no greater than ϵ_2, the total failure probability is upper bounded by $\epsilon_1 + \epsilon_2$ by the union bound.

Let's fix any pair of row indices (u_1, u_2). The probability for any column index $i \in [n]$ to be in their overlap $\mathcal{O}_{uv}$ is $\mathbb{P}(M(u_1, i) = 1)\,\mathbb{P}(M(u_2, i) = 1) = p^2$ and it follows that $|\mathcal{O}_{u_1 u_2}| \sim Bin(n, p^2)$ from the i.i.d. assumption on M. By the Chernoff bound for lower tail of a binomial distribution,

$$\epsilon_1 := \mathbb{P}\left(|\mathcal{O}_{u_1 u_2}| \leq \beta\right) \leq \exp\left(-\frac{np^2}{2}\left(1 - \frac{\beta}{np^2}\right)^2\right).$$

For a fixed u, there are $(m-1)$ candidate rows to test, and the probability of failing in the first condition

$$\begin{aligned}\epsilon_1 &:= \mathbb{P}\left(\exists v \in [m] \setminus u \ s.t.\ |\mathcal{O}_{uv}| \geq \beta\right) \\ &\leq (m-1)\exp\left(-\frac{np^2}{2}\left(1 - \frac{\beta}{np^2}\right)^2\right).\end{aligned}$$

From the assumption $\beta \leq \frac{np^2}{2}$, it follows that

$$\exp\left(-\frac{np^2}{2}\left(1-\frac{\beta}{np^2}\right)^2\right) \leq \exp\left(-\frac{np^2}{8}\right).$$

Now we claim that $|\mathcal{O}_i|$ concentrates around mp. Because $|\mathcal{O}_i| \sim Bin\left((m-1), p\right)$ for any $i \in [n]$, we have

$$\mathbb{P}\left(|\mathcal{O}_i| \notin (1 \pm \alpha)(m-1)p\right) \leq 2\exp\left(-\frac{\alpha^2}{3}(m-1)p\right), \quad \forall \delta \in (0,1).$$

By the union bound, the total failure probability is no greater than $\epsilon_1 + \epsilon_2$, and we can conclude that $\left|\mathcal{S}_u^{\beta}(i)\right| \approx (m-1)p$ for every (u,i) with high probability. □

Applying the union bound for every $(u,i) \in [m] \times [n]$ naïvely, we know that

$$\begin{aligned}
&\mathbb{P}\left(\exists (u,i) \in [m] \times [n] : \left|\mathcal{S}_u^{\beta}(i)\right| \notin (1 \pm \alpha)(m-1)p\right) \\
&\qquad \leq \sum_{(u,i) \in [m] \times [n]} \mathbb{P}\left(\left|\mathcal{S}_u^{\beta}(i)\right| \notin (1 \pm \alpha)(m-1)p\right) \\
&\qquad \leq m(m-1)n \exp\left(-\frac{np^2}{2}\left(1-\frac{\beta}{np^2}\right)^2\right) + 2mn\exp\left(-\frac{\alpha^2}{3}mp\right).
\end{aligned}$$

However, a slight modification in the proof of the previous lemma can provide a better result:

$$\begin{aligned}
&\mathbb{P}\left(\exists (u,i) \in [m] \times [n] : \left|\mathcal{B}_{row}^{\beta}(u,i)\right| \notin (1 \pm \alpha)(m-1)p\right) \\
&\qquad \leq \frac{m(m-1)}{2} \exp\left(-\frac{np^2}{2}\left(1-\frac{\beta}{np^2}\right)^2\right) + 2n\exp\left(-\frac{\alpha^2}{3}mp\right).
\end{aligned}$$

These two terms decay exponentially as long as $\beta < \frac{np^2}{2}$, $\log m(m-1) - \frac{np^2}{2} < 0$, and $\log n - \frac{\alpha^2}{3}mp < 0$, i.e., $p \gtrsim \max\{\sqrt{n^{-1}\log m}, m^{-1}\log n\}$.

5.2 Existence of a Good Neighbor

In order to show that a good-quality neighbor can be detected through sample variance, we need to show there exists a neighbor row whose population variance is small. For that purpose, we assume our latent space $\mathcal{X}_1$ is bounded, the distribution $P_{\mathcal{X}_1}$ allows every nontrivial ball to have positive measure, and f is Lipschitz. Under these assumptions, every (u, i) has nonzero probability of having a close neighbor. In fact, there exists a close neighbor for every entry with high probability.

Let's suppose row u_1 has a latent feature representation a and row u_2 has b. We cannot observe those feature representations, but we can define the popultion variance of the differences between those two features a and b as

$$\sigma_{ab}^2 \triangleq Var[f(a, \mathbf{x_2}) - f(b, \mathbf{x_2})],$$

for $a, b \in \mathcal{X}_1$ and $\mathbf{x}_2 \sim P_{\mathcal{X}_2}$. Abusing terminology, this will be also referred to as the population variance between two rows u_1 and u_2.

Lemma 2 (Existence of a good neighbor)**.** *Suppose that $(\mathcal{X}_1, P_{\mathcal{X}_1})$ admits a nondecreasing function $h : \mathbb{R}_{++} \to (0, 1]$ satisfying*

$$P_{\mathcal{X}_1}(\mathbf{x}_1 \in B(x_0, r)) \geq h(r), \quad \forall x_0 \in \mathcal{X}_1, \ \forall r > 0,$$

where $B(x_0, r) \triangleq \{x \in \mathcal{X}_1 \ s.t. \ d(x, x_0) \leq r\}$. If we fix $u \in [m]$, for any subset of indices $\mathcal{S} \subset [n] \setminus \{u\}$ and any $\rho > 0$, the probability of nonexistence of a neighbor within ρ in S satisfifes

$$\mathbb{P}\left(\bigcap_{v \in \mathcal{S}} \left\{\sigma_{x_1(u)x_1(v)}^2 > \rho\right\}\right) \leq \left[1 - h\left(\sqrt{\frac{\rho}{L^2}}\right)\right]^{|\mathcal{S}|}.$$

Proof. Recall that $\sigma_{ab}^2 \triangleq \mathrm{Var}[f(a, x) - f(b, x)]$, for some $a, b \in \mathcal{X}_1$, and $x \sim P_{\mathcal{X}_2}$. We can bound σ_{ab}^2 as a function of $d_1(a, b)$ by using the Lipschitz property of f, that

$|f(a,x_2) - f(b,x_2)| \leq Ld_1(a,b), \forall x_2 \in \mathcal{X}_2.$

$$\begin{aligned}
\sigma_{ab}^2 &= \mathrm{Var}[f(a,\mathbf{x}_2) - f(b,\mathbf{x}_2)] \\
&= \mathbb{E}\left[(f(a,\mathbf{x}_2) - f(b,\mathbf{x}_2))^2\right] - \mathbb{E}\left[f(a,\mathbf{x}_2) - f(b,\mathbf{x}_2)\right]^2 \\
&\leq \mathbb{E}\left[(f(a,\mathbf{x}_2) - f(b,\mathbf{x}_2))^2\right] \\
&\leq \mathbb{E}\left[(Ld_1(a,b))^2\right] \\
&= L^2 d_1(a,b)^2.
\end{aligned}$$

It follows that

$$\begin{aligned}
\mathbb{P}\left(\bigcap_{v\in\mathcal{S}} \{\sigma^2_{x_1(u)x_1(v)} > \rho\}\right) &= \int \mathbb{P}\left(\bigcap_{v\in\mathcal{S}} \{\sigma^2_{x_1(u)x_1(v)} > \rho\} \middle| x_1(u) = x_0\right) P_{\mathcal{X}_1}(x_0)dx_0 \\
&= \int \mathbb{P}\left(\sigma^2_{x_0,x_1(v)} > \rho\right)^{|\mathcal{S}|} P_{\mathcal{X}_1}(x_0)dx_0 \quad \because x_1(v) \textit{ drawn i.i.d.} \\
&\leq \int \mathbb{P}\left(L^2 d_1(x_0,x_1(v))^2 > \rho\right)^{|\mathcal{S}|} P_{\mathcal{X}_1}(x_0)dx_0 \\
&= \int \mathbb{P}\left(d_1(x_0,x_1(v)) > \sqrt{\frac{\rho}{L^2}}\right)^{|\mathcal{S}|} P_{\mathcal{X}_1}(x_0)dx_0 \\
&= \int P_{\mathcal{X}_1}\left[\mathbf{x}_1 \notin B\left(x_0, \sqrt{\frac{\rho}{L^2}}\right)\right]^{|\mathcal{S}|} P_{\mathcal{X}_1}(x_0)dx_0 \\
&\leq \int \left[1 - h\left(\sqrt{\frac{\rho}{L^2}}\right)\right]^{|\mathcal{S}|} P_{\mathcal{X}_1}(x_0)dx_0 \\
&= \left[1 - h\left(\sqrt{\frac{\rho}{L^2}}\right)\right]^{|\mathcal{S}|}.
\end{aligned}$$

□

How does h look like? In order to provide some understanding toward the assumption on distribution $P_{\mathcal{X}_1}$, observe that the function $h(\cdot)$ is a form of the cumulative distribution function (CDF) for $P_{\mathcal{X}_1}$. The only distribution which does not satisfy this property is a distribution which has non-atomic isolated points. However, these isolated points have measure zero, such that they will never appear in our datasets with probability 1. We provide a few examples of distributions and their corresponding functions $h(\cdot)$.

Example 1 (extremely uniform). *Suppose that* $\mathcal{X} = \times_{k=1}^{d}[a_i, b_i] \in \mathbb{R}^d$ *equipped with* L_∞ *norm and* $P_{\mathcal{X}}$ *is a uniform distribution over* $\mathcal{X}$. *We can see that the function* $h(r) := \prod_{k=1}^{d} \min\left\{1, \frac{r}{b_i - a_i}\right\}$ *satisfies the condition* $P_{\mathcal{X}}\left(\mathbf{x} \in B(x_0, r)\right) \geq h(r),\ \forall x_0 \in \mathcal{X},\ \forall r > 0$. *Note that* $h(r) \approx cr^d$ *for* $r \ll 1$.

Example 2 (extremely clustered). *Suppose that* $\mathcal{X} = \{x_1, \ldots, x_d\}$ *equipped with the discrete topology and* $P_{\mathcal{X}}$ *is expressed in terms of the probability masses* $P_{\mathcal{X}}(x_k) = p_k$ *with* $\sum_{k=1}^{d} p_k = 1$. *We can see that the function* $h(r) := \min_k p_k$ *works for* $(\mathcal{X}, P_{\mathcal{X}})$ *even when we do not know the metric.*

We show that a compact metric space admits such a function $h(\cdot)$ for regular points.

Definition 1 (regular points). *Let* (X, d) *be a compact metric space, and* P_X *be a Borel probability measure on it. A point* $x \in X$ *is called regular if*

$$P_X\left(B(x, r)\right) > 0, \quad \forall r > 0.$$

Lemma 3 (Existence of h function). *Let* (X, d) *be a compact metric space, and* P_X *be a Borel probability measure on it. Then there is a function* h *on* (X, P_X) *which satisfies*

1. $h : \mathbb{R}_{++} \to (0, 1]$ *is nondecreasing, and*
2. $P_X\left(B(x, r)\right) \geq h(r), \quad \forall x \in X$ *regular,* $\forall r > 0$.

Proof. For any $r > 0$, the family $\{B(x, r/2) : x \in X\}$ forms n open cover X. Since X is compact, there exists a finite subcover $C_0 := \{B_1, \ldots, B_{N_0}\}$. From this subcover, remove every measure-zero ball to obtain $C := \{B_1, \ldots, B_N\}$. It can be observed that for every regular point $x \in X$, there exists an $r/2$-ball $B_i \in C$ contained in $B(x, r)$. Therefore, for every regular $x \in X$, $B(x, r) \geq \min_{B_i \in C} P_X(B_i)$. We let $h(r) := \sup_C \min_{B_i \in C} P_X(B_i)$ over every finite subcover $C \subset C_0$. It is obvious that $P_X\left(B(x, r)\right) \geq h(r)$ and $h(r) > 0$ for $r > 0$.

For $r_1 > 0$, $h(r_1) := \sup_{C(r_1) \subset C_0(r_1)} \min_{B_i \in C(r_1)} P_X(B_i)$ by definition. If $C := \{B(x_1, r_1/2), \ldots, B(x_N, r_1/2)\}$ is a finite cover of X, so is the collection of balls with extended radii $C' := \{B(x_1, r_2/2), \ldots, B(x_N, r_2/2)\}$ when $r_2 \geq r_1$. Because $B(x_i, r_1/2) \subset B(x_i, r_2/2)$ and P_X is a measure, $P_X(B(x_i, r_1/2)) \leq P_X(B(x_i, r_2/2))$. Therefore, $\min_{B_i \in C} P_X(B_i) \leq \min_{B_i \in C'} P_X(B_i)$ for every finite subcover C of $C_0(r_1)$ and it follows that

$$\begin{aligned} h(r_2) &:= \sup_{C(r_2) \subset C_0(r_2)} \min_{B_i \in C(r_2)} P_X(B_i) \\ &\geq \sup_{C'(r_1) : C(r_1) \subset C_0(r_1)} \min_{B_i \in C'(r_1)} P_X(B_i) \\ &\geq \sup_{C(r_1) \subset C_0(r_1)} \min_{B_i \in C(r_1)} P_X(B_i) \\ &= h(r_1). \end{aligned}$$

□

5.3 Concentration of Sample Mean and Sample Variance

5.3.1 Concentration of Sample Means

Lemma 4 (Concentration of sample means). *Given $u, v \in [n]$, $i \in [m]$ and $\beta \geq 2$, for any $\alpha > 0$,*

$$\mathbb{P}\left(\left|\mu_{x_1(u)x_1(v)} - m_{uv}\right| > \alpha \,|\, v \in \mathcal{S}_u^\beta(i)\right) \leq \exp\left(-\frac{3\beta\alpha^2}{6B^2 + 2B\alpha}\right),$$

where recall that $B = 2(LB_{\mathcal{X}} + B_\eta)$.

Proof. Given $\mathbf{x}_1(u) = x_1(u), \mathbf{x}_1(v) = x_1(v)$, the mean $\mu_{x_1(u)x_1(v)}$ is a constant. Recall that empirical mean m_{uv} is defined as (see (3.4))

$$m_{uv} = \frac{1}{|\mathcal{O}_{uv}|}\left(\sum_{j\in\mathcal{O}_{uv}} A(u,j) - A(v,j)\right). \tag{5.1}$$

The variable $\mathbf{x}_2(j)$ is sampled as per P_2, independently from $x_1(u), x_1(v)$. And the noise term in each of the observation is independent zero-mean variable. Therefore, conditioned on $\mathbf{x}_1(u) = x_1(u), \mathbf{x}_1(v) = x_1(v)$, we have independent random variable, $Z(j) = A(u,j) - A(v,j)$ for $j \in \mathcal{O}_{uv}$, that have mean $\mu_{x_1(u)x_1(v)}$. That is, $\tilde{Z}(j) = Z(j) - \mu_{x_1(u)x_1(v)}$, $j \in \mathcal{O}_{uv}$ are zero-mean independent random variables. And by definition, each of them is bounded as

$$|\tilde{Z}(j)| \leq 2B_\eta + LB_{\mathcal{X}} \ \leq 2(LB_{\mathcal{X}} + B_\eta) \ = \ B. \tag{5.2}$$

In summary, conditioned on $\mathbf{x}_1(u) = x_1(u), \mathbf{x}_1(v) = x_1(v)$ and $\mathcal{O}_{uv}$, $\mu_{x_1(u)x_1(v)} - m_{uv}$ is the average of $\mathcal{O}_{uv}$ independent, zero mean random variables $\tilde{Z(j)}$, each of which have absolute value bounded above by B. Therefore, an application of Bernstein's inequality imply that

$$\mathbb{P}\left(\left|\mu_{x_1(u)x_1(v)} - m_{uv}\right| > \alpha \,\middle|\, \mathbf{x}_1(u) = x_1(u), \mathbf{x}_1(v) = x_1(v), \mathcal{O}_{uv}\right) \leq \exp\left(-\frac{3\,|\mathcal{O}_{uv}|\,\alpha^2}{6B^2 + 2B\alpha}\right). \tag{5.3}$$

When $v \in S_u^\beta(i)$, $|\mathcal{O}_{uv}| \geq \beta$. Further, since above holds for all possibilities of $x_1(u), x_2(v)$, we conclude that

$$\mathbb{P}\left(\left|\mu_{\mathbf{x}_1(u)\mathbf{x}_1(v)} - m_{uv}\right| > \alpha \,|\, v \in \mathcal{S}_u^\beta(i)\right) \leq \exp\left(-\frac{3\beta\alpha^2}{6B^2 + 2B\alpha}\right).$$

□

5.3.2 Concentration of Sample Variances

Next we establish the concentration of the sample variance.

To begin with, for every $\mathbf{x} = (x_1, \ldots, x_n) \in [0,1]^n$, we let

$$m_n(\mathbf{x}) := \frac{1}{n}\sum_{i=1}^{n} x_i,$$

and

$$V_n(\mathbf{x}) := \frac{1}{n(n-1)}\sum_{i,j=1}^{n} \frac{(x_i - x_j)^2}{2}.$$

It is easy to check that V_n is the same with the traditional sample variance:

$$\begin{aligned}
S_n^2 &= \frac{1}{n-1}\sum_{i=1}^{n} (x_i - m_n(\mathbf{x}))^2 \\
&= \frac{1}{n-1}\sum_{i=1}^{n} \left(x_i - \frac{1}{n}\sum_{j=1}^{n} x_j\right)^2 \\
&= \frac{1}{n-1}\sum_{i=1}^{n} \left[x_i^2 - \frac{2}{n}x_i\sum_{j=1}^{n} x_j + \frac{1}{n^2}\left(\sum_{j=1}^{n} x_j\right)^2\right] \\
&= \frac{1}{n-1}\sum_{i=1}^{n} x_i^2 - \frac{1}{n}\left(\sum_{i=1}^{n} x_i\right)^2 \\
&= \frac{1}{2(n-1)}\left(\sum_{i=1}^{n} x_i^2 - 2\sum_{i=1}^{n} x_i\sum_{j=1}^{n} x_j + \sum_{j=1}^{n} x_j^2\right) \\
&= \frac{1}{n(n-1)}\sum_{i,j=1}^{n} \frac{(x_i - x_j)^2}{2} \\
&= V_n(\mathbf{x}).
\end{aligned}$$

The following lemma provides concentration inequalities for sample variance of bounded independent random variables (see Appendix B.2.3 for more detail).

Lemma 5. *Let $n \geq 2$ and $\mathbf{X} = (X_1, \ldots, X_n)$ be a vector of independent random variables with values in $[c,d]$. Then for $t > 0$ we have*

1. *Upper tail: $\mathbb{P}\left(V_n(\mathbf{X}) - \mathbb{E}V_n > t\right) \leq \exp\left(-\frac{(n-1)t^2}{(d-c)^2(2\mathbb{E}V_n + t)}\right)$, and*
2. *Lower tail: $\mathbb{P}\left(V_n(\mathbf{X}) - \mathbb{E}V_n < -t\right) \leq \exp\left(-\frac{(n-1)t^2}{2(d-c)^2\mathbb{E}V_n}\right)$.*

Proof. We can normalize $\mathbf{Y} = \frac{\mathbf{X}-c}{d-c}$ so that $\mathbf{Y}$ is a vector of independent random variables with values in $[0,1]$. Note that $V_n(\mathbf{X}) = (d-c)^2 V_n(\mathbf{Y})$.

Write $Z(\mathbf{Y}) = nV_n(\mathbf{Y})$. Fix some k and choose any $z \in [0,1]$. Then

$$\begin{aligned} Z(\mathbf{Y}) - Z(\mathbf{Y}_k^z) &= \frac{1}{n-1} \sum_i \left((Y_k - Y_i)^2 - (z - Y_i)^2 \right) \\ &\leq \frac{1}{n-1} \sum_i (Y_k - Y_i)^2 \quad \because (z - Y_i)^2 \geq 0. \end{aligned}$$

It follows from $(Y_k - Y_i)^2 \in [0,1]$ that $Z(\mathbf{Y}) - \inf_{z \in [0,1]} Z(\mathbf{Y}_k^z) \leq 1$. We also get

$$\begin{aligned} \sum_{k=1}^{n} \left(Z(\mathbf{Y}) - \inf_{z \in [0,1]} Z(\mathbf{Y}_k^z) \right)^2 &\leq \sum_{k=1}^{n} \left(\frac{1}{n-1} \sum_{i=1}^{n} (Y_k - Y_i)^2 \right)^2 \\ &= \frac{n^3}{(n-1)^2} \left[\frac{1}{n} \sum_k \left(\frac{1}{n} \sum_i (Y_k - Y_i)^2 \right)^2 \right] \\ &\leq \frac{n^3}{(n-1)^2} \left[\frac{1}{2n^2} \sum_{i,k} (Y_i - Y_k)^2 \right] \quad \because \textit{Corollary } 12 \\ &= \frac{n}{n-1} Z(\mathbf{Y}). \end{aligned}$$

It follows that Z satisfies two conditions in Theorem 7with $a = \frac{n}{n-1}$.

Note that

$$\begin{aligned} \mathbb{P}\left(V_n(\mathbf{X}) - \mathbb{E}\left[V_n^X \right] > t \right) &= \mathbb{P}\left(V_n(\mathbf{Y}) - \mathbb{E}\left[V_n^Y \right] > \frac{t}{(d-c)^2} \right) \\ &= \mathbb{P}\left(Z(\mathbf{Y}) - \mathbb{E}\left[Z \right] > \frac{nt}{(d-c)^2} \right), \end{aligned}$$

and similarly,

$$\begin{aligned} \mathbb{P}\left(V_n(\mathbf{X}) - \mathbb{E}\left[V_n^X \right] < -t \right) &= \mathbb{P}\left(V_n(\mathbf{Y}) - \mathbb{E}\left[V_n^Y \right] < -\frac{t}{(d-c)^2} \right) \\ &= \mathbb{P}\left(Z(\mathbf{Y}) - \mathbb{E}\left[Z \right] < -\frac{nt}{(d-c)^2} \right). \end{aligned}$$

From Theorem 7, we can conclude that

$$\begin{aligned}\mathbb{P}\left(V_n(\mathbf{X}) - \mathbb{E}V_n > t\right) &\leq \exp\left(-\frac{(n-1)t^2}{(d-c)^2\left(2\mathbb{E}V_n + t\right)}\right), \text{ and}\\ \mathbb{P}\left(V_n(\mathbf{X}) - \mathbb{E}V_n < -t\right) &\leq \exp\left(-\frac{(n-1)t^2}{2(d-c)^2\mathbb{E}V_n}\right).\end{aligned}$$

□

Returning to our problem, we know that the ratings $A(u,i)$ are bounded in our model:

$$\begin{aligned}|A(u,i) - A(v,j)| &= \left|\left[f\left(x_1(u), x_2(i)\right) + E(u,i)\right] - \left[f\left(x_1(v), x_2(j)\right) + E(v,j)\right]\right|\\ &\leq \left|f\left(x_1(u), x_2(i)\right) - f\left(x_1(v), x_2(j)\right)\right| + |E(u,i) - E(v,j)|\\ &\leq LB + 2B_e.\end{aligned}$$

Alternatively, one might directly know that $A(u,i) \in [B_{A1}, B_{A2}]$. In either case, we know that A is bounded. Let $D = \min\{LB + 2B_e, B_{A2} - B_{A1}\}$ denote the range of observed ratings.

Lemma 6 (Concentration of sample variances)**.** *Given $u \in [m]$, $i \in [n]$, and $\beta \geq 2$, for any $\rho > 0$,*

$$\mathbb{P}\left(\left|s_{uv}^2 - (\sigma_{x_1(u)x_1(v)}^2 + 2\gamma^2)\right| > \rho \quad \middle| \quad v \in \mathcal{S}_u^\beta(i)\right) \leq 2\exp\left(-\frac{\beta\rho^2}{4B^2(2LB_{\mathcal{X}}^2 + 4\gamma^2 + \rho)}\right),$$

where recall that $B = 2(LB_{\mathcal{X}} + B_\eta)$.

Proof. Recall $\sigma_{ab}^2 \triangleq \mathrm{Var}[f(a,\mathbf{x}) - f(b,\mathbf{x})]$ for $a, b \in \mathcal{X}_1$, $\mathbf{x} \sim P_{\mathcal{X}_2}$, and sample variance between rows u and v is defined as

$$\begin{aligned}s_{uv}^2 &= \frac{1}{2\left|\mathcal{O}_{uv}\right|\left(\left|\mathcal{O}_{uv}\right| - 1\right)}\sum_{j,j' \in \mathcal{O}_{uv}}\left((y(u,j) - y(v,j)) - (y(u,j') - y(v,j'))\right)^2\\ &= \frac{1}{\left|\mathcal{O}_{uv}\right| - 1}\sum_{j \in \mathcal{O}_{uv}}\left(y(u,j) - y(v,j) - m_{uv}\right)^2.\end{aligned}$$

Conditioned on $\mathbf{x}_1(u) = x_1(u), \mathbf{x}_1(v) = x_1(v)$, we obtain that $\mathbb{E}\left[s_{uv}^2\right] = \sigma^2_{x_1(u)x_1(v)} + 2\gamma^2$, with respect to randomness induced by P_2 for sampling latent parameters for columns. Further, $Z(j) = A(u,j) - A(v,j)$ are independent random variables conditioned on $\mathbf{x}_1(u) = x_1(u), \mathbf{x}_1(v) = x_1(v)$. Using the fact that f is Lipschitz, space is bounded and noise is bounded, as before, we obtain that

$$|Z(j)| \;=\; |A(u,j) - A(v,j)| \leq 2(LB_{\mathcal{X}} + B_\eta) = B.$$

Given this, by an application of Maurer-Pontil inequality (see Lemma 5 above, and Theorem 7 in Appendix), we obtain that

$$\begin{aligned}\mathbb{P}\left(\left|s_{uv}^2 - (\sigma^2_{x_1(u)x_1(v)} + 2\gamma^2)\right| > \rho \,|\, v \in \mathcal{S}_u^\beta(i), \mathbf{x}_1(u) = x_1(u), \mathbf{x}_1(v) = x_1(v)\right) \\ \leq 2\exp\left(-\frac{\beta\rho^2}{4B^2(2(\sigma^2_{x_1(u)x_1(v)} + 2\gamma^2) + \rho)}\right), \qquad (5.4)\end{aligned}$$

where we used the property that $v \in \mathcal{S}_u^\beta(i)$ implies $|\mathcal{O}_{uv}| \geq \beta$. Using the Lipschitz property of f and boundedness of $\mathcal{X}_1$, we can bound $\sigma^2_{x_1(u)x_1(v)} \leq L^2B^2_{\mathcal{X}}$ as before. Therefore, the right hand side of (5.4) can be bounded as

$$\leq 2\exp\left(-\frac{\beta\rho^2}{4B^2(2L^2B^2_{\mathcal{X}} + 4\gamma^2 + \rho)}\right). \qquad (5.5)$$

Given that this bound is indepedent of $x_1(u), x_1(v)$, we can conclude the desired result. □

5.4 Concentration of Estimates

Now we establish the final step in the proof of Theorem 1. As in the proof of Theorem 1, for a given (u,i) with $u \in [m]$, $i \in [n]$ and $\beta \geq 2$, define events

- Let A denote the event that $|\mathcal{S}_u^\beta(i)| \in [(m-1)p/2, 3(m-1)p/2]$,
- Let B denote the event that $\min_{v \in \mathcal{S}_u^\beta(i)} \sigma^2_{\mathbf{x}_1(u)\mathbf{x}_1(v)} < \rho$,

- Let C denote the event that $\left|\mu_{\mathbf{x}_1(u)\mathbf{x}_1(v)} - m_{uv}\right| < \alpha$ for all $v \in \mathcal{S}_u^\beta(i)$,

- Let D denote the event that $\left|s_{uv}^2 - (\sigma^2_{\mathbf{x}_1(u)\mathbf{x}_1(v)} + 2\gamma^2)\right| < \rho$ for all $v \in \mathcal{S}_u^\beta(i)$.

Lemma 7. *Under the setting described above and given $\alpha > 0,\ \rho > 0$ and $\epsilon > \alpha$, under the algorithm user-user nearest neighbor, we have*

$$\mathbb{P}\Big(\ \left|R(u,i) - \hat{R}(u,i)\right| > \epsilon \,|\, A, B, C, D\Big) \leq \frac{3\rho + \gamma^2}{(\epsilon - \alpha)^2}.$$

Proof. Under the algorithm user-user nearest neighbor, the error of the estimate is given by

$$\begin{aligned} R(u,i) - \hat{R}(u,i) &= f(\mathbf{x}_1(u), \mathbf{x}_2(i)) - A(u^*, i) - m_{uu^*} \\ &= f(\mathbf{x}_1(u), \mathbf{x}_2(i)) - f(\mathbf{x}_1(u^*), \mathbf{x}_2(i)) - \eta_{u^*,i} - m_{uu^*}. \end{aligned}$$

Given $\mathbf{x}_1(u) = x_1(u), \mathbf{x}_1(u^*) = x_1(u^*)$ such that events A, B, C and D are satisfied, we have that

$$\mathbb{E}\left[f(x_1(u), \mathbf{x}_2(i)) - f(x_1(u^*), \mathbf{x}_2(i)) - \eta_{u^*,i}\right] = \mu_{x_1(u)x_1(u^*)}, \tag{5.6}$$

with respect to $\mathbf{x}_2(i) \sim P_2$.

Conditioned on event C, that is, $\left|\mu_{x_1(u)x_1(v)} - m_{uv}\right| < \alpha$ for all $v \in \mathcal{S}_u^\beta(i)$, included u^*, it is sufficient to bound the probability of event

$$E = \Big\{\ |f(x_1(u), \mathbf{x}_2(i)) - f(x_1(u^*), \mathbf{x}_2(i)) - \eta_{u^*,i} - \mu_{uu^*}| > \epsilon - \alpha\Big\}. \tag{5.7}$$

Conditioned on $\mathbf{x}_1(u) = x_1(u), \mathbf{x}_1(u^*) = x_1(u^*)$,

$$\mathrm{Var}\left[f(x_1(u), \mathbf{x}_2(i)) - f(x_1(u^*), \mathbf{x}_2(i)) - \eta_{u^*,i}\right] = \sigma^2_{x_1(u)x_1(u^*)} + \gamma^2. \tag{5.8}$$

Therefore, by standard Chebychev's inequality, we obtain

$$\mathbb{P}\left(|f(x_1(u), \mathbf{x}_2(i)) - f(x_1(u^*), \mathbf{x}_2(i)) - \eta_{u^*,i} - \mu_{x_1(u)x_1(u^*)}| > \epsilon - \alpha\right) \leq \frac{\sigma^2_{x_1(u)x_1(u^*)} + \gamma^2}{(\epsilon - \alpha)^2}. \tag{5.9}$$

The selection of u^* was done using empirical estimates s^2_{uv} across $v \in \mathcal{S}^{\beta}_{u}(i)$. By condition on event D happening, we have that for any $v \in \mathcal{S}^{\beta}_{u}(i)$, s^2_{uv} is within ρ of $(\sigma^2_{\mathbf{x}_1(u)\mathbf{x}_1(v)} + 2\gamma^2)$. And condition on event B, we have that there is at least one $v \in \mathcal{S}^{\beta}_{u}(i)$ so that $\sigma^2_{\mathbf{x}_1(u)\mathbf{x}_1(v)} < \rho$; let one such v be denoted as v^*. Therefore, we obtain that

$$\begin{aligned}\sigma^2_{x_1(u)x_1(u*)} + 2\gamma^2 - \rho &\leq s^2_{uu^*} \\ &\leq s^2_{uv} \\ &\leq \sigma^2_{x_1(u)\mathbf{x}_1(v)} + 2\gamma^2 + \rho \\ &\leq 2\gamma^2 + 2\rho. \end{aligned} \tag{5.10}$$

From above, we can conclude that $\sigma^2_{x_1(u)x_1(u^*)} \leq 3\rho$. Replacing this in (5.9), we obtain the bound on right hand side as

$$\leq \frac{3\rho + \gamma^2}{(\epsilon - \alpha)^2}. \tag{5.11}$$

Since this bound holds for all choices of $\mathbf{x}_1(u), \mathbf{x}_1(u^*)$ conditioned on events A, B, C and D, we conclude the desired result. □

Chapter 6

Experiments

In this chapter, we present some empirical observations on our algorithm. In Section 6.1, we evaluate our algorithm on two movie rating datasets and the experiments on MovieLens and Netflix datasets suggest that our algorithm provides principled improvements over basic collaborative filtering and matrix factorization methods. The blind regression framework naturally extends to the setting of higher order tensors by simply flattening the tensor into a matrix. In Section 6.2, we apply our method to the tensor completion problem for image reconstruction, showing that our simple and principled approach is competitive with respect to the state-of-art tensor completion algorithms. Lastly, in Section 6.3, we report additional observations which empircally justify the use of sample variance as a proxy for the squared distance.

6.1 Matrix Completion Experiments

We evaluated the performance of our algorithm to predict user-movie ratings on the MovieLens 1M and Netflix datasets. The MovieLens 1M data set contains about 1 million ratings by 6000 users of 4000 movies from the online movie recommendation service MovieLens. The Netflix data set consists of about 100 million movie ratings by 480,189 users of about 17,770 movies. In both data sets, the ratings are integers from 1 to 5. From each dataset, we generated 100 smaller user-movie rating matrices, in which we randomly subsampled 2000 users and 2000 movies.

For the implementation of our method, we used user-item Gaussian kernel weights for the final estimator. We chose overlap parameter $\beta = 2$ to ensure the algorithm is able to compute an estimate for all missing entries. When β is larger, the algorithm enforces rows (or columns) to have more commonly rated movies (or users). Although this increases the reliability of the estimates, it also reduces the fraction of entries for which the estimate is defined. We optimized the bandwidth parameter λ of the Gaussian kernel by evaluating the method with multiple values for λ and choosing the value which minimizes the error.

We compared our method with user-user collaborative filtering, item-item collaborative filtering, and softImpute from [39]. We chose the classic mean-adjusted collaborative filtering method, in which the weights are proportional to the cosine similarity of pairs of users or items (i.e. movies). SoftImpute is a matrix-factorization-based method which iteratively replaces missing elements in the matrix with those obtained from a soft-thresholded SVD.

For each rating matrix, we randomly select and withhold a percentage of the known ratings for the test set, while the remaining portion of the data set is revealed to the algorithm for computing the estimates. After the algorithm computes its predictions for unrevealed movie-user pairs, we evaluate the root mean squared error (RMSE) of the predictions compared with the withheld test set, where RMSE is defined as the square root of the mean of squared prediction error over the evaluation set. Figure 6-1 plots the RMSE of our method along with classic collaborative filtering and softImpute evaluated against 10%, 30%, 50%, and 70% withheld test sets. The RMSE is averaged over 100 subsampled rating matrices, and 95% confidence intervals are provided.

Figure 6-1 suggests that our algorithm achieves a systematic improvement over classical user-user and item-item collaborative filtering. SoftImpute performs worse than all methods on the MovieLens dataset, but it performs better than all methods on the Netflix dataset.

The reason behind this behavioral difference is not clear, but it could be due to the different nature of the dataset, for example, the density (or sparsity) of the

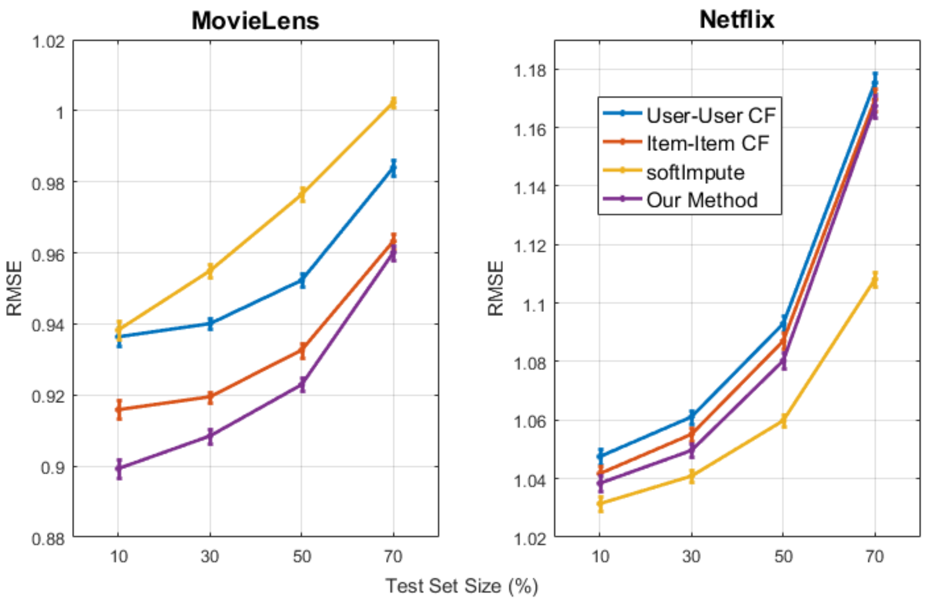

Figure 6-1: Performance of algorithms on Netflix and MovieLens data set with 95% confidence interval. λ values used by our algorithm are 2.8 (10%), 2.3 (30%), 1.7 (50%), 1 (70%) for MovieLens, and 1.8 (10%), 1.7 (30%), 1.6 (50%), 1.5 (70%) for Netflix.

dataset. For the MovieLens dataset, roughly 4.2% (1M out of 6,000 users and 4,000 movies) of the user-movie ratings are known, while only 1.2% (100M out of of 480,189 users and 17,770 movies) the ratings are available for the Netflix dataset. From the observations, we hypothesize that neighborhood-based methods provide more accurate predictions when there are abundant rating data available; however, their performance deteriorates sharply as data become sparse, compared to the low-rank matrix factorization method. This behavior could be due to the different underlying assumptions of low rank for matrix factorization methods as opposed to Lipschitz for collaborative filtering methods.

6.2 Tensor Completion Experiments

We evaluated and compared the performance of our tensor completion algorithm against existing methods in the literature on the image inpainting problem. An image can be represented as a 3$^{\text{rd}}$-order tensor where the dimensions are rows × columns ×

RGB. In particular we used three images (Lenna, Pepper, and Facade) of dimensions 256 × 256 × 3. For each image, a percentage of the pixels are randomly removed, and the missing entries are filled in by various tensor completion algorithms.

For the implementation of our tensor completion method, we collapsed the last two dimensions of the tensor (columns and RGB) to reduce the image to a matrix, and applied our method with user-item Gaussian kernel weights. We set the overlap parameter $\beta = 2$, and we optimized over the Gaussian kernel bandwidth parameter λ. We compared our method against fast low rank tensor completion (FaLRTC) [33], alternating minimization for tensor completion (TenAlt) [24], and fully Bayesian CP factorization (FBCP) [53], which extends the CANDECOMP/PARAFAC (CP) tensor factorization with automatic tensor rank determination.

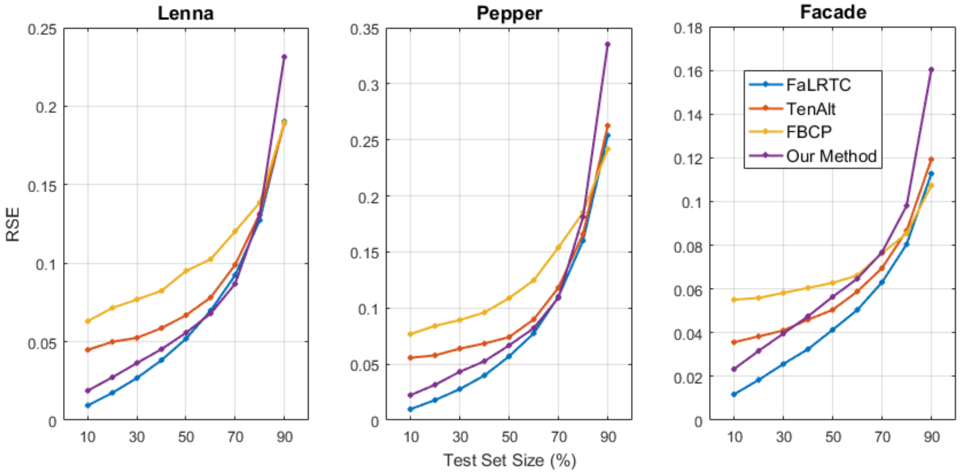

Figure 6-2: Performance comparison between different tensor completion algorithms based on RSE vs testing set size. For our method, we set overlap parameter β to 2.

To evaluate the outputs produced by each method, we computed the relative squared error (RSE), defined as

$$\text{RSE} := \frac{\sum_{i,j,k \in E} (\hat{R}_{ijk} - R_{ijk})^2}{\sum_{i,j,k \in E} (R_{ijk} - \bar{R})^2},$$

where $\bar{R}$ is the average value of the true entries. Figure $6-2$ plots the RSE achieved by each tensor completion method on the three images, as a function of the percentage of pixels removed. Figure 6-3 shows a sample of the image inpainting results for the

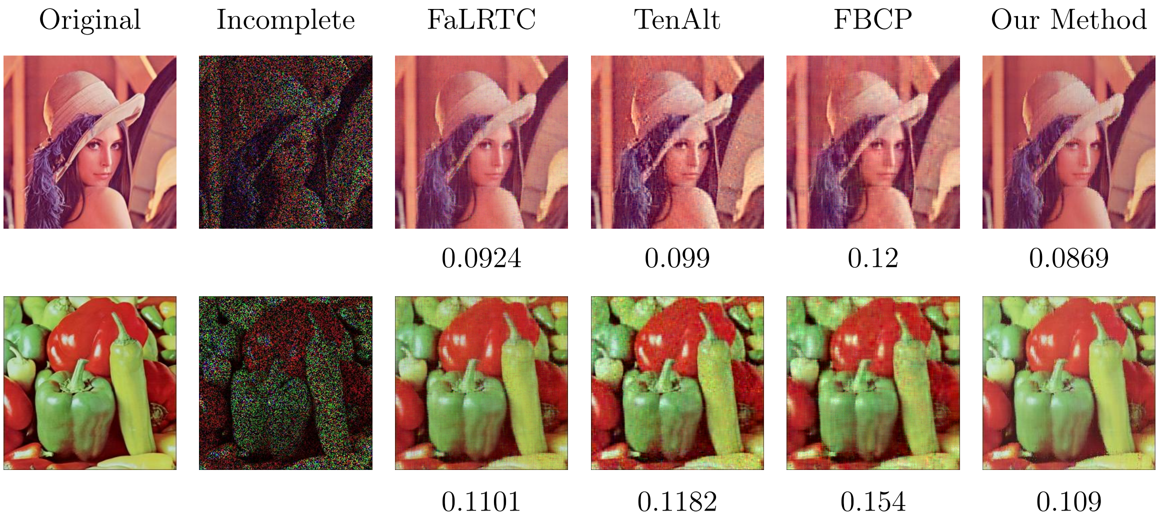

Figure 6-3: Recovery results for Lenna, Pepper and Facade images with 70% of missing entries. RSE is reported under the recovery images.

lenna and pepper images when 70% of the pixels are removed.

Again as discussed in the previous section about movie rating experiments, our algorithm outperforms TenAlt and FBCP in dense settings where a small portion of rating data is withheld as a test set, but the prediction accuracy steeply declines in a very sparse setting. However, the overall results demonstrate that our tensor completion method is competitive with existing tensor factorization based approaches, while maintaining a naive simplicity.

6.3 Additional Results from Experiments

We also compute the ϵ-risk achieved by each algorithm on the MovieLens data set when 10% of known ratings are withheld for evaluation. Figure $6-4$ shows that our method again outperforms classic collaborative filtering methods in the ϵ-risk. Since the rating scale is only the values 1 to 5, the scale is not fine enough to verify whether it decreases roughly as $O(\varepsilon^{-2})$.

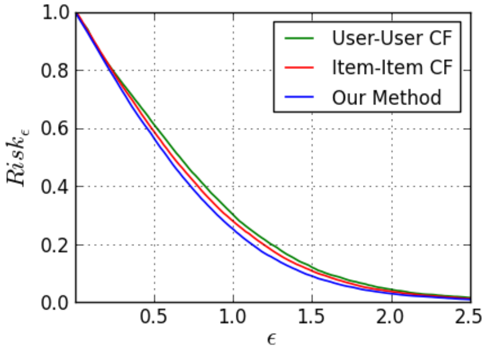

Figure 6-4: Risk$_\epsilon$ achieved on MovieLens data set (10% evaluation).

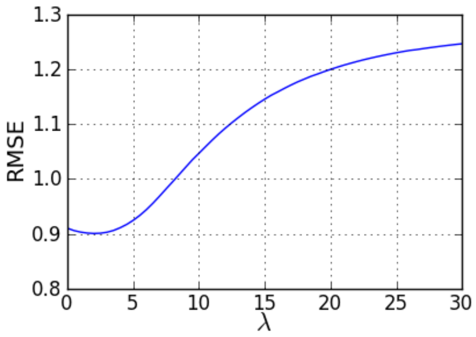

Figure 6-5: Effect of λ on RMSE from MovieLens data set (10% evaluation).

6.3.1 Choice of λ

The parameter λ corresponds to the inverse of variance in Gaussian kernel, in other words, it determines how much the algorithm emphasizes "close" neighbors with small sample variances over other neighbors. When $\lambda = 0$, the algorithm naively computes the average over all estimates, possibly allow "distant" neighbors with large sample variances to bias the estimate. When $\lambda = \infty$, the algorithm computes its estimate using only the closest neighbor with the minimum sample variance. This could also increase the noise and variance in the estimate, since it relies only on a few entries. This highlights the tradeoff between incorporating many datapoints into the estimate to reduce the noise through averaging, and emphasizing only the datapoints which seem to be closer in behavior to the target user or movie.

Figure 6-5 plots the RMSE as a function of λ for our algorithm applied to the MovieLens data set with 10% evaluation set. The figure shows that the performance of the algorithm first improves with increasing values of λ and then worsens as λ grows larger, with optimal $\lambda \approx 3$ (see Section 4.3 for further discussion on the trade-off).

In the caption of Figure 6-1, we reported the optimal value of λ for each size of the available data. We observe that when the percentage of ratings available to the algorithm decreases (i.e. the percentage of evaluation set increases), the optimal value of λ decreases, indicating that the algorithm needs to widen its circle to include estimates with larger sample variance. This intuitively makes sense, since the algorithm can depend more heavily on close neighbors as the matrix becomes denser, but needs to gather estimates more widely when the data is sparse.

6.3.2 Existence of Close Neighbors

For each (u, i) in the evaluation set (10%) for the MovieLens data set, we find the row v with minimum sample variance ($\min_v s^2_{uv}$) while requiring overlap of at least 5 ($\beta = 5$). Figure $6-6$ shows the distribution over the value of the minimum sample variances. Observe that the minimum sample variance $s^2_{uv} \leq 0.8$ for more than 90% of the entries, showing that it is unlikely for a user to have a closest neighbor with high sample variance, indicating that there is sufficient information to obtain good estimates through neighbor methods.

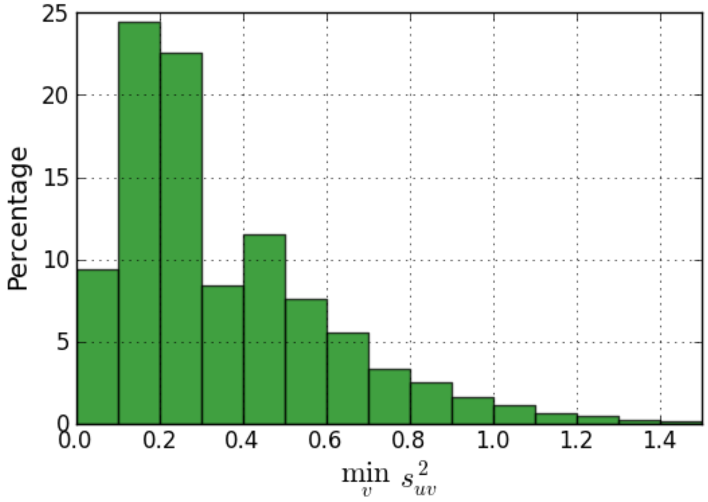

Figure 6-6: Distribution of minimum sample variance ($\beta = 5$).

We divided the entries into different buckets based on their minimum sample variances (intervals of width 0.1 as plotted in Figure 6-6). We computed the error for each bucket, with the prediction that for estimates such that the nearest neighbors have large sample variance, the error will also vary more widely. Recall that our algorithm computes the estimates by a weighted combination of many values, where the weights decay exponentially with the sample variance. Therefore, the minimum sample variance only indicates the lowest sample variance among all values incorporated into the estimate. However, we could observe that the minimum sample variance indeed provides a good indication of the reliability of the estimate.

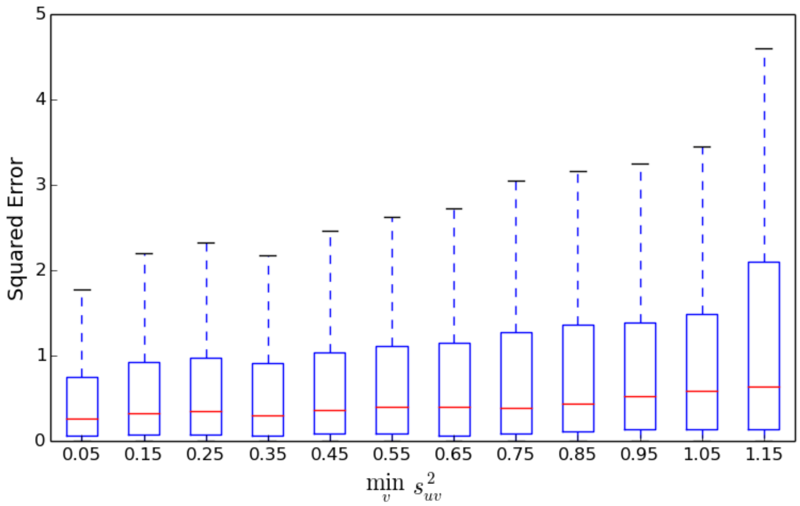

Figure 6-7: Variation of squared error across $\min_v s_{uv}^2$ buckets ($\beta = 5$). The red line plots the median, while the box shows the 25th and 75th percentile, and end of dashed lines extends to the most extreme data point within 1.5 interquartile range of the upper quartile.

Figure $6-7$ plots the squared error $(\hat{R}(u,i) - R(u,i))^2$ for each bucket. As predicted, the variance and variability of the prediction error indeed increase with the minimum sample variance, validating the theoretical prediction that the sample variance is an observable measure of the reliability of the estimate. In fact, this is quite useful in practice, since this implies that in addition to computing estimates for the missing entries, our algorithm can provide a confidence for each estimate obtained through a function of the sample variance of the entries involved in computing the final estimate. Note that attaining $Risk_\epsilon \leq \alpha$ is equivalent to acquiring an α-confidence interval for the estimate of length smaller than 2ϵ.

Chapter 7

Conclusion and Future Work

In this thesis, we provided a statistical framework for performing nonparametric regression over latent variable models. The investigation was motivated by recommender system applications. Inspired by local function approximation and kernel regression, we explored to construct a prediction algorithm which provides consistent estimates with provable performance guarantees. To overcome the challenge of unknown geometry of the latent space, we suggest to use a distance proxy which can be computed as a function of data. For example, the variance of difference between commonly observed entries in two rows (u, v) can mimic the squared L_2 distance between the latent features $x_1(u)$ and $x_2(v)$ of the rows. We proved that our framework can provide a prediction algorithm that is consistent for all Lipschitz functions, where the convergence rate depends on the model parameters. We also showed that our algorithm and analysis can be extended to higher-order tensor completion problems by flatteing a tensor to a matrix.

However, there are several interesting and important questions which we have not addressed in this thesis:

More general extension to higher dimension: Although we have shown in Section 4.2 that we can apply our matrix completion algorithm to tensors, this approach does not exploit the properties of tensor as a higher dimensional object. Recalling

Taylor's series approximation, we may approximate

$$A(u,i,t) \approx A(v,i,t) + A(u,j,t) + A(u,i,s) - 2A(v,j,s).$$

This alternative approach requires only 4 out of 8 points in the cube $\{u,v\} \times \{i,j\} \times \{t,s\}$ to make a prediction, which is much less as a ratio than 3 out of 4 required in the matrix case. However, it would be less likely to find a good neighbor v of u, which has a small variance of the difference $A(u,*,*) - A(v,*,*)$ as the dimensionality of slices increases from 1 to 2. It will be interesting to find a generalized extension of our blind regression framework in a higher-dimensional setting as well as to investigate the trade-off between the sample complexity and the accuracy of distance estimation.

Analysis for the algorithm with kernel weights: Theorem 1 shows the consistency of the nearest-neighbor algorithm; it is optimal when there is no noise. With the presence of noise, our analysis cannot surpass the Chebyshev bound for the noise. We briefly sketched the effect of averaging over rows in Section 4.3, and glimpsed the possiblility of smoothing out the noise. However, the examination is preliminary and the consistency of our algorithm with general user-item kernel weights is, yet, unclear. Therefore, it could be of interest to have an analysis on the algorithm with general kernel weights; it can also suggest a disciplined choice of the parameter λ.

Combining CF with content information: As discussed in the introduction, it is natural to use the content of data to make recommendations. In practice, recommendations are often made in a content-agnostic way via collaborative filtering, because such exogenous content information is not usually available. However, if we can find a systematic way to combine content information within the framework of collaborative filtering, the combination will yield more accurate and reliable recommendations. For example, information of users, such as age, gender, and geographic location, can be used to estimate the distance between two users in combination with the distance proxy computed from the variance of the difference in the ratings of commonly rated items.

Application to other types of problems: Lastly, we would like to point out that the underlying ideas of our blind regression are simple, but the results we obtained are powerful. It implies that our blind regression framework may possibly extend beyond the recommender system application - the main focus in this thesis. We believe that the concepts of blind regression as well as the 2-step regression framework itself can be extended to various other applications, especially where the latent variable model can be applied. The insights obtained in this thesis may find applications beyond.

Bibliography

[1] Animashree Anandkumar, Rong Ge, Daniel Hsu, Sham M Kakade, and Matus Telgarsky. Tensor decompositions for learning latent variable models. *The Journal of Machine Learning Research*, 15(1):2773–2832, 2014.

[2] Sanjeev Arora, Rong Ge, Ravindran Kannan, and Ankur Moitra. Computing a nonnegative matrix factorization–provably. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, pages 145–162. ACM, 2012.

[3] Sanjeev Arora, Rong Ge, and Ankur Moitra. Learning topic models–going beyond svd. In *Foundations of Computer Science (FOCS), 2012 IEEE 53rd Annual Symposium on*, pages 1–10. IEEE, 2012.

[4] Marko Balabanovic and Yoav Shoham. Fab: content-based, collaborative recommendation. *Communications of the ACM*, 40(3):66–72, 1997.

[5] Robert M Bell and Yehuda Koren. Lessons from the netflix prize challenge. *ACM SIGKDD Explorations Newsletter*, 9(2):75–79, 2007.

[6] Robert M Bell, Yehuda Koren, and Chris Volinsky. The bellkor solution to the netflix prize. 2007.

[7] James Bennett and Stan Lanning. The netflix prize. In *Proceedings of KDD cup and workshop*, page 35, 2007.

[8] Daniel Billsus and Michael J Pazzani. User modeling for adaptive news access. *User Modeling and User-Adapted Interaction*, 10(2):147–180, 2000.

[9] David Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003.

[10] Stéphane Boucheron, Gábor Lugosi, and Pascal Massart. Concentration inequalities using the entropy method. *Annals of Probability*, pages 1583–1614, 2003.

[11] John S Breese, David Heckerman, and Carl Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., 1998.

[12] Guy Bresler, George H Chen, and Devavrat Shah. A latent source model for online collaborative filtering. In *Advances in Neural Information Processing Systems*, pages 3347–3355, 2014.

[13] Guy Bresler, Devavrat Shah, and Luis F Voloch. Collaborative filtering with low regret. *arXiv preprint arXiv:1507.05371*, 2015.

[14] Deng Cai, Xiaofei He, Xiaoyun Wu, and Jiawei Han. Non-negative matrix factorization on manifold. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*, pages 63–72. IEEE, 2008.

[15] Emmanuel J Candès and Benjamin Recht. Exact matrix completion via convex optimization. *Foundations of Computational mathematics*, 9(6):717–772, 2009.

[16] Mukund Deshpande and George Karypis. Item-based top-n recommendation algorithms. *ACM Transactions on Information Systems (TOIS)*, 22(1):143–177, 2004.

[17] Maryam Fazel, Haitham Hindi, and Stephen P. Boyd. Log-det heuristic for matrix rank minimization with applications to hankel and euclidean distance matrices. In *American Control Conference, 2003. Proceedings of the 2003*, volume 3, pages 2156–2162. IEEE, 2003.

[18] Ravi S. Ganti, Laura Balzano, and Rebecca Willett. Matrix completion under monotonic single index models. In *Advances in Neural Information Processing Systems*, pages 1864–1872, 2015.

[19] David Goldberg, David Nichols, Brian M. Oki, and Douglas Terry. Using collaborative filtering to weave an information tapestry. *Commun. ACM*, 1992.

[20] Miha Grcar, Blaz Fortuna, Dunja Mladenic, and Marko Grobelnik. knn versus svm in the collaborative filtering framework. In *Data Science and Classification*, pages 251–260. Springer Berlin Heidelberg, 2006.

[21] Jonathan L Herlocker, Joseph A Konstan, Loren G Terveen, and John T Riedl. Evaluating collaborative filtering recommender systems. *Transactions on Information Systems*, 22(1):5–53, 2004.

[22] Will Hill, Larry Stead, Mark Rosenstein, and George Furnas. Recommending and evaluating choices in a virtual community of use. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 194–201. ACM Press/Addison-Wesley Publishing Co., 1995.

[23] Prateek Jain, Praneeth Netrapalli, and Sujay Sanghavi. Low-rank matrix completion using alternating minimization. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 665–674. ACM, 2013.

[24] Prateek Jain and Sewoong Oh. Provable tensor factorization with missing data. In *Advances in Neural Information Processing Systems*, pages 1431–1439, 2014.

[25] RH Keshavan, A Montanari, and S Oh. Matrix completion from a few entries. *Transactions on Information Theory*, 56(6), 2009.

[26] Tamara G Kolda and Brett W Bader. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009.

[27] Joseph A Konstan, Bradley N Miller, David Maltz, and Jonathan L Herlocker. Grouplens: applying collaborative filtering to usenet news. *Communications of the ACM*, 40(3):77–87, 1997.

[28] Yehuda Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 426–434. ACM, 2008.

[29] Yehuda Koren. The bellkor solution to the netflix grand prize. *Netflix prize documentation*, 81, 2009.

[30] Yehuda Koren and Robert Bell. Advances in collaborative filtering. In *Recommender Systems Handbook*, pages 77–118. Springer, 2015.

[31] Zhouchen Lin, Arvind Ganesh, John Wright, Leqin Wu, Minming Chen, and Yi Ma. Fast convex optimization algorithms for exact recovery of a corrupted low-rank matrix. *Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, 61, 2009.

[32] Greg Linden, Brent Smith, and Jeremy York. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*, 7(1):76–80, 2003.

[33] Ji Liu, Przemyslaw Musialski, Peter Wonka, and Jieping Ye. Tensor completion for estimating missing values in visual data. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 35(1):208–220, 2013.

[34] Zhang Liu and Lieven Vandenberghe. Interior-point method for nuclear norm approximation with application to system identification. *SIAM Journal on Matrix Analysis and Applications*, 31(3):1235 – 1256, 2010.

[35] YP Mack and Bernard W Silverman. Weak and strong uniform consistency of kernel regression estimates. *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete*, 61(3):405–415, 1982.

[36] Andreas Maurer. Concentration inequalities for functions of independent variables. *Random Structures & Algorithms*, 29(2):121–138, 2006.

[37] Andreas Maurer and Massimiliano Pontil. Empirical bernstein bounds and sample variance penalization. *arXiv preprint arXiv:0907.3740*, 2009.

[38] Rahul Mazumder, Trevor Hastie, and Robert Tibshirani. Spectral regularization algorithms for learning large incomplete matrices. *The Journal of Machine Learning Research*, 11:2287–2322, 2010.

[39] Rahul Mazumder, Trevor Hastie, and Robert Tibshirani. Spectral regularization algorithms for learning large incomplete matrices. *The Journal of Machine Learning Research*, 11:2287–2322, 2010.

[40] Sahand Negahban and Martin J Wainwright. Restricted strong convexity and weighted matrix completion: Optimal bounds with noise. *The Journal of Machine Learning Research*, 13(1):1665–1697, 2012.

[41] Xia Ning, Christian Desrosiers, and George Karypis. A comprehensive survey of neighborhood-based recommendation methods. In *Recommender Systems Handbook*, pages 37–76. Springer, 2015.

[42] Sewoong Oh and Devavrat Shah. Learning mixed multinomial logit model from ordinal data. In *Advances in Neural Information Processing Systems*, pages 595–603, 2014.

[43] Arkadiusz Paterek. Improving regularized singular value decomposition for collaborative filtering. In *Proceedings of KDD cup and workshop*, pages 5–8, 2007.

[44] Michael Pazzani and Daniel Billsus. Learning and revising user profiles: the identification of interesting web sites. *Machine Learning*, 27(3):313–331, 1997.

[45] Angelika Rohde, Alexandre B Tsybakov, et al. Estimation of high-dimensional low-rank matrices. *The Annals of Statistics*, 39(2):887–930, 2011.

[46] Ruslan Salakhutdinov, Andriy Minih, and Geoffrey Hinton. Restricted boltzmann machines for collaborative filtering. In *Proceedings of the 24th international conference on Machine learning*, pages 791–798. ACM, 2007.

[47] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*, pages 285–295. ACM, 2001.

[48] Upendra Shardanand and Pattie Maes. Social information filtering: algorithms for automating "word of mouth". In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 210–217. ACM Press/Addison-Wesley Publishing Co., 1995.

[49] Bao-Hong Shen, Shuiwang Ji, and Jieping Ye. Mining discrete patterns via binary matrix factorization. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 757–766. ACM, 2009.

[50] Nathan Srebro, Noga Alon, and Tommi S Jaakkola. Generalization error bounds for collaborative prediction with low-rank matrices. In *Advances In Neural Information Processing Systems*, pages 1321–1328, 2004.

[51] Gabor Takacs, Istvan Pilaszy, Bottyan Nemeth, and Domonkos Tikk. Scalable collaborative filtering approaches for large recommender systems. *The Journal of Machine Learning Research*, 10:623–656, 2009.

[52] Matt P Wand and M Chris Jones. *Kernel smoothing*. Crc Press, 1994.

[53] Qibin Zhao, Kiqing Zhang, and Andrzej Cichocki. Bayesian cp factorization of incomplete tensors with automatic rank determination. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 37(9):1751–1763, 2015.

[54] C. Lawrence Zitnick and Takeo Kanade. Maximum entropy for collaborative filtering. In *Proceedings of the 20th conference on Uncertainty in artificial intelligence*, pages 636–643. AUAI Press, 2004.

Appendix A

Geometry

A.1 Metric Space

The metric is a notion of separating one point from another, and a metric on a space induces topological properties like open and closed sets.

Definition 2 (Metric space)**.** *A metric space is an ordered pair (X, d) where X is a set and d is a metric on X, which is a function $d : X \times X \to \mathbb{R}$ such that the following holds for any $x, y, z \in X$:*

1. $d(x, y) \geq 0$.
2. $d(x, y) = 0$ *if and only if* $x = y$.
3. $d(x, y) = d(y, x)$.
4. $d(x, z) \leq d(x, y) + d(y, z)$.

Definition 3 (p-product metric)**.** *Let (X, d_X) and (Y, d_Y) be metric spaces and let $1 \leq p \leq +\infty$. Define the p-product metric d_p on the Cartesian product $X \times Y$ by*

$$d_p\left((x_1, y_1), (x_2, y_2)\right) := \left(d_X(x_1, x_2)^p + d_Y(y_1, y_2)\right)^{1/p} \quad for\ 1 \leq p < \infty;$$
$$d_\infty\left((x_1, y_1), (x_2, y_2)\right) := \max\left\{d_X(x_1, x_2), d_Y(y_1, y_2)\right\}.$$

Remark 1. An inner product $\langle \cdot, \cdot \rangle$ induces a norm $\| \cdot \|$ by letting $\|x\| := \langle x, x \rangle^{1/2}$, and a norm $\| \cdot \|$ induces a metric d by letting $d(x, y) := \|x - y\|$.

A.2 Lipschitz Continuity

For a map between two metric spaces, continuity preserves "closeness" or inseparability. Lipschitz continuity is a strong form of uniform continuity. Intuitively, a Lipschitz continuous map is limited in how fast it can expand.

Definition 4 (Lipschitz continuity)**.** *Given two metric spaces (X, d_X) and (Y, d_Y), a map $f : X \to Y$ is called (L-)Lipschitz continuous if there exists a real constant $L \geq 0$ such that*

$$d_Y\left(f(x_1), f(x_2)\right) \leq L d_X\left(x_1, x_2\right), \quad \forall x_1, x_2 \in X.$$

Any such L is referred to as a Lipschitz constant for the map f. The smallest constant is sometimes called the (best) Lipschitz constant.

Definition 5 (Hölder continuity)**.** *More generally, a map is said to be Hölder continuous of order $\alpha > 0$ on X if there exists a constant $M > 0$ such that*

$$d_Y\left(f(x_1), f(x_2)\right) \leq M d_X\left(x_1, x_2\right)^{\alpha}, \quad \forall x_1, x_2 \in X.$$

Lipschitz continuity is a special case of Hölder continuity with $\alpha = 1$.

Definition 6 (Bilipschitz)**.** *A map $f : X \to Y$ is called (L-)bilipschitz if there exists $L \geq 1$ with*

$$\frac{1}{L} d_X\left(x_1, x_2\right) \leq d_Y\left(f(x_1), f(x_2)\right) \leq L d_x\left(x_1, x_2\right), \quad \forall x_1, x_2 \in X.$$

A bilipschitz mapping is injective, and is in fact a homeomorphism onto its image. A bilipschitz mapping with $L = 1$ is an isometry.

A.3 Doubling Space

One way to define the dimension of a space is by quantifying how fast the volume of a ball in it grows as its radius increases. Doubling dimension is one measure for that, which also has connection to properties of the measures on the space.

Definition 7 (Doubling measure)**.** *A measure μ on a metric space X is said to be doubling if there is a constant $C > 0$ such that*

$$\mu\left(B(x, 2r)\right) \leq C\mu\left(B(x, r)\right), \quad \forall x \in X,\ \forall r > 0.$$

In this case, μ is said to be C-doubling.

Definition 8 (Doubling space)**.** *A metric space (X, d) is said to be doubling if there is a constant $M > 0$ such that for any $x \in X$ and $r > 0$, the ball $B(x, r)$ may be contained in a union of no more than M many balls of radius $r/2$. Here, $\log_2 M$ is referred to as the doubling dimension of X.*

A measure space that supports a C-doubling measure is necessarily a doubling metric space, where the doubling dimension depends on the constant C. Conversely, any complete doubling metric space supports a doubling measure.

Appendix B

Probability

B.1 Borel Probability Measures

B.1.1 Topological Spaces

Definition 9 (Topology)**.** *Let X be a set and let $\mathcal{T}$ be a family of subsets of X. Then $\mathcal{T}$ is called a topology of X if*

1. *$\phi, X \in \mathcal{T}$*

2. *Any union of elements of $\mathcal{T}$ is an element of $\mathcal{T}$.*

3. *Any finite intersection of elements of $\mathcal{T}$ is an element of $\mathcal{T}$.*

If $\mathcal{T}$ is a topology on X, then the pair $(X, \mathcal{T})$ is called a topological space. The elements in $\mathcal{T}$ are called open.

Remark 2. A metric d on X induces a topology $\mathcal{T}_d$ of which the open sets are all subsets that can be realized as the unions of open balls

$$B(x_0, r) := \{x \in X : d(x_0, x) < r\},$$

where $x_0 \in X$ and $r > 0$.

Definition 10. *A topological space is called separable if there exists a sequence* $\{x_n\}_{n=1}^{\infty}$ *of elements of the space such that every nonempty open subset of the space contains at least one element of the sequence.*

B.1.2 Borel Probability Measures

Definition 11 (σ-algebra)**.** *Let* X *be some set, and let* 2^X *represent its power set. A subset* $\Sigma \subset 2^X$ *is called a* σ*-algebra if it satisfies*

1. $X \in \Sigma$.

2. *If* $A \in \Sigma$, *then* $A^C = X \setminus A \in \Sigma$.

3. Σ *is closed under countable unions.*

Definition 12 (Borel σ-algebra)**.** *The Borel algebra* $\mathcal{B}(X)$ *on* X *is the smallest* σ*-algebra containing all open sets (equivalently, all closed sets).*

Proposition 8. $\mathcal{B}(X)$ *is the smallest sigma algebra with respect to which all continuous functions on* X *are measurable.*

Definition 13. *Let* (X, d) *be a metric space. A finite Borel measure on* X *is a map* $\mu : \mathcal{B}(X) \to [0, \infty)$ *such that*

1. $\mu(\phi) = 0$, *and*

2. *If* $A_1, A_2, \ldots \in \mathcal{B}(X)$ *are mutually disjoint, then* $\mu\left(\cup_{i=1}^{\infty}\right) = \sum_{i=1}^{\infty} \mu(A_i)$.

μ *is called a Borel probability measure if in addition* $\mu(X) = 1$.

Proposition 9. *Any finite Borel measure on* X *is regular, that is, for every* $B \in \mathcal{B}(X)$

$$\begin{aligned} \mu(B) &= \sup\{\mu(C) : C \subset B, \textit{closed}\} \quad \textit{(inner regular)} \\ &= \inf\{\mu(U) : U \supset B, \textit{open}\} \quad \textit{(outer regular)}. \end{aligned}$$

Definition 14. *A finite Borel measure* μ *on* X *is called tight if for every* $\epsilon > 0$ *there exists a compact set* $K \subset X$ *such that* $\mu(X \setminus K) < \epsilon$. *A tight finite Borel measure is also called a Radon measure.*

Theorem 3. *If (X, d) is a compact metric space, or (X, d) is a complete separable metric space, then every finite Borel measure on X is tight.*

B.2 Some Concentration Inequalities

B.2.1 Markov's and Chebyshev's Inequality

Theorem 4 (Markov's inequality). *If X is a nonnegative random variable, then for any $\epsilon > 0$,*

$$\mathbb{P}\left(X \geq \epsilon\right) \leq \frac{\mathbb{E}\left[X\right]}{\epsilon}.$$

Theorem 5 (Chebyshev's inequality). *For any random variable X and for any $\epsilon > 0$,*

$$\mathbb{P}\left(\left|X - \mathbb{E}\left[X\right]\right| \geq \epsilon\right) \leq \frac{Var(X)}{\epsilon^2}.$$

B.2.2 Chernoff Bounds

There are various forms of Chernoff bounds, each of which are tuned to different assumptions. The following theorem gives the bound for a sum of independent Bernoulli trials.

Theorem 6 (Chernoff bounds). *Let $X = \sum_{i=1}^{n} X_i$, where $X_i = 1$ with probability p_i, and $X_i = 0$ with probability $1-p_i$, and X_i's are independent. Let $\mu = \mathbb{E}\left[X\right] = \sum_{i=1}^{n} p_i$. Then*

1. *Upper tail: $\mathbb{P}\left(X \geq (1+\delta)\mu\right) \leq \exp\left(-\frac{\delta^2}{2+\delta}\mu\right)$ for all $\delta > 0$.*
2. *Lower tail: $\mathbb{P}\left(X \leq (1-\delta)\mu\right) \leq \exp\left(-\frac{\delta^2}{2}\mu\right)$ for all $0 < \delta < 1$.*

For $\delta \in (0, 1)$, we can combine the upper and lower tails to obtain the following simple bound:

Corollary 10. With the same assumptions as in Theorem 6,

$$\mathbb{P}\left(\left|X - \mu\right| \geq \delta\mu\right) \leq 2\exp -\frac{\mu\delta^2}{3} \quad \textit{for all } 0 < \delta < 1.$$

B.2.3 Concentration of Sample Variance of i.i.d. Bounded Random Variables

The results in this section is originally motivated by the entropy method. The entropy method is a novel way of deriving powerful inequalities based on logarithmic Sobolev inequalities, developed for proving sharp concentration bounds for maxima of empirical processes. It is shown that the methodology provides a general way of obtaining powerful results in a large variety of applications in [10].

Concentration Inequality for Self-Bounding Random Variables

Theorem 7. *[Theorem 7 in [37]; Theorem 13 in [36]] Let* $\mathbf{X} = (X_1, \dots, X_n)$ *be a vector of independent random variables with values in some set* $\mathcal{X}$*. For* $1 \leq k \leq n$ *and* $y \in \mathcal{X}$*, we use* $\mathbf{X}_k^y$ *to denote the vector obtained from* $\mathbf{X}$ *by replacing* X_k *by* y*. Suppose that* $a \geq 1$ *and a function* $Z = Z(\mathbf{X})$ *satisfies the inequalities*

$$Z(\mathbf{X}) - \inf_{y \in \mathcal{X}} Z(\mathbf{X}_k^y) \leq 1, \quad \forall k,$$
$$\sum_{k=1}^{n} \left(Z(\mathbf{X}) - \inf_{y \in \mathcal{X}} Z(\mathbf{X}_k^y) \right)^2 \leq aZ(\mathbf{X}),$$

almost surely. Then for $t > 0$*,*

1. *Upper tail:* $\mathbb{P}\left(Z - \mathbb{E}Z > t\right) \leq \exp\left(-\frac{t^2}{2a\mathbb{E}Z + at}\right)$.
2. *Lower tail:* $\mathbb{P}\left(Z - \mathbb{E}Z < -t\right) \leq \exp\left(-\frac{t^2}{2a\mathbb{E}Z}\right)$.

Variance of Bounded i.i.d. Random Variables

To begin with, for every $\mathbf{x} = (x_1, \dots, x_n) \in [0, 1]^n$, we let

$$m_n(\mathbf{x}) := \frac{1}{n} \sum_{i=1}^{n} x_i$$

and

$$V_n(\mathbf{x}) := \frac{1}{n(n-1)} \sum_{i,j=1}^{n} \frac{(x_i - x_j)^2}{2}.$$

It is easy to check that V_n is the same with the traditional sample variance:

$$\begin{aligned}
S_n^2 &= \frac{1}{n-1}\sum_{i=1}^n (x_i - m_n(\mathbf{x}))^2 \\
&= \frac{1}{n-1}\sum_{i=1}^n \left(x_i - \frac{1}{n}\sum_{j=1}^n x_j\right)^2 \\
&= \frac{1}{n-1}\sum_{i=1}^n \left[x_i^2 - \frac{2}{n}x_i\sum_{j=1}^n x_j + \frac{1}{n^2}\left(\sum_{j=1}^n x_j\right)^2\right] \\
&= \frac{1}{n-1}\sum_{i=1}^n x_i^2 - \frac{1}{n}\left(\sum_{i=1}^n x_i\right)^2 \\
&= \frac{1}{2(n-1)}\left(\sum_{i=1}^n x_i^2 - 2\sum_{i=1}^n x_i\sum_{j=1}^n x_j + \sum_{j=1}^n x_j^2\right) \\
&= \frac{1}{n(n-1)}\sum_{i,j=1}^n \frac{(x_i - x_j)^2}{2} \\
&= V_n(\mathbf{x}).
\end{aligned}$$

Before proceeding on to the concentration of sample variances, let us look at a technical lemma on conditional expectations.

Lemma 11. *[Lemma 8 in [37]] Let X, Y be i.i.d. random variables with values in an interval $[a, a+1]$. Then*

$$\mathbb{E}_X\left[\mathbb{E}_Y(X-Y)^2\right]^2 \le \frac{1}{2}\mathbb{E}(X-Y)^2.$$

Proof. Since X, Y are i.i.d., the RHS $\mathbb{E}(X-Y)^2 = \mathbb{E}\left[X^2 - XY\right] = \mathbb{E}\left[V_2(X,Y)\right]$. Meanwhile, one can compute the LHS $\mathbb{E}_X\left[\mathbb{E}_Y(X-Y)^2\right]^2 = \mathbb{E}\left[X^4 + 3X^2Y^2 - 4X^3Y\right]$. So, it suffices to show that $\mathbb{E}\left[g(X,Y)\right] \ge 0$ where

$$g(X,Y) = X^2 - XY - X^4 - 3X^2Y^2 + 4X^3Y.$$

With symmetrization, we obtain

$$\begin{aligned} g(X,Y)+g(Y,X) &= X^2 - XY - X^4 - 3X^2Y^2 + 4X^4Y \\ &\quad + Y^2 - XY - Y^4 - 3X^2Y^2 + 4Y^3X \\ &= (1+X-Y)(1+Y-X)(Y-X)^2 \\ &\geq 0, \quad \because X, Y \in [0,1]. \end{aligned}$$

Therefore, $\mathbb{E}\left[g(X,Y)\right] = \frac{1}{2}\mathbb{E}\left[g(X,Y)+g(Y,X)\right] \geq 0$, which completes the proof. □

When the random variables X and Y are uniformly distributed on a finite set, (treat x_i as a realization of X, and x_j as one of Y), Lemma 11 gives the following corollary.

Corollary 12. Suppose that $\{x_1, \ldots, x_n\} \subset [0,1]$. Then

$$\frac{1}{n}\sum_i \left(\frac{1}{n}\sum_j (x_i - x_j)^2\right)^2 \leq \frac{1}{2n^2}\sum_{i,j} (x_i - x_j)^2 .$$

B.2.4 Subgaussian and Subexponential Random Variables

Definition 15. *A random variable X with mean $\mu = \mathbb{E}\left[X\right]$ is said to be sub-Gaussian with variance proxy σ^2 if*

$$\mathbb{E}\left[\exp\left(s(X-\mu)\right)\right] \leq \exp\left(\frac{\sigma^2 s^2}{2}\right), \quad \forall s \in \mathbb{R}.$$

In this case we write $X \sim subG(\sigma^2)$. Note that $subG(\sigma^2)$ is a class of distributions.

Proposition 13. *The following are equivalent:*

1. *X is sub-Gaussian.*

2. *There exists $a > 0$ for which $\mathbb{P}\left(|X| > t\right) \leq 2e^{-at^2}$.*

3. *(If $\mathbb{E}\left[X\right] = 0$) There exist $b > 0$ for which $\mathbb{E}\left[e^{\lambda X}\right] \leq 2e^{\lambda^2 b}$ for all $\lambda \in \mathbb{R}$.*

4. There exists $c > 0$ *for which* $\|X\|_p \leq c\sqrt{p}, \quad \forall p \geq 1$ *where*

$$\|X\|_p = \mathbb{E}\left[|X|^p\right]^{1/p} = \sqrt{2}\left[\frac{\Gamma\left(\frac{1+p}{2}\right)}{\Gamma\left(\frac{1}{2}\right)}\right]^{1/p}.$$

Definition 16. *A random variable* X *with mean* $\mu = \mathbb{E}[X]$ *is said to be sub-exponential with nonnegative parameters* (λ, ν) *if*

$$\mathbb{E}\left[\exp\left(s(X-\mu)\right)\right] \leq \exp\left(\frac{\lambda^2 s^2}{2}\right), \quad \forall |s| \leq \frac{1}{\nu}.$$

Proposition 14. *Suppose that* X *is sub-exponential with parameters* (λ, ν)*. Then*

$$\mathbb{P}\left(X \geq \mu + t\right) \leq \begin{cases} \exp\left(-\frac{t^2}{2\lambda^2}\right), & \textit{if } 0 \leq t \leq \frac{\lambda^2}{\nu}; \\ \exp\left(-\frac{t}{2\nu}\right), & \textit{for } t > \frac{\lambda^2}{\nu}. \end{cases}$$

Hoeffding's Inequality

Theorem 8 (Hoeffding's inequality)**.** *Suppose that the variables* $X_1, \ldots, X_n$ *are independent, and* X_i *has mean* μ_i *and subGaussian with variance proxy* σ_i^2*. Then for all* $t \geq 0$*, we have*

$$\mathbb{P}\left(\sum_{i=1}^{n}(X_i - \mu_i) \geq t\right) \leq \exp\left\{-\frac{t^2}{2\sum_{i=1}^{n}\sigma_i^2}\right\}.$$

Bernstein's Inequality

Definition 17. *Given a random variable* X *with mean* $\mu = \mathbb{E}[X]$ *and variance* $\sigma^2 = \mathbb{E}[X^2] - \mathbb{E}[X]^2$*, we say that Bernstein's condition with parameter* b *holds if*

$$\left|\mathbb{E}\left[(X-\mu)^k\right]\right| \leq \frac{1}{2}k!\sigma^2 b^{k-2}, \quad \textit{for } k = 3, 4, \ldots$$

Theorem 9 (Bernstein's inequality)**.** *For any random variable satisfying the Bern-*

stein's condition, we have

$$\mathbb{E}\left[\exp\left(s(X-\mu)\right)\right] \leq \exp\left(\frac{s^2\sigma^2/2}{1-b|s|}\right), \quad \forall |s| < \frac{1}{b},$$

and

$$\mathbb{P}\left(|X-\mu| \geq t\right) \leq 2\exp\left(-\frac{t^2}{2(\sigma^2+bt)}\right), \quad \forall t \geq 0.$$

For example, we have the following simpler inequality for bounded random variables.

Corollary 15. If $X_1, \dots X_n$ are independent zero-mean r.v. such that $|X_i| \leq M$ almost surely, then for all t,

$$\begin{aligned}\mathbb{P}\left(\frac{1}{n}\sum_{i=1}^{n} X_i > t\right) &\leq \exp\left(-\frac{3n^2t^2}{2(3\sum_j \mathbb{E}\left[X_j^2\right] + Mnt)}\right)\\ &\leq \exp\left(-\frac{3nt^2}{6M^2+2Mt}\right).\end{aligned}$$