Learning with Matrix Factorizations

by

Nathan Srebro

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

Doctor of Philosophy in Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

August 2004 [September 2004]

Author ....
Department of Electrical Engineering and Computer Science
August 16, 2004

Certified by ....
Tommi S. Jaakkola
Associate Professor
Thesis Supervisor

Accepted by ....
Arthur C. Smith
Chairman, Department Committee on Graduate Students

Learning with Matrix Factorizations

by

Nathan Srebro

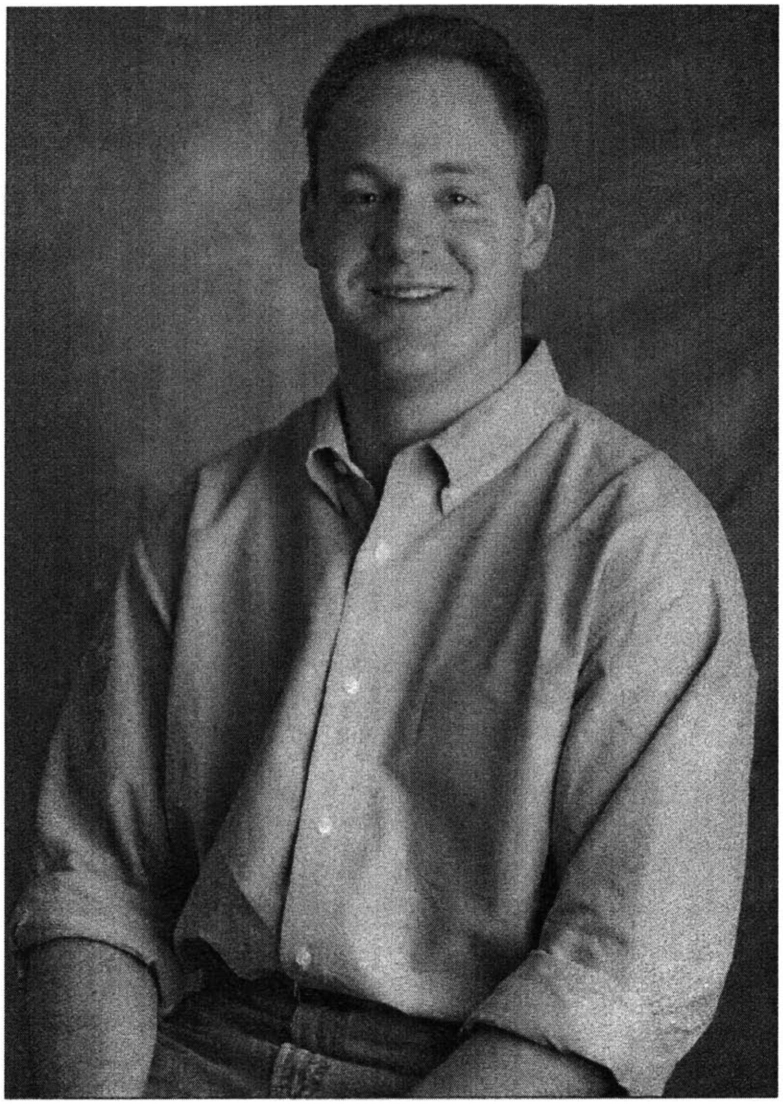
Submitted to the Department of Electrical Engineering and Computer Science
on August 16, 2004, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Computer Science

Abstract

Matrices that can be factored into a product of two simpler matrices can serve as a useful and often natural model in the analysis of tabulated or high-dimensional data. Models based on matrix factorization (Factor Analysis, PCA) have been extensively used in statistical analysis and machine learning for over a century, with many new formulations and models suggested in recent years (Latent Semantic Indexing, Aspect Models, Probabilistic PCA, Exponential PCA, Non-Negative Matrix Factorization and others). In this thesis we address several issues related to learning with matrix factorizations: we study the asymptotic behavior and generalization ability of existing methods, suggest new optimization methods, and present a novel maximum-margin high-dimensional matrix factorization formulation.

Thesis Supervisor: Tommi S. Jaakkola
Title: Associate Professor

In memory of Danny Lewin



May 14th, 1970 – September 11th, 2001

Acknowledgments

About five years ago, I first walked into Tommi's office to tell him about some ideas I had been thinking about. Slightly nervous, I laid out my ideas, and Tommi proceeded to show me how they were just a special case of a much broader context, which I got the impression was already completely understood, as Tommi corrected and generalized all the insights that just moments ago I thought were novel. After a few days I got over it and tried knocking on Tommi's always-open door again, hoping he would still be willing to take me as his student, perhaps working on something else. I was again surprised to be welcomed by Tommi for an unbounded time and even more surprised that at the end he inquired about the ideas I had mentioned two weeks before. Apparently he found them quite interesting... Since then, I've benefited over and over again from long meetings with Tommi, who showed me what I was *really* thinking about, and often enlightened me of how I *should* be thinking about it. I am truly thankful to Tommi for his advice, patience and flexibility.

Before, and also while, working with Tommi, I benefited greatly from David Karger's advice and guidance. Beyond the algorithmic and combinatorial techniques I learned from David (in his excellent courses and by working with him), I also learned a lot from the generous amounts of red ink David left on my papers, and the numerous talk rehearsals, late into the night in conference hotels. I thank David for continued advice, as my unofficial "theory adviser", even after my desertion to machine learning.

Throughout my graduate studies, Alan Willsky has been somewhat of a second adviser for me. The frequent meetings with Alan introduced structure to my research and forced me to summarize my work and think about what was important and what direction I was going in. It was those precious one-hour slots with Alan that were the mileposts of progress in my graduate research. I thank Alan for encouraging me, asking me all those good questions and pointing me in useful directions.

More recently, I benefited from my other committee members, Josh Tenenbaum and Tali Tishby. In stimulating and interesting conversations, Josh gave me many useful comments and asked enough questions for me to think about over the next several years. Tali provided a fresh, critical and often different viewpoint on many aspects of my work, which helped me look at it in a broader light. I thank Josh and Tali for agreeing to read my thesis, for attending my defense, and for their help along the way.

I also enjoyed the advice and assistance of many others at MIT. Eric Lander taught me to appreciate biology, and also to appreciate simplicity. In the theory group, Piotr Indyk, Madhu Sudan, Erik Demaine and Michel Goemns were always happy to chat, answer questions and direct me to references. Dmitry Panchenko guided me through the world of uniform bounds and concentration inequalities. In the last two years, I greatly enjoyed my many conversations with Michael Collins, who provided comments on my work, showed me some neat results, and gave me quite a bit of useful advice.

I had the pleasure of working with an old friend, Shie Mannor, and with newly acquired friends—John Barnett and Greg Shakhnarovich. I would also like to thank my collaborators on other projects during my studies at MIT: David Reich, Ziv Bar-Joseph, Ryan O'Donnell, Michael Wigler and his laboratory, David Gifford, Iftach Nachman and especially Percy Liang. I also appreciated discussions with, and many helpful comments from, Martin Wainwright, Polina Golland, Adrian Corduneanu, Lior Pachter, Mor Harcol-Balter, John Fisher, Ali Mohammad, Jason Rennie and many others. The single most beneficial aspect of my graduate studies was Tommi's weekly reading group meetings, and I am thankful to all who participated in them, and in particular to Erik Miller, Tony Jebara and Huizhen (Janey) Yu.

Outside of MIT, I am thankful to Adi Shraibman for discussions on the trace-norm and the max-norm, Noga Alon for a very fruitful meeting about counting sign configurations, Sam Roweis for pointing out Boyd's work, Peter Bickel for helpful discussions about consistency and matrix completion, Marina Meila for several interesting discussions about clustering and Ron Pinter for continued advice throughout the years. I am particularly indebted to Amir Ben-Dor and Zohar Yakhini, with whom I first worked on what turned out to be matrix factorization problems. Amir might not even know it, but the questions he asked me were the seeds of this thesis.

Keren Livescu deserves special and singular mention, not only for many discussion and comments, and for being an excellent friend and source of support, but for being a careful and dedicated editor. Keren reviewed almost every word I wrote in the past six years. Not only did she struggle with my own brand of spelling and "creative" grammatical constructions, but she also suggested many improvements on presentation and tried to help me make my writing clear and understandable. Unfortunately, the majority of this one last document did not pass Keren's critical eyes, and all the mistakes and obfuscations are purely mine.

I thank all the residents of the third floor at Tech Square for making it a vibrant and friendly research environment, and in particular Nicole, Sofia, Adam, Abhi, Adi, Yael, Eric, Alantha, Jon, Matthias, DLN, Tal, Anna, Marty, Eli Ben-Sassn, both Matts, Joan, Kathleen and of course Be (what would we do without you, Be?). I also enjoyed the company and camaraderie of many in the (former) AI lab, in particular Lilla, Agent K, Karen S, all Mikes and my fellow Nati. I thank all my officemates, Zuli, Brian, Mohammed, Yoav, both Adrians and especially both Johns (Dunagan and Barnett), who had lots of patience for my ramblings, ideas and gripes.

Mike Oltmans, beyond being the best apartment-mate one can hope for, is also responsible for organizing many of the IM sports I enjoyed at MIT. I would like to thank Mike, and also all the other IM sports organizers, captains and commissioners in the lab and at MIT (in particular Todd Stefanik and LP). Another great thing at MIT was the International Film club, which I am thankful to Krzysztof Gajos for creating.

Much of the work described in these pages was worked out on the Fung Wah and Lucky Star buses. I thank them for their cheap, efficient and often work-inducing buses (next time you take the Chinatown bus, please do pass this thanks to them).

Eli—I will thank you in person.

With sadness and sorrow, I am dedicating this thesis to the memory of Danny Lewin. I first met Danny at the Technion. He started MIT two years before me, and when he came to visit Israel he worked hard at convincing me to come study at MIT (I suspect he might have also done some convincing on the other end). Without Danny, I would not be at MIT, and would not be writing this thesis.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Factor models are often natural in the analysis of many kinds of tabulated data. This includes user preferences over a list of items (e.g. Section 5.1), microarray (gene expression) measurements (e.g. [4]), and collections of documents (e.g. [22]) or images (e.g. [47]). The underlying premise of such models is that important aspects of the data can be captured via a low-dimensional, or otherwise constrained, representation.

Consider, for example, a dataset of user preferences for movies. Such a data set can be viewed as a table, or matrix, with users corresponding to rows and movies to columns. The matrix entries specify how much each user likes each movie, i.e. the users' movie ratings.

The premise behind a factor model in this case is that there is only a small number of *factors* influencing the preferences, and that a user's preference vector is determined by how each factor applies to that user. In a linear factor model, each factor is a preference vector, and a user's preferences correspond to a linear combination of these factor vectors, with user-specific coefficients. These coefficients form a low-dimensional representation for the user.

Tabulated and viewed as a matrix, the preferences are modeled as the product of two smaller matrices: the matrix of per-user coefficients and the matrix of per-movie factors. Learning such a factor structure from the data amounts to *factorizing* the data matrix into two smaller matrices, or in the more likely case that this is impossible, finding a factorization that fits the data matrix well.

The factorization, or reduced (low dimensional) representation for each row (e.g. user), may be useful in several different ways:

Signal reconstruction The reduced representation, i.e. the per-row coefficients, may correspond to some hidden signal or process that is observed indirectly. Factor analysis was developed primarily for analyzing psychometric data: reconstructing the underlying characteristics of people that determine their observed answers to a series of questions. A more modern application can be found in gene expression analysis (e.g. [4]), where one aims at reconstructing cellular processes and conditions based on observed gene expression levels.

Lossy compression Traditional applications of Principal Component Analysis (PCA, see below) use the low-dimensional representation as a more compact representation that still contains most of the important information in the original high-dimensional input representation. Working with the reduced representation can reduce memory requirements, and more importantly, significantly reduce computational costs when the computational cost scales, e.g. exponentially, with the dimensionality.

Understanding structure Matrix factorization is often used in an unsupervised learning setting in order to model structure, e.g. in a corpus of documents or images. Each item in the corpus (document / image) corresponds to a row in the matrix, and columns correspond to item features (word appearances / pixel color levels). Matrix factorization is then used to understand the relationship between items in the corpus and the major modes of variation.

Prediction If the data matrix is only partially observed (e.g. not all users rated, or saw, all movies), matrix factorization can be used to predict unobserved entries (e.g. ratings).

Different applications of matrix factorization differ in the constraints that are sometimes imposed on the factorization, and in the measure of discrepancy between the factorization and the actual data (i.e. in the sense in which the factorization is required to "fit" the data).

If the factor matrices are unconstrained, the matrices which can be factored to two smaller matrices are exactly those matrices of rank bounded by the number of factors. Approximating a data matrix by an unconstrained factorization is equivalent to approximating the matrix by a low-rank matrix.

The most common form of matrix factorization is finding a low-rank approximation (unconstrained factorization) to a fully observed data matrix minimizing the sum-squared difference to it. Assuming the columns in the matrix are all zero mean (or correcting for this), this is known as Principal Component Analysis (PCA), as the factors represent the principal directions of variation in the data. Such a low-rank approximation is given in closed form in terms of the singular value decomposition (SVD) of the data matrix. The SVD essentially represents the eigenvalues and eigenvectors of the empirical covariance matrix of the rows, and of the columns, of the data matrix.

In many situations it is appropriate to consider other loss functions (e.g. when the targets are non-numerical, or corresponding to specific probabilistic models), or to impose constraints on the factorization (e.g. non-negativity [47] or sparsity). Such constraints can allow us to learn more factors, and can also be used to disambiguate the factors. Another frequent complication is that only some of the entries in the data matrix might be observed.

In this thesis we study various such generalizations. We study the problem of learning the factorizations, analyze how well we can learn them, and how they can be used for machine learning tasks.

We begin in Chapter 2 with a more thorough discussion of the various formulations of matrix factorization and the probabilistic models they correspond to.

In Chapter 3, we study the problem of *finding* a low-rank approximation subject to various measures of discrepancy. We focus on studying the resulting optimization problem: minimizing the discrepancy subject to low-rank constraints. We show that, unlike the sum-squared error, other measures of discrepancy lead to difficult optimization problems with non-global local minima. We discuss local-search optimization approaches, mostly based on minimizing the *weighted* sum-squared error (an interesting, and difficult, problem on its own right).

In Chapter 4 we aim at understanding the statistical properties of linear dimensionality reduction. We present a general statistical model for dimensionality reduction, and analyze the consistency of linear dimensionality reduction under various structural assumptions.

In Chapter 5 we focus on a specific learning task, namely collaborative filtering. We view this as completing unobserved entries in a partially observed matrix. We see how matrix factorization can be used to tackle the problem, and develop a novel approach, Maximum-Margin Matrix Factorization, with ties to current ideas in statistical machine learning.

In Chapter 6 we continue studying collaborative filtering, and present probabilistic post-hoc generalization error bounds for predicting entries in a partially observed data matrix. These are the first bounds of this type explicitly for collaborative filtering settings. We present bounds for prediction both using low-rank factorizations and using Maximum-Margin Matrix Factorization.

Notation

Throughout the thesis, we use uppercase letters to denote matrices, and lowercase letters for vectors and scalars. We use bold type to indicate random quantities, and plain roman type to indicate observed, or deterministic, quantities. The indexes i and j are used to index *rows* of the factored matrices and a and b to index *columns*. We use X_i to refer to the ith row of matrix X, but often treat it as a column vector. We use $X_{\cdot a}$ to refer to the ath column. The table below summarizes some of the notation used in the thesis.

$\lvert x\rvert$	The Euclidean (L_2) norm of vector x: $\lvert x\rvert = \sqrt{\sum_a x_a^2}$.
$\lvert x\rvert_\infty$	The L_∞ norm of vector x: $\lvert x\rvert_\infty = \max_a \lvert x_a\rvert$.
$\lvert x\rvert_1$	The L_1 norm of vector x: $\lvert x\rvert_1 = \sum_a \lvert x_a\rvert$.
$\lVert X\rVert_{\text{Fro}}$	The Frobenius norm of matrix X: $\lVert X\rVert_{\text{Fro}} = \sqrt{\sum_{ia} X_{ia}^2}$.
$\lVert X\rVert_2$	The spectral, or L_2 operator norm of matrix X, equal to the largest singular value of X: $\lVert X\rVert_2 = \max_{\lvert u\rvert=1} \lvert Xu\rvert$.
x', X'	Matrix or vector transposition
$X \otimes Y$	The element-wise product of two matrices: $(X \otimes Y)_{ia} = X_{ia} Y_{ia}$.
$X \bullet Y$	The matrix inner product of two matrices: $X \bullet Y = \operatorname{tr} X'Y$.
$X \succcurlyeq 0$	The square matrix X is positive semi-definite (all eigenvalues are non-negative).

Table 1.1: Linear algebra notation used in the thesis

Chapter 2

Matrix Factorization Models and Formulations

In this chapter we introduce the basic framework, models and terminology that are referred to throughout the thesis. We begin with a fairly direct statement of matrix factorization with different loss functions, mostly derived from probabilistic models on the relationship between the observations and the low-rank matrix (Section 2.1). In the remainder of the Chapter we discuss how these models, or slight variations of them, can arise from different modeling starting points and assumptions. We also relate the models that we study to other matrix factorization models suggested in the literature.

2.1 Low Rank Approximations

Consider tabulated data, organized in the observed matrix $Y \in \mathbb{R}^{n \times m}$, which we seek to approximate by a product of two matrices UV', $U \in \mathbb{R}^{n \times k}$, $V \in \mathbb{R}^{m \times k}$. Considering the rows of Y as data vectors Y_i, each such data vector is approximated by a linear combination $U_i V'$ of the the rows of V', and we can think of the rows of V' as *factors*, and the entries of U as coefficients of the linear combinations. Viewed geometrically, the data vectors $U_i \in \mathbb{R}^m$ are approximated by a k-dimensional linear subspace—the row subspace of V'.

This view is of course symmetric, and the columns of Y can be viewed as linear combinations of the columns of U. We will refer to both U and V as *factor matrices*.

If the factor matrices U and V are unconstrained, the matrices which can be exactly factored as $X = UV'$ are those matrices of rank at most k. Approximating a matrix Y by an unconstrained factorization is therefore equivalent to approximating it by a rank-k matrix[1].

An issue left ambiguous in the above discussion is the notion of "approximating" the data matrix. In what sense do we want to approximate the data? What is the measure of discrepancy between the data Y, and the model, X, that we want to minimize? Can this "approximation" be seen as fitting some probabilistic model?

2.1.1 Sum Squared Error

The most common, and in many ways simplest, measure of discrepancy is the sum-squared error, or the Frobenius distance (Frobenius norm of the difference) between X and Y:

$$\|Y - X\|^2_{\text{Fro}} = \sum_{ia}(Y_{ia} - X_{ia})^2 \tag{2.1}$$

We refer to the rank-k matrix X minimizing the Frobenius distance to Y as the Frobenius low-rank approximation.

In "Principal Component Analysis" (PCA) [44], an additional additive mean term is also allowed. That is, a data matrix $Y \in \mathbb{R}^{n \times m}$ is approximated by a rank-k matrix $X \in \mathbb{R}^{n \times m}$ and a row vector $\mu \in \mathbb{R}^m$, so as to minimize the Frobenius distance:

$$\sum_{ia}(Y_{ia} - (X_{ia} + \mu_a))^2. \tag{2.2}$$

The low-rank matrix X captures the principal directions of variation of the rows of Y from the mean row μ. In fact, it can be seen as the k-dimensional projection of the data that retains the greatest amount of variation.

Allowing a mean row term is usually straightforward. To simplify presentation, in this thesis we study homogeneous low-rank approximations, with no separate mean row term. Note also that by introducing a mean term, the problem is no longer symmetric, as rows

[1] In this Thesis, "rank-k matrices" refers to matrices of rank *at most* k

and columns are treated differently.

In terms of a probabilistic model, minimizing the Frobenius distance can be seen as maximum likelihood estimation in the presence of additive i.i.d. Gaussian noise with fixed variance. If we assume that we observe a random matrix generated as

$$\mathbf{Y} = X + \mathbf{Z} \tag{2.3}$$

where X is a rank-k matrix, and $\mathbf{Z}$ is a matrix of i.i.d. zero-mean Gaussians with constant variance σ^2, then the log-likelihood of X given the observation Y is:

$$\begin{aligned} \log \Pr(\mathbf{Y} = Y|X) &= -\frac{nm}{2} \ln 2\pi\sigma^2 - \sum_{ia} \frac{(Y_{ia} - X_{ia})^2}{2\sigma^2} \\ &= -\frac{1}{2\sigma^2} \|Y - X\|_{\text{Fro}} + \text{Const} \end{aligned} \tag{2.4}$$

Maximizing the likelihood of X is equivalent to minimizing the Frobenius distance.

In Section 4.2 we discuss how minimizing the Frobenius distance is appropriate also under more general assumptions.

The popularity of using the Frobenius low-rank approximation is due, to a great extent, to the simplicity of computing it. The Frobenius low-rank approximation is given by the k leading "components" of the singular value decomposition Y. This well-known fact is reviewed in Section 3.1.

As discussed in the remainder of Chapter 3, finding low-rank approximations that minimize other measures of discrepancy is not as easy. Nevertheless, the Gaussian noise model is not always appropriate, and other measures of discrepancy should be considered.

2.1.2 Non-Gaussian Conditional Models

Minimizing the Frobenius distance between a low-rank matrix X and the data matrix Y corresponds to a probabilistic model in which each entry Y_{ia} is seen as a single observation of a random variable $\mathbf{Y}_{ia} = X_{ia} + \mathbf{Z}_{ia}$, where $\mathbf{Z}_{ia} \sim \mathcal{N}(0, \sigma^2)$ is zero-mean Gaussian error with fixed variance σ^2. This can be viewed as specifying the conditional distribution of $\mathbf{Y}|X$, with $\mathbf{Y}_{ia}|X_{ia}$ following a Gaussian distribution with mean X_{ia} and some fixed variance σ^2, independently for entries (i, a) in the random matrix $\mathbf{Y}$.

Other models on the conditional distribution $\mathbf{Y}_{ia}|X_{ia}$ might be appropriate [31]. Such models are essentially specified by a single-parametric family of distributions $p(y; x)$.

A special class of conditional distributions are those that arise from additive, but not necessarily Gaussian, noise models, where $\mathbf{Y}_{ia} = X_{ia} + \mathbf{Z}_{ia}$, and $\mathbf{Z}_{ia}$ are independent and follow a fixed distribution. We refer to these as "additive noise models", and they receive special attention in some of our studies.

It is often appropriate to depart from an additive noise model, $\mathbf{Y} = X + \mathbf{Z}$, with $\mathbf{Z}$ independent of X. This is the case, for example, when the noise is multiplicative, or when the observations in Y are discrete.

Logistic Low Rank Approximation

For example, consider modeling an observed classification matrix of binary labels. It is possible to use standard low-rank approximation techniques by embedding the labels as real values (such as zero-one or ± 1) and minimizing the quadratic loss, but the underlying probabilistic assumption of a Gaussian model is inappropriate. Seeking an appropriate probabilistic model, a natural choice is a logistic model parameterized by a low-rank matrix $X \in \Re^{n \times m}$, such that $\Pr(\mathbf{Y}_{ia} = +1|X_{ia}) = g(X_{ia})$ independently for each ia, where g is the logistic function $g(x) = \frac{1}{1+e^{-x}}$. One then seeks a low-rank matrix X maximizing the likelihood $\Pr(\mathbf{Y} = Y|X)$. Such low-rank logistic models were recently studied by Schein *et al* [62].

Exponential PCA

Logistic low-rank approximation is only one instance of a general approach studied by Collins *et al* [19] as "Exponential-PCA". These are models in which the conditional distributions $\mathbf{Y}_{ia}|X_{ia}$ form an exponential family of distributions, with X_{ia} being the natural parameters.

Definition 1 (Exponential Fammily of Distributions). *A family of distributions $p(y; x)$, parametrized by a vector $x \in \mathbb{R}^d$, is an exponential family, with x being the* natural parameters, *if the distributions (either the density for continuous* y *or the probability mass for discrete* y*) can be be written as:*

$$p(y;x) = e^{\sum_a \phi_a(y)x_a + F(x) + G(y)}$$

for some real-valued features ϕ*, and real-valued functions* F *and* G*. The* mean parameterization *of the distributions family is given by* $\mu(x) = \mathbf{E}\left[\phi(\mathbf{y}); x\right]$.

In this thesis, we will usually refer to exponential families of distributions of random vectors $\mathbf{y} \in \mathbb{R}^m$, where the features are simply the coordinates of the vectors $\phi_a(y) = y_i$.

In Exponential-PCA the distributions $\mathbf{Y}_{ia}|X_{ia}$ form a single-parametric exponential family of distributions of (one dimensional) random variables $\mathbf{Y}_{ia}$, where the single parameter for $\mathbf{Y}_{ia}$ is given by X_{ia}. That is:

$$p(Y_{ia}|X_{ia}) = e^{Y_{ia}X_{ia} + F(X_{ia}) + G(Y_{ia})} \tag{2.5}$$

for some real-valued functions F and G.

Other than logistic low-rank approximation, other examples of exponential PCA include binomial and geometric conditional distributions. The Gaussian additive noise model can also be viewed as an exponential family, and it is the only conditional model which both corresponds to additive noise, and forms an exponential family.

Gous [32] also discusses exponential conditional models, viewed as selecting a linear subspace in the manifold of natural parameters for data-row distributions.

2.1.3 Other Loss Functions

So far, we have considered maximum likelihood estimation, and accordingly discrepancies that correspond to the negative log-likelihood of each entry in X:

$$\begin{aligned} \mathcal{D}(X;Y) &= \sum_{ia} \text{loss}(X_{ia}; Y_{ia}) \\ \text{loss}(x;y) &= -\log \Pr\left(y|x\right), \end{aligned} \tag{2.6}$$

up to scaling and constant additive terms. Departing from maximum likelihood estimation, it is sometimes desirable to discuss the measure of loss directly, without deriving it from a probabilistic model. For example, when the observations in Y are binary class labels, instead of assuming a logistic, or other, probabilistic model, loss functions commonly used for standard classifications tasks might be appropriate. These include, for example a zero/one-sign loss, matching positive labels with positive entries in X:

$$\text{loss}(x;y) = \begin{cases} 0 & \text{if } xy > 0 \\ 1 & \text{otherwise} \end{cases} \tag{2.7}$$

or convex loss functions such as the hinge loss often used in SVMs:

$$\text{loss}(x;y) = \begin{cases} 0 & \text{if } xy > 1 \\ 1 - xy & \text{otherwise} \end{cases} \tag{2.8}$$

2.1.4 Constrained Factorizations

We have so far referred only to *unconstrained* matrix factorizations, where U and V are allowed to vary over all matrices in $\mathbb{R}^{n\times k}$ and $\mathbb{R}^{m\times k}$ respectively, and so $X = UV'$ is limited only by its rank. It is sometimes appropriate to constrain the factor matrices. This might be necessary to match the interpretation of the factor matrices (e.g. as specifying probability distributions, see Section 2.3.1) or in order to reduce the complexity of the model, and allow identification of more factors. Imposing constraints on the factor matrices can also remove the degrees of freedom on the factorization UV' of a reconstructed X, and aid in interpretation.

Lee and Seung studied various constraints on the factor matrices including non-negativity constraints (Non-Negative Matrix Factorization [47]) and stochasticity constraints [46]. For a discussion of various non-negativity and stochasticity constraints, and the relationships between them, see Barnett's work [9].

2.2 Viewing the Matrix as an I.I.D. Sample

In the probabilistic view of the previous section, we regarded the entire matrix X as parameters, and estimated them according to a single observation Y of the random matrix $\mathbf{Y}$. The number of parameters is linear in the data, and even with more data, we cannot hope to estimate the parameters (entries in X) beyond a fixed precision. What we *can* estimate with more data rows is the rank-k row-space of X.

Here, we discuss probabilistic views in which the matrix Y is taken to be a sample of i.i.d. observations of a random vector $\mathbf{y}$. That is, each row y of Y is an independent observation of the random vector $\mathbf{y}$.

Focusing on a Gaussian additive noise model, the random vector $\mathbf{y}$ is modeled as

$$\mathbf{y} = \mathbf{x} + \mathbf{z} \tag{2.9}$$

where $\mathbf{x}$ is the low-rank "signal", to which Gaussian white noise $\mathbf{z} \sim \mathcal{N}(0, \sigma^2 I_m)$ is added. The main assumption here is that the signal $\mathbf{x}$ occupies only a k-dimensional subspace of $\mathbb{R}^m$. In other words, we can write $\mathbf{x} = \mathbf{u} V'$, where $V' \in \mathbb{R}^{k \times m}$ spans the support subspace of $\mathbf{x}$, and $\mathbf{u}$ is a k-dimensional random vector. The model (2.9) can thus be written as:

$$\mathbf{y} = \mathbf{u} V' + \mathbf{z} \tag{2.10}$$

where $\mathbf{u} \in \mathbb{R}^k$ and $V' \in \mathbb{R}^{k \times m}$.

In the previous section, we treated $\mathbf{u}$ as parameters, with a separate parameter vector u (a row of U) for each observed row y of Y. Here, we treat $\mathbf{u}$ as a random vector. A key issue is what assumptions are made on the distribution of $\mathbf{u}$.

2.2.1 Probabilistic Principal Component Analysis

Imposing a fixed, or possibly parametric, distribution on $\mathbf{u}$ yields a standard parametric model for $\mathbf{y}$. The most natural choice is to assume $\mathbf{u}$ follows a k-dimensional Gaussian distribution [74]. This choice yields a Gaussian distribution for $\mathbf{x}$, with a rank-k covariance matrix. Note that without loss of generality, we can assume $\mathbf{u} \sim \mathcal{N}(0, I_k)$, as the covariance matrix can be subsumed in the choice of V. As $\mathbf{z} \sim \mathcal{N}(0, \sigma^2 I_m)$, the observed random vector $\mathbf{y}$ also follows a Gaussian distribution:

$$\mathbf{y} \sim \mathcal{N}(0,\ VV' + \sigma^2 I_m). \tag{2.11}$$

This is, then, a fully parametric model, where the parameters are the rank-k matrix VV' and the noise covariance σ^2. Using this view, low-rank approximation becomes a standard problem of estimating parameters of a distribution given independent repeated observations. Interestingly, whether σ^2 is known or unknown, maximum likelihood estimation of the parameters agrees with Frobenius low-rank estimation (i.e. PCA) [74]: the maximum likelihood estimator of VV' under model (2.11) is the "covariance" $\frac{1}{n}X'X$ where X is the Frobenius low-rank approximation.

2.2.2 A Non-Parametric Model

The above analysis makes a significant additional assumption beyond those of Section 2.1—we assume a very specific form on the distribution of the "signal" $\mathbf{x}$, namely that it is Gaussian. This assumption is not necessary. A more general view, imposing less assumptions, is to consider the model (2.10) where $\mathbf{u}$ is a random vector that can follow any distribution. Considering the distribution over $\mathbf{u}$ as unconstrained, non-parametric nuisance, the maximum likelihood estimator of Section 2.1 can be seen as a maximum likelihood estimator for the "signal subspace" V—the k-dimensional subspace in $\mathbb{R}^m$ that spans the support of $\mathbf{x}$. The model class is non-parametric, yet we still desire, and are able, to estimate this parametric aspect of the model.

The discussion in this section so far refers to a Gaussian additive conditional model, but applies equally well to other conditional models.

The difference between this view and that of Section 2.1, is that here the rows of Y are viewed as i.i.d. observations, and the estimation is of finite number of parameters. We can therefore discuss the behavior of the estimator when the sample size (number of rows in Y) increases.

It is important to note that what we are estimating is the *subspace* V which spans the support of $\mathbf{x}$. Throughout this thesis, we overload notation and use V to denote both a matrix and the column-space it spans. However, we cannot estimate the *matrix* V for which $\mathbf{x} = \mathbf{u}V'$, as the estimation would be only up to multiplication by an invertible $k \times k$ matrix.

2.2.3 Canonical Angles

In order to study estimators for a subspace, we must be able to compare two subspaces. A natural way of doing so is through the *canonical angles* between them [71]. Define the angle between a vector v_1 and a subspace V_2 to be the minimal angle between v_1 and any $v_2 \in V_2$. The first (largest) canonical angle between two subspaces is then the maximal angle between a vector in $v_1 \in V_1$ and the subspace V_2. The second largest angle is the maximum over all vectors orthogonal to the v_1, and so on[2]. Computationally, if the columns of the matrices V_1 and V_2 form orthonormal bases of subspaces V_1 and V_2, then the cosines of the canonical angles between V_1 and V_2 are given by the singular values of $V_1'V_2$.

2.3 Low Rank Models for Occurrence, Count and Frequency Data

In this section we discuss several low-rank models of co-occurrence data, emphasizing the relationships, similarities and differences between them.

Entries in a co-occurrence matrix Y describe joint occurrences of row "entities" and column "entities". For example, in analyzing a corpus of text documents, the rows correspond to documents and columns to words, and entries of the matrix describe the (document, word) co-occurrence: entry Y_{ia} describes the occurrence of words a in document i. The order of words in a document is ignored, and the documents are considered as "bags of words". Entries Y_{ia} of the co-occurrence matrix Y can be binary, specifying whether the word a occurred in the document i or not; they can be non-negative integers specifying the number of times the word a occurred in the document i; or they can be reals in the

[2] if there are multiple vectors achieving the maximum angle, it does not matter which of them we take, as this will not affect subsequent angles

interval $[0, 1]$ specifying the frequency in which the word a occurs in the document i, or the frequency of the word-document co-occurrence (that is, the number of times the word a appeared in this document i divided by the total number of words in all documents).

In traditional Latent Semantic Analysis [22], a low-rank approximation of the co-occurrence matrix Y minimizing the sum-squared error is sought. However, this analysis does not correspond to a reasonable probabilistic model. In this section, we consider various probabilistic models.

We will see that models for vary in two significant aspects. The first is the whether each entry is seen as an observation of an independent random variable, or whether the entire matrix is seen as an observation of a joint distribution with dependencies between entries. The second is whether a low-rank structure is sought for the mean parameters or the natural parameters of the distribution.

2.3.1 Probabilistic Latent Semantic Analysis: The Aspect Model

We will first consider a fully generative model which views the "factors" as latent variables. In such a model, the observed random variables are the row and column indexes, i and a. The generative model describes a joint distribution over (i, a). The matrix Y is seen as describing some fixed number N of independent repeated observations of the random variable pair (i, a), where Y_{ia} is a non-negative integer describing the number of occurrences of $\mathrm{i} = i, \mathrm{a} = a$. The matrix Y is then an observation of a multinomially distributed random matrix $\mathbf{Y}$.

In the "aspect" model [37, 38], a latent (hidden) variable t is introduced, taking k discrete values, $\mathrm{t} \in [k]$. The variable t can be interpreted as a "topic". The constraint on the generative model for $(\mathrm{i}, \mathrm{t}, \mathrm{a})$ is that i and a are independent given t. This can be equivalently realized by any of the following directed graphical models:

$$\begin{gathered}\mathbf{i} \longrightarrow \mathbf{t} \longrightarrow \mathbf{a} \\ p(i,t,a) = p(i)p(t|i)p(a|t) = p(i,t)p(a|t)\end{gathered} \tag{2.12}$$

$$\begin{gathered}\mathbf{i} \longleftarrow \mathbf{t} \longrightarrow \mathbf{a} \\ p(i,t,a) = p(t)p(i|t)p(a|t)\end{gathered} \tag{2.13}$$

$$\begin{gathered}\mathbf{i} \longleftarrow \mathbf{t} \longleftarrow \mathbf{a} \\ p(i,t,a) = p(a)p(t|a)p(i|t)\end{gathered} \tag{2.14}$$

If we summarize the joint and conditional distributions in matrices:

$$X_{ia} = p(i,a) \qquad U_{it} = p(i,t) \qquad V_{at} = p(a|t) \tag{2.15}$$

we can write the joint distribution of $\mathbf{a}, \mathbf{i}$ as a product of two matrices with an inner dimension of k. The model imposes a rank-k constraint on the joint distribution of $\mathbf{a}, \mathbf{i}$. The constraint is actually a bit stronger, as the factorization of the joint distribution is to matrices which represent probability distributions, and must therefore be non-negative. What we are seeking is therefore a non-negative matrix factorization $X = UV'$ ([47], see Section 2.1.4) that is a distribution, i.e. such that $\sum_{ia} X_{ia} = 1$.

Given a count matrix Y, we seek a distribution matrix X with such a non-negative factorization $X = UV'$, maximizing the log-likelihood:

$$\log P(Y|X) = \sum_{ia} Y_{ia} \log X_{ia} \tag{2.16}$$

corresponding to an element-wise loss of

$$\text{loss}(x; y) = -y \log x \tag{2.17}$$

As in Section 2.1, the low-rank matrix X is interpreted as a parameter matrix. However, in the conditional models of Section 2.1, each entry Y_{ia} of Y was generated *independently* according to the parameter X_{ia}. Here, the parameters X, together with total count $N =$

$\sum_{ia} Y_{ia}$, specify a multinomial distribution over the entries in the matrix $\mathbf{Y}$, and different entries are not independent.

2.3.2 The Binomial and Bernoulli Conditional Models

Consider the binomial conditional model:

$$\mathbf{Y}_{ia}|X_{ia} \sim \text{Binom}(N, X_{ia}) \tag{2.18}$$

The marginal distribution of each entry $\mathbf{Y}_{ia}$ in this model, and in the multinomial aspect model, is the same. The difference between the two models is that in the binomial conditional model (2.18) each *entry* is independent, and the total number of occurrences is equal to N only on average (assuming $\sum_{ia} X_{ia} = 1$), while in the multinomial aspect model each *occurrence* is independent, and the number of occurrences is exactly N. Conditioned on the number of occurrences ($\sum_{ia} \mathbf{Y}_{ia}$) being exactly N, the two models agree. Since the total number of occurrences is tightly concentrated around N, the two models are extremely similar, and can be seen as approximations to one another.

We further note that if $NX_{ia} \ll 1$ for all ia, we will usually have $\mathbf{Y}_{ia} \in \{0, 1\}$, and the binomial conditional model (2.18) can be approximated by the Bernoulli conditional model:

$$\mathbf{Y}_{ia}|X_{ia} = \begin{cases} 0 & \text{with probability} 1 - NX_{ia} \\ 1 & \text{with probability} NX_{ia} \end{cases} \tag{2.19}$$

Mean Parameters and Natural Parameters

It is important to note the difference between Bernoulli conditional model (2.19) and Logistic Low Rank Approximation discussed in Section 2.1.2. In both models, the family of conditional distributions $p(y; x)$ includes the same distributions—all distributions of a single binary value. However, in Logistic Low Rank Approximation, entries X_{ia} are the *natural* parameters to the single parametric exponential family of distributions $p(y; x)$, whereas in Bernoulli conditional models, the entries X_{ia} are *mean* parameters. The difference then, is whether we seek a low rank subspace in the mean parameterization or in the natural parameterization.

Although less commonly used, another instance of Exponential PCA (models where X_{ia} serve as natural parameters) are low-rank models with Binomial conditional models, where X_{ia} is the *natural* parameter to the Binomially distributed $\mathbf{Y}_{ia}$. In the binomial conditional model as discussed before, as well as in the multinomial aspect model, the entries X_{ia} are scaled mean parameters, and we have:

$$X_{ia} = \frac{1}{N}\mathbf{E}\left[\mathbf{Y}_{ia}; X_{ia}\right] \tag{2.20}$$

Another related loss function for binary data is the hinge loss, defined in equation (2.8). The hinge loss and the logistic loss (i.e. the loss equal to the negative log likelihood for the logistic conditional model) are actually very similar—both are convex upper bounds (after appropriate scaling) on the zero-one loss (equation (2.7)), and have the same asymptotic behavior, while differing in their local behavior around zero (see e.g. [10] for a discussion on different convex loss functions).

2.3.3 KL-Divergence Loss

So far in this Section, the data matrix Y was taken to be a matrix of co-occurrence counts. A related loss function, which is appropriate for non-negative real-valued data matrices Y was suggested by Lee and Seung in their work on Non-Negative Matrix Factorizations (NMF) [48]. The loss is a "corrected" KL-divergence between unnormalized "distributions" specified by X and Y:

$$\begin{aligned} \mathcal{D}(X; Y) &= \sum_{ia} \text{loss}(X_{ia}; Y_{ia}) \\ \text{loss}(x; y) &= y \log \frac{y}{x} - y + x \end{aligned} \tag{2.21}$$

When X and Y specify distributions over index pairs, i.e. $\sum_{ia} X_{ia} = \sum_{ia} Y_{ia} = 1$, the discrepancy (2.21) is exactly the KL-divergence between the two distributions. Furthermore, when X specified a distribution ($\sum_{ia} X_{ia} = 1$) the discrepancy (2.21) agrees, up to an additive term independent of X, with the multinomial maximum likelihood loss (2.17).

Recalling that the factorization of the joint distribution in the aspect model is a non-negative matrix factorization, we see that Probabilistic Latent Semantic Analysis and Non-Negative Matrix Factorization with the KL-loss (2.21) are almost equivalent: the only difference is the requirement $\sum X_{ia} = 1$. Buntine [18] discusses this relationship between pLSA and NMF.

2.3.4 Sufficient Dimensionality Reduction

Also viewing the data matrix Y as describing the joint distribution of $(\mathbf{i}, \mathbf{a})$, Globerson and Tishby [28] arrive at low-rank approximation from an information-theoretic standpoint. In their *Sufficient Dimensionality Reduction* (SDR) formulation, one seeks the k features of $\mathbf{i}$ that are most informative about $\mathbf{a}$ (for a fixed, predetermined, k). Globerson and Tishby show that the features U of $\mathbf{i}$ most informative about $\mathbf{a}$ are dual to the features V of $\mathbf{a}$ most informative about $\mathbf{i}$, and together they specify a joint distribution

$$p_X(i, a) \propto e^{X_{ia}} \qquad X = UV'. \tag{2.22}$$

The most informative features U and V correspond to the joint distribution of the form (2.22) minimizing the KL divergence from the actual distribution given by Y. SDR is therefore equivalent to finding the rank-k matrix X minimizing the KL-divergence from Y, i.e. minimizing:

$$\begin{aligned} D(Y \| p_X) &= \sum_{ia} Y_{ia} \log \frac{Y_{ia}}{p_X(i,a)} \\ &= \sum_{ia} -Y_{ia} \log p_X(i,a) + \text{Const} \\ &= \sum_{ia} -Y_{ia} \log \frac{e^{X_{ia}}}{\sum_{jb} e^{X_{jb}}} + \text{Const} \end{aligned} \tag{2.23}$$

SDR and pLSA are therefore similar in that both seek a low-rank representation of a joint distribution on $\mathbf{i}, \mathbf{a}$ which minimizes the KL-divergence from the specified distribution Y. That is, in both cases we seek the distribution "closest" to Y (in the same sense of minimizing the KL-divergence from Y, also referred to as *projecting* Y) among distributions in a limited class of distributions parameterizes by rank-k matrices. The difference between SDR and pLSA is in how the low-rank matrix X parametrizes the joint distribution, and therefore in the resulting limited class of distributions.

2.3.5 Subfamilies of Joint Distributions

Consider the nm "indicator" features of the random variables $(\mathtt{i}, \mathtt{a})$:

$$\phi_{ia}(\mathtt{i}, \mathtt{a}) = \begin{cases} 1 & \text{if } \mathtt{i} = i \text{ and } \mathtt{a} = a \\ 0 & \text{otherwise} \end{cases} . \tag{2.24}$$

The family of all joint distributions of $(\mathtt{i}, \mathtt{a})$ is an exponential family with respect to these features. In SDR, we project Y to the subfamily of distributions where the *natural* parameters (with respect to these indicator features) form a low-rank matrix. In pLSA, we project Y to the subfamily of distributions where the *mean* parameters form a low-rank matrix. Note that neither of these subfamilies is an exponential family itself!

It is important to note that although SDR can be seen as the problem of finding a low-rank matrix minimizing some discrepancy to the target Y (namely the discrepancy given by (2.23)), unlike all previous models that we discussed, this discrepancy does *not* decompose to a sum of element-wise losses. This is because the normalization factor $\frac{1}{\sum_{jb} e^{X_{jb}}}$ appearing inside the logarithm of each term of the sum, depends on *all* the entries in the matrix. In pLSA, and mean parameter models in general, this normalization can be taken care of by requiring the global constraint $\sum_{ia} X_{ia} = 1$. However, in low-rank natural parameter models, this is not possible as every two matrices correspond to different probability distributions.

In Tishby and Globerson's formulation of SDR, the matrix X was not precisely a rank-k matrix, and additional constant-row and constant-column terms were allowed, corresponding to allowing information from the $\mathtt{i}$ and $\mathtt{a}$ marginals in the information theoretic formulation. This is a more symmetric version of the mean term usually allowed in PCA.

Gous [32] also discusses exponential conditional models, viewed as selecting a linear subspace in the manifold of natural parameters for data-row distributions.

2.3.6 Latent Dirichlet Allocation

In the aspect model of Probabilistic Latent Semantic Analysis (2.12), as in all other models discussed in this section, the distributions $p(i, t)$ and $p(a|t)$ are considered as parameters. This is similar to the Low Rank Approximation models of Section 2.1, where $X = UV'$ are considered parameters. Similar to the probabilistic PCA model described in Section 2.2.1, Blei *et al* [15] propose viewing the rows of Y as independent observations from a fully generative model. In this model, Latent Dirichlet Allocation (LDA), the conditional distribution $p(a|t)$ is viewed as a parameter to be estimated (analogous to the matrix V in Probabilistic PCA). The conditional distribution $p(t|i)$, however, is generated for each row i according to a Dirichlet distribution. It is important to note that unlike probabilistic PCA, which shares the same maximum likelihood solutions with "standard" PCA (where U are treated as parameter), assuming a Dirichlet generative model on U (i.e. on $p(t|i)$) does change the maximum likelihood reconstruction relative to "standard" probabilistic latent semantic analysis.

2.4 Dependent Dimensionality Reduction

Low-rank approximation can also be seen as a method for dimensionality reduction. The goal of dimensionality reduction is to find a low-dimensional representation $\mathbf{u}$ for data $\mathbf{y}$ in a high-dimensional feature space, such that the low-dimensional representation captures the important aspects of the data. In many situations, including collaborative filtering and structure exploration, the "important" aspects of the data are the dependencies between different attributes.

In this Section, we present a formulation of dimensionality reduction that seeks to identify a low-dimensional space that captures the *dependent* aspects of the data, and separate them from *independent* variations. Our goal is to relax restrictions on the form of each of these components, such as Gaussianity, additivity and linearity, while maintaining a principled rigorous framework that allows analysis of the methods. Doing so, we wish to provide a unifying probabilistic framework for dimensionality reduction, emphasizing what assumptions are made and what is being estimated, and allowing us to discuss asymptotic behavior.

Our starting point is the problem of identifying linear dependencies in the presence of independent identically distributed Gaussian noise. In this formulation, discussed in Section 2.2, we observe a data matrix $Y \in \Re^{n \times d}$, which we take as n independent observations of a random vector $\mathbf{y}$, generated as in (2.10), where the dependent, low-dimensional component $\mathbf{x} = \mathbf{u} V'$ (the "signal") has support of rank k, and the independent component $\mathbf{z}$ (the "noise") is i.i.d. zero-mean Gaussian with variance σ^2. The dependencies inside $\mathbf{y}$ are captured by $\mathbf{u}$, which, through the parameters V and σ specifies how each entry $\mathbf{y}_i$ is generated *independently* given u.

As we would like to relax parametric assumptions about the model, and focus only on structural properties about dependencies and independecies, we take the semi-parametric approach of Section 2.2.2 and consider $\mathbf{x} = \mathbf{u} V'$ where $\mathbf{u} \in \mathbb{R}^k$ is an arbitrarily distributed k-dimensional random vector.

Doing so, we do not impose any form on the distribution $\mathbf{u}$, but we do impose a strict form on the conditional distributions $\mathbf{y}_i|\mathbf{u}$: we required them to be Gaussian with fixed variance σ^2 and mean $\mathbf{u}V_i'$. We would like to relax these requirements, and require only that $\mathbf{y}|\mathbf{u}$ be a product distribution, i.e. that its coordinates $\mathbf{y}_i|\mathbf{u}$ be (conditionally) independent. This is depicted as a graphical model in Figure 2-1.

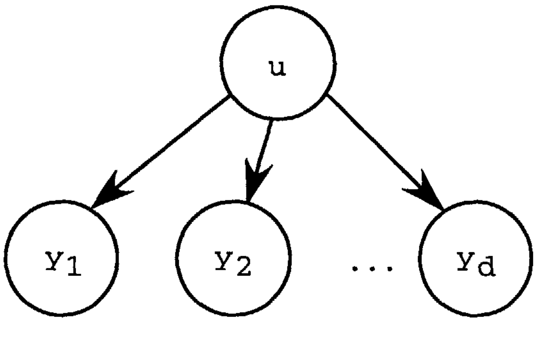


Figure 2-1: Dependent Dimensionality Reduction: the components of $\mathbf{y}$ are independent given $\mathbf{u}$

Since $\mathbf{u}$ is continuous, we cannot expect to forgo all restrictions on $\mathbf{y}_i|\mathbf{u}_i$, but we can expect to set up a semi-parametric problem in which $\mathbf{y}|\mathbf{u}$ may lie in an infinite dimensional family of distributions, and is not strictly parameterized.

Relaxing the Gaussianity leads to linear additive models $\mathbf{y} = \mathbf{u}V' + \mathbf{z}$, with $\mathbf{z}$ independent of $\mathbf{u}$, but not necessarily Gaussian. As discussed earlier, relaxing the additivity is appropriate, e.g., when the noise has a multiplicative component, or when the features of $\mathbf{y}$ are not real numbers. These types of models, with a *known* distribution $\mathbf{y}_i|\mathbf{x}_i$, have been suggested for classification using logistic loss, when $\mathbf{y}_i|\mathbf{x}_i$ forms an exponential family [19], and in a more abstract framework [31].

Relaxing the linearity assumption $\mathbf{x} = \mathbf{u}V'$ is also appropriate in many situations, and several non-linear dimensionality reduction methods have recently been popularized [59, 73]. Fitting a non-linear manifold by minimizing the sum-squared distance can be seen as a ML estimator for $\mathbf{y}|\mathbf{u} = g(\mathbf{u}) + \mathbf{z}$, where $\mathbf{z}$ is i.i.d. Gaussian and $g : \Re^k \rightarrow \Re^d$ specifies some smooth manifold. Combining these ideas leads us to discuss the conditional distributions $\mathbf{y}_i|g_i(\mathbf{u})$, or $\mathbf{y}_i|\mathbf{u}$ directly.

In this Thesis we take our first steps in studying this problem, and in relaxing restrictions on $\mathbf{y}|\mathbf{u}$. We continue to assume a linear model $\mathbf{x} = \mathbf{u}V'$. In Section 3.4 we consider general additive noise models and present a general method for maximum likelihood estimation under this model, even when the noise distribution is unknown, and is regarded as nuisance. In Section 4.2 we consider both additive noise models and a more general class of unbiased models, in which $\mathbf{E}[\mathbf{y}|\mathbf{x}] = \mathbf{x}$. We show how “standard” Frobenius low-rank approximation is appropriate for additive models, and suggest a modification for unbiased models.

Chapter 3

Finding Low Rank Approximations

Low-rank matrix approximation with respect to the Frobenius norm—minimizing the sum squared differences to the target matrix—can be easily solved with Singular Value Decomposition (SVD). This corresponds to finding a maximum likelihood low-rank matrix X maximizing the likelihood of the observation matrix Y, which we assume was generated as $\mathbf{Y} = X + \mathbf{Z}$, where $\mathbf{Z}$ is a matrix of i.i.d. zero-mean Gaussians with constant variance.

For many applications, however, it is appropriate to minimize a different measure of discrepancy between the observed matrix and the low-rank approximation. In this chapter, we discuss alternate measures of discrepancy, and the corresponding optimization problems of finding the low-rank matrix minimizing these measures of discrepancy. Most of these measures correspond to likelihoods with respect to various probabilistic models on $\mathbf{Y}|X$, and minimizing them corresponds to (conditional) maximum likelihood estimation.

In Section 3.2 *weighted* Frobenius norm is considered. Beyond being interesting on its own right, optimization relative to a weighted Frobenius norm also serves us as a basic procedure in methods developed in subsequent sections. In Section 3.3 we show how weighted Frobenius low-rank approximation can be used as a proceedure in a Newton-type appraoch to findind low-rank approximation for general convex loss functions. In Section 3.4 general additive noise, $\mathbf{Y} = X + \mathbf{Z}$, with $\mathbf{Z}$ independent of X, is considered where the distribution of entires $\mathbf{Z}_{ia}$ is modeled as a mixture of Gaussian distributions. Weighted Frobenius low-rank approximation is used in order to find a maximum likelihood estimator in this setting.

The research described in this chapter was mostly reported in conference presentations [69, 70]. The methods are implemented in a Python/Numeric Python library.

3.1 Frobenius Low Rank Approximations

We first revisit the well-studied case of finding a low-rank matrix minimizing the (unweighted) sum-squared error (i.e. the Frobenius norm of the difference) versus a given target matrix. We call such an approximation a Frobenius low-rank approximation. It is a standard result that the low-rank matrix minimizing the sum-squared distance to A is given by the leading components of the singular value decomposition of A. It will be instructive to consider this case carefully and understand why the Frobenius low-rank approximation has such a clean and easily computable form. We will then be able to move on to weighted and other loss functions, and understand how, and why, the situation becomes less favorable.

Problem Formulation

Given a target matrix $A \in \mathbb{R}^{n\times m}$ and a desired (integer) rank k, we would like to find a matrix $X \in \Re^{n\times m}$ of rank (at most) k, that minimizes the Frobenius distance

$$J(X) = \sum_{ia} \left(X_{ia} - A_{ia}\right)^2 .$$

A Matrix-Factorization View

It will be useful to consider the decomposition $X = UV'$ where $U \in \mathbb{R}^{n\times k}$ and $V \in \mathbb{R}^{m\times k}$. Since any rank-$k$ matrix can be decomposed in such a way, and any pair of such matrices yields a rank-k matrix, we can think of the problem as an unconstrained minimization problem over pairs of matrices (U, V) with the minimization objective

$$\begin{aligned} J(U, V) &= \sum_{i,a} W_{i,a} \left(A_{i,a} - (UV')_{i,a} \right)^2 \\ &= \sum_{i,a} W_{i,a} \left(A_{i,a} - \sum_{\alpha} U_{i,\alpha} V_{a,\alpha} \right)^2 . \end{aligned}$$

This decomposition is not unique. For any invertible $R \in \mathbb{R}^{k \times k}$, the pair (UR, VR^{-1}) provides a factorization equivalent to (U, V), i.e. $J(U, V) = J(UR, VR^{-1})$, resulting in a k^2-dimensional manifold of equivalent solutions. The singularities in the space of invertible matrices R yield equivalence classes of solutions actually consisting of a collection of such manifolds, asymptotically tangent to one another.

In particular, any (non-degenerate) solution (U, V) can be orthogonalized to a (non-unique) equivalent orthogonal solution $\bar{U} = UR$, $\bar{V} = VR^{-1}$ such that $\bar{V}'\bar{V} = I$ and $\bar{U}'\bar{U}$ is a diagonal matrix.[1] Instead of limiting our attention only to orthogonal decompositions, it is simpler to allow any matrix pair (U, V), resulting in an unconstrained optimization problem (but remembering that we can always focus on an orthogonal representative).

3.1.1 Characterizing the Low-Rank Matrix Minimizing the Frobenius Distance

Now that we have formulated the Frobenius Low-Rank Approximations problem as an unconstrained optimization problem, with a differentiable objective, we can identify the minimizing solution by studying the derivatives of the objective.

The partial derivatives of the objective J with respect to U, V are

$$\begin{aligned} \tfrac{\partial J}{\partial U} &= 2(UV' - A)V \\ \tfrac{\partial J}{\partial V} &= 2(VU' - A')U \end{aligned} \tag{3.1}$$

[1] We slightly abuse the standard linear-algebra notion of "orthogonal" since we cannot always have both $\bar{U}'\bar{U} = I$ and $\bar{V}'\bar{V} = I$.

Solving $\frac{\partial J}{\partial U} = 0$ for U yields

$$U = AV(V'V)^{-1}. \tag{3.2}$$

Focusing on an orthogonal solution, where $V'V = I$ and $U'U = \Lambda$ is diagonal, yields $U = AV$. Substituting back into $\frac{\partial J}{\partial V} = 0$, we have

$$0 = VU'U - A'U = V\Lambda - A'AV. \tag{3.3}$$

The columns of V are mapped by $A'A$ to multiples of themselves, i.e. they are eigenvectors of $A'A$. Thus, the gradient $\frac{\partial J}{\partial(U,V)}$ vanishes at an orthogonal (U, V) if and only if the columns of V are eigenvectors of $A'A$ and the columns of U are corresponding eigenvectors of AA', scaled by the square root of their eigenvalues. More generally, the gradient vanishes at any (U, V) if and only if the columns of U are spanned by eigenvectors of AA' and the columns of V are correspondingly spanned by eigenvectors of $A'A$. In terms of the singular value decomposition $A = U_0SV_0'$, the gradient vanishes at (U, V) if and only if there exist matrices $Q_U'Q_V = I_k$ (or more generally, a zero/one diagonal matrix rather than I) such that $U = U_0SQ_U$, $V = V_0Q_V$. This provides a complete characterization of the critical points of J. We now turn to identifying the global minimum and understanding the nature of the remaining critical points.

The global minimum can be identified by investigating the value of the objective function at the critical points. Let $\sigma_1 \geq \cdots \geq \sigma_m$ be the eigenvalues of $A'A$. For critical (U, V) that are spanned by eigenvectors corresponding to eigenvalues $\{\sigma_q | q \in Q\}$, the error of $J(U, V)$ is given by the sum of the eigenvalues *not* in Q ($\sum_{q \notin Q} \sigma_q$), and so the global minimum is attained when the eigenvectors corresponding to the highest eigenvalues are taken. As long as there are no repeated eigenvalues, all (U, V) global minima correspond to the same low-rank matrix $X = UV'$, and belong to the same equivalence class. If there are repeated eigenvalues, the global minima correspond to a polytope of low-rank approximations in X space; in U, V space, they form a collection of higher-dimensional asymptotically tangent manifolds.

In order to understand the behavior of the objective function, it is important to study the remaining critical points. For a critical point (U, V) spanned by eigenvectors corresponding to eigenvalues as above (assuming no repeated eigenvalues), the Hessian has exactly $\sum_{q \in Q} q - \binom{k}{2}$ negative eigenvalues: we can replace any eigencomponent with eigenvalue σ with an alternate eigencomponent not already in (U, V) with eigenvalue $\sigma' > \sigma$, decreasing the objective function. The change can be done gradually, replacing the component with a convex combination of the original and the improved components. This results in a line between the two critical points which is a monotonic improvement path. Since there are $\sum_{q \in Q} q - \binom{k}{2}$ such pairs of eigencomponents, there are at least this many directions of improvement. Other than these directions of improvement, and the k^2 directions along the equivalence manifold corresponding to the k^2 zero eigenvalues of the Hessian, all other eigenvalues of the Hessian are positive (or zero, for very degenerate A).

Hence, when minimizing the unweighted Frobenius distance, all critical points that are not global minima are saddle points. This is an important observation: Despite $J(U, V)$ not being a convex function, all of its local minima are global.

3.2 Weighted Low Rank Approximations

For many applications the discrepancy between the observed matrix and the low-rank approximation should be measured relative to a weighted Frobenius norm. While the extension to the weighted-norm case is conceptually straightforward, and dates back to early work on factor analysis [82], standard algorithms (such as SVD) for solving the unweighted case do not carry over to the weighted case. Only the special case of a rank-one weight matrix (where the weights can be decomposed into row weights and column weights) can be solved directly, analogously to SVD [41].

Weighted norms can arise in a number of situations. Zero/one weights, for example, arise when some of the entries in the matrix are not observed. More generally, we may introduce weights in response to some external estimate of the noise variance associated with each measurement. This is the case, for example, in gene expression analysis, where the error model for microarray measurements provides entry-specific noise estimates. Setting the weights inversely proportional to the assumed noise variance can lead to a better reconstruction of the underlying structure. In other applications, entries in the target matrix may represent aggregates of many samples. The standard *un*weighted low-rank approximation (e.g., for separating style and content [72]) would in this context assume that the number of samples is uniform across the entries. Non-uniform weights are needed to appropriately capture any differences in the sample sizes.

Weighted low-rank approximations also arise as a sub-routine in more complex low-rank approximation tasks, with non-quadratic loss functions. Examples of such uses are demonstrated in Sections 3.4 add 3.3.

Prior and related work

Despite its usefulness, the weighted extension has attracted relatively little attention. Shpak [66] and Lu *et al* [50] studied weighted-norm low-rank approximations for the design of two-dimensional digital filters where the weights arise from constraints of varying importance. Shpak developed gradient-based optimization methods while Lu et al. suggested alternating-optimization methods. In both cases, rank-k approximations are greedily combined from k rank-one approximations. Unlike for the unweighted case, such a greedy procedure is sub-optimal.

Shum *et al* [67] extends the work of Ruhe [60] and Wibger [81], who studied the zero-one weight case, to general weighted low-rank approximation, suggesting alternate optimization of U given V and visa versa. The special case of zero-one weights, which can be seen as low rank approximation with missing data, was also confronted recently by several authors, mostly suggesting simple ways of 'filling in' the missing (zero weight) entries, with zeros (e.g. Berry [12]) or with row and column means (e.g. Sarwar *et al* [61]). Brand [17] suggested an incremental update method, considering on data row at a time, for efficiently finding low-rank approximations. Brand's method can be adapted to handle rows with missing data, but the resulting low rank approximation is not precisely the weighted low-rank approximation. Troyanskaya *et al* [75] suggests an iterative fill-in procedure, essentially identical to the one we discuss in Section 3.2.4 for zero-one weights.

Problem formulation

Given a target matrix $A \in \mathbb{R}^{n \times m}$, a corresponding non-negative weight matrix $W \in \mathbb{R}_{+}^{n \times m}$, and a desired (integer) rank k, we would like to find a matrix $X \in \mathbb{R}^{n \times m}$ of rank (at most) k, that minimizes the weighted Frobenius distance

$$J(X) = \sum_{ia} W_{ia} \left(X_{ia} - A_{ia}\right)^2 .$$

3.2.1 Structure of the Optimization Problem

As was discussed previously, the *unweighted* Frobenius low-rank approximation can be computed in closed form and is given by the leading components of the singular value decomposition. We now move on to the weighted case, and try to take the same path as before. Unfortunately, when weights are introduced, the critical point structure changes significantly.

The partial derivatives become (with $\otimes$ denoting element-wise multiplication):

$$\begin{aligned} \tfrac{\partial J}{\partial U} &= 2(W \otimes (UV' - A))\,V \\ \tfrac{\partial J}{\partial V} &= 2(W \otimes (VU' - A'))\,U \end{aligned}$$

The equation $\frac{\partial J}{\partial U} = 0$ is still a linear system in U, and for a fixed V, it can be solved, recovering the global minima U_V^* for fixed V (since $J(U, V)$ is convex in U):

$$U_V^* = \arg\min_U J(U, V). \tag{3.4}$$

However, the solution cannot be written using a single pseudo-inverse. Instead, a separate pseudo-inverse is required for each row $(U_V^*)_i$ of U_V^*:

$$\begin{aligned} (U_V^*)_i &= (V'\underline{W_i}V)^{-1}V'\underline{W_i}A_i \\ &= \text{pinv}(\sqrt{\underline{W_i}}V)(\sqrt{\underline{W_i}}A_i) \end{aligned} \tag{3.5}$$

where $\underline{W_i} \in \Re^{k \times k}$ is a diagonal matrix with the weights from the i^{th} row of W on the diagonal, and A_i is the i^{th} row of the target matrix. In order to proceed as in the unweighted case, we would have liked to choose V such that $V'\underline{W_i}V = I$ (or is at least diagonal). This can certainly be done for a single i, but in order to proceed we need to diagonalize all $V'\underline{W_i}V$ *concurrently*. When W is of rank one, such concurrent diagonalization is possible, allowing for the same structure as in the unweighted case, and in particular an eigenvector-based solution [41]. However, for higher-rank W, we cannot achieve this concurrently for all rows. The critical points of the weighted low-rank approximation problem, therefore, lack the eigenvector structure of the unweighted case. Another implication of this is that the incremental structure of unweighted low-rank approximations is lost: an optimal rank-k factorization cannot necessarily be extended to an optimal rank-$(k+1)$ factorization.

3.2.2 Gradient-Based Optimization

Lacking an analytic solution, we revert to numerical optimization methods to minimize $J(U, V)$. But instead of optimizing $J(U, V)$ by numerically searching over (U, V) pairs, we can take advantage of the fact that for a fixed V, we can calculate U_V^*, and therefore also the projected objective

$$J^*(V) = \min_U J(U, V) = J(U_V^*, V). \tag{3.6}$$

The parameter space of $J^*(V)$ is of course much smaller than that of $J(U, V)$, making optimization of $J^*(V)$ more tractable. This is especially true in many typical applications where the the dimensions of A are highly skewed, with one dimension several orders of magnitude larger than the other (e.g. in gene expression analysis one often deals with thousands of genes, but only a few dozen experiments).

Recovering U_V^* using (3.5) requires n inversions of $k \times k$ matrices. The dominating factor is actually the matrix multiplications: Each calculation of $V'\underline{W_i}V$ requires $O(mk^2)$ operations, for a total of $O(nmk^2)$ operations. Although more involved than the unweighted case, this is still significantly less than the prohibitive $O(n^3k^3)$ required for each iteration suggested by Lu *et al* [50], or for Hessian methods on (U, V) [66], and is only a factor of k larger than the $O(nmk)$ required just to compute the prediction UV'.

After recovering U_V^*, we can easily compute not only the value of the projected objective, but also its gradient. Since $\left.\frac{\partial J(V,U)}{\partial U}\right|_{U=U_V^*} = 0$, we have

$$\frac{\partial J^*(V)}{\partial V} = \left.\frac{\partial J(V,U)}{\partial V}\right|_{U=U_V^*} = 2(W \otimes (VU_V^{*\prime} - A'))U_V^*.$$

The computation requires only $O(nmk)$ operations, and is therefore "free" after U_V^* has been recovered.

The Hessian $\frac{\partial^2 J^*(V)}{\partial V^2}$ is also of interest for optimization. The mixed second derivatives with respect to a pair of rows V_a and V_b of V is (where δ_{ab} is the Kronecker delta):

$$\mathbb{R}^{k\times k} \ni \frac{\partial^2 J^*(V)}{\partial V_a \partial V_b} = 2\sum_i \left(W_{ia}\delta_{ab}(U_V^*)_i(U_V^*)_i' - G_{ia}'(V'\underline{W_i}V)^{-1}G_{ja}(V_a)\right), \tag{3.7}$$

$$\text{where:}\quad G_{ia}(V_a) \overset{\text{def}}{=} W_{ia}(V_a(U_V^*)_i' + ((U_V^*)_i'V_a - A_{ia})I) \in \Re^{k\times k}. \tag{3.8}$$

By associating the matrix multiplications efficiently, the Hessian can be calculated with $O(nm^2k)$ operations, significantly more than the $O(nmk^2)$ operations required for recovering U_V^*, but still manageable when m is small enough.

3.2.3 Local Minima

Equipped with the above calculations, we can use standard gradient-descent techniques to optimize $J^*(V)$. Unfortunately, though, unlike in the unweighted case, $J(U, V)$, and $J^*(V)$, might have local minima that are not global. Figure 3-1 shows the emergence of a non-global local minimum of $J^*(V)$ for a rank-one approximation of $A = \left(\begin{smallmatrix}1 & 1.1\\ 1 & -1\end{smallmatrix}\right)$. The matrix V is a two-dimensional vector. But since $J^*(V)$ is invariant under invertible scalings, V can be specified as an angle θ on a semi-circle. We plot the value of $J^*([\cos\theta, \sin\theta])$ for each θ, and for varying weight matrices of the form $W = \left(\begin{smallmatrix}1+\alpha & 1\\ 1 & 1+\alpha\end{smallmatrix}\right)$. At the front of the plot, the weight matrix is uniform and indeed there is only a single local minimum, but at the back of the plot, where the weight matrix emphasizes the diagonal, a non-global local minimum emerges.

Despite the abundance of local minima, we found gradient descent methods on $J^*(V)$,

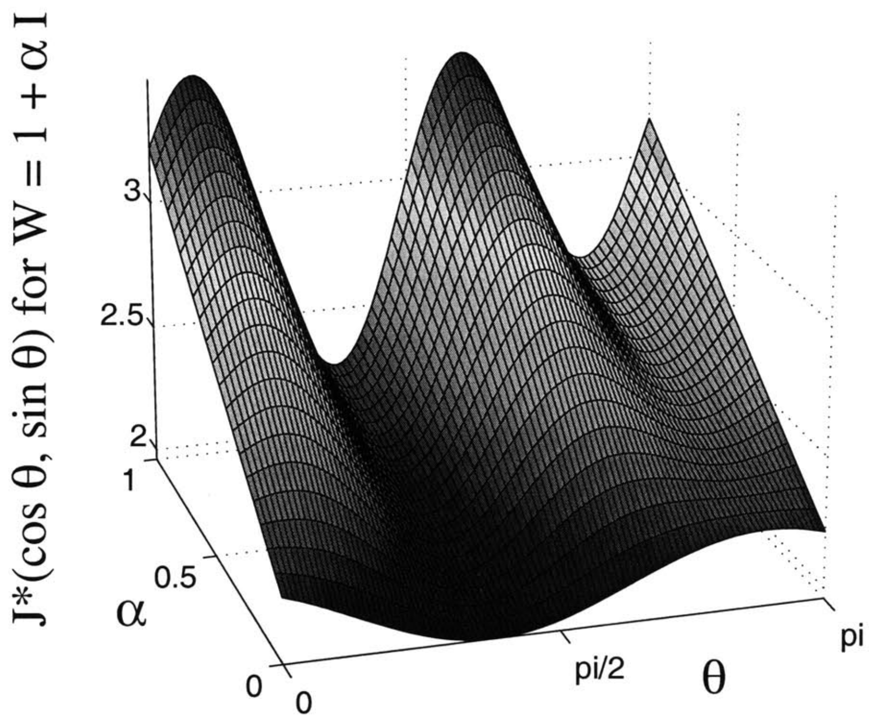


Figure 3-1: Emergence of local minima when the weights become non-uniform.

and in particular conjugate gradient descent, equipped with a long-range line-search for choosing the step size, very effective in avoiding local minima and quickly converging to the global minimum.

The function $J^*(V)$ also has many saddle points, their number far surpassing the number of local minima. In most regions, the function is not convex. Therefore, Newton-Raphson methods are generally inapplicable except very close to a local minimum.

3.2.4 A Missing-Values View and an EM Procedure

In this section we present an alternative optimization procedure, which is much simpler to implement. This procedure is based on viewing the weighted low-rank approximation problem as a maximum-likelihood problem with missing values.

Zero-One Weights

Consider first systems with only zero/one weights, where only some of the elements of the target matrix A are observed (those with weight one) while others are missing (those with weight zero). Referring to a probabilistic model parameterized by a low-rank matrix X, where $\mathbf{Y} = X + \mathbf{Z}$ and $\mathbf{Z}$ is white Gaussian noise, the weighted cost of X is equivalent to the log-likelihood of the observed variables.

This suggests an Expectation-Maximization procedure:

In the M-step, we would like to maximize the expected log-likelihood, where the expectation is over the missing values of $\mathbf{Y}$, with respect the the conditional distribution over these values imposed by the current estimate of the parameters X. For unobserved $\mathbf{Y}_{ia}$, we have $\mathbf{Y}_{ia} = X_{ia} + \mathbf{Z}_{ia}$ with $\mathbf{Z}_{ia} \sim \mathcal{N}(0, \sigma^2)$, and so given X_{ia}, we have $\mathbf{Y}_{ia}|X_{ia} \sim \mathcal{N}(X_{ia}, \sigma^2)$. Let us now evaluate the contribution of $X_{ia}^{(t+1)}$ to the expected log-likelihood:

$$\begin{aligned}
&\mathbf{E}_{\mathbf{Y}_{ia}\sim\mathcal{N}(X_{ia}^{(t)},\sigma^2)}\left[\log \Pr(\mathbf{Y}_{ia}|X_{ia}^{(t+1)})\right] \\
&= \tfrac{1}{\sigma^2}\mathbf{E}_{\mathbf{Y}_{ia}\sim\mathcal{N}(X_{ia}^{(t)},\sigma^2)}\left[(\mathbf{Y}_{ia} - X_{ia}^{(t+1)})^2\right] + \text{Const} \\
&= \tfrac{1}{\sigma^2}\mathbf{E}_{\mathbf{Y}_{ia}\sim\mathcal{N}(X_{ia}^{(t)},\sigma^2)}\left[\mathbf{Y}_{ia}^2 - 2\mathbf{Y}_{ia}X_{ia}^{(t+1)} + X_{ia}^{(t+1)^2}\right] + \text{Const} \\
&= \tfrac{1}{\sigma^2}\left(\mathbf{E}\left[\mathbf{Y}_{ia}^2\right] - 2\mathbf{E}\left[\mathbf{Y}_{ia}\right]X_{ia}^{(t+1)} + X_{ia}^{(t+1)^2}\right) + \text{Const} \\
&= \tfrac{1}{\sigma^2}\left((\sigma^2 + X_{ia}^{(t)^2}) - 2X_{ia}^{(t)}X_{ia}^{(t+1)} + X_{ia}^{(t+1)^2}\right) + \text{Const} \\
&= \tfrac{1}{\sigma^2}(X_{ia}^{(t)} - X_{ia}^{(t+1)})^2 + \text{Const}.
\end{aligned}$$

The contribution corresponding to non-missing values is, as before, $\frac{1}{\sigma^2}(Y_{ia} - X_{ia}^{(t)})^2 + \text{Const}$, and so the total expected log-likelihood is proportional to the Frobenius difference between $X^{(t+1)}$ and the matrix Y with missing values filled in from $X^{(t)}$.

In each **EM** update we would like to find a new parameter matrix maximizing the expected log-likelihood of a filled-in A, where missing values are filled in according to the distribution imposed by the current estimate of X. This maximum-likelihood parameter matrix is the (unweighted) low-rank approximation of the mean filled-in Y, which is Y with missing values filled in from X. To summarize: in the **E**xpectation step values from the current estimate of X are filled in for the missing values in Y, and in the Maximization step X is re-estimated as a low-rank approximation of the filled-in Y.

Such an approach was also suggested by Troyanskaya *et al* [75].

Multiple Target Matrices

In order to extend this approach to a general weight matrix, consider a probabilistic system with several target matrices, $\mathbf{Y}_{(1)}, \mathbf{Y}_{(2)}, \ldots, \mathbf{Y}_{(N)}$, but with a single low-rank parameter matrix X, where $\mathbf{Y}_{(r)} = X + \mathbf{Z}_{(r)}$ and the random matrices $\mathbf{Z}_{(r)}$ are independent white Gaussian noise with fixed variance. When all target matrices are fully observed, the maximum likelihood setting for X is the low-rank approximation of the their average $\frac{1}{N}\sum Y_{(r)}$. Now, if some of the entries of some of the target matrices are not observed, we can use a similar **EM** procedure, where in the expectation step values from the current estimate of X are filled in for all missing entries in the target matrices, and in the maximization step X is updated to be a low-rank approximation of the mean of the filled-in target matrices:

Real-valued Weights

To see how to use the above procedure to solve weighted low-rank approximation problems, consider systems with weights limited to $W_{ia} = \frac{Q_{ia}}{N}$ with integer $Q_{ia} \in \{0, 1, \ldots, N\}$. Such a low-rank approximation problem can be transformed to a missing value problem of the form above by "observing" the value A_{ia} in Q_{ia} of the target matrices $Y_{(1)}, \ldots, Y_{(Q_{ia})}$ (for each entry i, a), and leaving the entry as missing in the rest of the target matrices. The **EM** update then becomes:

$$X^{(t+1)} = \mathrm{LRA}_k \left(W \otimes A + (1 - W) \otimes X^{(t)} \right) \tag{3.9}$$

where $\mathrm{LRA}_k(X)$ is the unweighted rank-k approximation of X, as can be computed from the SVD. Note that this procedure is independent of N. For any weight matrix (scaled to weights between zero and one) the procedure in equation (3.9) can thus be seen as an expectation-maximization procedure. This provides for a very simple, tweaking-free method for finding weighted low-rank approximations.

Initialization and a Heuristic Improvement

Although we found this EM-inspired method effective in many cases, in some other cases the procedure converges to a local minimum which is not global. Since the method is completely deterministic, initialization of X plays a crucial role in promoting convergence to a global, or at least deep local, minimum, as well as the speed with which convergence is attained.

Two obvious initialization methods are to initialize X to A, and to initialize X to zero. Initializing X to A works reasonably well if the weights are bounded away from zero, or if the target values in A have relatively small variance. However, when the weights are zero, or very close to zero, the target values become meaningless, and can throw off the search. Initializing X to zero avoids this problem, as target values with zero weights are completely ignored (as they should be), and works well as long as the weights are fairly dense. However, when the weights are sparse, it often converges to local minima which consistently under-predict the magnitude of the target values.

As an alternative to these initialization methods, we found the following procedure very effective: we initialize X to zero, but instead of seeking a rank-k approximation right away, we start with a full rank matrix, and gradually reduce the rank of our approximations. That is, the first $m - k$ iterations take the form:

$$X^{(t+1)} = \mathrm{LRA}_{m-t}\left(W \otimes A + (1 - W) \otimes X^{(t)}\right), \tag{3.10}$$

resulting in $X^{(t)}$ of rank $(m - t + 1)$. After reaching rank k, we revert back to the iterations of equation (3.9) until convergence. Note that with iterations of the form $X^{(t+1)} = W \otimes A + (1 - W) \otimes X^{(t)}$, without rank reductions, we would have $X_{ia}^{(t)} = (1-(1-W_{ia})^t))A_{ia} \rightarrow (1 - e^{-tW_{ia}})A_{ia}$, which converges exponentially fast to A for positive weights. Of course, because of the rank reduction, this does not hold, but even the few high-rank iterations set values with weights away from zero close to their target values, as long as they do not significantly contradict other values.

3.2.5 Reconstruction Experiments

Since the unweighted or simple low-rank approximation problem permits a closed-form solution, one might be tempted to use such a solution even in the presence of non-uniform weights (i.e., ignore the weights). We demonstrate here that this procedure results in a substantial loss of reconstruction accuracy as compared to the EM algorithm designed for the weighted problem.

To this end, we generated 1000×30 low rank matrices combined with Gaussian noise models to yield the observed (target) matrices. For each matrix entry, the noise variance σ_{ia}^2 was chosen uniformly in some noise level range characterized by a *noise spread ratio* $\max \sigma^2 / \min \sigma^2$. The planted matrix was subsequently reconstructed using both a weighted low-rank approximation with weights $W_{ia} = 1/\sigma_{ia}^2$, and an unweighted low-rank approximation (using SVD). The quality of reconstruction was assessed by an unweighted squared distance from the "planted" matrix.

Figure 3-2 shows the quality of reconstruction attained by the two approaches as a function of the signal (weighted variance of planted low-rank matrix) to noise (average noise variance) ratio, for a noise spread ratio of 100 (corresponding to weights in the range 0.01–1). The reconstruction error attained by the weighted approach is generally over twenty times smaller than the error of the unweighted solution. Figure 3-3 shows this improvement in the reconstruction error, in terms of the error ratio between the weighted and unweighted solutions, for the data in Figure 3-2, as well as for smaller noise spread ratios of ten and two. Even when the noise variances (and hence the weights) are within a factor of two, we still see a consistent ten percent improvement in reconstruction.

The weighted low-rank approximations in this experiment were computed using the EM algorithm of Section 3.2.4. For a wide noise spread, when the low-rank matrix becomes virtually undetectable (a signal-to-noise ratio well below one, and reconstruction errors in excess of the variance of the signal), EM often converges to a non-global minimum. This results in weighted low-rank approximations with errors far higher than could otherwise be expected, as can be seen in both figures. In such situations, conjugate gradient descent methods proved far superior in finding the global minimum.

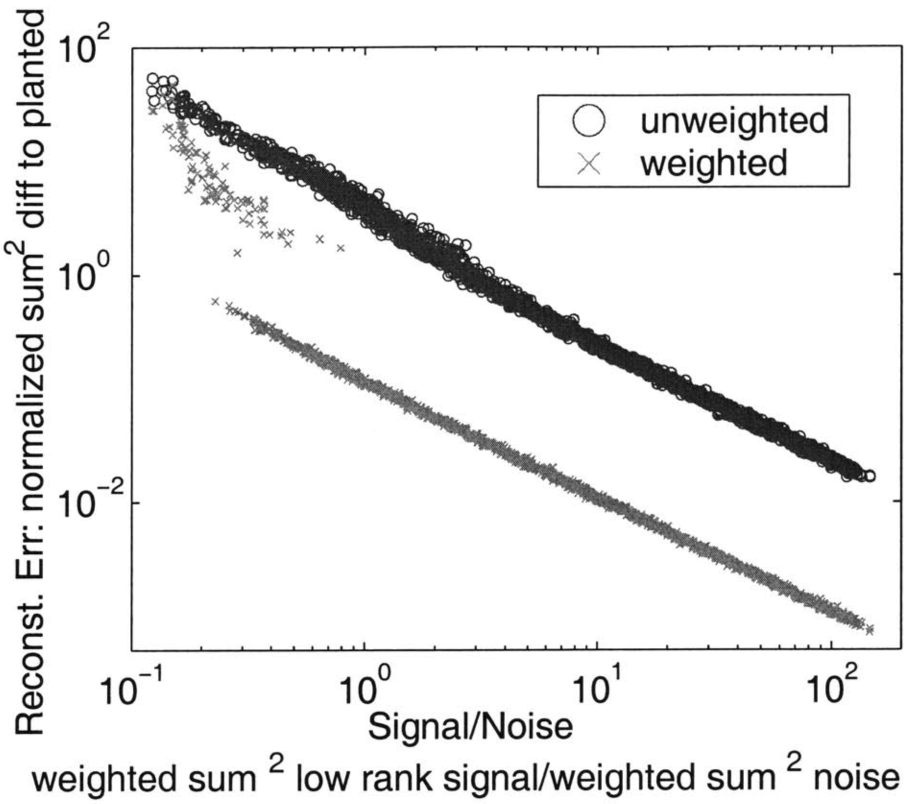


Figure 3-2: Reconstruction of a 1000×30 rank-three matrix: weighted and unweighted reconstruction with a noise spread of 100.

3.3 Low Rank Approximation with Other Convex Loss Functions

In this section we depart from the sum-squared error as a measure of loss, and consider other loss functions. We consider the optimization problem:

$$\min_{X,\text{rank}(X)=k} \sum_{ia} \text{loss}(X_ia;\, Y_ia) \tag{3.11}$$

where the loss function, the target rank k and the target matrix Y are given. Specifically, we consider the case in which the loss function is convex. By *convex* we mean that it is convex in X_ia for any value of Y_ia.

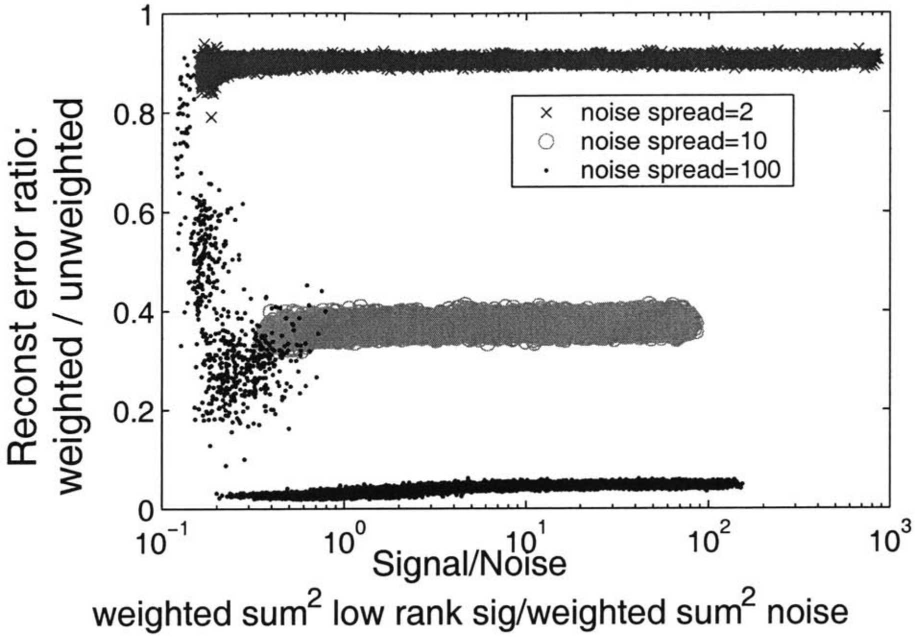


Figure 3-3: Reconstruction of a 1000×30 rank-three matrix with varying noise spreads.

3.3.1 A Newton Approach

Using a weighted low-rank approximation, we can fit a low-rank matrix X minimizing a quadratic loss from the target. In order to fit a convex, but non-quadratic loss, we use a quadratic approximation to the loss. At each iteration, we consider a quadratic approximation to the overall loss. The quadratic approximation can be written as a weighted low-rank approximation problem, and we can apply the methods of Section 3.2.

For a twice continuously differentiable loss function, the second-order Taylor expansion of $\text{loss}(x, y)$ around $\tilde{x}$ can be written as:

$$\begin{aligned}
\text{loss}(x; y) &\approx \text{loss}(\tilde{x}; y) + \text{loss}'(\tilde{x}; y)(x - \tilde{x}) + \frac{\text{loss}''(\tilde{x}; y)}{2}(x - \tilde{x})^2 \\
&= \left(\text{loss}(\tilde{x}; y) - \text{loss}'(\tilde{x}; y)\tilde{x} + \frac{\text{loss}''(\tilde{x}; y)}{2}\tilde{x}^2\right) \qquad (3.12)\\
&\quad + \left(\text{loss}'(\tilde{x}; y) - \text{loss}''(\tilde{x}; y)\tilde{x}\right) x + \frac{\text{loss}''(\tilde{x}; y)}{2} x^2 \\
&= \frac{\text{loss}''(\tilde{x}; y)}{2}\left(x - \frac{\text{loss}''(\tilde{x}; y)\tilde{x} - \text{loss}'(\tilde{x}; y)}{\text{loss}''(\tilde{x}; y)}\right)^2 + \text{Const} \qquad (3.13)
\end{aligned}$$

where the constant term depends on $\tilde{x}$ and y, but not on x.

We can now write a second-order approximation for the total discrepancy of X about an origin matrix $\tilde{X}$:

$$\begin{aligned}\mathcal{D}(X; Y) &= \sum_{ia} \text{loss}(X_{ia}; Y_{ia}) \\ &\approx \sum_{ia} \frac{\text{loss}''(\tilde{X}_{ia}; Y_{ia})}{2} \left(X_{ia} - \frac{\text{loss}''(\tilde{X}_{ia}; Y_{ia})\tilde{X}_{ia} - \text{loss}'(\tilde{X}_{ia}; Y_{ia})}{\text{loss}''(\tilde{X}_{ia}; Y_{ia})} \right)^2 + \text{Const} \\ &= \frac{1}{2} \sum_{ia} W_{ia}(X_{ia} - A_{ia})^2 + \text{Const} \qquad (3.14) \\ & \qquad (3.15)\end{aligned}$$

where

$$W_{ia} = \text{loss}''(\tilde{X}_{ia}; Y_{ia}) \qquad A_{ia} = \frac{\text{loss}''(\tilde{X}_{ia}; Y_{ia})\tilde{X}_{ia} - \text{loss}'(\tilde{X}_{ia}; Y_{ia})}{\text{loss}''(\tilde{X}_{ia}; Y_{ia})}. \qquad (3.16)$$

Maximizing (3.14) is a weighted low-rank approximation problem. Note that for each entry (i, a), we use a second-order expansion about a *different* point $\tilde{X}_{ia}$. The closer the origin $\tilde{X}_{ia}$ is to X_{ia}, the better the approximation.

This suggest a Newton-type iterative approach, where at each iteration we set

$$W_{ia}^{(t+1)} = \text{loss}''(X_{ia}^{(t)}; Y_{ia}) \qquad A_{ia}^{(t+1)} = \frac{\text{loss}''(X_{ia}^{(t)}; Y_{ia})X_{ia}^{(t)} - \text{loss}'(X_{ia}^{(t)}; Y_{ia})}{\text{loss}''(X_{ia}^{(t)}; Y_{ia})} \qquad (3.17)$$

and then update the current solution $X^{(t)}$ by optimizing a weighted low-rank approximation problem:

$$X^{(t+1)} = \arg\min_X \sum_{ia} W_{ia}^{(t+1)}(X_{ia} - A_{ia}^{(t+1)})^2 \qquad (3.18)$$

3.3.2 Low-rank Logistic Regression

As a specific example of optimizing a convex non-quadratic loss, we consider the problem of logistic low-rank regression: entries of an observed binary matrix are modeled as Bernoulli variables with natural parameters forming a low-rank matrix X, and the maximum likelihood low-rank matrix X is sought. The negative log-likelihood, which is our objective to be minimized, can then be written as a sum of logistic losses:

$$\mathcal{D}(X;Y) = -\log \Pr(Y|X) = \sum_{ia}(-\log g(Y_{ia}X_{ia})) \tag{3.19}$$

where $g(z) = \frac{1}{1+e^{-z}}$ is the logistic function.

Taking the derivatives of the logistic loss $\mathrm{loss}(x;y) = -\log g(yx)$, we get the following Newton updates for the weight and target matrices (3.17):

$$A_{ia}^{(t+1)} = X_{ia}^{(t)} + \frac{Y_{ia}}{g(Y_{ia}X_{ia}^{(t)})} \qquad W_{ia}^{(t+1)} = g(X_{ia}^{(t)})g(-X_{ia}^{(t)}) \tag{3.20}$$

Optimizing a Second Order Variational Bound

For the Taylor expansion, the improvement of the approximation is not always monotonic. This might cause the method outlined above not to converge. In order to provide for a more robust method, we use the following variational bound on the logistic [42]:

$$\begin{aligned}\log g(yx) &\geq \log g(y\tilde{x}) + \tfrac{yx-y\tilde{x}}{2} - \tfrac{\tanh(\tilde{x}/2)}{4\tilde{x}}\left(x^2 - \tilde{x}^2\right)\\ &= -\tfrac{1}{4}\tfrac{\tanh(\tilde{x}/2)}{\tilde{x}}\left(x - \tfrac{y\tilde{x}}{\tanh(\tilde{x}/2)}\right) + \mathrm{Const},\end{aligned}$$

with equality if and only if $x = \tilde{x}$. Bounding each entry, we get the corresponding bound on the overall objective:

$$\mathcal{D}(X;Y) \leq \tfrac{1}{4}\sum_{ia}\tfrac{\tanh(\tilde{X}_{ia}/2)}{\tilde{X}_{ia}}\left(X_{ia} - \tfrac{Y_{ia}\tilde{X}_{ia}}{\tanh(\tilde{X}_{ia}/2)}\right) + \mathrm{Const} \tag{3.21}$$

with equality if and only if $X = \tilde{X}$. This bound suggests an iterative update of the parameter matrix $X^{(t)}$ by seeking a low-rank approximation $X^{(t+1)}$ for the following target and weight matrices:

$$A_{ia}^{(t+1)} = Y_{ia}/W_{ia}^{(t+1)} \qquad W_{ia}^{(t+1)} = \tanh(X_{ia}^{(t)}/2)/X_{ia}^{(t)} \tag{3.22}$$

Fortunately, we do not need to confront the severe problems associated with nesting iterative optimization methods. In order to increase the likelihood of our logistic model, we do not need to find a low-rank matrix minimizing the objective specified by (3.22), just one improving it. Any low-rank matrix $X^{(t+1)}$ with a lower objective value than $X^{(t)}$, with respect to $A^{(t+1)}$ and $W^{(t+1)}$, is guaranteed to have a lower overall discrepancy $\mathcal{D}(X; Y)$ (i.e. higher likelihood): A lower objective corresponds to a lower lower bound in (3.21), and since the bound is tight for $X^{(t)}$, $\mathcal{D}(X^{(t+1)}; Y)$ must be lower than $\mathcal{D}(X^{(t)}; Y)$. Moreover, if the discrepancy of $X^{(t)}$ is not already minimal are guaranteed to be matrices with lower objective values. Therefore, we can mix weighted low-rank approximation iterations and logistic bound update iterations, while still ensuring convergence.

In many applications we may also want to associate external weights with each entry in the matrix (e.g. to accommodate missing values), or more generally, weights (counts) of positive and negative observations in each entry (e.g. to capture the likelihood with respect to an empirical distribution). This can easily be done by multiplying the weights in (3.22) or (3.20) by the external weights.

Note that the target and weight matrices corresponding to the Taylor approximation and those corresponding to the variational bound are different: The variational target is always closer to the current value of X, and the weights are more subtle (less variation between the weights). This ensures the guaranteed convergence (as discussed above), but the price we pay is a much lower convergence rate. Although we have observed many instances in which a 'Taylor' iteration increases, rather then decreases, the objective, overall convergence was attained much faster using 'Taylor', rather than 'variational' iterations.

The same approach outlines here is applicable for any twice differentiable convex loss function, and especially loss functions for which a second order variational bound is known in closed form.

3.4 Low Rank Approximations with Non-Gaussian Additive Noise

We now depart from Gaussian noise models, but still assume additive noise:

$$\mathbf{Y} = X + \mathbf{Z}, \tag{3.23}$$

where the entries in $\mathbf{Z}$ are independent of each other and of X. We model the noise distribution as a mixture of Gaussian distributions:

$$p_{\mathbf{Z}}(Z_{ia}) = \sum_{c=1}^{m} p_c (2\pi\sigma_c^2)^{1/2} \exp((Z_{ia} - \mu_c)^2/(2\sigma_c^2)). \tag{3.24}$$

Given such a mixture model, an observed data matrix $Y \in \mathbb{R}^{n \times m}$, and a target rank k, we would like to find the rank-k matrix X maximizing the likelihood $\Pr(Y = X + \mathbf{Z}; X)$, where the entries of $\mathbf{Z}$ are i.i.d. with distribution $p_{\mathbf{Z}}$.

3.4.1 An EM optimization procedure

To do so, we introduce latent variables $\mathbf{C}_{ia}$ specifying the mixture component of the noise at $\mathbf{Y}_{ia}$, and solve the problem using EM.. In the Expectation step, we compute the posterior probabilities $\Pr(\mathbf{C}_{ia}|Y_{ia}; X)$ based on the current low-rank parameter matrix X. In the Maximization step we need to find the low-rank matrix X that maximizes the posterior expected log-likelihood

$$\begin{aligned}\mathbf{E}_{\mathbf{C}|Y}\left[\log \Pr(Y = X + \mathbf{Z}|\mathbf{C}; X)\right] &= -\sum_{ia} \mathbf{E}_{\mathbf{C}_{ia}|Y_{ia}}\left[\tfrac{1}{2}\log 2\pi\sigma^2_{\mathbf{C}_{ia}} + \tfrac{\left((X_{ia}-Y_{ia})-\mu_{\mathbf{C}_{ia}}\right)^2}{2\sigma^2_{\mathbf{C}_{ia}}}\right] \\ &= -\sum_{ia}\sum_{c} \tfrac{\Pr(\mathbf{C}_{ia}=c)|Y_{ia}}{2\sigma_c^2}(X_{ia}-(Y_{ia}+\mu_c))^2 + \text{Const} \\ &= -\tfrac{1}{2}\sum_{ia} W_{ia}\left(X_{ia} - A_{ia}\right)^2 + \text{Const} \end{aligned} \tag{3.25}$$

where

$$W_{ia} = \sum_{c} \tfrac{\Pr(\mathbf{C}_{ia}=c)|Y_{ia}}{\sigma_c^2} \qquad A_{ia} = Y_{ia} + \sum_{c} \tfrac{\Pr(\mathbf{C}_{ia}=c)|Y_{ia}\mu_c}{\sigma_c^2 W_{ia}} \tag{3.26}$$

This is a *weighted* Frobenius low-rank approximation (WLRA) problem. Equipped with a WLRA optimization method (Section 3.2), we can now perform EM iteration in order to find the matrix X maximizing the likelihood of the observed matrix Y. At each **M** step it is enough to perform a single WLRA optimization iteration, which is guaranteed to improve the WLRA objective, and so also the likelihood. The resulting iterative optimization method can be viewed as iterative WLRA method, where the target and weight matrices are dynamically updated as a function of the current solution.

Unknown Gaussian Mixtures

So far we discussed the situation in which the noise was i.i.d. according to a known Gaussian mixture. However, we can easily augment our method to handle an *unknown* noise distribution, so long as it is a Gaussian mixture. This can be done by introducing an optimization of the mixture parameters of p (with respect to the current posteriors and low-rank matrix X) at each M iteration. Note that again, this is a weakened EM method since the mixture parameters and the low-rank matrix X are not concurrently optimized, but rather are alternatively optimized, leading to a sub-optimal setting. Still, we do improve the objective, and are guaranteed convergence to a local minimum.

Infinite Gaussian Mixtures

We do not have to limit ourselves only to finite Gaussian mixtures. The maximum-likelihood problem can be well-defined also for classes of noise distributions with an unbounded, or infinite, number of Gaussian components. The target and weight matrices for the WLRA in the **M** step can be written as

$$\begin{aligned} W_{ia} &= \mathbf{E}_{\mathbf{C}_{ia}|Y_{ia}}\left[\frac{1}{\sigma^2_{\mathbf{C}_{ia}}}\right] \\ A_{ia} &= Y_{ia} + \mathbf{E}_{\mathbf{C}_{ia}|Y_{ia}}\left[\frac{u_{\mathbf{C}_{ia}}}{\sigma^2_{\mathbf{C}_{ia}}}\right] / W_{ia}. \end{aligned}$$

Any class of Gaussian mixture distributions which we can efficiently fit using EM, and compute these two quantities for, is amenable to our approach.

3.4.2 Reconstruction Experiments with GSMs

We report here experiments with ML estimation using bounded Gaussian scale mixtures [6, 77], i.e. a mixture of Gaussians with zero mean, and variance bounded from bellow. Gaussian scale mixtures (GSMs) are a rich class of symmetric distributions, which include non-log-concave, and heavy tailed distributions. We investigated two noise distributions: a 'Gaussian with outliers' distribution formed as a mixture of two zero-mean Gaussians with widely varying variances; and a Laplace distribution $p(z) \propto e^{-|z|}$, which is an infinite scale mixture of Gaussians. Figures 3-4,3-5 show the quality of reconstruction of the L_2 estimator and the ML bounded GSM estimator, for these two noise distributions, for a fixed sample size of 300 rows, under varying signal strengths. We allowed ten Gaussian components, and did not observe any significant change in the estimator when the number of components increases.

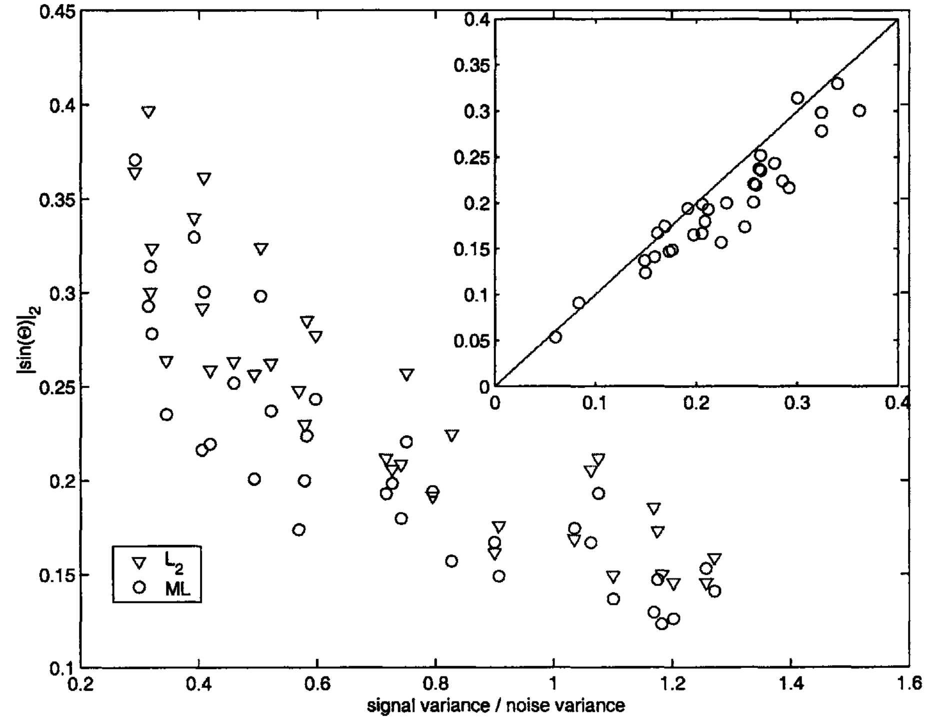


Figure 3-4: Norm of sines of canonical angles to correct subspace for a random rank-3 subspace in $\mathbb{R}^{10}$ with Laplace noise. Insert: sine norm of ML estimation plotted against sine norm of Frobenius estimation (label "L_2").

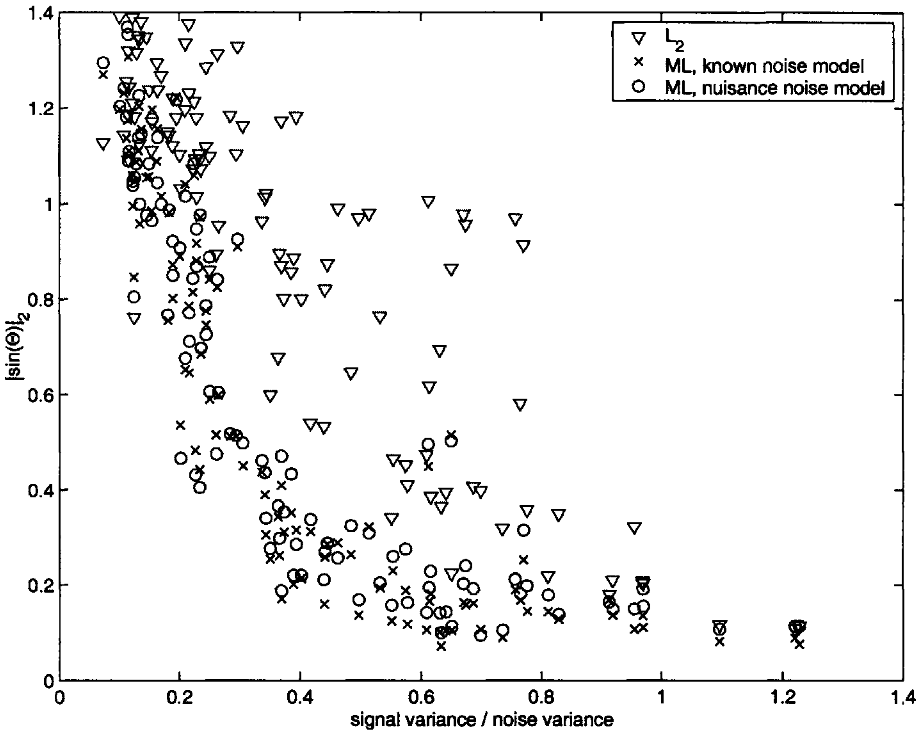


Figure 3-5: Norm of sines of canonical angles to correct subspace for a random rank-2 subspace in $\mathbb{R}^{10}$ with $0.99\mathcal{N}(0,1) + 0.01\mathcal{N}(0,100)$ noise. Frobenius estimator is denoted "L_2".

The ML estimator is overall more accurate than the Frobenius estimator (standard PCA)—it succeeds in reliably reconstructing the low-rank signal for signals which are approximately three times weaker than those necessary for reliable reconstruction using the Frobenius estimator. The improvement in performance is not as dramatic, but still noticeable, for Laplace noise.

Most usual caveats of learning a distribution as a Gaussian mixture apply, and we will want to limit the admissible models, both in terms of complexity (e.g. number of components) and in order to prevent singularities.

One particularly problematic situation, which is not specific to Gaussian mixtures, should be pointed out. This situation occurs if we allow the density to attain unbounded values at a point z_0, while remaining bounded from bellow by some strictly positive function elsewhere. As the density at z_0 increases, it becomes increasingly profitable to fit some, even a few, values of y exactly, while paying only a constant penalty for completely missing all other entries. But it is always possible for X to fit at least k values in each row of y exactly (e.g. when the columns of X are spanned by k columns of y). The likelihood is thus unbounded, and will go to infinity for those X fitting some values of y exactly, as $p(z_0)$ goes to infinity.

3.4.3 Comparing Newton's Methods and Using Gaussian Mixtures

Confronted with a general additive noise distribution, the approach suggested in Section 3.4 would be to rewrite, or approximate, it as a Gaussian mixture and use WLRA in order to learn X using EM. A different option is to write down the log likelihood with respect to the additive noise distribution, and to use Newton's method of Section 3.3.1, considering the second order Taylor expansions of the log-likelihood, with respect to the entries of X, and iteratively maximize them using WLRA. Such an approach requires calculating the first and second derivatives of the density. If the density is not specified analytically, or is unknown, these quantities need to be estimated. But beyond these issues, which can be overcome, lies the major problem of Newton's method: the noise density must be strictly log-concave and differentiable. If the distribution is not log-concave, the quadratic expansion of the log-likelihood will be unbounded and will not admit an optimum. Attempting to ignore this fact, and for example "optimizing" U given V using the equations derived for non-negative weights would actually drive us towards a saddle-point rather then a local optimum. The non-concavity does not only mean that we are not guaranteed a global optimum (which we are not guaranteed in any case, due to the non-convexity of the low-rank requirement)— it does not yield even local improvements. On the other hand, approximating the distribution as a Gaussians mixture and using the EM method, might still get stuck in local minima, but is at least guaranteed local improvement.

Limiting ourselves to only log-concave distributions is a rather strong limitation, as it precludes, for example, additive noise with any heavy-tailed distribution. Consider even the "balanced tail" Laplace distribution $p(z) \propto e^{-|z|}$. Since the log-density is piecewise linear, a quadratic approximation of it is a line, which of course does not attain a minimum value.

Chapter 4

Consistency of Low Rank Approximation

Viewing dimensionality reduction as a subspace estimation problem (Section 2.2.2) allows us to investigate its properties as an estimator. In this chapter, we begin doing so, by studying the most basic property, namely asymptotic consistency, under various assumptions. We will see that even this minimal requirement cannot be taken for granted.

Maximum likelihood estimation assuming a fully parametric model, e.g. low rank Gaussian signal and white Gaussian noise (Seciton 2.2.1), is certainly consistent by virtue of it being a maximum likelihood estimator in a finite dimensional parameter space. However, in line with Section 2.4, our interests lay in focusing on the structural, non-parametric, aspects of the model, and understanding what the minimal requirements for consistency are.

We begin by studying the asymptotic consistency of maximum likelihood estimation for various conditional models, under the framework of Section 2.2.2. The question we ask is: if the observed data is generated from a low-rank matrix using the assumed conditional model, does the maximum likelihood estimator for the low-rank subspace spanning the low-rank matrix converge to the true subspace when more data rows are available? To answer this question, we first develop a series of necessary and sufficient conditions for consistency (Section 4.1.1) and then proceed to analyze a number of specific conditional models. We show that maximum likelihood estimation of the subspace in most of these conditional models is, in fact, *not* consistent.

On the other hand, in Section 4.2 we show how the simple Frobenius low-rank approximation *is* consistent for the general class of additive noise models. The Frobenius low-rank approximation is not appropriate for non-additive conditional models, but in Section 4.2.2 we suggest a correction that is appropriate for *unbiased* conditional models.

Most of the research described in this chapter is reported in a conference presentation [70]. Matterial which appears here for the first time includes formulation of the necessary and sufficient conditions for non-additive models, as well as the analyzis of the consistency of maximum likelihood estimation for the logistic and Bernoulli conditional models.

4.1 Consistency of Maximum Likelihood Estimation

We consider maximum likelihood estimation for low-rank linear models with a known conditional distribution $\mathbf{y}|\mathbf{x}$, where $\mathbf{x} = \mathbf{u}V'$ lies in the low-rank subspace V, which is the "parameter" of interest. These include both additive models $\mathbf{y} = \mathbf{x} + \mathbf{z}$, where $\mathbf{z}$ is i.i.d. with known distribution p_Z, and non-additive models with a known conditional distribution, such as Exponential-PCA.

The question we ask is: assuming the data $\mathbf{Y}$ does follow the known conditional distribution $\mathbf{y}|\mathbf{x}$, where $\mathbf{x} = \mathbf{u}\,V_0{}'$, will the maximum likelihood estimator

$$\hat{V} = \arg\max_V \sup_U \Pr\left(Y \,|\, UV'\right) \tag{4.1}$$

converge to the true subspace V_0 as more data rows of Y are available? We would like this to hold for *any* distribution over $\mathbf{u}$. That is, we would like to reconstruct the correct support subspace for any distribution over $\mathbf{x}$ with low-rank support.

Note that due to degrees of freedom in the representation (rotations of V, and in fact, any linear invertible transformations), we cannot expect the *matrix* $\hat{V}$ to always converge to V_0. Instead, we must discuss the convergence of the *subspace* it represents. Throughout the presentation, we will slightly overload notation and use a matrix to denote also its column subspace. In particular, we will denote by V_0 the true signal subspace. In order to study estimators for a subspace, we must be able to compare two subspaces. A natural way of doing so is through the *canonical angles* between them (Section 2.2.3). All results described bellow refer to convergence with respect to the canonical angles.

4.1.1 Necessary and sufficient conditions

In order to answer this question, we present a necessary and sufficient conditions for the consistency of the maximum likelihood estimator, under a known conditional distribution $p_{Y|X}$.

A necessary condition

Consider the random function:

$$\Phi(V) = \inf_{u} - \log p_{Y|X}(\mathbf{y}|uV'). \tag{4.2}$$

Here $\mathbf{y}$ is a random vector, hence $\Phi(V)$ is a random variable. The maximum log-likelihood of V for the data Y can be written in terms of the empirical mean of $\Phi(V)$:

$$\begin{aligned} \log \Pr(Y|V) &= \sup_{U} \log \Pr(Y|UV') \\ &= \sup_{U} \sum \log \Pr(Y_a|U_aV') \\ &= \sum_{a} \sup_{u} \log \Pr(Y_a|uV') \\ &= -n\hat{\mathbf{E}}[\Phi(V)] \end{aligned} \tag{4.3}$$

where $\hat{\mathbf{E}}[]$ denotes the empirical mean with respect to the data μ. Maximizing the likelihood of V is equivalent to minimizing the empirical mean $\hat{\mathbf{E}}[\Phi(V)]$.

When the number of samples increase, the empirical means converge to the true means, and if $\mathbf{E}[\Phi(V_1)] < \mathbf{E}[\Phi(V_2)]$, then with probability approaching one V_2 will not minimize $\hat{\mathbf{E}}[\Phi(V)]$. For the ML estimator to be consistent, $\mathbf{E}[\Phi(V)]$ *must* be minimized by V_0, establishing a necessary condition for consistency:

Condition 1. $\mathbf{E}_{\mathbf{y}}[\Phi(V)]$ *is minimized by* V_0.

Proposition 1. *Condition 1 is necessary for the consistency of the maximum likelihood estimator* $\hat{V}$.

The sufficiency of this condition rests on the *uniform* convergence of $\{\hat{\mathbf{E}}[\Phi(V)]\}$, which does not generally hold, or at least on uniform *divergence* from $\mathbf{E}[\Phi(V_0)]$. It should be noted that the issue here is whether the ML estimator at all converges, since if it does converge, it must converge to the minimizer of $\mathbf{E}[\Phi(V)]$.

Conditions independent of the signal distribution

When discussing $\mathbf{E}[\Phi(V)]$, the expectation is with respect to the conditional distribution $p_{Y|X}$ *and* the signal distribution $p_X = p_U$. This is not quite satisfactory, as we would like results which are independent of the signal distribution, beyond the rank of its support. To do so, we must ensure the expectation of $\Phi(V)$ is minimized on V_0 for *all* possible signals (and not only in expectation).

Denote the negative maximum likelihood of a data vector $y \in \mathbb{R}^m$:

$$\phi(V;y) = \inf_u(-\log p(y|uV')). \tag{4.4}$$

and for any *signal* vector $x \in \mathbb{R}^m$ consider the expected conditional negative log-likelihood:

$$\Psi(V;x) = \mathbf{E}_{\mathbf{y}|\mathbf{x}}[\phi(V;\mathbf{y})] = \mathbf{E}_{\mathbf{y}|\mathbf{x}}\left[\inf_u -\log p_{\mathbf{y}|\mathbf{x}}(\mathbf{y}|uV)\middle| x\right] \tag{4.5}$$

This is the expected contribution of a signal vector x to the maximum log-likelihood of V. For additive models $\mathbf{y} = \mathbf{x} + \mathbf{z}$, $\Psi(V;x)$ can be written in terms of the error distribution p_Z:

$$\Psi(V;x) = \mathbf{E}_{\mathbf{z}}\left[\inf_u -\log p_{\mathbf{z}}((x+\mathbf{z}) - uV)\right] \tag{4.6}$$

We can now formulate a condition which is independent of the signal distribution, and is necessary for the maximum likelihood estimator being consistent for *all* signal distribution:

Condition 2. *For all* $x \in \Re^m$, $\Psi(V;x)$ *is minimized with respect to* V *exactly when* $x \in \operatorname{span} V$.

Proposition 2. *For any conditional distribution* $\mathrm{y}|\mathrm{x}$ *(e.g. for any additive noise distribution), Condition 2 is necessary for the consistency of the maximum likelihood estimator* $\hat{V}$ *for all distributions* $\mathrm{x} = \mathrm{u}\, V_0$. *That is, if Condition 2 does not hold, then it cannot be the case that for all* V_0 *and all distributions* u, *we have* $\hat{V} \to V_0$.

We will also consider the following more specific condition, which focuses on the expected contribution to the log-likelihood $\Psi(V; 0)$ when no signal is present (i.e. $x = 0$):

Condition 3. $\Psi(V; 0)$ *is constant for all* V.

Proposition 3. *If the conditional distribution* $\mathrm{y}|\mathrm{x}$ *has a continuous density (e.g. if it is additive and* p_Z *is continuous), then Condition 3 is necessary for the consistency of the maximum likelihood estimator* $\hat{V}$ *for all distributions* $\mathrm{x} = \mathrm{u}\, V_0$.

If Condition 3 does not hold, we have $\Psi(V_1; 0) < \Psi(V_2; 0)$ for some V_1, V_2, and for small enough $x \in V_2$, $\Psi(V_1; x) < \Psi(V_2; x)$. A non-constant $\Psi(V; 0)$ indicates an a-priori bias towards certain sub-spaces.

Sufficiency of the conditions

The sufficiency of Conditions 1 and 2 rests on the *uniform* convergence of $\{\hat{\mathbf{E}}\,[\Phi(V)]\}$, which does not generally exist, or at least on uniform *divergence* from $\mathbf{E}\,[\Phi(V_0)]$. It should be noted that the issue here is whether the ML estimator at all converges, since if it does converge, it must converge to the minimizer of $\mathbf{E}\,[\Phi(V)]$.

Such convergence can be guaranteed at least in the special case of additive noise when the marginal noise density $p_Z(z_a)$ is continuous, strictly positive, and has finite variance and differential entropy. Under these conditions, the maximum likelihood estimator is guaranteed to converge to the minimizer of $\mathbf{E}\,[\Phi(V)]$ [76, Theorem 5.7].

Condition 4. $\mathrm{y} = \mathrm{x} + \mathrm{z}$, *where* z *is i.i.d and the density of each component of* z *is continuous, strictly positive, and has finite variance and differential entropy.*

Lemma 4. *Condition 4 gurantees the uniform law of large numbers for* $\{\Phi(V)\}$.

Proof. We will show that $\Phi(V)$ is continuous in V, which varies over a compact space (since we can view V as unit-norm matrices), and has a finite envelope:

$$\mathbf{E}\left[\max|\Phi(V)|\right] \leq \infty. \tag{4.7}$$

For additive noise models, $p_{Y|X}(\mathbf{y}|uV') = p_Z(\mathbf{y} - uV')$ goes to zero when u is far enough from the origin, and the infimum in (4.2) is always attained at finite u and can be replaced with a minimum. The random function

$$\Phi(V) = \min_u -\log p_Z(\mathbf{y} - uV') \tag{4.8}$$

is then continuous in V for all $\mathbf{y}$. Furthermore, we have:

$$\Phi(V) \geq -\max_z \log p_Z(z) \geq -\infty \tag{4.9}$$

ensuring

$$\mathbf{E}_Y\left[\max_V \Phi(V)\right] \geq -\infty. \tag{4.10}$$

On the other side, we have:

$$\begin{aligned}\mathbf{E}_Y\left[\Phi(V)\right] &\leq \mathbf{E}_Y\left[-\max_u \log p_{Y|X}(\mathbf{y}|uV')\right] \\ &\leq \mathbf{E}_Y\left[-\max_u \log p_{Y|X}(\mathbf{y}|0)\right] \\ &= D\left(p_Y \| p_{Y|0}\right) - Hp_Y \leq \infty \end{aligned} \tag{4.11}$$

where p_Y is the true distribution of $\mathbf{y}$ and $p_{Y|0}$ is the distribution $Y|X = 0$. Together, (4.11) and (4.10) provide a finite envolope for $\Phi(V)$. □

Proposition 5. *Conditions 4 and 1 together are sufficient for the consistency of the maximum likelihood estimator $\hat{V}$. Conditions 4 and 2 together are sufficient for the consistency of the maximum likelihood estimator $\hat{V}$ for any "signal" distribution* $\mathrm{x} = \mathrm{u}\, V_0$.

4.1.2 Additive Gaussian Noise

We first analyze the maximum likelihood estimator $\hat{V}$ of V_0 in the presence of i.i.d. addative Gaussian noise:

$$\mathbf{y} = \mathbf{x} + \mathbf{z}$$

where $\mathbf{x} = \mathbf{y}\, V_0$ and the the noise $\mathbf{z}$ is i.i.d. Gaussian. This is the standard Frobenius-distance minimizing estimator (PCA). By using Proposition 5 it is possible to show that the maximum likelihood estimator in this case is consistent.

Theorem 6. *For any signal distribution* $\mathbf{x} = \mathbf{u}\, V_0$, *and for additive i.i.d. Gaussian noise, the maximum likelihood estimator* $\hat{V}$ *is consistent.*

Proof. For a fixed subspace V, consider the decomposition $y = y_{\parallel} + y_{\perp}$ of vectors into their projection onto V, and the residual. As $\mathbf{z}$ is an isotropic Gaussian random vector, any rotation of $\mathbf{z}$ is also isotropic Gaussian, and so $\mathbf{z}_{\perp}$ and $\mathbf{z}_{\parallel}$ are independent, and we can decompose:

$$p_Y(y) = p_{\parallel}(y_{\parallel}) p_{\perp}(y_{\perp}) \tag{4.12}$$

We can now analyze:

$$\phi(V; y) = \inf_u (-\log p_{\parallel}(y_{\parallel} - uV') - \log p_{\perp}(y_{\perp})) = -\log p_{\parallel}(0) + \frac{1}{\sigma^2} |y_{\perp}|_2^2 + \text{Const} \tag{4.13}$$

yielding

$$\Psi(V; x) \propto \mathbf{E}_{\mathbf{z}_{\perp}} \left[|x_{\perp} + \mathbf{z}_{\perp}|_2 \right] + \text{Const}, \tag{4.14}$$

which is minimized when $x_{\perp} = 0$, i.e. x is spanned by V. This fulfills Condition 2. The Gaussian density fulfills Condition 4 and so by Proposition 5 $\hat{V}$ is consistent. □

This consistency proof employed a key property of the isotropic Gaussian: rotations of an isotropic Gaussian random variable remain i.i.d. As this property is unique to Gaussian random variables, maximum likelihood estimators in the presence of other conditional distributions (or even additive noise distributions) might not be consistent.

4.1.3 Inconsistency

To show inconsistency, we will analyze $\Psi(V;0)$ and use Condition 3 and Proposition 3. For some distributions, it is possible to evaluate $\Psi(V;0)$ analytically. In addition to the additive Gaussian noise, this can be done, for example, for additive Laplace noise, as well as for a logistic model.

Additive Laplace Noise

We now analyze the maximum likelihood estimator $\hat{V}$ of V_0 in the presence of i.i.d. addative Laplace noise:

$$\mathbf{y} = \mathbf{x} + \mathbf{z}$$

where $\mathbf{x} = \mathbf{y}\, V_0$ and the the noise $\mathbf{z}$ is i.i.d. with:

$$p_Z(z_a) = \frac{1}{2} e^{-|z_a|} \tag{4.15}$$

The log-likelihood

$$\log p_{\mathbf{y}|\mathbf{x}}(y|x) = -\log p_Z(y-x) = -\sum |y_a - x_a| - m = |y-x|_1 - m \tag{4.16}$$

is essentially the L_1-norm $|y-x|_1$. We focus on rank-one approximation in a two-dimensional space, that is, finding the direction in the plane (line through the origin) in which the signal resides.

Consider first a one dimensional subspace at an angle of $0 \leq \theta \leq \frac{\pi}{4}$ to the y_1 axis. That is, a one dimensional subspace spanned by $V_\theta = (1, \tan\theta)$, where $0 \leq \tan\theta \leq 1$. For any $y = (y_1, y_2)$ the L_1-norm $|y - uV'|_1 = |y_1 - u| + |y_2 - u\tan\theta|$ is minimized when $y_1 - u = 0$, i.e. $u = y_1$, yielding

$$\phi(V_\theta; z) = |z - z_1 V_\theta'|_1 + 2 = |z_2 - \tan\theta z_1| + 2. \tag{4.17}$$

We can now calculate:

$$
\begin{aligned}
\Psi(V_\theta;0) &= \mathbf{E}_{\mathbf{z}}\left[\phi(V_\theta;z)\right] = \int dz_1 \int dz_2 p_Z(z)\phi(V_\theta;z) \\
&= \int dz_1 \int dz_2 \frac{1}{4} e^{-|z_1|-|z_2|} |z_2 - \tan\theta z_1| dz_1 dz_2 + 2 \\
&= 2 + \frac{\tan^2\theta + \tan\theta + 1}{\tan\theta + 1} \qquad (4.18)
\end{aligned}
$$

which is monotonic increasing in θ in the valid range $[0, \frac{\pi}{4}]$, and $\Psi(V;0)$ is certainly not a constant function of V. The necessary Condition 3 does not hold, and the maximum likelihood estimator is not consistent.

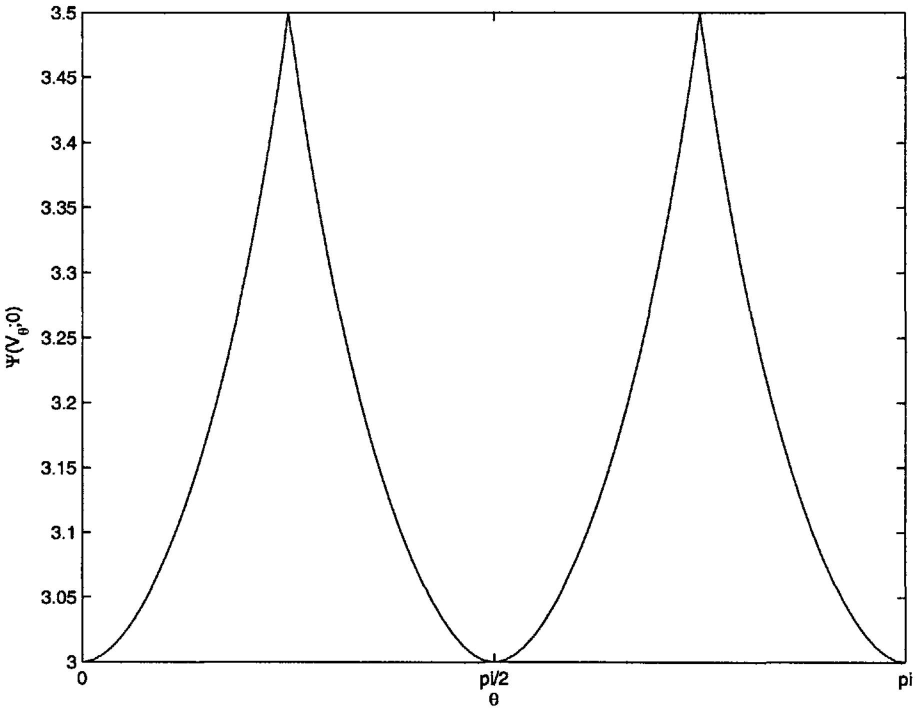


Figure 4-1: The function $\Psi(V_\theta;0)$ for one-dimensional subspaces $V_\theta \subset \mathbb{R}^2$ spanned by $(\cos\theta, \sin\theta)$, as a function of θ in the presence of additive Laplace noise.

To understand the asymptotic bias of the maximum likelihood estimator, Figure 4.1.3 displays the function $\Psi(V;0)$ as calculated above, for the entire range of directions (symmetry arguments apply to $\theta > \frac{\pi}{4}$). In particular, we have $3 = \Psi(V_0;0) < \Psi(V_{\frac{\pi}{4}};0) = \frac{7}{2}$. When no signal is present, the likelihood is maximized (Ψ is minimized) by axis-aligned subspaces. Even when a signal is present, the maximum likelihood estimator will be biased towards being axis-aligned.

Gaussian mixture additive noise

The asymptotic bias of the maximum likelihood estimator can also be observed empirically. To do so, we consider a two-component Gaussian mixture additive noise distribution. Although it is not possible to analyze the Ψ analytically in closed form for such a distribution, using the methods of Section 3.4, the maximum likelihood estimator $\hat{V}$ can be found.

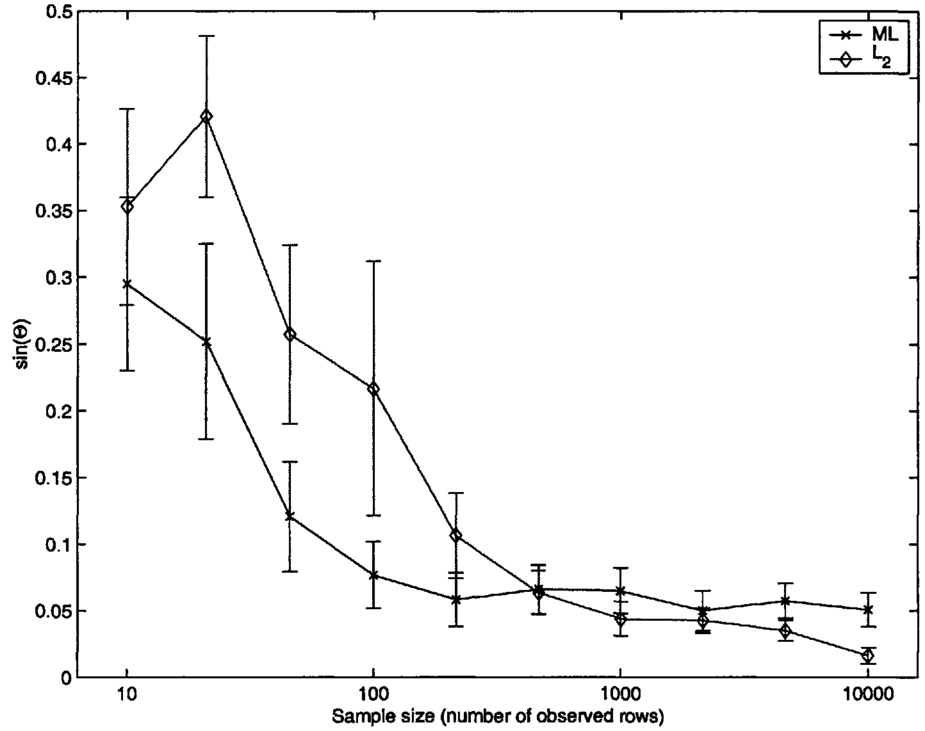


Figure 4-2: Norm of sines of canonical angles to the correct subspace span$(2, 1, 1)' \subset \mathbb{R}^3$ with a two-component Gaussian mixture $0.9\mathcal{N}(0, 1) + 0.1\mathcal{N}(0, 25)$ additive noise. The maximum likelihood estimator converges to $(2.34, 1, 1)$. Bars are one standard deviation tall.

Figure 4.1.3 demonstrates the asymptotic bias of the maximum likelihood estimator empirically. Two-component Gaussian mixture noise was added to rank-one signal in $\mathbb{R}^3$, and the signal subspace was estimated using a maximum likelihood estimator with known noise model, and a Frobenius estimator for comparison. For small data sets, the maximum likelihood estimator is more accurate, but as the number of samples increase, the error of the Frobenius estimator vanishes (see Section 4.2.1), while the maximum likelihood estimator converges to the wrong subspace.

Logistic conditional model

So far we discussed additive noise models. The Gaussian additive noise model can also be viewed as an instance of Exponential PCA, i.e. a conditional model in which the conditional distributions $\mathbf{y}_a|x_a$ form an exponential family, with x_a being the natural parameters. In fact, as mentioned in Section 2.1.2, Gaussian models are the only instance of additive noise models which form an exponential family distributions, with a natural parameterization.

We turn now to studying a different instance of Exponential PCA, which does not correspond to an additive noise model. We consider a logistic conditional model, where $\mathbf{y}_a \in \pm$ and x_a are natural parameters:

$$p(\mathbf{y}_a = +|x_a) = \frac{1}{1+e^{-x_a}} = g(x_a) \tag{4.19}$$

where $g(\cdot)$ is the logistic function.

We again focus on a estimating a rank-one subspace in two dimensions and analyze $\Psi(V;0)$ for all rank-one subspaces, i.e. lines through the origin, V. A setting of $x = 0$ implies a uniform distribution on $\mathbf{y}$ over $(+,+), (+,-), (-,+), (-,-)$ and we have:

$$\begin{aligned}\Psi(V;0) &= \mathbf{E}_{\mathbf{y}|x=0}\left[\phi(V;\mathbf{y})\right]\\ &= \frac{1}{4}\phi(V;{+}{+}) + \frac{1}{4}\phi(V;{-}{+}) + \frac{1}{4}\phi(V;{+}{-}) + \frac{1}{4}\phi(V;{-}{-})\end{aligned} \tag{4.20}$$

We analyze the axis-parallel and non-axis-parallel subspaces V separately.

For an axis-parallel subspace, e.g. $V_0 = (1,0)$ without loss of generality, we have, for any $y \in \{\pm\}^2$:

$$\begin{aligned}\phi((1,0);y) &= -\sup_u \log p(y|u(1,0)) = -\sup_u (\log p(y_1|u) + \log p(y_2|0)) \\ &= -\sup_u \log p(y_1|u) - \log p(y_2|0) = -\log 1 - \log\frac{1}{2} = 1 \qquad (4.21)\end{aligned}$$

and so:

$$\Psi(V_0;0) = \Psi(V_{\frac{\pi}{2}};0) = 1 \qquad (4.22)$$

We now turn to analyzing non-axis-parallel subspaces $V_\theta = (1, \tan\theta)$, where without loss of generality we concentrate on $0 < \theta \leq \frac{\pi}{2}$. For $y = ++$ and $y = --$ we can push the likelihood $p(y|u(1,\lambda))$ to one by pushing u to infinity or negative infinity, and therefore:

$$\phi((1,\tan\theta);++) = \phi((1,\tan\theta);--) = -\log 1 = 0 \qquad (4.23)$$

For $y = +-$, increasing u increases $p(\mathbf{y}_1 = +|u)$ but decreases $p(\mathbf{y}_2 = -|u\tan\theta)$. The value of u maximizing

$$\begin{aligned}\phi((1,\tan\theta);+-) &= -\sup_u \log p(+-|u, u\tan\theta) \\ &= -\sup_u (\log g(u) + \log g(-u\tan\theta)), \qquad (4.24)\end{aligned}$$

can be found by setting the derivative of the log-likelihood to zero:

$$0 = \frac{\partial}{\partial u} = \log p(+-|u, u\tan\theta) = g(-u) - (\tan\theta)g(u\tan\theta) = \frac{1}{1+e^u} - \frac{\tan\theta}{1+e^{-u\tan\theta}}$$

$$\Downarrow$$

$$1 + \tan\theta + \tan\theta\,(e^u) + (e^u)^{-\tan\theta} = 0 \qquad (4.25)$$

Due to $u \leftrightarrow -u$ symmetry, we have $\phi(V;+-) = \phi(V;-+)$.

For the mid-axis diagonal $V_{\frac{\pi}{4}} = (1,1)$, the maximum in (4.24) is obtained at $u = 0$

(this can also be seen from symmetry considerations), yielding

$$\phi(V_{\frac{\pi}{4}};+-) = \phi(V_{\frac{\pi}{4}};-+) = -\log p(+-|0,0) = -\log\frac{1}{4} = 2 \tag{4.26}$$

and combined with (4.23),

$$\Psi(V_{\frac{\pi}{4}};0) = \frac{1}{4}0 + \frac{1}{4}0 + \frac{1}{4}2\frac{1}{4}2 = 1. \tag{4.27}$$

However, for $0 < \theta < \frac{\pi}{4}$, a lower value is obtained for $m(V_\theta;+-)$. In particular, when θ approaches zero, the optimizing u (the solution of (4.25)) approaches $-\ln\frac{\tan\lambda}{2}$ and

$$\phi(V_\theta;+-) = \phi(V_\theta;-+) = \xrightarrow{\theta\rightarrow 0} 1. \tag{4.28}$$

Combined with (4.23), we therefore have:

$$\Psi(V_\theta;0) \xrightarrow{\theta\rightarrow 0} \frac{1}{4}0 + \frac{1}{4}0 + \frac{1}{4}1\frac{1}{4}1 = \frac{1}{2}. \tag{4.29}$$

The expected contribution of the negative log-likelihood from a distribution generated by parameters on the origin $\Psi(V_\theta;0)$, decreases as the subspace becomes close to being axis aligned, as long as it is not completely axis aligned. At the axis-aligned subspaces, we observe a discontinuity, with the value at these subspaces substantially higher (in fact, the highest possible, and equal to the value on the mid-axis diagonals). The quantitative behavior of $\Psi(V_\theta;0)$ as a function of θ is presented in Figure 4-3. This behavior indicates that the maximum likelihood estimator for this setting is biased towards being axis aligned (though not on the axis itself).

Bernoulli conditional models

The next model we analyze is also a binary observation model, $\mathbf{y}_a \in \pm$, but unlike the logistic conditional model in which x_a are *natural* parameters, here x_a are taken to be *mean* parameters:

$$p(\mathbf{y}_a = +|x_a) = x_a \tag{4.30}$$

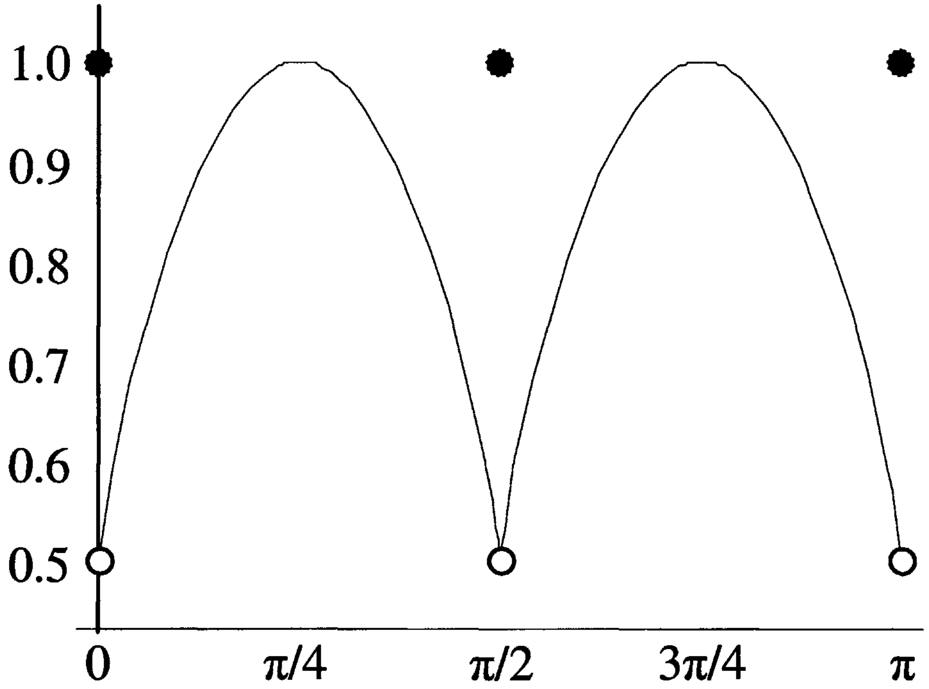


Figure 4-3: $M(V_\theta; 0)$ for a logistic conditional model

where $0 \leq x_a \leq 1$.

We again analyze estimation of a rank-one subspace (line through the origin) in two dimensions.

For $x_a = 0$ we have $\Pr(y = --|0) = 1$. For any subspace V, by choosing $u = 0$, we have $\phi(V_\theta; --) = -\log 1 = 0$ and so $\Psi(V; 0) = 0$. The necessary condition 3 is satisfied, and the estimator might be consistent. However, this is only a necessary condition, and does not imply the consistency of the estimator. In order to study the consistency, we now turn to analyzing $\Psi(V; x)$ for any $x \in [0, 1]^2$.

Consider $V_\theta = (1, \tan\theta)$, where without loss of generality $0 \leq \theta \leq \frac{\pi}{4}$, i.e. $0 \leq tan\theta \leq 1$ (calculations for other subspaces can be extrapolated by symmetry). We can calculate (all maximization are constrained by $0 \leq u \leq 1$ and $0 \leq u \tan\theta \leq 1$):

$$\phi(V_\theta; ++) = -\sup_u \log\left(u(u\tan\theta)\right) = -\log\tan\theta \tag{4.31}$$

$$\phi(V_\theta; --) = -\sup_u \log\left((1-u)(1-u\tan\theta)\right) = -\log 1 = 0 \tag{4.32}$$

$$\phi(V_\theta; +-) = -\sup_u \log\left(u(1-u\tan\theta)\right)$$

Solving $0 = \frac{\partial u(1-u\tan\theta)}{\partial u} = 1 - 2u\tan\theta$ yields $u = \frac{1}{2\tan\theta}$ which is in the legal domain only when $\tan\theta \geq \frac{1}{2}$, and so:

$$= \begin{cases} \text{when } \tan\theta \geq \frac{1}{2}, & -\log\left(\frac{1}{2\tan\theta}(1-\frac{1}{2})\right) = 2 + \log\tan\theta \\ \text{when } \tan\theta \leq \frac{1}{2}, & -\log\left(1-\tan\theta\right) \end{cases} \tag{4.33}$$

Symmetrically:

$$\phi(V_\theta; -+) = \begin{cases} \text{when } \cot\theta \leq \frac{1}{2}, & 2 + \log\cot\theta \\ \text{when } \cot\theta \leq \frac{1}{2}, & -\log\left(1-\cot\theta\right) \end{cases} \tag{4.34}$$

From (4.31)-(4.34) we can calculate

$$\Psi(V_\theta; x) = x_1 x_2 \phi(V_\theta; ++) \; + \; x_1(1-x_2)\phi(V_\theta; +-) \; + \\ (1-x_1)x_2\phi(V_\theta; -+) \; + \; (1-x_1)(1-x_2)\phi(V_\theta; --) \tag{4.35}$$

and check whether it is indeed minimized, with respect to θ, when $x \in V_\theta$, as Condition 2 requires.

Numerical calculations reveal that $M(V_\theta; x)$ is generally *not* minimized when $x \in V_\theta$. Biases in different directions, and different magnitudes, are observed for different $x \in [0,1]^2$. Figure 4.1.3 displays the bias, from the true direction of x to the direction θ minimizing $M(V_\theta; x)$.

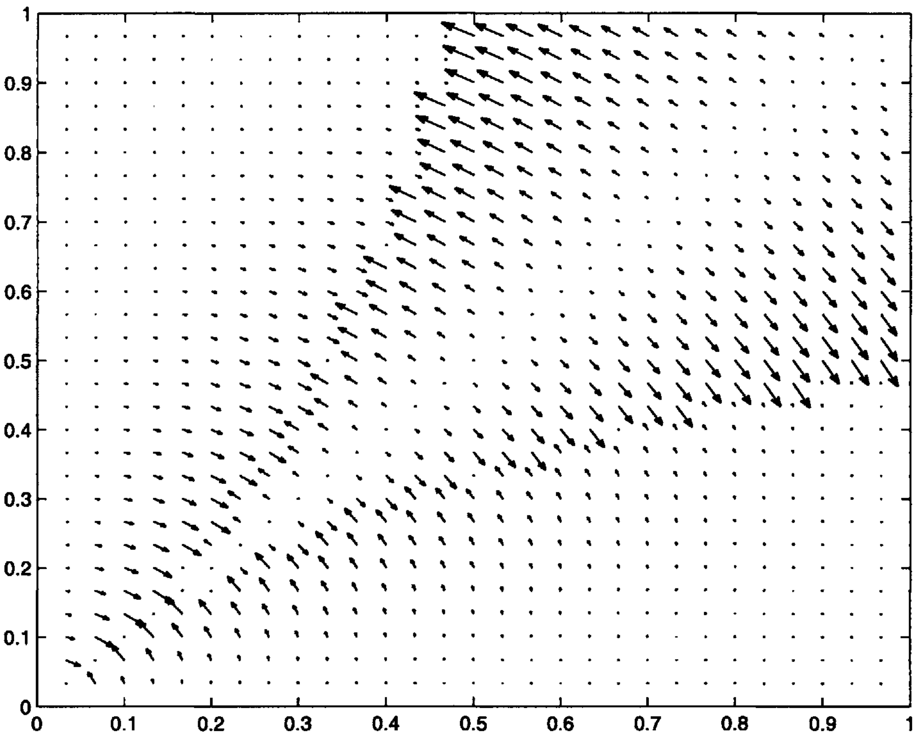


Figure 4-4: Bias in maximum likelihood estimation of rank-one subspace in $\mathbb{R}^2$ for a Bernoulli conditional model. The arrow at each point $x \in [0, 1]^2$ indicates the offset from the true direction of x to the direction θ minimizing $\Psi(V_\theta; x$, to which maximum likelihood estimation will converge.

Uniform additive noise

Before concluding that the maximum likelihood estimator is only consistent with additive Gaussian noise, consider maximum likelihood estimation in the presence of additive i.i.d. uniform noise. The likelihood in a uniform model is either some positive constant, if all errors are within the support, or zero otherwise. Thus, the maximum likelihood estimator is any low-rank matrix that is consistent with the required margin of error. But for any incorrect low-rank subspace, there exists some positive probability of producing noise incompatible with it. Hence, the only low-rank subspace which is compatible with probability one is V_0, and the estimator is consistent.

4.2 Universal Consistency of Subspace Estimation

In contrast to the lack of consistency of maximum likelihood estimation, we show here that the "standard" approach to low-rank approximation, minimizing the sum squared error (i.e. Frobenius estimation), yields a universal estimator which is consistent for any additive noise model. Minimizing the sum squared error corresponds to maximum likelihood estimation in the presence of Gaussian additive noise. We already saw in the previous section that in the presence of additive noise that does indeed follow a Gaussian distribution, this estimator is consistent. Here, we establish a much stronger result: the Frobenius estimator is consistent for *any* additive i.i.d. noise distribution.

For non-additive conditional models $\mathrm{y}|\mathrm{x}$, the Frobenius estimator might not be consistent. In Section 4.2.2 we relax this requirement, and require only that the conditional model is *unbiased*, i.e. $\mathbf{E}\left[\mathrm{y}|\mathrm{x}\right] = \mathrm{x}$. This happens, for example, in the presence of multiplicative noise (a constant bias can easily be corrected) or when the conditional distribution $\mathrm{y}|\mathrm{x}$ form an exponential family with x being the *mean* parameters.

We suggest a modified universal estimator, the variance-ignoring estimator, that is appropriate for unbiased conditional models.

4.2.1 Additive noise

Consider the "standard" approach to low-rank approximation, minimizing the sum squared error, and the corresponding Frobenius estimator for signal subspace V:

$$\hat{V}_{\mathrm{ML}} = \arg\min_{V} \inf_{U} \| Y - UV' \|_{\mathrm{Fro}} \tag{4.36}$$

Theorem 7. *For any i.i.d. additive noise model* $\mathrm{y} = \mathrm{u}V_0 + \mathrm{z}$ *(coordinates of* z *are i.i.d. and independent of* x*), where* u *and* z *have finite fourth moments, the Frobenius estimator* (4.36) *is a consistent estimator of* V_0.

Proof. The Frobenius estimator of the signal subspace is the subspace spanned by the leading eigenvectors of the empirical covariance matrix $\hat{\Lambda}_n$ of y. Assuming the fourth moments of the distribution of y are finite, the empirical covariance matrix $\hat{\Lambda}_n$ converges to the true covariance matrix Λ_Y, which in turn is the sum of the covariance matrices of $\mathbf{x}$ and $\mathbf{z}$:

$$\hat{\Lambda}_n \rightarrow \Lambda_Y = \Lambda_X + \Lambda_Z \tag{4.37}$$

The covariance Λ_X of $\mathbf{x}$ is a matrix of rank k, and since $\mathbf{z}$ is i.i.d., $\Lambda_Z = \sigma^2 I$. We should also be careful about signals that occupy only a proper subspace of V_0, and be satisfied with any rank-k subspace containing the support of $\mathbf{x}$, but for simplicity of presentation we assume this does not happen and $\mathbf{x}$ is of full rank k.

Let $s_1 \geq s_2 \geq \cdots \geq s_k > 0$ be the non-zero eigenvalues of Λ_X. Since $\mathbf{z}$ has variance exactly σ^2 in any direction, the principal directions of variation are not affected by it, and the eigenvalues of Λ_Y are exactly $s_1+\sigma^2, \ldots, s_k+\sigma^2, \sigma^2, \ldots, \sigma^2$, with the leading k eigenvectors being the eigenvectors of Λ_X. This ensures an eigenvalue gap of $s_k > 0$ between the invariant subspace of Λ_Y spanned by the eigenvectors of Λ_X and its complement, and we can bound the norm of the canonical sines between V_0 and the leading k eigenvectors of $\hat{\Lambda}_n$ by $\frac{|\hat{\Lambda}_n - \Lambda_Y|}{s_k}$ [71]. Since $|\hat{\Lambda}_n - \Lambda_Y| \rightarrow 0$ a.s., we conclude that the estimator is consistent. □

It is interesting to note that even though the standard Frobenius estimator is consistent, while the maximum likelihood estimator is not consistent, empirical results (Figure 4.1.3) demonstrate that on reasonable-sized samples the maximum likelihood estimator outperforms the Frobenius estimator. But as the sample size increases, the Frobenius estimated subspace converges to the correct subspace, whereas the maximum likelihood subspace converges to the wrong subspace.

4.2.2 Unbiased Noise and the Variance-Ignoring Estimator

Before discussing unbiased noise, let us turn our attention to additive noise with independent, not not identically distributed, coordinates. This is essentially the classic setting of factor analysis.

Non-identical Additive Noise

Consider an additive model $\mathbf{y} = \mathbf{x}+\mathbf{z}$ where the components of the noise $\mathbf{z}$ are independent, but not necessarily identical, Gaussians. If the noise variances are known, the ML estimator corresponds to minimizing the column-weighted (inversely proportional to the variances) Frobenius norm of $Y - X$, and can be calculated from the leading eigenvectors of a scaled empirical covariance matrix [41]. If the variances are not known, e.g. when the scale of different coordinates is not known, there is no maximum likelihood estimator: at least k coordinates of each y can always be exactly matched, and so the likelihood is unbounded when up to k variances approach zero.

The Frobenius estimator is not appropriate in this scenario. The covariance matrix Λ_Z is still diagonal, but is no longer a scaled identity. The additional variance introduced by the noise is different in different directions, and these differences may overwhelm the "signal" variance along V_0, biasing the leading eigenvectors of Λ_Y, and thus the Frobenius estimator, toward axes with high "noise" variance. The fact that this variability is independent of the variability in other coordinates is ignored, and the Frobenius estimator is asymptotically biased.

The Variance-Ignoring Estimator

Instead of recovering the directions of greatest variability, we can recover the covariance structure directly. In the limit, we still have $\hat{\Lambda}_n \to \Lambda_Y = \Lambda_X + \Lambda_Z$, a sum of a rank-$k$ matrix and a diagonal matrix. In particular, the non-diagonal entries of $\hat{\Lambda}_n$ approach those of Λ_X. We can thus seek a rank-k matrix $\hat{\Lambda}_X$ approximating $\hat{\Lambda}_n$, e.g. in a sum-squared sense, except on the diagonal. This is a (zero-one) *weighted* low-rank approximation problem, and the methods of Section 3.2 apply. The row-space of the resulting $\hat{\Lambda}_X$ is then an estimator for the signal subspace. Note that the Frobenius estimator is the row-space of the rank-k matrix minimizing the *unweighted* sum-squared distance to $\hat{\Lambda}_n$.

Although in most cases the Variance-Ignoring estimator *will* converge to the correct subspace, discussing consistency in the presence of non-identical noise with unknown variances is problematic: The signal subspace is not necessarily identifiable. For example, the combined covariance matrix $\Lambda_Y = \left(\begin{smallmatrix}2&1\\1&2\end{smallmatrix}\right)$ can arise from a rank-one signal covariance $\Lambda_X = \left(\begin{smallmatrix}a&1\\1&1/a\end{smallmatrix}\right)$ for any $\frac{1}{2} \leq a \leq 2$, each corresponding to a different signal subspace. Counting the number of parameters and constraints suggests identifiability when $k < m - \frac{\sqrt{8m+1}-1}{2}$, but this is by no means a precise guarantee. Anderson and Rubin [5] present several conditions on Λ_X which are sufficient for identifiability but require $k < \left\lfloor \frac{m}{2} \right\rfloor$, and other weaker conditions which are necessary.

Non-Additive Models

The above estimation method is also useful in a less straight-forward situation. Until now we have considered only additive noise, in which the distribution of $\mathrm{y}_a - \mathrm{x}_a$ was independent of x_a. We will now relax this restriction and allow more general conditional distributions $\mathrm{y}_a|\mathrm{x}_a$, requiring only that $\mathbf{E}\left[\mathrm{y}_a|\mathrm{x}_a\right] = \mathrm{x}_a$. With this requirement, together with the structural constraint (y_a independent given x), for any $i \neq j$:

$$\begin{aligned}\mathrm{Cov}\left[\mathrm{y}_a, \mathrm{y}_b\right] &= \mathbf{E}\left[\mathrm{y}_a\mathrm{y}_b\right] - \mathbf{E}\left[\mathrm{y}_a\right]\mathbf{E}\left[\mathrm{y}_b\right] = \mathbf{E}\left[\mathbf{E}\left[\mathrm{y}_a\mathrm{y}_b|\mathrm{x}\right]\right] - \mathbf{E}\left[\mathbf{E}\left[\mathrm{y}_a|\mathrm{x}\right]\right]\mathbf{E}\left[\mathbf{E}\left[\mathrm{y}_b|\mathrm{x}\right]\right] \\ &= \mathbf{E}\left[\mathbf{E}\left[\mathrm{y}_a|\mathrm{x}\right]\mathbf{E}\left[\mathrm{y}_b|\mathrm{x}\right]\right] - \mathbf{E}\left[\mathrm{x}_a\right]\mathbf{E}\left[\mathrm{x}_b\right] = \mathbf{E}\left[\mathrm{x}_a\mathrm{x}_b\right] - \mathbf{E}\left[\mathrm{x}_a\right]\mathbf{E}\left[\mathrm{x}_b\right] = \mathrm{Cov}\left[\mathrm{x}_a, \mathrm{x}_b\right]. \qquad (4.38)\end{aligned}$$

As in the non-identical additive noise case, Λ_Y agrees with Λ_X except on the diagonal. Even if $\mathrm{y}_a|\mathrm{x}_a$ is identically conditionally distributed for all i, the difference $\Lambda_Y - \Lambda_X$ is *not* in general a scaled identity:

$$\begin{aligned}\mathrm{Var}\left[\mathrm{y}_a\right] &= \mathbf{E}\left[\mathrm{y}_a^2\right] - \mathbf{E}\left[\mathrm{y}_a\right]^2 \\ &= \mathbf{E}\left[\mathbf{E}\left[\mathrm{y}_a^2|\mathrm{x}_a\right] - \mathbf{E}\left[\mathrm{y}_a|\mathrm{x}_a\right]^2\right] + \mathbf{E}\left[\mathbf{E}\left[\mathrm{y}_a|\mathrm{x}_a\right]^2\right] - \mathbf{E}\left[\mathrm{y}_a\right]^2 \\ &= \mathbf{E}\left[\mathrm{Var}\left[\mathrm{y}_a|\mathrm{x}_a\right]\right] + \mathbf{E}\left[\mathrm{x}_a^2\right] - \mathbf{E}\left[\mathrm{x}_a\right]^2 \\ &= \mathbf{E}\left[\mathrm{Var}\left[\mathrm{y}_a|\mathrm{x}_a\right]\right] + \mathrm{Var}\left[\mathrm{x}_a\right]. \qquad (4.39)\end{aligned}$$

Unlike the additive noise case, the variance of $\mathrm{y}_a|\mathrm{x}_a$ depends on x_a, and so its expectation depends on the distribution of x_a.

These observations suggest using the variance-ignoring estimator. Figure 4.2.2 demonstrates how such an estimator succeeds in reconstruction when $\mathrm{y}_a|\mathrm{x}_a$ is exponentially dis-

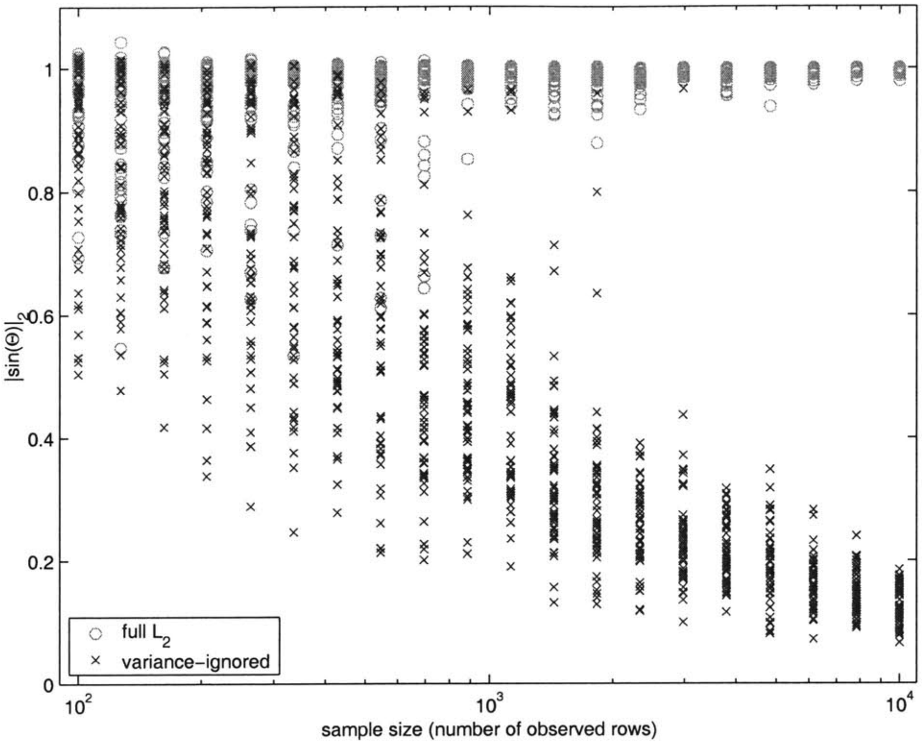


Figure 4-5: Norm of sines of canonical angles to correct subspace: Observations are exponentially distributed with means in rank-2 subspace $\left(\begin{smallmatrix}1&1&1&1&1&1&1&1&1&1\\1&0&1&0&1&0&1&0&1&0\end{smallmatrix}\right)'$.

tributed with mean $\mathbf{x}_a$, even though the standard Frobenius estimator is not applicable. We cannot guarantee consistency because the decomposition of the covariance matrix might not be unique, but when $k < \left\lfloor \frac{d}{2} \right\rfloor$ this is not likely to happen. Note that if the conditional distribution $\mathbf{y}|\mathbf{x}$ is known, even if the decomposition is not unique, the correct signal covariance might be identifiable based on the relationship between the signal marginals and the expected conditional variance of of $\mathbf{y}|\mathbf{x}$, but this is not captured by the variance-ignoring estimator.

4.2.3 Biased noise

The variance-ignoring estimator is also applicable when $\mathbf{y}$ can be transformed such that $\mathbf{E}\left[g(\mathbf{y})|\mathbf{u}\right]$ lie in a low-rank linear space, e.g. in log-normal models. If the conditional distribution $\mathbf{y}|\mathbf{x}$ is known, this amount to obtaining an unbiased estimator for $\mathbf{x}_a$. When such a transformation is not known, we may wish to consider it as nuisance. In particular, this might be possible when $g(\mathbf{y}) - \mathbf{x}$ is Gaussian. However, this does not provide a general *consistent* estimator for V if no *unbiased* estimator exists for $\mathbf{x}_a$.

Of particular interest are distributions $\mathbf{y}_a|\mathbf{x}_a$ that form exponential families where $\mathbf{x}_a$ are the *natural* parameters (Exponential-PCA). When the *mean* parameters form a low-rank linear subspace, the variance-ignoring estimator is applicable, but when the natural parameters form a linear subspace, the means are in general curved, and there is no unbiased estimator for the natural parameters.

The problem of finding a consistent estimator for the linear-subspace of natural parameters when $\mathbf{y}_a|\mathbf{x}_a$ forms an exponential family, or in other general settings, remains open.

Chapter 5

Maximum Margin Matrix Factorization

In the previous chapters of this thesis we limited ourselves to learning with low-rank matrices. In terms of the factorization $X = UV'$, we constrained the dimensionality of U and V. In this chapter we suggest constraining the factorization by constraining the norms of U and V, yielding a novel class of factorisable matrices. We show how these constraints arise naturally when matrix factorizations are viewed as feature learning for large-margin linear prediction, and how they lead to *convex* optimization problems that can be formulated as semi-definite programs.

We present maximum margin matrix factorizations in the context of "collaborative prediction": predicting unobserved entries of a target matrix, based on a subset of observed entries. We begin the chapter by discussing this setting and defining the framework.

5.1 Collaborative Filtering and Collaborative Prediction

"Collaborative filtering" refers to the general task of providing users with information on what items they might like, or dislike, based on their preferences so far (perhaps as inferred from their actions), and how they relate to the preferences of other users. For example, in a collaborative filtering movie recommendation system, the inputs to the system are user ratings on movies they have already seen. Prediction of user preferences on movies they have not yet seen are then based on patterns in the partially observed rating matrix, e.g. predicting preferences which correlate with the rating of the movie by other users with overall similar preferences, but anti-correlate with users with distinctly opposite ratings. This approach contrasts with a more traditional feature-based approach where predictions are made based on features of the movies (e.g. genre, year, actors, external reviews) and the users (e.g. age, gender, explicitly specified preferences). Users "collaborate" by sharing their ratings instead of relying on external information.

In some collaborative filtering tasks, the preferences might not be given explicitly by the user, but rather inferred from the user's actions. The input for such tasks typically consists of the items each user has already requested (e.g. web pages visited, items purchased), and the goal is to predict which further items the user is likely to request. Note that generally no negative data is available in such situations.

The desired output of collaborative filtering varies by application. One type of often useful output, is to predict for each user a few items the user is highly likely to like. This answers a user query of the form "What movie should I go see?", or can be used to place recommended links on a web page. Here, we will focus on *"collaborative prediction"*: predicting the user's preference regarding each item, answering queries of the form "Will I like this movie?". Although it is certainly possible to use this output in order to generate a list of the predicted most strongly preferred items, answering the first type of query, this requires going over all possible items, which is often impractical. Furthermore, it might be possible to find a few items with high certainty the user will like them, even when the collaborative prediction problem is hard. Therefore, although the basic ideas studied in the thesis may be relevant for both types of tasks, methods for efficiently predicting the top items may be substantially different, and lie outside the scope of the thesis.

5.1.1 Matrix Completion

In this thesis, collaborative prediction is formalized as a simple matrix-completion problem: predicting the unobserved entries of a partially observed target matrix. A subset of entries S of a target matrix Y is observed. Based on the observed values Y_S, and no other external information, we would like to predict all other values in Y.

A key issue is how the *discrepancy* between the target values in Y and the predictions

X is measured. For the collaborative prediction tasks studied, we would like to ensure each entry is correctly predicted, and accordingly use a per-entry loss function:

$$\mathcal{D}(X;Y) = \sum_{ia} \operatorname{loss}(X_{ia};Y_{ia}). \tag{5.1}$$

The matrix completion formulation is also appropriate for other applications, such as filling in missing values in a mostly observed matrix of experiment results. Such a situation is often encountered in gene expression analysis, where the expression levels of thousands of genes are measured across different "experiments" (different experimental conditions, time points, cell-types, etc), but where some entries in the experiment-gene matrix might be missing [75]. The main difference between such applications and typical collaborative filtering applications is the observation sparseness: whereas when completing experimental results, only a small proportion matrix entries are missing, the typical situation in collaborative filtering is that only a small fraction of entries are observed.

A learning task with a somewhat similar formulation, but different measure of discrepancy, is completing a partially observed covariance matrix. In such situations, the measure of discrepancy is not a per-entry sum-of-losses measure, as we are not usually interested in each covariance separately, but rather a measure of discrepancy between the implied joint distributions.

5.2 Matrix Factorization for Collaborative Prediction

Using matrix factorization for matrix completion is fairly straight forward. A factorization (U, V) is sought that minimizes the discrepancy between the observed entries Y_S and the corresponding entries of $X = UV'$. Unobserved entries in Y are then predicted according to the corresponding entries in X.

5.2.1 Low Rank Matrix Completion

Several authors have recently suggested, and experimented with, low-rank (unconstrained or almost unconstrained) matrix factorization for collaborative prediction. Methods mostly differ in how they relate real-valued entries in X to preferences in Y, and in the associated measure of discrepancy.

Hoffman [39] suggests a collaborative prediction method based on a probabilistic latent variable model, which corresponds to view the entries in X as mean parameters for a probabilistic model of the entries of Y, and fitting X by maximizing the likelihood. Marlin [52] also studied low-rank collaborative prediction, viewing the entries of X as mean parameters. In [69] we study low-rank collaborative prediction, both with a sum-squared loss, and viewing X as natural parameters to Bernoulli distributions on the entries of Y, yielding to a logistic loss. In a recent paper, Marlin *et al* [53] also implicitly suggest fitting X as natural parameters.

Azar *et al* [8] proved asymptotic consistency of a method in which unobserved entries are replaced by zeros, observed entries are scaled inversely proportionally to the probability of them being observed, and a squared error loss is used. No guarantees are provided for finite data sets.

Other have suggested using a low-rank approximation in combination with other methods. Goldberg *et al* [29] use a low-rank approximation of a fully-observed subset of columns of the matrix, thus avoiding the need to introduce weights. Billsus *et al* [13] use a singular value decomposition of a sparse binary observation matrix. Both Goldberg and Billsus use the low-rank approximation only as a preprocessing step, and then use clustering (Goldberg) and neural networks (Billsus) to learn the preferences.

Extensive experiments with various collaborative prediction methods can be found in Marlin's MSc thesis [51].

5.2.2 Matrix factorization and linear prediction

If one of the factor matrices, say U, is fixed, and only the other factor matrix V' needs to be learned, then fitting each column of the target matrix Y is a separate linear prediction problem. Each row of U functions as a "feature vector", and each column of V' is linear predictor, predicting the entries in the corresponding column of Y based on the "features" in U.

In the matrix factorization approach to matrix completion, both U and V are unknown and need to be estimated. This can be thought of as learning feature vectors (rows in U) for each of the rows of Y, enabling good linear prediction of all of the prediction problems (columns of Y) concurrently, each with a different linear predictor (columns of V'). Since the factorization is symmetric, the symmetric view, of learning features for the column enabling good linear prediction of the rows, is equally valid.

In collaborative prediction, both U and V are unknown and need to be estimated. This can be thought of as learning feature vectors (rows in U) for each of the rows of Y, enabling good linear prediction across all of the prediction problems (columns of Y) concurrently, each with a different linear predictor (columns of V'). The features are learned without any external information or constraints which is impossible for a single prediction task (we would use the labels as features). The underlying assumption that enables us to do this in a collaborative filtering situation is that the prediction tasks (columns of Y) are *related*, in that the same features can be used for all of them, though possibly in different ways.

The symmetric view, of learning features for the column enabling good linear prediction of the rows, is equally valid.

Low-rank collaborative prediction corresponds to regularizing by limiting the dimensionality of the feature space—each column is a linear prediction problem in a k-dimensional space. Instead, we suggest allowing an unbounded dimensionality for the feature space, and regularizing by requiring a low-norm factorization, while predicting with large-margin.

5.2.3 Maximum Margin Matrix Completion

Consider adding to the loss a penalty term which is the sum of squares of entries in U and V, i.e. $||U||^2_{\text{Fro}} + ||V||^2_{\text{Fro}}$ ($|| ||_{\text{Fro}}$ denotes the Frobenius norm). Each "conditional" problem (fitting U given V and vice versa) again decomposes into a collection of standard, this time regularized, linear prediction problems. With an appropriate loss function, or constraints on the observed entries, these correspond to large-margin linear discrimination problems. For example, if we learn a binary observation matrix by minimizing a hinge loss plus such a regularization term, each conditional problem decomposes into a collection of SVMs.

5.3 Maximum Margin Matrix Factorizations

We present two variations of maximum margin matrix factorizations, corresponding to two different norms, or constraints on U and V.

5.3.1 Constraining the Factor Norms on Average: The Trace-Norm

The squared Frobenius norms $\|U\|^2_{\text{Fro}}$ and $\|V\|^2_{\text{Fro}}$ are the sums of the L_2 norms of the rows of U and V. That is, constraining the Frobenius norm is in a sense a constraint *on average*: taking the features-and-predictors view, any particular predictor may have large norm (small margin), but on average, the predictors must have a small norm.

The Trace-Norm

Matrices with a factorization $X = UV'$, where U and V have low Frobenius norm (recall that the dimensionality of U and V is no longer bounded!), can be characterized in several equivalent ways, and are known as low *trace-norm* matrices:

Lemma 8. *For any matrix X the following are all equal:*

1. $\min_{\substack{U,V \\ X=UV'}} \|U\|_{Fro} \|V\|_{Fro}$
2. $\min_{\substack{U,V \\ X=UV'}} \frac{1}{2}(\|U\|^2_{Fro} + \|V\|^2_{Fro})$
3. *The sum of the singular values of X, i.e.* $\operatorname{tr} S$ *where $X = U\Lambda V'$ is the singular value decomposition of X.*

Furthermore, If $X = U\Lambda V'$ is the singular value decomposition of X, then the matrices $U\sqrt{S}$ and $V\sqrt{S}$ minimize the first quantity.

Definition 2. *The trace-norm $\|X\|_{\text{tr}}$ of a matrix is given by the above quantities.*

It is also known as the *nuclear norm* [26] and the *Ky-Fan n-norm* (e.g. [40]).

The trace-norm is, in fact, a matrix norm, and in particular it is a convex function, and the set of bounded trace-norm matrices is a convex set. For convex loss functions, seeking a bounded trace-norm matrix minimizing the loss versus some target matrix is a convex optimization problem. This contrasts sharply with minimizing loss over low-rank matrices—a non-convex problem.

In fact, the trace-norm has been suggested as a convex surrogate to the rank for various rank-minimization problems [26], noting that:

Lemma 9 ([26, Theorem 1]). *The convex envelope (smallest bounding convex function) of the rank function, on matrices with unit spectral norm, is the trace-norm.*

Here, we justify the trace-norm directly, both as a natural extension of large-margin methods and by providing generalization error bounds (Section 6.2).

The relationship of the trace-norm to the Frobenius norm and the rank of a matrix is given by the following bounds:

Lemma 10. *For any matrix X:*

$$\|X\|_{Fro} \leq \|X\|_{\mathrm{tr}} \leq \sqrt{\operatorname{rank} X}\,\|X\|_{Fro}$$

Proof. Recall that the Frobenius norm $\|X\|_{\mathrm{Fro}}$ is equal to the L_2 (Euclidean) vector norm of the singular values of $\|X\|_{\mathrm{Fro}}$, while the trace-norm $\|X\|_{\mathrm{tr}}$ is the L_1 norm of the singular values. The relationship between the L_1 and L_2 vector norms establish the left inequality. To establish the right inequality, recall that the number of non-zero singular values is equal to the rank. □

Furthermore, we can characterize the unit ball of the trace-norm

$$\mathcal{B}_{\mathrm{tr}} \doteq \{X \mid \|X\|_{\mathrm{tr}} \leq 1\} \tag{5.2}$$

in terms of the convex hull of unit-norm rank-one matrices

$$\mathcal{X}_1[1] \doteq \left\{uv' \,|u \in \mathbb{R}^n, v \in \mathbb{R}^m, |u|^2 = |v|^2 = 1\right\} \tag{5.3}$$

Lemma 11. $\mathcal{B}_{\mathrm{tr}} = \mathrm{conv}\mathcal{X}_1[1]$

Proof. Lemma 10 ensures that $\mathcal{X}_1[1] \subseteq \mathcal{B}_{\mathrm{tr}}$, and combined with the convexity of the trace-norm we have $\mathrm{conv}\mathcal{X}_1[1] \subseteq \mathcal{B}_{\mathrm{tr}}$. For a unit trace-norm matrix X with singular value decomposition $X = USV'$ with tr $S = 1$, we can write $X = \sum_i S_i U_i V_i'$ which is a convex combination of matrices in $\mathcal{X}_1[1]$, establishing $\mathcal{B}_{\text{tr}} \subseteq \text{conv}\mathcal{X}_1[1]$. □

This characterization is both helpful in understanding the class, and serves a vital role in Section 6.2.2 for proving generalization error bounds on learning with low trace-norm matrices.

Maximum Margin Low Trace-Norm Matrix Factorization

To simplify presentation, we focus on binary labels, $Y \in \{\pm 1\}^{n \times m}$. We consider *hard-margin matrix factorization*, where we seek a minimum trace-norm matrix X that matches the observed labels with a margin of one:

$$\begin{aligned} &\text{minimize } \|X\|_{\text{tr}} \\ &\text{subject to } Y_{ia}X_{ia} \geq 1 \text{ for all } ia \in S \end{aligned} \tag{5.4}$$

We also consider *soft-margin* learning, where we minimize a trade-off between the trace-norm of X and its hinge-loss relative to Y_S:

$$\text{minimize } \|X\|_{\text{tr}} + c \sum_{ia \in S} \max(0, 1 - Y_{ia}X_{ia}). \tag{5.5}$$

As in large-margin linear discrimination, there is an inverse dependence between the norm and the margin. Fixing the margin and minimizing the trace-norm is equivalent to fixing the trace-norm and maximizing the margin. As in large-margin discrimination with certain infinite dimensional (e.g. radial) kernels, the data is always separable with sufficiently high trace-norm (a trace-norm of $\sqrt{n|S|}$ is sufficient to attain a margin of one).

5.3.2 Constraining the Factor Norms Uniformly: The Max-Norm

Instead of constraining the norms of rows in U and V on average (by considering their squared Frobenius norms), we can also constrain all rows of U and V to have small L_2 norm.

The Max-Norm

We refer to matrices with a factorization $X = UV'$, where all rows of U and V have bounded L_2 norm, as low *max-norm* matrices:

Definition 3. *The max-norm* $\|X\|_{\max}$ *is given by:*

$$\|X\|_{\max} \doteq \min_{\substack{U,V \\ X=UV'}} \left(\max_i |U_i|\right) \left(\max_a |V_a|\right)$$

where U_i *and* V_a *are the row-vectors of* U *and* V*.*

The max-norm is also known as the γ_2-norm [43].

Note that if we would have bounded the norm of the column vectors of U and V we would have defined the spectral norm. Unlike the spectral norm (largest singular value), Frobenius norm (euclidean norm of singular values) and trace-norm (sum of singular values), the max-norm is not a function of the singular values. In fact, calculating the max-norm requires using quadratic programming in order to solve the optimization problem in the definition of the max-norm.

By considering the first characterization in Lemma 8 we can establish the connection between the max-norm and the trace-norm:

Lemma 12. *For any* $X \in \mathbb{R}^{n\times m}$, $\|X\|_{\max} \leq \|X\|_{\text{tr}} \leq \sqrt{nm}\,\|X\|_{\max}$.

Although it is easy to verify that the max-norm is also convex, and so its unit ball:

$$\mathcal{B}_{\max} \doteq \{X \mid \|X\|_{\max} \leq 1\} \tag{5.6}$$

is convex, we cannot provide an exact characterization of the unit ball in terms of a simple class of matrices, as we did for the trace-norm in Lemma 11. We can, however, provide an approximate characterization in terms of the class of rank-one sign matrices:

$$\mathcal{X}_{\pm} \doteq \{uv' \,|u \in \{\pm 1\}^n, v \in \{\pm 1\}^m\} \tag{5.7}$$

Lemma 13.

$$\text{conv}\mathcal{X}_{\pm} \subseteq \mathcal{B}_{\max} \subseteq K_G \text{conv}\mathcal{X}_{\pm}$$

where K_G is Grothendiek's constant, and:

$$1.67 \leq K_G \leq 1.79$$

Proof. The left inclusion is immediate from the convexity of the max-norm and the fact that the factorization uv', with u and v sign matrices, establishes $\|uv'\|_{\max} \leq 1$. The right inclusion is consequence of Grothendiek's inequality (see Appendix F). □

Maximum Margin Low Max-Norm Matrix Factorization

Parallel to the low trace-norm problems (5.4)-(5.5), we also consider the hard-margin max-norm minimization problem:

$$\begin{aligned} &\text{minimize } \|X\|_{\max} \\ &\text{subject to } Y_{ia}X_{ia} \geq 1 \text{ for all } ia \in S \end{aligned} \tag{5.8}$$

and the soft-margin max-norm minimize problem:

$$\text{minimize } \|X\|_{\text{tr}} + c \sum_{ia \in S} \max(0, 1 - Y_{ia}X_{ia}). \tag{5.9}$$

As with the trace-norm counterpart, the problem is always separable.

A Geometric Interpretation

Low max-norm learning has a clean geometric interpretation. First, note that predicting the target matrix with the signs of a rank-k matrix corresponds to mapping the "items" (columns) to points in $\mathbb{R}^k$, and the "users" (rows) to homogeneous hyperplanes, such that each user's hyperplane separates his positive items from his negative items. Hard-margin low-max-norm prediction corresponds to mapping the users and items to points and hyperplanes in a high-dimensional unit sphere such that each user's hyperplane separates his positive and negative items with a large-margin (the margin being the inverse of the max-norm).

5.4 Learning Large-Margin Matrix Factorizations

In this section we investigate the optimization problems (5.4),(5.5),(5.8) and (5.9) of learning with low trace-norm and max-norm matrices. We show how these optimization problems can be written as a semi-definite programs.

5.4.1 Trace-Norm Minimization as Semi Definite Programming

Bounding the trace-norm of UV' by $\frac{1}{2}(\|U\|_{\text{Fro}}^2 + \|V\|_{\text{Fro}}^2)$, we can characterize the trace-norm in terms of the trace of a positive semi-definite matrix:

Lemma 14 ([26, Lemma 1]). *For any $X \in \mathbb{R}^{n\times m}$ and $t \in \mathbb{R}$: $\|X\|_{\text{tr}} \leq t$ iff there exists $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{m\times m}$ such that $\left[\begin{smallmatrix} A & X \\ X' & B \end{smallmatrix}\right] \succcurlyeq 0$ and $\operatorname{tr} A + \operatorname{tr} B \leq 2t$.*

Proof. Note that for any matrix W, $\|W\|_{\text{Fro}} = \operatorname{tr} WW'$. If $\left[\begin{smallmatrix} A & X \\ X' & B \end{smallmatrix}\right]$ is p.s.d. with $\operatorname{tr} A + \operatorname{tr} B \leq 2t$, we can write it as a product $\left[\begin{smallmatrix} U \\ V \end{smallmatrix}\right]\left[\begin{smallmatrix} U' & V' \end{smallmatrix}\right]$. We have $X = UV'$ and $\frac{1}{2}(\|U\|_{\text{Fro}}^2 + \|V\|_{\text{Fro}}^2) = \frac{1}{2}(\operatorname{tr} A + \operatorname{tr} B) \leq t$, establishing $\|X\|_{\text{tr}} \leq t$. Conversely, if $\|X\|_{\text{tr}} \leq t$ we can write it as $X = UV'$ with $\operatorname{tr} UU' + \operatorname{tr} VV' \leq 2t$ and consider the p.s.d. matrix $\left[\begin{smallmatrix} UU' & X \\ X' & VV' \end{smallmatrix}\right]$. □

Lemma 14 can be used in order to formulate minimizing the trace-norm as a semi-definite optimization problem (SDP).

The hard-margin matrix factorization problem (5.4) can be written as:

$$\min \frac{1}{2}(\operatorname{tr} A + \operatorname{tr} B) \quad \textbf{s.t.} \quad \begin{array}{c} \begin{bmatrix} A & X \\ X' & B \end{bmatrix} \succcurlyeq 0 \\ Y_{ia}X_{ia} \geq 1 \quad \forall ia \in S \end{array} \tag{5.10}$$

And introducing slack, soft-margin matrix factorization (5.5), can be written as:

$$\min \frac{1}{2}(\operatorname{tr} A + \operatorname{tr} B) + c\sum_{ia\in S} \xi_{ia} \quad \textbf{s.t.} \quad \begin{array}{c} \begin{bmatrix} A & X \\ X' & B \end{bmatrix} \succcurlyeq 0 \\ Y_{ia}X_{ia} \geq 1 - \xi_{ia} \\ \xi_{ia} \geq 0 \end{array} \quad \forall ia \in S \tag{5.11}$$

5.4.2 The Dual Problem

Both to aid in optimization, and to better understand the problems, we study the dual of the above problems.

Introducing the Lagrange multipliers $\Gamma, \Delta, \Upsilon, Q_{ia}$ we can write the optimization problem (5.11) as (we arbitrarily choose a scaling of $\frac{1}{2}$ for the multiplier $\left[\begin{smallmatrix}\Gamma & \Upsilon\\ \Upsilon' & \Delta\end{smallmatrix}\right]$):

$$\min_{\substack{A,B,X\\ \xi_{ia}\geq 0\ \forall ia\in S}} \ \max_{\substack{\left[\begin{smallmatrix}\Gamma & \Upsilon\\ \Upsilon' & \Delta\end{smallmatrix}\right]\succcurlyeq 0\\ Q_{ia}\geq 0\ \forall ia\in S}} \frac{1}{2}(\operatorname{tr} A + \operatorname{tr} B) + c\sum_{ia\in S}\xi_{ia}$$
$$-\frac{1}{2}\begin{bmatrix}\Gamma & \Upsilon\\ \Upsilon' & \Delta\end{bmatrix}\bullet\begin{bmatrix}A & X\\ X' & B\end{bmatrix} - \sum_{ia\in S} Q_{ia}(Y_{ia}X_{ia} + \xi_{ia} - 1)$$

The existence of an interior feasible solution guarantees there is no duality gap and allows us to change the order of minimization and maximization without a duality gap:

$$= \max_{\substack{\left[\begin{smallmatrix}\Gamma & \Upsilon\\ \Upsilon' & \Delta\end{smallmatrix}\right]\succcurlyeq 0\\ Q_{ia}\geq 0\ \forall ia\in S}} \ \min_{\substack{A,B,X\\ \xi_{ia}\geq 0\ \forall ia\in S}} \frac{1}{2}(\operatorname{tr} A + \operatorname{tr} B) + c\sum_{ia\in S}\xi_{ia}$$
$$-\frac{1}{2}\begin{bmatrix}\Gamma & \Upsilon\\ \Upsilon' & \Delta\end{bmatrix}\bullet\begin{bmatrix}A & X\\ X' & B\end{bmatrix} - \sum_{ia\in S} Q_{ia}(Y_{ia}X_{ia} + \xi_{ia} - 1) \quad (5.12)$$

Treating Q and Y as sparse matrices (with zero where $ia \notin S$), and collecting terms of primal variables, we can write:

$$= \max_{\substack{\left[\begin{smallmatrix}\Gamma & \Upsilon\\ \Upsilon' & \Delta\end{smallmatrix}\right]\succcurlyeq 0\\ Q_{ia}\geq 0\ \forall ia\in S}} \ \min_{\substack{A,B,X\\ \xi_{ia}\geq 0\ \forall ia\in S}} \sum_{ia\in S} Q_{ia} + \sum_{ia\in S}(c - Q_{ia})\xi_{ia} + \frac{1}{2}(I - \Gamma)\bullet A$$
$$+\frac{1}{2}(I - \Delta)\bullet B - \frac{1}{2}(\Upsilon + q\otimes y)\cdot X \quad (5.13)$$

In order for the minimization to be finite, we must have $c - Q_{ia} > 0$ (since ξ_{ia} is constrained to be positive) as well as $\Gamma = I$, $\Delta = I$ and $\Upsilon = -Q\otimes Y$ (since A,B and X and unconstrained). We can therefore write the dual to (5.11) as:

$$\max \sum_{ia \in S} Q_{ia} \quad \textbf{s.t.} \quad \begin{bmatrix} I & (-Q \otimes Y) \\ (-Q \otimes Y)' & I \end{bmatrix} \succcurlyeq 0, \quad 0 \leq Q_{ia} \leq c \tag{5.14}$$

where Q_{ia} is a dual variable associated with each $ia \in S$ and $Q \otimes Y$ denotes the sparse matrix $(Q \otimes Y)_{ia} = Q_{ia} Y_{ia}$ for $ia \in S$ and zeros elsewhere. The dual of the hard-margin problem is similar, but without the box constraints $Q_{ia} \leq c$.

In either case, problem is strictly feasible, and there is no duality gap.

The p.s.d. constraint in the dual (5.14) is equivalent to bounding the spectral norm of $Q \otimes Y$, and the dual can also be written as an optimization problem subject to a bound on the spectral norm, i.e. a bound on the singular values of $Q \otimes Y$:

$$\max \sum_{ia \in S} Q_{ia} \quad \textbf{s.t.} \quad \begin{matrix} \|Q \otimes Y\|_2 \leq 1 \\ 0 \leq Q_{ia} \leq c \quad \forall ia \in S \end{matrix} \tag{5.15}$$

In typical collaborative prediction problems, we observe only a small fraction of the entries in a large target matrix. Such a situation translates to a sparse dual semi-definite program, with the number of variables equal to the number of observed entries. Large-scale SDP solvers can take advantage of such sparsity.

5.4.3 Recovering the Primal Optimal from the Dual Optimal

Most SDP solvers use internal point methods and return a pair of primal and dual optimal solutions. The prediction matrix X^* minimizing (5.5) is part of the primal optimal solution of (5.11), and can be extracted from it directly.

Nevertheless, it is interesting to study how the optimal prediction matrix X^* can be directly recovered from a dual optimal solution Q^* alone. Although unnecessary when relying on standard internal point SDP solvers, this might enable us to use specialized optimization methods, taking advantage of the simple structure of the dual.

As for linear programming, recovering a primal optimal solution directly from a dual optimal solution is not always possible for SDPs in general. However, at least for the hard-margin problem (5.10) this is possible, and we describe below how an optimal prediction matrix X^* can be recovered from a dual optimal solution Q^* by calculating a singular value decomposition and solving linear equations.

Complimentary Slackness Considerations

Consider the hard-margin trace-norm minimization problem (5.10) and its dual, and let (X^*, A^*, B^*) be primal optimal and Q^* be dual optimal solutions.

Strong complimentary slackness for the SDPs guarantees that not only the matrix inner product $\begin{bmatrix} I & (-Q^*\otimes Y) \\ (-Q^*\otimes Y)' & I \end{bmatrix} \bullet \begin{bmatrix} A^* & X^* \\ X^{*\prime} & B^* \end{bmatrix}$ is zero for any primal and dual optimal solutions, but in fact the matrix product

$$\begin{bmatrix} I & (-Q^*\otimes Y) \\ (-Q^*\otimes Y)' & I \end{bmatrix} \begin{bmatrix} A^* & X^* \\ X^{*\prime} & B^* \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{5.16}$$

is zero everywhere for the optimal solutions. The blocks of this matrix equality yield the following necessary condition for dual and primal optimality:

$$\begin{aligned} A^* &= (Q^*\otimes Y)X^{*\prime} & X^* &= (Q^*\otimes Y)B^* \\ X^{*\prime} &= (Q^*\otimes Y)'A^* & B^* &= (Q^*\otimes Y)'X^* \end{aligned} \tag{5.17}$$

Together with complimentary slackness of the label constraints, this condition is also sufficient. Recalling that $\begin{bmatrix} A^* & X^* \\ X^{*\prime} & B^* \end{bmatrix} \succcurlyeq 0$, we can write $\begin{bmatrix} A^* & X^* \\ X^{*\prime} & B^* \end{bmatrix} = \begin{bmatrix} \tilde{U} \\ \tilde{V} \end{bmatrix} \begin{bmatrix} \tilde{U}' & \tilde{V}' \end{bmatrix}$ with $\begin{bmatrix} \tilde{U}' & \tilde{V}' \end{bmatrix}$ of full row rank. Expressing $A^* = \tilde{U}\tilde{U}'$, $B^* = \tilde{V}\tilde{V}'$ and $X^* = \tilde{U}\tilde{V}'$, we can now rewrite (5.17) as:

$$\begin{aligned} \tilde{U}\tilde{U}' &= (Q^*\otimes Y)\tilde{V}\tilde{U}' & \tilde{U}\tilde{V}' &= (Q^*\otimes Y)\tilde{V}\tilde{V}' \\ \tilde{V}\tilde{U}' &= (Q^*\otimes Y)'\tilde{U}\tilde{U}' & \tilde{V}\tilde{V}' &= (Q^*\otimes Y)'\tilde{U}\tilde{V}' \end{aligned} \tag{5.18}$$

Combining each row of matrix equations into a single matrix equation yields:

$$(\tilde{U} - (Q^*\otimes Y)\tilde{V})\begin{bmatrix} \tilde{U}' & \tilde{V}' \end{bmatrix} \qquad (\tilde{V} - (Q^*\otimes Y)'\tilde{U})\begin{bmatrix} \tilde{U}' & \tilde{V}' \end{bmatrix} \tag{5.19}$$

Since $\left[\tilde{U}' \; \tilde{V}'\right]$ is of full row rank, we have:

$$\tilde{U} = (Q^* \otimes Y)\tilde{V} \qquad\qquad \tilde{V} = (Q^* \otimes Y)'\tilde{U} \tag{5.20}$$

Together these equations specify that the columns of $\tilde{U}$ and $\tilde{V}$ are corresponding eigenvectors of eigenvalue one of $(Q^* \otimes Y)(Q^* \otimes Y)'$ and $(Q^* \otimes Y)'(Q^* \otimes Y)$. Or in other words, $\tilde{U}$ and $\tilde{V}$ are correspondingly spanned by the singular rows and vectors of $(Q^* \otimes Y)$ with singular value one.

Note that since $\left[\begin{smallmatrix} I & (-Q^*\otimes Y) \\ (-Q^*\otimes Y)' & I \end{smallmatrix}\right] \succcurlyeq 0$, the singular values of $(Q^* \otimes Y)$ are all less than or equal to one, and those that are equal to one correspond to zero eigenvectors of $\left[\begin{smallmatrix} I & (-Q^*\otimes Y)) \\ (-Q^*\otimes Y)' & I \end{smallmatrix}\right]$.

The singular rows and columns of $(Q^* \otimes Y)$ corresponding to singular value one define a linear space in which $\tilde{U}$ and $\tilde{V}$ lie. The optimal $\tilde{U}$ and $\tilde{V}$ from this space can be recovered by solving the linear equations

$$X_{ia} = Y_{ia} \quad \forall S_{ia} > 0. \tag{5.21}$$

Recovering X^* from Q^*

To summarize, we obtain the following procedure for recovering an optimal solution of (5.4) from an optimal solution of (5.15) (with no slack, i.e. no box constraints):

1. Let Q^* be a dual optimal solution.

2. Calculate the singular value decomposition $Q^* \otimes Y = U\Lambda V'$.

3. Let $\tilde{U} \in \mathbb{R}^{n \times p}$ and $\tilde{V} \in \mathbb{R}^{m \times p}$ be the columns of U and V with singular value exactly one.

4. For every $Q^*_{ia} > 0$, consider the equation $X^*_{ia} = \tilde{U}_i RR' \tilde{V}'_a = Y_{ia}$ and solve these as linear equations in the entries of RR'.

5. $X^* = \tilde{U}RR'\tilde{V}$ is an optimal solution of (5.4).

5.4.4 Using the Dual: Recovering Specific Entries of X^*

The approach described above requires solving a large system of linear equations (with as many variables as observations). Furthermore, especially when the observations are very sparse (only a small fraction of the entries in the target matrix are observed), the dual solution is much more compact then the prediction matrix: the dual involves a single number for each *observed* entry. It might be desirable to avoid storing the prediction matrix X^* explicitly, and calculate a desired entry $X^*_{i_0a_0}$, or at least its sign, directly from the dual optimal solution Q^*.

Consider adding the constraint $X_{i_0a_0} > 0$ to the primal SDP (5.11). If there exists an optimal solution X^* to the original SDP with $X^*_{i_0a_0} > 0$, then this is also an optimal solution to the modified SDP, with the same objective value. Otherwise, the optimal solution of the modified SDP is not optimal for the original SDP, and the optimal value of the modified SDP is higher (worse) than the optimal value of the original SDP.

Introducing the constraint $X_{i_0a_0} > 0$ to the primal SDP (5.11) corresponds to introducing a new variable $Q_{i_0a_0}$ to the dual SDP (5.14), appearing in $Q \otimes Y$ (with $Y_{i_0a_0} = 1$) but *not* in the objective. In this modified dual, the optimal solution Q^* of the original dual would always be feasible. But, if $X^*_{i_0a_0} \leq 0$ in all primal optimal solutions, and the modified primal SDP has a higher value, then so does the dual, and Q^* is no longer optimal for the new dual. By checking the optimality of Q^* for the modified dual, e.g. by attempting to re-optimize it, we can recover the sign of $X^*_{i_0a_0}$.

We can repeat this test once with $Y_{i_0a_0} = 1$ and once with $Y_{i_0a_0} = -1$, corresponding to $X_{i_0a_0} < 0$. If $Y_{i_0a_0} X^*_{i_0a_0} < 0$ (in all optimal solutions), then the dual solution can be improved by introducing $Q_{i_0a_0}$ with a sign of $Y_{i_0a_0}$.

5.4.5 Max-Norm Minimization as a Semi-Definite Program

The max-norm can also be characterized with a similar positive semi-definite matrix. However, if before we were interested in summing the norms of the rows of U and V, now we are interested in bounding them:

Lemma 15. *For any* $X \in \mathbb{R}^{n \times m}$ *and* $t \in \mathbb{R}$: $\|X\|_{\max} \leq t$ *if and only if there exists*

$A \in \mathbb{R}^{n \times n}$ and $B \in \mathbb{R}^{m \times m}$ such that $\begin{bmatrix} A & X \\ X' & B \end{bmatrix}$ is positive semi-definite with diagonal elements at most t (i.e. $A_{ii} \leq t$ and $B_{aa} \leq t$ for all i, a).

Similarly to (5.11) we have the following SDP for the soft-margin max-norm minimization problem (5.9):

$$\min t + c \sum_{ia \in S} \xi_{ia} \quad \textbf{s.t.} \quad \begin{bmatrix} A & X \\ X' & B \end{bmatrix} \succcurlyeq 0 \quad \begin{array}{c} A_i i \leq t \quad \forall i \\ B_a a \leq t \quad \forall a \\ y_{ia} X_{ia} \geq 1 - \xi_{ia} \quad \forall ia \in S \\ \xi_{ia} \geq 0 \forall ia \in S \end{array} \tag{5.22}$$

And the corresponding dual:

$$\max \sum_{ia \in S} q_{ia} \quad \textbf{s.t.} \quad \begin{bmatrix} \Gamma & (-Q \otimes Y) \\ (-Q \otimes Y)' & \Delta \end{bmatrix} \succcurlyeq 0 \quad \begin{array}{c} \Gamma, \Delta \text{ are diagonal} \\ \operatorname{tr} \Gamma + \operatorname{tr} \Delta = 1 \\ 0 \leq q_{ia} \leq c \quad \forall ia \in S \end{array} \tag{5.23}$$

where again the dual variables are Q_{ia} for each $ia \in S$ and $Q \otimes Y$ denotes the sparse matrix $(Q \otimes Y)_{ia} = Q_{ia} Y_{ia}$ for $ia \in S$ and zeros elsewhere.

5.4.6 Predictions for new users

So far, we have assumed that learning is done on the known entries in all rows. It is commonly desirable to predict entries in a new partially observed row of Y (a new user in a collaborative filtering task), not included in the original training set. This essentially requires solving a "conditional" problem, where V is already known, and a new row of U is learned (the predictor for the new user) based on a new partially observed row of X. Using large-margin matrix factorization, this is a standard SVM problem.

5.5 Loss Functions for Ratings

In our discussion of matrix completion so far, we focused on binary classification target matrices, where target values take only two possible values. In Chapter 3 we also discussed various loss functions appropriate for real-valued observations, and in particular the ubiquitous sum-squared loss. But for many collaborative filtering problems, the user preferences are specified as discrete ratings, or levels, in some integer range, e.g. one to five "stars".

This type of labels falls between real-valued labels and multi-class labels. Although discrete like multi-class labels, they have a particular ordering, like real-valued labels. In order to apply the suggested matrix completion methods to such data, we need to choose an appropriate loss function.

Rating data has been considered both in the statistical literature [54, 27] and in the machine learning literature [64, 23, 20, 36]. Some approaches extract from the rating levels binary comparison relationships on the rated items and thus map the problem to a partial ordering problem [36]. Here we focus on approaches that use real-valued predictors x, and assign a loss $\text{loss}(x; y)$ relative to each rating level y. We are particularly interested in loss functions imposing a margin between predictors corresponding to different levels.

5.5.1 Threshold-based loss functions

Several loss functions, which are generalizations of standard loss functions for binary classification, have recently been suggested in the machine learning literature. Most of these are based on separating the real line to (possibly infinite) intervals corresponding to the rating levels.

For binary classification, a single threshold (zero) separates between positive and negative predictions. For R rating levels, $R - 1$ threshold $b_1 < \cdots < b_{R-1}$ are necessary. We will also denote $b_0 = -\infty$ and $b_R = \infty$ for convenience. A predictor x then corresponds to the rating level $r(x)$ such that $b_{r(x)-1} \leq x < b_{r(x)}$.

The simple zero-one loss ($\text{loss}_{01}(x) = 0$ if $r(x) = y$ and zero otherwise) does not reflect how far away the correct rating is from the predicted rating. A simple alternative is the absolute rank difference loss

$$\text{loss}_{ard}(x; y) = |r(x) - y|.$$

This loss might correspond well to the actual cost of rating errors. In fact, we might want to measure the generalization error (over the entire matrix) in terms of this loss. However, it is not convex, and does not impose a separation margin.

Shashua and Levin[64] suggested imposing a hinge loss on each end of the interval corresponding to y. That is:

$$\text{loss}(x; y) = \max(0, b_{y-1} + 1 - x) + \max(0, x - b_y - 1).$$

Other standard convex loss functions $l(x)$, such as logistic or exponential loss, can be used instead of the hinge loss in a similar way, with the general form:

$$\text{loss}(x; y) = l(x - b_{y-1}) + l(b_y - x),$$

where $l(\infty) = 0$.

However, such loss functions do not bound the absolute rank difference, which might be desirable if our true objective is minimizing the overall absolute rank difference [10]. Although they penalize predictions which are far away on the real line from the "target segment", they do not take into account that variations in some regions of the real line (corresponding to large target segments) are not very expensive, while variations in other regions, across densely concentrated threshold, are more expensive.

An alternative is to superimpose the loss functions associated with all thresholds, not only those bounding the segment corresponding to y:

$$\text{loss}(x; y) = \sum_{r=1}^{y-1} l(x - b_r) + \sum_{r=y}^{R-1} l(b_r - x).$$

This loss function might lead to a slightly more complicated objective. In particular, for the hinge it corresponds to more constraints in the learning optimization problems. However, it might reduce the generalization error, particularly with respect to the absolute rank difference, and perhaps also yield better generalization error bounds.

In either case, the thresholds need to be determined. Although it is possible to fix the thresholds in advance, significant flexibility can be gained by fitting the threshold from the data [64, 20]. In the context of matrix factorization, it is possible to fit one set of thresholds for the entire matrix, or fit separate thresholds for each row, or for each column. This can allow us, for example, to account for user variations in the use of different ratings, as an alternative to explicit normalization.

5.6 Implementation and Experiments

All the methods described in this Chapter were implemented in MATLAB, optionally using YALMIP [49].

We conducted preliminary experiments on a subset of the MovieLens dataset[1], consisting of the 100 users and 100 movies with the most ratings. The ratings are on a discrete scale of one through five, and we experimented with both generalizations of the hinge loss described above, allowing per-user thresholds. We used CSDP [16] to solve the resulting SDPs. Solving with the immediate-threshold loss took about 25 CPU minutes on a 3.06GHz Intel Xeon. Solving with the all-threshold loss took up to eight hours. We compared against methods described in [51], randomly selecting 50% of the entries for training and 50% for testing. We tested a range of regularization parameters (C/K) and present the best zero-one agreement error (ZOE) and mean-absolute-error (MAE) result for each method, with the regularization parameters attaining it.

	all-θ	immediate-θ	K-medians	WLRA	WLRA
	LMMF,c=0.2	LMMF,c=0.3	K=2	K=1	K=2
MAE	0.508 / **0.670**	0.621 / 0.715	0.620 / 0.674	0.679 / 0.698	0.622 / 0.714
ZOE	0.450 / 0.553	0.462 / **0.542**	0.510 / 0.558	0.550 / 0.559	0.519 / 0.553

Table 5.1: Lowest train/test errors for various methods.

[1] `http://www.cs.umn.edu/Research/GroupLens/`

5.7 Discussion

Learning a large-margin matrix factorization requires solving a sparse semi-definite program. We experimented with generic SDP solvers, and were able to learn with up to tens of thousands of labels. We propose that just as generic QP solvers do not perform well on SVM problems, special purpose techniques, taking advantage of the very simple structure of the dual (5.14), might be necessary in order to solve large-scale large-margin matrix factorization problems. An itterative update procedure for the dual would not only allow us to find the optimal dual, but also extract entries in the primal optimal solution from the dual optimal, using the methods of Section 5.4.4.

SDPs were recently suggested for a related, but different, problem: learning the features (or equivalently, kernel) that are best for a *single* prediction task [45]. This task is hopeless if the features are completely unconstrained, as they are in our formulation. Lanckriet *et al* suggest constraining the allowed features, e.g. to a linear combination of a few "base feature spaces" (or base kernels), which represent the external information necessary to solve a single prediction problem. It is possible to combine the two approaches, seeking constrained features for multiple related prediction problems, as a way of combining external information (e.g. details of users and of items) and collaborative information.

An alternate method for introducing external information into our formulation is by adding to U and/or V additional fixed (non-learned) columns representing the external features. This method degenerates to standard SVM learning when Y is a vector rather than a matrix.

Chapter 6

PAC-type Bounds for Matrix Completion

A central type of results in machines learning are probabilistic post-hoc bounds on the generalization error of predictors. The classic setting for such result is learning a classification based on a random supervised training set. An important aspect is that no assumptions are made about the source of examples, other than the central assumption that all examples are drawn i.i.d. from the same unknown source distribution. PAC (Probably Approximately Correct) bounds then assure us that regardless of the source distribution, with certain probability over the random training set, the expected error over future samples from the same source will not be much more than the average error over the training set. Although such bounds do not provide for an a-priori guarantee on the performance of the predictor, and such an a-priori guarantee cannot be expected without assumptions on the source distribution, they do provide a post-hoc guarantee in terms of an observed quantity—the training error. The relationship between the probability of failure, the degree of approximation and the sample size is generally governed by the complexity of the class from which the predictor is chosen.

Similar types of bounds can be shown on the generalization error of matrix completion via matrix factorization. The major assumption made, paralleling the i.i.d. source assumption, is that entries in the target matrix to be observed are chosen randomly. The bounds will then be stated with high probability over the choice of the random subset of observed

arbitrary source distribution	⇔	target matrix Y
random training set	⇔	random set S of observed entries
hypothesis	⇔	predicted matrix X
training error	⇔	observed discrepancy $\mathcal{D}_S(X;Y)$
generalization error	⇔	true discrepancy $\mathcal{D}(X;Y)$

Figure 6-1: Correspondence with post-hoc bounds on the generalization error for standard prediction tasks

entries. With this probability, we will bound the overall discrepancy between the entire predicted matrix X and the target Y as a function of the discrepancy on the observed entries. The bounds will hold for *any* target matrix Y.

More formally, bounds of the following type will be shown:

$$\forall_{Y \in R^{n \times d}} \Pr_S \left(\forall_{X \in \mathcal{X}} \mathcal{D}(X;Y) < \mathcal{D}_S(X;Y) + \epsilon(n, d, |S|, \mathcal{X}, \delta) \right) > 1 - \delta$$

where the distribution on S is uniform over all subsets of $|S|$ entries, $\mathcal{X}$ is a class of matrices,

$$\mathcal{D}(X;Y) = \frac{1}{nm} \sum_{ia} \text{loss}(X_{ia}; Y_{ia})$$

is the average discrepancy over the entire prediction matrix and

$$\mathcal{D}_S(X;Y) = \frac{1}{|S|} \sum_{ia} \text{loss}(X_{ia}; Y_{ia})$$

is the average observed discrepancy. Such results ensure that the bound on the overall prediction error hold also for the specific matrix in the class $\mathcal{X}$ chosen by the learning algorithm.

6.1 Bounds for Low-Rank Matrix Factorization

In this section, we consider generalization error bounds for the class of rank-k matrices, $\mathcal{X}_k = \{X \,|\, \operatorname{rank} X = k\}$. The allowed rank k is a complexity parameter that will determine the relationship between the sample size $|S|$ and the error ϵ.

6.1.1 Prior Work

Previous results bounding the error of collaborative prediction using a low-rank matrix all assume the true target matrix Y is well-approximated by a low-rank matrix. This corresponds to a large *eigengap* between the top few singular values of Y and the remaining singular values. Azar *et al* [8] gives asymptotic results on the convergence of the predictions to the true preferences, assuming they have an eigengap. Drineas *et al* [25] analyzes the sample complexity needed to be able to predict a matrix with an eigengap, and suggests strategies for actively querying entries in the target matrix. To our knowledge, this is the first analysis of the generalization error of low-rank methods that do not make any assumptions on the true target matrix.

Generalization error bounds (and related online learning bounds) were previously discussed for collaborative prediction applications, but only when prediction was done for each user separately, using a feature-based method, with the other user's preferences as features [20, 21]. Although these address a collaborative prediction application, the learning setting is a standard feature-based setting. These methods are also limited, in that learning must be performed separately for each user.

Shaw-Taylor *et al* [65] discuss assumption-free post-hoc bounds on the residual errors of low-rank approximation. These results apply to a different setting, where a subset of the rows are fully observed, and bound a different quantity—the distance between rows and the learned *subspace*, rather then the distance to predicted entries.

6.1.2 Bound on the Zero-One Error

We begin by considering binary labels $Y_{ia} \in \pm$ and a zero-one sign agreement loss:

$$\text{loss}(X_{ia}; Y_{ia}) = 1_{Y_{ia}X_{ia} \leq 0} \tag{6.1}$$

Theorem 16. *For* any *matrix* $Y \in \{\pm 1\}^{n \times m}$, $n, m > 2$, $\delta > 0$ *and integer* k, *with probability at least* $1 - \delta$ *over choosing a subset* S *of entries in* Y *uniformly among all subsets* *of* $|S|$ *entries:*

$$\forall_{X,\operatorname{rank} X<k}\mathcal{D}(X;Y) < \mathcal{D}_S(X;Y) + \sqrt{\frac{k(n+m)\log\frac{8em}{k} - \log\delta}{2|S|}}$$

where the discrepancies are with respect to the zero-one loss (6.1). *The logarithms are base two, and* e *is the natural base.*

To prove the theorem we employ standard arguments about the generalization error for finite hypothesis classes with bounded cardinality (e.g. [24, Theorem 8.3]).

To prove the theorem, first fix Y as well as $X \in \mathbb{R}^{n\times m}$. When an index pair (i, a) is chosen uniformly at random, $\operatorname{loss}(X_{ia}; Y_{ia})$ is a Bernoulli random variable with probability $\mathcal{D}(X;Y)$ of being one. If the entries of S are chosen independently and uniformly, $|S|\mathcal{D}_S(X;Y)$ is Binomially distributed with mean $|S|\mathcal{D}(X;Y)$ and using Chernoff's inequality:

$$\Pr_S\left(\mathcal{D}(X;Y) \geq \mathcal{D}_S(X;Y) + \epsilon\right) \leq e^{-2|S|\epsilon^2} \tag{6.2}$$

The distribution of S in Theorem 16 is slightly different, as S is chosen without repetitions. The mean of $\mathcal{D}_S(X;Y)$ is the same, but it is more concentrated, and (6.2) still holds.

Now consider all rank-k matrices. Noting that $\operatorname{loss}(X_{ia}; Y_{ia})$ depends only on the *sign* of X_{ia}, it is enough to consider the equivalence classes of matrices with the same sign patterns. Let $f(n, m, k)$ be the number of such equivalence classes, i.e. the number of possible sign configurations of $n \times m$ matrices of rank at most k:

$$F(n,m,k) = \{\operatorname{sign} X \in \{-,0,+\}^{n\times m} | X \in \mathbb{R}^{n\times m}, \operatorname{rank} X \leq k\}$$

$$f(n,m,k) = \sharp F(n,m,k)$$

where sign X denotes the element-wise sign matrix $(\operatorname{sign} X)_{ia} = \begin{cases} 1 & \text{If } X_{ia} > 0 \\ 0 & \text{If } X_{ia} = 0\,. \\ -1 & \text{If } X_{ia} < 1 \end{cases}$

For all matrices in an equivalence class, the random variable $\mathcal{D}_S(X;Y)$ is the same, and taking a union bound of the events $\mathcal{D}(X;Y) \geq \mathcal{D}_S(X;Y) + \epsilon$ for each of these $f(n, m, k)$

random variables we have:

$$\Pr_S\left(\exists_{X,\operatorname{rank} X\leq k}\mathcal{D}(X;Y)\geq\mathcal{D}_S(X;Y)+\sqrt{\frac{\log f(n,m,k)-\log\delta}{2|S|}}\right)\leq\delta \qquad (6.3)$$

by using (6.2) and setting $\epsilon=\sqrt{\frac{\log f(n,m,k)-\log\delta}{2|S|}}$. The proof of Theorem 16 rests on bounding $f(n,m,k)$, which we will do in the next section.

Note that since the equivalence classes we defined do not depend on the sample set, no symmetrization argument is necessary. One might suggest improving the bound using more specific equivalence classes, considering only the sign configurations of entries in S. However, not much can be gained from such refinements. Consider, for example, bounding the number of S-specific equivalence classes by $f(n,m,k,|S|)\leq|S|^V$ using VC-dimension arguments. Then we have $f(n,m,k)\leq(nm)^V$, and since for meaningful sample sizes $|S|\geq\max(n,m)$ (otherwise we cannot hope to generalize), the improvement in the bound is by at most a constant factor of two, which is lost in the symmetrization arguments. Bounding the growth function $f(n,m,k,|S|)$ directly might yield improvements for specific sample size, but since $f(n,m,k)\leq f(n,m,k,|S|)^{\log nm}$, the improvement would not be by more than a factor of $\log nm$.

6.1.3 Sign Configurations of a Low-Rank Matrix

In this section, we bound the number $f(n,m,k)$ of sign configurations of $n\times m$ rank-k matrices over the reals. We follow a course outlined by Alon [3].

Any matrix X of rank at most k can be written as a product $X=UV'$ where $U\in\mathbb{R}^{n\times k}$ and $V\in\mathbb{R}^{k\times m}$. In order to bound the number of sign configurations of X, we consider the $k(n+m)$ entries of U,V as variables, and the nm entries of X as polynomials of degree two over these variables:

$$X_{ia}=\sum_{\alpha=1}^{k}U_{i\alpha}V_{a\alpha}$$

Appendix A presents a bound on the number of sign configurations of polynomials of bounded degree. Applying Theorem 34 from the Appendix, we obtain:

Lemma 17. $f(n,m,k) \leq \left(\frac{8e\cdot 2\cdot nm}{k(n+m)}\right)^{k(n+m)} \leq (16em/k)^{k(n+m)}$

Using this bound in (6.3) would yield a factor of $\log \frac{16em}{k}$ in the bound. In considering sign configurations, we differentiate between zero entries and non-zero entries. However, for each entry, we only care about two possible states of X_{ia}: either it has the same sign as Y_{ia}, or it does not, in which case we do not care if it is zero or of opposite sign. For any fixed matrix Y it is therefore enough to consider the configuration of sign agreements with Y, which can by bounded using Theorem 35:

Lemma 18. *For any $Y \in \{+,-\}^{n\times m}$, the number of configurations of sign agreements of rank-k matrices with Y is bounded by*

$$\left(\frac{4e\cdot 2\cdot nm}{k(n+m)}\right)^{k(n+m)} \leq (8em/k)^{k(n+m)}$$

This establishes Theorem 16

Lower bound on the number of sign configurations These upper bounds on the number of low-rank matrices are tight up to multiplicative factors in the exponent.

Lemma 19. *For $m > k^2$, $f(n,m,k) \geq m^{\frac{1}{2}(k-1)n}$*

Proof. Recall that rank-k matrices are those that can be written as $X = UV'$ with an inner dimension of k. Fix any matrix $V \in \mathbb{R}^{m\times k}$ with rows in general position, and consider the number $f(n,V,k)$ of sign configurations of matrices UV', where U varies over all $n \times k$ matrices. Each row of sign UV' is a homogeneous linear classification of the rows of V, i.e. of m vectors in general position in $\mathbb{R}^k$. Focusing only on $+/-$ sign configurations (no zeros in UV'), there are exactly $\left(2\sum_{i=0}^{k-1}\binom{m}{i}\right)$ possible homogeneous linear classifications of m vectors in general position in $\mathbb{R}^k$, and so these many options for each row of sign UV'. We can therefore bound:

$$\begin{aligned} f(n,m,k) \geq f(n,V,k) &\geq \left(2\sum_{i=0}^{k-1}\binom{m}{i}\right)^n \\ &\geq \binom{m}{k-1}^n \geq \left(\frac{m}{k-1}\right)^{n(k-1)} \geq \sqrt{m}^{(k-1)n} = m^{\frac{1}{2}(k-1)n} \end{aligned}$$

□

Related work The number of sign configurations of low-rank matrices was previously considered in the context of unbounded error communication complexity.

Consider two parties, Alice and Bob, who would like to jointly calculate a function $X(i,a) : [n] \times [m] \to \pm 1$ where Alice holds the input i and Bob holds the input a. Alice and Bob would like to communicate as little as possible between them, so that at the end of the computation each one of them would hold the correct answer with probability greater than half (both Alice and Bob are unlimited computationally). The ***unbounded error communication complexity*** of a function A is the minimum number c such that there exists a probabilistic protocol, under which no more than c bits are exchanged between Alice and Bob for any input, and for any input, at the end of the computation, both Alice and Bob hold $A(i,a)$ with probability greater than half. Viewing A as an $n \times m$ matrix, its unbounded communication complexity is roughly the logarithm of its rank, and more precisely bounded by [57, Theorem 2]:

$$\lceil \log \operatorname{rank} A \rceil \leq c \leq \lceil \log \operatorname{rank} A \rceil + 1$$

In order to show the existence of functions with high unbounded error communication complexity, Alon, Frankl and Rodl [1] bound the number of sign configurations of low-rank matrices. They then use counting arguments to establish that some (in fact, most) binary matrices can only be realized by high-rank matrices, and therefore correspond to functions with high unbounded error communication complexity.

Alon Frankl and Rodl's bound is based on a two-step approach similar to a preliminary result independently obtained by the author of this thesis [68], and made obsolete by the stronger results discussed above. First, a fixed matrix U is considered, and a bound on the number of sign configurations of $X = UV'$ is obtained, where only V is variable. Each column of UV' is a linear separation of the rows of U. The number of linear separations of n points in $\mathbb{R}^k$ is less than $2(k+1)n^{k-1}$, and so the number of sign configurations for any fixed U is less than $\left(2(k+1)n^{k-1}\right)^m$. This bound should be multiplied by the number of different matrices U, i.e. the number of matrices U yielding different sets of possible sign configurations. The important aspect of U is the different ways its rows can be linearly separated, i.e. the set of covectors the rows of U define. And so, what we are after is the number of possible different sets of covectors realizable by n vectors in $\mathbb{R}^k$, i.e. the number of possible realizable oriented matroids [14]. There are at most $n^{k(k+1)n}$ oriented matroids realizable by n points in $\mathbb{R}^k$ [30, 2], yielding a bound of

$$f(n, m, k) \leq \left(2(k+1)n^{k-1}\right)^m n^{k(k+1)n} < 2^{km \log 2n + k(k+1)n \log n}. \tag{6.4}$$

The bound of Theorem 34 avoids the quadratic dependence on k in the exponent. Alon, Frankl and Rodl used a different bound on the number realizable oriented matroids, bounding it by $2^{n^3+O(n^2)}$, which is looser for small k, but slightly tighter then (6.4) when $k = \Theta(n)$.

6.1.4 Other Loss Functions

In Section 6.1.2 we considered generalization error bounds for a zero-one loss function. More commonly, though, other loss functions are used, and it is desirable to obtain generalization error bounds for general loss functions.

When dealing with other loss functions, the magnitude of the entries in the matrix are important, and not only their signs. It is therefore no longer enough to bound the number of sign configurations. Instead, we will bound not only the number of ways low rank matrices behave with regards to a threshold of zero, but the number of possible ways low-rank matrices can behave relative to any set of thresholds. That is, for any threshold matrix $T \in \mathbb{R}^{n \times m}$, we will show that the number of possible sign configurations of $(X - T)$, where X is low-rank, is small. Intuitively, this captures the complexity of the class of low-rank matrices not only around zero, but throughout all possible values.

We then use standard results from statistical machine learning to obtain generalization error bounds from the bound on the number of relative sign configurations. The number of relative sign configurations serves as a bound on the *pseudodimension*—the maximum number of entries for which there exists a set of thresholds such that all relative sign configurations (limited to these entries) is possible. The pseudodimension can in turn be used to show the existence of a small ϵ-net. Roughly speaking, and ϵ-net is a finite set of matrices (the number of which we will bound) such that every low-rank matrix has entries withing ϵ of some matrix in the ϵ-net. We can then use arguments similar to those in the proof of Theorem 16, taking a union bound only over the matrices in the ϵ-net and arguing that the error for any other matrix is within ϵ of one of these matrices. The ϵ-net we actually use is not an ϵ-net for the low-rank matrices themselves, but of the element-wise losses of these matrices relative to a fixed target matrix.

The Pseudodimension of Low-Rank Matrices

Recall the definition of the pseudodimension of a class of real-valued functions:

Definition 4. *A class $\mathcal{F}$ of real-valued functions* pseudo-shatters *the points $x_1, \dots, x_n$ with thresholds $t_1, \dots, t_n$ if for every binary labeling of the points $(s_1, \dots, s_n) \in \{+, -\}^n$ there exists $f \in \mathcal{F}$ s.t. $f(x_i) < t_i$ iff $s_i = -$. The* pseudodimension *of a class $\mathcal{F}$ is the supremum over n for which there exist n points and thresholds that can be shattered.*

The pseudodimension is a generalization of the VC-dimension, which is defined only for classes of indicator functions. The pseudodimension is also equal to the VC-dimension of the *subgraphs* of $\mathcal{F}$, that is, the class of the sets $\{(x, y) \mid f(x) < y\}$ for each $f \in \mathcal{F}$. Classes with finite pseudodimension are known as *VC subgraph* classes.

In order to apply known results linking the pseudodimension to covering numbers, we consider matrices $X \in \mathbb{R}^{n \times m}$ as real-valued functions $X : [n] \times [m] \to \mathbb{R}$ over index pairs to entries in the matrix. The class $\mathcal{X}_k$ of rank-k matrices can now be seen as a class of real-valued functions over the domain $[n] \times [m]$. We bound the pseudodimension of this class by bounding, for any threshold matrix $T \in \mathbb{R}^{n \times m}$ the number of *relative sign matrices*:

$$G_T(n, m, k) = \{\operatorname{sign}^{\pm}(X - T) \in \{-, +\}^{n \times m} | X \in \mathbb{R}^{n \times m}, \operatorname{rank} X \leq k\}$$
$$g_T(n, m, k) = \sharp G_T(n, m, k)$$

where $\operatorname{sign}^{\pm} X$ denotes the element-wise *binary* sign matrix $(\operatorname{sign} X)_{ia} = \begin{cases} 1 & \text{If } X_{ia} \geq 0 \\ -1 & \text{If } X_{ia} < 1 \end{cases}$, where zero is considered as positive, in accordance with our definition of shattering.

Lemma 20. *For any $T \in \mathbb{R}^{n \times m}$, we have $g_T(n, m, k) \leq \left(\frac{8em}{k}\right)^{k(n+m)}$.*

Proof. We take a similar approach to that of Lemmas 17 and 18. Any matrix X of rank at most k can be written as a product $X = UV'$ where $U \in \mathbb{R}^{n \times k}$ and $V \in \mathbb{R}^{k \times m}$. we consider the $k(n+m)$ entries of U, V as variables, and the nm entries of $X - T$ as polynomials of degree two over these variables:

$$(X - T)_{ia} = \sum_{\alpha=1}^{k} U_{i\alpha} V_{a\alpha} - T_{ia}$$

Applying Theorem 35 yields the desired bound. □

Corollary 21. *The pseudodimension of the class $\mathcal{X}_k$ of $n \times m$ matrices over the reals of rank at most k, is less than $k(n+m)\log\frac{8em}{k}$.*

A Generalization Error Bound

Viewing rank-k matrices as real-valued functions over index pairs, standard results in statistical machine learning provide us with a bound on the generalization error in terms of the pseudodimension. Substituting the bound on the pseudodimension from Corollary 21 in Theorem 44 we obtain:

Theorem 22. *For any monotone loss function with $|loss| \leq M$, any matrix $Y \in \{\pm 1\}^{n \times m}$, $n, m > 2$, $\delta > 0$ and integer k, with probability at least $1 - \delta$ over choosing a subset S of entries in Y uniformly among all subsets of $|S|$ entries:*

$$\forall_{X, \mathrm{rank}\, X < k} \mathcal{D}(X; Y) < \mathcal{D}_S(X; Y) + \epsilon$$

where:

$$\epsilon = 6\sqrt{\frac{k(n+m)\log\frac{8em}{k}\log\frac{M|S|}{k(n+m)} - \log\delta}{|S|}}$$

6.2 Bounds for Large-Margin Matrix Factorization

In this section, we consider generalization error bounds for Large-Margin Matrix Factorization. More specifically, we provide bounds that hold for *all* learned matrices X, but where the bound on the generalization error depends on either the trace-norm or the max-norm of X. Since the norms are a scale sensitive measure of complexity, the bounds depend on the scale in which the loss function changes, as captured by the Lipschitz continuity constant of the loss function. Recall that:

Definition 5. *A function $f : \mathbb{R} \to \mathbb{R}$ is Lipschitz continuous with constant L if for every x_1, x_2:*

$$|f(x_1) - f(x_2)| \leq L|x_1 - x_2|$$

In particular, differentiable functions with a bounded derivative are Lipschitz continuous with a constant equal to the bound on the derivative. We say that a loss function loss : $\mathbb{R} \times Y \to \mathbb{R}$ is Lipschitz continuous with constant L if $\mathrm{loss}(x, y)$ is Lipschitz continuous in x for every y, with constant L.

6.2.1 Bounding with the Trace-Norm

Theorem 23. *For all target matrices $Y \in \{\pm 1\}^{n \times m}$ and sample sizes $|S| > n \log n$, and for a uniformly selected sample S of $|S|$ entries in Y, with probability at least $1 - \delta$ over the sample selection, the following holds for all matrices $X \in \mathbb{R}^{n \times m}$:*

$$\frac{1}{nm}\sum loss(X_{ia};Y_{ia}) < \frac{1}{|S|}\sum_{ia \in S} loss(X_{ia};Y_{ia}) + KL\frac{\|X\|_{\mathrm{tr}}}{\sqrt{nm}}\sqrt[4]{\ln m}\sqrt{\frac{(n+m)\ln n}{|S|}} + \sqrt{\frac{\ln(1+|\log\|X\|_{\mathrm{tr}}|)}{|S|}} + \sqrt{\frac{\ln(4/\delta)}{2|S|}}$$

Where K is a universal constant that does not depend on Y,n,m, the loss function or any other quantity, and loss is Lipschitz continuous with constant L, and we assume $n \geq m$.

By bounding the zero-one error in terms of a piecewise linear margin loss $\mathrm{loss}(x, y) = \max(0, \min(yx - 1, 1))$, which in turn is bounded by the zero-one margin loss, the generalization error bound can be specialized to bounding the true zero-one error in terms of the empirical zero-one margin error:

Corollary 24. *For all target matrices $Y \in \{\pm 1\}^{n\times m}$ and sample sizes $|S| > n\log n$, and for a uniformly selected sample S of $|S|$ entries in Y, with probability at least $1-\delta$ over the sample selection, the following holds for all matrices $X \in \mathbb{R}^{n\times m}$ and all $\gamma > 0$:*

$$\frac{1}{nm}|\{ia|X_{ia}Y_{ia} \leq 0\}| < \frac{1}{|S|}|\{ia \in S|X_{ia}Y_{ia} \leq \gamma\}| + \\ K\frac{\|X\|_{\text{tr}}}{\gamma\sqrt{nm}}\sqrt[4]{\ln m}\sqrt{\frac{(n+m)\ln n}{|S|}} + \sqrt{\frac{\ln(1+|\log\|X\|_{\text{tr}}/\gamma|)}{|S|}} + \sqrt{\frac{\ln(4/\delta)}{2|S|}}$$

To understand the scaling of this bound, it is useful to consider the scaling of the trace-norm for matrices that can be factored into $X = UV'$ where the norm of each row of U and V is bounded by r. The trace-norm of such matrices is at most $r^2\sqrt{nm}$, leading to a complexity term of r^2. Recall that the conditional problem, where V is fixed and only U is learned, is a collection of low-norm (large-margin) linear prediction problems. When the norms of rows in U and V are bounded by r, a similar generalization error bound on the conditional problem would include the term $r^2\sqrt{\frac{n}{|S|}}$, matching the term in Theorem 23 up to log-factors. We see, then, that learning *both* U and V does not introduce significantly more structural risk than learning just one of them.

Also of interest are low-rank matrices. Since the rank is not a scale-sensitive measure, we must impose a scale constraint, and we do so by bounding the entries in the matrix. For low rank matrices we have

$$\|X\|_{\text{tr}} \leq \sqrt{\operatorname{rank} X}\,\|X\|_{\text{Fro}} \leq \sqrt{\operatorname{rank} X nm}B \tag{6.5}$$

where B is a bound on the entries in the matrix. This inequality yields:

Corollary 25. *For all target matrices $Y \in \{\pm 1\}^{n\times m}$ and sample sizes $|S| > n\log n$, and for a uniformly selected sample S of $|S|$ entries in Y, with probability at least $1-\delta$ over the sample selection, the following holds for all rank-k matrices with bounded entries*

$X \in [-B, B]^{n \times m}$:

$$\frac{1}{nm}\sum loss(X_{ia};Y_{ia}) < \frac{1}{|S|}\sum_{ia\in S} loss(X_{ia};Y_{ia}) + KB\sqrt[4]{\ln m}\sqrt{\frac{k(n+m)\ln n}{|S|}} + \sqrt{\frac{\ln(4/\delta)}{2|S|}}$$

Where K is a universal constant that does not depend on Y,n,m the loss function or any other quantity, and loss is L-Lipschitz.

When the loss function is bounded only by the Lipschitz continuity and the bound on the entries in X, this bound provides the same guarantee as Theorem 22, up to log factors. However, Theorem 22 avoids the dependence on the magnitude of the entries in X when the loss function is explicitly bounded.

This is the best (up to log factors) that can be achieved without explicitly bounding the loss function. But for bounded loss functions, analyzing the covering number of bounded low-rank matrices directly, yields a bound that scales only logarithmically with B.

6.2.2 Proof of Theorem 23

To prove the theorem, we consider matrices $X \in \mathbb{R}^{n \times m}$ as functions $X : [n] \times [m] \to \mathbb{R}$ from index pairs to entries in the matrix, and bound their *Rademacher complexity* (see Appendix D) as such. The proof is then an application of Theorem 45 (Theorem 2 of [56]).

In order to calculate the Rademacher complexity of matrices with bounded trace-norm, we calculate the Rademacher complexity of unit-norm rank-one matrices,

$$\mathcal{X}_1[1] \doteq \{uv' \mid u \in \mathbb{R}^n, v \in \mathbb{R}^m, |u| = |v| = 1\}, \tag{6.6}$$

and use the fact that the Rademacher complexity does not change when we take the convex hull of this class. We first analyze the empirical Rademacher complexity, for any fixed sample S, possibly with repeating index pairs. We then bound the (average) Rademacher complexity for a sample of $|S|$ index pairs drawn uniformly at random from $[n] \times [m]$ (with repetitions). The resulting generalization error bound applies to samples selected by this process, and therefore also bounds the more concentrated situation of samples drawn without repetitions.

The Empirical Rademacher Complexity

For an empirical sample $S = \{(i_1, a_1), (i_2, a_2), \ldots\}$ of $|S|$ index pairs, the empirical Rademacher complexity of rank one unit norm matrices is the expectation:

$$\hat{R}_S(\mathcal{X}_1[1]) = \mathbf{E}_\sigma \left[\sup_{|u|=|v|=1} \left| \frac{2}{|S|} \sum_{\alpha=1}^{|S|} \sigma_\alpha u_{i_\alpha} v_{a_\alpha} \right| \right]$$

Where σ_α are uniform ± 1 random variables. For each index pair (i, a) we will denote s_{ia} the number of times it appears in the empirical sample S, and consider the random variables

$$\sigma_{ia} = \sum_{\substack{\alpha \\ (i_\alpha, a_\alpha)=(i,a)}} \sigma_\alpha.$$

Since the variables σ_α are independent,

$$\mathbf{E}\left[\sigma_{ia}^2\right] = \sum_{\substack{\alpha \\ (i_\alpha, a_\alpha)=(i,a)}} \mathbf{E}\left[\sigma_\alpha^2\right] = s_{ia} \cdot 1 = s_{ia}$$

We can now calculate:

$$\begin{aligned} \hat{R}_S(\mathcal{X}_1[1]) &= \mathbf{E}_\sigma \left[\sup_{|u|=|v|=1} \left| \frac{2}{|S|} \sum_{i,a} \sigma_{ia} u_i v_a \right| \right] \\ &= \frac{2}{|S|} \mathbf{E}_\sigma \left[\sup_{|u|=|v|=1} |u' \sigma v| \right] \\ &= \frac{2\mathbf{E}_\sigma \left[\|\sigma\|_2 \right]}{|S|} \end{aligned} \tag{6.7}$$

where σ is an $n \times m$ matrix of σ_{ia}.

The Rademacher complexity is equal to the expectation of the spectral norm of the random matrix σ (with a factor of $\frac{2}{|S|}$). Using the Frobenius norm to bound the spectral norm, we have:

$$\begin{aligned}\hat{R}_S(\mathcal{X}_1[1]) &= \frac{2}{|S|}\mathbf{E}_\sigma\left[\|\sigma\|_2\right] \le \frac{2}{|S|}\mathbf{E}_\sigma\left[\|\sigma\|_{\text{Fro}}\right] \\ &= \frac{2}{|S|}\mathbf{E}_\sigma\left[\sqrt{\sum_{ia}\sigma_{ia}^2}\right] \\ &\le \frac{2}{|S|}\sqrt{\sum_{ia}\mathbf{E}_\sigma\left[\sigma_{ia}^2\right]} \\ &= \frac{2}{|S|}\sqrt{\sum_{ia} s_{ia}} = \frac{2}{|S|}\sqrt{|S|} = \frac{2}{\sqrt{|S|}} \end{aligned} \tag{6.8}$$

As a supremum over all sample sets S, this bound is tight.

Examples of worst-case empirical Rademacher complexity

Consider a sample of $|S|$ identical index pairs, i.e. $s_{11} = |S|$ and $s_{ia} = 0$ elsewhere. The maximizing u and v have $u_1 = v_1 = 1$ and the Rademacher complexity is essentially the expectation of the distance from the origin of a 1-D $|S|$-step random walk: $\mathbf{E}\left[|\sigma_{11}|\right] \approx \sqrt{\frac{2s_{ia}}{\pi}}$ and $\hat{R}_S = \frac{2\sqrt{2}}{\sqrt{\pi|S|}}$ [80, 55].

As an even tighter example of a bad sample without repeated entries, consider a sample of $|S|$ index pairs, all in the same column. The rank-one unit-norm matrix attaining the supremum would match the signs of the matrix with $\pm\frac{1}{\sqrt{|S|}}$ yielding an empirical Rademacher complexity of $\frac{2}{\sqrt{|S|}}$.

The form of (6.8) is very disappointing, as it would lead to a term of $\frac{\|X\|_{\text{tr}}}{\sqrt{|S|}}$ in a generalization error bound using Theorem 45. Even a matrix of constant sign requires a trace-norm of $\sqrt{nm}$ to represent with margin 1. This would indicate that to get a meaningful bound we would need $|S| > nm$, i.e. more sample entries than entries in the matrix—not a very useful situation.

In order to get a meaningful bound, we must analyze the expected spectral norm more carefully.

Bounding the Expected Spectral Norm $\mathbf{E}_\sigma\left[\|\sigma\|_2\right]$

Instead of using the Frobenius norm, we bound the expected spectral norm directly. We do so by applying Theorem 3.1 of [63] (see Appendix E), which bounds the expected spectral norm of matrices with entries of fixed magnitudes but random signs in terms of the maximum row and column magnitude norms. If S contains no repeated index pairs ($s_{ia} = 0$ or 1), we are already in this situation, as the magnitudes of σ are equal to s. When some index pairs are repeated, we consider a different random matrix, $\tilde{\sigma}$ which consists of sign flips of s_{ia}:

$$\tilde{\sigma}_{ia} = \epsilon_{ia} s_{ia} \tag{6.9}$$

where ϵ_{ia} are i.i.d. unbiased signs. Applying Theorem 3.1 to $\tilde{\sigma}_{ia}$ we obtain:

$$\begin{aligned} \mathbf{E}_\epsilon\left[\|\tilde{\sigma}\|_2\right] &\leq K(\ln m)^{\frac{1}{4}}\left(\max_i |s_{i\cdot}| + \max_a |s_{\cdot a}|\right) \\ &= K(\ln m)^{\frac{1}{4}}\left(\max_i \sqrt{\sum_a s_{ia}^2} + \max_a \sqrt{\sum_i s_{ia}^2}\right) \end{aligned} \tag{6.10}$$

where $s_{i\cdot}$ and $s_{\cdot a}$ are row and column vectors of the matrix s, and K is the absolute constant guaranteed by Theorem 3.1 of [63].

To see that $\mathbf{E}_\epsilon\left[\|\tilde{\sigma}\|_2\right]$ provides an upper bound on the Rademacher complexity, we prove that such "sign consolidation" can only increase the Rademacher complexity.

Consolidating Signs of Repeated Points

We show that for any function class and distribution, the Rademacher complexity can be bounded from above by consolidating all random signs corresponding to the same point, into a single sign. We first show that consolidating a single sign can only increase the Rademacher complexity:

Lemma 26. *For any function class $\mathcal{F}$ and sample $S = (x_1, \ldots, x_n)$ with $x_1 = x_2$:*

$$\mathbf{E}_\sigma\left[\sup_{f\in\mathcal{F}}\left|\sum_{i=1}^{n} \sigma_i f(x_i)\right|\right] \leq \mathbf{E}_\sigma\left[\sup_{f\in\mathcal{F}}\left|\sigma_2 2f(x_2) + \sum_{i=3}^{n} \sigma_i f(x_i)\right|\right]$$

where σ_i are i.i.d. unbiased signs (the expectation on the right is over $n-1$, rather then n, random signs).

Proof. We first note that removing x_1, x_2 can only decrease the expectation:

$$\begin{aligned}
&\mathbf{E}_\sigma \left[\sup_{f\in\mathcal{F}} \left| \sum_{i=1}^{n} \sigma_i f(x_i) \right| \right] && (6.11)\\
&= \mathbf{E}_{\sigma_{3:n}} \left[\mathbf{E}_{\sigma_1,\sigma_2} \left[\sup_{f\in\mathcal{F}} \left| \sigma_1 f(x_1) + \sigma_2 f(x_2) + \sum_{i=3}^{n} \sigma_i f(x_i) \right| \right] \right] \\
&\geq \mathbf{E}_{\sigma_{3:n}} \left[\sup_{f\in\mathcal{F}} \left| \mathbf{E}_{\sigma_1} [\sigma_1 f(x_1)] + \mathbf{E}_{\sigma_2} [\sigma_2 f(x_2)] + \sum_{i=3}^{n} \sigma_i f(x_i) \right| \right] \\
&= \mathbf{E}_{\sigma_{3:n}} \left[\sup_{f\in\mathcal{F}} \left| \sum_{i=3}^{n} \sigma_i f(x_i) \right| \right] && (6.12)
\end{aligned}$$

And now calculate, using (6.12) for the inequality:

$$\begin{aligned}
&\mathbf{E}_\sigma \left[\sup_{f\in\mathcal{F}} \left| \sum_{i=1}^{n} \sigma_i f(x_i) \right| \right] \\
&= \Pr(\sigma_1 \neq \sigma_2) \mathbf{E}_\sigma \left[\sup_{f\in\mathcal{F}} \left| \sum_{i=3}^{n} \sigma_i f(x_i) \right| \right] \\
&\quad + \Pr(\sigma_1 = \sigma_2) \mathbf{E}_\sigma \left[\sup_{f\in\mathcal{F}} \left| \sigma_2 2 f(x_2) + \sum_{i=3}^{n} \sigma_i f(x_i) \right| \right] \\
&\leq \frac{1}{2} \mathbf{E}_\sigma \left[\sup_{f\in\mathcal{F}} \left| \sum_{i=1}^{n} \sigma_i f(x_i) \right| \right] + \frac{1}{2} \mathbf{E}_\sigma \left[\sup_{f\in\mathcal{F}} \left| \sigma_2 2 f(x_2) + \sum_{i=3}^{n} \sigma_i f(x_i) \right| \right]
\end{aligned}$$

Subtracting the first term on the right hand side from the original left hand side, gives us the desired inequality. □

By iteratively consolidating identical sample points, we get:

Lemma 27 (Sign Consolidation). *For any function class $\mathcal{F}$ and sample $S = (x_1, \ldots, x_n)$, denote s_x the number of times a sample appears in the class, and let σ_x be i.i.d. unbiased random signs, then:*

$$\mathcal{R}_S(\mathcal{F}) \leq \mathbf{E}_\sigma \left[\sup_{f\in\mathcal{F}} \left| \frac{2}{|S|} \sum_{x\in S} \sigma_x s_x f(x) \right| \right]$$

Bounding the Row and Column Norms of a Uniformly Random Sample

We now consider the Rademacher complexity with respect to a sample S of $|S|$ (where $|S|$ is fixed) index pairs, chosen independently and uniformly at random:

$$\begin{aligned} R_S(\mathcal{X}_1[1]) &= \frac{2}{|S|}\mathbf{E}_S\left[\hat{R}_S(\mathcal{X}_1[1])\right] \\ &\leq \frac{K}{|S|}(\ln m)^{\frac{1}{4}}\left(\mathbf{E}_S\left[\max_i |s_{i\cdot}|\right] + \mathbf{E}_S\left[\max_a |s_{\cdot a}|\right]\right) \end{aligned} \tag{6.13}$$

For the worst samples, the norm of a single row or column vector of s might be as high as $|S|$, but for random uniformly drawn samples, we would expect the norm of row vectors to be roughly $\frac{|S|}{n}$ and of column vectors to be roughly $\frac{|S|}{m}$. To make this estimate precise we proceed in two steps.

We will first bound the maximum value of s_{ia}, uniformly over all index pairs. When the maximum entry in s is bounded, the norm of a row can be bounded by the number of observations in the row. In the second step we will bound the number of observations in a row and conclude a bound on the maximal row (and similarly column) norm.

In deriving these bounds, we assume $m \leq n < |S| < nm$. We also assume $m > 3$ in order to simplify some of the logarithmic factors and constants.

Lemma 28.

$$\Pr_S\left(\max_{ia} s_{ia} > 9\ln n\right) \leq \frac{1}{|S|}$$

Proof. Using Bernstein inequality for the binomial distribution (Corollary 39), with $t = 2\ln(nm|S|^2)$, for every s_{ia}:

$$\Pr\left(s_{ia} > \frac{|S|}{nm} + 2\ln(nm|S|)\right) \leq \exp\left(-\frac{t^2}{t + 2\frac{S}{nm}}\right) \leq \exp\left(-\frac{t}{2}\right) = \frac{1}{nm|S|}$$

Taking a union bound over all nm entries in s, and bounding $\ln(nm|S|) \leq 4\ln n$ and $\frac{|S|}{nm} \leq 1 \leq \ln(n)$ we get the desired bound. □

We now bound $\mathbf{E}_S\left[\max_i |s_{i\cdot}|\right]$, for samples in which $s_{ia} < B$ for all s_{ia}:

$$\begin{aligned}\mathbf{E}_S\left[\max_i |s_{i\cdot}| \mid \forall s_{ia} < B\right] &= \mathbf{E}_S\left[\max_i \sqrt{\sum_a s_{ia}^2} \mid \forall s_{ia} < B\right] \\ &= \mathbf{E}_S\left[\sqrt{\max_i \sum_a B s_{ia}}\right] \leq \sqrt{B\mathbf{E}_S\left[\max_i s_i\right]}\end{aligned}$$

Where s_i is the number of observation in the row i. Viewing the sample as $|S|$ independent and uniform selections of i rows, we can bound $\mathbf{E}_S\left[\max_i s_i\right] \leq 6(\frac{|S|}{n} + \ln|S|)$ using Theorem 43.

Combining this bound with Lemma 28, we can now meaningfully bound the Rademacher complexity, for a random sample set where each index pair is chosen uniformly and independently at random (on each line, K designates some fixed universal constant, but this constant changes from line to line):

$$\begin{aligned}&R_{|S|}^{\text{uniform}}(\mathcal{X}_1[1]) = \mathbf{E}_S\left[R_S(\mathcal{X}_1[1])\right] \\ &\leq \Pr\left(\max_{ia} s_{ia} > 9\ln n\right) \sup_S R_S(\mathcal{X}_1[1]) + \mathbf{E}_S\left[R_S(\mathcal{X}_1[1]) \,\middle|\, \max_{ia} s_{ia} \leq 9\ln n\right] \\ &\leq \frac{1}{|S|}\frac{2}{\sqrt{|S|}} + \frac{K}{|S|}(\ln m)^{\frac{1}{4}}\mathbf{E}_S\left[\max_i |s_{i\cdot}| + \max_a |s_{\cdot a}| \,\middle|\, \max_{ia} s_{ia} \leq 9\ln n\right] \\ &\leq \frac{2}{|S|} + \frac{K}{|S|}(\ln m)^{\frac{1}{4}}\left(\sqrt{6\cdot 9\ln n(\frac{|S|}{n} + \ln|S|)} + \sqrt{6\cdot 9\ln n(\frac{|S|}{m} + \ln|S|)}\right) \\ &\leq K\frac{1}{\sqrt{nm}}(\ln m)^{\frac{1}{4}}(\ln n)^{\frac{1}{2}}\sqrt{\frac{n + m + \frac{nm}{|S|}\ln n}{|S|}}\end{aligned}$$

(in the last inequality we also used $2\ln n \geq \ln|S|$)

So far, we bounded the Rademacher complexity of unit-norm rank-one matrices, $\mathcal{X}_1[1]$. Taking the convex hull of this class (Lemma 11) and scaling by the desired norm we have (following Theorem 46):

$$R(\mathcal{B}[M]) = R(M\text{conv}\mathcal{X}_1[1]) = MR(\mathcal{X}_1[1])$$

establishing:

Theorem 29. *For some universal constant K, the Rademacher complexity of matrices of trace-norm at most M, over uniform samplings of index pairs is at most (for $n \geq m$):*

$$R(\mathcal{B}[M]) \leq K\frac{M}{\sqrt{nm}}(\ln m)^{\frac{1}{4}}\sqrt{\frac{(n+m+\frac{nm}{|S|}\ln n)\ln n}{|S|}}$$

When $|S| > n \ln n$, the last term can be subsumed in the constant K.

6.2.3 Bounding with the Max-Norm

Since the max-norm gives us a bound on the trace-norm:

$$\|X\|_{\text{tr}} \leq \sqrt{nm}\,\|X\|_{\max} \quad \text{for every } X \in \mathbb{R}^{n \times m}$$

we can apply Theorem 23 also to matrices of bounded max-norm, replacing $\frac{\|X\|_{\text{tr}}}{\sqrt{nm}}$ with $\|X\|_{\max}$. However, when the max-norm is bounded it is possible to more simply obtain slightly better bounds, avoiding the log-terms and with explicit constants:

Theorem 30. *For all target matrices $Y \in \{\pm 1\}^{n \times m}$ and sample sizes $|S| > n \log n$, and for a uniformly selected sample S of $|S|$ entries in Y, with probability at least $1 - \delta$ over the sample selection, the following holds for all matrices $X \in \mathbb{R}^{n \times m}$:*

$$\frac{1}{nm}\sum loss(X_{ia}; Y_{ia}) < \frac{1}{|S|}\sum_{ia \in S} loss(X_{ia}; Y_{ia}) +$$
$$12\,\|X\|_{\max}\sqrt{\frac{n+m}{|S|}} + \sqrt{\frac{\ln(1+|e \log \|X\|_{\max}|)}{|S|}} + \sqrt{\frac{\ln(4/\delta)}{2|S|}}$$

Where loss is Lipschitz continuous with constant L. For large enough n, m, the constant 12 can be reduced to $k_R\sqrt{8 \ln 2} < 4.197$, where k_R is Grothendiek's constant (see Appendix F).

Moreover, unlike the bound in terms of the trace-norm (Theorem 23), the bound in terms of the max-norm can be generalized to index pairs chosen under an arbitrary distribution, with the generalization error measured appropriately:

Theorem 31. *For all target matrices $Y \in \{\pm 1\}^{n \times m}$ and sample sizes $|S| > n \log n$, and for any distribution D over index pairs, for a sample S of $|S|$ i.i.d. index pairs selected according to D, with probability at least $1 - \delta$ over the sample selection, the following holds for all matrices $X \in \mathbb{R}^{n \times m}$:*

$$\mathbf{E}_{ia \sim D}\left[loss(X_{ia}; Y_{ia})\right] < \frac{1}{|S|} \sum_{ia \in S} loss(X_{ia}; Y_{ia}) + \\ 12 \|X\|_{\max} \sqrt{\frac{n+m}{|S|}} + \sqrt{\frac{\ln(1 + |e \log \|X\|_{\max}|)}{|S|}} + \sqrt{\frac{\ln(4/\delta)}{2|S|}}$$

Where loss is Lipschitz continuous with constant L and the constant can be improved as in Theorem 30.

This generalization is a consequence of the empirical Rademacher complexity of low max-norm matrices being bounded for any sample set of indexes. As was discussed in Section 6.2.2, this is not the case for low trace-norm matrices, for which the empirical Rademacher complexity might be high, and only the average Rademacher complexity over uniformly selected index pairs is low.

6.2.4 Proof of Theorem 30

To prove Theorems 30 and 31 we bound the empirical Rademacher complexity of low max-norm matrices, again viewing them as functions $X : [n] \times [m] \to \mathbb{R}$ from index pairs to entries in the matrix.

As we did for low trace-norm matrices, we bound the Rademacher complexity of low max-norm matrices by characterizing the unit ball of the max-norm (i.e. unit max-norm matrices) as a convex hull. Unlike the trace-norm unit ball, we cannot exactly characterize the max-norm unit ball as a convex hull. However, using Grothendiek's Inequality (see Appendix F) we can bound the unit ball as with the convex hull of rank-one sign matrices:

$$\mathrm{conv}\mathcal{X}_{\pm} \subset \mathcal{B}_{\max} \subset 2\mathrm{conv}\mathcal{X}_{\pm} \tag{6.14}$$

where

$$\mathcal{B}_{\max} = \{X \mid \|X\|_{\max} \leq 1\}$$

is the unit max-norm ball and

$$\mathcal{X}_{\pm} = \{uv' \mid u \in \{-1,+1\}^n, v \in \{-1,+1\}^m\} = \{X \in \{-1,+1\} \mid \operatorname{rank} X = 1\}.$$

The class of rank-one sign matrices is a finite class of size $|\mathcal{X}_{\pm}| = 2^{n+m-1}$, and so its Rademacher complexity can be bounded by (Theorem 47):

$$\hat{\mathcal{R}}_S(\mathcal{X}_{\pm}) < \sqrt{7\frac{2(n+m)+\log|S|}{|S|}} \tag{6.15}$$

Taking the convex hull of this class and scaling by the desired norm we have (following Theorem 46):

$$\begin{aligned} R(\mathcal{B}_{\max}[M]) &< R(2M\operatorname{conv}\mathcal{X}_{\pm}) \\ &< 2M\sqrt{7\frac{2(n+m)+\log|S|}{|S|}} < 12M\sqrt{\frac{n+m}{|S|}} \end{aligned} \tag{6.16}$$

where in the last inequality we use $2 < |S| < nm$. This establishes:

Theorem 32. *The Rademacher complexity of matrices of trace-norm at most M, for any index-pair distribution, is at most:*

$$R(\mathcal{B}_{\max}[M]) \leq 12M\sqrt{\frac{n+m}{|S|}}$$

For large enough n, m, the constant 12 can be reduced to $k_R\sqrt{8\ln 2} < 4.197$, where k_R is Grothendiek's constant (see Appendix F).

6.3 On assuming a random observations

A major assumption we make throughout the treatment of the matrix completion problem is that the entries to be observed (indexes in S) are selected randomly, independent of one another (except perhaps avoiding repetitions), and independent of the target Y. This assumption underlies the learning method suggested, and is made explicit in the generalization error bounds.

However, this assumption is often unrealistic. For example, it would imply a binomial distribution on the number of movies people rate, and on the number of people who rate a movie. A heavier tailed distribution is probably more realistic, implying dependencies among the choice of observations.

More importantly, whether a rating is observed or not can be related to the rating itself. User are more likely to see, and rate, movies they like. An extreme example of such dependencies is in collaborative filtering situations where the preferences are implied by user requests, where all observations are assumed positive.

Even in more subtle situations, significant benefit can probably be gained by modeling the observation process and its relationship with the target values.

The bound in terms of the trace-norm (Theorem 23) heavily relies on the uniformity of the sample selection. Since the trace-norm is *average* over rows and columns, it is not surprising that it is an effective constraint only when all rows and column are used uniformly. Indeed, as was discussed in Section 6.2.2, bounding the trace-norm is not effective when only a subset of rows or columns is used.

Requiring low rank or low max-norm constrains all rows and columns uniformly. Indeed, the generalization error bounds in terms of these complexity measures apply also when the indexes are not chosen at random, and even when the observation process is dependent on the ratings themselves (i.e. to the target matrix Y). However, in such cases, the guarantee is on the expected loss when future entries are sampled under the observed subset. In a collaborative filtering setting, this is extremely unsatisfying, as with would guarantee low error on items the user is likely to want anyway, but not on items we predict he would like.

Chapter 7

Summary

In this thesis, we examined several aspects of learning with matrix factorizations. Learning with matrix factorizations is not a new idea: low-rank and factor models have been extensively used in statistical analysis of tabulated data for over a century [58]. Throughout this century of matrix factorization, new formulations, methods and analysis have been devised, based on the central motivating assumption that tabulated data can be modeled by underlying factors. Continued research on learning with matrix factorizations is fueled by evolving trends and approaches to statistical analysis and machine learning, such as the study of exponential families and generalized linear models, high-dimensional large-margin linear methods, and distribution-free post-hoc generalization error bounds, as well as by advances in convex optimization and the constant growth in sheer processing power. This thesis continues this tradition, and offers several novel contributions to the field:

Study of Weighted Low Rank Approximations We show how the structure of the optimization function breaks down when weights are introduced, and suggest novel local search heuristics for finding weighted low-rank approximations. These include a very simple update inspired by Expectation Maximization and a more complex conjugate gradient method. We show how weighted low-rank approximations can be used as a procedure in other low-rank optimization problems, including ones with convex loss functions or with additive noise modeled as a (possibly unknown) Gaussian mixture.

Asymptotic Consistency and Inconsistency We show that asymptotic consistency of maximum likelihood low-rank approximations should not be taken for granted. For a Gaussian noise model, estimation of the low-rank subspace is consistent, but we show that for a variety of other conditional models, including some that have recently been suggested and studied (e.g. Exponential PCA and Logistic Low-Rank Approximation in particular) estimation is not consistent even when the data follows the modeling assumptions. On the other hand, we show that simple Frobenius low-rank approximation (using the SVD) is consistent for any additive noise, even when the noise distribution is not Gaussian. For non-additive noise models, we are able to provide an appropriate correction to Frobenius low-rank approximation only for *unbiased* conditional models, i.e. only when a low-rank approximation to the *mean* parameters is sought. This leaves open the important problem of consistent estimation of a linear subspace of the *natural* parameters (as in Exponential PCA).

Maximum Margin Matrix Factorizations We propose a novel method for completing entries in a partially observed matrix: instead of approximating the observed entries with a low-rank factorization, we approximate the observed entries with a low-norm factorization while maintaining a large-margin. Unlike low-rank matrix approximation of a partially observed matrix, which is a non-convex optimization problem for which no efficient solutions are known, maximum-margin low-norm matrix factorization is a convex optimization problem that can be formalized and solved as a semi-definite program. Using generic optimization methods for sparse semi-definite programs we are able to find maximum-margin matrix factorizations for problems with up to a few tens of thousands of observations—far from the size of actual data sets. The applicability of the methods to large data sets is contingent on developing specialized optimization techniques which take advantage of the very simple structure of the dual semi-definite programs.

Post-hoc Generalization Error Bounds We present, for the first time, post-hoc generalization error bounds, without assumptions on the "true" preferences, for collaborative filtering viewed as a matrix completion problem. We present bounds both for low-rank approximation, based on combinatorial results on the number of sign configurations of low-rank matrices, and for maximum-margin matrix factorization, based on bounding the Rademacher complexity of low trace-norm and low sum-norm (γ_2-norm) matrices. All of our results assume a random observation process—an assumption which often does not hold in practice. An important challenge is to develop generalization error bounds for the more realistic scenario in which the observation process is dependent on the value of the entries in the matrix.

Appendix A

Sign Configurations of Polynomials

We briefly quote here a discussion from Alon [3] on the number of sign configurations of a set of real polynomials.

Let $P_1, \ldots, P_m$ be real polynomials in q variables, and let V be the complement of the variety defined by $\Pi_i P_i$, i.e. the set of points in which all the m polynomials are non-zero:

$$V = \{x \in \mathbb{R}^q | \forall_i P_i(x) \neq 0\}$$

Theorem 33 (Warren [78], Theorem 5.2 [3]). *If all m polynomials are of degree at most d, then the number of connected components of V is at most:*

$$c(V) \leq 2(2d)^q \sum_{i=0}^{q} 2^i \binom{m}{i} \leq \left(\frac{4edm}{q}\right)^q$$

where the second inequality holds when $m > q > 2$.

We are interested in the number of sign configurations of the polynomials, i.e. the cardinality of:

$$S = \{(\text{sign } P_1(x), \text{sign } P_2(x), \ldots, \text{sign } P_m(x)) \in \{-, 0, +\}^m \mid x \in \mathbb{R}^q\}$$

Each connected component of V maps to a single sign vector. And so, the number of connected components of V bounds the number of sign configurations that do not contains zeros:

$$|S \cap \{-,+\}^m| \leq c(V)$$

To bound the number of sign configurations, including those with zeros, we will modify the polynomials slightly. Consider a set $C \subset \mathbb{R}^q$ containing one variable configuration for each sign pattern in S, i.e. $|C| = |S|$ and such that $S = \{(\text{sign } P_i(x)) \in \{-,0,+\}^m \mid x\, C\}$. Define ϵ as:

$$\epsilon \doteq \frac{1}{2} \min_{1\leq i\leq m, x\in CP_i(x)\neq 0} |P_i(x)| \tag{A.1}$$

Since S is finite (at most 3^m sign vectors are possible), C is also finite, the minimum is justifies, and $\epsilon > 0$. We can now consider the $2m$ polynomials $P_i^+(x) = P_i(x) + \epsilon$ and $P_i^-(x) = P_i(x) - \epsilon$ and:

$$V' = \left\{x \in \mathbb{R}^q | \forall_i P_i^+(x) \neq 0, P_i^-(x) \neq 0\right\}$$

Different points in C lie in different connected components of V', and so $|S| = |C| \leq |c(V')|$ establishing:

Theorem 34 ([3, Proposition 5.5]). *The number of sign configurations of m polynomials, each of degree at most d, over q variables, is at most* $(8edm/q)^q$ *(for* $2m > q > 2$*).*

If we consider only +/- signs by identifying zero as (arbitrarily) positive, instead of ignoring zeros, that is $\text{sign}^{\pm} p = \begin{cases} + & p \geq 0 \\ - & \text{otherwise} \end{cases}$, it is enough to take the modified polynomials $P_i^-(x) = P_i(x) - \epsilon$, obtaining a bound of $(4edm/q)^q$. This is true also if we identify zero as positive or negative differently for each polynomial:

Theorem 35. *Let* $P_1, \ldots, P_m$ *be polynomials over q variables, each of degree at most d and* $y_1, \ldots, y_m \in \pm 1$. *Define the relative sign configuration for a variables assignment* $x \in \mathbb{R}^q$ *as*

$$s_i(x) = \begin{cases} + & y_i P_i(x) > 0 \\ - & y_i P_i(x) \leq 0 \end{cases}.$$

The number of different relative sign configurations is at most $(4edm/q)^q$ *(for* $m > q > 2$*).*

Appendix B

Basic Concentration Inequalities and Balls in Bins

We quote here basic concentration inequalities about the sums of random variables that are used in the Thesis. In Section B.2.1 we use Bernstein's inequality to bound the expected number of balls in the fullest bin, when balls are tossed randomly into bins.

B.1 Chernoff and Heoffding Bounds

Chernoff's original inequality applies to a binomial distribution, i.e. a sum of i.i.d. Bernouli random variables:

Theorem 36. *Let S Binom(n,p) be a binomial random variable. For any $\epsilon > 0$:*

$$\Pr\left(S > \mathbf{E}\left[S\right] + \epsilon\right) \leq e^{-2\epsilon^2/n}$$

and

$$\Pr\left(S < \mathbf{E}\left[S\right] - \epsilon\right) \leq e^{-2\epsilon^2/n}$$

Heoffding relaxed the assumptions that the variables are identically distributed, and that they are Bernoulli:

Theorem 37 (Heoffding 1963, [24, Theorem 8.1]). *Let X_i be independent random vari-*

ables such that $a_i \leq X_i \leq b_i$ with probability one for all i, then for any $\epsilon > 0$:

$$\Pr\left(\sum_{i=1}^{n} X_i > \sum_{i=1}^{n} \mathbf{E}\left[X_i\right] + \epsilon\right) \leq e^{-2\epsilon^2/\sum_{i=1}^{n}(b_i-a_i)^2}$$

and

$$\Pr\left(\sum_{i=1}^{n} X_i < \sum_{i=1}^{n} \mathbf{E}\left[X_i\right] - \epsilon\right) \leq e^{-2\epsilon^2/\sum_{i=1}^{n}(b_i-a_i)^2}$$

Chernoff's and Heoffding's inequalities are *pessimistic.* They do not depend on the distribution of the summed random variables X_i, and assume the worst possible variance. For Bernoulli random variables, this variance is achieved, and the inequality is tightest, when $P(X_i = 1) = \frac{1}{2}$. However, when the probability of 'success' in each trial is very low, i.e. $\mathbf{E}\left[S\right]$ is small, the inequality is very loose. Bernstein's and Bennett's inequalities are tighter when the variance is small.

B.2 The Bernstein Bound and Balls in Bins

Theorem 38 (Bernstein 1946, [24, Theorem 8.2]). *Let X_i be independent random variables such that $X_i < c$ with probability one for all i, then for any $\epsilon > 0$:*

$$\Pr\left(\sum_{i=1}^{n} X_i > \sum_{i=1}^{n} \mathbf{E}\left[X_i\right] + \epsilon\right) \leq \exp\left(-\frac{\epsilon^2}{2\sum_{i=1}^{n} \mathit{Var}\left[X_i\right] + \frac{2}{3}nc\epsilon}\right)$$

Specialized to the binomial distribution, Bernstein's bound can be written as:

Corollary 39.

$$\Pr\left(\mathit{Binom}(n,p) > \frac{n}{p} + t\right) \leq \exp\left(-\frac{t^2}{t + 2np}\right)$$

Note that the $(1-p)$ term was dropped from the variance, and the factor $\frac{2}{3}$ was also dropped, giving a slightly looser version of the bound, specialized to small p.

B.2.1 The Expected Number of Balls in the Fullest Bin

Consider an experiment in which n balls are independently and uniformly tossed, each ball to one of m bins. Denote by s_i the number of balls in bin i. We can use Bernstein's inequality to bound the expected number of balls in the fullest bin, i.e. $\mathbf{E}\left[\max_i s_i\right]$.

Lemma 40. *For $7 < n \leq m$, if n balls are tossed into m bins:*

$$\mathbf{E}\left[\max_i s_i\right] \leq 3\ln m$$

Proof. Each s_i is Binomially distributed with n trials and success probability $\frac{1}{m}$, and so using Corollary 39 of Bernstein's bound, and setting $t = (2\ln(nm) - 1 - \frac{n}{m})$:

$$\begin{aligned}
\Pr\left(s_i > 3\ln m - 1\right) &\leq \exp\left(-\frac{(3\ln m - 1 - \frac{n}{m})^2}{3\ln m - 1 - \frac{n}{m} + 2n\frac{1}{m}}\right) \\
&\leq \exp\left(-\frac{(3\ln m - 2)^2}{3\ln m}\right) \\
&\leq \exp\left(-\frac{9(\ln m)^2 - 6\ln m + 4}{2\ln m}\right) \\
&\leq \exp\left(-3\ln m + 2\right) \leq exp(-2\ln m) = \frac{1}{m^2}
\end{aligned}$$

Taking the union of these events over all bins, the probability that at least one bin has more than $3\ln m - 1$ balls is at most $\frac{m}{m^2} = \frac{1}{m}$. Noting that in any case, the maximum is at most n, we have:

$$\begin{aligned}
\mathbf{E}\left[\max_i s_i\right] &\leq \Pr\left(\max_i s_i > 3\ln m - 1\right)n + 3\ln m - 1 \\
&\leq \frac{1}{m}n + 3\ln m - 1 \leq 3\ln m
\end{aligned}$$

□

Lemma 41. *For $\frac{n}{\ln n} \leq m \leq n$:*

$$\mathbf{E}\left[max_i s_i\right] \leq 6\ln n$$

Proof. We use the same argument, with $t = 4\ln n$, getting:

$$\Pr\left(s_i > \frac{n}{m} + 4\ln n\right) \leq \exp\left(-\frac{(4\ln n)^2}{4\ln n + 2\frac{n}{m}}\right)$$
$$\leq \exp\left(-\frac{16(\ln n)^2}{6\ln n}\right) \leq \exp(-2\ln n) = \frac{1}{n^2}$$

And so:

$$\mathbf{E}\left[\max_i s_i\right] \leq \Pr\left(\max_i s_i > \frac{n}{m} + 4\ln n\right)n + \frac{n}{m} + 4\ln n$$
$$\leq \frac{m}{n^2}n + \frac{n}{m} + 4\ln n \leq 6\ln n$$

□

Lemma 42. *For* $m \leq \frac{n}{\ln n}$*:*

$$\mathbf{E}\left[\max_i s_i\right] \leq \frac{n}{m} + 4\sqrt{\frac{n}{m}\ln n} \leq 5\frac{n}{m}$$

Proof. Choose $t = \sqrt{11\frac{n}{m}\ln n}$:

$$\Pr\left(s_i > \frac{n}{m} + \sqrt{11\frac{n}{m}\ln n}\right) \leq \exp\left(-\frac{11\frac{n}{m}\ln n}{\sqrt{11\frac{n}{m}\ln n} + 2\frac{n}{m}}\right)$$
$$\leq \exp\left(-\frac{11\frac{n}{m}\ln n}{\sqrt{\frac{n}{m}}(\sqrt(11)\sqrt{\ln n} + 2\sqrt{\frac{n}{m}})}\right)$$

Since $\ln n \leq \frac{n}{m}$ and $\sqrt{\ln n} \leq \sqrt{\frac{n}{m}}$:

$$\leq \exp\left(-\frac{11\frac{n}{m}\ln n}{(\sqrt{11}+2)\frac{n}{m}}\right) < \exp(-2\ln n) = \frac{1}{n^2}$$

And so:

$$\mathbf{E}\left[\max_i s_i\right] \leq \Pr\left(\max_i s_i > \frac{n}{m} + \sqrt{11\frac{n}{m}\ln n}\right)n + \frac{n}{m} + \sqrt{11\frac{n}{m}\ln n}$$
$$\leq \frac{m}{n^2}n + \frac{n}{m} + \sqrt{11\frac{n}{m}\ln n} < \frac{n}{m} + 4\sqrt{\frac{n}{m}\ln n} \leq 5\frac{n}{m}$$

□

Combined, these three lemmas cover the entire range of ratios of balls to bins:

Theorem 43. *If n balls are tossed into m bins uniformly and independently at random, then the expectation of the number of balls in the fullest bin is at most:*

$$6\max\left(1, \frac{n}{m}, \ln n, \ln m\right)$$

Appendix C

Generalization Error Bounds in Terms of the Pseudodimension

We quote here standard results from statistical machines theory (e.g. [7, Chapters 17-18]).

Definition 6. *A class $\mathcal{F}$ of real-valued functions* pseudo-shatters *the points $x_1, \ldots, x_n$ with thresholds $t_1, \ldots, t_n$ iff for every binary labeling of the points $(s_1, \ldots, s_n) \in \{+, -\}^n$ there exists $f \in \mathcal{F}$ s.t. $f(x_i) < t_i$ iff $s_i = -$. The* pseudodimension *of a class $\mathcal{F}$ is the supremum over n for which there exist n points and thresholds that can be shattered.*

Definition 7. *We say a loss function $loss : \mathbb{R} \times \mathcal{Y} \rightarrow \mathbb{R}$ is* monotone *if for all $y \in Y$, $loss(x, y)$ is a monotone (either increasing or decreasing) function of y.*

Theorem 44. *Let $\mathcal{F}$ be a class of real-valued functions $f : \mathcal{X} \rightarrow \mathbb{R}$ with pseudodimension d, and $loss : \mathbb{R} \times \mathcal{Y} \rightarrow \mathbb{R}$ be a bounded monotone loss function, with $loss < M$. For any joint distribution over (X, Y), consider an i.i.d. sample $S = (X_1, Y_1), \ldots, (X_n, Y_n)$. Then for any $\epsilon > 0$:*

$$\Pr_S \left(\exists_{f \in \mathcal{F}} \mathbf{E}_{X,Y} \left[loss(f(X), Y) \right] > \frac{1}{n} \sum_{i=1} n loss(f(X_i), Y_i) + \epsilon \right) < 4e(d+1) \left(\frac{32eM}{\epsilon} \right)^d e^{-\frac{epsilon^2 n}{32}}$$

The bound is a composition of a generalization error bound in terms of the L_1 covering number [7, Theorem 17.1] and a bound on the L_1 covering number in terms of the pseudodimension [34], as well as the observation that composition with a monotone function does not increase the pseudodimension [7, Theorem 12.3],[33].

Appendix D

Bounding the Generalization Error Using Rademacher Complexity

The Rademacher complexity is a scale-sensitive measure of complexity for a class of real valued function.

Definition 8. *The empirical Rademacher complexity of a class of function $\mathcal{F} : \mathcal{X} \to \mathbb{R}$ over a specific sample $S = (x_1, x_2, \ldots) \in \mathcal{X}^{|S|}$ is given by:*

$$\hat{\mathcal{R}}_S(\mathcal{F}) = \frac{2}{|S|}\mathbf{E}_\sigma\left[\sup_{f\in\mathcal{F}}\left|\sum_i \sigma_i f(x_i)\right|\right]$$

where the expectation is over the i.i.d. random signs σ_i, with $\Pr(\sigma_i = +1) = \Pr(\sigma_i = -1) = \frac{1}{2}$.

The Rademacher complexity of $\mathcal{F}$ with respect to a distribution D over $\mathcal{X}$ is the expectation of the empirical Rademacher complexity:

$$\mathcal{R}_n^D(\mathcal{F}) = \mathbf{E}_{S\sim D^n}\left[\hat{\mathcal{R}}_S(\mathcal{F})\right]$$

where the expectation is over and i.i.d. sample of $|S|$ points chosen according to D.

The distribution-free Rademacher complexity of $\mathcal{F}$ is the supremum of over all samples of $|S|$ points:

$$\mathcal{R}_n^{\sup}(\mathcal{F}) = \sup_{S\in\mathcal{X}^n}\hat{\mathcal{R}}_S(\mathcal{F})$$

and is an upper bound of the Rademacher complexity over any distribution.

The Rademacher complexity can be used to bound the generalization error:

Theorem 45 ([56, Theorem 2]). *For any class of function $\mathcal{F} : \mathcal{X} \rightarrow \mathbb{R}$ and any joint distribution D of (X, Y) over $\mathcal{X} \times \{\pm 1\}$, for a sample of a D-i.i.d. sample (X_i, Y_i) of size of size n, with probability at least $1 - \delta$ with respect to choosing the sample, all functions $f \in \mathcal{F}$ satisfy:*

$$\Pr_{(X,Y)\sim D}(Yf(X) \leq 0) \leq \frac{1}{n}\sum_{i=1}^{n} loss^{\gamma}(f(X_i); Y_i) + \frac{2}{\gamma}\mathcal{R}_n^D(\mathcal{F}) + \sqrt{\frac{\ln(2/\delta)}{2|S|}} \tag{D.1}$$

Furthermore, with probability at least $1 - \delta$ with respect to choosing the sample, all function $f \in \mathcal{F}$ satisfy:

$$\Pr_{(X,Y)\sim D}(Yf(X) \leq 0) \leq$$
$$\inf_{\gamma>0}\left(\frac{1}{n}\sum_{i=1}^{n} loss^{\gamma}(f(X_i); Y_i) + \frac{4}{\gamma}\mathcal{R}_n^D(\mathcal{F}) + \sqrt{\frac{\ln(1 + |\log\gamma|)}{n}}\right) + \sqrt{\frac{\ln\frac{4}{\delta}}{2|S|}} \tag{D.2}$$

The presentation here differs from the original presentation [56]:

- Only the form (D.2) was presented in [56]. The form (D.1) is an intermediate result, and is the presentation in, e.g. [11].
- The presentation here is specialized to the zero-one margin loss loss$^{\gamma}$, as a bound on the truncated hinge loss. The original presentation applies to any Lipschitz-bounded bound on the zero-one sign loss.
- The original presentation takes an infimum over $0 < \gamma \leq 1$. Here, a slightly modified form is presented, where the bound is over any $0 < \gamma$. This form can be easily derived from Theorem 1 in [56] by taking a union bound over the two cases $\gamma < 1$ and $\gamma > 1$. As a result, the failure probability doubles (expressed in the bound by a $\frac{4}{\delta}$ instead of $\frac{2}{\delta}$ inside the logarithm of the last term). The expression of the third term also has to

change to accommodate $\gamma > 1$, and $\ln(1 + |\log\gamma|)$ replaces $\ln\log\frac{2}{\gamma}$ in the original presentation.

Some properties of the Rademacher complexity (e.g. from [11, Theorem 12]):

Theorem 46. *For any classes of functions:*

1. *If $F \subset H$ then $\mathcal{R}_n(F) \leq \mathcal{R}_n(H)$.*
2. *$\mathcal{R}_n(F) = \mathcal{R}_n(\text{conv}F)$.*
3. *For $c \in \mathbb{R}$, $\mathcal{R}_n(cF) = |c|\mathcal{R}_n(F)$.*
4. *$\mathcal{R}_n(\sum F_i) \leq \sum_i \mathcal{R}_n(\sum F_i)$.*

Rademacher Complexity of a Finite Class of Sign Functions

Theorem 47. *Let $\mathcal{F}$ be a finite class of sign functions functions $\mathcal{X} \to \{-1, +1\}$, then its empirical Rademacher complexity, and so also its distribution free Rademacher complexity, is bounded by:*

$$\hat{\mathcal{R}}_S(\mathcal{F}) < \sqrt{7\frac{2\log|\mathcal{F}| + \log|S|}{|S|}}$$

The constant can be reduced to $4\ln 2 < 2.773$ *for large enough* $|\mathcal{F}|$.

Proof. For any $f \in \mathcal{F}$, $\sum_i f(x_i)\sigma_i$ are all identically distributed (when σ are random), and it is enough to analyze $sum_i\sigma_i$. Using Chernoff's bound (Theorem 36), for any $\alpha > 0$:

$$\Pr_{\sigma}\left(\frac{2}{|S|}\sum_i \sigma_i \geq \frac{\alpha}{|S|}\right) = \Pr\left(\text{Binom}(|S|, \frac{1}{2}) \geq \frac{|S|}{2} + \frac{alpha}{4}\sqrt{|S|}\right) \leq e^{-\alpha^2/8} \quad \text{(D.3)}$$

And so for any $f \in \mathcal{F}$ we have:

$$\Pr_{\sigma}\left(\left|\frac{2}{|S|}\sum_i f(x_i)\sigma_i\right| \geq \frac{\alpha}{|S|}\right) \leq 2e^{-\alpha^2/8} \quad \text{(D.4)}$$

Taking a union bound over all $f \in \mathcal{F}$:

$$\Pr_{\sigma}\left(\sum_{f\in\mathcal{F}}\left|\frac{2}{|S|}\sum_i f(x_i)\sigma_i\right| \geq \frac{\alpha}{|S|}\right) \leq 2|\mathcal{F}|e^{-\alpha^2/8} \quad \text{(D.5)}$$

Noting that $\left|\frac{2}{|S|}\sum_i f(x_i)\sigma_i\right| < 2$, and setting $\alpha = \sqrt{8\ln(4|\mathcal{F}|) + 4\ln|S|}$ we can now bound:

$$\begin{aligned}
\hat{\mathcal{R}}_S(\mathcal{F}) &= \mathbf{E}_\sigma\left[\sup_{f\in\mathcal{F}}\left|\frac{2}{|S|}\sum_i \sigma_i f(x_i)\right|\right] \\
&\leq \frac{\alpha}{\sqrt{|S|}} + 4Ne^{-\alpha^2/8} \\
&= \frac{\sqrt{8\ln(4|\mathcal{F}|) + 4\ln|S|} + 1}{\sqrt{n}} \\
&\leq \sqrt{7\frac{2\log|\mathcal{F}| + \log|S|}{|S|}} \qquad \text{(D.6)}
\end{aligned}$$

for $|\mathcal{F}|, |S| > 2$. □

Appendix E

The Expected Spectral Norm of A Random Matrix

Many results are known about the asymptotic distribution of the spectral norm of a random matrix, where the entries in the matrix are i.i.d. Gaussians. Seginer [63] provides a bound on the expectation for finite size matrices, for matrices with i.i.d., but not necessarily Gaussian entries, and for matrices with independent sign-flips, but not necessarily identical magnitudes.

Theorem 48 ([63, Corollary 2.2]). *There exists a constant K such that, for any m, m, any $h \leq 2 \ln \max(m, n)$ and any $m \times n$ random matrix $A = (a_{ij})$, where a_{ij} are i.i.d. zero mean random variables, the following inequality holds:*

$$\max \left\{ \mathbf{E}\left[\max_i |a_{i\cdot}|^h\right], \mathbf{E}\left[\max_j |a_{\cdot j}|^h\right] \right\} \leq \mathbf{E}\left[\|A\|_2^h\right] \leq K^h \left(\mathbf{E}\left[\max_i |a_{i\cdot}|^h\right] + \mathbf{E}\left[\max_j |a_{\cdot j}|^h\right] \right)$$

where $a_{i\cdot}$ is a row of A and $a_{\cdot j}$ is a column of A.

Theorem 49 ([63, Theorem 3.1]). *There exists a constant K such that, for any n, m, any $h \leq 2 \ln \max(m, n)$, and any $m \times n$ deterministic matrix $A = (a_{ij})$, the following* *inequality holds:*

$$\mathbf{E}_\epsilon\left[\|\epsilon \otimes A\|_2^h\right] \leq \left(K \ln^{1/4} \min(m,n)\right)^h \left(\max_i |a_{i\cdot}|^h + \max_j |a_{\cdot j}|^h\right)$$

where ϵ *is an i.i.d. sign matrix with* $\Pr(\epsilon_{ij} = 1) = \Pr(\epsilon_{ij} = -1) = \frac{1}{2}$, *the operation* $\otimes$ *denotes an element-wise product, and* $a_{i\cdot}$ *and* $a_{\cdot j}$ *are row and column vectors of* A.

Appendix F

Grothendiek's Inequality

We quote here Grothendiek's inequality, and a corollary of the inequality that we use in the Thesis.

Theorem 50 ([79]). *For any $A \in \mathbb{R}^{n \times m}$:*

$$max_{X=uv',|u|_\infty=|v|_\infty=1} A \bullet X \geq k_R \max_{X,\|X\|_{\max}\leq 1} A \bullet X$$

where k_R is Grothendiek's constant, and:

$$1.67 \leq k_R \leq 1.79$$

Let

$$B_{\max} = \{X| \, \|X\|_{\max} \leq 1\}$$

denote the unit ball of the max-norm,

$$C_{\inf} = \{uv'| \, |u|_\infty = |v|_\infty = 1\}$$

denote rank-one matrices with unit-bounded entries and

$$C_\pm = \{uv'|u \in \{-1,+1\}^n, v \in \{-1,+1\}^m\} = \{X \in \{-1,+1\} | \operatorname{rank} X = 1\}$$

denote rank-one sign matrices.

Corollary 51.

$$\mathrm{conv} C_{\pm} = \mathrm{conv} C_{\mathrm{inf}} \subset B_{\max} \subset 2\mathrm{conv} C_{\pm}$$

Bibliography

[1] N. Alon, P Frankl, and V. Rödel. Geometrical realization of set systems and probabilistic communication complexity. In *Proceedings of the 26th Annual Symposium on the Foundations of Computer Science (FOCS)*, pages 227–280, 1985.

[2] Noga Alon. The number of polytopes, configurations and real matroids. *Mathematika*, 33:62–71, 1986.

[3] Noga Alon. Tools from higher algebra. In M. Grötschel R.L. Graham and L. Lovász, editors, *Handbook of Combinatorics*, chapter 32, pages 1749–1783. North Holland, 1995.

[4] Orly Alter, Patrick O. Brown, and David Botstein. Singular value decomposition for genome-wide expression data processing and modeling. *PNAS*, 97(18):10101–10106, 2000.

[5] T. W. Anderson and Herman Rubin. Statistical inference in factor analysis. In *Third Berleley Symposium on Mathematical Statistics and Probability*, volume V, pages 111–150, 1956.

[6] D. F. Andrews and C. L. Mallows. Scale mixtures of normal distributions. *Journal of the Royal Statistical Society. Series B (Methodological)*, 36(1):99–102, 1974.

[7] Martin Anthony and Peter L. Bartlett. *Neural Networks Learning: Theoretical Foundations*. Cambridge University Press, 1999.

[8] Yossi Azar, Amos Fiat, Anna R. Karlin, Frank McSherry, and Jared Saia. Spectral analysis of data. In *ACM Symposium on Theory of Computing*, pages 619–626, 2001.

[9] John Barnett. Convex matrix factorization for gene expression analysis. Master's thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, February 2004.

[10] Peter L. Bartlett, Michael I. Jordan, and Jon D. McAuliffe. Large margin classifiers: Convex loss, low noise, and convergence rates. In *Advances in Neural Information Processing Systems*, volume 16. MIT Press, 2004.

[11] Peter L. Bartlett and Shahar Mendelson. Rademacher and Gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3:463–482, 2002.

[12] M. W. Berry. Large-scale sparse singular value computations. *The International Journal of Supercomputer Applications*, 6(1):13–49, Spring 1992.

[13] Daniel Billsus and Michael J. Pazzani. Learning collaborative information filters. In *Proc. 15th International Conf. on Machine Learning*, pages 46–54. Morgan Kaufmann, San Francisco, CA, 1998.

[14] A. Bj orner, M. Las Vergnas, B. Strumfels, N. White, and G. Ziegler, editors. *Oriented Matroids*. Cambridge University Press, 2nd edition edition, 1999.

[15] D. Blei, A. Ng, and M. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[16] B. Borchers. CSDP, a C library for semidefinite programming. *Optimization Methods and Software*, 11(1):613–623, 1999.

[17] Matthew Brand. Incremental singular value decomposition of uncertain data with missing values. In *European Conference on Computer Vision*, 2002.

[18] Wray Buntine. Variational extensions to em and multinomial pca. In *Proceedings of the European Conference on Machine Learning*, volume 2430 of *Lecture Notes in Artificial Intelligence*. Springer-Verlag, 2002.

[19] M. Collins, S. Dasgupta, and R. Schapire. A generalization of principal component analysis to the exponential family. In *NIPS*, 2001.

[20] K. Crammer and Y. Singer. Pranking with ranking. In *NIPS*14*, 2002.

[21] Sanjoy Dasgupta, Wee Sun Lee, and Philip M. Long. A theoretical analysis of query selection for collaborative filtering. *Machine Learning*, 51(3):283–298, 2003.

[22] S. Deerwester, G. W. Dumias, S. R. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Sience*, 41:391–407, 1990.

[23] Ofer Dekel, Christopher Manning, and Yoram Singer. Log-linear models for label ranking. In *Advances in Neural Information Processing Systems*, volume 16. MIT Press, 2004.

[24] Luc Devroye, Lászl6 Gy ofri, and Gábor Lugosi. *A Probabilistic Theory of Pattern Recongnition*. Springer, 1996.

[25] Petros Drineas, Iordanis Kerenidis, and Prabhakar Raghavan. Competitive recommendation systems. In *ACM Symposium on Theory of Computing*, 2002.

[26] Maryam Fazel, Haitham Hindi, and Stephen P. Boyd. A rank minimization heuristic with application to minimum order system approximation. In *Proceedings American Control Conference*, volume 6, 2001.

[27] Limin Fu and Douglas G. Simpson. Conditional risk models for ordinal response data: simultaneous logistic regression analysis and generalized score tests. *Journal of Statistical Planning and Inference*, 108(1-2):201–217, Nov 2002.

[28] Amir Globerson and Naftali Tishby. Sufficient dimensionality reduction. *Journal of Macchine Learning Research*, 3:1307–1331, 2003.

[29] Ken Goldberg, Theresa Roeder, Dhruv Gupta, and Chris Perkins. Eigentaste: A constant time collaborative filtering algorithm. *Information Retrieval*, 4(2):133–151, 2001.

[30] Jacob Goodman and Richard Pollack. Upper bounds for configurations and polytopes in $\mathbb{R}^d$. *Discrete and Computational Geometry*, 1:219–227, 1986.

[31] Geoff Gordon. Generalized^2 linear^2 models. In *NIPS*2002*, 2002.

[32] Alan Gous. *Exponential and Spherical Subfamily Models.* PhD thesis, Stanford University, 1998.

[33] David Haussler. Decision theoretic generalizations of the PAC model for neural net and other learning applications. *Information and Computation*, 100(1):78–150, 1992.

[34] David Haussler. Sphere packing numbers for subsets of the boolean n-cube with bounded Vapnick-Chernovenkis dimension. *Jounral or Combinatorial Thoery, Series A*, 69(2):217–232, 1995.

[35] R. Herbrich, T. Graepel, and K. Obermayer. Support vector learning for ordinal regression. In *Proceedings of 9th International Conference on Artificial Neural Networks: ICANN '99, 7-10 Sept. 1999*, pages 97–102, 1999.

[36] R Herbrich, T Graepel, and K Obermayer. *Advances in Large Margin Classifiers*, chapter Large Margin Rank Boundaries for Ordinal Regression, pages 115–132. MIT Press, 2000.

[37] T. Hofmann, J. Puzicha, and M. Jordan. Unsupervised learning from dyadic data. In *Advances in Neural Information Processing Systemts*, volume 11, 1999.

[38] Thomas Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning Journal*, 42(1):177–196, 2001.

[39] Thomas Hofmann. Latent semantic models for collaborative filtering. *ACM Trans. Inf. Syst.*, 22(1):89–115, 2004.

[40] Roger A. Horn and Charles R. Johnson. *Topics in Matrix Analysis.* Cambridge University Press, 1991.

[41] Michal Irani and P Anandan. Factorization with uncertainty. In *European Conference on Computer Vision*, June 2000.

[42] Tommi Jaakkola and Michael Jordan. Bayesian parameter estimation via variational methods. *Statistics and Computing*, 10:25–37, 2000.

[43] G. J. O. Jameson. *Summing and nuclear norms in Banach space theory.* Cambridge University Press, 1987.

[44] I. T. Jolliffe. *Principal Component Analysis.* Springer, 1986.

[45] Gert R.G. Lanckriet, Nello Cristianini, Peter Bartlett, Laurent El Ghaoui, and Michael I. Jordan. Learning the kernel matrix with semidefinite programming. *Journal of Machine Learning Research*, 5:27–72, 2004.

[46] Daniel D. Lee and H. Sebastian Seung. Unsupervised learning by convex and conic coding. In *Advances in Neural Information Processing Systems*, volume 9, pages 515–521, 1997.

[47] Daniel D. Lee and H. Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401:788–791, 1999.

[48] Daniel D. Lee and H. Sebastian Seung. Algorithms for non-negative matrix factorization. In *Advances in Neural Information Processing Systems*, volume 13, pages 556–562, 2001.

[49] J Löfberg. *YALMIP 3*, 2004.

[50] W.-S. Lu, S.-C. Pei, and P.-H. Wang. Weighted low-rank approximation of general complex matrices and its application in the design of 2-D digital filters. *IEEE Transactions on Circuits and Systems—I*, 44(7):650–655, July 1997.

[51] B. Marlin. Collaborative filtering: A machine learning perspective. Master's thesis, University of Toronto, 2004.

[52] Benjamin Marlin. Modeling user rating profiles for collaborative filtering. In *NIPS*17*, 2004.

[53] Benjamin Marlin and Richard S. Zemel. The multiple multiplicative factor model for collaborative filtering. In *To appear in ICML*, 2004.

[54] Peter McCullagh. Regression models for ordinal data. *Journal of the Royal Statistical Society. Series B (Methodological)*, 42(2):109–142, 1980.

[55] F. Mosteller, R. E. K. Rourke, and G. B. Thomas. *Probability and Statistics*. Addison-Wesley, 1961.

[56] Dmitry Panchenko and Vladimir Koltchinskii. Empirical margin distributions and bounding the generalization error of combined classifiers. *Annals of Statistics*, 30(1), 2002.

[57] R. Paturi and J. Simon. Probabilistic communication complexity. In *Proceedings of the 25th Annual Symposium on the Foundations of Computer Science (FOCS)*, pages 118–126, 1984.

[58] K. Pearson. On lines and planes of closets fit to systems of points in space. *The London, Edinburgh and Dublin Philosophical Magazine and Journal of Science*, 2:559–572, 1901.

[59] Sam T. Roweis and Lawrence K. Saul. Nonlinear Dimensionality Reduction by Locally Linear Embedding. *Science*, 290(5500):2323–2326, 2000.

[60] A. Ruhe. Numerical computation of principal components when seveal observations are missing. Technical Report UMINF-48-74, Dept. Information Processing, Umea University, Sweden, 1974.

[61] B. M. Sarwar, G. Karypis, J. A. Konstan, and J. Riedl. Application of dimensionality reduction in recommender system–a case study. In *ACM WebKDD 2000 Web Mining for E-Commerce Workshop*, 2000.

[62] Andrew I. Schein, Lawrence K. Saul, and Lyle H. Ungar. A generalized linear model for principal component analysis of binary data. In *Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics*, 2003.

[63] Yoav Seginer. The expected norm of random matrices. *Comb. Probab. Comput.*, 9(2):149–166, 2000.

[64] Amnon Shashua and Anat Levin. Ranking with large margin principle: Two approaches. In *NIPS*14*, 2003.

[65] John Shawe-Taylor, Nello Cristianini, and Jaz Kandola. On the concentration of spectral properties. In *NIPS*15*, 2002.

[66] Dale Shpak. A weighted-least-squares matrix decomposition method with application to the design of two-dimensional digital filters. In *IEEE Midwest Symposium Circuits Systems*, pages 1070–1073, Calgary, AB, Canada, August 1990.

[67] Heung-Yeung Shum, Karsushi Ikeuchi, and Raj Reddy. Principal component analysis with missing data and its application to plyhedral object modeling. *IEEE transactions on pattern analysis and machine inteligence*, 17(9), 1995.

[68] Nathan Srebro. Counting sign configurations of low-rank matrices. `http://www.csail.mit.edu/people/nati/mmmf`, April 2004.

[69] Nathan Srebro and Tommi Jaakkola. Weighted low rank approximation. In *20th International Conference on Machine Learning*, 2003.

[70] Nathan Srebro and Tommi Jaakkola. Linear dependent dimensionality reduction. In *Advances in Neural Information Processing Systems*, volume 16. MIT Press, 2004.

[71] G. W. Stewart and Ji-guang Sun. *Matrix Perturbation Theory*. Academic Press, Inc, 1990.

[72] Joshua B. Tenenbaum and William T. Freeman. Separating style and content with bilinear models. *Neural Computation*, 12(6):1247–1283, 2000.

[73] Joshua B. Tenenbaum, Vin de Silva, and John C. Langford. A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science*, 290(5500):2319–2323, 2000.

[74] M. E. Tipping and C. M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society, Series B*, 21(3):611–622, 1999.

[75] Olga Troyanskaya, Michael Cantor, Gavin Sherlock, Pat Brown, Trevor Hastie, Robert Tibshirani, David Botstein, and Russ B. Altman. Missing value estimation methods for dna microarrays. *Bioinformatics*, 17(6):520–525, 2001.

[76] A. W. van der Vaart. *Asymptotic Statistics*. Cambridge University Press, 1998.

[77] M J Wainwright and E P Simoncelli. Scale mixtures of Gaussians and the statistics of natural images. In S. A. Solla, T. K. Leen, and K.-R. Müller, editors, *Adv. Neural Information Processing Systems*, volume 12, pages 855–861, Cambridge, MA, 2000. MIT Press.

[78] H. E. Warren. Lower bounds for approximation by nonlinear manifolds. *Transactions of the American Mathematical Society*, 133:167–178, 1968.

[79] Eric W. Weisstein. Grothendieck's constant. Web Resource, `http://mathworld.wolfram.com/GrothendiecksConstant.html`.

[80] Eric W. Weisstein. Random walk–1-dimensional. Web Resource, `http://mathworld.wolfram.com/RandomWalk1-Dimensional.html`.

[81] T. Wibger. Computation of principal components when data are missing. In *Proceedings of the Second Symposium on Computational Statistics*, pages 229–236, 1976.

[82] Gale Young. Maximum likelihood estimation and factor analysis. *Psychometrika*, 6(1):49–53, 1940.