GLOBAL OPTIMALITY IN REPRESENTATION LEARNING

by

Benjamin D. Haeffele

A dissertation submitted to The Johns Hopkins University in conformity with the requirements for the degree of Doctor of Philosophy.

Baltimore, Maryland

November, 2015

Abstract

A majority of data processing techniques across a wide range of technical disciplines require a representation of the data that is meaningful for the task at hand in order to succeed. In some cases one has enough prior knowledge about the problem that a fixed transformation of the data or set of features can be pre-calculated, but for most challenging problems with high dimensional data, it is often not known what representation of the data would give the best performance. To address this issue, the field of representation learning seeks to learn meaningful representations directly from data and includes methods such as matrix factorization, tensor factorization, and neural networks. Such techniques have achieved considerable empirical success in many fields, but common to a vast majority of these approaches are the significant disadvantages that 1) the associated optimization problems are typically non-convex due to a multilinear form or other convexity destroying transformation and 2) one is forced to specify the size of the learned representation a priori.

This thesis presents a very general framework which allows for the mathematical analysis of a wide range of non-convex representation learning problems. The framework allows the derivation of sufficient conditions to guarantee that a local minimizer of the non-convex optimization problem is a global minimizer and that from any initialization it is possible to find a global minimizer using a purely local descent algorithm. Further, the framework also allows for a wide range of regularization to be incorporated into the model to capture known features of data and to adaptively fit the size of the learned representation to the data instead of defining it a priori. Multiple implications of this work are discussed as they relate to modern practices in deep learning, and the advantages of the approach are demonstrated in applications of automated spatio-temporal segmentation of neural calcium imaging data and reconstructing hyperspectral image volumes from compressed measurements.

Primary Reader: Dr. René Vidal

Secondary Reader: Dr. Daniel Robinson

Acknowledgments

My first week of grad school I was buying furniture off of Craigslist for my new apartment, and by chance I ended up getting a pair of chairs from another Hopkins student who had just finished her Ph.D. and was getting ready to move. During the small-talk as I was loading chairs into my car she gave me the usual "a Ph.D. takes a lot of work, but you can do it" encouragement, but the weary look on her face made it clear through her attempt at a smile that I had no idea what "a lot" really meant. Over the years, one quickly learns that science does not progress in a smooth and orderly fashion. Experiments fail; equipment breaks; ideas don't pan out after months of work; and grad students get left in the wake asking serious questions about their life choices. A lot goes into a thesis, but it takes much more than work. For this thesis, it has taken the support of many individuals over the years, a few of which I have remembered to properly thank below, but to those who I have inevitably forgotten, thank you.

First, I have had the privilege of working with two fantastic advisers, Eric and René. When you tell other grad students around Hopkins that you're working in

Eric's lab, people start talking in reverent tones and recount stories of some brilliant insight Eric had given them about the project they were working on. A nickel for each time this happened may not have made me rich, but it would have at least gotten me a cappuccino or something. This reputation is well deserved, and I could have not asked for a better adviser. My Ph.D. finished with work that was a long way from what I started out doing, and I am extremely grateful for all the support and guidance over the years as I followed my academic whims.

It was almost by accident that I started working with René. I was assigned as a teaching assistant in Rene's control theory course, and an offhand question one day about some algorithms I was using to analyze my data grew into the work that makes up the bulk of this thesis. Thank you for devoting the time to work with me in the early days, the ever insightful comments, the unwavering support, and for ultimately becoming my adviser as my work evolved over the years. Your high standards for rigor and ability to communicate the essence of complicated ideas have undoubtedly made me a better scientist.

Thank you, Daniel, for agreeing to join my committee late in the game. I regret that we didn't get more time to work together, but I will always appreciate you patiently listening to me as I rambled through some poorly presented proof or half thought out idea.

There have been many other grad students who have helped me over the years and shared in my struggles and successes. Thank you Chong and Manolis for the hours of

interesting discussions, as well as John and Manu for struggling and commiserating with me through broken pressure injectors, months of failed experiments, and the infamous [unnamed professor] lab space incident during the dark ages of the calcium imaging project. To the rest of the BME crew, Aggy, Alex, Ben L, Connie, Lei, Lukas, Nupura, Rishab, Yoonju, and the others that I'm forgetting, your friendship has been invaluable.

Finally, I thank my family for being my greatest source of support. To Mom and Dad, I would never be where I am today without you. I of course have my flaws, but I cannot blame any of them on the job you did raising me. I will never be able to thank you enough for your unconditional love and support over the years. To Jon, thank you for always being there whenever I've needed a friend and that sense of humor that only gets appreciated by family. You've turned out to be a wonderful brother despite my best efforts to traumatize you as a small child. To Juleen, you have been my constant through all of this. You brightened me on the dark days, kept me sane on the crazy ones, and cheered me through the sad ones. You kept me grounded in the world outside of neurons and theorems and explored that world with me. I might have finished this thesis without you, but it's unclear.

Dedication

To David T. Yue, who passed away before this thesis could be completed.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

A large majority of modern data processing techniques rely on finding or having access to a meaningful representation of data in order to make sense of complicated, high-dimensional datasets. For tasks such as data exploration, finding an easily interpretable and problem-meaningful representation of the data is itself the end-goal, while more focused tasks, like classifying entries in a dataset into an appropriate class or removing noise and corruptions from the data, critically depending on the data representation being suitable for the particular task at hand to ensure success of the method. In some sense, this idea of needing a useful representation of a dataset forms a founding principle in many technical disciplines. For example, in classical signal processing a fundamental concept is the notion that signals can be simultaneously represented as either a sequence of values or as the coefficients of a weighted summation of sinusoids obtained via the Fourier transform. When designing a filter to extract signals with a known frequency spectrum, one derives little insight from studying a sequence of signal values as a function of time, while similarly one would be hard pressed to identify if a given image was a dog or a tiger after being shown an image of Fourier coefficients. Of course, the above example is somewhat contrived as the Fourier transform provides a simple means to switch between the two representations, but it speaks to the importance of choosing a representation of the data appropriate for the problem one wishes to solve.

To make this idea more concrete, consider the problem of classification using linear classifiers. The top panel of figure 1.1 shows a collection of 2-dimensional data points from 3 different classes (red, green, and blue) which are arranged in concentric rings (left). If we only have access to linear classifiers to separate the 3 classes, one would have to find many linear decision boundaries (to approximate circular decision boundaries) to separate the 3 circles using the raw data, but by simply converting the data to polar coordinates (right) it is easy to separate the classes with 2 linear decision boundaries. Certainly in this example it is relatively trivial to recognize that a suitable data representation is to convert the data into polar coordinates, but for more challenging classification tasks, such as classifying an image as being either a dog or a tiger (bottom panel of figure 1.1), it is highly non-trivial to find a transformation of the data that allows the images to be linearly separated.

Historically, a great deal of work has approached such problems by constructing and analyzing "hand-designed" data representations, where one fixes *a priori* a

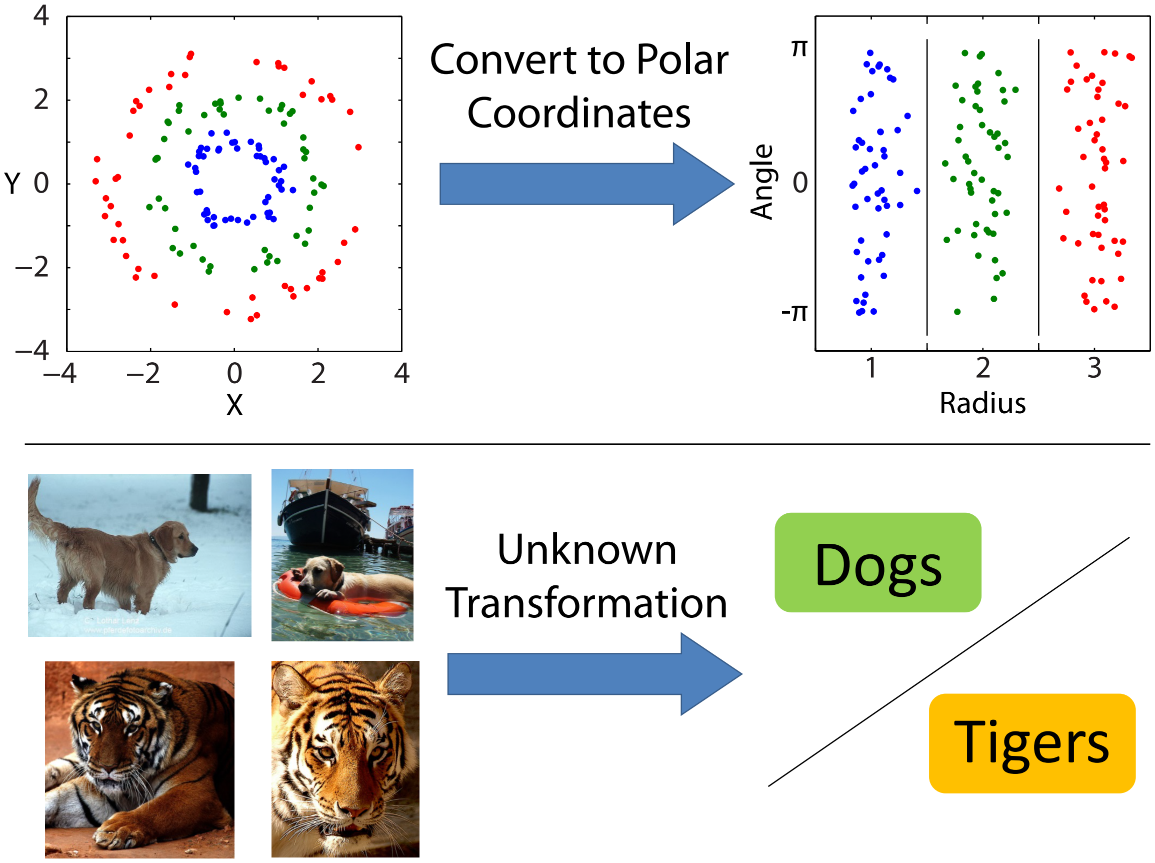

Figure 1.1: Classification with Linear Classifiers *Top Panel:* Hypothetical 2-dimensional data points from 3 classes (red, blue, and green) in Cartesian (left) and polar coordinates (right). In polar coordinates the 3 classes are easily separated via linear classifiers. *Bottom Panel:* Example images from the dog and tiger classes of the ImageNet database (left). Separation of dog and tiger classes after applying an ideal transformation of the data (right).

transformation of the data or calculates a predetermined set of features from the data to use in subsequent analysis. Transforming a signal from the time domain to the frequency domain via a Fourier transform is one such example, as are more modern signal processing techniques like representing the signal via a collection of wavelets coefficients [1]. Further, in specific applications one can similarly find a development of specialized problem-specific transformations. For example, in image classification one finds techniques and hand-crafted features such as morphological image processing [2], local binary patterns (LBP) [3], histogram of oriented gradients (HOG) [4], or the scale-invariant feature transforms (SIFT) [5]. While in some cases there is a strong theoretical justification for choosing a particular data representation based on known information about the problem, such as a physical model that describes how the data is generated, often times the practitioner is left with little guidance as to what features or transformations would be most advantageous for a particular problem and is forced to perform trial and error to choose the best representation.

Due to the somewhat arbitrary choice of a hand-designed data representation, an alternative approach is to instead try to find a relevant representation for the data which is tailored to the task at hand by learning a representation directly from the dataset. While this idea sounds attractive in principle, there are many challenges that arise in practice. The first is simply the fact that attempting to learn representations directly from the data greatly increases the scope of the problem we are trying to solve. For example, in an image classification task, learning a few parameters for a classifier that operates on a predefined set of features typically requires significantly fewer parameters than learning a full transformation of the input image, which must then be fed into a classifier (whose parameters also have to be learned). As a result of this significantly increased problem scope, great care must be taken to ensure that what one is learning is not simply "over-fitting" to noise in the dataset. This requires either 1) a very large amount of data (which is becoming more feasible in the era of "big data"), 2) careful mathematical modeling of the problem to ensure that the representations we learn are "good" in some sense, or 3) some combination of the above two points. Further, beyond the greatly increased scope of the problem and increased risk of overfitting, another significant challenge to learning representations directly from the data is the fact that typically one is required to solve a very challenging optimization problem whose solution cannot be found in polynomial time. However, despite these numerous practical challenges, learning representations directly from the data remains a very powerful concept and has achieved very significant empirical success in multiple real-world applications.

1.1 Relevance to Biology

To motivate the relevance of these general ideas in biology, consider the problem of processing neural calcium imaging data. Calcium imaging is a recently developed biological technique that records changes in the intracellular calcium level of individual cells through the use of either synthetic or genetically encoded calcium sensitive fluorescent molecules. The most common application of calcium imaging is in electrically excitable cells, such as cardiac myocytes or neurons, where the large changes in intracellular calcium that occur during a depolarizing action potential can induce large changes in the fluorescent signal which can be observed by recording videos of the cells with a fluorescent microscope [6]. Given a movie containing the fluorescent signals recorded from a population of electrically excitable cells (which from here on we will assume are neurons), the overall goal when processing a calcium signal movie is to recover 3 pieces of information. First, we simply need to estimate how many active neurons are in the movie. Then, for each active neuron, we wish to estimate both the temporal fluorescent signal of the neuron (and in particular when the action potentials occur) as well as a segmentation of that neuron in space. This is depicted in the top panel of Figure 1.2, where for a given neuron, we wish to estimate the temporal signal, which can be completely defined by the action potential times (red dots) and an (assumed known) model of the calcium dynamics given the action potential times, as well as the spatial segmentation of the neuron. The bottom panel of figure 1.2 then depicts the estimated temporal signals and spatial segmentations for all 10 neurons in the movie.

The problem of estimating these properties from a given calcium image movie highlights many of the common challenges one encounters in learning representations directly from data. First, the size of the representation has to be estimated somehow,

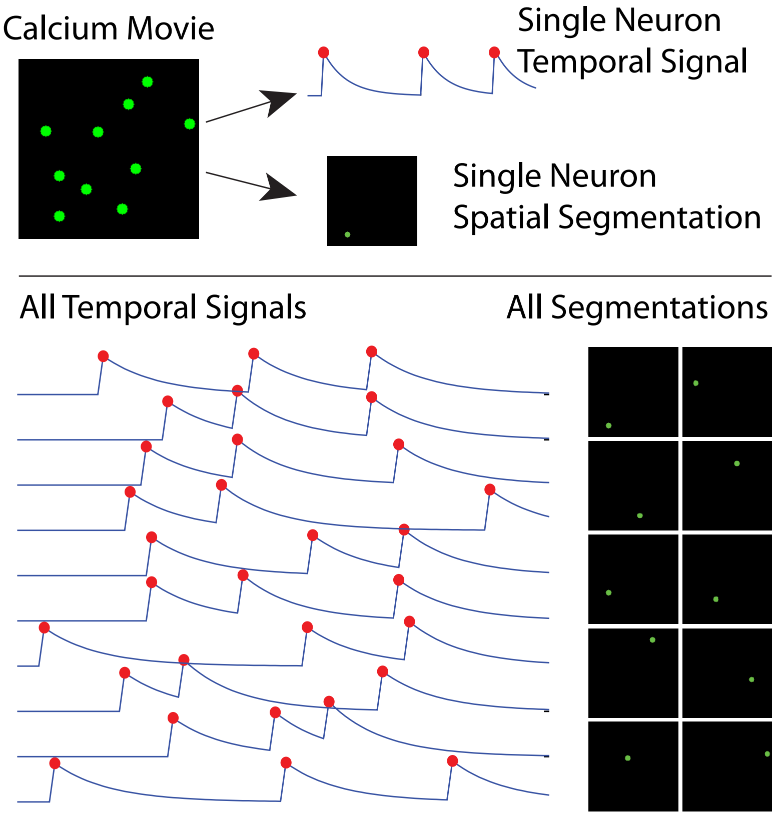
Calcium Movie
Single Neuron Temporal Signal
Single Neuron Spatial Segmentation
All Temporal Signals
All Segmentations

Figure 1.2: Cartoon depiction of a calcium imaging dataset. *Top Panel:* Estimated temporal signal (with action potential times shown as red dots) and spatial segmentation of a signal neuron from the calcium imaging movie. *Bottom Panel:* Estimated temporal signals and spatial segmentations of all 10 neurons in the dataset.

which in this case corresponds to estimating the number of active neurons in the data. However, even if one knows the number of active neurons in the data, it is still unclear how to estimate both the temporal signals and the spatial segmentations simultaneously. If we are provided with the spatial segmentations of all of the neurons, then it is straight-forward to estimate the temporal signals by averaging the pixels within the segmentation of a neuron at each frame of the video; then from the temporal signal there are established techniques to estimate the action potential times [7]. Likewise, if we were given the temporal signals, one could envision a method to estimate the spatial segmentations by looking for pixels with strong correlations with the provided temporal signal. When both components of the representation are unknown, however, the problem becomes significantly more challenging. This "chicken-or-the-egg" problem is not unique to calcium imaging but rather encapsulates a fundamental challenge in almost every technique that seeks to learn meaningful representations directly from the data. For example, in the case of classifying images, if one wishes to learn discriminative features directly from the data, one must choose the number of features to learn, while simultaneously learning parameters that describe both the features and the classifier (which takes as input the learned features).

1.2 Thesis Contributions

At a high level, this thesis will be largely devoted to addressing the challenges associated with learning representations directly from high-dimensional data. In particular, it will focus on how one can model representation learning problems in a way that allows known prior information to be incorporated into the model to combat the effects of over-fitting while also providing a general mathematical framework to ensure that the associated mathematical optimization problem has nice properties that make it conducive to being solved efficiently. While representation learning techniques have achieved great empirical success in many application areas, the examples above illustrate that in practice there are still a few fundamental challenges one faces when trying to learn representations directly from the data. The first issue is that one is must typically select the size of the learned representation *a priori*. Choosing too small of a representation will prevent the model from having a sufficient degree of expressive power to accomplish the task at hand, while choosing too large of a representation leads to over-fitting the representation to noise in the data. A second major issue that is common to the vast majority of representation learning techniques is the fact that the associated optimization problems are non-convex (see section 2.2). As a result, one is typically only able to find an approximate solution to the proposed optimization problem, and the approximate solution that is found will heavily depend on the particular choice of initialization one uses and specific details of how one attempts to solve the optimization problem.

To address these issues, this thesis will focus largely on how carefully constructed regularization can be used to improve representation learning techniques. At a high level, this will afford two major advantages. First, by correct construction of the regularization function in the representation learning formulations, it will possible to derive techniques that effectively fit the size of the learned representation to the data directly, as opposed to forcing one to choose a representation size *a priori*. Second, although the overall problem of learning data representations is typically a non-convex optimization problem, under the conditions of the mathematical framework developed here, it is possible to provide sufficient conditions to guarantee that local minima are globally optimal and that a global minimizer can always be found from local descent.

1.3 Thesis Outline

The remainder of this thesis is organized as follows. Chapter 2 introduces the necessary mathematical background material and notation. First, an overview of the basic principles of optimization is provided, which includes the various types of optimality, the benefits of convex optimization problems, and discussion of several forms of duality. Next, a review of common methods used for representation learning is discussed, such as matrix factorization, sparse/low-rank methods, tensor factorization, and both classical and modern developments of neural networks.

Chapter 3 will focus on the problem of solving structured matrix factorization problems. By using a particular form of regularization that allows for prior assumptions to be placed on the matrix factors, it is shown that a wide range of structured matrix factorization problems can be done within a framework that guarantees that local minima will be globally optimal and which fits the size of the learned matrix factorization to the data. A few practical algorithms are also provided to solve the matrix factorization problem, and bounds regarding how closely a given approximate solution of the optimization problem is to the global optimum are derived.

In chapter 4, these ideas are significantly extended and generalized to a wide range of representation learning problems. In particular, the framework is generalized to any mapping which is a positively homogeneous function of the factorized variables. This includes a wide variety of representation learning problems, such as tensor factorization and training deep neural networks. Results are again derived to guarantee that local minima of the non-convex factorization problem are globally optimal. Moreover, it is shown that if the size of the representation is initialized to be sufficiently large, then from any arbitrary initialization there must always exist a non-increasing path to a global minimizer (i.e., from any point one can follow a path to a global minimizer such that the objective function never increases along that path). Further, a meta-algorithm is constructed that allows one to use any local descent strategy to find a global minimizer.

Chapter 5 concludes the thesis by demonstrating the proposed structured matrix factorization techniques on real-world applications. First, a formulation is derived to do a spatio-temporal segmentation of neural calcium imaging data, which simultaneously estimates neural spike trains and spatial segmentations from a raw calcium imaging video. Results of the proposed method are provided for both phantom experiments and on real data taken from awake mice. Second, experiments are provided on the application of recovering hyperspectral images from a series of compressed measurements.

Chapter 2

Mathematical Preliminaries

Before proceeding further, basic relevant mathematical background material will be introduced, and the notational conventions used in this thesis will be defined.

2.1 Optimization Basics

The topics discussed in this thesis will largely revolve around solving optimization problems, which take the general form

$$\min_{X} f(X) \quad \text{s.t.} \quad X \in C, \tag{2.1}$$

where the set $C \subseteq \mathbb{R}^D$ will be referred to as the ***constraint set***, and the function $f : C \to \mathbb{R}$ will be referred to as the ***objective function***. If $C = \mathbb{R}^D$ then the optimization problem will be said to be ***unconstrained***, while if $C \subset \mathbb{R}^D$ the optimization problem will be said to be ***constrained***. Note that a constrained problem can always be converted to an unconstrained problem via the use of the indicator function, which is defined as follows.

Definition 1 *The **indicator function of a set** C is defined as*

$$\delta_C(x) = \begin{cases} 0 & x \in C \\ \infty & x \notin C. \end{cases} \tag{2.2}$$

Given this definition, the constrained problem in (2.1) can be rewritten in unconstrained form as

$$\min_X f(X) + \delta_C(X). \tag{2.3}$$

When solving optimization problems, one can discuss several different types of optimality. Outside of the formal optimization literature, the specific type of optimality being referenced by an author can sometimes be somewhat ill-defined, with various authors using differing concepts of things such as "local minima". To ensure clarity, the definitions assumed by this thesis are formalized below, with the first, and most straightforward, being global optimality, which is defined as follows:

Definition 2 *A point X_{opt} is said to be a **global minimizer** of the optimization problem given in* (2.1) *if $X_{opt} \in C$ and $\forall Z \in C$ we have $f(X_{opt}) \leq f(Z)$.*

A weaker notion of optimality is that of local optimality.

Definition 3 *A point X_{local} is said to be a **local minimizer** of the optimization* *problem given in* (2.1) *if* $X_{local} \in C$ *and* $\exists \epsilon > 0$ *such that* $\forall Z \in C \cap \{X' : \|X' - X\| \leq \epsilon\}$ *we have* $f(X_{local}) \leq f(Z)$.

From the definitions, it is clear that local optimality is weaker than global optimality and all global minimizers must also be local minimizers. A still weaker form of optimality is the concept of first order optimality, but before discussing this concept, one must first introduce the notion of a subgradient.

Definition 4 *Given a function* $f : \mathbb{R}^D \to \mathbb{R}$, $Z' \in \mathbb{R}^D$ *is said to be a* ***regular subgradient*** *of* f *at* X, *notated as* $Z' \in \hat{\partial} f(X)$, *if*

$$\liminf_{X' \to X : X' \neq X} \frac{f(X') - f(X) - \langle Z', X' - X \rangle}{\|X' - X\|} \geq 0. \tag{2.4}$$

Further, $Z \in \mathbb{R}^D$ *is said to be a* ***general subgradient*** *of* f *at* X, *notated as* $Z \in \partial f(X)$, *if there exists sequences* (X^k, Z^k) *such that* $X^k \to X$, $f(X^k) \to f(X)$, *and* $Z^k \in \hat{\partial} f(X^k) \to Z$.

This thesis will only very briefly rely on this general form of the subgradient, which is mentioned primarily for completeness, and instead a significantly simplified notion of a subgradient will be discussed below in the context of convex functions. Interested readers can find an extremely detailed analysis of subgradients in [8], but it is worth mentioning the basic facts that in general $\hat{\partial} f(X) \subseteq \partial f(X)$ and if $f(X)$ is differentiable at X with gradient $\nabla f(X)$, then $\hat{\partial} f(X) = \partial f(X) = \nabla f(X)$.

Having introduced the notion of a subgradient, first order optimality can now be defined and shown to be a necessary condition for the other forms of optimality discussed above.

Definition 5 *Given a function $f : \mathbb{R}^D \to \mathbb{R}$, a point $\bar{X}$ is said to be a **critical point** of f, or equivalently $\bar{X}$ is said to satisfy the **first order optimality conditions**, if $0 \in \partial f(\bar{X})$.*

Theorem 1 *[8, Thm. 10.1] If a function $f : \mathbb{R}^D \to \mathbb{R}$ has a local minimum at X_{local}, then X_{local} must be a critical point of f.*

If f is a differentiable function, then the above theorem is simply Fermat's theorem from basic calculus, but the use of general subgradients allows for the theorem to be extended to an arbitrary function. Also, recall that with the use of indicator functions discussed above, this also allows one to consider first order optimality conditions for constrained optimization problems.

2.2 Convexity and Duality

A fundamental concept in optimization is the notion of convexity, which can be summarized by the following two definitions:

Definition 6 *A set $C \subseteq \mathbb{R}^D$ is a **convex set** if $\forall(X \in C, Z \in C)$ and $\forall\mu \in [0,1]$ we have $\mu X + (1-\mu)Z \in C$.*

Definition 7 *Given a convex set C, a function $f : C \to \mathbb{R}$ is said to be a* ***convex function*** *on the set C if $\forall (X \in C, Z \in C)$ and $\forall \mu \in [0,1]$ we have $f(\mu X + (1-\mu)Z) \le \mu f(X) + (1-\mu) f(Z)$.*

An optimization problem of form (2.1) is then said to be a ***convex optimization problem*** if C is a convex set and f is a convex function on C. In general, the difference between a convex and a non-convex optimization problem represents a major bifurcation in the field of optimization. While a full review of the benefits of convex optimization is of course beyond the scope of this thesis and can be found in any optimization text [9, 10], one of the major advantages afforded by convexity is that first order optimality conditions are sufficient to guarantee global optimality. In particular, one has the following result.

Proposition 1 *[8, Thm. 10.1 and Prop. 8.12] If $f : \mathbb{R}^D \to \mathbb{R}$ is a convex function, then*

$$\hat{\partial} f(X) = \partial f(X) = \{Z : f(X') \ge f(X) + \langle Z, X' - X \rangle\,, \quad \forall X' \in \mathbb{R}^D\} \tag{2.5}$$

and X_{opt} is a global minimizer of f if and only if $0 \in \partial f(X_{opt})$.

From the above discussion, the relations between various forms of optimality for convex and non-convex functions are succinctly described by the relation,

Non-Convex: Global minimizer $\implies$ Local minimizer $\implies$ Critical point

Convex: Global minimizer $\iff$ Local minimizer $\iff$ Critical point.

The equivalence between the three forms of optimality in convex optimization is a major advantage as it guarantees that one needs to only consider local information about the function to test if the point is globally optimal. For general non-convex optimization, this fails to be the case, and global optimality can only be assured by exhaustively exploring the entire search space, which requires performing a number of computations that potentially grows exponentially with the number of variables.

2.2.1 Fenchel Dual

A second major benefit provided by working with convex optimization problems is the notion of duality. There are multiple different notions of duality, but the first that will be discussed is that of Fenchel Duality, which is defined as follows.

Definition 8 *Given a function $f : \mathbb{R}^D \to \mathbb{R}$, the **Fenchel dual** (also sometimes referred to as the conjugate function) will be notated $f^* : \mathbb{R}^D \to \mathbb{R}$ and is defined as*

$$f^*(Z) \equiv \sup_X \langle Z, X \rangle - f(X). \tag{2.6}$$

The Fenchel dual has many interesting connections with convex functions, which are too numerous to describe here (and full details can be found in [8, 9]), but a few important facts of the Fenchel dual are described in the following theorem:

Theorem 2 *[8, Thm. 11.1 and Prop. 11.3] Given a function $f : \mathbb{R}^D \to \mathbb{R}$, then its Fenchel dual, f^*, and the dual of the dual, $(f^*)^*$, are convex functions; $(f^*)^* \leq f$; and $f(X) + f^*(Z) \geq \langle X, Z \rangle$ for all (X, Z). Further, if f is a convex function, then $(f^*)^* = f$ and the following holds:*

$$Z \in \partial f(X) \iff X \in \partial f^*(Z) \iff f(X) + f^*(Z) = \langle X, Z \rangle \, . \tag{2.7}$$

The above theorem provides several useful results that will be used in various forms in this thesis, with the first being the fact that equation (2.7) gives a convenient means to characterize the subgradients of certain convex functions. Second, the dual of the dual, $(f^*)^*$, often referred to as the ***convex envelope*** of the function f, can be informally defined as the convex function which most closely lower bounds f (see [8, Thm. 11.1] for the formal definition). As will be discussed below, replacing a non-convex function with its convex envelope has led to the development of many commonly used convex regularizers in sparse and low-rank methods.

2.2.2 Lagrangian Duality

Another notion that often arises in the context of optimization is that of ***Lagrangian duality***. Lagrangian duality is largely a special case of Fenchel duality, as many of the results in Lagrangian duality can be derived from the concepts of Fenchel duality [8,9]. Lagrangian duality is often used in the context of a constrained optimization problem with the form

$$\min_X f(X) \ \text{ s.t. } \begin{array}{l} f_i(X) \leq 0 \quad i \in \{1, \dots, m\} \\ \mathcal{A}(X) = b, \end{array} \tag{2.8}$$

where $\mathcal{A}$ denotes a linear operator, $b \in \mathbb{R}^d$ is a vector of constants, and the f_i functions are used to enforce various inequality constraints[1]. Given a problem in the above form, one can introduce additional ***Lagrange multiplier*** terms, $\Lambda \in \mathbb{R}^d$ and $\gamma \in \mathbb{R}^m$, to enforce the constraints from the original problem (referred to as the ***primal problem***) and define the ***Lagrangian function*** as

$$L(X, \Lambda, \gamma) = f(X) + \langle \Lambda, \mathcal{A}(X) - b \rangle + \sum_{i=1}^{m} \gamma_i f_i(X). \tag{2.9}$$

Given this form, the theory of Lagrange duality guarantees the following:

$$\sup_{\Lambda, \gamma \geq 0} \inf_X L(X, \Lambda, \gamma) \leq \inf_X \sup_{\Lambda, \gamma \geq 0} L(X, \Lambda, \gamma). \tag{2.10}$$

[1] Note that this is the most common form of Lagrangian duality, but more general forms of Lagrangian duality also exist [8].

Note that for any point X which does not satisfy the constraints from the primal problem, $\sup_{\Lambda,\gamma\geq 0} L(X,\Lambda,\gamma) = \infty$, while if X is a feasible point of the primal problem then $\sup_{\Lambda,\gamma\geq 0} L(X,\Lambda,\gamma) = f(X)$. As a result, the right-hand side of the inequality in (2.10) is equivalent to the primal problem. The left-hand side of (2.10) is referred to as the ***dual problem***, and the inequality comes from a property known as ***weak duality*** which holds for any choice of functions f and f_i, $i = 1,\ldots,m$. If f and f_i, $i = 1,\ldots,m$ are convex functions, then provided a few technical constraint qualifications are satisfied, the inequality in (2.10) becomes an equality and the problem is said to have ***strong duality*** [8, 9]. Strong duality thus provides an alternative means to approach a convex optimization problem, as it guarantees that one can equivalently solve either the primal or the dual problem. Additionally, strong duality provides a means of guaranteeing global optimality, as one can verify if the value of the primal problem equals the value of the dual problem. If the two are not equal, then the given solution is known to be non-optimal, and the difference between the primal and the dual values is referred to as the ***duality gap***.

2.2.3 Polar Duality and Gauges

The final form of duality that will be discussed is that of ***polar duality***. Polar duality is based on the notion of a polar set, which is defined as follows:

Definition 9 *Given any set $C \subset \mathbb{R}^D$ such that $0 \in C$, the **polar set of** C is notated* *as $C^\circ \subset \mathbb{R}^D$ and defined as*

$$C^\circ \equiv \{Z : \langle Z, X\rangle \leq 1 \ \forall X \in C\}. \tag{2.11}$$

Similar to the Fenchel dual function, the polar set will always be convex regardless of the choice of C, and if C is itself closed and convex then the polar of the polar will be the original set C. This is formalized by the following theorem.

Theorem 3 *[8, Ex. 11.19] Given any set $C \subset \mathbb{R}^D$ such that $0 \in C$, then $C^\circ \subset \mathbb{R}^D$ is a convex set with $0 \in C^\circ$. Further, if C is also closed and convex, then $(C^\circ)^\circ = C$.*

The later case of the above theorem, where C is a closed and convex set which contains the origin, is particularly useful in convex analysis as it allows one to define a gauge function.

Definition 10 *Given a closed, convex set $C \subset \mathbb{R}^D$ such that $0 \in C$, the* ***gauge function*** *on the set C, $\sigma_C : \mathbb{R}^D \to \mathbb{R}_+ \cup \infty$, is defined as*

$$\sigma_C(X) \equiv \inf\{\mu \geq 0 : X \in \mu C\}. \tag{2.12}$$

Note that gauge functions are generalizations of norms, and can be equivalently defined through the following result:

Theorem 4 *[11, Chap. 15] A function $\sigma : \mathbb{R}^D \to \mathbb{R}_+ \cup \infty$ is a gauge function if and only if*

1. *$\sigma(0) = 0$, and $\forall X \neq 0, \quad \sigma(X) > 0$.*

2. *$\forall (X,Y), \ \sigma(X+Y) \leq \sigma(X) + \sigma(Y)$.*

3. *$\forall \alpha \geq 0, \quad \sigma(\alpha X) = \alpha \sigma(X)$.*

To see how gauge functions are generalizations of norms, note that norms satisfy all of the above conditions, and for $C = \{Z : \|Z\| \leq 1\}$ then $\sigma_C(X) = \|X\|$. Further, for a gauge function to be a norm, it must also be invariant to negative scaling, $\sigma(-X) = \sigma(X)$, which can be assured if C is a symmetric set, i.e., $\forall X, \quad X \in C \iff -X \in C$.

Combining the concepts of a polar set, a gauge function, and Fenchel duality, one can now show many properties regarding the relationships between gauge functions, their subgradients, and polar sets. For notational purposes, we will notate the gauge function induced by the polar of set as σ_C° and refer to it as the ***polar function***,

$$\sigma_C^\circ(X) \equiv \sigma_{C^\circ}(X). \tag{2.13}$$

Having introduced the notation, we then have the following results.

Theorem 5 *[11] Given a closed, convex set $C \subset \mathbb{R}^D$ such that $0 \in C$, then one has the following relations*

1. *$\forall (X,Z) \quad \langle X, Z \rangle \leq \sigma_C(X)\sigma_C^\circ(Z)$.*

2. *$(\sigma_C)^*(Z) = \delta_{C^\circ}(Z)$ and $(\sigma_C^\circ)^*(X) = \delta_C(X)$.*

3. *$\partial \sigma_C(X) = \{Z : \langle X, Z \rangle = \sigma_C(X), \ \sigma_C^\circ(Z) \leq 1\}$.*

4. $\partial\sigma_C^\circ(Z) = \{X : \langle X, Z\rangle = \sigma_C^\circ(Z),\ \sigma_C(X) \leq 1\}$.

To provide some intuition for the above result, consider as an example the case of the l_q norms, defined for $q \in [1, \infty]$ as

$$\|x\|_q \equiv \left(\sum_{i=1}^{D} |x_i|^q\right)^{(1/q)}. \tag{2.14}$$

For a given l_q norm, its corresponding dual norm (or equivalently polar function) is the $l_{q'}$ norm ($\|x\|_q^\circ = \|x\|_{q'}$) where q and q' are related as $1/q + 1/q' = 1$, with the well known examples of this result being $(\|x\|_1)^\circ = \|x\|_\infty$, $(\|x\|_\infty)^\circ = \|x\|_1$, and $(\|x\|_2)^\circ = \|x\|_2$. Further, for the l_q norms, condition 1 of the above theorem gives the well known Hölder inequality. Likewise, condition 3 (and 4) gives well known results for the subgradients of norms, e.g., $\partial\|x\|_1 = \{z : \langle x, z\rangle = \|x\|_1,\ \|z\|_\infty \leq 1\} = SGN(x)$, where $SGN(x)$ is the point to set mapping

$$[SGN(x)]_i \equiv \begin{cases} 1 & x_i > 0 \\ -1 & x_i < 0 \\ [-1, 1] & x_i = 0. \end{cases} \tag{2.15}$$

More generally, the basic properties given in Theorem 5 form foundational principles in convex analysis and will be used multiple times during the development of the ideas in this thesis.

2.2.4 Proximal Operators

The final piece of optimization background material that will be necessary to introduce is the concept of a proximal operator, which is defined as follows:

Definition 11 *Given a closed, convex function g, the* ***proximal operator*** *of g is notated* $\mathbf{prox}_g$ *and defined as*

$$\mathbf{prox}_g(x) \equiv \arg\min_z \tfrac{1}{2}\|x - z\|_F^2 + g(z), \tag{2.16}$$

where $\|\cdot\|_F^2$ denotes the squared Frobenius norm, $\|z\|_F^2 = \sum_i z_i^2$. Further, the ***Moreau envelope*** *of g is notated $\mathcal{M}_g(x)$ and defined as*

$$\mathcal{M}_g(x) \equiv \tfrac{1}{2}\|x - \bar{z}_x\|_F^2 + g(\bar{z}_x), \tag{2.17}$$

where $\bar{z}_x = \mathbf{prox}_g(x)$.

Proximal operators arise in a variety of optimization algorithms but are most commonly used as a means to optimize non-differentiable objective functions. This is largely due to the facts that the Moreau envelope is continuous, with gradient equal to $\mathbf{prox}_g(x)$, and that the minimum of the Moreau envelope of g is equal to the minimum of g and the minima of the two functions are achieved at the same point [12]. As a result, if one needs to minimize a function which consists of a convex, differentiable function f and a convex, non-differentiable function g,

$$\min_x f(x) + g(x), \tag{2.18}$$

a common strategy is to update the variables x via the update equation

$$x^{k+1} = \mathbf{prox}_{\alpha g}(x^k - \alpha \nabla f(x^k)), \tag{2.19}$$

where $\alpha > 0$ is some step size parameter. The above iteration is typically referred to as ***proximal gradient descent*** and will converge to the global optimum of the objective function, provided α is chosen appropriately and f satisfies a few technical conditions [12]. Proximal gradient descent has many interpretations, but perhaps the most intuitive is that the $x^k - \alpha \nabla f(x^k)$ term is a simple gradient descent step on the differentiable part of the objective, f. Then, the proximal operator attempts to minimize g while penalizing solutions that are far from the gradient descent step; a more formal treatment of proximal descent and further interpretations are given in [12] and related works.

2.2.4.1 Deriving and Evaluating Proximal Operators

For optimization involving proximal operators to be efficient in practice, one needs a means to rapidly solve the proximal operator of the function of interest. Fortunately, due to the relatively simple form of the proximal operator, for a wide range of functions the proximal operator can be solved in closed form. A review of all known proximal operators is beyond the scope of this thesis, but many well known results can be found in [12, 13]. One property that is particularly useful in deriving proximal operators is the ***Moreau identity***, given by the equation

$$\mathbf{prox}_f(x) + \mathbf{prox}_{f^*}(x) = x, \tag{2.20}$$

where recall that f^* denotes the Fenchel dual. As a simple example application of the Moreau identity, consider the proximal operator of a gauge function.

Proposition 2 *Given a gauge function, σ_C, the proximal operator of σ_C is given by*

$$\mathbf{prox}_{\sigma_C}(x) = x - \operatorname*{arg\,min}_{z \in C^\circ} \|x - z\|_F^2 \tag{2.21}$$

Proof. Recall from Theorem 5 that $(\sigma_C)^*(x) = \delta_{C^\circ}(x)$. Thus, from the Moreau identity, we have

$$\mathbf{prox}_{\sigma_C}(x) = x - \mathbf{prox}_{\delta_{C^\circ}}(x) = x - \operatorname*{arg\,min}_{z} \tfrac{1}{2}\|x - z\|_F^2 + \delta_{C^\circ}(z), \tag{2.22}$$

which completes the result as the indicator function can be replaced by the constraint in the problem statement, and since we are searching for the arg min the 1/2 multiplier can be removed. ■

This above result is well known and can be used to derive many proximal operators. For example, the proximal operator of the l_1 is given by

$$\begin{aligned} \mathbf{prox}_{\|\cdot\|_1}(x) &= x - \arg\min_z \|x - z\|_F^2 \ \text{ s.t. } \ \|z\|_\infty \leq 1 \\ &= x - CLIP_1(x), \end{aligned} \tag{2.23}$$

where $CLIP_\lambda$ denotes the clipping operator which is given by

$$CLIP_\lambda(x) = \text{sign}(x) \odot \min\{|x|, \lambda\}, \tag{2.24}$$

and $\odot$ denotes an element-wise product. Likewise, using the above result of the proximal operator for the l_1 norm, one finds that the Moreau envelope of the l_1 norm is given by the once-differentiable Huber function,

$$\mathcal{M}_{\lambda\|\cdot\|_1}(x) = \sum_i \begin{cases} \frac{1}{2}x_i^2 & |x_i| \leq \lambda \\ \lambda(|x_i| - \frac{1}{2}\lambda) & |x_i| > \lambda. \end{cases} \tag{2.25}$$

2.3 Representation Learning

As discussed in the introduction, the overall goal of this thesis is to study the mathematical properties for learning structured representations from data, and this section will review several well known techniques that have been developed for this task. These techniques will include both ***unsupervised learning*** and ***supervised***

learning methods. Unsupervised learning methods take a dataset as input and seek to find a representation that closely (or exactly) approximates the dataset while ensuring that the representation has particular properties to facilitate subsequent analysis. Supervised learning methods, on the other hand, require both a dataset and a corresponding collection of labels or target values, and the overall goal of the learning method is not to find a close approximation of the data, but rather to find a data representation which allows one to efficiently and robustly predict the target value for a given data point from the learned representation. As a result, learning good representations in a supervised setting largely depends on finding data representations that preserve the relevant information for the task at hand while at the same time are stable and robust against noise and corruptions that might be present in the data.

2.3.1 Sparse and Low-Rank Methods

A major focus of research in machine learning, computer vision, signal processing, and other technical disciplines involving large datasets has been work on a broad array of techniques referred to as sparse and low-rank methods. As a general definition, to say that a set of numbers (typically a matrix or vector) is ***sparse*** simply implies that the number of non-zero entries in the set of numbers is much smaller than the dimensionality of the set (i.e., "most" of the entries in the set are zero), and the term ***sparse methods*** simply refers to methods that model some aspect of the problem under the assumption that a portion of the model is sparse. ***Low-rank methods***, as the name implies, are similar to sparse methods except that instead of assuming a set is sparse, one assumes that a matrix in the model has low-rank. As we will see, low-rank methods can actually be thought of as a special case of sparse methods, by assuming sparsity on one specific aspect of the model.

2.3.1.1 Sparsity

The fundamental component of sparse methods is the assumption that some portion of a model is sparse, but to make this concept meaningful, one needs some measure of how sparse a given signal is. Since sparsity implies a small number of non-zero entries, a natural measure of sparsity is to simply count the number of non-zero entries. This is usually notated mathematically with the l_0 pseudo-norm, defined as

$$\|x\|_0 = \sum_i \begin{cases} 1 & x_i \neq 0 \\ 0 & x_i = 0. \end{cases} \tag{2.26}$$

While the l_0 pseudo-norm provides a direct measure of sparsity and is simple to calculate for a fixed vector x, it is not a convex function, and solving problems involving the l_0 pseudo-norm can be very challenging. For example, given an optimization problem of the form

$$\min_x f(x) \;\; \text{s.t.} \;\; \|x\|_0 \leq k, \tag{2.27}$$

where $x \in \mathbb{R}^D$ is a vector we are trying to solve for and k is a given positive integer, then one must solve $\binom{D}{k}$ separate optimization problems to minimize $f(x)$ for each possible subset of k variables in x that is allowed to be non-zero. Clearly, this combinatorial search is infeasible for large problems with non-trivial values of k (i.e., $k \neq \{1, D\}$), so instead one is typically forced to accept approximate solutions which might be found through some form of greedy search algorithm.

As an alternative to the l_0 pseudo-norm, another popular measure of sparsity is the l_1 norm, defined simply as the sum of the absolute values of x,

$$\|x\|_1 = \sum_i |x_i|. \tag{2.28}$$

At first, it is not immediately intuitive why the l_1 norm should provide a meaningful measure of sparseness, but the relaxation comes from the fact that the l_1 norm is the convex envelope of the l_0 pseudo-norm, i.e., $(\|x\|_0^*)^* = \|x\|_1$ [14]. As a result, although the l_1 norm does not provide an exact measure of sparseness, it does provide a useful convex heuristic of sparseness and under certain conditions one can prove that by relaxing a l_0 pseudo-norm into a l_1 norm, solutions to the l_1 norm regularized problem will be equal to solutions of the l_0 regularized problem. As an example, solutions to the convex problem

$$\min_x \|x\|_1 \quad \text{s.t.} \quad \mathcal{A}(x) = b \tag{2.29}$$

will also be solutions to the non-convex problem

$$\min_x \|x\|_0 \quad \text{s.t.} \quad \mathcal{A}(x) = b \tag{2.30}$$

if $\mathcal{A}$ is a linear operator that satisfies certain requirements [15].

2.3.1.2 Structured Sparsity and Low-Rank Models

Extending the idea of using the l_1 norm as a measure of sparsity, the l_1 norm can also be composed with other functions to produce sparsity in many different aspects of the model. For example, if one is given a matrix $X \in \mathbb{R}^{D \times N}$ and an arbitrary vector norm $\| \cdot \|_u$, one can define the ***mixed norm*** $\|X\|_{u,1}$ as

$$\|X\|_{u,1} = \sum_{i=1}^{N} \|X_i\|_u, \tag{2.31}$$

which is essentially just the sum of the $\| \cdot \|_u$ vector norms of each column of X. One interpretation of the $\|X\|_{u,1}$ mixed norm is that it first constructs a vector of all the $\| \cdot \|_u$ column norms, $z = [\ \|X_1\|_u,\ \ldots,\ \|X_N\|_u\ ]$, and then the mixed norm is the l_1 norm (or sum because the norms are non-negative) of that vector of norms, $\|X\|_{u,1} = \|z\|_1$. With this interpretation, by taking an appropriate choice of $\|\cdot\|_u$ norm (namely choosing it to be a norm other than the l_1 norm) the mixed norm effectively encourages X to have a small number of columns which are non-zero, but within a given non-zero column the entries can be dense. Such a structure is typically referred to as ***group sparsity*** or ***structured sparsity***, and the idea can be generalized to taking the sums of norms of arbitrarily defined groups of variables (other than the matrix columns) to induce a wide array of sparsity patterns [13].

Moving beyond sparseness of the variables themselves, consider the singular value decomposition of a matrix $X \in \mathbb{R}^{D \times N}$, given by $X = \sum_{i=1}^{\min\{D,N\}} \sigma_i U_i V_i^T$, where U_i and V_i denote the singular vectors and σ_i denotes the singular values. To say that X is low-rank implies that most of the singular values (σ_i) are 0 (or the set of singular values is sparse). Returning to the idea that the l_1 norm encourages sparseness, one could then intuitively expect that taking the l_1 norm of the singular values of a matrix would encourage low-rank matrices, and this is exactly the motivation for the ***nuclear norm*** of a matrix, defined as the sum of the singular values,

$$\|X\|_* = \sum_{i=1}^{\min\{D,N\}} \sigma_i(X). \tag{2.32}$$

While the above discussion about why one might expect the nuclear norm to encourage low-rank solutions is only an intuitive argument, again using the notion of a convex envelope, one can show that the nuclear norm is the closest convex relaxation to the rank of a matrix [16], and similar to the l_1 case, one can also guarantee correct recovery of low-rank solutions using the nuclear norm for linear equality constrained problems provided the linear operator satisfies certain conditions [17].

2.3.2 Matrix Factorization

The next form of representation learning that will be discussed is ***matrix factorization***. As the name suggests, if one is given a data matrix Y, the goal of matrix factorization is to find two matrices (U, V) such that their product closely approximates Y, i.e., $Y \approx UV^T$. Of course, for a given matrix Y there are infinitely many possible factorizations such that $Y = UV^T$, so for the factorization problem to be well posed, one must look for factorizations (U, V) which satisfy certain properties that are beneficial for a particular application. The difference between most matrix factorization methods rests largely on what properties in particular are enforced on the factors.

2.3.2.1 Principal Component Analysis

Arguably the most well-known and wide-spread form of representation learning is ***principal component analysis*** (PCA). To understand the intuition behind PCA, consider a matrix $Y \in \mathbb{R}^{D\times N}$ where the columns of Y are data points in $\mathbb{R}^D$, $Y = [Y_1,\ Y_2, \ldots, Y_N]$, and the rows of Y have zero mean, i.e., $\sum_{i=1}^N Y_i = 0$. Now, given the factorized matrices $U \in \mathbb{R}^{D\times r}$ and $V \in \mathbb{R}^{N\times r}$, one sees that the approximate representation for an individual data point, Y_n, is given by

$$Y_n \approx (UV^T)_n = \sum_{i=1}^{r} U_i V_{n,i} \tag{2.33}$$

which implies that each data point in Y is being approximately represented by a linear combination of the r columns of U, with weight coefficients for the n^{th} data point contained in the n^{th} row of V. In this sense, the columns of U can be interpreted as a set of features (or components) that will be used to represent each of the N points in the dataset, and information about how to encode the N data points in terms of these features is stored in the appropriate rows of V. Given this interpretation of matrix factorization, the PCA model then seeks to find the features, U, and encodings, V, which will produce the best approximation of the data using a fixed number of features, r. This is accomplished by minimizing the least-squares error between the data and the approximation in the optimization problem

$$\min_{U \in \mathbb{R}^{D \times r}, V \in \mathbb{R}^{N \times r}} \|Y - UV^T\|_F^2 \;\; \text{s.t.} \;\; U^T U = I. \tag{2.34}$$

Note that the above optimization problem is not a convex optimization problem due both to the multiplication of U and V and the orthonormal constraint on U. However, from the Eckart-Young theorem [18], the above optimization problem can be solved via a singular value decomposition of Y and is one of the few examples of a non-convex optimization problem that can be solved efficiently.

In addition to having an algorithm that can efficiently solve (2.34), PCA also provides a convenient means to perform ***dimensionality reduction***. In particular, if the data points that makeup Y lie in a linear subspace with dimension $r \ll \min\{D, N\}$, then each data point in Y can be compactly represented by just $r \ll \min\{D, N\}$ parameters. Due to the potential for dimensionality reduction and the availability of an efficient algorithm to solve (2.34), PCA is a very powerful method for representation learning which has seen application in a very wide-range of problems [19]; however, there are several significant limitations to PCA. The first is that while PCA is good at finding a set of features, U, which can compactly represent all of the data points in Y, the learned features typically have little meaningful interpretation in terms of the problem at hand. As a result, one is often forced to do some sort of post-processing on the PCA representation to obtain useful information. Another limitation of PCA is that it is very fragile to corruptions in the dataset. From a statistical standpoint, one interpretation of PCA is that it assumes the datapoints are generated as points in a linear subspace with the addition of Gaussian noise, $Y_n = Ux_n + \epsilon$, $\epsilon \sim \mathcal{N}(0, \sigma)$ [19], so when this assumption of Gaussian noise is violated by the presence of large outliers in the dataset, the representations learned by PCA can also be grossly corrupted. Due to these limitations, multiple alternative matrix factorization approaches have been proposed. A few well known approaches are reviewed below, and Chapter 3 will develop a significantly generalized and unifying approach to learning representations via matrix factorization.

2.3.2.2 Non-Negative Matrix Factorization

A well-known alternative to PCA, ***non-negative matrix factorization*** (NMF) operates similarly to PCA, but instead of requiring the features, U, to be orthonormal, NMF instead requires both the features, U, and the encodings, V, to be non-negative. In particular, for a given number of features, r, NMF attempts to solve the optimization problem given by

$$\min_{U \in \mathbb{R}^{D \times r}, V \in \mathbb{R}^{N \times r}} \ell(Y, UV^T) \ \text{ s.t. } \ U \geq 0, \ V \geq 0, \tag{2.35}$$

where ℓ is some form of loss function that measures how closely Y is approximated by UV^T. One example loss function is the Frobenius norm as in PCA, $\ell(Y, UV^T) = \|Y - UV^T\|_F^2$, and another popular loss function in the NMF literature is given by

$$\ell(Y, X) = D(Y \,||\, X) = \sum_{i,j} \left(Y_{i,j} \log \frac{Y_{i,j}}{X_{i,j}} - Y_{i,j} + X_{i,j} \right), \tag{2.36}$$

which reduces to the Kullback-Leibler divergence when X and Y sum to 1 and is useful to model mixtures of probability distributions [20].

The motivation behind NMF is that in many applications the data itself is non-negative (i.e., $Y \geq 0$), and by constraining U and V to be non-negative one is searching for a representation that reproduces a datapoint Y_n in a purely additive way. As a result, NMF is often described as learning "parts" of a dataset which then allows one to unmix a collection of signals into their respective parts [21]. In many applications this approach has a well founded interpretation for the problem when the data is known to be generated in an additive manner. For example, in spectrometry, if one is given different mixtures of r different materials, then the spectrum of each material is known to be non-negative and the amount of a given material present in each mixture must also be non-negative, so by performing NMF on a collection of recorded spectra from different mixtures, one would then hope to recover the spectra of the various materials in the columns of U and the rows of V would contain the amount of each material present in a given mixture [22].

While NMF has many potential applications and better captures the physical constraints of many problems compared to PCA, NMF also has a few significant drawbacks. The first is that the number of "parts" that are present in the mixture, r, is not necessarily known *a priori* and one must employ some model selection strategy to determine an appropriate value (note that PCA also requires one to choose an appropriate value for r). Second, like PCA, the NMF optimization problem in (2.35) is non-convex, but unlike PCA, there is no known algorithm to efficiently solve the NMF optimization problem. As a result, in practice one can only obtain approximate solutions to the optimization problem and choices such as how the optimization algorithm is initialized can have a significant impact on the approximate solution one obtains [20].

2.3.2.3 Sparse Dictionary Learning

Another approach to representation learning based on matrix factorization is ***sparse dictionary learning***. The general idea behind sparse dictionary learning is that instead of trying to find a small number of features with which to represent the data as in PCA, we will instead try to find a potentially large number of features (referred to as a ***dictionary*** of features), but we will require that the representation for each particular data point will be generated by only using a small number of features (or the representation is ***sparse***). Sparse dictionary learning was first motivated by attempts to model the early stages of the mammalian visual system, where it was noted that if sparse dictionaries were learned from local patches of natural images, then the learned features were highly similar to the receptive fields of simple cells in primary visual cortex [23]. Later work has since established that modeling small patches of natural images via sparse dictionary learning has many benefits in application, and such approaches have achieved state-of-the-art results in standard image processing problems such as image denoising [24] and image in-painting [25].

Mathematically, sparse dictionary learning typically tries to solve an optimization problem with the form

$$\min_{U \in \mathbb{R}^{D \times r}, V \in \mathbb{R}^{N \times r}} \tfrac{1}{2}\|Y - UV^T\|_F^2 + \Theta(V) \quad \text{s.t.} \quad \|U_i\|_2 = 1 \ \forall i \in \{1, \ldots, r\}, \tag{2.37}$$

where $\Theta(V)$ is some function that promotes V to be sparse, such as the l_0 pseudo-norm or the l_1 discussed above.

Despite the success of sparse dictionary learning, it too suffers from many of the technical challenges associated with techniques like NMF. Namely, one must choose *a priori* the size of the dictionary, r, and even if one uses the l_1 relaxed form of sparse dictionary learning, the optimization problem is still non-convex overall and the success of the method will depend strongly on implementation details like how the optimization algorithm is initialized [26, 27].

2.3.3 Tensor Factorization

In many applications involving multi-modal data, one would like to find sets of features that explain the data along multiple modes of the data. For example, in something like census data a dataset might have entries that depend on spatial, temporal, and demographic factors, and one would like to find sets of spatial, temporal, and demographic features to explain a particular aspect of the data. For such multi-modal data, ***tensor factorization*** provides a natural generalization of matrix factorization and decomposes the data in multiple different types of features. Tensor methods are, in general, extremely flexible with regards to the dimensionality of both the dataset as well as the tensors factors that one is trying to recover. As a result, there have been a large number of different forms of tensor factorization or decomposition proposed in the literature [28]. Below, the two most common forms of tensor factorization, the CANDECOMP/PARAFAC factorization and the Tucker decomposition, are reviewed, and the notational convention that will be used for tensors in this thesis is introduced.

2.3.3.1 Tensor Notation

The mathematical formulations considered in this thesis will be very general in regards to the dimensionality of the data and variables. As a result, the notation will be based around the concept of a ***tensor***, which is essentially just a matrix generalized to more than two dimensions. For example, a matrix with m rows and n columns, $X \in \mathbb{R}^{m \times n}$, is a second order tensor. Generalizing this to a third order tensor, $X \in \mathbb{R}^{m \times n \times p}$, results in a cube with height m, width n, and depth p, and this can be extended to a K^{th} order tensor, $X \in \mathbb{R}^{d_1 \times \ldots \times d_K}$. To simplify this notation, capital letters will be used as a shorthand for a set of dimensions, and individual dimensions will be denoted with lower case letters. For example, $X \in \mathbb{R}^{d_1 \times \ldots \times d_K} \equiv X \in \mathbb{R}^D$ for $D = d_1 \times \ldots \times d_K$; similarly, $X \in \mathbb{R}^{D \times R} \equiv X \in \mathbb{R}^{d_1 \times \ldots \times d_K \times r_1 \times \ldots \times r_M}$ for $D = d_1 \times \ldots \times d_K$ and $R = r_1 \times \ldots \times r_M$. The ***cardinality*** of $X \in \mathbb{R}^D$ will be denoted as $\text{card}(X) = \prod_{i=1}^{K} d_i$. Given two tensors with matching dimensions except for the last dimension, $X \in \mathbb{R}^{D \times r_x}$ and $Z \in \mathbb{R}^{D \times r_z}$, $[X \ Z] \in \mathbb{R}^{D \times (r_x + r_z)}$ will be used to denote the concatenation of the two tensors along the last dimension.

A ***slice*** of a K^{th} order tensor is a $(K-1)$ order tensor which is a subset of the original tensor formed by holding one index of the tensor fixed. For example, given a matrix $X \in \mathbb{R}^{m \times n}$, the slices of X along the first dimension correspond to the m row vectors of X (which are of dimension $1 \times n$), and the slices of X along the second dimension correspond to the n column vectors of X (which are of dimension $m \times 1$). For a multidimensional tensor, X, a single subscript, X_i, will denote a slice along the last dimension of the tensor. For example, given a matrix $X \in \mathbb{R}^{d_1 \times r}$, then $X_i \in \mathbb{R}^{d_1}, i \in \{1, \ldots, r\}$, denotes the i^{th} column of X and $X = [X_1 \ \ldots \ X_r]$. Similarly, given a third order tensor $X \in \mathbb{R}^{d_1 \times d_2 \times r}$ then $X_i \in \mathbb{R}^{d_1 \times d_2}, i \in \{1 \ldots, r\}$, denotes the

i^{th} slice along the third dimension. Tensors which have a size of 1 along the last dimension and are not slices from a larger tensor will be denoted with lower-case letters. For example, $x \in \mathbb{R}^{D\times 1}$ denotes a tensor of size 1 along its last dimension, while $X_i \in \mathbb{R}^{D\times 1}$ is a slice from a larger tensor $X \in \mathbb{R}^{D\times r}$.

2.3.3.2 CANDECOMP/PARAFAC (CP) Factorization

Having introduced the tensor notation, this section returns to the idea of using tensor factorization to do representation learning by discussing the closest analogue to PCA for multidimensional tensors, the ***CANDECOMP/PARAFAC (CP) factorization*** of a tensor. In the rank-r CP factorization, one tries to approximate a multidimensional tensor, $Y \in \mathbb{R}^{d_1\times\cdots\times d_K}$, via a factorization into K factors, $(X^1, \ldots, X^K) \in \mathbb{R}^{d_1\times r} \times \ldots \times \mathbb{R}^{d_K\times r}$, given by

$$Y \approx \sum_{i=1}^{r} X_i^1 \otimes \cdots \otimes X_i^K, \tag{2.38}$$

where $X_i^1 \otimes \cdots \otimes X_i^K$ denotes the outer product of K vectors; e.g., $(a\otimes b\otimes c\otimes d)_{i,j,k,l} = a_i b_j c_k d_l$. The CP factorization clearly generalizes PCA, as if one takes $K = 2$, then the CP factorization is equal to the PCA factorization, modulo the fact that the CP factorization does not have orthogonality constraints on the factors, which PCA includes to guarantee uniqueness. Based on this analogy, the CP factorization has been used in a wide range of applications with multi-modal data, such as tracking text email conversations over time [29], analyzing functional MRI and electroencephalogram (EEG) data [30, 31], and image classification and compression [32]. However, despite the close similarity to PCA, the difficulty of the factorization problem increases significantly for factorizations with $K > 2$ factors. In particular, calculating the CP factorization of a high dimensional tensor is a known NP-hard problem [33] and thus there is no known algorithm to calculate it in polynomial time.

2.3.3.3 Tucker Decomposition

Generalizing further, the ***Tucker decomposition*** of a tensor again decomposes a multidimensional tensor into a set of K factors, but instead of simply taking the sum of rank-1 tensors, the Tucker decomposition includes a fixed "core tensor", $\kappa \in \mathbb{R}^{r_1,\ldots,r_K}$, which models more potential interactions among the various factors. Specifically, the Tucker decomposition attempts to solve the problem

$$Y \approx \sum_{i_1=1}^{r_1} \cdots \sum_{i_K=1}^{r_K} (X^1_{i_1} \otimes \cdots \otimes X^K_{i_K}) \kappa(i_1, \ldots, i_K). \tag{2.39}$$

Note that the Tucker decomposition is a generalization of the CP factorization, as if one takes the core tensor to be diagonal,

$$\kappa(i_1, \ldots, i_K) = \begin{cases} 1 & i_1 = i_2 = \cdots = i_k \\ 0 & \text{else} \end{cases}, \tag{2.40}$$

then the Tucker decomposition reverts to the CP factorization. One advantage of the Tucker decomposition over CP factorization is that it is typically easier to fit in practice, as for certain choices of core tensors the Tucker decomposition can be approximated by solving a sequence of PCA problems after reshaping the original tensor, Y, into a matrix. A discussion of specific algorithms to approximate CP and Tucker factorizations is beyond the scope of this work but can be found in [28].

In applications, the Tucker decomposition has been used from problems such as signal processing [34], facial recognition [35], and human motion recognition [36]. Similar to matrix factorization and CP tensor factorizations, solving the Tucker decomposition problem is also a non-convex optimization problem, and thus one cannot typically find the globally optimal Tucker decomposition in polynomial time.

2.3.4 Neural Networks

The final form of representation learning that will be discussed is ***neural networks***, which seek to perform computations based on roughly approximating the behavior of neurons in biological nervous systems. The basic computational unit in neural networks is a simulated neuron, which takes a vector of inputs, $z \in \mathbb{R}^d$, and produces an output x based on the formula

$$x = \psi(w^T z + b) \tag{2.41}$$

where $w \in \mathbb{R}^d$ is a vector of "weight parameters" that model synaptic connection weights in biological neurons, $\psi : \mathbb{R} \to \mathbb{R}$ is a non-decreasing (typically non-linear) function to mimic the thresholding behavior of biological neurons, and $b \in \mathbb{R}$ is a scalar "bias term" that sets the output for an input $z = 0$.

While the computation performed by a single artificial neuron is relatively simple, by forming interconnected networks with large numbers of neurons, artificial neural networks are capable of performing very complicated computations. The most common network arrangement for computer science purposes is known as a ***feed-forward network***. In a feed-forward network, neurons are arranged into layers, and the inputs of neurons in a given layer only connect to the outputs of the neurons in the layer below them.

Neural networks are known to be capable of approximating any smooth function to an arbitrary level of precision provided the network contains enough neurons [37], but this only guarantees the existence of neuron weights w and bias terms b that will result in a network which will approximate the function. It says nothing about how to find such parameters. The task of learning the appropriate network weight parameters, given a set of training data and corresponding desired outputs, is known as ***network training*** and is the primary computational challenge that must be solved to apply neural networks to problems in practice. Similar to other forms of representation learning, training neural networks is an inherently non-convex problem, and this non-convexity presents a significant challenge. Chapter 4 will address the issue of non-convexity and provide a mathematical framework to analyze certain forms of neural network training formulations and derive sufficient conditions to guarantee the global optimality of local minimizers.

2.3.4.1 Classical Neural Networks

Classically, the first form of neural network was based on the ***perceptron***, which is a single neuron that uses a step-function non-linearity [38]. The step-function was used initially to mimic the "binary" nature of biological neurons "a neuron fires an action potential or it does not", but the non-differentiability of the step function and the fact that the step function has 0 gradient almost everywhere in its domain makes it very hard to learn the appropriate weight parameters during the network training stage. In the late 1980s, it was realized that if the non-linearities were changed to be differentiable then it would be possible to update the network weight parameters by doing gradient descent on an objective function, which led to smooth approximations of the step function being used as non-linearities, such as a sigmoid or hyperbolic tangent function. Due to the feed-forward architecture of the neural network, the task of calculating the gradient of the objective function with respect to the weights in the i^{th} layer ends up being independent of the other weights in the network if one has access to the gradient of the $(i+1)^{\text{th}}$ layer. As a result, the gradients of all the layers can be calculated sequentially by traversing the network from output-to-input, which led to the term ***back-propagation*** [39]. While back-propagation provided a means to train neural networks with differentiable non-linearities, a great deal of work in the 1990s and 2000s was devoted to "non-neural" learning methods, such as support vector machines, decision trees, and boosting [40–43]. This was largely due to the fact that, although neural networks achieved reasonable performance in certain applications, the large numbers of free parameters that needed to be learned in neural networks combined with relatively little access to training data limited the performance of many classical back-propagation networks.

2.3.4.2 Modern Neural Networks and Deep Learning

Over the past several years, large scale neural networks have seen a massive resurgence in popularity and achieved state-of-the-art performance in many challenging machine learning applications, particularly in areas such as image and speech recognition [43–47]. Several factors have been attributed to this recent success. The first is simply the matter of scale. With access to millions of potential images from which to train a network and significantly expanded computing power, training very large scale networks (often referred to as ***deep learning***) with enough training data to prevent over-fitting is now feasible in a reasonable amount of time. However, beyond simply training larger networks with more data, many aspects are common to current state-of-the-art networks.

The first is that most modern networks have replaced traditional sigmoid or hyperbolic tangent non-linearities with what is known as the ***rectified-linear unit***

(ReLU), which is simply given by $\psi(x) = \max\{0, x\}$. One of the initial motivations for the ReLU non-linearity was that the computation of its gradient is very simple (just a 1 or a 0 depending on whether x is positive or not) and its gradient does not saturate for very large inputs; however, in addition to these computational advantages, ReLU non-linearities have been shown experimentally to achieve significant performance boosts over more traditional sigmoid-style non-linearities [44–46, 48].

Beyond using ReLU non-linearities, many modern networks are ***convolutional neural networks***, meaning that instead of taking a dot-product between the inputs to a neuron and the neuron's weight parameter vector, a convolution between the inputs to a neuron and a given convolutional kernel is taken. The convolutional kernel is then shared between all of the neurons in that given layer, which greatly reduces the number of parameters that need to be learned in the network.

To date, many of the practices common to the deep learning field have been arrived at largely through experimentation, and well founded theoretical principles for why something like a ReLU non-linearity achieves better performance than a traditional sigmoid non-linearity are lacking. In Chapter 4 these issues will be explored in much greater depth, and an initial theoretical framework from which to approach the analysis of deep neural networks will be presented.

Chapter 3

Structured Matrix Factorization

In many large datasets, relevant information often lies in a subspace of much lower dimension than the ambient space, and thus the goal of many learning algorithms can be broadly interpreted as trying to find or exploit this underlying "structure" that is present in the data. One structure that is particularly useful both due to its wide-ranging applicability and efficient computation is the linear subspace model. Generally speaking, if one is given N data points from a D dimensional ambient space, $Y = [Y_1,\ Y_2,\ \ldots, Y_N] \in \mathbb{R}^{D \times N}$, a linear subspace model simply implies that there exists matrices (U, V) such that $Y \approx UV^T$. For problems where either U or V is known *a priori* the problem simplifies considerably, but if both U and V are allowed to be totally arbitrary one can always find an infinite number of (U, V) matrices that satisfy this requirement. As a result, to accomplish anything meaningful one must impose some restrictions on the properties of (U, V), and this basic idea captures a wide variety of common techniques. A few well known examples can be summarized as follows:

- **Principal Component Analysis (PCA):** The number of columns, r, in (U, V) is typically constrained to be small, $r \ll \min\{D, N\}$, and U is constrained to have orthonormal columns.
- **Non-Negative Matrix Factorization (NMF):** The number of columns in (U, V) is similarly constrained to be small, and (U, V) are also required to be non-negative [20, 21].
- **Sparse Dictionary Learning (SDL):** The number of columns in (U, V) is allowed to be larger than $\min\{D, N\}$, but the columns of U are typically required to have unit Euclidean norm and V is required to be "sparse" as measured by something like the l_1 norm or the l_0 pseudo-norm [26, 27][1].

Mathematically, the general problem of recovering structured linear subspaces from a dataset can be captured by a structured matrix factorization problem of the form

$$\min_{U,V} \ell(Y, UV^T) + \lambda\Theta(U, V), \tag{3.1}$$

where ℓ is some ***loss function*** that measures how well Y is approximated by UV^T and Θ is a ***regularizer*** that encourages or enforces specific properties in (U, V). By

[1] As a result, in sparse dictionary learning, one does not assume that there exists a single low-dimensional subspace to model the data, but rather that the data lies in a union of a large number of low-dimensional subspaces

taking an appropriate combination of ℓ and Θ one can formulate both unsupervised learning techniques, such as PCA, NMF, and SDL, or supervised learning techniques like discriminative dictionary learning [49, 50] and learning max-margin factorized classifiers [51]. However, while there are wide-ranging applications for structured matrix factorization methods that have achieved good empirical success, the associated optimization problem (3.1) is non-convex regardless of the choice of ℓ and Θ functions due to the presence of the matrix product UV^T. As a result, aside from a few special cases (such as PCA), finding solutions to (3.1) poses a significant challenge, which often requires one to instead consider approximate solutions that depend on a particular choice of initialization and optimization method.

Given the challenge of non-convex optimization, one possible approach to matrix factorization problems is to relax the non-convex problem into a problem which is convex on the product of the factorized matrices, $X = UV^T$, and then recover the factors of X after solving the convex relaxation. As a concrete example, in low-rank matrix factorization, one might be interested in solving a problem of the form

$$\min_X \ell(Y, X) \text{ subject to } \operatorname{rank}(X) \leq r, \tag{3.2}$$

which is equivalently defined as a factorization problem

$$\min_{U,V} \ell(Y, UV^T) \tag{3.3}$$

where the rank constraint is enforced by limiting the number of columns in the U and V matrices to be less than or equal to r. However, aside from a few special choices of ℓ, solving (3.2) or (3.3) is in general a NP-hard problem. Instead, one can relax (3.2) into a convex problem by using a convex regularization that promotes low-rank solutions, such as the nuclear norm $\|X\|_*$ (sum of the singular values of X), and then solve

$$\min_X \ell(Y,X) + \lambda\|X\|_*, \tag{3.4}$$

which can often be done efficiently if $\ell(Y,X)$ is convex with respect to X [17, 52]. Given a solution to (3.4), X_{opt}, it is then simple to find a low-rank factorization $UV^T = X_{opt}$ via a singular value decomposition. Unfortunately, however, while the nuclear norm provides a nice convex relaxation for low-rank matrix factorization problems, nuclear norm relaxation does not capture the full generality of problems such as (3.1) as it does not necessarily ensure that X_{opt} can be factorized as $X_{opt} = UV^T$ for some (U,V) pair which has the desired structure encouraged by $\Theta(U,V)$ (e.g., in non-negative matrix factorization we require U and V to be non-negative), nor does it provide a means to find the desired factors.

Based on the above discussion, optimization problems in the factorized space, such as (3.1), versus problems in the product space, with (3.4) as a particular example, both present various advantages and disadvantages. Factorized problems attempt to solve for the desired factors (U,V) directly, provide significantly increased modeling flexibility by permitting one to model structure on the factors (sparsity, non-negativity,

Table 3.1: Typical properties of problems in the factorized vs product space. (Items in bold are desirable.)

	Product Space (X)	Factorized Space (U, V)
Convex	**Yes**	No
Problem Size	Large	**Small**
Structured Factors	No	**Yes**

etc.), and allow one to potentially work with a significantly reduced number of variables if the number of columns in (U, V) is $\ll \min\{D, N\}$; however, they suffer from the significant challenges associated with non-convex optimization. Problems in the product space, on the other hand, can be formulated to be convex, which affords many practical algorithms and analysis techniques, but one is required to optimize over a potentially large number of variables and solve a second factorization problem in order to recover the factors (U, V) from the solution X. These various pros and cons are briefly summarized in Table 3.1.

To bridge this gap between the two classes of problems, here we explore the link between non-convex matrix factorization problems, which have the general form

$$\textbf{Factorized Problems:} \quad \min_{U,V} \ell(Y, UV^T) + \lambda\Theta(U, V), \tag{3.5}$$

and a closely related family of convex problems in the product space, given by

$$\textbf{Convex Problems:} \quad \min_{X} \ell(Y, X) + \lambda\Omega_\Theta(X), \tag{3.6}$$

where the function Ω_Θ will be defined based on the choice of the regularization function Θ and will have the desirable property of being a convex function of X. Unfortunately, while the optimization problem in (3.6) is convex w.r.t. X, it will typically be non-tractable to solve. Moreover, even if a solution to (3.6) could be found, solving a convex problem in the product space does not necessarily achieve our goal, as we still must solve another matrix factorization problem to recover the (U, V) factors with the desired properties encouraged by the Θ function (sparsity, non-negativity, etc.). Nevertheless, the two problems given by (3.5) and (3.6) will be tightly coupled. Specifically, the convex problem in (3.6) will be shown to be a global lower-bound to the non-convex factorized problem in (3.5), and solutions to the factorized problem will also be solutions to the convex problem for $X = UV^T$. As a result, we will tailor our results to the non-convex factorization problem (3.5) using the convex function (3.6) as an analysis tool. While the optimization problem in the factorized space is not convex, by analyzing this tight interconnection between the two problems we will show that local minima of the non-convex factorized problem will be global minima if the factorized matrices, (U, V), have sufficiently many columns, and the number of the columns in (U, V) can be adapted to the data instead of being fixed *a priori*. In addition, a practical optimization strategy that is parallelizable and often requires a much smaller set of variables is discussed; in chapter 5 experimental results are presented for several real-world applications.

3.1 Mathematical Background and Prior Work

As discussed above, relaxing low-rank matrix factorization problems via nuclear norm formulations fails to capture the full generality of factorized problems as it does not allow one to find "structured" factors, (U,V), with desired properties encouraged by $\Theta(U,V)$ (sparseness, non-negativity, etc.). To address this issue, several studies have explored a more general convex relaxation via the matrix norm given by

$$\begin{aligned} \|X\|_{u,v} &\equiv \inf_{r\in\mathbb{N}_+} \inf_{U,V:UV^T=X} \sum_{i=1}^{r} \|U_i\|_u \|V_i\|_v \\ &\equiv \inf_{r\in\mathbb{N}_+} \inf_{U,V:UV^T=X} \sum_{i=1}^{r} \tfrac{1}{2}(\|U_i\|_u^2 + \|V_i\|_v^2) \end{aligned} \tag{3.7}$$

where (U_i,V_i) denote the i^{th} columns of U and V, respectively, $\|\cdot\|_u$ and $\|\cdot\|_v$ are arbitrary vector norms, and the number of columns (r) in the U and V matrices is allowed to be variable [53–57]. The norm in (3.7) has appeared under multiple names in the literature, including the projective tensor norm, decomposition norm, and atomic norm, and by replacing the column norms in (3.7) with gauge functions the formulation can be generalized to incorporate additional regularization on (U,V), such as non-negativity, while still being a convex function of X [55]. Further, it is worth noting that for particular choices of the $\|\cdot\|_u$ and $\|\cdot\|_v$ vector norms, $\|X\|_{u,v}$ reverts to several well known matrix norms and thus provides a generalization of many commonly used regularizers. Notably, when the vector norms are both l_2 norms, the form in (3.7) becomes the well known variational definition of the nuclear norm. More explicitly,

$$\begin{aligned} \|X\|_* = \|X\|_{2,2} &\equiv \inf_{r \in \mathbb{N}_+} \inf_{U,V:UV^T=X} \sum_{i=1}^{r} \|U_i\|_2 \|V_i\|_2 \\ &\equiv \inf_{r \in \mathbb{N}_+} \inf_{U,V:UV^T=X} \sum_{i=1}^{r} \tfrac{1}{2}(\|U_i\|_2^2 + \|V_i\|_2^2). \end{aligned} \tag{3.8}$$

Beyond nuclear norm relaxations, the $\|\cdot\|_{u,v}$ norm has the appealing property that by an appropriate choice of vector norms $\|\cdot\|_u$ and $\|\cdot\|_v$ (or more generally gauge functions), one can promote desired properties in the factorized matrices (U, V) while still working with a problem which is convex w.r.t. X.

3.1.1 Matrix Factorization as Semidefinite Optimization

Due to the increased modeling opportunities it provides, several studies have explored structured matrix factorization formulations based on the $\|\cdot\|_{u,v}$ norm in a way that allows one to work with a highly reduced set of variables while still providing some guarantees of global optimality. In particular, it is possible to explore optimization problems over factorized matrices (U, V) of the form

$$\min_{U,V} \ell(Y, UV^T) + \lambda \|UV^T\|_{u,v}. \tag{3.9}$$

While (3.9) is a convex function of the product $X = UV^T$, the problem is still non-convex with respect to (U, V) jointly due to the matrix product. However, if we define a matrix Γ to be the concatenation of U and V

$$\Gamma \equiv \begin{bmatrix} U \\ V \end{bmatrix} \implies \Gamma\Gamma^T = \begin{bmatrix} UU^T & UV^T \\ VU^T & VV^T \end{bmatrix}, \tag{3.10}$$

we see that UV^T is a submatrix of the positive semidefinite matrix $\Gamma\Gamma^T$. After defining the function $H : S_n^+ \to \mathbb{R}$

$$H(\Gamma\Gamma^T) = \ell(Y, UV^T) + \lambda \|UV^T\|_{u,v}, \tag{3.11}$$

it is clear that the proposed formulation (3.9) can be recast as an optimization over a positive semidefinite matrix $X = \Gamma\Gamma^T$.

3.1.1.1 Semidefinite Optimality: Standard Form and Differentiable Problems

At first the above discussion seems to be a circular argument, since while $H(X)$ is a convex function of (X), this says nothing about finding Γ (or U and V). However, results for semidefinite programs in standard form show that one can minimize $H(X)$ by solving for Γ directly without introducing any additional local minima, provided that the rank of Γ is larger than the rank of the true solution X_{opt} [58]. Further, if the rank of the true solution is not known *a priori*, if $H(X)$ is a twice differentiable function, then any local minima w.r.t. Γ such that Γ is a rank-deficient matrix give a global minimum of $H(\Gamma\Gamma^T)$. Formally, one has the following result.

Proposition 3 *[54] Let $H : S_n^+ \to \mathbb{R}$ be a twice differentiable convex function with compact level sets. If Γ is a rank deficient local minimum of $h(\Gamma) = H(\Gamma\Gamma^T)$, then $X = \Gamma\Gamma^T$ is a global minimum of $H(X)$.*

While these results provide encouragement that it is sometimes possible to solve problems in the factorized domain, the projective tensor norm is, unfortunately, not twice differentiable in general, so the above result can not be applied directly.

3.1.1.2 Semidefinite Optimality: Non-Differentiable Problems

Due to the fact that many problems of interest involve non-differentiable components, the above result is often challenging to apply. However, if $H(X)$ is composed of the sum of a twice differentiable and a non-differentiable convex function, then our prior work has shown that it is still possible to guarantee that rank-deficient local minima w.r.t. Γ give global minima of $H(\Gamma\Gamma^T)$. In particular, the above proposition can be extended to non-differentiable functions via the following result.

Proposition 4 *[56] Let $F : S_n^+ \to \mathbb{R}$ be a twice differentiable convex function with compact level sets and let $G : S_n^+ \to \mathbb{R}$ be a proper, lower semi-continuous convex function that is potentially non-differentiable. If Γ is a rank deficient local minimum of $h(\Gamma) = H(\Gamma\Gamma^T) = F(\Gamma\Gamma^T) + G(\Gamma\Gamma^T)$, then $X = \Gamma\Gamma^T$ is a global minimum of $H(X) = F(X) + G(X)$.*

Taken together, these results allow one to solve (3.9) using a potentially highly reduced set of variables if the rank of the true solution is much smaller than the dimensionality of X.

Unfortunately, while the above results from semidefinite programming are sufficient if we only wish to find general factors such that $UV^T = X$, for the purposes of solving structured matrix factorizations, we are interested in finding factors (U, V) that achieve the infimum in the definition of (3.7), which is not provided by a solution to (3.9). To make this point more explicit, note that the problem of optimizing (3.9) can be equivalently written as

$$\begin{aligned} &\min_{U,V} \ell(Y, UV^T) + \lambda \|UV^T\|_{u,v} = \\ &\min_{U,V} \ell(Y, UV^T) + \lambda \inf_{U',V'} \sum_{i=1}^{r} \|U_i'\|_u \|V_i'\|_v \;\; \text{s.t.} \;\; U'(V')^T = UV^T. \end{aligned} \tag{3.12}$$

Note that in the equation above, there is an additional degree of freedom in the sense that the (U, V) factors that are found from the semidefinite optimization do not necessarily need to be the same (U', V') factors that achieve the infimum in (3.7). As a result, the results from semidefinite optimization are not directly applicable to problems such as (3.9) as they deal with different optimization problems. Here we will show that results regarding global optimality can still be derived for the non-convex optimization problem given in (3.9) as well as for more general matrix factorization formulations.

3.2 Structured Matrix Factorization Problem Formulation

Our analysis will be based on a convex regularization function which is a generalization of the $\|\cdot\|_{u,v}$ norm and is similarly defined in the product space but allows one to enforce structure in the factorized space. The basic idea behind the regularization function is to note that any matrix factorization can be interpreted as the sum of rank-1 matrices, $X = UV^T = \sum_{i=1}^{r} U_i V_i^T$, and the number of rank-1 matrices in the factorization, r, will be fit to the data via a sparsity promoting regularization.

3.2.1 Matrix Factorization Regularizers

To define the regularization, it will be necessary to have a function that regularizes rank-1 matrices, which can be defined as follows:

Definition 12 *A function* $\theta : \mathbb{R}^D \times \mathbb{R}^N \to \mathbb{R}_+ \cup \infty$ *is said to be a* ***rank-1 regularizer*** *if*

1. $\theta(u, v)$ *is positively homogeneous with degree 2:* $\theta(\alpha u, \alpha v) = \alpha^2 \theta(u, v)\ \forall \alpha \geq 0$.

2. $\theta(u, v)$ *is positive semi-definite:* $\theta(0, 0) = 0$ *and* $\theta(u, v) \geq 0\ \forall (u, v)$.

3. $\min_{\alpha > 0} \theta(\alpha u, \alpha^{-1} v) = \inf_{\alpha > 0} \theta(\alpha u, \alpha^{-1} v) > 0$ *for all* $\{(u, v) : uv^T \neq 0\}$.

Note that this is a very general set of requirements, and one can propose a very wide range of rank-1 regularizers that will satisfy these three properties. Specific examples of regularizers that can be used for well known problems will be described below, and we will prove our theoretical results using this general definition of a rank-1 regularizer. Later, when discussing specific algorithms that can be used to solve structured matrix factorization problems in practice, we will require that $\theta(u, v)$ satisfies a few additional requirements.

Using the notion of a rank-1 regularizer, we are now prepared to define a regularization function on matrices of arbitrary rank as follows:

Definition 13 *Given a rank-1 regularizer $\theta(u,v) : \mathbb{R}^D \times \mathbb{R}^N \to \mathbb{R}_+ \cup \infty$, the* ***matrix factorization regularizer*** $\Omega_\theta : \mathbb{R}^{D\times N} \to \mathbb{R}_+ \cup \infty$ *is defined as*

$$\Omega_\theta(X) \equiv \inf_{r\in\mathbb{N}_+} \inf_{U\in\mathbb{R}^{D\times r}, V\in\mathbb{R}^{N\times r}} \sum_{i=1}^{r} \theta(U_i, V_i) \ \text{ s.t. } \ X = UV^T. \tag{3.13}$$

The function defined in (3.13) is very closely related to other regularizers that have appeared in the literature. In particular, taking $\theta(u,v) = \|u\|_u \|v\|_v$ or $\theta(u,v) = \frac{1}{2}(\|u\|_u^2 + \|v\|_v^2)$ for arbitrary vector norms $\|\cdot\|_u$ and $\|\cdot\|_v$ gives the $\|\cdot\|_{u,v}$ norm discussed above. Note, however, there is no requirement for $\theta(u,v)$ to be convex w.r.t. (u,v) or to be composed of norms. As long as θ satisfies the requirements from Definition 12 one can show that Ω_θ satisfies the following proposition[2]:

Proposition 5 *Given a rank-1 regularizer θ, the matrix factorization regularizer, Ω_θ satisfies the following properties.*

1. $\Omega_\theta(0) = 0$ *and* $\Omega_\theta(X) > 0 \ \forall X \neq 0$.

2. $\Omega_\theta(\alpha X) = \alpha \Omega_\theta(X) \ \forall \alpha \geq 0$.

3. $\Omega_\theta(X+Y) \leq \Omega_\theta(X) + \Omega_\theta(Y) \ \forall (X,Y)$.

[2] Note that Properties 1-3 almost satisfy the requirements for a gauge function on X. The missing condition for Ω_θ to be a norm is that $\Omega_\theta(X)$ must be invariant w.r.t. negative scaling (i.e., $\Omega_\theta(-X) = \Omega_\theta(X)$), and if θ satisfies the final condition, then it is easily shown that this will be true. Also in this case, Ω_θ becomes a special case of the atomic norm in [57] for atoms $\{uv^T : \theta(u,v) \leq 1\}$.

4. $\Omega_\theta(X)$ is convex w.r.t. X.

5. The infimum in (3.13) *can always be achieved with $r \leq DN$.*

6. If $\theta(-u,v) = \theta(u,v)$ or $\theta(u,-v) = \theta(u,v)$, then Ω_θ is a norm on X.

We do not show the proof here, as proofs for very similar forms can be found in related works [53–55, 57, 59] and use largely identical arguments. A full proof of the above results is also provided in Proposition 10 in the next chapter for a more general regularization function which includes Ω_θ as a special case.

From the above proposition, note that the first 3 properties show that Ω_θ is a gauge function on X (and further it will be a norm if property 6 is satisfied), which also implies that it must be a convex function of X. Note that while $\Omega_\theta(X)$ is a convex function of X, it can still be very challenging to evaluate or optimize functions involving Ω_θ due to the fact that it requires solving a non-convex optimization problem by definition. However, by exploiting the convexity of Ω_θ, we are able to use it to study the optimality conditions of many associated non-convex matrix factorization problems, several examples of which are provided below.

3.2.2 Examples of Structured Matrix Factorization Problems

The matrix factorization regularizer provides a natural bridge between convex formulations in the product space (3.6) and non-convex functions in the factorized space (3.5) due to the fact that $\Omega_\theta(X)$ is a convex function of X while from the definition (3.13) one can induce a wide range of properties in (U, V) by an appropriate choice of $\theta(u, v)$ function. In what follows, we give a number of examples which lead to variants of several structured matrix factorization problems that have been studied previously in the literature.

Low-Rank: The first example of note, which was introduced in the introduction to this chapter, is to relax low-rank constraints into nuclear norm regularized problems. Taking $\theta(u, v) = \frac{1}{2}(\|u\|_2^2 + \|v\|_2^2)$ gives the well-known variational form of the nuclear norm, $\Omega_\theta(X) = \|X\|_*$, and thus provides a means to solve problems in the factorized space where the size of the factorization gets controlled by regularization. In particular we have the conversion,

$$\begin{aligned} &\min_X \ell(Y, X) + \lambda\|X\|_* \iff \\ &\min_{r,U,V} \ell(Y, UV^T) + \tfrac{\lambda}{2}(\|U\|_F^2 + \|V\|_F^2) \iff \\ &\min_{r,U,V} \ell(Y, UV^T) + \lambda \sum_{i=1}^{r} \|U_i\|_2 \|V_i\|_2, \end{aligned} \tag{3.14}$$

where the $\iff$ notation implies that solutions to all 3 objective functions will have identical values at the global minimum and any global minimum w.r.t. (U, V) will be a global minimum for $X = UV^T$. While the above equivalence is well known for the nuclear norm [17, 60], the factorization is "unstructured" in the sense that the Euclidean norms do not bias the columns of U and V to have any particular properties, so to find factors with additional structure, such as non-negativity, sparseness, etc., more general $\theta(u,v)$ functions need to be considered.

Non-Negative Matrix Factorization: Recall the variational form of the nuclear norm from above uses $\theta(u,v) = \frac{1}{2}(\|u\|_2^2 + \|v\|_2^2)$ or $\theta(u,v) = \|u\|_2\|v\|_2$. If we extend this to now add non-negative constraints on (u,v), we get $\theta(u,v) = \frac{1}{2}(\|u\|_2^2 + \|v\|_2^2) + \delta_{\mathbb{R}_+}(u) + \delta_{\mathbb{R}_+}(v)$, which acts similar to the variational form of the nuclear norm in the sense that it limits the number of non-zero columns in (U,V), but it also imposes the constraints that U and V must be non-negative. As a result, one gets a convex relaxation of traditional non-negative matrix factorization

$$\begin{aligned}&\min_{U,V} \ell(Y,UV^T) \;\text{ s.t. }\; U \geq 0,\; V \geq 0 \implies \\ &\min_{r,U,V} \ell(Y,UV^T) + \tfrac{\lambda}{2}(\|U\|_F^2 + \|V\|_F^2) \;\text{ s.t. }\; U \geq 0,\; V \geq 0.\end{aligned} \tag{3.15}$$

Now note the $\implies$ notation is meant to imply that the two problems are not strictly equivalent as in the nuclear norm example. The key difference between the two forms above is that in the top equation the number of columns in (U,V) is fixed *a priori*, while in the bottom form the number of columns in (U,V) is allowed to be variable and adapted to the data via the low-rank regularization induced by the two Frobenius norms on (U,V).

Row or Columns Norms: Taking $\theta(u,v) = \|u\|_1\|v\|_v$ is known to result in $\Omega_\theta(X) = \sum_{i=1}^D \|(X^T)_i\|_v$, i.e., the sum of the $\|\cdot\|_v$ norms of the rows of X, while taking $\theta(u,v) = \|u\|_u\|v\|_1$ results in $\Omega_\theta(X) = \sum_{i=1}^N \|X_i\|_u$, i.e., the sum of the $\|\cdot\|_u$

norms of the columns of X [53, 54]. As a result, the regularizer $\Omega_\theta(X)$ generalizes the $\|X\|_{u,1}$ and $\|X\|_{1,v}$ mixed norms, but the factorization problem in this case is relatively uninteresting as taking either U or V to be the identity (depending on whether the l_1 norm is on the columns of U or V, respectively) and the other matrix to be X (or X^T) results in one of the possible optimal factorizations. The resulting reformulations into a factorized form gives

$$\begin{aligned} \min_X \ell(Y,X) + \lambda\|X\|_{1,v} &\iff \min_{r,U,V} \ell(Y,UV^T) + \lambda\sum_{i=1}^r \|U_i\|_1\|V_i\|_v \\ \min_X \ell(Y,X) + \lambda\|X\|_{u,1} &\iff \min_{r,U,V} \ell(Y,UV^T) + \lambda\sum_{i=1}^r \|U_i\|_u\|V_i\|_1. \end{aligned} \tag{3.16}$$

Sparse Dictionary Learning: Similar to the non-negative matrix factorization case, convex relaxations of sparse dictionary learning can also be obtained by combining l_2 norms with sparsity-inducing regularization. For example, taking $\theta(u,v) = \frac{1}{2}(\|u\|_2^2 + \|v\|_2^2 + \gamma\|v\|_1^2)$ results in a relaxation

$$\begin{aligned} &\min_{U,V} \ell(Y,UV^T) + \lambda\|V\|_1 \ \text{ s.t. } \ \|U_i\|_F = 1 \implies \\ &\min_{r,U,V} \ell(Y,UV^T) + \tfrac{\lambda}{2}(\|U\|_F^2 + \|V\|_F^2 + \gamma\sum_{i=1}^r \|V_i\|_1^2) \end{aligned} \tag{3.17}$$

which was considered as a convex relaxation of sparse dictionary learning in [54], where now the number of atoms in the dictionary is fit to the dataset via the low-rank regularization induced by the Frobenius norms. A similar approach would be to take $\theta(u,v) = \|u\|_F(\|v\|_F + \gamma\|v\|_1)$.

Sparse PCA: If both the rows and columns of U and V are regularized to be sparse, then one can obtain convex relaxations of sparse PCA [61]. One example of this is to take $\theta(u,v) = \frac{1}{2}(\|u\|_2^2 + \gamma_u\|u\|_1^2 + \|v\|_2^2 + \gamma_v\|v\|_1^2)$. Alternatively, one can also place constraints on the number of elements in the non-zero support of each column in (u,v) via a rank-1 regularizer of the form $\theta(u,v) = \frac{1}{2}(\|u\|_2^2 + \|v\|_2^2) + \delta_{\|\cdot\|_0 \leq k}(u) + \delta_{\|\cdot\|_0 \leq q}(v)$, where $\delta_{\|\cdot\|_0 \leq k}(u)$ denotes the indicator function that u has k or fewer non-zero elements. Such a form was analyzed in [62] and gives a relaxation of sparse PCA that regularizes the number of sparse components via the Frobenius norms while requiring that a given component must have the specified level of sparseness.

General Structure: More generally, this theme of using a combination of l_2 norms and additional regularization on the factors can be used to model additional forms of structure on the factors. For example one can take $\theta(u,v) = \|u\|_2\|v\|_2 + \gamma\hat{\theta}(u,v)$ or $\theta(u,v) = \|u\|_2^2 + \|v\|_2^2 + \gamma\hat{\theta}(u,v)$ with a function $\hat{\theta}$ that promotes the desired structure in U and V provided that $\theta(u,v)$ satisfies the necessary properties in the definition of a rank-1 regularizer. Additional example problems can be found in [55, 56].

Symmetric Factorizations: Assuming that X is a square matrix, it is also possible to learn symmetrical formulations with this framework, as the indicator function $\delta_{u=v}(u,v)$ that requires u and v to be equal is also positively homogeneous. As a result, one can use regularization such as $\theta(u,v) = \delta_{u=v}(u,v) + \|u\|_2^2$ to learn low-rank symmetrical factorizations of X, and adding additional regularization can be done to encourage additional structures. For example $\theta(u,v) = \delta_{u=v}(u,v) + \|u\|_2^2 + \|u\|_1^2 + \delta_{\mathbb{R}_+}(u)$ learns symmetrical factorizations where the factors are required to be non-negative and encouraged to be sparse.

3.3 Problem Formation

Returning to the motivation from the introductory discussion, in this section we describe the link between convex problems (3.6), which offer guarantees of global optimality, and factorized formulations (3.5), which offer additional flexibility in modeling the data structure and recovery of features that can be used in subsequent analysis. Using the matrix factorization regularizer introduced in the previous section, we will consider problems of the form

$$\min_{X,Q} F(X,Q) = \ell(Y,X,Q) + \lambda\Omega_\theta(X). \tag{3.18}$$

Here the term Q allows for modeling additional variables that will not be factorized. For example in robust PCA (RPCA) [63], the Q term can be used to account for sparse outlying entries, and a formulation in which the data is corrupted by both large corruptions and Gaussian noise can be modeled as,

$$\min_{X,Q} F(X,Q)_{RPCA} = \tfrac{1}{2}\|Y - X - Q\|_F^2 + \gamma\|Q\|_1 + \lambda\|X\|_*. \tag{3.19}$$

From this convex function, $F(X,Q)$, one can also consider the closely related non-convex factorized function defined as

$$\min_{r,U,V,Q} f(U,V,Q) = \ell(Y, UV^T, Q) + \lambda \sum_{i=1}^{r} \theta(U_i, V_i) \tag{3.20}$$

where recall the number of columns in the U and V matrices (r) is allowed to be arbitrary. We will assume throughout that $\ell(Y,X,Q)$ is lower semicontinuous, jointly convex w.r.t. (X,Q), and once differentiable w.r.t. X.

3.4 Theoretical Results

Given the non-convex optimization problem (3.20), note that from the definition of $\Omega_\theta(X)$ for any $UV^T = X$ we must have $\Omega_\theta(X) \leq \sum_{i=1}^r \theta(U_i, V_i)$, so this also results in a global lower bound between the convex and non-convex objective functions, i.e., for all $UV^T = X$,

$$F(X,Q) = \ell(Y,X,Q) + \lambda\Omega_\theta(X) \leq \ell(Y, UV^T, Q) + \lambda \sum_{i=1}^{r} \theta(U_i, V_i) = f(U,V,Q). \tag{3.21}$$

From this, if we let X_{opt} denote an optimal solution to the convex optimization problem $\min_{X,Q} F(X,Q)$, then any factorization $UV^T = X_{opt}$ such that $\sum_{i=1}^r \theta(U_i, V_i) = \Omega_\theta(X_{opt})$ will also be an optimal solution to the non-convex optimization problem $\min_{U,V,Q} f(U,V,Q)$. Further this link between the two problems can be analyzed by noting that the subgradient of the matrix regularization function can be characterized as

$$\partial\Omega_\theta(X) = \left\{W : \langle W, X\rangle = \Omega_\theta(X),\ u^T W v \leq \theta(u,v)\ \forall(u,v)\right\}. \tag{3.22}$$

This result will be shown formally in Lemma 1 of the next chapter, but the intuition for the result can be seen by recalling that Ω_θ is a gauge function, so from Theorem 5 the subgradient will have the form $\langle \partial\Omega_\theta(X), X\rangle = \Omega_\theta(X)$ and $\Omega_\theta^\circ(\partial\Omega_\theta(X)) \leq 1$. This is exactly the form of (3.22), with the polar function being given by $\Omega_\theta^\circ(Z) = \sup_{u,v:\theta(u,v)\leq 1} u^T Z v$. Also note that a factorization $UV^T = X$ is an optimal factorization of X - i.e., it achieves the infimum in (3.13), if and only if $\exists W' \in \partial\Omega_\theta(X)$ such that $\sum_{i=1}^r U_i^T W' V_i = \sum_{i=1}^r \theta(U_i, V_i)$. Again, these results will be shown in detail by Lemma 1 in the next chapter and form the foundation of the following result.

Theorem 6 *Given a function $\ell(Y,X,Q)$ which is lower-semicontinuous, jointly convex in (X,Q), and once differentiable w.r.t. X; a rank-1 regularizer θ which satisfies the conditions in definition 13; and a constant $\lambda > 0$, then local minima of the non-convex optimization problem*

$$\min_{U,V,Q} \ell(Y, UV^T, Q) + \lambda \sum_{i=1}^r \theta(U_i, V_i) \tag{3.23}$$

are globally optimal if $(U_i, V_i) = (0,0)$ for some $i \in \{1,\ldots,r\}$.

Proof. Recall from Section 3.3 that the non-convex factorized objective $f(U,V,Q)$ provides a global upper bound for convex objective $F(X,Q)$. The result follows from the fact that a local minimizer of $f(U,V,Q)$ which satisfies the conditions of the theorem also satisfies the first order conditions for optimality of $F(X,Q)$. More specifically, from the characterization of the subgradient described in (3.22), we have that (X,Q) is a global minimum of the convex objective $F(X,Q)$ iff

$$\begin{aligned} -\tfrac{1}{\lambda}\nabla_X \ell(Y,X,Q) \in \partial\Omega_\theta(X) \\ 0 \in \partial_Q \ell(Y,X,Q). \end{aligned} \tag{3.24}$$

If (U,V,Q) is a local minimum of (3.23) then it is necessary that $0 \in \partial_Q \ell(Y, UV^T, Q)$ from first-order optimality. From the characterization of the subgradient of $\Omega_\theta(X)$ given if (3.22), we also have that $-\frac{1}{\lambda}\nabla_X \ell(Y,X,Q) \in \partial\Omega_\theta(X)$ is equivalent to the conditions

$$u^T(-\tfrac{1}{\lambda}\nabla_X \ell(Y,\tilde{U}\tilde{V}^T,Q))v \leq \theta(u,v)\ \forall (u,v) \tag{3.25}$$

$$\sum_{i=1}^r \tilde{U}_i^T(-\tfrac{1}{\lambda}\nabla_X \ell(Y,\tilde{U}\tilde{V}^T,Q))\tilde{V}_i = \sum_{i=1}^r \theta(\tilde{U}_i,\tilde{V}_i). \tag{3.26}$$

for an optimal factorization $X = \tilde{U}\tilde{V}^T$, i.e., $\Omega_\theta(X) = \sum_i \theta(\tilde{U}_i,\tilde{V}_i)$. Note also that the condition in (3.26) can also be equivalently stated as

$$\tilde{U}_i^T(-\tfrac{1}{\lambda}\nabla_X \ell(Y,\tilde{U}\tilde{V}^T,Q))\tilde{V}_i = \theta(\tilde{U}_i,\tilde{V}_i)\ \forall i \in \{1,\ldots,r\}. \tag{3.27}$$

Considering the local minimum (U, V, Q), recall that we have one column pair of (U, V) which is entirely 0, and assume without loss of generality that the final column of (U, V) is the one that is all 0. Then, due to the fact that (U, V, Q) is a local minimum, we have $\forall(u, v)$ there exists $\delta > 0$ such that $\forall \epsilon \in (0, \delta)$

$$\ell(Y, [U_1, \ldots, U_{r-1}, \epsilon^{1/2} u][V_1, \ldots, V_{r-1}, \epsilon^{1/2} v]^T, Q) + \lambda \sum_{i=1}^{r} \theta(U_i, V_i) + \lambda\theta(\epsilon^{1/2} u, \epsilon^{1/2} v) = \tag{3.28}$$

$$\ell(Y, UV^T + \epsilon uv^T, Q) + \lambda \sum_{i=1}^{r} \theta(U_i, V_i) + \epsilon\lambda\theta(u, v) \geq \tag{3.29}$$

$$\ell(Y, UV^T, Q) + \lambda \sum_{i=1}^{r} \theta(U_i, V_i). \tag{3.30}$$

Rearranging terms and using the positive homogeneity of θ, we then have

$$\tfrac{-1}{\lambda}[\ell(Y, UV^T + \epsilon uv^T, Q) - \ell(Y, UV^T, Q)]\epsilon^{-1} \leq \theta(u, v). \tag{3.31}$$

Taking the limit $\epsilon \searrow 0$, note that from the differentiability of $\ell(Y, X, Q)$ w.r.t. X, this gives $\left\langle \tfrac{-1}{\lambda} \nabla_X \ell(Y, UV^T, Q), uv^T \right\rangle \leq \theta(u, v)$ for any (u, v) vector pair, showing (3.25). To show (3.26), note again that because we have a local minimum then for $\epsilon > 0$ and sufficiently small we must have

$$\begin{aligned}&\ell(Y,[(1+\epsilon)^{1/2}U][(1+\epsilon)^{1/2}V]^T,Q)+\lambda\sum_{i=1}^r\theta((1+\epsilon)^{1/2}U_i,(1+\epsilon)^{1/2}V_i)=\\&\ell(Y,UV^T+\epsilon UV^T,Q)+\lambda(1+\epsilon)\sum_{i=1}^r\theta(U_i,V_i)\geq\\&\ell(Y,UV^T,Q)+\lambda\sum_{i=1}^r(U_i,V_i)\end{aligned}\tag{3.32}$$

and also

$$\begin{aligned}&\ell(Y,[(1-\epsilon)^{1/2}U][(1-\epsilon)^{1/2}V]^T,Q)+\lambda\sum_{i=1}^r((1-\epsilon)^{1/2}U_i,(1-\epsilon)^{1/2}V_i)=\\&\ell(Y,UV^T-\epsilon UV^T,Q)+\lambda(1-\epsilon)\sum_{i=1}^r\theta(U_i,V_i)\geq\\&\ell(Y,UV^T,Q)+\lambda\sum_{i=1}^r(U_i,V_i)\end{aligned}\tag{3.33}$$

Rearranging terms and taking the limit $\epsilon\searrow 0$ as before, we get

$$\sum_{i=1}^r\theta(U_i,V_i)\leq\left\langle\tfrac{-1}{\lambda}\nabla_X\ell(Y,UV^T,Q),UV^T\right\rangle\leq\sum_{i=1}^r\theta(U_i,V_i)\implies\tag{3.34}$$

$$\left\langle\tfrac{-1}{\lambda}\nabla_X\ell(Y,UV^T,Q),UV^T\right\rangle=\sum_{i=1}^r U_i^T(\tfrac{-1}{\lambda}\nabla_X\ell(Y,UV^T,Q))V_i=\sum_{i=1}^r\theta(U_i,V_i)\tag{3.35}$$

which completes the result. ■

Note that the above proof also proves a simple corollary that provides sufficient conditions to guarantee global optimality of any point.

Corollary 1 *Given a function $\ell(Y,X,Q)$ which is lower-semicontinuous, jointly con-*

vex in (X, Q), and once differentiable w.r.t. X; a rank-1 regularizer θ which satisfies the conditions in definition 13; and a constant $\lambda > 0$, then any point $(\tilde{U}, \tilde{V}, \tilde{Q})$ is a global minimum of

$$\min_{U,V,Q} \ell(Y, UV^T, Q) + \lambda \sum_{i=1}^{r} \theta(U_i, V_i) \tag{3.36}$$

if it satisfies the following conditions:

1. $0 \in \partial_Q \ell(Y, \tilde{U}\tilde{V}^T, \tilde{Q})$

2. $\tilde{U}_i^T(\frac{-1}{\lambda}\nabla_X \ell(Y, \tilde{U}\tilde{V}^T, \tilde{Q}))\tilde{V}_i = \theta(\tilde{U}_i, \tilde{V}_i)\ \forall i \in \{1, \ldots, r\}$

3. $u^T(\frac{-1}{\lambda}\nabla_X \ell(Y, \tilde{U}\tilde{V}^T, \tilde{Q}))v \leq \theta(u, v)\ \forall (u, v)$.

Condition 1 is fairly easy to verify, as one can hold (U, V) constant and solve a convex optimization problem for Q. Likewise, condition 2 is simple to test, and if a (U_i, V_i) pair exists which does not satisfy the equality, then one can decrease the objective function by scaling (U_i, V_i) by a non-negative constant. Further, for many problems, it is possible to show that points that satisfy first-order optimality will satisfy conditions 1 and 2, such as in the following result.

Proposition 6 *Given a function $\ell(Y, X, Q)$ which is lower-semicontinuous, jointly convex in (X, Q), and once differentiable w.r.t. X; a constant $\lambda > 0$; and two gauge functions $(\sigma_u(u), \sigma_v(v))$, then for $\theta(u, v) = \sigma_u(u)\sigma_v(v)$ or $\theta(u, v) = \frac{1}{2}(\sigma_u(u)^2 + \sigma_v(v)^2)$,* *any first-order optimal point $(\tilde{U}, \tilde{V}, \tilde{Q})$ of the function*

$$\min_{U,V,Q} \ell(Y, UV^T, Q) + \lambda \sum_{i=1}^{r} \theta(U_i, V_i) \tag{3.37}$$

satisfies conditions 1 and 2 of Corollary 1.

Proof. Note that condition 1 is trivially satisfied, as this is simply the first-order optimality requirement w.r.t. Q. Thus, we are left to show that condition 2 is also satisfied. Let $\theta_p(u, v) = \sigma_u(u)\sigma_v(v)$ and $\theta_s(u, v) = \frac{1}{2}(\sigma_u(u)^2 + \sigma_v(v)^2)$. Note that the following are easily shown from basic properties of subgradients of gauge functions

$$\begin{aligned} &\langle u, \partial_u \theta_p(u, v)\rangle = \langle v, \partial_v \theta_p(u, v)\rangle = \theta_p(u, v) \\ &\langle u, \partial_u \theta_s(u, v)\rangle = \sigma_u(u)^2 \\ &\langle v, \partial_v \theta_s(u, v)\rangle = \sigma_v(v)^2. \end{aligned} \tag{3.38}$$

Considering the first-order optimality conditions w.r.t. U_i and V_i, one gets

$$0 \in \nabla_X(Y, \tilde{U}\tilde{V}^T, \tilde{Q})\tilde{V}_i + \lambda \partial_u \theta(\tilde{U}_i, \tilde{V}_i) \tag{3.39}$$

$$0 \in \nabla_X(Y, \tilde{U}\tilde{V}^T, \tilde{Q})^T \tilde{U}_i + \lambda \partial_v \theta(\tilde{U}_i, \tilde{V}_i) \tag{3.40}$$

and left multiplying the two above inclusions by $\tilde{U}_i^T$ and $\tilde{V}_i^T$, respectively gives

$$0 \in \tilde{U}_i^T \nabla_X(Y, \tilde{U}\tilde{V}^T, \tilde{Q})\tilde{V}_i + \lambda \left\langle \tilde{U}_i, \partial_u \theta(\tilde{U}_i, \tilde{V}_i) \right\rangle \tag{3.41}$$

$$0 \in \tilde{V}_i^T \nabla_X(Y, \tilde{U}\tilde{V}^T, \tilde{Q})^T \tilde{U}_i + \lambda \left\langle \tilde{V}_i, \partial_v \theta(\tilde{U}_i, \tilde{V}_i) \right\rangle . \tag{3.42}$$

Since this is true for all $(\tilde{U}_i, \tilde{V}_i)$ pairs, substituting (3.38) and rearranging terms then shows that condition 2 of Corollary 1 is satisfied for both $\theta_s(u, v)$ and $\theta_p(u, v)$, completing the result. ■

From this result and the above discussion, it is clear that the primary challenge in verifying if a given point is globally optimal is to test if condition 3 of Corollary 1 is satisfied. This is known as the polar problem and is discussed in detail below.

3.4.1 Polar Problem

Note that because the overall matrix factorization optimization problem is non-convex, first-order optimality is not sufficient to guarantee a local minimum, and to apply these results in practice one needs to verify that condition 3 from Corollary 1 is satisfied. This problem is known as the polar problem and is a generalization of the concept of a dual norm. In particular given a matrix factorization regularizer $\Omega_\theta(X)$, the polar function of Ω_θ is denoted as Ω_θ°, defined as

$$\Omega_\theta^\circ(Z) = \sup_{u,v} u^T Z v \quad \text{s.t.} \quad \theta(u, v) \leq 1, \tag{3.43}$$

and condition 3 of Corollary 1 corresponds to $\Omega_\theta^\circ(\frac{-1}{\lambda}\nabla_X \ell(Y, \tilde{U}\tilde{V}^T, \tilde{Q})) \leq 1$. The difficulty of calculating the polar problem heavily depends on the particular choice of the θ function. For example for $\theta(u,v) = \|u\|_1\|v\|_1$ the polar problem reduces to simply finding the largest entry of Z in absolute value, while for $\theta(u,v) = \|u\|_\infty\|v\|_\infty$ solving the polar problem is known to be NP-hard [64].

While for general $\theta(u,v)$ functions it is not necessarily known how to efficiently solve the polar problem, given a point $(\tilde{U}, \tilde{V}, \tilde{Q})$ that satisfies conditions 1 and 2 of Corollary 1, the value of the polar problem solution at a given point and how closely the polar problem can be approximated provides a bound on how far a particular point is from being globally optimal. This bound is based on the following proposition:

Proposition 7 *Given a function $\ell(Y,X,Q)$ which is lower-semicontinuous, jointly convex in (X,Q), and once differentiable w.r.t. X; a rank-1 regularizer θ which satisfies the conditions in definition 13; and a constant $\lambda > 0$, let $F(X,Q) = \ell(Y,X,Q) + \lambda\Omega_\theta(X)$. Then for any point $(\tilde{U}, \tilde{V}, \tilde{Q})$ that satisfies conditions 1 and 2 of Corollary 1, we have the following bound*

$$\begin{aligned} &\ell(Y, \tilde{U}\tilde{V}^T, \tilde{Q}) + \lambda \sum_i \theta(\tilde{U}_i, \tilde{V}_i) - F(X_{opt}, Q_{opt}) \leq \\ &\lambda\Omega_\theta(X_{opt})[\Omega_\theta^\circ(\tfrac{-1}{\lambda}\nabla_X \ell(Y, \tilde{U}\tilde{V}^T, \tilde{Q})) - 1] - \tfrac{m_X}{2}\|\tilde{U}\tilde{V}^T - X_{opt}\|_F^2 - \tfrac{m_Q}{2}\|\tilde{Q} - Q_{opt}\|_F^2 \end{aligned} \tag{3.44}$$

where $m_X \geq 0$ and $m_Q \geq 0$ denote the constants of strong-convexity of ℓ w.r.t. X

and Q, respectively, (note that both m constants can be 0 if ℓ is not strongly convex) and (X_{opt}, Q_{opt}) denotes a global minimizer of $F(X,Q)$.

Proof. Let $\tilde{X} = \tilde{U}\tilde{V}^T$. From (strong) convexity of ℓ, we have

$$\begin{aligned}\ell(Y, X_{opt}, Q_{opt}) \geq & \ell(Y, \tilde{X}, \tilde{Q}) + \tfrac{m_X}{2}\|\tilde{X} - X_{opt}\|_F^2 + \tfrac{m_Q}{2}\|\tilde{Q} - Q_{opt}\|_F^2 \\ & + \left\langle \nabla_X \ell(Y, \tilde{X}, \tilde{Q}), X_{opt} - \tilde{X} \right\rangle + \left\langle \partial_Q \ell(Y, \tilde{X}, \tilde{Q}), Q_{opt} - \tilde{Q} \right\rangle.\end{aligned} \tag{3.45}$$

From condition 1 of Corollary 1 we can take $0 \in \partial_Q \ell(Y, \tilde{X}, \tilde{Q})$, and from condition 2 we have $\left\langle -\nabla_X \ell(Y, \tilde{X}, \tilde{Q}), \tilde{X} \right\rangle = \lambda \sum_i \theta(\tilde{U}_i, \tilde{V}_i)$. Applying these facts and rearranging terms gives

$$\begin{aligned}& \ell(Y, \tilde{X}, \tilde{Q}) - \ell(Y, X_{opt}, Q_{opt}) + \lambda \sum_i \theta(\tilde{U}_i, \tilde{V}_i) \\ & \leq \lambda \left\langle \tfrac{-1}{\lambda} \nabla_X \ell(Y, \tilde{X}, \tilde{Q}), X_{opt} \right\rangle - \tfrac{m_X}{2}\|\tilde{X} - X_{opt}\|_F^2 - \tfrac{m_Q}{2}\|\tilde{Q} - Q_{opt}\|_F^2.\end{aligned} \tag{3.46}$$

Recall that from polar duality we also have $\forall (X, Z)$, $\langle X, Z \rangle \leq \Omega_\theta(X)\Omega_\theta^\circ(Z)$, which implies

$$\left\langle \tfrac{-1}{\lambda} \nabla_X \ell(Y, \tilde{X}, \tilde{Q}), X_{opt} \right\rangle \leq \Omega_\theta(X_{opt})\Omega_\theta^\circ(\tfrac{-1}{\lambda} \nabla_X \ell(Y, \tilde{X}, \tilde{Q})). \tag{3.47}$$

Substituting this into (3.46) we then have

$$\begin{aligned}& \ell(Y, \tilde{X}, \tilde{Q}) - \ell(Y, X_{opt}, Q_{opt}) + \lambda \sum_i \theta(\tilde{U}_i, \tilde{V}_i) \\ & \leq \lambda \Omega_\theta(X_{opt})\Omega_\theta^\circ(\tfrac{-1}{\lambda} \nabla_X \ell(Y, \tilde{X}, \tilde{Q})) - \tfrac{m_X}{2}\|\tilde{X} - X_{opt}\|_F^2 - \tfrac{m_Q}{2}\|\tilde{Q} - Q_{opt}\|_F^2.\end{aligned} \tag{3.48}$$

and subtracting $\lambda\Omega_\theta(X_{opt})$ from both sides of the above inequality completes the result. ■

There are a few interpretations one can draw from the above proposition. First, if $X_{opt} = 0$, then the only (U, V) pair that will satisfy conditions 1 and 2 of Corollary 1 is the global optimum $UV^T = 0$. Second, for $X_{opt} \neq 0$ recall that $\Omega_\theta(X_{opt}) > 0$ and $\Omega_\theta^\circ(\frac{-1}{\lambda}\nabla_X \ell(Y, \tilde{U}\tilde{V}^T, \tilde{Q})) \geq 1$, since if condition 2 of Corollary 1 is satisfied then the polar is clearly at least equal to 1 by definition of the polar. Further, from Theorem 6, if we can find any (u, v) pair such that $\theta(u, v) \leq 1$ and $u^T(\frac{-1}{\lambda}\nabla_X \ell(Y, UV^T, Q))v > 1$, then we can decrease the objective function by appending (u, v) to the factorization, i.e., $(U, V) \to ([U\ \epsilon u], [V\ \epsilon v])$ will decrease the objective for some $\epsilon > 0$. As a result, we always have a means to decrease the objective function by either doing gradient descent or adding a (u, v) pair to the factorization, unless we arrive at a first-order optimal point and we cannot find a (u, v) pair such that $u^T(\frac{-1}{\lambda}\nabla_X \ell(Y, UV^T, Q))v > 1$. If the polar is truly greater than 1, then the $[\Omega_\theta^\circ(\frac{-1}{\lambda}\nabla_X \ell(Y, \tilde{U}\tilde{V}^T, \tilde{Q})) - 1]$ in the above proposition effectively measures the error between the true value of the polar and our lower-bound estimate of the polar. Further, the maximum difference between a first-order optimal point and the global minimum is upper bounded by the value of the polar at that point, and if the loss function ℓ is strongly convex, the error in the objective function is decreased further. As a result, if one can guarantee solutions to the polar problem to within a given error level or provide an upper-bound on the polar problem, one can also guarantee solutions that are within a given error level of the global optimum.

A final interpretation of Proposition 7 that can be made is to note that the final condition of Corollary 1 is essentially a check that the size of the representation (i.e., the number of columns in U and V) is sufficiently large to represent the global optimum. If, instead, we find a local minimum with a smaller representation than the global optimum, $r < r_{opt}$, where r_{opt} denotes the number of columns in the global optimum, then the value of the $[\Omega_\theta^\circ(\frac{-1}{\lambda}\nabla_X \ell(Y, \tilde{U}\tilde{V}^T, \tilde{Q})) - 1]$ bounds how far from the global minimum we are by using a more compact representation (i.e., using only r instead of r_{opt} columns).

As a concrete example of these ideas, consider the case where $\theta(u,v) = \|u\|_2\|v\|_2$. Recall that this choice of θ gives the nuclear norm, $\Omega_\theta(X) = \|X\|_*$. From this, the polar function then is given by

$$\|Z\|_*^\circ = \sup_{u,v} u^T Z v \quad \text{s.t.} \quad \|u\|_2\|v\|_2 \leq 1 \tag{3.49}$$

$$= \sup_{u,v} u^T Z v \quad \text{s.t.} \quad \|u\|_2 \leq 1, \ \|v\|_2 \leq 1 \tag{3.50}$$

$$= \sigma_{max}(Z), \tag{3.51}$$

where $\sigma_{max}(Z)$ denotes the largest singular value of Z (and thus $\|\cdot\|_*^\circ$ is the spectral norm). In this case, given any first order optimal point $(\tilde{U}, \tilde{V}, \tilde{Q})$, then Proposition 7 guarantees that the distance of the current point from the global minimum is bounded by $\lambda\|X_{opt}\|_*[\sigma_{max}(\frac{-1}{\lambda}\nabla_X \ell(Y, \tilde{U}\tilde{V}^T, \tilde{Q})) - 1]$. If $(\tilde{U}, \tilde{V}, \tilde{Q})$ is a global minimizer, then the largest singular value term will be equal to 1 (and hence the bound is 0), while if the largest singular value term is greater than 1 this indicates that $(\tilde{U}, \tilde{V})$ do not have sufficiently many columns to represent the global optimum, and the size of the representation should be increased. Further, by appending the largest singular vector pair (u, v) to the factorization $U \leftarrow [\tilde{U} \ \tau u]$ and $V \leftarrow [\tilde{V} \ \tau v]$ (as this is the vector pair that achieves the supremum of the polar function) will be guaranteed to reduce the objective function for some step size $\tau > 0$. This strategy is described in the meta-algorithm given by Algorithm 2.

3.4.1.1 Upper Bounding the Polar

In many cases, it is possible to derive semidefinite relaxations of the polar problem that upper-bound the polar solution. Specifically, note that (3.43) is equivalently reformulated as

$$\Omega_\theta^\circ(Z) = \sup_{u,v} \tfrac{1}{2} \left\langle \begin{bmatrix} 0 & Z \\ Z^T & 0 \end{bmatrix}, \begin{bmatrix} uu^T & uv^T \\ vu^T & vv^T \end{bmatrix} \right\rangle \ \text{ s.t. } \ \theta(u,v) \leq 1. \tag{3.52}$$

If we make the change of variables $M = [u; v][u; v]^T$, the problem is equivalent to optimizing over rank-1 semidefinite matrices M, provided there exists an equivalent function $\theta'(M)$ to enforce the constraint $\theta(u, v) \leq 1$ if M is a rank-1 matrix. For example, consider the case $\theta(u, v) = \frac{1}{2}(\|u\|_F^2 + \|v\|_F^2)$, which gives $\theta(u, v) = \frac{1}{2}(\mathrm{Tr}(uu^T) + \mathrm{Tr}(vv^T)) = \frac{1}{2}\mathrm{Tr}(M)$, so we have the following equivalent problems

$$\begin{aligned}\Omega_\theta^\circ(Z) =& \max_{u,v} u^T Z v \;\; \text{s.t.} \;\; \tfrac{1}{2}(\text{Tr}(uu^T) + \text{Tr}(vv^T)) \leq 1 \\ =& \max_M \tfrac{1}{2} \left\langle \begin{bmatrix} 0 & Z \\ Z^T & 0 \end{bmatrix}, M \right\rangle \;\; \text{s.t.} \;\; \text{rank}(M) = 1, \; \tfrac{1}{2}\text{Tr}(M) \leq 1, \; M \succeq 0.\end{aligned} \tag{3.53}$$

By removing the rank$(M) = 1$ constraint we then have a convex optimization problem on positive semidefinite matrices that upper-bounds the polar problem,

$$\Omega_\theta^\circ(Z) \leq \max_M \tfrac{1}{2} \left\langle \begin{bmatrix} 0 & Z \\ Z^T & 0 \end{bmatrix}, M \right\rangle \;\; \text{s.t.} \;\; \tfrac{1}{2}\text{Tr}(M) \leq 1, \; M \succeq 0 \tag{3.54}$$

and if the solution to the above problem results in a rank-1 solution matrix M, which in this special case of $\theta(u, v)$ can be shown to be true via the S-procedure [9], then the inequality becomes an equality and the desired (u, v) factors can be recovered from the largest singular vector of M.

This same idea can be extended to more general $\theta(u, v)$ regularization functions [55] and has been used in techniques such as sparse PCA [65]. A few example functions on vectors x and their equivalent function on xx^T are provided in Table 3.2, and these equivalences can be used to derive $\theta'(M)$ functions from a given $\theta(u, v)$ function.

While, unfortunately, in general there is no guarantee that the solution to the semidefinite relaxation will be a rank-1 M matrix for an arbitrary $\theta(u, v)$ regulariza-

Table 3.2: Equivalent forms of polar problem regularizers.

$f(x)$	$F(xx^T)$
$\|x\|_F^2$	$\mathrm{Tr}(xx^T)$
$\|x\|_1^2$	$\|xx^T\|_1$
$\|x\|_\infty^2$	$\|xx^T\|_\infty$
$\|Ax\|_1^2$	$\|Axx^TA^T\|_1$
$\|Ax\|_1\|x\|_F$	$\sum_i \|(xx^TA)_i\|_F$
$\delta_{\mathbb{R}_+}(x)$	$\delta_{\mathbb{R}_+}(xx^T)$

tion function, for some cases of $\theta(u, v)$ one can prove bounds about how close the upper-bound of the polar obtained from semidefinite relaxation will be to the true value of the polar [55].

3.5 Minimization Algorithm

Before we begin the discussion of the algorithm, note that in addition to the conditions included in Theorem 6, the particular method we present here assumes that the gradients of the loss function $\ell(Y, UV^T, Q)$ w.r.t. U and w.r.t. V (denoted as $\nabla_U \ell(Y, UV^T, Q)$ and $\nabla_V \ell(Y, UV^T, Q)$, respectively) are Lipschitz continuous (i.e. the gradient w.r.t. U is Lipschitz continuous for any fixed value of V and vice versa). Under these assumptions on ℓ, the bilinear structure of our objective function (3.20) gives convex subproblems if we update U or V independently while holding the other fixed, making an alternating minimization strategy efficient and easy to implement. Further, we assume that $\ell(Y, UV^T, Q) = \hat{\ell}(Y, UV^T, Q) + H(Q)$ where $\hat{\ell}(Y, UV^T, Q)$ is a convex, once differentiable function of Q with Lipschitz continuous gradient with constants L_Q^k and $H(Q)$ is convex but possibly non-differentiable[3].

The updates to our variables are made using accelerated proximal-linear steps similar to the FISTA algorithm, which entails solving a proximal operator of an extrapolated gradient step to update each variable [66, 67]. The general structure of the alternating updates we use is given in Algorithm 1, and the key point is that to update either U, V, or Q the primary computational burden lies in calculating the gradient of the loss function and then calculating a proximal operator. The structure of the non-differentiable term in (3.20) allows the proximal operators for U and V to be separated into columns, greatly reducing the complexity of calculating the proximal operator and offering the potential for parallelization.

3.5.1 Proximal Operators of Structured Factors

Recall from the introductory discussion that one means to induce general structure in the factorized matrices is to regularize the columns of a factorized matrix with an l_2 norm, to limit the rank of the solution, plus a general gauge function, to induce specific structure in the factors. For example, potential forms of the rank-1 regularizers could be of the form $\theta(u, v) = \|u\|_2\|v\|_2 + \gamma\sigma_u(u)\sigma_v(v)$ or $\theta(u, v) = (\|u\|_2 + \gamma_u\sigma_u(u))(\|v\|_2 + \gamma_v\sigma_v(v))$, where the σ_u and σ_v gauge functions

[3] Note that the assumption that there is a component of the objective function that is differentiable w.r.t. Q, $\hat{\ell}$, is only needed for use the particular update strategy we describe here. In general one could also optimize objective functions that are totally non-differentiable w.r.t. Q (but which do need to be convex w.r.t. Q) by doing a full minimization w.r.t. Q at each iteration instead of just a proximal gradient update. See [66] for more details.

Algorithm 1 (Structured Matrix Factorization)

```
Input: $Y$, $U^0$, $V^0$, $Q^0$, $\lambda$, NumIter
Initialize $\hat{U}^1 = U^0$, $\hat{V}^1 = V^0$, $\hat{Q}^1 = Q^0$, $t^0 = 1$
for $k = 1$ to NumIter do
    \\Calculate gradient of loss function w.r.t. $U$
    \\evaluated at the extrapolated point $\hat{U}$
    $G_U^k = \nabla_U \ell(Y, \hat{U}^k (V^{k-1})^T, Q^{k-1})$
    $P = \hat{U}^k - G_U^k / L_U^k$
    \\Calculate proximal operator of $\theta$
    \\for every column of $U$
    for $i = 1$ to number of columns in $A$ do
        $U_i^k = \mathbf{prox}_{\lambda\theta(\cdot, V_i^{k-1})/L_U^k}(P_i)$
    end for
    \\Repeat similar process for $V$
    $G_V^k = \nabla_V \ell(Y, U^k (\hat{V}^k)^T, Q^{k-1})$
    $W = \hat{V}^k - G_V^k / L_V^k$
    for $i = 1$ to number of columns in $V$ do
        $V_i^k = \mathbf{prox}_{\lambda\theta(U_i^k, \cdot)/L_V^k}(W_i)$
    end for
    \\Repeat again for $Q$
    $G_Q^k = \nabla_Q \hat{\ell}(Y, U^k (V^k)^T, \hat{Q}^k)$
    $R = \hat{Q}^k - G_Q^k / L_Q^k$
    $Q^k = \mathbf{prox}_{H(Y, U^k (V^k)^T, \cdot)/L_Q^k}(R)$
    \\Update extrapolation based on prior iterates
    \\Check if objective decreased
    if $obj(U^k, V^k, Q^k) < obj(U^{k-1}, V^{k-1}, Q^{k-1})$ then
        \\The objective decreased, update extrapolation
        $t^k = (1 + \sqrt{1 + 4(t^{k-1})^2})/2$
        $\mu = (t^{k-1} - 1)/2$
        $\mu_U = \min\{\mu, \sqrt{L_U^{k-1}/L_U^k}\}$
        $\mu_V = \min\{\mu, \sqrt{L_V^{k-1}/L_V^k}\}$
        $\mu_Q = \min\{\mu, \sqrt{L_Q^{k-1}/L_Q^k}\}$
        $\hat{U}^{k+1} = U^k + \mu_U(U^k - U^{k-1})$
        $\hat{V}^{k+1} = V^k + \mu_V(V^k - V^{k-1})$
        $\hat{Q}^{k+1} = Q^k + \mu_Q(Q^k - Q^{k-1})$
    else
        \\The objective didn't decrease.
        \\Run again without extrapolation.
        $t^k = t^{k-1}$
        $\hat{U}^{k+1} = U^{k-1}$
        $\hat{V}^{k+1} = V^{k-1}$
        $\hat{Q}^{k+1} = Q^{k-1}$
    end if
end for
```

are chosen to encourage specific properties in U and V, respectively. In this case, to apply Algorithm 1 we need a way to solve the proximal operator of the l_2 norm plus a general gauge function. While the proximal operator of the l_2 norm is simple to calculate, even if the proximal operator of the gauge function is know, in general the proximal operator of the sum of two functions is not necessarily easy to compute or related to the proximal operators of the individual functions. Fortunately, however, the following result shows that for the sum of the l_2 norm plus a general gauge function, the proximal operator can be solved by sequentially calculating the two proximal operators.

Theorem 7 *Let σ_C be any gauge function. The proximal operator of $\theta(x) = \lambda\sigma_C(x) + \lambda_2\|x\|_2$ is the composition of the proximal operator of the l_2 norm and the proximal operator of σ_C, i.e., $\mathbf{prox}_\theta(y) = \mathbf{prox}_{\lambda_2\|\cdot\|_2}(\mathbf{prox}_{\lambda\sigma_C}(y))$.*

Proof. Note that we can equivalently solve the proximal operator by introducing another variable subject to an equality constraint,

$$\mathbf{prox}_\theta(y) = \operatorname*{arg\,min}_{x,z:x=z} \frac{1}{2}\|y - x\|_2^2 + \lambda\sigma_C(z) + \lambda_2\|x\|_2. \tag{3.55}$$

This gives the Lagrangian

$$L(x, z, \gamma) = \frac{1}{2}\|y - x\|_2^2 + \lambda\sigma_C(z) + \lambda_2\|x\|_2 + \langle \gamma, x - z \rangle. \tag{3.56}$$

Minimizing the Lagrangian w.r.t. z, we obtain the negative Fenchel dual of $\lambda\sigma_C$, which is an indicator on the polar set

$$\min_z \lambda\sigma_C(z) - \langle\gamma, z\rangle = -(\lambda\sigma_C(\gamma))^* = -\delta^\circ_{\lambda C}(\gamma). \tag{3.57}$$

Minimizing the Lagrangian w.r.t. x, we obtain

$$\min_x \frac{1}{2}\|y - x\|_2^2 + \lambda_2\|x\|_2 + \langle\gamma, x\rangle = \tag{3.58}$$

$$\min_x \frac{1}{2}\|y - \gamma - x\|_2^2 + \lambda_2\|x\|_2 + \langle\gamma, y\rangle - \frac{1}{2}\|\gamma\|_2^2 = \tag{3.59}$$

$$\begin{cases} \frac{1}{2}\|y\|_2^2 & \|y-\gamma\|_2 \le \lambda_2 \\ \frac{1}{2}\|y\|_2^2 - \frac{1}{2}\left(\|y-\gamma\|_2 - \lambda_2\right)^2 & \text{else} \end{cases} \tag{3.60}$$

where the minimum value for x is achieved at $x = \mathbf{prox}_{\lambda_2\|\cdot\|_2}(y - \gamma)$. The relation between (3.58) and (3.59) is easily seen by expanding the quadratic terms, while the relation between (3.59) and (3.60) is given by the fact that (3.59) is the standard proximal operator for the l_2 norm plus terms that do not depend on x. Plugging the solution of the proximal operator of the l_2 norm (noting that the l_2 norm is self dual) into (3.59) gives (3.60). The dual of the original problem thus becomes maximizing (3.60) w.r.t. γ subject to $\sigma^\circ_C(\gamma) \le \lambda$. We note that (3.60) is monotonically non-decreasing as $\|y - \gamma\|_2$ decreases, so the dual problem is equivalent to minimizing $\|y - \gamma\|_2$ (or equivalently $\|y - \gamma\|_F^2$) subject to $\sigma^\circ_C(\gamma) \le \lambda$. Combining these results with the primal-dual relation $x = \mathbf{prox}_{\lambda_2\|\cdot\|_2}(y-\gamma)$, we have

$$\mathbf{prox}_\theta(y) = \mathbf{prox}_{\lambda_2\|\cdot\|_2}(y-\gamma_{opt}), \tag{3.61}$$

where γ_{opt} is the solution to the optimization problem

$$\gamma_{opt} = \arg\min_{\gamma} \|y-\gamma\|_F^2 \ \text{ s.t. } \ \sigma_C^\circ(\gamma) \leq \lambda. \tag{3.62}$$

Recall that from the Moreau identity, the proximal operator of the $\lambda\sigma_C$ gauge is given by

$$\mathbf{prox}_{\lambda\sigma_C}(y) = y - \arg\min_{\gamma} \|y-\gamma\|_F^2 \ \text{ s.t. } \ \sigma_C^\circ(\gamma) \leq \lambda, \tag{3.63}$$

which completes the result, as the above equation implies $\mathbf{prox}_{\lambda\sigma_C}(y) = y - \gamma_{opt}$. ■

Combining these results with Theorem 6 and our previously discussed points, we now have a potential strategy to search for structured low-rank matrix factorizations as we can guarantee global optimality if we can find a local minimum with an all-zero column in (U,V), and the above proposition provides a means to efficiently solve proximal operator problems that one typically encounters in structured factorization formulations. However, there are a few critical caveats to note about the optimization problem. In the next section we discuss these caveats along with a potential meta-algorithm to address them.

3.5.2 Optimization Meta-Algorithm

While Algorithm 1 provides an easily implementable algorithm to perform structured matrix factorization, a critical caveat of the algorithm is that alternating minimization does not necessarily guarantee convergence to a local minimum. It has been shown that, subject to a few conditions, block convex functions will globally converge to a Nash equilibrium point via the alternating minimization algorithm we use here, and any local minima must also be a Nash equilibrium point (although unfortunately the converse is not true) [66]. A Nash equilibrium point implies that we satisfy first order optimality, so in general Algorithm 1 will converge to a point that satisfies conditions 1-2 of Corollary 1, but perhaps not condition 3. From the discussion of the polar problem above, one can search for (u, v) pairs such that $\theta(u, v) \leq 1$ and $u^T(\frac{-1}{\lambda}\nabla_X \ell(Y, UV^T, Q))v > 1$ which can then be used to decrease the objective function by appending the (u, v) pair to the factorization and the algorithm can be rerun from that new location. This approach is outlined in the meta-algorithm described in Algorithm 2.

Note that the main computational challenge from a theoretical standpoint is to find a (u, v) pair such that $\theta(u, v) \leq 1$ and $u^T(\frac{-1}{\lambda}\nabla_X \ell(Y, UV^T, Q))v > 1$, as in general to find such a pair (if a pair exists) we would need to be able to solve the polar problem, as discussed above. However, from Proposition 7, as it becomes harder to find (u, v) pairs that can be used to decrease the objective function (i.e., the value of the polar function moves closed to 1) we are also guaranteed to be closer to the

Algorithm 2 (Structured Matrix Factorization Meta-Algorithm)

```
input Initialization for variables, $(U_{init}, V_{init}, Q_{init})$
  while Not Converged do
    Do local descent via Algorithm 1 until arriving at a critical point $(\tilde{U}, \tilde{V}, \tilde{Q})$.
    Search for $(u, v)$ such that $\theta(u, v) \leq 1$ and $u^T(\frac{-1}{\lambda}\nabla_X \ell(Y, \tilde{U}\tilde{V}^T, \tilde{Q}))v > 1$.
    if $(u, v)$ found then
      Choose a step size $\tau$ by line search and append $(u, v)$ to the factorization $U = [\tilde{U} \tau u]$, $V = [\tilde{V} \tau v]$.
    else
      Return $(\tilde{U}, \tilde{V}, \tilde{Q})$.
    end if
  end while
```

global minimum.

3.6 Conclusions

We have proposed a highly flexible approach to structured matrix factorization, which allows specific structure to be promoted directly on the factors. While our proposed formulation is not jointly convex in all of the variables, we have shown that under certain criteria a local minimum of the factorization is sufficient to find a global minimum of the product, offering the potential to solve the factorization using a highly reduced set of variables.

Chapter 4

Generalized Factorizations

Models involving factorization or decomposition are ubiquitous across a wide variety of technical fields and application areas. As an example illustrated by the previous chapter, many forms of ***matrix factorization***, such as Principle Component Analysis, Non-Negative Matrix Factorization, and Sparse Dictionary Learning, have been developed and achieved considerable empirical success [21, 26, 27]. However, it was noted that common to almost all matrix factorization formulations is the significant disadvantage that the associated optimization problems are typically non-convex in the factorized space due to the bilinear form of the matrix product.

This issue speaks to an apparent dichotomy one is confronted with when choosing a model for a particular problem: Should the problem be approached with a non-convex model which affords greater modeling flexibility and is perhaps better suited for the problem at hand but leads to significant optimization challenges, or should the problem be relaxed into a convex form which provides a set of well developed optimization tools, guarantees of global optimality, and robustness to choice of initialization? Seminal work over the past decade in fields such as compressed sensing and matrix completion has shown that for problems satisfying certain requirements, solutions to convex relaxations of non-convex problems will faithfully recover the solution of the non-convex problem [15, 68–70], which naturally lead one to question if more general convex relaxations are possible. However, as was the case in matrix factorization, solving for X in a relaxed problem is often unsatisfactory at a fundamental level, since in many factorization problems we are interested in finding the factors (U, V) themselves, which implies that even if we have a solution X_{opt} for the relaxed problem, we must still solve yet another non-convex factorization problem to find (U, V). In the case of low-rank matrix factorization, one is fortunate in the sense that since there is no need to enforce any structure on U or V, efficient algorithms (such as singular value decomposition) exist to solve the non-convex factorization problem given X_{opt}. For more general problems (including those possibly beyond matrix factorization), this quickly fails to be a viable solution, and one is forced to consider other options.

To address these issues, in this chapter we consider the task of solving non-convex optimization problems directly in a factorized space with potentially more than 2 factors while using ideas inspired from the convex relaxation of matrix factorizations as a means to analyze the non-convex factorization problem. This framework includes matrix factorization as a special case but also applies much more broadly to a wide range of non-convex optimization problems, several of which we describe below.

4.1 Generalized Factorization

As we alluded to above, the often unavoidable challenge in optimizing factorization problems is the fact that the variables we wish to optimize undergo a convexity destroying transformation (or mapping). In the case of matrix factorization, this takes the form of the matrix product $(U, V) \to UV^T$, but a natural generalization is to consider an arbitrary convexity destroying mapping of the variables $(X^1, \ldots, X^K) \to \Phi(X^1, \ldots, X^K)$, where now we might be interested in optimizing over K blocks of variables, for some $K \geq 1$.

For example, ***tensor factorization*** models provide a natural extension to matrix factorization and have been employed in a wide variety of applications [28, 71]. The resulting optimization problem is similar to matrix factorization, with the difference that we now consider more general factorizations which decompose a multidimensional tensor $Y \approx \Phi(X^1, \ldots, X^K)$ into a set of K different factors $(X^1, \ldots, X^K)$, where each factor is also possibly a multidimensional tensor and Φ is an arbitrary multilinear mapping; i.e., Φ is a linear function of each X^i term if the other X^j terms, $i \neq j$, are held constant. Clearly tensor factorization is a generalization of matrix factorization by taking $(X^1, X^2) = (U, V)$ and $\Phi(U, V) = UV^T$. Moreover, similar to matrix factorization, the optimization problem will typically be non-convex regardless of the choice of regularization function, Θ, or loss function, ℓ, due to the presence of the multilinear mapping Φ.

While the tensor factorization framework is very general with regards to the dimensionality of the data and the factors, the mapping Φ from the factorized space to the output space (the codomain of Φ) is typically assumed to be multilinear. However, if we consider more general mappings from the factorized space into the output space (i.e., Φ mappings which are not restricted to be multilinear) then we can capture a much broader array of models in the “factorized model” family. For example, in ***deep neural network training*** the output of the network is typically generated by applying an alternating series of linear and non-linear functions. More concretely, if one is given training data consisting of N data points of dimension d, $V \in \mathbb{R}^{N \times d}$, the output of the network in response to the training data is described by the mapping

$$\Phi(X^1, \ldots, X^K) = \psi_K(\psi_{K-1}(\ldots \psi_2(\psi_1(VX^1)X^2) \ldots X^{K-1})X^K), \tag{4.1}$$

where each X^i factor (the variables we are trying to optimize) is an appropriately sized matrix which contains the connection weight coefficients between layers $i-1$ and i of the network, and the $\psi_i(\cdot)$ functions apply some form of non-linearity after each matrix multiplication, e.g., a sigmoid function, rectification, max-pooling. Note that although here we have shown the linear operations to be simple matrix multiplications for notational simplicity (which implies each layer is fully connected), this is easily generalized to other linear operators (e.g., in a convolutional network each linear operator could be a set of convolutions with a group of various convolution kernels with parameters contained in the $(X^1, \dots, X^K)$ variables).

Clearly there is an extremely broad range of possible Φ mappings that can destroy convexity, so the focus of this paper will be on one particular family of Φ mappings which captures many problems of interest (such as those described above) and allows for an analysis of sufficient conditions to guarantee global optimality of the non-convex optimization problem.

4.1.1 Contributions

The primary goal of this chapter is to consider non-convex optimization problems of the form

$$\min_{X^1,\dots,X^K,Q} \ell(Y, \Phi(X^1, \dots, X^K), Q) + \lambda\Theta(X^1, \dots, X^K) + H(Q), \tag{4.2}$$

where it is assumed that $\ell(Y, X, Q)$ is jointly convex w.r.t. (X, Q) and once differentiable, $H(Q)$ is convex w.r.t. Q, but the overall problem is non-convex due to the convexity destroying mapping $X = \Phi(X^1, \dots, X^K)$ and possibly non-convex $\Theta(X^1, \dots, X^K)$. Given a non-convex factorization problem of the form in (4.2), our first contribution is to show that if Φ and Θ satisfy a few basic properties then (4.2) can be recast as the convex problem

$$\min_{X,Q} \ell(Y, X, Q) + \lambda \Omega_{\Phi,\Theta}(X) + H(Q), \tag{4.3}$$

where $\Omega_{\Phi,\Theta}$ is a convex function of $X = \Phi(X^1, \ldots, X^K)$ derived from Φ and Θ. However, as mentioned previously in the context of matrix factorization, solving a convex relaxation of the original factorization problem in the output space of the mapping Φ does not achieve our goal, as we still do not know the desired factorization $(X^1, \ldots, X^K)$ such that $X_{opt} = \Phi(X^1, \ldots, X^K)$. For example, in neural network training the output of Φ is simply the response of the network to the training data, and without knowing the factorized variables $(X^1, \ldots, X^K)$ that describe the network weights it is impossible to apply new input data to the network. As a result, we will tailor our results to the non-convex factorization problem (4.2) and use the convex re-formulation (4.3) simply as an analysis tool.

Using this convex framework we are then able to show that local minima of the non-convex factorization problem achieve the global minimum if they satisfy a simple condition. Further, we also show that if the non-convex problem is initialized with factorized variables of sufficient dimension (e.g., in matrix factorization the number of columns in U and V is sufficiently large; in neural networks the size of network hidden layers is sufficiently large), then from any initialization of the factorized variables there must always exist a non-increasing path to a global minimizer and a global minimizer can always be found from local descent.

Before proceeding further, we pause for a moment to clarify from the outset what our results will and also *will not* imply. First, those wishing to apply our results in practice should be cautioned that our results apply to local minima of the non-convex objective function, not critical points. For non-convex optimization problems finding a local minimum can still be an NP-hard problem in general, and many optimization methods can only ensure convergence to a critical point of a general non-convex problem. That being said, however, our results guarantee that the optimization landscape is significantly simplified for the class of non-convex problems that can be captured in our framework. Figure 4.1 provides a cartoon depiction of what our results imply in one dimension. In the left panel, a few possible critical points of a non-convex function are shown in red; these can include single points (b,e,g,h,i) or entire regions of the function domain (a,c,d,f). The single point local maxima (e,g) are of little concern from an optimization standpoint as any reasonable optimization method will avoid these points with overwhelming probability, but all of the other critical points/regions, except for the two global optima (b and d), present possible failure points for an optimization method based on local descent. The right panel of figure 4.1 shows a depiction of what is guaranteed by our framework. First, local minima such as (f) and (h), which require that we must increase the objective to escape from them, are guaranteed to not exist. Second, if we are on one of the

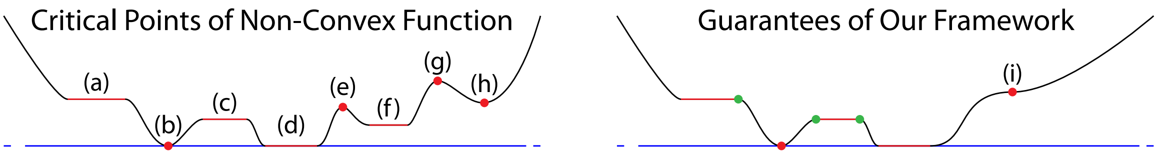
Critical Points of Non-Convex Function
(a)
(b)
(c)
(d)
(e)
(f)
(g)
(h)
Guarantees of Our Framework
(i)

Figure 4.1: *Left:* Example critical points of a non-convex function (shown in red). (a) Saddle plateau (b,d) Global minima (c,e,g) Local maxima (f,h) Local minima (i - right panel) Saddle point. *Right:* Guaranteed properties of our framework. From any initialization a non-increasing path exists to a global minimum. From points on a flat plateau a simple method exists to find the edge of the plateau (green points).

non-optimal plateaus (a,c) for which there is no local descent direction[1], there is a simple method to find the edge of the plateau from which there will be a descent direction (green points). Taken together, these results will imply a theoretical meta-algorithm that is guaranteed to find a global minimum of the non-convex factorization problem if from any point one can either find a local descent direction or verify the non-existence of a local descent direction. The primary challenge from a theoretical perspective (which is not solved by our results and is potentially NP-hard for certain problems within our framework) is thus how to find a local descent direction (which is guaranteed to exist) from a non-globally-optimal critical point.

Two concepts will be key to establishing our analysis framework: 1) the dimensionality of the factorized elements is not assumed to be fixed, but instead fit to the data through regularization (for example, in matrix factorization the number of columns in U and V is allowed to change) 2) we require the mapping, Φ, and the

[1] Note that points in the interior of these plateaus could be considered both local maxima and local minima as there exists a neighborhood around these points such that the point is both maximal and minimal on that neighborhood.

regularization on the factors, Θ, to be positively homogeneous (defined below).

Definition 14 *A function g is **positively homogeneous with degree p** if $\forall \alpha \geq 0$, $g(\alpha x^1, \ldots, \alpha x^N) = \alpha^p g(x^1, \ldots, x^N)$.*

Interestingly, the deep learning field has increasingly moved to using non-linearities such as Rectified Linear Units (ReLU) and Max-Pooling, both of which satisfy the positive homogeneity property. Additionally, it has been noted empirically that both the speed of training the neural network and the overall performance of the network is increased significantly when ReLU non-linearities are used instead of the more traditional hyperbolic tangent or sigmoid non-linearities [44–46, 48]. We suggest that our framework provides a partial theoretical explanation to this phenomena and also offers guidance on simple concepts to take into consideration in the design of learning systems to facilitate efficient optimization.

4.2 Prior Work

Despite the significant empirical success and wide ranging applications of the models discussed above (and many others not discussed), it is not immediately apparent why one should expect them to succeed. From an optimization perspective, the algorithms often used to solve factorization problems – including (but certainly not limited to) alternating minimization, gradient descent, stochastic gradient descent, block coordinate descent, back-propagation, and quasi-Newton methods – are typically only guaranteed to converge to a critical point or local minimum of the objective function [10, 26, 39, 72, 73], so the non-convexity of the problem leaves the model somewhat ill-posed in the sense that it is not just the model formulation that is important but also implementation details, such as how the model is initialized and particulars of the optimization algorithm, which can have a significant impact on the performance of the model. Yet, although there is little in the way of theoretical guarantees regarding the optimization of these methods, it is often reported empirically that many different solutions achieve equal performance in practice and have very similar objective values.

In the previous chapter, prior work relating to factorized semidefinite programming (SDP) that guarantees global optimality of non-convex optimization problems was discussed, and these results provide some initial support for the idea that solving optimization problems in the factorized domain was possible. Beyond matrix factorization, in the context of neural networks, [74] showed that for neural networks with a single hidden layer, if the number of neurons in the hidden layer is not fixed, but instead fit to the data through a sparsity inducing regularization, then the process of training a globally optimal neural network is analogous to selecting a finite number of hidden units from a potentially infinite dimensional space of all possible hidden units. The selected hidden units are then combined by taking a weighted summation of these units to produce the output. The specific optimization problem is of the form

$$\min_w \ell(Y, \sum_i h_i(V) w_i) + \lambda \|w\|_1, \tag{4.4}$$

where $h_i(V)$ represents one possible hidden unit activation in response to the training data V from an infinite dimensional space $h_i(V) \in \mathcal{H}$ of all possible hidden unit activations. Clearly (4.4) is a convex optimization problem (assuming $\ell(Y, X)$ is convex w.r.t. X) and straightforward to solve for a finite set of $h_i(V)$ activations. However, because $\mathcal{H}$ is an infinite dimensional space the primary difficulty lies in how to select the appropriate hidden unit activations. Nonetheless, by using arguments from gradient boosting, it is possible to show that problem (4.4) can be globally optimized by sequentially adding hidden units to the network until one can no longer find a hidden unit whose addition will decrease the objective function [74–76]. Here, our work takes a conceptually similar approach while extending and refining these ideas. The key innovation is that by considering a well defined family of hidden unit mappings we can analyze the problem directly in the space of the parameters that define the hidden unit mappings (i.e., the network weight parameters of potentially multilayer networks). This allows us to show that if the size of the network is sufficiently large (with an upper bound on the sufficient size of the network that is linearly proportional to the number of training examples) then the non-convex optimization problem with respect to the network weight parameters directly has the simplified landscape outlined in Figure 4.1. Our work also provides sufficient conditions on the network architecture and regularization of the network weight parameters to guarantee that from any initialization a globally optimal solution can be found by performing purely local descent on the network weights.

Finally, we note that several recent works have also explored the error surface of multilayer neural networks using tools derived from random matrix theory and statistical physics. Applying ideas from random matrix theory to high-dimensional non-convex optimization, [77] argue that, under certain assumptions, for high-dimensional optimization problems if one is given a particular critical point, it is vastly more likely that the critical point will be a saddle point rather than a local minimum and thus avoiding saddle points is the key difficulty in high-dimensional, non-convex optimization. Using arguments from statistical physics, [78] show that, under certain assumed distributions of the training data and the network weight parameters, as the number of hidden units in a network increases the distribution of local minima becomes increasingly concentrated in a small band of objective function values near the global optimum (and thus all local minima become increasingly close to being global minima). Our results will largely echo these two general ideas, but we note that we take a markedly different approach. Specifically, we analyze the problem from a purely deterministic approach which does not require any assumptions regarding the distribution of the inputs or the network weight parameters. With this approach, we show that saddle points and plateaus are the *only* critical points that one needs to be concerned with due to the fact that for networks of sufficient size, local minima that require one to climb the objective surface to escape from, such as (f) and (h) in Figure 4.1, are guaranteed not to exist.

4.3 Preliminaries

Before we present our main results, we first describe our notation system and recall a few definitions.

4.3.1 Notation

Our formulation is fairly general in regards to the dimensionality of the data and factorized variables. As a result, to simplify the notation, we will use capital letters as a shorthand for a set of dimensions, and individual dimensions will be denoted with lower case letters. For example, the tensor $X \in \mathbb{R}^{d_1 \times \ldots \times d_N}$ will be denoted as $X \in \mathbb{R}^D$ for $D = d_1 \times \ldots \times d_N$, and the cardinality of $X \in \mathbb{R}^D$ will be denoted as $\operatorname{card}(X) = \prod_{i=1}^N d_i$. Similarly, $X \in \mathbb{R}^{D \times R} \equiv X \in \mathbb{R}^{d_1 \times \ldots \times d_N \times r_1 \times \ldots \times r_M}$ for $D = d_1 \times \ldots \times d_N$ and $R = r_1 \times \ldots \times r_M$. Given two tensors with matching dimensions except for the last dimension, $X \in \mathbb{R}^{D \times r_x}$ and $Z \in \mathbb{R}^{D \times r_z}$, we will use $[X \ Z] \in \mathbb{R}^{D \times (r_x + r_z)}$ to denote the concatenation of the two tensors along the last dimension.

Given an element from a tensor space, we will use a subscript to denote a slice of the tensor along the last dimension. For example, given a matrix $X \in \mathbb{R}^{d_1 \times r}$, then $X_i \in \mathbb{R}^{d_1}, i \in \{1, \ldots, r\}$, denotes the i^{th} column of X. Similarly, given a third order tensor $X \in \mathbb{R}^{d_1 \times d_2 \times r}$ then $X_i \in \mathbb{R}^{d_1 \times d_2}, i \in \{1 \ldots, r\}$, denotes the i^{th} slice along the third dimension. Tensors which have a size of 1 along the last dimension and are not slices from a larger tensor will be denoted with lower-case letters. For example, $x \in \mathbb{R}^{D \times 1}$ denotes a tensor of size 1 along its last dimension, while $X_i \in \mathbb{R}^{D \times 1}$ is a slice from a larger tensor $X \in \mathbb{R}^{D \times r}$.

We will denote the dot product between two elements from a tensor space ($X \in \mathbb{R}^D, Z \in \mathbb{R}^D$) as $\langle X, Z \rangle = \text{vec}(X)^T \text{vec}(Z)$, where $\text{vec}(\cdot)$ denotes flattening the tensor into a vector. For a function $g(x)$, we will denote its image as $\text{Im}(g)$ and its Fenchel dual as $g^*(x) \equiv \sup_z \langle x, z \rangle - g(z)$. The gradient of a differentiable function $g(x)$ will be denoted $\nabla g(x)$, and the subgradient of a convex (but possibly non-differentiable) function $g(x)$ will be denoted $\partial g(x)$. For a multivariate differentiable function $g(x^1, \ldots, x^K)$, we will use $\nabla_{x^i} g(x^1, \ldots, x^K)$ to denote the portion of the gradient corresponding to x^i. The space of non-negative real numbers will be denoted $\mathbb{R}_+$, and the space of positive integers will be denoted $\mathbb{N}_+$.

4.3.2 Definitions

We now make/recall a few general definitions and well known facts which will be used in our analysis.

Definition 15 *A* ***size-r set of K factors*** $(X^1, \ldots, X^K)_r$ *is defined to be a set of* K

tensors where the last dimension of each tensor is equal to r*. That is,* $(X^1, \ldots, X^K)_r \in \mathbb{R}^{(D^1 \times r)} \times \ldots \times \mathbb{R}^{(D^K \times r)}$.

Definition 16 *The **indicator function of a set** C is defined as*

$$\delta_C(x) = \begin{cases} 0 & x \in C \\ \infty & x \notin C \end{cases} . \tag{4.5}$$

Definition 17 *A function* $g : \mathbb{R}^{D^1} \times \ldots \times \mathbb{R}^{D^N} \to \mathbb{R}_+$ *is **positive semidefinite** if* $g(0, \ldots, 0) = 0$ *and* $g(x^1, \ldots, x^N) \geq 0,\ \forall (x^1, \ldots, x^N)$.

Definition 18 *The **one-sided directional derivative** of a function* $g(x)$ *at a point* x *in the direction* z *is denoted* $dg(x)(z)$ *and defined as* $dg(x)(z) \equiv \lim_{\epsilon \searrow 0}\ (g(x+\epsilon z) - g(x))\epsilon^{-1}$.

Also, recall that for a differentiable function $g(x)$, $dg(x)(z) = \langle \nabla g(x), z \rangle$.

4.4 Problem Formulation

Returning to the problem from the introduction (4.2), we now define the family of mapping functions from the factors into the output space and the family of regularization functions on the factors (Φ and Θ, respectively) which we will study in our framework.

4.4.1 Factorization Mappings

In this paper, we consider mappings Φ which are based on a sum of what we refer to as an **elemental mapping**. Specifically, if we are given a size-r set of K factors $(X^1, \dots, X^K)_r$, the elemental mapping $\phi : \mathbb{R}^{D^1} \times \dots \times \mathbb{R}^{D^K} \to \mathbb{R}^D$ takes a slice along the last dimension from each tensor in the set of factors and maps it into the output space. We then define the full mapping to be the sum of these elemental mappings along each of the r slices in the set of factors. The only requirement we impose on the elemental mapping is that it must be positively homogeneous with a positive degree. More formally,

Definition 19 *An **elemental mapping**, $\phi : \mathbb{R}^{D^1} \times \dots \times \mathbb{R}^{D^K} \to \mathbb{R}^D$ is any mapping which is positively homogeneous with degree $p > 0$. The **r-element factorization mapping** $\Phi_r : \mathbb{R}^{(D^1 \times r)} \times \dots \times \mathbb{R}^{(D^K \times r)} \to \mathbb{R}^D$ is defined as*

$$\Phi_r(X^1, \dots, X^K) = \sum_{i=1}^{r} \phi(X_i^1, \dots, X_i^K). \tag{4.6}$$

From the definition of Φ_r it is easy to verify that if ϕ is positively homogeneous with degree p, then Φ_r is also positively homogeneous with degree p and satisfies the following proposition.

Proposition 8 *Given a size-r_x set of K factors, $(X^1, \dots, X^K)_{r_x}$, and a size-r_z set* *of K factors, $(Z^1, \ldots, Z^K)_{r_z}$, then $\forall \alpha \geq 0, \beta \geq 0$ we have*

$$\Phi_{(r_x+r_z)}([\alpha X^1 \, \beta Z^1], \ldots, [\alpha X^K \, \beta Z^K]) = \alpha^p \Phi_{r_x}(X^1, \ldots, X^K) + \beta^p \Phi_{r_z}(Z^1, \ldots, Z^K) \tag{4.7}$$

where recall, $[X \;\; Z]$ denotes the concatenation of X and Z along the final dimension of the tensor.

As we do not place any restrictions on the elemental mapping, ϕ, beyond the requirement that it must be positively homogeneous, there are a wide range of problems that can be captured by a mapping with form (4.6). Several example problems which can be placed in this framework include:

Matrix Factorization: The elemental mapping, $\phi : \mathbb{R}^{d_1} \times \mathbb{R}^{d_2} \to \mathbb{R}^{d_1 \times d_2}$

$$\phi(u, v) = uv^T \tag{4.8}$$

is positively homogeneous with degree 2 and $\Phi_r(U, V) = \sum_{i=1}^r U_i V_i^T = UV^T$ is simply matrix multiplication for matrices with r columns.

Tensor Decomposition - CANDECOMP/PARAFAC (CP): Slightly more generally, the elemental mapping $\phi : \mathbb{R}^{d_1} \times \ldots \times \mathbb{R}^{d_K} \to \mathbb{R}^{d_1 \times \ldots \times d_K}$

$$\phi(x^1, \ldots, x^K) = x^1 \otimes \cdots \otimes x^K, \tag{4.9}$$

where $\otimes$ denotes the tensor outer product, results in $\Phi_r(X^1, \ldots, X^K)$ being the mapping used in the rank-r CANDECOMP/PARAFAC (CP) tensor decomposition model [28],

$$\Phi_r(X^1, \ldots, X^K) = \sum_{i=1}^{r} X_i^1 \otimes \cdots \otimes X_i^K, \tag{4.10}$$

which is visualized for a 3rd order tensor in figure 4.2. Further, instead of choosing ϕ to be a simple outer product, we can also generalize this to be any multilinear function of the factor slices $(X_i^1, \ldots, X_i^K)$. For example, the output could be formed by taking convolutions between the factor slices. We note that more general tensor decompositions, such as the general form of the Tucker decomposition, do not explicitly fit inside the framework we describe here; however, by using similar arguments to the ones we will develop here, it is possible to show analogous results to those we derive in this paper for more general tensor decompositions, and we will briefly discuss these extensions in Section 4.6.2.

Neural Networks with Rectified Linear Units (ReLU): Let $\psi^+(x) \equiv \max\{x, 0\}$ be the linear rectification function, which is applied element-wise to a tensor x of arbitrary dimension. Then if we are given a matrix of training data $V \in \mathbb{R}^{N \times d_1}$, the elemental mapping $\phi(x^1, x^2) : \mathbb{R}^{d_1} \times \mathbb{R}^{d_2} \rightarrow \mathbb{R}^{N \times d_2}$

$$\phi(x^1, x^2) = \psi^+(Vx^1)(x^2)^T \tag{4.11}$$

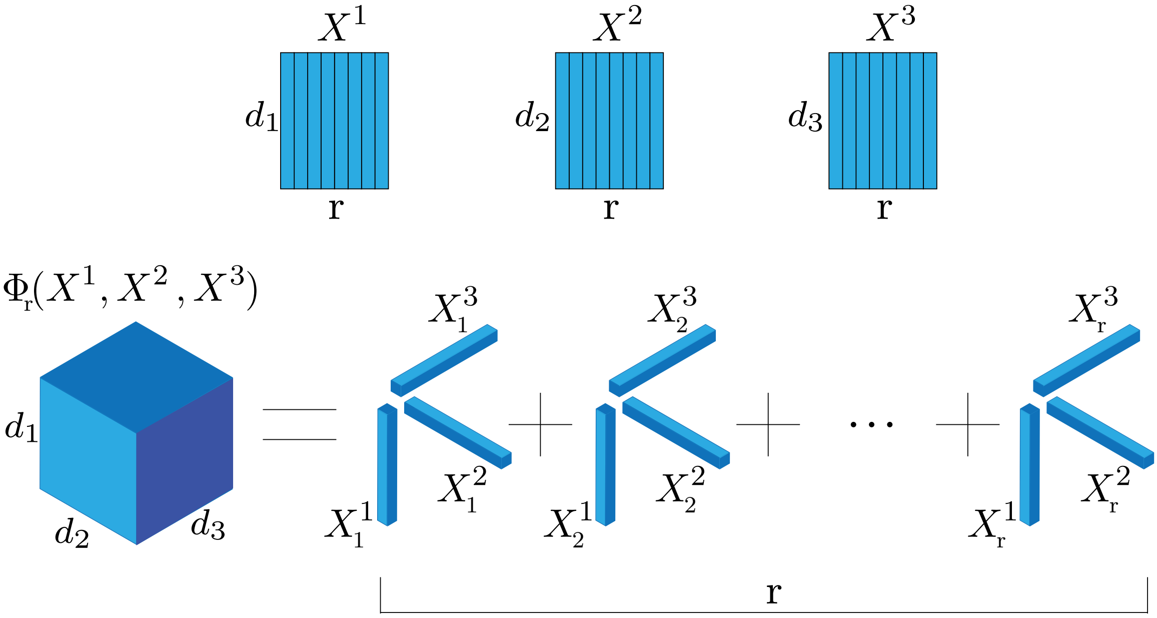

Figure 4.2: Rank-r CP decomposition of a 3rd order tensor.

results in a mapping $\Phi_r(X^1, X^2) = \psi^+(VX^1)(X^2)^T$, which can be interpreted as producing the d_2 outputs of a neural network with r neurons in a single hidden layer in response to the input of N data points of d_1 dimensional data, V. The hidden units have a ReLU non-linearity; the other units are linear; and the $(X^1, X^2) \in \mathbb{R}^{d_1 \times r} \times \mathbb{R}^{d_2 \times r}$ matrices contain the connection weights from the input-to-hidden and hidden-to-output layers, respectively. The left panel of figure 4.3 illustrates such a network with $(r, d_1, d_2) = (4, 3, 2)$.

By utilizing more complicated definitions of ϕ, it is possible to consider a broad range of neural network architectures. As a simple example of networks with multiple hidden layers, an elemental mapping such as $\phi : \mathbb{R}^{d_1 \times d_2} \times \mathbb{R}^{d_2 \times d_3} \times \mathbb{R}^{d_3 \times d_4} \times \mathbb{R}^{d_4 \times d_5} \to$

$\mathbb{R}^{N\times d_5}$

$$\phi(x^1,x^2,x^3,x^4) = \psi^+(\psi^+(\psi^+(Vx^1)x^2)x^3)x^4 \tag{4.12}$$

gives a $\Phi_r(X^1,X^2,X^3,X^4)$ mapping which is the output of a 5 layer neural network in response to the inputs in the $V \in \mathbb{R}^{N\times d_1}$ matrix with ReLU non-linearities on all of the hidden layer units. In this case, the network has the architecture that there are r, 4 layer fully-connected subnetworks, with each subnetwork having the same number of units in each layer as defined by the dimensions $\{d_2,d_3,d_4\}$. The r subnetworks are all then fed into a fully connected linear layer to produce the output. This is visualized in figure 4.3 for $(d_1,d_2,d_3,d_4,d_5) = (5,3,5,1,2)$ and with $r = 4$.

More general still, since *any* positively homogenous transformation is a potential elemental mapping, by an appropriate definition of ϕ, one can describe neural networks with very general architectures, provided the non-linearities in the network are compatible with positive homogeneity (ReLUs are one example, but non-linearities such as the absolute value, raising each element to a non-zero power, max-out, and max-pooling are also positively homogeneous). For example, the well-known "AlexNet" network from [44], which consists of a series of convolutional layers, linear-rectification, max-pooling layers, response normalization layers, and fully connected layers, can be described by taking $r = 1$ and defining ϕ to be the entire transformation of the network (with the removal or slight redefinition of the response normalization layers, which are not strictly positively homogenous, see Section 4.6.3). Note, however, that our results will rely on r potentially changing size or being initialized to

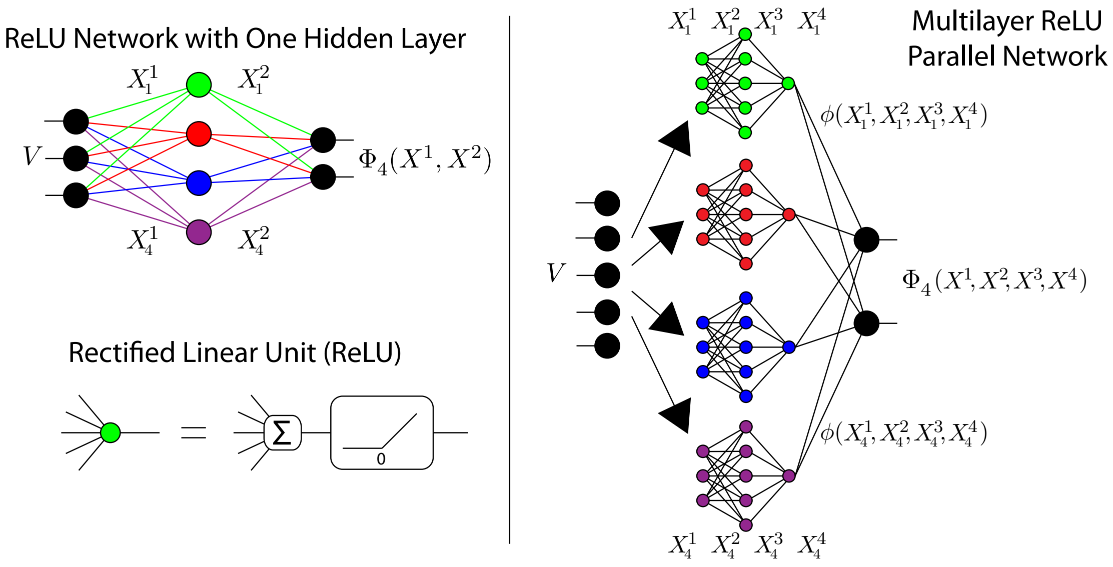
ReLU Network with One Hidden Layer
X_1^1
X_1^2
V
$\Phi_4(X^1, X^2)$
X_4^1
X_4^2
Rectified Linear Unit (ReLU)
=
Σ
0
Multilayer ReLU Parallel Network
X_1^1 X_1^2 X_1^3 X_1^4
$\phi(X_1^1, X_1^2, X_1^3, X_1^4)$
V
$\Phi_4(X^1, X^2, X^3, X^4)$
$\phi(X_4^1, X_4^2, X_4^3, X_4^4)$
X_4^1 X_4^2 X_4^3 X_4^4

Figure 4.3: Example ReLU networks. (Left panel) ReLU network with a single hidden layer with the mapping described by the equation in (4.11) with $(r = 4, d_1 = 3, d_2 = 2)$. Each color corresponds to one element of the elemental mapping $\phi(X_i^1, X_i^2)$. The colored hidden units have rectifying non-linearities, while the black units are linear. (Right panel) Multilayer ReLU network with 4 fully connected parallel subnetworks $(r = 4)$ with elemental mappings defined by (4.12) with $(d_1 = 5, d_2 = 3, d_3 = 5, d_4 = 1, d_5 = 2)$. Each color corresponds to the subnetwork described by one element of the elemental mapping $\phi(X_i^1, X_i^2, X_i^3, X_i^4)$.

be sufficiently large, which limits the applicability of our results to current state-of-the-art network architectures with $r = 1$. Essentially, the main limitation is that the analysis we develop here relies on a network with multiple parallel subnetworks which are linearly combined to produce the output. This has potentially interesting interpretations in relation to techniques such as drop-out [47] as we discuss in Section 4.6.3. Also, in Section 4.6.2 we briefly describe how ideas from our framework can be extended to more general Φ mappings to capture additional potential network architectures.

Here we have provided a few examples of common factorization mappings that can be cast in form (4.6), but certainly there are a wide variety of other problems for which our framework is relevant. Additionally, while most of the mappings described above are positively homogeneous with degree equal to the degree of the factorization (K), this is not a requirement; $p > 0$ is sufficient. For example, non-linearities such as raising each element to a power or convolutional neural network techniques such as contrast normalization will affect the degree of positive homogeneity but can still be included in our framework. What will turn out to be essential, however, is that we require p to match the degree of positive homogeneity used to regularize the factors, which we will discuss in the next section.

4.4.2 Factorization Regularization

Inspired by the ideas from structured convex matrix factorization, rather than trying to analyze the optimization over a size-r set of K factors $(X^1, \ldots, X^K)_r$ for a fixed r, we instead consider the optimization problem where r is possibly allowed to vary and adapted to the data through regularization. To do so, we will define a regularization function similar to the $\| \cdot \|_{u,v}$ norm discussed in matrix factorization, which is convex with respect to the output tensor, $X = \Phi_r(X^1, \ldots, X^K)$, but which still allows for regularization to be placed on the factors, $(X^1, \ldots, X^K)_r$. Similar to our definition in (4.6), we will begin by first defining an **elemental regularization function** $\theta : \mathbb{R}^{D^1} \times \ldots \times \mathbb{R}^{D^K} \rightarrow \mathbb{R}_+ \cup \infty$ which takes as input slices of the factorized tensors along the last dimension and returns a non-negative number. The requirements we place on θ are that it must be positively homogeneous and positive semidefinite. Formally,

Definition 20 *We define an* ***elemental regularization function*** $\theta : \mathbb{R}^{D^1} \times \ldots \times \mathbb{R}^{D^K} \to \mathbb{R}_+ \cup \infty$*, to be any function which is positive semidefinite and positively homogeneous.*

Again, due to the generality of our framework, there are a wide variety of possible elemental regularization functions. We highlight two positive semidefinite, positively homogeneous functions which are commonly used and note that functions can be composed with summations, multiplications, and raising to non-zero powers to change the degree of positive homogeneity and combine various functions.

Norms: Any norm $\|x\|$ is positively homogeneous with degree 1. Note that because we make no requirement of convexity on θ, this framework can also include functions such as the l_q pseudo-norms for $q \in (0, 1)$.

Conic Indicators: The indicator function $\delta_C(x)$ of any conic set C is positively homogeneous for all degrees. Recall that a conic set, C, is simply any set such that if $x \in C$ then $\alpha x \in C$, $\forall \alpha \geq 0$. A few popular conic sets which can be of interest include the non-negative orthant $\mathbb{R}^D_+$, the kernel of a linear operator $\{x : Ax = 0\}$, inequality constraints for a linear operator $\{x : Ax \geq 0\}$, and the set of positive semidefinite matrices. Constraints on the non-zero support of x are also typically conic sets. For example, the set $\{x : \|x\|_0 \leq n\}$ is a conic set, where $\|x\|_0$ is simply the number of non-zero elements in x and n is a positive integer. More abstractly, conic sets can also be used to enforce invariances w.r.t. positively homogeneous transformations. For example, given two positively homogeneous functions $g(x)$, $g'(x)$ with equal degrees of positive homogeneity, the sets $\{x : g(x) = g'(x)\}$ and $\{x : g(x) \geq g'(x)\}$ are also conic sets.

From this, we now define our main regularization function:

Definition 21 *Given an elemental mapping ϕ and an elemental regularization function θ, we define the* ***factorization regularization function****, $\Omega_{\phi,\theta}(X) : \mathbb{R}^D \to \mathbb{R}_+ \cup \infty$ to be*

$$\Omega_{\phi,\theta}(X) \equiv \inf_{r \in \mathbb{N}_+} \inf_{(X^1,\ldots,X^K)_r} \sum_{i=1}^{r} \theta(X_i^1, \ldots, X_i^K) \ \text{ s.t. } \ \Phi_r(X^1, \ldots, X^K) = X \tag{4.13}$$

with the additional condition that $\Omega_{\phi,\theta}(X) = \infty$ if $X \notin \bigcup_r \operatorname{Im}(\Phi_r)$.

For the above $\Omega_{\phi,\theta}$ function to be useful in our analysis, it will be necessary that ϕ and θ have equal degrees of positive homogeneity. The necessity of this requirement will become apparent when we begin our analysis, and we discuss this importance further in section 4.6.1. A few typical formulations of a θ which are positively homogeneous with degree K and nondegenerate with the example ϕ mappings we have given might include:

$$\theta(x^1,\dots,x^K) = \prod_{i=1}^{K} \|x^i\|_{(i)} \tag{4.14}$$

$$\theta(x^1,\dots,x^K) = \tfrac{1}{K}\sum_{i=1}^{K} \|x^i\|_{(i)}^K \tag{4.15}$$

$$\theta(x^1,\dots,x^K) = \prod_{i=1}^{K} (\|x^i\|_{(i)} + \delta_{C_i}(x^i)) \tag{4.16}$$

where all of the norms, $\|\cdot\|_{(i)}$, and conic sets, δ_{C_i}, are arbitrary. Forms (4.14) and (4.15) can be shown to be equivalent, in the sense that they give rise to the same $\Omega_{\phi,\theta}$ function (see the following proposition) for all of the example mappings ϕ we have discussed above. By an appropriate choice of norm one can induce various properties in the factorized elements (such as sparsity) with forms (4.14) and (4.15), while form (4.16) is similar but additionally constrains each factor to be an element of a conic set C_i (see [54–56, 62] for examples from matrix factorization).

Proposition 9 *For any elemental mapping ϕ which is positively homogenous with degree 1 in each factor – that is $\phi(x^1,\dots,\alpha x^k,\dots,x^K) = \alpha\phi(x^1,\dots,x^k,\dots,x^K)$ $\forall\alpha \geq 0$ and $\forall k \in \{1,\dots,K\}$, then the elemental regularization functions* (4.14) *and* (4.15) *produce the same regularization function $\Omega_{\phi,\theta}$.*

Proof. For the case of the $\|\cdot\|_{u,v}$ norm this result is known and is what allows for the two equivalent forms of definition in (3.7) [54]. The equivalence between (4.14) and (4.15) is easily extended to cases with $K > 2$ by starting from (4.15) and considering the following geometric programming optimization problem:

$$\min_{\alpha_1,\ldots,\alpha_K} \tfrac{1}{K}\sum_{i=1}^{K} \|\alpha_i x^i\|_{(i)}^K \text{ subject to } \prod_{i=1}^{K}\alpha_i = 1 \text{ and } \alpha_i > 0 \ \forall i. \tag{4.17}$$

Making the change of variables $z_i = \ln(\alpha_i)$ we have the equivalent problem

$$\min_{z_1,\ldots,z_K} \tfrac{1}{K}\sum_{i=1}^{K} e^{z_i K}\|x^i\|_{(i)}^K \text{ subject to } \sum_{i=1}^{K} z_i = 0, \tag{4.18}$$

which gives the KKT conditions

$$e^{z_{i_{opt}}K}\|x^i\|_{(i)}^K = \gamma_{opt} \ \ \forall i \in \{1,\ldots,K\}, \tag{4.19}$$

where γ is a Lagrange multiplier to enforce the equality constraint. Taking the product of (4.19) over i and raising the result to $1/K$, we have

$$\gamma_{opt} = e^{(z_{1_{opt}}+\ldots+z_{K_{opt}})}\prod_{i=1}^{K}\|x^i\|_{(i)} = \prod_{i=1}^{K}\|x^i\|_{(i)} = (4.14), \tag{4.20}$$

where the second equality is due to the constraint that the z terms sum to 0. Substituting (4.19), (4.20), and $\alpha_{i_{opt}} = e^{z_{i_{opt}}}$ into (4.17) then gives that (4.17) = (4.14). Due to the requirement that $\prod_{i=1}^{K}\alpha_i = 1$ we have $\phi(\alpha_1 x^1,\ldots,\alpha_K x^K) = \phi(x^1,\ldots,x^K)$ from ϕ being positively homogeneous with degree 1 in each factor. As a result, the above discussion has shown that for any θ of form (4.15), for any factorization

$\Phi(X^1, \ldots, X^K) = X$ the infimum will always be achieved when (4.15) = (4.14), since if the two were not equal we could scale the factors by the *alpha* constants which are the solution to (4.17) and decrease the value of the factorization, which completes the result. ■

Note that the above proposition is also easily generalized to other positive semidefinite functions that are positively homogeneous with degree 1 (other than norms) by using identical arguments.

4.4.2.1 Nondegenerate Factorization Regularization

While the above forms of elemental regularizers θ result in useful regularizers in the product space, $\Omega_{\phi,\theta}$, note that in general simply ensuring that the degrees of positive homogeneity are matched between ϕ and θ is not necessarily sufficient to guarantee a useful factorization regularization function $\Omega_{\phi,\theta}$. For example, in matrix factorization with $\phi(u, v) = uv^T$, taking $\theta(u, v) = \delta_C(u)\|v\|^2$ for any arbitrary norm and conic set C, we have matched degrees of positive homogeneity between ϕ and θ (i.e., 2); however, we can always reduce the value of $\theta(u, v)$ by scaling v by a constant $\alpha \in (0, 1)$ and scaling u by α^{-1} without changing the value of $\phi(u, v)$. As such, this implies that $\Omega_{\phi,\theta}(X) = 0 \; \forall X$ and the infimum in (4.13) can never be achieved. As a result, to make the $\Omega_{\phi,\theta}$ function well defined for our analysis purposes, we will require that the (ϕ, θ) pair satisfy a nondegeneracy property, defined below:

Definition 22 *Given an elemental mapping ϕ and an elemental regularization func-*

tion θ, will we say that (ϕ, θ) is a ***nondegenerate pair*** *if*

1. *θ and ϕ are both positively homogeneous with degree p, for some $p > 0$*

2. *$\forall X \in \mathrm{Im}(\phi)\backslash 0$, we have*

$$\min_{(z^1,\ldots,z^K):\phi(z^1,\ldots,z^K)=X} \theta(z^1,\ldots,z^K) = \inf_{(z^1,\ldots,z^K):\phi(z^1,\ldots,z^K)=X} \theta(z^1,\ldots,z^K) > 0. \tag{4.21}$$

(Note that in property 2 this is ϕ, not Φ).

We will assume for the remainder of this work that ϕ and θ satisfy this condition.

Assumption 1 *(ϕ, θ) is a nondegenerate pair as defined by Definition 22.*

4.4.3 Properties of the Factorization Regularization Function

We now show a few properties regarding $\Omega_{\phi,\theta}$, with the key points being that it is a convex function of X and in general the infimum in (4.13) can always be achieved with a finitely sized factorization (i.e., r does not need to approach ∞)[2]. In particular, $\Omega_{\phi,\theta}$ satisfies the following proposition:

[2] In particular, the largest r needs to be is $\mathrm{card}(X)$, and we note that $\mathrm{card}(X)$ is a worst case upper bound on the size of the factorization. In certain cases the bound can be shown to be lower. As an example, $\Omega_{\phi,\theta}(X) = \|X\|_*$ when $\phi(u,v) = uv^T$ and $\theta(u,v) = \|u\|_2\|v\|_2$. In this case the infimum can be achieved with $r \leq \mathrm{rank}(X) \leq \min\{\mathrm{card}(u), \mathrm{card}(v)\}$.

Proposition 10 *The factorization regularization function* $\Omega_{\phi,\theta} : \mathbb{R}^D \to \mathbb{R} \cup \infty$ *as defined in* (4.13)*, such that* (ϕ, θ) *is a nondegenerate pair, has the following properties:*

1. $\Omega_{\phi,\theta}(0) = 0$ *and* $\Omega_{\phi,\theta}(X) > 0 \ \ \forall X \neq 0$.
2. $\Omega_{\phi,\theta}$ *is positively homogeneous with degree 1, i.e.,* $\Omega_{\phi,\theta}(\alpha X) = \alpha \Omega_{\phi,\theta}(X) \ \forall \alpha \geq 0$.
3. $\Omega_{\phi,\theta}(X + Z) \leq \Omega_{\phi,\theta}(X) + \Omega_{\phi,\theta}(Z) \ \ \forall (X, Z)$.
4. $\Omega_{\phi,\theta}(X)$ *is convex w.r.t.* $X \in \mathbb{R}^D$.
5. $\forall X$ *s.t.* $\Omega_{\phi,\theta}(X) < \infty$*, the infimum in* (4.13) *can be achieved with* $r \leq \operatorname{card}(X)$.
6. *If for some* $k \in \{1, \dots, K\}$ *we have that* $\phi(x^1, \dots, -x^k, \dots, x^K) = -\phi(x^1, \dots, x^k, \dots, x^K)$ *and* $\theta(x^1, \dots, -x^k, \dots, x^K) = \theta(x^1, \dots, x^k, \dots, x^K)$*, then* $\Omega_{\phi,\theta}(X)$ *is also a norm on* X.

Before proving the above result, we will first characterize the Fenchel dual of $\Omega_{\phi,\theta}$, which will be needed for many points of our analysis.

Proposition 11 *The Fenchel dual of* $\Omega_{\phi,\theta}(X)$ *is given by*

$$\Omega^*_{\phi,\theta}(W) = \begin{cases} 0 & \Omega^\circ_{\phi,\theta}(W) \leq 1 \\ \infty & \text{otherwise} \end{cases} \tag{4.22}$$

where

$$\Omega^\circ_{\phi,\theta}(W) \equiv \sup_{(z^1,\dots,z^K)} \left\langle W, \phi(z^1, \dots, z^K) \right\rangle \quad \text{s.t.} \quad \theta(z^1, \dots, z^K) \leq 1. \tag{4.23}$$

Proof. Recall, $\Omega^*_{\phi,\theta}(W) \equiv \sup_Z \langle W, Z\rangle - \Omega_{\phi,\theta}(Z)$, so for Z to approach the supremum we must have $Z \in \bigcup_r \operatorname{Im}(\Phi_r)$. As result, the problem is equivalent to

$$\Omega^*_{\phi,\theta}(W) = \sup_{r\in\mathbb{N}_+} \sup_{(Z^1,\dots,Z^K)_r} \left\langle W, \Phi_r(Z^1,\dots,Z^K)\right\rangle - \sum_{i=1}^r \theta(Z^1_i,\dots,Z^K_i) \tag{4.24}$$

$$= \sup_{r\in\mathbb{N}_+} \sup_{(Z^1,\dots,Z^K)_r} \sum_{i=1}^r \left[\left\langle W, \phi(Z^1_i,\dots,Z^K_i)\right\rangle - \theta(Z^1_i,\dots,Z^K_i)\right]. \tag{4.25}$$

If $\Omega^\circ_{\phi,\theta}(W) \leq 1$ then all the terms in the summation of (4.25) will be non-positive, so taking $(Z^1,\dots,Z^K) = (0,\dots,0)$ will achieve the supremum. This can be seen by noting that because of the balanced degrees of homogeneity, if $\Omega^\circ_{\phi,\theta}(W) \leq 1$ then we will always have $\left\langle W, \phi(z^1,\dots,x^K)\right\rangle \leq \theta(z^1,\dots,z^K)$ since we can always rescale the $(z^1,\dots,z^K)$ terms by a positive constant α so that $\theta(\alpha z^1,\dots,\alpha z^K) = 1$. To make this point explicit, consider any $(z^1,\dots,z^K)$ and $\alpha > 0$ such that $\theta(\alpha z^1,\dots,\alpha z^K) = 1$, giving

$$\begin{aligned}&\alpha^p \left\langle W, \phi(z^1,\dots,z^K)\right\rangle = \left\langle W, \phi(\alpha z^1,\dots,\alpha z^K)\right\rangle \leq 1 = \\ &\theta(\alpha z^1,\dots,\alpha z^K) = \alpha^p \theta(z^1,\dots,z^K).\end{aligned} \tag{4.26}$$

The inequality above comes from the fact that $\Omega^\circ_{\phi,\theta}(W) \leq 1$, and since $\alpha^p > 0$ we can cancel it from both sides of the inequality to give $\left\langle W, \phi(z^1,\dots,x^K)\right\rangle \leq \theta(z^1,\dots,z^K)$.

Conversely, if $\Omega^\circ_{\phi,\theta}(W) > 1$, then $\exists(z^1,\dots,z^K)$ such that $\left\langle W, \phi(z^1,\dots,z^K)\right\rangle > \theta(z^1,\dots,z^K)$. This result, combined with the positive homogeneity of ϕ and θ gives that (4.25) is unbounded by considering $(\alpha z^1,\dots,\alpha z^K)$ as $\alpha \to \infty$. ∎

Having this characterization of the Fenchel dual, we are now prepared to prove Proposition 10.

Proof. (Proposition 10) Many of these properties can be shown in a similar fashion to results from the $\|\cdot\|_{u,v}$ norm discussed previously [54, 57, 59]. For brevity of notation, we will notate the optimization problem in (4.13) as

$$\Omega_{\phi,\theta} \equiv \inf_{\Phi_r(X^1,\ldots,X^K)=X} \sum_{i=1}^{r} \theta(X_i^1,\ldots,X_i^K), \tag{4.27}$$

where recall that r is variable although it is not explicitly notated.

1. By definition and the fact that θ is positive semidefinite, we always have $\Omega_{\phi,\theta}(X) \geq 0 \quad \forall X$. Trivially, $\Omega_{\phi,\theta}(0) = 0$ since we can always take $(X^1,\ldots,X^K) = (0,\ldots,0)$ to achieve the infimum. For $X \neq 0$, because (ϕ,θ) is a non-degenerate pair then $\sum_{i=1}^{r}\theta(X_i^1,\ldots,X_i^K) > 0$ for any $(X^1,\ldots,X^K)_r$ s.t. $\Phi_r(X^1,\ldots,X^K) = X$ and r finite. Property 5 shows that the infimum can be achieved with r finite, completing the result.

2. The result is easily seen from the positive homogeneity of ϕ and θ,

$$\begin{aligned}\Omega_{\phi,\theta}(\alpha X) = \inf_{\Phi_r(X^1,\ldots,X^K)=\alpha X} \sum_{i=1}^{r} \theta(X_i^1,\ldots,X_i^K) = \\ \inf_{\Phi_r(\alpha^{-1/p}X^1,\ldots,\alpha^{-1/p}X^K)=X} \sum_{i=1}^{r} \theta(X_i^1,\ldots,X_i^K) = \\ \inf_{\Phi_r(Z^1,\ldots,Z^K)=X} \alpha \sum_{i=1}^{r} \theta(Z_i^1,\ldots,Z_i^K) = \alpha\Omega_{\phi,\theta}(X),\end{aligned} \tag{4.28}$$

where the equality between the middle and final lines is simply due to the change of variables $(Z^1, \ldots, Z^K) = (\alpha^{-1/p} X^1, \ldots, \alpha^{-1/p} X^K)$.

3. If either $\Omega_{\phi,\theta}(X) = \infty$ or $\Omega_{\phi,\theta}(Z) = \infty$ then the inequality is trivially satisfied. Considering any (X, Z) pair such that $\Omega_{\phi,\theta}$ is finite for both X and Z, for any $\epsilon > 0$ let $(X^1, \ldots, X^K)_{r_x}$ be an ϵ optimal factorization of X. Specifically, $\Phi_{r_x}(X^1, \ldots, X^K) = X$ and $\sum_{i=1}^{r_x} \theta(X_i^1, \ldots, X_i^K) \leq \Omega_{\phi,\theta}(X) + \epsilon$. Similarly, let $(Z^1, \ldots, Z^K)_{r_z}$ be an ϵ optimal factorization of Z. From Proposition 8 we have $\Phi_{r_x+r_z}([X^1\, Z^1], \ldots, [X^K\, Z^K]) = X+Z$, so $\Omega_{\phi,\theta}(X+Z) \leq \sum_{i=1}^{r_x} \theta(X_i^1, \ldots, X_i^K) + \sum_{j=1}^{r_z} \theta(Z_j^1, \ldots, Z_j^K) \leq \Omega_{\phi,\theta}(X) + \Omega_{\phi,\theta}(Y) + 2\epsilon$. Letting ϵ tend to 0 completes the result.

4. Convexity is given by the combination of properties 2 and 3. Further, note that properties 2 and 3 also show that $\{X \in \mathbb{R}^D : \Omega_{\phi,\theta}(X) < \infty\}$ is a convex set.

5. Let $\Gamma \subset \mathbb{R}^D$ be defined as

$$\Gamma = \{X : \exists (x^1, \ldots, x^K),\ \phi(x^1, \ldots, x^K) = X,\ \theta(x^1, \ldots, x^K) \leq 1\}. \tag{4.29}$$

Note that because (ϕ, θ) is a nondegenerate pair, for any non-zero $X \in \Gamma$ there exists $\alpha \in [1, \infty)$ such that αX is on the boundary of Γ, so Γ and its convex hull are compact sets.

Further, note that Γ contains the origin by definition of ϕ and θ, so as a result, we can define σ_Γ to be a gauge function on the convex hull of Γ,

$$\sigma_\Gamma(X) = \inf_\mu \{\mu : \mu \geq 0,\ X \in \mu \operatorname{conv}(\Gamma)\}. \tag{4.30}$$

Since the infimum w.r.t. μ is linear and constrained to a compact set, it must be achieved. Therefore, there must exist $\mu_{opt} \geq 0$, $\{\beta \in \mathbb{R}^{\operatorname{card}(X)} : \beta_i \geq 0\ \forall i,\ \sum_{i=1}^{\operatorname{card}(X)} \beta_i = 1\}$, and $\{(Z_i^1, \ldots, Z_i^K) : \phi(Z_i^1, \ldots, Z_i^K) \in \Gamma\}_{i=1}^{\operatorname{card}(X)}$ such that $X = \mu_{opt} \sum_{i=1}^{\operatorname{card}(X)} \beta_i \phi(Z_i^1, \ldots, Z_i^K)$ and $\sigma_\Gamma(X) = \mu_{opt}$.

Combined with positive homogeneity, this gives that σ_Γ can be defined identically to $\Omega_{\phi,\theta}$, but with the additional constraint $r \leq \operatorname{card}(X)$,

$$\sigma_\Gamma(X) \equiv \inf_{r \in [1, \operatorname{card}(X)]} \inf_{(X^1, \ldots, X^K)_r} \sum_{i=1}^r \theta(X^1, \ldots, X^K) \text{ s.t. } \Phi_r(X^1, \ldots, X^K) = X. \tag{4.31}$$

This is seen by noting that we can take $(X_i^1, \ldots, X_i^K) = ((\mu_{opt}\beta_i)^{1/p} Z_i^1, \ldots, (\mu_{opt}\beta_i)^{1/p} Z_i^K)$ to give

$$\begin{aligned} \mu_{opt} = \sigma_\Gamma(X) \leq \sum_{i=1}^{\operatorname{card}(X)} \theta(X_i^1, \ldots, X_i^K) = \mu_{opt} \sum_{i=1}^{\operatorname{card}(X)} \beta_i \theta(Z_i^1, \ldots, Z_i^K) \\ \leq \mu_{opt} \sum_{i=1}^{\operatorname{card}(X)} \beta_i = \mu_{opt}, \end{aligned} \tag{4.32}$$

and shows that a factorization of size $r \leq \operatorname{card}(X)$ which achieves the infimum $\mu_{opt} = \sigma_\Gamma(X)$ must exist. Clearly from (4.31) σ_Γ is very similar to $\Omega_{\phi,\gamma}$. To show that the two functions are, in fact, the same function, recall that the proof of the Fenchel dual of $\Omega_{\phi,\theta}$ given in Proposition 11 does not depend on the size of r but only on the existence (or non-existence) of a single $(z^1, \dots, z^K)$ element. As a result, using an identical series of arguments to derive the Fenchel dual of σ_Γ, one finds that $\sigma_\Gamma^* = \Omega_{\phi,\theta}^*$, and since both σ_Γ and $\Omega_{\phi,\theta}$ are convex function, the one-to-one correspondence between convex functions and their Fenchel duals gives that $\sigma_\Gamma(X) = \Omega_{\phi,\theta}(X)$, completing the result.

6. Note that from properties 1-3 we have established all of the requirements for a norm, except for invariance w.r.t. negative scaling, i.e., we must show that $\Omega_{\phi,\theta}(-X) = \Omega_{\phi,\theta}(X)$. This is easily seen from the definition of $\Omega_{\phi,\theta}$ and the conditions of the proposition,

$$\begin{aligned}\Omega_{\phi,\theta}(-X) = \inf_{\Phi(X^1,\dots,X^K)=-X} \sum_{i=1}^r \theta(X_i^1, \dots, X_i^K) = \\ \inf_{\Phi(X^1,\dots,-X^k,\dots,X^K)=X} \sum_{i=1}^r \theta(X_i^1, \dots, X_i^K) = \\ \inf_{\Phi(X^1,\dots,Z,\dots,X^K)=X} \sum_{i=1}^r \theta(X_i^1, \dots, Z_i, \dots, X_i^K) = \Omega_{\phi,\theta}(X).\end{aligned} \tag{4.33}$$

■

While $\Omega_{\phi,\theta}$ suffers from many of the practical issues associated with the matrix norm $\|\cdot\|_{u,v}$ discussed earlier (namely that in general it cannot be evaluated in polynomial time due to the complicated definition), because $\Omega_{\phi,\theta}(X)$ is a convex function on X, it allows us to use $\Omega_{\phi,\theta}$ as an analysis tool to derive results for a more tractable factorized formulation. In particular, from the Fenchel dual, one can characterize the subgradient of $\Omega_{\phi,\theta}(X)$ through the following result.

Proposition 12 *The subgradient of $\Omega_{\phi,\theta}(X)$ is given by*

$$\partial\Omega_{\phi,\theta}(X) = \{W : \langle X, W\rangle = \Omega_{\phi,\theta}(X),\ \Omega^{\circ}_{\phi,\theta}(W) \leq 1\}. \tag{4.34}$$

Proof. This is simply due to the fact that because $\Omega_{\phi,\theta}(X)$ is convex we have $W \in \partial\Omega_{\phi,\theta}(X) \iff \langle W, X\rangle = \Omega_{\phi,\theta}(X) + \Omega^{*}_{\phi,\theta}(W)$, and since $\Omega^{*}_{\phi,\theta}$ is just the indicator function on the set $\{W : \Omega^{\circ}_{\phi,\theta}(W) \leq 1\}$ we have the stated result. ■

From this simple result, we now have the basis for the following lemma which will be used in our main results

Lemma 1 *Given a factorization $X = \Phi_r(X^1, \ldots, X^K)$ and a regularization function $\Omega_{\phi,\theta}(X)$, then the following conditions are equivalent:*

1. *$(X^1, \ldots, X^K)$ is an optimal factorization of X; i.e., $\sum_{i=1}^{r} \theta(X^1_i, \ldots, X^K_i) = \Omega_{\phi,\theta}(X)$.*

2. *$\exists W$ such that $\Omega^{\circ}_{\phi,\theta}(W) \leq 1$ and $\big\langle W, \Phi_r(X^1, \ldots, X^K)\big\rangle = \sum_{i=1}^{r} \theta(X^1_i, \ldots, X^K_i)$.*

3. *$\exists W$ such that $\Omega^{\circ}_{\phi,\theta}(W) \leq 1$ and $\forall i \in \{1, \ldots, r\}$, $\big\langle W, \phi(X^1_i, \ldots, X^K_i)\big\rangle = \theta(X^1_i, \ldots, X^K_i)$.*

Further, any W which satisfies condition 2 or 3 satisfies both conditions 2 and 3 and $W \in \partial\Omega_{\phi,\theta}(X)$.

Proof. 2 $\iff$ 3) 3 trivially implies 2 from the definition of Φ_r. For the opposite direction, recall from the proof of Proposition 11 that because $\Omega^\circ_{\phi,\theta}(W) \leq 1$ we have $\big\langle W, \phi(X^1_i, \ldots, X^K_i)\big\rangle \leq \theta(X^1_i, \ldots, X^K_i)$ $\forall i$. Taking the sum over i, we can only achieve equality in 2 if we have equality $\forall i$ in condition 3. This also shows that any W which satisfies condition 2 or 3 must also satisfy the other condition.

We next show that if W satisfies conditions 2/3 then $W \in \partial\Omega_{\phi,\theta}(X)$. First, from condition 2/3 and the definition of $\Omega_{\phi,\theta}$, we have $\Omega_{\phi,\theta}(X) \leq \sum_{i=1}^r \theta(X^1_i, \ldots, X^K_i) = \langle W, X\rangle < \infty$. Thus, recall that because $\Omega_{\phi,\theta}(X)$ is convex and finite at X, we have $\langle W, X\rangle \leq \Omega_{\phi,\theta}(X) + \Omega^*_{\phi,\theta}(W)$ with equality iff $W \in \partial\Omega_{\phi,\theta}(X)$. Now, by contradiction assume W satisfies conditions 2/3 but $W \notin \partial\Omega_{\phi,\theta}(X)$. From condition 2/3 we have $\Omega^*_{\phi,\theta}(W) = 0$, so $\Omega_{\phi,\theta}(X) = \Omega_{\phi,\theta}(X) + \Omega^*_{\phi,\theta}(W) > \langle X, W\rangle = \sum_{i=1}^r \theta(X^1_i, \ldots, X^K_i)$ which contradicts the definition of $\Omega_{\phi,\theta}(X)$.

1 $\implies$ 2) Any $W \in \partial\Omega_{\phi,\theta}(X)$ satisfies $\langle X, W\rangle = \Omega_{\phi,\theta}(X) + \Omega^*_{\phi,\theta}(W) = \sum_{i=1}^r \theta(X^1_i, \ldots, X^K_i)$.

2 $\implies$ 1) By contradiction, assume $(X^1, \ldots, X^K)_r$ was not an optimal factorization of X. This gives, $\Omega_{\phi,\theta}(X) < \sum_{i=1}^r \theta(X^1_i, \ldots, X^K_i) = \langle W, X\rangle = \Omega_{\phi,\theta}(X) + \Omega^*_{\phi,\theta}(W) = \Omega_{\phi,\theta}(X)$, producing the contradiction. ■

Before presenting our main results, we briefly note that the optimization problem associated with (4.23) is typically referred to as the polar problem and is a generalization of the concept of a dual norm. In practice solving the polar can be very challenging (NP-hard in general) and is often the limiting factor in being able to escape non-optimal saddle points and applying our results in practice (see [55, 79] for further information on solving matrix factorization polar problems). In the following sections we will build our analysis of the main problem and discuss these points in further detail.

4.5 Main Results

In the previous section we introduced and established several properties of the $\Omega_{\phi,\theta}$ factorization regularization function. In this section we will utilize the $\Omega_{\phi,\theta}$ function to analyze a wide variety of non-convex factorization problems. To build our analysis, we will start by defining the convex (but typically non-tractable) problem, given by

$$\min_{X,Q} F(X,Q) = \ell(Y,X,Q) + \lambda\Omega_{\phi,\theta}(X) + H(Q). \tag{4.35}$$

Here $X \in \mathbb{R}^D$ is the output of the factorization mapping $X = \Phi(X^1,\ldots,X^K)$ as we have been discussing. For our analysis we will assume the following:

Assumption 2 *$\ell(Y,X,Q)$ is once differentiable and jointly convex in (X,Q).*

Assumption 3 *$H(Q)$ is convex (but possibly non-differentiable).*

Assumption 4 *A minimum of $F(X,Q)$ exists, i.e., $\emptyset \neq \arg\min_{X,Q} F(X,Q)$.*

As we have noted on multiple occasions, it is typically impractical to optimize over functions involving $\Omega_{\phi,\theta}(X)$, and, even if one were given an optimal solution to (4.35),

X_{opt}, one would still need to solve the problem given in (4.13) to recover the desired $(X^1, \ldots, X^K)$ factors. Therefore, we instead focus on the non-convex optimization problem given by

$$\begin{aligned}\min_{(X^1,\ldots,X^K)_r,Q} & f_r(X^1, \ldots, X^K, Q) \equiv \\ & \ell(Y, \Phi_r(X^1, \ldots, X^K), Q) + \lambda \sum_{i=1}^{r} \theta(X_i^1, \ldots, X_i^K) + H(Q).\end{aligned} \tag{4.36}$$

We will show that any local minima of (4.36) is a global minima if it satisfies the condition that one slice from each of the factorized tensors is all zero. Further, we will also show that if r is taken to be large enough then from any initialization we can always find a global minimum of (4.36) by doing an optimization based purely on local descent.

4.5.1 Local Minima Achieve Global Minima

To show our results, we will rely on the fact that the convex function, $F(X, Q)$, is a global lower bound of $f_r(X^1, \ldots, X^K, Q)$ for all factorizations $X = \Phi_r(X^1, \ldots, X^K)$ due to the definition of $\Omega_{\phi,\theta}$. As a result, one can use standard first-order optimality conditions to characterize the globally optimal solutions of $\min_{X,Q} F(X, Q)$. Then, we show that local minima of $f_r(X^1, \ldots, X^K, Q)$ which satisfy the condition that one slice of the factors $(X^1, \ldots, X^K)$ is all zero will also satisfy the optimality conditions for $F(X, Q)$, which implies a global minima of both problems due to the global lower bound.

Before showing our main results, we develop one additional lemma.

Lemma 2 *If* $(X^1,\dots,X^K,Q)$ *is a local minimum of* $f_r(X^1,\dots,X^K,Q)$ *as given in* (4.36)*, then for any* $\beta \in \mathbb{R}^r$

$$\left\langle -\tfrac{1}{\lambda}\nabla_X \ell(Y,\Phi_r(X^1,\dots,X^K),Q), \sum_{i=1}^r \beta_i \phi(X_i^1,\dots,X_i^K) \right\rangle = \sum_{i=1}^r \beta_i \theta(X_i^1,\dots,X_i^K). \tag{4.37}$$

Proof. Let $(Z_i^1,\dots,Z_i^K) = (\beta_i X_i^1,\dots,\beta_i X_i^K)$ for all $i \in \{1\dots r\}$ and let $\Lambda = \sum_{i=1}^r \beta_i \phi(X_i^1,\dots,X_i^K)$. From positive homogeneity and the fact that we have a local minimum, then $\exists \delta > 0$ such that $\forall \epsilon \in (0,\delta)$ we must have

$$f_r(X^1,\dots,X^K,Q) \le f_r(X^1+\epsilon Z^1,\dots,X^K+\epsilon Z^K,Q) \implies \tag{4.38}$$

$$\begin{aligned}&\ell(Y,\Phi_r(X^1,\dots,X^K),Q) + \lambda\sum_{i=1}^r \theta(X_i^1,\dots,X_i^K) + H(Q) \le \\ &\ell\left(Y,\sum_{i=1}^r (1+\epsilon\beta_i)^p \phi(X_i^1,\dots,X_i^K),Q\right) + \lambda\sum_{i=1}^r (1+\epsilon\beta_i)^p \theta(X_i^1,\dots,X_i^K) + H(Q).\end{aligned} \tag{4.39}$$

Taking the first order approximation $(1+\epsilon\beta_i)^p = 1 + p\epsilon\beta_i + O(\epsilon^2)$ and rearranging the terms of (4.39), we arrive at

$$\begin{aligned} 0 \leq & \ell\left(Y, \Phi_r(X^1, \ldots, X^K) + p\epsilon\Lambda + O(\epsilon^2), Q\right) - \ell(Y, \Phi_r(X^1, \ldots, X^K), Q) \\ & + p\epsilon\lambda \sum_{i=1}^{r} \beta_i \theta(X_i^1, \ldots, X_i^K) + O(\epsilon^2), \end{aligned} \tag{4.40}$$

After dividing by ϵ and taking $\lim_{\epsilon \searrow 0}[\frac{(4.40)}{\epsilon}]$, we note that the difference in the $\ell(\cdot, \cdot, \cdot)$ terms gives the one-sided directional derivative $d\ell(Y, \Phi_r(X^1, \ldots, X^K), Q)(p\Lambda, 0)$, thus from the differentiability of ℓ we get

$$0 \leq \left\langle \nabla_X \ell(Y, \Phi_r(X^1, \ldots, X^K), Q), p\Lambda \right\rangle + p\lambda \sum_{i=1}^{r} \beta_i \theta(X_i^1, \ldots, X_i^K). \tag{4.41}$$

Noting that for $\epsilon > 0$ but sufficiently small, we also must have $f_r(X^1, \ldots, X^K, Q) \leq f_r(X^1 - \epsilon Z^1, \ldots, X^K - \epsilon Z^K)$, using identical steps as before and taking the first order approximation $(1 - \epsilon\beta_i)^p = 1 - p\epsilon\beta_i + O(\epsilon^2)$, we get

$$\begin{aligned} 0 \leq & \ell(Y, \Phi_r(X^1, \ldots, X^K) - p\epsilon\Lambda + O(\epsilon^2), Q) - \ell(Y, \Phi_r(X^1, \ldots, X^K), Q) \\ & - p\epsilon\lambda \sum_{i=1}^{r} \beta_i \theta(X_i^1, \ldots, X_i^K) + O(\epsilon^2). \end{aligned} \tag{4.42}$$

Dividing by ϵ and taking the limit $\lim_{\epsilon \searrow 0}[\frac{(4.42)}{\epsilon}]$, we arrive at

$$0 \leq \left\langle \nabla_X \ell(Y, \Phi_r(X^1, \ldots, X^K), Q), -p\Lambda \right\rangle - p\lambda \sum_{i=1}^{r} \beta_i \theta(X_i^1, \ldots, X_i^K) \tag{4.43}$$

Combining (4.41) and (4.43) and rearranging terms gives the result. ■

Based on the above preliminary results, we are now ready to state our main results and several immediate corollaries.

Theorem 8 *Given a function $f_r(X^1,\dots,X^K,Q)$ of the form given in* (4.36)*, any local minimizer of the optimization problem*

$$\begin{aligned}\min_{(X^1,\dots,X^K)_r,Q} \; & f_r(X^1,\dots,X^K,Q) \equiv \\ & \ell(Y,\Phi_r(X^1,\dots,X^K),Q) + \lambda\sum_{i=1}^r \theta(X_i^1,\dots,X_i^K) + H(Q)\end{aligned} \tag{4.44}$$

such that $(X_{i_0}^1,\dots,X_{i_0}^K) = (0,\dots,0)$ for some $i_0 \in \{1,\dots,r\}$ is a global minimizer.

Proof. We begin by noting that from the definition of $\Omega_{\phi,\theta}(X)$, for any factorization $X = \Phi_r(X^1,\dots,X^K)$

$$\begin{aligned}&F(X,Q) = \ell(Y,X,Q) + \lambda\Omega_{\phi,\theta}(X) + H(Q) \le \\ &\ell(Y,\Phi_r(X^1,\dots,X^K),Q) + \lambda\sum_{i=1}^r \theta(X_i^1,\dots,X_i^K) + H(Q) = f_r(X^1,\dots,X^K,Q)\end{aligned} \tag{4.45}$$

with equality at any factorization which achieves the infimum in (4.13). We will show that a local minimum of $f_r(X^1,\dots,X^K,Q)$ satisfying the conditions of the theorem also satisfies the conditions for $(\Phi_r(X^1,\dots,X^K),Q)$ to be a global minimum of the convex function $F(X,Q)$, which implies a global minimum of $f_r(Y,X^1,\dots,X^K,Q)$ due to the global bound in (4.45).

First, because (4.35) is a convex function, a simple subgradient condition gives that (X, Q) is a global minimum of $F(X, Q)$ iff the following two conditions are satisfied

$$-\tfrac{1}{\lambda}\nabla_X \ell(Y, X, Q) \in \partial\Omega_{\phi,\theta}(X) \tag{4.46}$$

$$-\nabla_Q \ell(Y, X, Q) \in \partial H(Q), \tag{4.47}$$

where $\nabla_X \ell(Y, X, Q)$ and $\nabla_Q \ell(Y, X, Q)$ denote the portions of the gradient of $\ell(Y, X, Q)$ corresponding to X and Q, respectively. If $(X^1, \dots, X^K, Q)$ is a local minimum of $f_r(X^1, \dots, X^K, Q)$, then (4.47) must be satisfied at $(X, Q) = (\Phi_r(X^1, \dots, X^K), Q)$, as this is implied by the first order optimality condition for a local minimum [80, Chap. 10], so we are left to show that (4.46) is also satisfied.

Turning to the factorization objective, if $(X^1, \dots, X^K, Q)$ is a local minimum of $f_r(X^1, \dots, X^K, Q)$, then $\forall (Z^1, \dots, Z^K)_r$ there exists $\delta > 0$ such that $\forall \epsilon \in (0, \delta)$ we have $f_r(X^1 + \epsilon^{1/p} Z^1, \dots, X^K + \epsilon^{1/p} Z^K, Q) \geq f_r(X^1, \dots, X^K, Q)$. If we now consider search directions $(Z^1, \dots, Z^K)_r$ of the form

$$(Z_j^1, \dots, Z_j^K) = \begin{cases} (0, \dots, 0) & j \neq i_0 \\ (z^1, \dots, z^K) & j = i_0 \end{cases}, \tag{4.48}$$

where i_0 is the index such that $(X_{i_0}^1, \dots, X_{i_0}^K) = (0, \dots, 0)$, then for $\epsilon \in (0, \delta)$, we have

$$\ell(Y, \Phi_r(X^1, \ldots, X^K), Q) + \lambda \sum_{i=1}^{r} \theta(X_i^1, \ldots, X_i^K) + H(Q) \leq \tag{4.49}$$

$$\begin{aligned}&\ell(Y, \Phi_r(X^1 + \epsilon^{1/p} Z^1, \ldots, X^K + \epsilon^{1/p} Z^K), Q) + \\ &\lambda \sum_{i=1}^{r} \theta(X_i^1 + \epsilon^{1/p} Z_i^1, \ldots, X_i^K + \epsilon^{1/p} Z_i^K) + H(Q) =\end{aligned} \tag{4.50}$$

$$\begin{aligned}&\ell(Y, \sum_{i \neq i_0} \phi(X_i^1, \ldots, X_i^K) + \phi(X_{i_0}^1 + \epsilon^{1/p} Z_{i_0}^1, \ldots, X_{i_0}^K + \epsilon^{1/p} Z_{i_0}^K), Q) + \\ &\lambda \sum_{i \neq i_0} \theta(X_i^1, \ldots, X_i^K) + \lambda \theta(X_{i_0}^1 + \epsilon^{1/p} Z_{i_0}^1, \ldots, X_{i_0}^K + \epsilon^{1/p} Z_{i_0}^K) + H(Q) =\end{aligned} \tag{4.51}$$

$$\begin{aligned}&\ell(Y, \Phi_r(X^1, \ldots, X^K) + \epsilon \phi(z^1, \ldots, z^K), Q) + \\ &\lambda \sum_{i=1}^{r} \theta(X_i^1, \ldots, X_i^K) + \epsilon \lambda \theta(z^1, \ldots, z^K) + H(Q).\end{aligned} \tag{4.52}$$

The equality between (4.51) and (4.52) comes from the special form of Z given by (4.48), the fact that $(X_{i_0}^1, \ldots, X_{i_0}^K) = (0, \ldots, 0)$, and the positive homogeneity of ϕ and θ. Rearranging terms, we now have

$$\begin{aligned}&\epsilon^{-1}[\ell(Y, \Phi_r(X^1, \ldots, X^K) + \epsilon \phi(z^1, \ldots, z^K), Q) - \ell(Y, \Phi_r(X^1, \ldots, X^K), Q)] \\ &\quad \geq -\lambda \theta(z^1, \ldots, z^K).\end{aligned} \tag{4.53}$$

Taking the limit of (4.53) as $\epsilon \searrow 0$, we note that the left side of the inequality is simply the definition of the one-sided directional derivative of $\ell(Y, \Phi_r(X^1, \ldots, X^K), Q)$ in the direction $(\phi(z^1,\dots,z^K),0)$, which combined with the differentiability of $\ell(X,Q)$, gives

$$\left\langle \phi(z^1,\dots,z^K), \nabla_X \ell(Y, \Phi_r(X^1,\dots,X^K),Q)\right\rangle \geq -\lambda\theta(z^1,\dots,z^K). \tag{4.54}$$

Because $(z^1,\dots,z^K)$ was arbitrary, we have established that

$$\begin{aligned} &\left\langle \phi(z^1,\dots,z^K), -\tfrac{1}{\lambda}\nabla_X \ell(Y, \Phi_r(X^1,\dots,X^K),Q)\right\rangle \leq \theta(z^1,\dots,z^K) \;\; \forall (z^1,\dots,z^K) \\ &\qquad \iff \Omega^\circ_{\phi,\theta}(-\tfrac{1}{\lambda}\nabla_X \ell(Y, \Phi_r(X^1,\dots,X^K),Q)) \leq 1, \end{aligned} \tag{4.55}$$

where the equivalence is seen by identical arguments to those used in the proof of Proposition 11. Further, if we choose β to be vector of all ones in Lemma 2, we get

$$\sum_{i=1}^r \theta(X_i^1,\dots,X_i^K) = \left\langle \Phi_r(X^1,\dots,X^K), -\tfrac{1}{\lambda}\nabla_X \ell(Y, \Phi_r(X^1,\dots,X^K),Q)\right\rangle. \tag{4.56}$$

This fact, combined with (4.55), Lemma 1, and Proposition 12 shows that $-\frac{1}{\lambda}\nabla_X \ell(Y,\Phi_r(X^1,\dots,X^K),Q) \in \partial\Omega_{\phi,\theta}(\Phi_r(X^1,\dots,X^K))$, completing the result. ■

From this result, we can then test the global optimality of any local minimum (regardless of whether it has an all-zero slice or not) from the immediate corollary:

Corollary 2 *Given a function $f_r(X^1,\dots,X^K,Q)$ of the form given in (4.36), any* *local minimizer of the optimization problem*

$$\min_{(X^1,\ldots,X^K)_r,Q} f_r(X^1,\ldots,X^K,Q) \tag{4.57}$$

is a global minimizer if $f_{r+1}([X^1\ 0],\ldots,[X^K\ 0],Q)$ *is a local minimizer of* f_{r+1}.

Proof. Note that from the structure of $f_r(X^1,\ldots,X^K,Q)$ the following two problems are equivalent

$$\begin{aligned}&\min_{(X^1,\ldots,X^K)_r,Q} f_r(X^1,\ldots,X^K,Q) \equiv \\ &\min_{([X^1\ x^1],\ldots,[X^K\ x^K]),Q} f_{r+1}([X^1\ x^1],\ldots,[X^K\ x^k],Q)\ \text{ s.t. }\ (x^1,\ldots,x^K)=(0,\ldots,0).\end{aligned} \tag{4.58}$$

If we remove the equality constraint we then have that $\min f_{r+1} \leq \min f_r$, and if the condition of the corollary is satisfied, then $([X^1\ 0],\ldots,[X^K\ 0],Q)$ is a global minimizer for f_{r+1} due to Theorem 8. This then implies that $(X^1,\ldots,X^K,Q)$ is global minimizer of f_r due to the equivalence in (4.58). ■

4.5.2 Finding Global Minima

From the results of Theorem 8, we have a sufficient condition to guarantee the global optimality of a given local minimum. Building on this result, we now are also able to show that if we let the size of the factorized variables (r) become large enough, then from any initialization we can always find a global minimizer of $f_r(X^1, \ldots, X^K, Q)$ using a purely local descent strategy. Specifically, we have the following result.

Theorem 9 *Given a function* $f_r(X^1, \ldots, X^K, Q)$ *as defined by* (4.36)*, if* $r >$ card(X) *then from any point* $(Z^1, \ldots, Z^K, Q)$ *such that* $f_r(Z^1, \ldots, Z^K, Q) < \infty$ *there must exist a non-increasing path from* $(Z^1, \ldots, Z^K, Q)$ *to a global minimizer of* $f_r(X^1, \ldots, X^K, Q)$*.*

Proof. Clearly if $(Z^1, \ldots, Z^K, Q)$ is not a local minimum, then we can follow a decreasing path until we reach a local minimum. Having arrived at a local minimum, $(\tilde{X}^1, \ldots, \tilde{X}^K, \tilde{Q})$, if $(\tilde{X}^1_i, \ldots, \tilde{X}^K_i) = (0, \ldots, 0)$ for any $i \in \{1, \ldots, r\}$ then from Theorem 8 we must be at a global minimum. Similarly, if for any $i_0 \in \{1, \ldots, r\}$ we have $\phi(\tilde{X}^1_{i_0}, \ldots, \tilde{X}^K_{i_0}) = 0$ then we can scale the slice $(\alpha\tilde{X}^1_{i_0}, \ldots, \alpha\tilde{X}^K_{i_0})$ as α goes from $1 \to 0$ without increasing the objective function. Once $\alpha = 0$ we will then have an all zero slice in the factor tensors, so from Theorem 8 we are either at a global minimum or a local descent direction must exist from that point. We are thus left to show that a non-increasing path to a global minimizer must exist from any local minima such that $\phi(\tilde{X}^1_i, \ldots, \tilde{X}^K_i) \neq 0$ for all $i \in \{1, \ldots, r\}$.

First, note that because $r >$ card(X) there must exist $\hat{\beta} \in \mathbb{R}^r$ such that $\hat{\beta} \neq 0$ and $\sum_{i=1}^r \hat{\beta}_i \phi(\tilde{X}^1_i, \ldots, \tilde{X}^K_i) = 0$. Further, from Lemma 2 we must have that $\sum_{i=1}^r \hat{\beta}_i \theta(\tilde{X}^1_i, \ldots, \tilde{X}^K_i) = \left\langle -\frac{1}{\lambda}\nabla_X \ell(Y, \Phi_r(X^1, \ldots, X^K), Q), \sum_{i=1}^r \hat{\beta}_i \phi(X^1_i, \ldots, X^K_i) \right\rangle = 0$. Due to the nondegeneracy of the (ϕ, θ) pair we must have $\theta(\tilde{X}_i^1, \ldots, \tilde{X}_i^K) > 0, \quad \forall i \in \{1, \ldots, r\}$, which implies that at least one entry of $\hat{\beta}$ must be strictly less than zero.

Without loss of generality, assume $\hat{\beta}$ is scaled so that $\min_i \hat{\beta}_i = -1$. Now, for all $(\gamma, i) \in \{[0,1]\} \times \{1, \ldots, r\}$, let us define

$$(R_i^1(\gamma), \ldots, R_i^K(\gamma)) \equiv ((1+\gamma\hat{\beta}_i)^{1/p}\tilde{X}_i^1, \ldots, (1+\gamma\hat{\beta}_i)^{1/p}\tilde{X}_i^K) \tag{4.59}$$

where p is the degree of positive homogeneity of (ϕ, θ). Note that by construction $(R^1(0), \ldots, R^K(0)) = (\tilde{X}^1, \ldots, \tilde{X}^K)$ and that for $\gamma = 1$ there must exist $i_0 \in \{1, \ldots, r\}$ such that $(R_{i_0}^1(1), \ldots, R_{i_0}^K(1)) = (0, \ldots, 0)$.

Further, from the positive homogeneity of (ϕ, θ) we have $\forall \gamma \in [0,1]$

$$\begin{aligned} f_r(R^1(\gamma), \ldots, R^K(\gamma), \tilde{Q}) =& \ell\left(Y, \sum_{i=1}^r \phi(\tilde{X}_i^1, \ldots, \tilde{X}_i^K) + \gamma \sum_{i=1}^r \hat{\beta}_i \phi(\tilde{X}_i^1, \ldots, \tilde{X}_i^K),\, \tilde{Q}\right) + \\ & \lambda\gamma \sum_{i=1}^r \hat{\beta}_i \theta(\tilde{X}_i^1, \ldots, \tilde{X}_i^K) + \lambda \sum_{i=1}^r \theta(\tilde{X}_i^1, \ldots, \tilde{X}_i^K) + H(\tilde{Q}) \end{aligned} \tag{4.60}$$

$$= \ell(Y, \Phi_r(\tilde{X}^1, \ldots, \tilde{X}^K), \tilde{Q}) + \lambda \sum_{i=1}^r \theta(\tilde{X}_i^1, \ldots, \tilde{X}_i^K) + H(\tilde{Q}) \tag{4.61}$$

$$= f_r(\tilde{X}^1, \ldots, \tilde{X}^K, \tilde{Q}), \tag{4.62}$$

where the equality between (4.60) and (4.61) is seen by recalling that $\sum_{i=1}^r \hat{\beta}_i \phi(\tilde{X}_i^1, \ldots, \tilde{X}_i^K) = 0$ and $\sum_{i=1}^r \hat{\beta}_i \theta(\tilde{X}_i^1, \ldots, \tilde{X}_i^K) = 0$.

As a result, as γ goes from $0 \to 1$ we can traverse a path from $(\tilde{X}^1, \ldots, \tilde{X}^K, \tilde{Q}) \to (R^1(1), \ldots, R^K(1), \tilde{Q})$ without changing the value of f_r. Also recall that by construction $(R^1_{i_0}(1), \ldots, R^K_{i_0}(1)) = (0, \ldots, 0)$, so if $(R^1(1), \ldots, R^K(1), \tilde{Q})$ is a local minimizer of f_r then it must be a global minimizer due to Theorem 8. If $(R^1(1), \ldots, R^K(1), \tilde{Q})$ is not a local minimizer then there must exist a descent direction and we can iteratively apply this result until we reach a global minimizer, completing the proof. ■

We note that our proof is constructive in nature and describes a meta-algorithm (outlined in Algorithm 3) which can be used with any local-descent optimization strategy to guarantee convergence to a global minimum. Further, note also that our definition of a local minimum includes the "saddle plateau" critical regions from Figure 4.1. As a result, our proof also guarantees that for points on such plateaus we can rescale the tensor slices by the β terms in Algorithm 3 to arrive at a point from where a descent direction is guaranteed to exist (the green points in figure 4.1), and finding the necessary β terms is equivalent to finding a vector in the null space of a $\operatorname{card}(X) \times r$ matrix.

Corollary 3 *Algorithm 3 will find a global minimum of* $f_r(X^1, \ldots, X^K, Q)$ *as defined in* (4.36)*. If* r *is initialized to be greater than* $\operatorname{card}(X)$*, then the size of the factorized variables will not increase. Otherwise, the algorithm will terminate with* $r \leq \operatorname{card}(X) + 1$*.*

While in general the size of the factorization (r) might increase as the algorithm proceeds, as a worst case, it is guaranteed that a global minimum can be found with a finite r never growing larger than $\text{card}(X) + 1$. Also note that this is a worst case upper bound on r for the most general form of our framework and that for specific choices of ϕ and θ the bound on the maximum r required can be significantly lowered. The main requirement for lowering the upper bound on r is whether from a given local minimum there exists a transformation of the variables that allows us to set one slice of the tensors to 0 without increasing the objective function. For example, in nuclear norm matrix factorization problems we have $\Phi_r(U, V) = UV^T$ and $\theta(U_i, V_i) = \frac{1}{2}(\|U_i\|_F^2 + \|V_i\|_F^2)$. Due to the rotational invariance of the Frobenius norm, if either U or V is rank deficient we can multiply by a orthonormal matrix R to make one of the columns all zeros without changing the objective function, i.e., $\Phi_r(UR, VR) = \Phi(U, V)$ and $\theta(U_iR, V_iR) = \theta(U_i, V_i)$, which implies a non-increasing path to a global minimizer must exist as soon as $r > \min\{\text{card}(U_i), \text{card}(V_i)\}$.

4.6 Discussion and Conclusions

We begin our discussion by noting the limitations of our results and cautioning that many challenges still exist to applying them in practice. In particular, many algorithms based on alternating minimization can typically only guarantee convergence to a critical point, and with the inherent non-convexity of the problem, verifying whether a given critical point is also a local minima can be a challenging problem on its own. Further, to use our results to guarantee global optimality, it is necessary

Algorithm 3 (Local Descent Generalized Factorization Meta-Algorithm)

```
input $p$ - Degree of positive homogeneity for $(\phi, \theta)$
input $\{(X^1, \dots, X^K)_r,\ Q\}$ - Initialization for variables
  while Not Converged do
    Perform local descent on variables $\{(X^1, \dots, X^K), Q\}$ until arriving at a local minimum $\{(\tilde{X}^1, \dots, \tilde{X}^K), \tilde{Q}\}$
    if $\exists i_0 \in \{1, \dots, r\}$ such that $(\tilde{X}^1_{i_0}, \dots, \tilde{X}^K_{i_0}) = (0, \dots, 0)$ then
      $\{(\tilde{X}^1, \dots, \tilde{X}^K), \tilde{Q}\}$ is a global minimum. Return.
    else
      if $\exists \beta \in \mathbb{R}^r \backslash 0$ such that $\sum_{i=1}^r \beta_i \phi(\tilde{X}^1_i, \dots, \tilde{X}^K_i) = 0$ then
        Scale $\beta$ so that $\min_i \beta_i = -1$
        Set $(X^1_i, \dots, X^K_i) = ((1+\beta_i)^{1/p} \tilde{X}^1_i, \dots, (1+\beta_i)^{1/p} \tilde{X}^K_i),\ \forall i \in \{1, \dots, r\}$
      else
        Increase size of factorized variables by appending an all zero slice
        $(X^1, \dots, X^K)_{r+1} = ([\tilde{X}^1\ 0], \dots, [\tilde{X}^K\ 0])$
      end if
      Set $Q = \tilde{Q}$
      Continue loop
    end if
  end while
```

to verify whether a descent direction exists from a point where one of the tensor slices is all 0, which is analogous to the results from gradient boosting that require us to test if adding another element to the factorization can reduce the objective function [74, 76]. As shown in Theorem 8, in general this requires solving the polar problem (4.23), which as we noted above can be quite challenging. For example, even in the seemingly simple case of matrix factorization with the $\|\cdot\|_{u,v}$ norm, choosing both vector norms to be l_q norms with one the commonly used values of $q \in \{1, 2, \infty\}$ results in polar problems with widely varying computational complexity depending on the particular choice of norm: the $\|X\|_{1,1}$ polar is simply the largest absolute value of all the entries of X; the $\|X\|_{2,2}$ polar is the largest singular value of X; but the

$\|X\|_{\infty,\infty}$ polar is NP-hard to compute [64]. More complicated elemental mappings and regularizers, such as those associated with tensor decompositions or ReLU neural networks, also typically result in NP-hard polar problems [33, 81].

Nevertheless, despite these practical challenges, we emphasize that our results guarantee that global minimizers can be found from purely local descent if the optimization problem falls within the general framework we have described here. As a result, even if the particular local descent strategy one chooses for a specific problem does not come with guaranteed convergence to a local minimum, the scope of the problem is still vastly reduced from a full global optimization. There is no need, in theory, to consider multiple initializations or more complicated (and much larger scale) techniques to explore the entire search space. Further, our analysis also provides multiple insights into the behavior of factorization problems and offers simple guiding principles regarding the design of factorization problems, several of which we discuss below.

4.6.1 Balanced Degrees of Homogeneity

The first key principle for our analysis is that balancing the degree of positive homogeneity between the regularization function and the mapping function is crucial. Here we have analyzed a mapping Φ with the particular form given in (4.6). We conjecture our results can likely be generalized to include additional positively homogeneous factorization mappings and regularizers (which we briefly discuss in the next section), but even for more general mappings and regularization functions, requiring the degrees of positive homogeneity to match between the regularization function and the mapping function will be critical to showing results similar to those we present here. In general, if the degrees of positive homogeneity do not match between the factorization mapping and the regularization function, then it either becomes impossible to make guarantees regarding the global optimality of a local minimum, or the regularization function does nothing to limit the size of the factorization, so the degrees of freedom in the model are largely determined by the user defined choice of r.

As a demonstration of these phenomena, first consider the case where we have a general mapping, $\Phi(X^1, \ldots, X^K)$, which is positively homogeneous with degree p (but which is not assumed to have form (4.6)). Now, consider a general regularization function, $G(X^1, \ldots, X^K)$, which is positively homogeneous with degree $p' < p$, then the following proposition provides a simple counter-example demonstrating that in general it is not possible to guarantee that a global minimum can be found from local descent from an arbitrary initialization.

Proposition 13 *Let $\ell : \mathbb{R}^D \to \mathbb{R}$ be a convex function with $\partial\ell(0) \neq \emptyset$; let $\Phi : \mathbb{R}^{D^1} \times \ldots \times \mathbb{R}^{D^K} \to \mathbb{R}^D$ be a positively homogeneous mapping with degree p; and let $G : \mathbb{R}^{D^1} \times \ldots \times \mathbb{R}^{D^K} \to \mathbb{R}_+$ be a positively homogeneous function with degree $p' < p$ such that $G(0, \ldots, 0) = 0$ and $G(X^1, \ldots, X^K) > 0 \ \ \forall \{(X^1, \ldots, X^K) : \Phi(X^1, \ldots, X^K) \neq$*

$0\}$. *Then, the optimization problem given by*

$$\min_{(X^1,\ldots,X^K)} \tilde{f}(X^1,\ldots,X^K) = \ell(\Phi(X^1,\ldots,X^K)) + G(X^1,\ldots,X^K) \tag{4.63}$$

has a local minimum at $(X^1,\ldots,X^K) = (0,\ldots,0)$*. Additionally,* $\forall(X^1,\ldots,X^K)$ *such that* $\Phi(X^1,\ldots,X^K) \neq 0$ *there exists a* δ *such that* $\forall\epsilon \in (0,\delta)$ $\tilde{f}(\epsilon X^1,\ldots,\epsilon X^K) > \tilde{f}(0,\ldots,0)$*.*

Proof. Consider $\tilde{f}(\epsilon X^1,\ldots,\epsilon X^K) - \tilde{f}(0,\ldots,0)$. This gives

$$\ell(\Phi(\epsilon X^1,\ldots,\epsilon X^K)) + G(\epsilon X^1,\ldots,\epsilon X^K) - \ell(0) - G(0,\ldots,0) = \tag{4.64}$$

$$\ell(\epsilon^p\Phi(X^1,\ldots,X^K)) - \ell(0) + \epsilon^{p'}G(X^1,\ldots,X^K) \geq \tag{4.65}$$

$$\epsilon^p \left\langle \partial\ell(0), \Phi(X^1,\ldots,X^K)\right\rangle + \epsilon^{p'}G(X^1,\ldots,X^K), \tag{4.66}$$

where the inequality is simply due to the definition of the subgradient of a convex function. Recall that $p > p'$ and $\Phi(X^1,\ldots,X^K) \neq 0 \iff G(X^1,\ldots,X^K) > 0$, so $\forall(X^1,\ldots,X^K)$, $\tilde{f}(\epsilon X^1,\ldots,\epsilon X^K) - \tilde{f}(0,\ldots,0) \geq 0$ for $\epsilon > 0$ and sufficiently small, with equality iff $G(X^1,\ldots,X^K) = 0 \iff \Phi(X^1,\ldots,X^K) = 0$, giving the result. ■

The above proposition shows that unless we have the special case where $(X^1,\ldots,X^K) = (0,\ldots,0)$ happens to be a global minimizer, then there will always exist a local minimum at the origin, and from the origin it will always be necessary to take an increasing path to escape the local minimum. The case described above, where $p > p'$, is arguably the more common situation for mismatched degrees of homogeneity (as opposed to $p < p'$), and a typical example might be an objective function such as

$$\ell(\Phi(X^1, \ldots, X^K)) + \lambda \sum_{i=1}^{K} \|X^i\|_{(i)}^{p'}, \tag{4.67}$$

where Φ is a positively homogeneous mapping with degree $K > 2$ (e.g., the mapping of a deep neural network) but p' is typically taken to be only 1 or 2 depending on the particular choice of norms (e.g., $\|X^i\|_F^2$ or $\|X^i\|_1$).

Conversely, in the situation where $p' > p$, then it is often the case that the regularization function is not sufficient to "limit" the size of the factorization, in the sense that the objective function can always be decreased by allowing the size of the factors to grow. As a simple example, consider the case of matrix factorization with the objective function

$$\ell(UV^T) + \lambda(\|U\|^{p'} + \|V\|^{p'}). \tag{4.68}$$

If the size of the factorization doubles, then we can always take $[\frac{\sqrt{2}}{2}U \ \frac{\sqrt{2}}{2}U][\frac{\sqrt{2}}{2}V \ \frac{\sqrt{2}}{2}V]^T = UV^T$, so if $(\frac{\sqrt{2}}{2})^{p'}(\|[U \ U]\|^{p'} + \|[V \ V]\|^{p'}) < \|U\|^{p'} + \|V\|^{p'}$, then the objective function can always be decreased by simply duplicating and scaling the existing factorization. It is easily verified that the above inequality is satisfied for many choices of norms (for example, all the l_q norms with $q \geq 1$) when $p' > 2$. As a result, this implies that the degrees of freedom in the model will be largely dependent on the particular choice of the number of columns in (U, V), since in general the objective function is typically decreased by having all entries of (U, V) be non-zero.

4.6.2 Further Generalization

In our analysis we have focused on mappings with the particular form given in (4.6) for simplicity of presentation, but we note that by similar arguments to those presented above it is possible to consider many other potential mappings. For example, a more general factorization regularizer could be defined as

$$\Omega_{\Phi,\Theta}(X) = \inf_{(X^1,\ldots,X^K):\Phi(X^1,\ldots,X^K)=X} \Theta(X^1,\ldots,X^K), \tag{4.69}$$

where now Φ is an arbitrary positively homogeneous mapping defined over factors $(X^1,\ldots,X^K)$ that are allowed to change size along multiple dimensions, and Θ is an arbitrary positively homogeneous, positive semidefinite function. Assuming the degrees of positive homogeneity match between Φ and Θ, it is easy to show that (4.69) will be positively homogeneous with degree 1, so if it can also be shown that for any (X, Z) there must exist a factorization that satisfies the triangle inequality $\Omega_{\Phi,\Theta}(X + Z) \leq \Omega_{\Phi,\Theta}(X) + \Omega_{\Phi,\Theta}(Z)$, then $\Omega_{\Phi,\Theta}$ is a convex function and it has a Fenchel dual which will be an indicator function on a convex set, similar to the form in Proposition 11. As such generalizations substantially broaden the scope of the work and require a significantly expanded notational system, we save a full development of these ideas for future work in particular application domains.

4.6.3 Implications for Neural Networks

Examining our results specifically as they apply to deep neural networks, we note that there are a few simple principles suggested by our work to take into consideration when designing deep neural network systems. First note that from our analysis we have shown that neural networks which are based on positively homogeneous mappings can be regularized in the way we have outlined in our framework so that the optimization problem of training the network induces a convex regularization on the output of the network that limits the degrees of freedom within the network. We suggest that these results provide a partial theoretical explanation of the recently observed empirical phenomenon where replacing the traditional sigmoid or hyperbolic tangent non-linearities with positively homogeneous non-linearities, such as rectification and max-pooling, significantly boosts the speed of optimization and the performance of the network [44–46,48]. This has very recently been explored experimentally by [82] who note that the optimization problem of training a fully connected network with a single hidden layer using weight decay in the update of the network weights results in an optimization problem of the form

$$\min_{X^1,X^2} \ell(Y, \psi^+(VX^1)(X^2)^T) + \tfrac{\lambda}{2}(\|X^1\|_F^2 + \|X^2\|_F^2) \tag{4.70}$$

and has very strong analogies to the variational form of the nuclear norm. They then show empirically that such a network is robust to over-fitting even in the case where there are a large number of hidden units and noise deliberately added to the labels in the training set. Our results provide a generalization of this idea to multilayer networks, but note that standard weight decay typically implies a squared Frobenius norm term on the network weight variables as in (4.70), and given our discussion above regarding the importance of balanced degrees of homogeneity between the mapping and the regularizer, this is only appropriate for networks which are positively homogeneous of degree 2. In fact, many works have reported that traditional regularization on the network weight parameters, such as an l_1 or l_2 norm, does not result in good performance with multilayer ReLU networks and use other regularization strategies instead [44,47,83], and an immediate prediction of our analysis is that simply ensuring that the degrees of homogeneity are balanced between the mapping and the regularizer could be a significant factor in improving the performance of deep networks.

With regards to the degree of positive homogeneity of a network mapping, it is clear that adding an extra layer to the network with a positively homogeneous non-linearity typically increases the overall degree of the mapping by 1, but there are a few points to consider that can complicate the overall positive homogeneity of a network mapping. The first is contrast normalization. This is typically used in convolutional networks and takes the form of applying a transformation such as

$g_i = z_i/f(N(z_i))$, where g_i denotes the i^{th} output of the normalization layer, z_i denotes the i^{th} input to the normalization layer, and $f(N(z_i))$ denotes a function of the inputs to the normalization layer in a neighborhood surrounding z_i. If $f(N(z_i))$ is positively homogeneous with degree p', such as a norm raised to p', then the normalization layer is also a positively homogeneous transformation[3], but it "resets" the degree of positive homogeneity to be $1-p'$ at that stage in the network. As a result, care must be taken to ensure that sufficiently many layers exist following the normalization layer so that the overall degree of the network mapping becomes larger that 0. The second issue to consider with regards to staying strictly within the positively homogenous framework is the use of bias terms. For example, the output of a fully connected ReLU layer with bias terms is given by $G = \psi^+(ZW + B)$, where again G denotes the output of the layer, Z denotes the input to the layer, W denotes the connection weights, and B denotes the bias terms. If the input, Z, comes from lower layers of the network then it can already be a positively homogeneous function of the weight parameters in the lower layers, so B must be raised to an appropriate power to preserve the overall homogeneity of the mapping with respect to all the variables we are optimizing over (including B). For example, if Z is positively homogeneous of degree 3, then we could instead use bias terms of the form $G = \psi^+(Z * W + B_p^{(4)} - B_n^{(4)})$, where $B^{(4)}$ denotes raising each element to the 4'th power entry-wise, and the use of both the B_p and

[3] Usually, most response normalization layers are not strictly positively homogeneous as they add a small non-zero constant to the denominator to avoid division by 0, but if the constant is significantly smaller than the value of $f(N(z_i))$ it is a very close approximation of a positively homogeneous transformation.

B_n terms allows for negative bias terms. This then results in a mapping which is positively homogeneous with respect to all of the connection weights and bias terms in the network. Note that in this case, the θ regularization should also include the bias parameters as input.

We conclude by noting that a main limitation of our current framework in the analysis of currently existing state-of-the-art neural networks is that the form of the mapping we study here (4.6) implies that the network architecture must consist of r parallel subnetworks, where each subnetwork has a particular architecture defined by the elemental mapping ϕ. While many modern architectures have a certain degree of parallelization (for example, low level convolutional layers are often split onto multiple GPUs and then combined via fully connected layers), they do not typically approach the level of parallelization we consider here. Clearly, this is a limitation of our current results, but it also suggests at several interesting concepts to guide future work. The first concept is that neural networks which generate the output by taking the sum of multiple parallel subnetworks are highly conducive to efficient optimization. This idea, of linearly combining the outputs of multiple subnetworks, has clear analogies to ensemble methods like boosting and bagging and was a large motivation in the development of techniques such as drop-out, which stochastically approximates the average output of an exponential number of subnetworks [47]. The framework we present here is not an exact analogy to drop-out, as drop-out couples all of the subnetwork weights by a common parametrization, but combining the concept of summing multiple subnetworks along with considering more general forms of network mappings, which allow for common parametrization of the subnetworks, presents many opportunities for future work. Finally, as our framework is very general with respect to the particular choice of the elemental mapping, ϕ, and the elemental regularizer, θ, there exists a considerable potential for analyzing how these results can be improved and used in applications by considering specific choices of ϕ and θ.

4.6.4 Conclusions

We have presented a general framework which allows for a wide variety of non-convex factorization problems to be analyzed with tools from convex analysis and induces a convex regularizer on the output of the non-convex mapping. In particular, we have shown that for problems which can be placed in our framework, any local minimum can be guaranteed to be a global minimum of the non-convex factorization problem if one slice of the factorized tensors is all zero. Additionally, we have shown that if the non-convex factorization problem is done with factors of sufficient size, then from any feasible initialization it is always possible to find a global minimizer using a purely local descent algorithm.

Chapter 5

Applications

This chapter will explore applications of the structured matrix factorization theory introduced in Chapter 3. In particular, the matrix factorization method will be applied to two image processing problems: spatiotemporal segmentation of neural calcium imaging data and hyperspectral compressed recovery. Such problems are well modeled by low-rank linear models with square loss functions under the assumption that the spatial component of the data has low total variation (and is optionally sparse in the row and/or column space). Specifically, in this section we consider the following objective

$$\min_{U,V,Q} \frac{1}{2}\|Y - \mathcal{A}(UV^T) - \mathcal{B}(Q)\|_F^2 + \lambda \sum_i \|U_i\|_u \|V_i\|_v \tag{5.1}$$

$$\text{(optionally s.t.) } U \geq 0, V \geq 0$$

where $\mathcal{A}(\cdot)$ and $\mathcal{B}(\cdot)$ are linear operators, and the $\|\cdot\|_u$ and $\|\cdot\|_v$ norms have the form

$$\|\cdot\|_u = \nu_{u_1}\|\cdot\|_1 + \nu_{u_{TV}}\|\cdot\|_{TV} + \nu_{u_2}\|\cdot\|_2 \tag{5.2}$$

$$\|\cdot\|_v = \nu_{v_1}\|\cdot\|_1 + \nu_{v_{TV}}\|\cdot\|_{TV} + \nu_{v_2}\|\cdot\|_2, \tag{5.3}$$

for non-negative scalars ν. Recall that the anisotropic total variation of x is defined as [84]

$$\|x\|_{TV} \equiv \sum_i \sum_{j \in N_i} |x_i - x_j|, \tag{5.4}$$

where N_i denotes the set of pixels in the neighborhood of pixel i. Further, note that this objective function exactly fits within the framework introduced in Chapter 3 as we can define a rank-1 regularizer $\theta(u,v) = \|u\|_u\|v\|_v$ and optionally add indicator functions on u and/or v to enforce non-negativity constraints.

5.1 Solving the L1-TV Proximal Operator

In Chapter 3, Algorithm 1 was introduced as a general algorithm that could be used to solve structured matrix factorization problems. Further, from Theorem 7 we have an efficient means to solve the proximal operator of (5.2) and (5.3) optionally subject to non-negativity constraints, as we can first solve a proximal operator of the form

$$\arg\min_x \tfrac{1}{2}\|y - x\|_F^2 + \nu_1\|x\| + \nu_{TV}\|x\|_{TV} \text{ (optionally s.t.) } x \geq 0 \tag{5.5}$$

and then calculate the proximal operator of the l_2 norm with the solution to (5.5) as the argument. The only component that is missing to apply Algorithm 1 to the proposed objective function (5.1) is how to solve the proximal operator of the l_1 norm plus the total-variation pseudo-norm. To address this issue, note that (5.5) is equivalent to solving the problem

$$\arg\min_x \tfrac{1}{2}\|y - x\|_F^2 + \|Gx\|_1 \text{ (optionally s.t.) } x \geq 0 \tag{5.6}$$

$$G = \begin{bmatrix} \nu_1 I \\ \\ \nu_{TV}\Delta \end{bmatrix}, \tag{5.7}$$

where Δ is a matrix that takes the difference between neighboring elements of x. Using standard Lagrangian duality arguments, such as those presented in [85], it is easily shown that the dual problem of (5.6) is equivalent to

$$\min_\gamma \tfrac{1}{2}\|y - G^T\gamma\|_F^2 \;\text{ s.t. }\; \|\gamma\|_\infty \leq 1 \tag{5.8}$$

$$\text{(optionally s.t.) } y - G^T\gamma \geq 0$$

with the primal-dual relationship $x = y - G^T\gamma$. To solve (5.8), we note that due to the special structure of G, one can calculate the global optimum of an individual element of γ extremely quickly if the other elements in γ are held constant. Thus, to solve (5.8) we cycle through making updates to the elements of γ while checking the duality gap for convergence. For the values of the ν parameters typically used in our experiments, this strategy converges after a relatively small number of cycles through the γ variables, and due to the fact that the updates to the γ variables themselves are very easy to calculate this strategy provides a very efficient means of solving the proximal operator in (5.5).

5.2 Neural Calcium Imaging Segmentation

Returning now to applications of the method, the first application considered is the segmentation of calcium image data. Calcium imaging is a rapidly growing microscopy technique in neuroscience that records fluorescent images from neurons that have been loaded with either synthetic or genetically encoded fluorescent calcium indicator molecules. When a neuron fires an electrical action potential (or spike), calcium enters the cell and binds to the fluorescent calcium indicator molecules, changing the fluorescence properties of the molecule. By recording movies of the calcium-induced fluorescent dynamics it is possible to infer the spiking activity from large populations of neurons with single neuron resolution [86]. If we are given the fluorescence time series from a single neuron, inferring the spiking activity from the fluorescence time series is well modeled via a Lasso style estimation,

$$\hat{s} = \arg\min_{s \geq 0} \frac{1}{2} \left\| y - Ds \right\|_2^2 + \lambda \left\| s \right\|_1 , \tag{5.9}$$

where $y \in \mathbb{R}^t$ is the fluorescence time series (normalized by the baseline fluorescence), $\hat{s} \in \mathbb{R}^t$ denotes the estimated spiking activity (each entry of $\hat{s}$ is monotonically related to the number of action potentials the neuron has during that imaging frame), and $D \in \mathbb{R}^{t \times t}$ is a matrix that applies a convolution with a known decaying exponential to model the change in fluorescence resulting from a neural action potential [7].

One of the challenges in neural calcium imaging is that the data can have a significant noise level, making manual segmentation challenging. Additionally, it is also possible to have two neurons overlap in the spatial domain if the focal plane of the microscope is thicker than the size of the distinct neural structures in the data, making simultaneous spatiotemporal segmentation necessary. A possible strategy to address these issues would be to extend (5.9) to estimate spiking activity for the whole data volume via the objective

$$\hat{S} = \arg\min_{S \geq 0} \frac{1}{2} \left\| Y - DS \right\|_F^2 + \lambda \left\| S \right\|_1 , \tag{5.10}$$

where now each column of $Y \in \mathbb{R}^{t \times p}$ contains the fluorescent time series for a single pixel and the corresponding column of $\hat{S} \in \mathbb{R}^{t \times p}$ contains the estimated spiking activity for that pixel. However, due to the significant noise often present in the actual data, solving (5.10) directly typically gives poor results. To address this issue, [87] have suggested adding an additional low-rank regularization to (5.10) based on the knowledge that if two pixels are from the same neural structure they should have identical spiking activities, giving S a low-rank structure with the rank of S corresponding to the number of neural structures in the data. Specifically, they propose an objective to promote low-rank and sparse spike estimates,

$$\hat{S} = \arg\min_{S \geq 0} \frac{1}{2} \|Y - DS\|_F^2 + \lambda \|S\|_1 + \lambda_2 \|S\|_* \tag{5.11}$$

and then estimate the temporal and spatial features by performing a non-negative matrix factorization of $\hat{S}$.

While (5.11) provides a nice model of spiking activity within a dataset, recall from the introductory discussion that in factorization problems solving a problem in the product space (i.e. solving for X) is somewhat unsatisfactory as it does not provide us with the desired factors. Fortunately, it can be shown that problem (5.1) is equivalent to a standard Lasso estimation when both the row space and column space are regularized by the l_1 norm [54], while combined l_1, l_2 norms of the form (5.2) and (5.3) with $\nu_{u_{TV}} = 0$ promote solutions that are simultaneously sparse and low rank. Thus, the projective tensor norm can generalize the two prior methods for calcium image processing by providing regularizations that are sparse or simultaneously sparse and low-rank, while also having the advantage of solving for the desired factors directly.

Further, by working in the factorized space we can also model additional known structure in the factors. In particular, we extend the two above formulations by noting that if two pixels are neighboring each other it is likely that they are from the same neural structure and thus have identical spiking activity, implying low total variation in the spatial domain. We demonstrate the flexible nature of our formulation (5.1) by using it to process calcium image data with regularizations that are either sparse, simultaneously sparse and low-rank, or simultaneously sparse, low-rank, and with low total variation. Additionally, by optimizing (5.1) to simultaneously estimate temporal spiking activity U and neuron shape V, with $\mathcal{A}(UV^T) = DUV^T$, we inherently find spatial and temporal features in the data (which are largely non-negative even though we do not explicitly constrain them to be) directly from our optimization without the need for an additional matrix factorization step. Finally, note that the $\mathcal{B}(Q)$ term can be used to fit the background intensity of the pixels by taking $\mathcal{B}(Q) = \mathbf{1}Q^T$ for a vector $Q \in \mathbb{R}^p$, and if the data exhibits temporal variations in pixel intensities not due to calcium activity, such as from slow movements of the sample or photo-bleaching, this can also be modeled via an appropriate choice of $\mathcal{B}$ operator. For the experiments presented here the data has been normalized by background intensity, so the $\mathcal{B}(Q)$ term is not used.

5.2.1 Simulation Data

We first tested our algorithm on a simulated phantom dataset which was constructed with 19 non-overlapping spatial regions (see Figure 5.4, left panel) and 5 randomly timed action potentials and corresponding calcium dynamics per region. The phantom was 200 frames of 120x125 images, and the decaying exponentials in D had a time constant of $1.33\bar{3}\,sec$ with a simulated sampling rate of $10\,Hz$. Gaussian white noise was added to the modeled calcium signal to produce a SNR of approximately -16dB.

Using this phantom, we used Algorithm 1 to solve the formulation given in (5.1) with different ν parameters for the norms in (5.2) and (5.3). In particular we did experiments using just sparse and low-rank regularization by taking $[\nu_{u_1}, \nu_{u_{TV}}, \nu_{u_2}] = [\nu_{v_1}, \nu_{v_{TV}}, \nu_{v_2}] = [1, 0, 1]$ and $\lambda = 1.5\sigma$, where σ denotes the standard deviation of the Gaussian noise. Then, to demonstrate the benefit of adding total-variation regularization in the spatial domain, we used ν parameters given by $\lambda = 0.4\sigma$, $[\nu_{u_1}, \nu_{u_{TV}}, \nu_{u_2}] = [1, 0, 1]$, and $[\nu_{v_1}, \nu_{v_{TV}}, \nu_{v_2}] = [1, 1, 1]$, with an 8-connected lattice for the total-variation graph[1]. For the sparse + low-rank condition U was initialized to be an identity matrix. For the experiments that include the total-variation regularization we again conducted experiments with U initialized to be an identity matrix, and to study the effects of different initializations, we additionally also performed experiments with U initialized with 50 columns, where each entry in U was

[1] The regularization parameters were roughly tuned by hand to produce the best qualitative results for the two experimental conditions (i.e. sparse + low-rank w/wo total-variation)

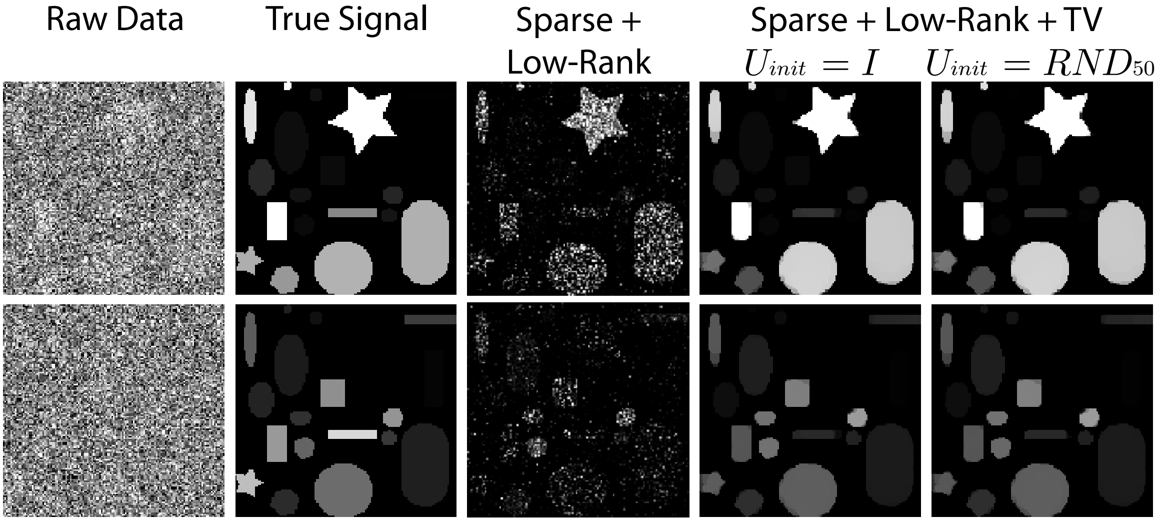

Figure 5.1: Example reconstructed calcium signal from phantom dataset. The two rows correspond to two different example image frames. *From left to right*: Raw data. True calcium signal. Reconstruction with sparse + low-rank regularization. Reconstruction with sparse + low-rank + total-variation regularization with U initialized as an identity matrix. Reconstruction with sparse + low-rank + total-variation regularization with U initialized as 50 columns of random values uniformly distributed between $[0, 1]$.

initialized to a random value uniformly distributed between $[0, 1]$ (in all cases V was initialized as 0).

Figure 5.1 shows two example reconstructions of the calcium signal estimated with our algorithm with different regularization conditions. Figure 5.2 shows example spatial components recovered by our algorithm as well as spatial components recovered by PCA for comparison. For each case, the components shown are the first 9 most significant components (i.e. those with the largest value of $\|U_i\|_u \|V_i\|_v)^2$. Note that

[2] Note that the differences in the specific components shown in Figure 5.2 between the two initializations of U is due to the fact that the structure of the objective function (5.1) allows for components to be duplicated without changing the value of the objective function. For example, suppose we have $U = [U_1 \; U_2]$ and $V = [V_1 \; V_2]$, then taking $\tilde{U} = [U_1 \; 0.2U_2 \; 0.8U_2]$ and $\tilde{V} = [V_1 \; V_2 \; V_2]$ will give identical objective function values.

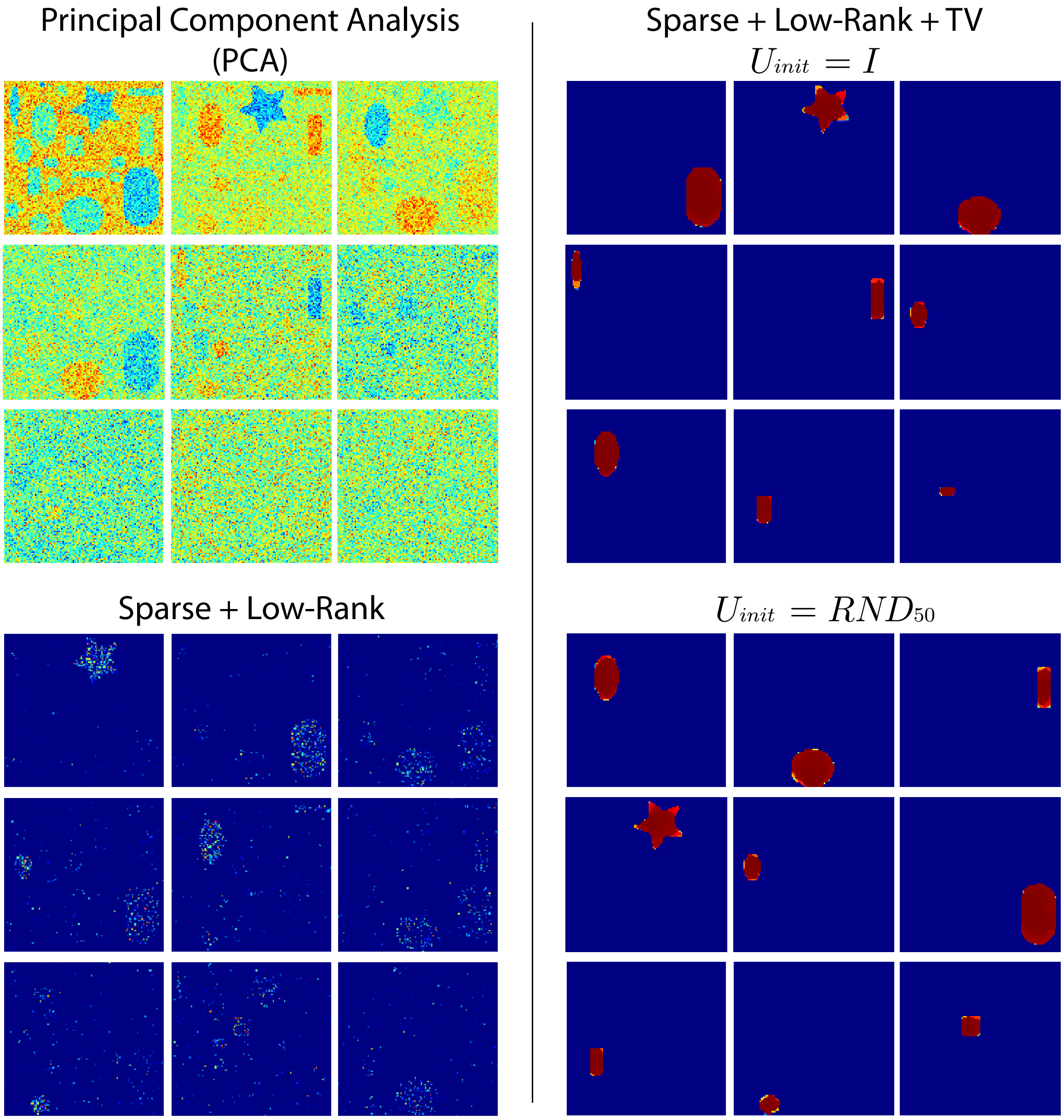

Figure 5.2: Example recovered spatial components from phantom dataset. *Top Left*: First 9 most significant spatial components recovered via Principal Component Analysis (PCA). *Bottom Left*: First 9 most significant spatial components recovered with sparse and low-rank regularization. *Top Right*: First 9 most significant spatial components recovered using sparse, low-rank, and total variation regularization, with U initialized as an identity matrix. *Bottom Right*: Same as the top right panel but with U initialized as 50 columns of random values uniformly distributed between $[0, 1]$

.

although we only show the first 9 spatial components here for compactness, the remaining components also closely correspond to the true spatial regions and allow for the true spatial segmentation to be recovered (see below).

The recovered temporal components for the 9 regions shown in Figure 5.2 are plotted in Figure 5.3 along with the corresponding true temporal spike times (red dots) for the sparse + low-rank + total-variation regularization conditions. The final recovered spatial segmentation is shown in 5.4 for the sparse + low-rank + total-variation experiments with the two different initializations for U. This segmentation was generated by simply finding connected components of the non-zero support of the spatial components, then any two connected components that overlapped by more than 10% were merged (note that this step is largely only necessary to combine duplicate components – see footnote 2 – and the results are very insensitive to the choice of the percentage of overlap as any duplicated components had almost identical non-zero supports). Despite the very high noise level, adding the appropriate structure of sparse + low-rank + total-variation regularization recovers the true spatial and temporal components with very high accuracy and faithfully reconstructs the true calcium signal. Further, this performance is robust to the choice of initialization as initializing U as either an identity matrix or random values still faithfully recovers true spatial and temporal components. Additionally, despite the very different initializations, the relative error in the final objective value between the two final objective values (given as $|obj_1 - obj_2|/\min\{obj_1, obj_2\}$, where obj_1 and obj_2 denote the final

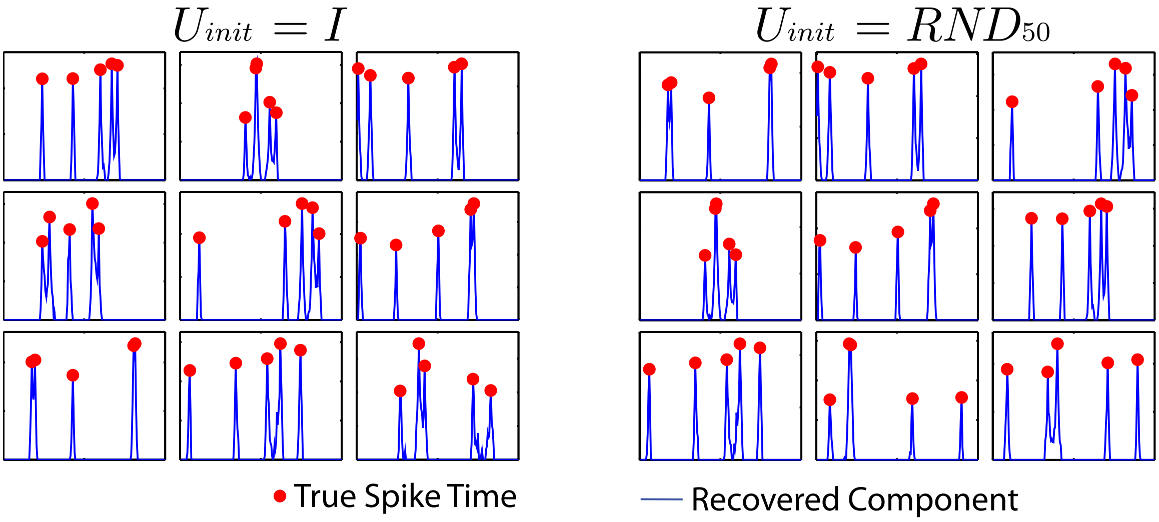

Figure 5.3: Reconstructed spike trains from phantom dataset with sparse + low-rank + total variation for the components shown in Figure 5.2. Blue lines are the estimated temporal components recovered by our algorithm, while the red dots correspond to the true temporal spike times. *Left Panel*: Reconstruction with U initialized as an identity matrix. *Right Panel*: Reconstruction with U initialized as 50 columns of random values uniformly distributed between $[0, 1]$.

objective values for the 2 different initializations) was only 3.8833×10^{-5}.

5.2.2 *In vivo* Calcium Image Data

We next tested our algorithm on actual calcium image data taken *in vivo* from the primary auditory cortex of a mouse that was transfected with the genetic calcium indicator GCaMP5 [88]. The top panel of Figure 5.5 shows 5 manually labeled regions from the dataset (top row) and the corresponding spatial features recovered by our algorithm (bottom 3 rows) under the various regularization conditions. The bottom panel of Figure 5.5 displays a frame from the dataset taken at a time point when the

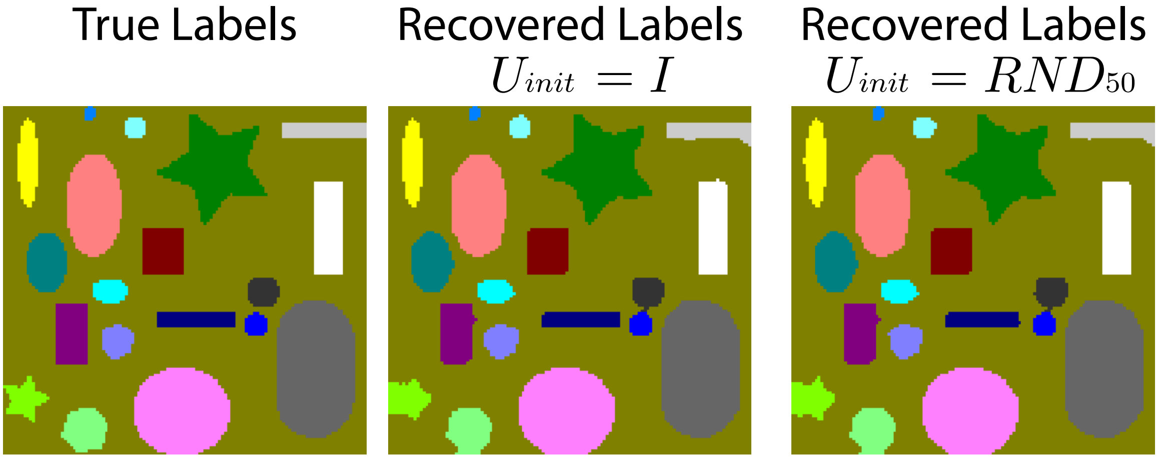
True Labels
Recovered Labels $U_{init} = I$
Recovered Labels $U_{init} = RND_{50}$

Figure 5.4: Recovered spatial segmentations from phantom dataset. *Left*: True spatial labels. *Middle*: Spatial labels recovered with sparse + low-rank + total-variation regularization, with U initialized as an identity matrix. *Right*: Same as the middle panel but with U initialized as 50 columns of random values uniformly distributed between $[0, 1]$.

corresponding region had a significant calcium signal, with the actual data shown in the top row and the corresponding reconstructed calcium signal for that time point under the various regularization conditions shown in the bottom 3 rows. We note that regions 1 and 2 correspond to the cell body and a dendritic branch of the same neuron. The manual labeling was purposefully split into two regions due to the fact that dendrites can have significantly different calcium dynamics from the cell body and thus it is often appropriate to treat calcium signals from dendrites as separate features from the cell body [89].

The data shown in Figure 5.5 are particularly challenging to segment as the two large cell bodies (regions 1 and 3) are largely overlapping in space, necessitating a spatiotemporal segmentation. In addition to the overlapping cell bodies there are

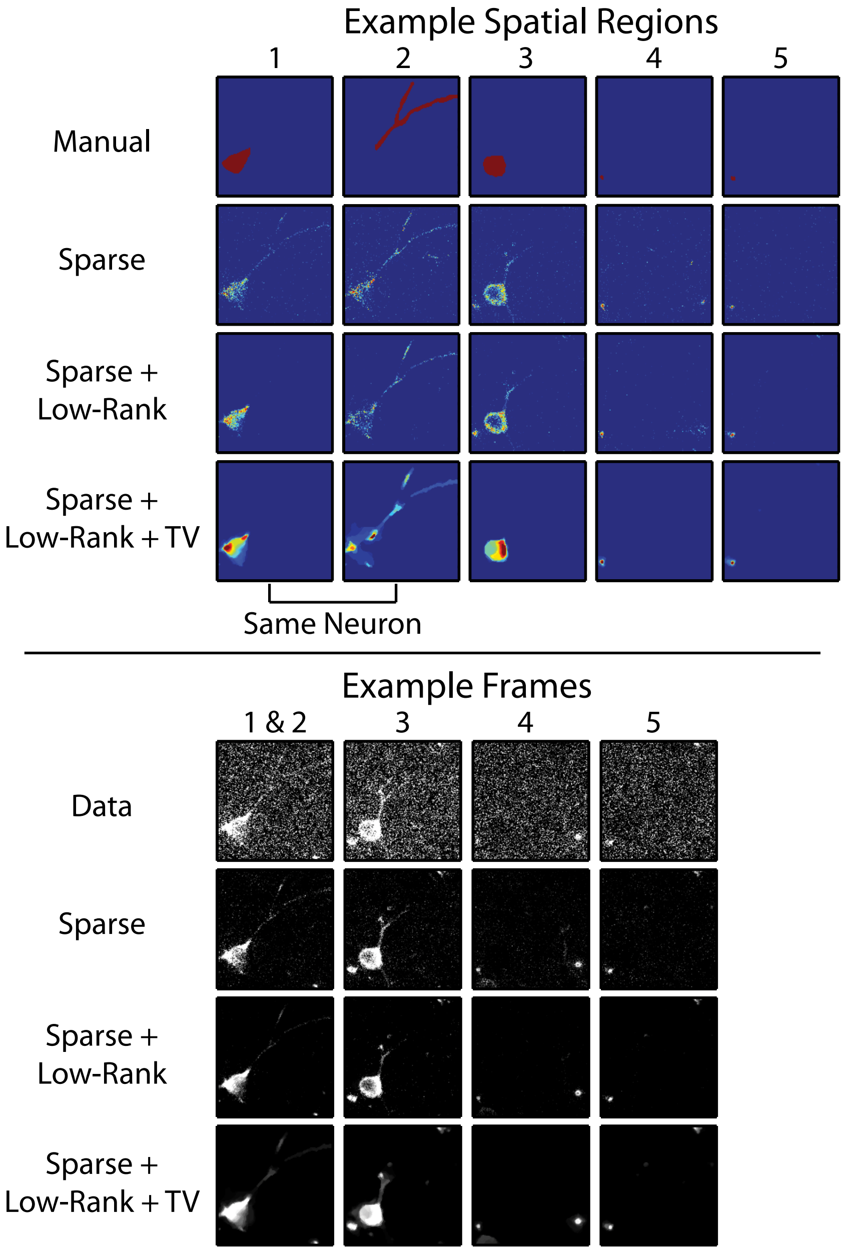

Figure 5.5: Results from an *in vivo* calcium imaging dataset. *Top*: Demonstration of spatial features for 5 example regions. (Top Row) Manually segmented regions. (Bottom 3 Rows) Corresponding spatial feature recovered by our method with various regularizations. Note that regions 1 and 2 are different parts of the same neurons - see discussion in the text. *Bottom*: Example frames from the dataset corresponding to time points where the example regions display a significant calcium signal. (Top Row) Actual Data. (Bottom 3 Rows) Estimated signal for the example frame with various regularizations.

various small dendritic processes radiating perpendicular to (regions 4 and 5) and across (region 2) the focal plane that lie in close proximity to each other and have significant calcium transients. Additionally, at one point during the dataset the animal moves, generating a large artifact in the data. Nevertheless, optimizing (5.1) under the various regularization conditions, we observe that, as expected, the spatial features recovered by sparse regularization alone are highly noisy (Fig. 5.5, row 2). Adding low-rank regularization improves the recovered spatial features, but the features are still highly pixelated and contain numerous pixels outside of the desired regions (Fig. 5.5, row 3). Finally, by incorporating the total variation regularization our method produces coherent spatial features which are highly similar to the desired manual labellings (Fig. 5.5, rows 1 and 4), noting again that these features are found directly from the alternating minimization of (5.1) without the need to solve a secondary matrix factorization. For comparison purposes, the top 5 spatial components recovered via PCA along with example image frames that are reconstructed using the top 20 principal components are shown in Figure 5.6. Note that while the PCA spatial components have a rough correspondence to the neural structures in the data a significant amount of post-processing would be required to recover the segmentation of a specific neural structure from the PCA representation. Likewise, the example image frames recovered via PCA still contain a very large amount of noise.

To initialize our structured matrix factorization algorithm for the *in vivo* dataset, U was initialized to be 100 uniformly sampled columns from an identity matrix (out

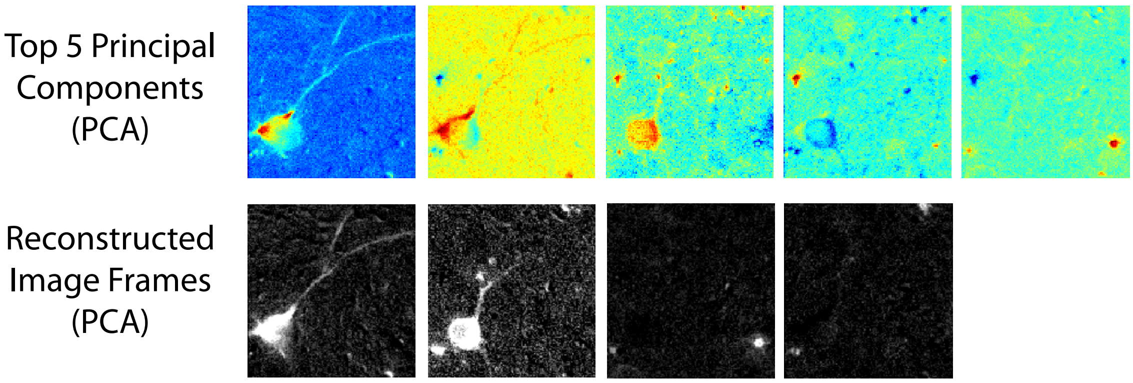
Top 5 Principal Components (PCA)
Reconstructed Image Frames (PCA)

Figure 5.6: Results of PCA applied to an *in vivo* calcium imaging dataset. *Top Row*: The first 5 most significant spatial components from PCA analysis. *Bottom Row*: Example image frames reconstructed from the first 20 most significant Principal Components. The example frames are the same is in Figure 5.5.

Table 5.1: Regularization parameters for *in vivo* calcium imaging experiments. σ denotes the standard deviation of all of the voxels in the data matrix, Y.

	λ	$[\nu_{u1}, \nu_{uTV}, \nu_{u2}]$	$[\nu_{v1}, \nu_{vTV}, \nu_{v2}]$
Sparse	2σ	$[1, 0, 0]$	$[1, 0, 0]$
Sparse + Low-Rank	1.75σ	$[1, 0, 1]$	$[1, 0, 1]$
Sparse + Low-Rank + TV	0.5σ	$[1, 0, 2.5]$	$[1, 0.5, 1]$

of a possible 559) and V was initialized as $V = 0$, demonstrating the potential to reduce the problem size and achieve good results despite a very trivial initialization. Similar to the phantom experiments, choosing U to be initialized as random variables in $[0, 1]$ produced nearly identical results (not shown). The regularization parameters were tuned manually to produce good qualitative performance for each regularization condition, and the specific values of the parameters are given in Table 5.1.

We conclude by noting that while adding total variation regularization improves performance for a segmentation task, it also can cause a dilative effect when reconstructing the estimated calcium signal (for example, distorting the size of the thin

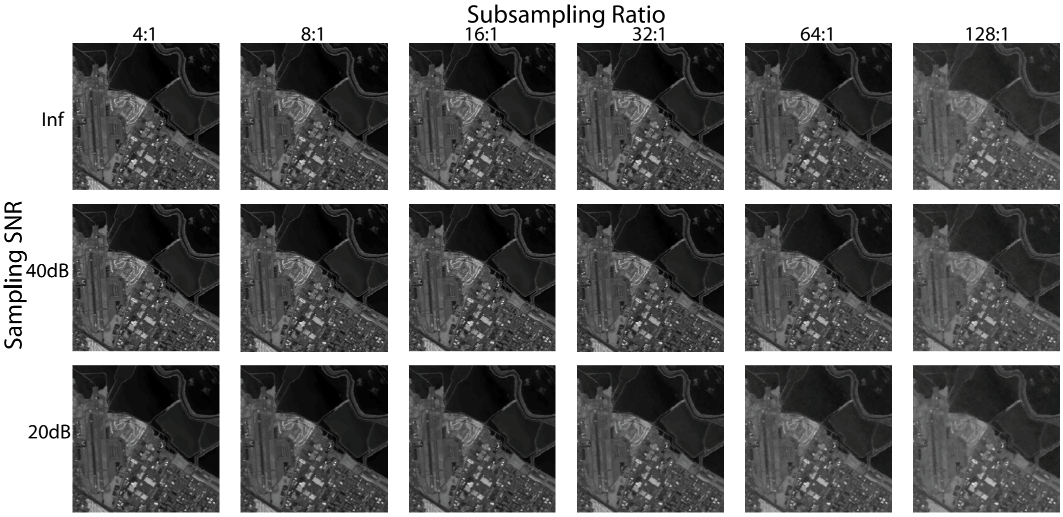

Figure 5.7: Hyperspectral compressed recovery results. Example reconstructions from a single spectral band ($i = 50$) under different subsampling ratios and sampling noise levels. Compare with [90, Fig. 2].

dendritic processes in the left two columns of the example frames in Figure 5.5). As a result, in a denoising task it might instead be desirable to only impose sparse and low-rank regularization. The fact that we can easily and efficiently adapt our model to account for many different features of the data depending on the desired task highlights the flexible nature and unifying framework of our proposed formulation (5.1).

5.3 Hyperspectral Compressed Recovery

The second application we considered is recovering a hyperspectral image volume from a set of compressed measurements. Hyperspectral imaging (HSI) is similar to regular digital photography, but instead of recording the intensities of light at just 3 wavelengths (red, green, blue) as in a typical camera, HSI records images for a very large number of wavelengths (typically hundreds or more). Due to the way the image volumes are acquired, the data often displays a low-rank structure. For example, consider hyperspectral images taken during aerial reconnaissance. If one was given the spectral signatures of various materials in the hyperspectral image volume (trees, roads, buildings, dirt, etc.), as well as the spatial distributions of those materials, then one could construct a matrix $U \in \mathbb{R}^{t\times r}$ where each column, U_i, contains the spectral signature of a material (recorded at t wavelengths) along with a matrix $V \in \mathbb{R}^{p\times r}$ which contains the spatial distribution of the i^{th} material in the column V_i (where p denotes the number of pixels in the image). Then, r corresponds to the number of materials present in the given HSI volume, and since typically $r \ll \min\{t, p\}$ the overall HSI volume can be closely approximated by the low-rank factorization $Y \approx UV^T$.

This fact, combined with the large data sizes typically encountered in HSI applications, has led to a large interest in developing compressed sampling and recovery techniques to compactly collect and reconstruct HSI datasets. Further, an HSI volume also displays significant structure in the spatial domain, as if two pixels are neighboring each other it is highly likely that they are the same material [91]. This combination of low-rank structure along with strong correlation between neighboring pixels in the spatial domain of an HSI dataset led the authors of [90] to propose a combined nuclear norm and total variation regularization (NucTV) method to reconstruct HSI volumes from compressed measurements with the form

$$\min_{X} \|X\|_* + \lambda \sum_{i=1}^{t} \|(X^i)^T\|_{TV} \quad \text{s.t.} \quad \|Y - \mathcal{A}(X)\|_F^2 \leq \epsilon. \tag{5.12}$$

Here $X \in \mathbb{R}^{t \times p}$ is the estimated HSI reconstruction with t spectral bands and p pixels, X^i denotes the i^{th} row of X (or the ith spectral band), $Y \in \mathbb{R}^{t \times m}$ contains the observed samples (compressed at a subsampling ratio of m/p), and $\mathcal{A}(\cdot)$ denotes the compressed sampling operator. To solve (5.12), [90] implemented a proximal gradient method, which required solving a total variation proximal operator for every spectral slice of the data volume in addition to solving the proximal operator of the nuclear norm (singular value thresholding) at every iteration of the algorithm [92]. For the large data volumes typically encountered in HSI, this can require significant computation per iteration.

Here we demonstrate the use of our matrix factorization method to perform hyperspectral compressed recovery by optimizing (5.1), where $\mathcal{A}(\cdot)$ is a compressive sampling function that applies a random-phase spatial convolution at each wavelength [90, 93], U contains estimated spectral features, and V contains estimated spatial abundance features.[3] Compressed recovery experiments were performed on the dataset from [90][4] at various subsampling ratios and with different levels of sam-

[3] For HSI experiments, we set $\nu_u = \nu_{v_1} = 0$ in (5.2) and (5.3).

[4] The data used are a subset of the publicly available AVARIS Moffet Field dataset. We made an effort to match the specific spatial area and spectral bands of the data for our experiments to that used in [90] but note that slightly different data may have been used in our study.

Table 5.2: Hyperspectral imaging compressed recovery error rates.

	Our Method			NucTV		
Sample	Sampling SNR (dB)			Sampling SNR (dB)		
Ratio	∞	40	20	∞	40	20
4:1	0.0209	0.0206	0.0565	0.01	0.02	0.06
8:1	0.0223	0.0226	0.0589	0.03	0.04	0.08
16:1	0.0268	0.0271	0.0663	0.09	0.09	0.13
32:1	0.0393	0.0453	0.0743	0.21	0.21	0.24
64:1	0.0657	0.0669	0.1010			
128:1	0.1140	0.1186	0.1400			

pling noise. We limited the number of columns of U and V to 15 (the dataset has 256 × 256 pixels and 180 spectral bands), initialized one randomly selected pixel per column of V to one and all others to zero, and initialized U as $U = 0$.

Figure 5.7 shows examples of the recovered images at one wavelength (spectral band $i = 50$) for various subsampling ratios and sampling noise levels and Table 5.2 shows the reconstruction recovery rates $\left\|X_{true} - UV^T\right\|_F / \left\|X_{true}\right\|_F$, where $X_t rue$ denotes the true hyperspectral image volume. We note that even though we optimized over a highly reduced set of variables ($[256 \times 256 \times 15 + 180 \times 15]/[256 \times 256 \times 180] \approx 8.4\%$) with very trivial initializations, we were able to achieve reconstruction error rates equivalent to or better than those in [90][5]. Additionally, by solving the reconstruction in a factorized form, our method offers the potential to perform blind hyperspectral unmixing directly from the compressed samples without ever needing to reconstruct the full dataset, an application extension we leave for future work.

[5] The entries for NucTV in Table 5.2 were adapted from [90, Fig. 1]

Bibliography

[1] S. Mallat, *A Wavelet Tour of Signal Processing: The Sparse Way.* Academic Press, 2008.

[2] E. R. Dougherty and R. A. Lotufo, *Hands-on Morphological Image Processing.* SPIE Press Bellingham, 2003, vol. 71.

[3] T. Ojala, M. Pietikäinen, and D. Harwood, "A comparative study of texture measures with classification based on featured distributions," *Pattern Recognition*, vol. 29, no. 1, pp. 51–59, 1996.

[4] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, 2005, pp. 886–893.

[5] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[6] C. Grienberger and A. Konnerth, “Imaging calcium in neurons,” *Neuron*, vol. 73, no. 5, pp. 862–885, 2012.

[7] J. T. Vogelstein, A. M. Packer, T. A. Machado, T. Sippy, B. Babadi, R. Yuste, and L. Paninski, “Fast nonnegative deconvolution for spike train inference from population calcium imaging,” *Journal of Neurophysiology*, vol. 104, no. 6, pp. 3691–3704, 2010.

[8] R. T. Rockafellar and R. J.-B. Wets, *Variational Analysis*, 3rd ed. Springer, 2009.

[9] S. Boyd and L. Vandenberghe, *Convex Optimization.* Cambridge University Press, 2004.

[10] S. J. Wright and J. Nocedal, *Numerical Optimization.* Springer New York, 1999, vol. 2.

[11] R. T. Rockafellar, *Convex Analysis.* Princeton University Press, 1970.

[12] N. Parikh and S. Boyd, “Proximal algorithms,” *Foundations and Trends in Optimization*, vol. 1, pp. 123–231, 2013.

[13] F. Bach, R. Jenatton, J. Mairal, and G. Obozinski, “Optimization with sparsity-inducing penalties,” *Foundations and Trends in Machine Learning*, vol. 4, no. 1, pp. 1–106, 2012.

[14] V. Jojic, S. Saria, and D. Koller, "Convex envelopes of complexity controlling penalties: the case against premature envelopment," in *International Conference on Artificial Intelligence and Statistics*, 2011, pp. 399–406.

[15] D. L. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[16] M. Fazel, H. Hindi, and S. Boyd, "Rank minimization and applications in system theory," in *American Control Conference*, vol. 4, 2004, pp. 3273–3278.

[17] B. Recht, M. Fazel, and P. Parrilo, "Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization," *SIAM Review*, vol. 52, no. 3, pp. 471–501, 2010.

[18] C. Eckart and G. Young, "The approximation of one matrix by another of lower rank," *Psychometrika*, vol. 1, no. 3, pp. 211–218, 1936.

[19] I. Jolliffe, *Principal Component Analysis.* Wiley Online Library, 2002.

[20] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Neural Information Processing Systems*, 2001, pp. 556–562.

[21] ——, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.

[22] M. W. Berry, M. Browne, A. N. Langville, V. P. Pauca, and R. J. Plemmons,

"Algorithms and applications for approximate nonnegative matrix factorization," *Computational Statistics and Data Analysis*, vol. 52, no. 1, pp. 155–173, 2007.

[23] B. A. Olshausen and B. J. Field, "Sparse coding with an overcomplete basis set: a strategy employed by V1?" *Vision Research*, 1997.

[24] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Transactions on Image Processing*, vol. 15, no. 12, pp. 3736–3745, 2006.

[25] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online dictionary learning for sparse coding," in *International Conference on Machine Learning*, 2009.

[26] ——, "Online learning for matrix factorization and sparse coding," *The Journal of Machine Learning Research*, vol. 11, pp. 19–60, 2010.

[27] M. Aharon, M. Elad, and A. M. Bruckstein, "K-SVD: an algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4311–4322, 2006.

[28] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Review*, vol. 51, no. 3, pp. 455–500, 2009.

[29] B. W. Bader, M. W. Berry, and M. Browne, "Discussion tracking in enron email using parafac," in *Survey of Text Mining II*. Springer, 2008, pp. 147–163.

[30] E. Martınez-Montes, P. A. Valdés-Sosa, F. Miwakeichi, R. I. Goldman, and M. S. Cohen, “Concurrent eeg/fmri analysis by multiway partial least squares,” *NeuroImage*, vol. 22, no. 3, pp. 1023–1034, 2004.

[31] F. Miwakeichi, E. Martınez-Montes, P. A. Valdés-Sosa, N. Nishiyama, H. Mizuhara, and Y. Yamaguchi, “Decomposing eeg data into space–time–frequency components using parallel factor analysis,” *NeuroImage*, vol. 22, no. 3, pp. 1035–1045, 2004.

[32] A. Shashua and A. Levin, “Linear image coding for regression and classification using the tensor-rank principle,” in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, 2001, pp. I–42.

[33] C. J. Hillar and L.-H. Lim, “Most tensor problems are np-hard,” *Journal of the ACM*, vol. 60, no. 6, p. 45, 2013.

[34] L. De Lathauwer and J. Vandewalle, “Dimensionality reduction in higher-order signal processing and rank-(r1, r2, ... , rn) reduction in multilinear algebra,” *Linear Algebra and its Applications*, vol. 391, pp. 31–55, 2004.

[35] M. A. O. Vasilescu and D. Terzopoulos, “Multilinear analysis of image ensembles: Tensorfaces,” in *European Conference on Computer Vision*. Springer, 2002, pp. 447–460.

[36] M. A. O. Vasilescu, “Human motion signatures: Analysis, synthesis, recogni-

tion," in *International Conference on Pattern Recognition*, vol. 3, 2002, pp. 456–460.

[37] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of Control, Signals and Systems*, vol. 2, no. 4, pp. 303–314, 1989.

[38] F. Rosenblatt, "The perceptron: a probabilistic model for information storage and organization in the brain." *Psychological Review*, vol. 65, no. 6, p. 386, 1958.

[39] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Cognitive Modeling*, vol. 5, 1988.

[40] V. Vapnik, *The Nature of Statistical Learning Theory.* N.Y.: Springer, 1995.

[41] Y. Freund, R. Schapire, and N. Abe, "A short introduction to boosting," *Journal of the Japanese Society For Artificial Intelligence*, vol. 14, no. 771-780, p. 1612, 1999.

[42] L. Rokach and O. Maimon, *Data Mining with Decision Trees: Theroy and Applications.* River Edge, NJ, USA: World Scientific Publishing Co., Inc., 2008.

[43] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, 2015.

[44] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Neural Information Processing Systems*, 2012, pp. 1097–1105.

[45] G. E. Dahl, T. N. Sainath, and G. E. Hinton, “Improving deep neural networks for lvcsr using rectified linear units and dropout,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013, pp. 8609–8613.

[46] A. L. Maas, A. Y. Hannun, and A. Y. Ng, “Rectifier nonlinearities improve neural network acoustic models,” in *International Conference on Machine Learning*, vol. 30, 2013.

[47] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting,” *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[48] M. D. Zeiler, M. Ranzato, R. Monga, M. Mao, K. Yang, Q. V. Le, P. Nguyen, A. Senior, V. Vanhoucke, J. Dean, and G. E. Hinton, “On rectified linear units for speech processing,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013, pp. 3517–3521.

[49] Z. Jiang, Z. Lin, and L. S. Davis, “Learning a discriminative dictionary for sparse coding via label consistent k-svd,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 1697–1704.

[50] J. Mairal, J. Ponce, G. Sapiro, A. Zisserman, and F. R. Bach, “Supervised dictionary learning,” in *Neural Information Processing Systems*, 2009, pp. 1033–1040.

[51] N. Srebro, J. Rennie, and T. S. Jaakkola, “Maximum-margin matrix factorization,” in *Neural Information Processing Systems*, 2004, pp. 1329–1336.

[52] J.-F. Cai, E. J. Candés, and Z. Shen, “A singular value thresholding algorithm for matrix completion,” *SIAM Journal of Optimization*, vol. 20, no. 4, pp. 1956–1982, 2008.

[53] R. A. Ryan, *Introduction to Tensor Products of Banach Spaces.* Springer, 2002.

[54] F. Bach, J. Mairal, and J. Ponce, “Convex sparse matrix factorizations,” *arXiv:0812.1869v1*, 2008.

[55] F. Bach, “Convex relaxations of structured matrix factorizations,” *arXiv:1309.3117v1*, 2013.

[56] B. Haeffele, E. Young, and R. Vidal, “Structured low-rank matrix factorization: Optimality, algorithm, and applications to image processing,” in *International Conference on Machine Learning*, 2014.

[57] V. Chandrasekaran, B. Recht, P. A. Parrilo, and A. S. Willsky, “The convex geometry of linear inverse problems,” *Foundations of Computational Mathematics*, vol. 12, no. 6, pp. 805–849, 2012.

[58] S. Burer and R. D. C. Monteiro, “Local minima and convergence in low-rank semidefinite programming,” *Mathematical Programming, Series A*, no. 103, pp. 427–444, 2005.

[59] Y. Yu, X. Zhang, and D. Schuurmans, "Generalized conditional gradient for sparse estimation," *arXiv preprint arXiv:1410.4828*, 2014.

[60] R. Cabral, F. De la Torre, J. P. Costeira, and A. Bernardino, "Unifying nuclear norm and bilinear factorization approaches for low-rank matrix decomposition," in *IEEE International Conference on Computer Vision*, 2013, pp. 2488–2495.

[61] H. Zou, T. Hastie, and R. Tibshirani, "Sparse principal component analysis," *Journal of Computational and Graphical Statistics*, vol. 15, no. 2, pp. 265–286, 2006.

[62] E. Richard, G. R. Obozinski, and J.-P. Vert, "Tight convex relaxations for sparse matrix factorization," in *Neural Information Processing Systems*, 2014, pp. 3284–3292.

[63] E. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *Journal of the ACM*, vol. 58, no. 3, 2011.

[64] J. M. Hendrickx and A. Olshevsky, "Matrix p-norms are np-hard to approximate if p\neq1,2,," *SIAM Journal on Matrix Analysis and Applications*, vol. 31, no. 5, pp. 2802–2812, 2010.

[65] A. d'Aspremont, L. El Ghaoui, M. I. Jordan, and G. R. Lanckriet, "A direct formulation for sparse pca using semidefinite programming," *SIAM Review*, vol. 49, no. 3, pp. 434–448, 2007.

[66] Y. Xu and W. Yin, “A block coordinate descent method for regularized multi-convex optimization with applications to nonnegative tensor factorization and completion,” *SIAM Journal of Imaging Sciences*, vol. 6, no. 3, pp. 1758–1789, 2013.

[67] A. Beck and M. Teboulle, “A fast iterative shrinkage-thresholding algorithm for linear inverse problems,” *SIAM Journal on Imaging Sciences*, vol. 2, pp. 183–202, 2009.

[68] E. J. Candès, J. Romberg, and T. Tao, “Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information,” *IEEE Transactions on Information Theory*, vol. 52, no. 2, pp. 489–509, 2006.

[69] E. Candes and J. Romberg, “Sparsity and incoherence in compressive sampling,” *Inverse Problems*, vol. 23, no. 3, p. 969, 2007.

[70] E. Candès and B. Recht, “Exact matrix completion via convex optimization,” *Foundations of Computational Mathematics*, vol. 9, pp. 717–772, 2009.

[71] A. Cichocki, R. Zdunek, A. H. Phan, and S.-i. Amari, *Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-Way Data Analysis and Blind Source Separation.* John Wiley & Sons, 2009.

[72] J. Ngiam, A. Coates, A. Lahiri, B. Prochnow, Q. V. Le, and A. Y. Ng, “On op-

timization methods for deep learning," in *International Conference on Machine Learning*, 2011, pp. 265–272.

[73] Y. Xu and W. Yin, "A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion," *SIAM Journal on Imaging Sciences*, vol. 6, no. 3, pp. 1758–1789, 2013.

[74] Y. Bengio, N. L. Roux, P. Vincent, O. Delalleau, and P. Marcotte, "Convex neural networks," in *Neural Information Processing Systems*, 2005, pp. 123–130.

[75] J. H. Friedman, "Greedy function approximation: a gradient boosting machine," *Annals of Statistics*, pp. 1189–1232, 2001.

[76] L. Mason, J. Baxter, P. L. Bartlett, and M. R. Frean, "Boosting algorithms as gradient descent," in *Neural Information Processing Systems*, 2000, pp. 512–518.

[77] Y. N. Dauphin, R. Pascanu, C. Gulcehre, K. Cho, S. Ganguli, and Y. Bengio, "Identifying and attacking the saddle point problem in high-dimensional non-convex optimization," in *Neural Information Processing Systems*, 2014, pp. 2933–2941.

[78] A. Choromanska, M. Henaff, M. Mathieu, G. Ben Arous, and Y. LeCun, "{The Loss Surfaces of Multilayer Networks}," in *International Conference on Artificial Intelligence and Statistics*, 2015, pp. 192–204.

[79] X. Zhang, Y.-L. Yu, and D. Schuurmans, "Polar operators for structured sparse estimation," in *Neural Information Processing Systems*, 2013, pp. 82–90.

[80] R. T. Rockafellar and R. J.-B. Wets, *Variational Analysis.* Springer, 1998, vol. 317.

[81] F. Bach, "Breaking the curse of dimensionality with convex neural networks," *arXiv preprint arXiv:1412.8690*, 2014.

[82] B. Neyshabur, R. Tomioka, and N. Srebro, "In search of the real inductive bias: On the role of implicit regularization in deep learning," *arXiv preprint arXiv:1412.6614*, 2014.

[83] L. Wan, M. Zeiler, S. Zhang, Y. L. Cun, and R. Fergus, "Regularization of neural networks using dropconnect," in *International Conference on Machine Learning*, 2013, pp. 1058–1066.

[84] H. Birkholz, "A unifying approach to isotropic and anisotropic total variation denoising models," *Journal of Computational and Applied Mathematics*, vol. 235, pp. 2502–2514, 2011.

[85] R. J. Tibshirani and J. Taylor, "The solution path of the generalized lasso," *Annals of Statistics*, vol. 39, no. 3, pp. 1335–1371, 2011.

[86] C. Stosiek, O. Garaschuk, K. Holthoff, and A. Konnerth, "In vivo two-photon

calcium imaging of neuronal networks," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 100, no. 12, pp. 7319–7324, 2003.

[87] E. A. Pnevmatikakis, T. A. Machado, L. Grosenick, B. Poole, J. T. Vogelstein, and L. Paninski, "Rank-penalized nonnegative spatiotemporal deconvolution and demixing of calcium imaging data," Abstract: Computational and Systems Neuroscience (Cosyne), 2013.

[88] J. Akerboom, T.-W. Chen, T. J. Wardill, L. Tian, J. S. Marvin, S. Mutlu, $\cdots$, and L. L. Looger, "Optimization of a GCaMP calcium indicator for neural activity imaging," *The Journal of Neuroscience*, vol. 32, pp. 13 819–13 840, 2012.

[89] N. Spruston, "Pyramidal neurons: Dendritic structure and synaptic integration," *Nature Reviews Neuroscience*, vol. 9, pp. 206–221, 2008.

[90] M. Golbabaee and P. Vandergheynst, "Joint trace/tv minimization: A new efficient approach for spectral compressive imaging," in *IEEE Internation Conference on Image Processing*, 2012, pp. 933–936.

[91] H. Zhang, W. He, L. Zhang, H. Shen, and Q. Yuan, "Hyperspectral image restoration using low-rank matrix recovery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. PP, pp. 1–15, 2013.

[92] P. L. Combettes and J. C. Pesquet, "Proximal splitting methods in signal pro-

cessing," in *Fixed-Point Algorithms for Inverse Problems in Science and Engineering.* Springer-Verlag, 2011, vol. 49, pp. 185–212.

[93] J. Romberg, "Compressive sensing by random convolution," *SIAM Journal of Imaging Sciences*, vol. 2, no. 4, pp. 1098–1128, Nov. 2009.

Vita

Ben Haeffele received a B.S. in Electrical Engineering from the Georgia Institute of Technology in 2006 and enrolled in the Biomedical Engineering Ph.D. program at Johns Hopkins University in 2007. His research has focused on neuroscience, microscopy, machine learning, optimization, and computer vision, with a particular emphasis on analyzing the mathematical properties of challenging, non-convex optimization problems that arise in the field of representation learning. Starting in November of 2015, he will be an associate research scientist in the Center for Imaging Science at Johns Hopkins University.