Learning Representations With Linear-Algebraic Structure

by

Abraham Frandsen

Department of Computer Science
Duke University

Date:_____

Approved:

_____
Rong Ge, Supervisor

_____
Debmalya Panigrahi

_____
Ronald Parr

_____
Cynthia Rudin

Dissertation submitted in partial fulfillment of the
requirements for the degree of Doctor of Philosophy
in the Department of Computer Science
in the Graduate School of
Duke University

2022

ABSTRACT

Learning Representations With Linear-Algebraic Structure

by

Abraham Frandsen

Department of Computer Science
Duke University

Date:_____

Approved:

_____
Rong Ge, Supervisor

_____
Debmalya Panigrahi

_____
Ronald Parr

_____
Cynthia Rudin

An abstract of a dissertation submitted in partial fulfillment of the
requirements for the degree of Doctor of Philosophy
in the Department of Computer Science
in the Graduate School of
Duke University

2022

Abstract

Representation learning is a key step for enabling algorithms to make sense of data and output good decisions. Good data representations preserve useful information, discard irrelevant features, and simplify complex relationships between data. We approach the problem of representation learning through the lens of latent variable models. In such models, the representations are directly given as unobserved variables encoding the core structure of the data. We utilize linear algebraic structure to specify the properties of the representations, which enables rigorous analysis and efficient, provable algorithms.

We first consider the area of natural language processing, where the data are comprised of words. We propose a novel model for word representations that encodes compositional syntactic and semantic structure as latent multilinear structure. We prove that the representations can be efficiently recovered and develop a practical learning algorithm.

We show that learning the word embedding model is closely connected to the Tucker decomposition, an important basic operation in tensor analysis that also arises in the context of other latent variable models. We formulate the Tucker decomposition as a nonconvex optimization problem and prove that its landscape is benign. We then give a local search algorithm that provably finds the global optimum.

We finally consider the area of reinforcement learning and control, where the time dynamics of the data are vital. We propose a model in which the state observations are high-dimensional with nonlinear dynamics, but depend on a latent low-dimensional linear control system. We develop state representation learning algorithms based on both forward and inverse dynamics that provably and efficiently recover the hidden linear system.

Contents

List of Tables

List of Figures

Acknowledgements

Many people helped me through my years of doctoral education and research that culminate in this thesis. Foremost is my advisor, Rong Ge, whose patient mentorship pushed me to become a more capable researcher, and with whom it was a privilege to work. I have learned so much through coursework and conversations with the fantastic faculty in the departments of Computer Science and Mathematics at Duke. I have benefitted greatly from the supportive and collegial culture of the computer science graduate program, which was created by wonderful fellow graduate students and excellent staff. I will miss it (not to mention all the free food). My partner Ryan has given amazing support and helped me maintain a healthy life balance. My parents and siblings have been a constant source of encouragement.

I owe you all my gratitude.

Chapter 1

Introduction

Machine learning has become a core discipline and technology across many areas of society. Wherever data and decision-making come together there are or soon will be people attempting to utilize its methods. Increases in data and computational power combined with algorithmic advances have fueled this tremendous growth over the last several years, and it seems that this is only accelerating.

The success of machine learning rests on the ability of algorithms to make sense of data. As humans, we are constantly processing data from the world around us. Our ability to effectively use and understand these data depends in large measure on how they are presented to us: our computers have monitors and speakers to transform digital data into electromagnetic and acoustic waves, text must be translated if we can't read its original language, measurements from scientific experiments are often denoised before scientists can draw conclusions. The same can broadly be said for algorithms.

A machine learning model is commonly formulated as a parametric function $f_\theta : \mathbb{R}^d \to \mathbb{R}^k$ that takes in a feature vector and outputs a prediction or decision of some kind. Here, $\theta \in \Theta \subset \mathbb{R}^n$ is a parameter vector usually selected through solving an optimization problem. This idealized view glosses over the significant complexities that can arise when dealing with the input data. There are many domains where the data we collect aren't vectorial by nature, such as human language and social networks. In other cases, the observed data are high-dimensional, noisy, and contain irrelevant signals, such as video and images. Finally, the chosen machine learning model may

be ill-suited to leverage the structure in the data. In such cases, it is necessary to transform the raw data to produce feature vectors, or *representations*, that are suited to standard machine learning models. In particular, if the raw data come from a domain $\mathcal{X}$, we seek a mapping $\phi : \mathcal{X} \to \mathbb{R}^d$ that produces a suitable representation of the data for f_θ. Representation learning has emerged as an important field that proposes algorithms for learning such data transformations.

Objectives and Overview In this thesis, we study the representation learning problem with an emphasis on developing efficient algorithms with provable guarantees. We pursue this course and make contributions along the following objectives:

1. Propose data models that exhibit rich latent linear-algebraic structure.
2. Develop representation learning algorithms for these models based on nonconvex optimization.
3. Analyze and provide theoretical guarantees for these algorithms, and demonstrate their empirical effectiveness.

To do this, we consider representation learning for two domains: natural language processing and control. In each case, we propose natural latent variable models wherein ground-truth linear-algebraic structure can be recovered under the correct–but initially unknown–data transformation. We then take up the central theoretical problem: to design efficient representation learning algorithms that provably identify these transformations. In addition to theoretical considerations, we also extend our algorithms to practical settings using modern deep learning techniques. By doing so, we expand the foundation on which representation learning has provable guarantees and provide principled motivation for algorithms used in practice.

In the remainder of this chapter, we discuss existing representation learning techniques and motivate our approach. We also review particular lines of research upon which we build.

In Chapter 2, we establish common mathematical notation and definitions used throughout the thesis, as well as highlight important basic facts that will be frequently used.

In Chapter 3, we focus on the domain of natural language processing. We propose a model for word embeddings that encodes syntactical relationships between words. We show that the latent representations in this model can be recovered through a tensor decomposition problem. We then design a practical learning algorithm for the model and discuss experimental results.

In Chapter 4, we study the tensor decomposition problem that emerged from Chapter 3. We design a nonconvex objective function that encodes this decomposition and analyze the resulting optimization landscape. We give an efficient local search algorithm that provably solves the decomposition problem.

In Chapter 5, we consider the problem of state representation learning for control. We propose a latent variable model with rich, nonlinear observations and a latent linear control system driving the dynamics. We then give two provably correct representation learning algorithms for identifying the latent linear structure, and conduct empirical experiments with both synthetic and simulation data.

The results presented in this thesis reflect the effort and original work of the author, conducted under the supervision of and in collaboration with Rong Ge. The content of Chapter 5 was additionally developed in collaboration with Holden Lee.

1.1 Overview of Representation Learning

The need for proper data representation has long been recognized as fundamental part of data analysis. Designing hand-crafted features, such as cepstral coefficients for speech recognition (Kim und Stern, 2016), wavelet transforms for signal processing (Pittner und Kamarthi, 1999), and SIFT (Lowe, 2004) for computer vision tasks, is an important traditional approach. However, human-engineered features are often tailored to very specific tasks and may not fully capture the requisite structure in the data. It can be difficult or impossible to directly specify effective representations by hand. Because of this, it has become common to approach data representation itself as a learning task. Supervised deep learning models couple the representation and prediction problems by training deep models end-to-end, from raw data to output. The intermediate layers of such models can be viewed as data representations optimized for the particular predictive task at hand. While the performance of these models can outstrip approaches based on hand-crafted features in many domains, they are still limited both by the availability of labeled training data as well as the flexibility and transferability of the learned representations to other tasks.

Unsupervised Representation Learning In contrast to hand-crafted and end-to-end approaches, unsupervised representation learning focuses on data transformation as a learning problem in its own right, separate from any particular predictive task. This approach uses large unlabeled datasets to train general-purpose representations that aren't tied to a single use. It sets itself apart from other paradigms in unsupervised learning, such as clustering, through its anticipation of future tasks: rather than being an end in itself, the utility of representation learning ultimately depends on how well the learned representations facilitate subsequent tasks. To illustrate this, consider a dataset $X = \{x_1, \ldots, x_N\} \subset \mathcal{X}$. Clustering methods seek to partition X into k groups

that capture meaningful shared structure. While a learned clustering can highlight important properties of the data, cluster membership on its own is insufficient for many predictive tasks. Representation learning, on the other hand, seeks a mapping $\phi : \mathcal{X} \to \mathbb{R}^d$ that captures and elucidates relevant patterns and structure in X. For any subsequent machine learning task involving data from the same domain $\mathcal{X}$, the representation map ϕ is used to produce feature vectors which stand in place of the raw data.

There are multiple ways to formalize the intuition that learned representations should capture the structure of the data. In many settings, the data don't have known ground-truth structure, and so general principles can guide representation learning. In their influential review of the field from a few years ago, Bengio u. a. (2013) argue that good representations should be distributed and expressive (i.e. dense or k-sparse vectors as opposed to 1-sparse vectors), exhibit invariance to local changes in the raw data, capture abstract concepts, and disentangle factors of variation (i.e. different coordinates in the representation correspond to different, independent explanatory factors). Of course when there is known, relevant latent structure in the data, it is natural to require the representations to preserve and identify this structure. For example, Schölkopf u. a. (2021) advocate for representations to capture causal relationships rather than just statistical properties.

A few broad algorithmic frameworks for unsupervised representation learning have become popular in recent years. Autoencoder methods (Tschannen u. a., 2018) take the approach of learning both a representation function ϕ (the "encoder") and a "decoder" function ψ such that the difference between the original input datapoint x and its reconstruction $\psi(\phi(x))$ is minimized. This ensures that the learned representation retains information about the original data, since it must be effective at reproducing them. Self-supervised representation learning (Ericsson u. a., 2021) is another class

of algorithms that has become a very popular. It formulates representation learning as a predictive task, but relies on labels that can be quickly generated from the data itself without human annotation. Specifying relevant artificial predictive tasks is a potentially important way to inject domain-knowledge into the learning algorithm and representations, somewhat analogous to the role of human feature engineering in older methods. Contrastive representation learning (Le-Khac u. a., 2020) is a related technique suitable for domains in which there is a natural notion or similarity or proximity between datapoints, as is the case with e.g. graph, text, and image data. Here, given a datapoint x, a similar point y, and a dissimilar point z, the representation map is trained so that $\phi(x)$ can discriminate between $\phi(y)$ and $\phi(z)$. This ensures that the similarity structure of the raw data is preserved and reflected in the geometry of the representations.

Latent Variable Models While the above approaches can be very effective general tools for many different problems and applications, it can be difficult to prove strong guarantees for these algorithms without understanding the particular structure of the input data. Accordingly, in this thesis we study unsupervised representation learning in the context of latent variable models, which allows us to directly specify and leverage structure within the data. Latent variable models relate observed data to unseen explanatory variables. In such models, the latent variables are often in correspondence with the observed datapoints and transparently encode the key structure that is obscured by the full observation. In this case, we can view these variables themselves as the latent representations. The representation learning task is then to infer these latent variables and produce a mapping from observation to latent variable.

A classic example of this is probabilistic latent semantic analysis (PLSA) (Hofmann, 2013), which models text documents as mixtures of latent topics, each topic being a

distribution over the vocabulary. Learning this model involves infering the topics as well as the mixture weights of each document over the topics. These mixture weights act as representations for the original documents, facilitating document classification, information retrieval, and other tasks.

Latent variable modeling offers advantages both in theory and practice. The concrete mathematical setting of latent variable models enables rigorous analysis and the design of algorithms with provable guarantees. Moreover, the representation learning problem has a direct target of identifying the latent variables and structure, which gives a prinicipled basis on which we can evaluate the quality of the learned representations. Additionally, the latent variables often encode *interpretable* structure related to the observed data, and this interpretability transfers naturally to the representations. For example, in PLSA, the learned topics can be interpreted by observing which words are most prevalent in each topic, and the mixture weights representing each document indicate which topics are most represented.

We focus in particular on latent variable models that encode *linear-algebraic structure*. Identifying and exploiting linear structure in data has a long and fruitful history in machine learning and beyond. Perhaps the oldest and most foundational representation learning algorithm is principal components analysis (PCA), which identifies the linear subspace that captures the most variation in the input data, and projects the data to this subspace. Although simple, PCA provides an important starting point for many other ideas in the field such as disentangled features, dimensionality reduction, and autoencoders. The latent structure in PLSA and related topic models is known to be characterized by the low-rank structure particular statistic matrices in the model (Ge, 2013). Targeting linear-algebraic structure for representation learning allows us to use and build upon the deep well of mathematical theory and efficient, robust algorithms tailored to linear algebra.

1.2 Background and Related Work

We now review specific lines of research that connect to this thesis.

Matrix and Tensor Decomposition Identifying the linear-algebraic structure in latent variable models can often be reduced to matrix and tensor decomposition problems. The singular value decomposition plays a particularly central role, forming the basis of many representation learning algorithms including PCA. More recent problems that have risen to prominence include nonnegative matrix factorization (Lee und Seung, 2000; Wang und Zhang, 2012; Ding u. a., 2006) and low-rank matrix recovery and completion (Zhang u. a., 2013; Davenport und Romberg, 2016; Nguyen u. a., 2019).

The two primary tensor decompositions are the CP/PARAFAC (Carroll und Chang, 1970; Harshman u. a., 1970) decomposition and the Tucker decomposition (Hitchcock, 1927; Tucker, 1966). Unlike the CP decomposition, the Tucker decomposition can be computed efficiently if the original tensor has low multilinear rank through, for example, the higher-order SVD (De Lathauwer u. a., 2000a). Many other algorithms have also been proposed for tensor decomposition, see for example De Lathauwer u. a. (2000b); Eldén und Savas (2009); Phan u. a. (2014).

Spectral Method Several latent variable models have been found to be efficiently learnable using spectral methods for matrix and tensor decomposition. These include hidden Markov models (Hsu u. a., 2012; Anandkumar u. a., 2012b), mixture of Guassians (Hsu und Kakade, 2013), topic models (Anandkumar u. a., 2012a), and mixed membership community models (Anandkumar u. a., 2013). In particular, see Anandkumar u. a. (2014) for additional references as well as a unified view of many of these models. The basic approach in this line of work is to relate the model parameters

to orthogonal decompositions of certain matrices or tensors that can be estimated from observed data.

Local Search and Nonconvex Optimization The representation learning algorithms we study in this thesis, like most machine learning algorithms in general, are usually based on nonconvex optimization problems. Although in the worst case, optimizing nonconvex objective functions is NP-hard, there are many natural settings in which global optimization is tractable. A recent line of work (Ge u. a., 2015; Bhojanapalli u. a., 2016; Sun u. a., 2016a; Ge u. a., 2016; Sun u. a., 2016b; Bandeira u. a., 2016) showed that many nonconvex problems can still be solved by local search algorithms because they have a simple *optimization landscape*.

In the particular case of matrix and tensor decompositions, it has become common to apply local search algorithms directly to nonconvex objective functions (Koren, 2009; Recht und Ré, 2013). Despite the nonconvexity, for matrix problems such as matrix sensing (Bhojanapalli u. a., 2016; Park u. a., 2016) and matrix completion (Ge u. a., 2016, 2017), it was shown that all local minima are globally optimal. Similar results have also been shown for special cases of tensor CP decomposition(Ge u. a., 2015).

Chapter 2

Notation and Preliminaries

In this chapter we specify some of the common notation and definitions used throughout the thesis. We also highlight some common useful facts about linear and tensor algebra.

Linear Algebra We use lower-case letters like u to denote vectors, and upper-case letters like M to denote matrices. We reserve the symbol I to denote the identity matrix; its particular dimension will be clear from context. For a vector u, we use $\|u\|$ and $\|u\|_2$ to denote its Euclidean norm. For vectors u, v we write $\langle u, v \rangle$ to denote their inner-product. For a matrix M, we use $\|M\|$ to denote its spectral norm, $\|M\|_F = \sqrt{\sum_{i,j} M_{i,j}^2}$ to denote its Frobenius norm, $M_{i,:}$ to denote its i-th row, and $M_{:,j}$ to denote its j-th column. For matrices M, A of the same dimensions, we write $\langle M, A \rangle = \mathrm{tr}(M^\top A)$ to denote the Frobenius inner product, where tr denotes the matrix trace operator. We make use of the fact that the trace is invariant to transpose and cyclical permutations, i.e. $\mathrm{tr}(M^\top A) = \mathrm{tr}(A^\top M) = \mathrm{tr}(AM^\top)$. The matrix M^+ denotes the Moore-Penrose pseudoinverse of M.

We use caligraphic font like $\mathcal{V}$ to denote linear subspaces of $\mathbb{R}^n$. Given a subspace $\mathcal{V}$, we write $\mathcal{V}^\perp$ to denote its orthogonal complement and $\Pi_\mathcal{V}$ to denote the orthogonal projection matrix onto $\mathcal{V}$. For a matrix A, we let $\mathrm{col}(A)$ denote its column-space.

Tensor Basics In this thesis we deal with third-order tensors, which we denote with upper-case letters like T. We view a tensor T as both a three-dimensional array of real numbers, i.e. an element of $\mathbb{R}^{d_1 \times d_2 \times d_3}$, and also as a multilinear form. We use $\otimes$ to denote the tensor product: if $u \in \mathbb{R}^{d_1}$, $v \in \mathbb{R}^{d_2}$, and $w \in \mathbb{R}^{d_3}$, then $T = u \otimes v \otimes w$

is a $d_1 \times d_2 \times d_3$ tensor whose entries are $T_{i,j,k} = u_i v_j w_k$.

Just as matrices are often viewed as bilinear functions, third order tensors can be interpreted as trilinear functions over three vectors. Concretely, let T be a $d_1 \times d_2 \times d_3$ tensor, and let x, y, z be real vectors of size d_1, d_2, and d_3, respectively. We define the scalar $T(x, y, z) \in \mathbb{R}$ as follows

$$T(x, y, z) = \sum_{i,j,k=1}^{d} T_{i,j,k} x_i y_j z_k.$$

This operation is linear individually in x, y, and z. Analogous to applying a matrix M to a vector v (with the result vector Mv), we can also apply a tensor T to one or two vectors, resulting in a matrix and a vector, respectively:

$$T(x, \cdot, \cdot)_{j,k} = \sum_{i=1}^{d} T_{i,j,k} x_i, \quad T(x, y, \cdot)_k = \sum_{i,j=1}^{d} T_{i,j,k} x_i y_j$$

These definitions extend in the obvious ways to $T(x, \cdot, y)$, $T(\cdot, x, y)$, $T(\cdot, x, \cdot)$, and $T(\cdot, \cdot, x)$. We will make use of the simple facts that $\langle z, T(x, y, \cdot) \rangle = T(x, y, z)$ and $[T(x, \cdot, \cdot)]^\top y = T(x, y, \cdot)$.

We equip $\mathbb{R}^{d_1 \times d_2 \times d_3}$ with the Frobenius inner product $\langle \cdot, \cdot \rangle$ and associated norm $\| \cdot \|_F$ given by

$$\langle T, U \rangle = \sum_{i,j,k=1}^{d_1, d_2, d_3} T_{ijk} U_{ijk} \qquad \|T\|_F = \sqrt{\langle T, T \rangle}$$

We also define the operator 2-norm $\| \cdot \|_2$ (i.e. the spectral norm) by

$$\|T\|_2 = \sup \{ T(u, v, w) \, : \, \|u\|_2 = \|v\|_2 = \|w\|_2 = 1 \}$$

These two norms are related as follows (Wang u. a., 2017):

$$\left(\frac{\max(d_1, d_2, d_2)}{d_1 d_2 d_3}\right)^{1/2} \|T\|_F \leq \|T\|_2 \leq \|T\|_F.$$

In the special case of $d_1 = d_2 = d_3 = d$, we have $\|T\|_F \leq d\|T\|_2$. Another important fact is that for $\sigma = \|T\|_2$, there exist unit vectors $u \in \mathbb{R}^{d_1}$, $v \in \mathbb{R}^{d_2}$, and $w \in \mathbb{R}^{d_3}$ such that the following hold (Lim, 2005):

$$T(u, v, w) = \sigma \quad T(\cdot, v, w) = \sigma u \quad X(u, \cdot, w) = \sigma v \quad X(u, v, \cdot) = \sigma w$$

We can also view a tensor $T \in \mathbb{R}^{d_1 \times d_2 \times d_3}$ as an operator on matrices. Given matrices $A \in \mathbb{R}^{d_1 \times r_1}$, $B \in \mathbb{R}^{d_2 \times r_2}$, $C \in \mathbb{R}^{d_3 \times r_3}$, we define $T(A, B, C) \in \mathbb{R}^{r_1 \times r_2 \times r_3}$ by

$$T(A, B, C)_{ijk} = \sum_{x,y,z=1}^{d_1,d_2,d_3} T_{x,y,z} A_{x,i} B_{y,j} C_{z,k}.$$

In the special case where one or more of r_1, r_2, r_3 equals 1, we view $T(A, B, C)$ appropriately as a matrix, column vector, or scalar.

We can relate these tensor-matrix operations to standard matrix multiplication via *flattening*. Let $T_{(i)} \in \mathbb{R}^{d_i \times \Pi_{j \neq i} d_j}$ denote the factor-i flattening of X (for $i = 1, 2, 3$), defined by $(T_{(1)})_{i,(j,k)} = (T_{(2)})_{j,(i,k)} = (T_{(3)})_{k,(i,j)} = T_{i,j,k}$. Here we use multi-index notation where $(j, k) = (j-1)d_2 + k$, $(i, k) = (i-1)d_1 + k$, and $(i, j) = (i-1)d_1 + j$. Now we can state the desired relationship:

$$T(A, B, C)_{(1)} = A^\top X_{(1)}(B \otimes C)$$

where $\otimes$ denotes the Kronecker product of matrices.

Tensor Decompositions Low-rank tensor decompositions are an important topic in this thesis. Unlike matrices, there are several different definitions for the *rank* of a tensor. In this paper we mostly use the notions of multilinear rank and the related *Tucker rank* (Tucker (1966)). A tensor $T \in \mathbb{R}^{d_1 \times d_2 \times d_3}$ has multilinear rank (r_1, r_2, r_3), if there exists a core tensor $S \in \mathbb{R}^{r_1 \times r_2 \times r_3}$ and matrices $A \in \mathbb{R}^{r_1 \times d_1}$, $B \in \mathbb{R}^{r_2 \times d_2}$, and $C \in \mathbb{R}^{r_3 \times d_3}$ of minimal dimension such that $T = S(A, B, C)$. When $r_1 = r_2 = r_3 = r$, we simply say that T has Tucker rank r.

The tuple (S, A, B, C) gives a *Tucker decomposition* of X. This decomposition can be computed efficiently, for example using the higher-order singular value decomposition, which is based on computing the matrix SVD of each flattening of T. We note that T has multilinear rank (r_1, r_2, r_3) if and only if $T_{(i)}$ has rank r_i for $i = 1, 2, 3$.

When the core tensor S is restricted to a diagonal tensor (only nonzero at entries $S_{i,i,i}$), the decomposition $T = S(A, B, C)$ is called a CP decomposition (Carroll und Chang (1970); Harshman u. a. (1970)) which can also be written as $T = \sum_{i=1}^{d} s_i A_{i,:} \otimes B_{i,:} \otimes C_{i,:}$. In this case, the tensor T is the sum of d rank-1 tensors $(A_{i,:} \otimes B_{i,:} \otimes C_{i,:})$. However, unlike matrix factorizations and the Tucker decomposition, the CP decomposition of a tensor is hard to compute in the general case (Håstad (1990); Hillar und Lim (2013)).

Probability We denote the expectation operator on random variables and vectors by $\mathbb{E}$. For centered random vectors a and b, let Σ_{ab} denote the matrix $\mathbb{E}[ab^\top]$ and Σ_a denote $\mathbb{E}[aa^\top]$. Likewise, if $A \in \mathbb{R}^{m \times k}$ and $B \in \mathbb{R}^{n \times k}$ are random matrices whose respective columns are centered, i.i.d. random vectors, let Σ_{AB} denote the empirical cross-covariance matrix $k^{-1}AB^\top$ and let Σ_A denote the empirical covariance matrix $k^{-1}AA^\top$.

One important way to measure the relationship between two random vectors is

canonical correlation analysis. This seeks to find the directions of maximal linear correlation between the vectors. We say $\rho(a, b)$ is the top canonical correlation between a and b, defined by

$$\rho(a, b) = \max_{a', b'} \frac{\mathbb{E}[\langle a, a' \rangle \langle b, b' \rangle]}{\sqrt{\mathbb{E}[\langle a, a' \rangle^2] \mathbb{E}[\langle b, b' \rangle^2]}}.$$

The empirical version of canonical correlation is similar:

$$\rho(A, B) = \max_{a', b'} \frac{a'^\top \Sigma_{AB} b'}{\sqrt{a'^\top \Sigma_A a'} \sqrt{b'^\top \Sigma_B b'}}.$$

The directions a', b' that maximize these objectives are called the canonical correlation directions. Subsequent canonical correlations can be found by optimizing the same objective subject to the additional constraints that the candidate directions a', b' must be orthogonal to previous canonical correlation directions with respect to the inner products induced by Σ_a and Σ_b. There are various spectral characterizations of canonical correlation (Borga, 2001), but we use the particular fact that the eigenvalues of the matrix $\Sigma_a^{-1/2} \Sigma_{ab} \Sigma_b^{-1} \Sigma_{ba} \Sigma_a^{-1/2}$ give the squared canonical correlations between a and b.

Chapter 3

Compositional Word Embeddings

In this chapter, we consider the domain of natural language processing. Text data are available in massive amounts on the internet, but the common ways words are encoded digitally completely ignore their semantic meaning. Unsupervised representation learning has accordingly made a tremendous impact in this area. We focus on learning representations for words, commonly called *word embeddings*, that capture semantic and syntactic structure.

Acknowledgements The results in this chapter are joint work with Rong Ge, and are published in Frandsen und Ge (2019).

3.1 Introduction

Word embeddings have become one of the most popular techniques in natural language processing. A word embedding maps each word in the vocabulary to a low dimensional vector. Several algorithms (e.g., Mikolov u. a. (2013); Pennington u. a. (2014)) can produce word embedding vectors whose distances or inner-products capture semantic relationships between words. The vector representations are useful for solving many NLP tasks, such as analogy tasks (Mikolov u. a. (2013)) or serving as features for supervised learning problems (Maas u. a. (2011)).

While word embeddings are good at capturing the semantic information of a single word, a key challenge is the problem of *composition*: how to combine the embeddings of two co-occurring, syntactically related words to an embedding of the entire phrase.

In practice composition is often done by simply adding the embeddings of the two words, but this may not be appropriate when the combined meaning of the two words differ significantly from the meaning of individual words (e.g., "complex number" should not just be "complex"+"number").

In this chapter, we try to learn a latent variable model for word embeddings that incorporates syntactic information and naturally leads to better compositions for syntactically related word pairs. Our model is motivated by the principled approach for understanding word embeddings initiated by Arora u. a. (2015), and models for composition similar to Coecke u. a. (2010).

Arora u. a. (2015) gave a generative model (RAND-WALK) for word embeddings, and showed several previous algorithms can be interpreted as finding the hidden parameters of this model. However, the RAND-WALK model does not treat syntactically related word-pairs differently from other word pairs. We give a generative model called syntactic RAND-WALK (see Section 3.2) that is capable of capturing specific syntactic relations (e.g., adjective-noun or verb-object pairs). Taking adjective-noun pairs as an example, previous works (Socher u. a. (2012); Baroni und Zamparelli (2010); Maillard und Clark (2015)) have tried to model the adjective as a linear operator (a matrix) that can act on the embedding of the noun. However, this would require learning a $d \times d$ matrix for each adjective while the normal embedding only has dimension d. In our model, we use a core tensor $T \in \mathbb{R}^{d \times d \times d}$ to capture the relations between a pair of words and its context. In particular, using the tensor T and the word embedding for the adjective, it is possible to define a matrix for the adjective that can be used as an operator on the embedding of the noun. Therefore our model allows the same interpretations as many previous models while having much fewer parameters to train.

One salient feature of our model is that it makes good use of high order statistics. Standard word embeddings are based on the observation that the semantic information

of a word can be captured by words that appear close to it. Hence most algorithms use pairwise co-occurrence between words to learn the embeddings. However, for the composition problem, the phrase of interest already has two words, so it would be natural to consider co-occurrences between at least three words (the two words in the phrase and their neighbors).

Based on the model, we can prove an elegant relationship between high order co-occurrences of words and the model parameters. In particular, we show that if we measure the Pointwise Mutual Information (PMI) between three words, and form an $n \times n \times n$ tensor that is indexed by three words a, b, w, then the tensor has a Tucker decomposition that exactly matches our core tensor T and the word embeddings (see Theorem 1 and Corollary 1). This suggests a natural way of learning our model using a tensor decomposition algorithm.

Our model also allows us to approach the composition problem with more theoretical insights. Based on our model, if words a, b have the particular syntactic relationships we are modeling, their composition will be a vector $v_a + v_b + T(v_a, v_b, \cdot)$. Here v_a, v_b are the embeddings for word a and b, and the tensor gives an additional correction term. By choosing different core tensors it is possible to recover many previous composition methods. We discuss this further in Section 3.2.

Finally, we train our new model on a large corpus and give experimental evaluations. In the experiments, we show that the model learned satisfies the new assumptions that we need. We also give both qualitative and quantitative results for the new embeddings. Our embeddings and the novel composition method can capture the specific meaning of adjective-noun phrases in a way that is impossible by simply "adding" the meaning of the individual words. Quantitative experiment also shows that our composition vector are better correlated with humans on a phrase similarity task.

3.1.1 Related Work

Syntax and word embeddings Many well-known word embedding methods (e.g., Pennington u. a. (2014); Mikolov u. a. (2013)) don't explicitly utilize or model syntactic structure within text. Andreas und Klein (2014) find that such syntax-blind word embeddings fail to capture syntactic information above and beyond what a statistical parser can obtain, suggesting that more work is required to build syntax into word embeddings.

Several syntax-aware embedding algorithms have been proposed to address this. Levy und Goldberg (2014a) propose a syntax-oriented variant of the well-known skip-gram algorithm of Mikolov u. a. (2013), using contexts generated from syntactic dependency-based contexts obtained with a parser. Cheng und Kartsaklis (2015) build syntax-awareness into a neural network model for word embeddings by indroducing a negative set of samples in which the order of the context words is shuffled, in hopes that the syntactic elements which are sensitive to word order will be captured.

Word embedding composition Several works have addressed the problem of composition for word embeddings. On the theoretical side, Gittens u. a. (2017) give a theoretical justification for additive embedding composition in word models that satisfy certain assumptions, such as the skip-gram model, but these assumptions don't address syntax explicitly. Coecke u. a. (2010) present a mathematical framework for reasoning about syntax-aware word embedding composition that motivated our syntactic RAND-WALK model. Our new contribution is a concrete and practical learning algorithm with theoretical guarantees. Mitchell und Lapata (2008, 2010) explore various composition methods that involve both additive and multiplicative interactions between the component embeddings, but some of these are limited by the need to learn additional parameters post-hoc in a supervised fashion.

Guevara (2010) get around this drawback by first training word embeddings for each word and also for tokenized adjective-noun pairs. Then, the composition model is trained by using the constituent adjective and noun embeddings as input and the adjective-noun token embedding as the predictive target. Maillard und Clark (2015) treat adjectives as matrices and nouns as vectors, so that the composition of an adjective and noun is just matrix-vector multiplication. The matrices and vectors are learned through an extension of the skip-gram model with negative sampling. In contrast to these approaches, our model gives rise to a syntax-aware composition function, which can be learned along with the word embeddings in an unsupervised fashion, and which generalizes many previous composition methods (see Section 3.2.3 for more discussion).

Tensor factorization for word embeddings As Levy und Goldberg (2014b) and Li u. a. (2015) point out, some popular word embedding methods are closely connected matrix factorization problems involving pointwise mutual information (PMI) and word-word co-occurrences. It is natural to consider generalizing this basic approach to tensor decomposition. Sharan und Valiant (2017) demonstrate this technique by performing a CP decomposition on triple word co-occurrence counts. Bailey und Aeron (2017) explore this idea further by defining a third-order generalization of PMI, and then performing a symmetric CP decomposition on the resulting tensor. In contrast to these recent works, our approach arives naturally at the more general Tucker decomposition due to the syntactic structure in our model. Our model also suggests a different (yet still common) definition of third-order PMI.

3.2 Syntactic RAND-WALK Model

In this section, we introduce our syntactic RAND-WALK model and present formulas for inference in the model. We also derive a novel composition technique that emerges from the model.

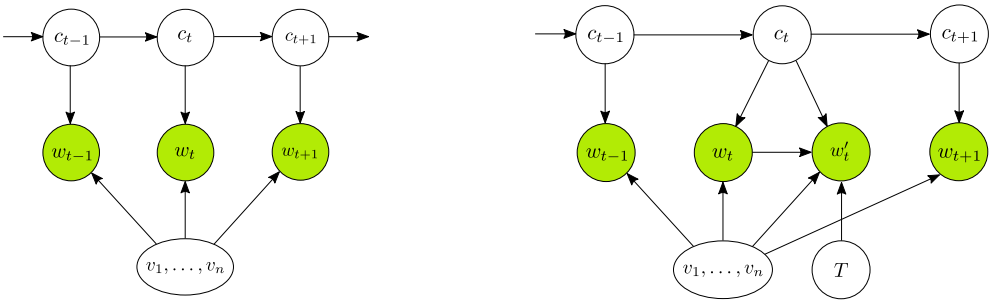
RAND-WALK Model We first briefly review the RAND-WALK model (Arora u. a. (2015)). In this model, a corpus of text is considered as a sequence of random variables $w_1, w_2, w_3, \ldots$, where w_t takes values in a vocabulary V of n words. Each word $w \in V$ has a word embedding $v_w \in \mathbb{R}^d$. The prior for the word embeddings is $v_w = s \cdot \hat{v}$, where s is a positive bounded scalar random variable with constant expectation τ and upper bound κ, and $\hat{v} \sim N(0, I)$.

The distribution of each w_t is determined in part by a random walk $\{c_t \in \mathbb{R}^d \,|\, t = 1, 2, 3 \ldots\}$, where c_t – called a *discourse vector* – represents the topic of the text at position t. This random walk is slow-moving in the sense that $\|c_{t+1} - c_t\|$ is small, but mixes quickly to a stationary distribution that is uniform on the unit sphere, which we denote by $\mathcal{C}$.

Let $\mathscr{C}$ denote the sequence of discourse vectors, and let $\mathscr{V}$ denote the set of word embeddings. Given these latent variables, the model specifies the following conditional probability distribution:

$$\Pr[w_t = w \,|c_t] \propto \exp(\langle v_w, c_t \rangle). \tag{3.1}$$

The graphical model depiction of RAND-WALK is shown in Figure 3.1a.



(a) RAND-WALK

(b) Syntactic RAND-WALK

Figure 3.1: Graphical models of RAND-WALK (left) and our new model (right), depicting a syntactic word pair (w_t, w'_t)

3.2.1 Syntactic RAND-WALK

One limitation of RAND-WALK is that it can't deal with syntactic relationships between words. Observe that conditioned on c_t and $\mathscr{V}$, w_t is independent of the other words in the text. However, in natural language, words can exhibit more complex dependencies, e.g. adjective-noun pairs, subject-verb-object triples, and other syntactic or grammatical structures.

In our syntactic RAND-WALK model, we start to address this issue by introducing direct pairwise word dependencies in the model. When there is a direct dependence between two words, we call the two words a *syntactic word pair*. In RAND-WALK, the interaction between a word embedding v and a discourse vector c is mediated by their inner product $\langle v, c \rangle$. When modeling a syntactic word pair, we need to mediate the interaction between *three* quantities, namely a discourse vector c and the word embeddings v and v' of the two relevant words. A natural generalization is to use a trilinear form defined by a tensor T, i.e.

$$T(v, v', c) = \sum_{i,j,k=1}^{d} T_{i,j,k} v(i) v'(j) c(k).$$

Here, $T \in \mathbb{R}^{d \times d \times d}$ is also a latent random variable, which we call the *composition tensor*.

We model a syntactic word pair as a single semantic unit within the text (e.g. in the case of adjective-noun phrases). We realize this choice by allowing each discourse vector c_t to generate a pair of words w_t, w_t' with some small probability p_{syn}. To generate a syntactic word pair w_t, w_t', we first generate a *root word* w_t conditioned on c_t with probability proportional to $\exp(\langle c_t, w_t \rangle)$, and then we draw w_t' from a conditional distribution defined as follows:

$$\Pr[w_t' = b \,|\, w_t = a, \mathscr{C}, \mathscr{V}] \propto \exp(\langle c_t, v_b \rangle + T(v_a, v_b, c_t)). \tag{3.2}$$

Here $\exp(\langle c_t, v_b \rangle)$ would be proportional to the probability of generating word b in the original RAND-WALK model, without considering the syntactic relationship. The additional term $T(v_a, v_b, c_t)$ can be viewed as an adjustment based on the syntactic relationship.

We call this extended model Syntactic RAND-WALK. Figure 3.1b gives the graphical model depiction for a syntactic word pair, and we summarize the model below.

Definition 1 (Syntactic RAND-WALK model)**.** *The model consists of the following:*

1. *Each word w in vocabulary has a corresponding embedding $v_w \sim s \cdot \hat{v}_w$, where $s \in \mathbb{R}_{\geq 0}$ is bounded by κ and $\mathbb{E}[s] = \tau$; $\hat{v}_w \sim N(0, I_{d \times d})$.*

2. *The sequence of discourse vectors $c_1, ..., c_t$ are generated by a random walk on the unit sphere, $\|c_t - c_{t+1}\| \leq \epsilon_w / \sqrt{d}$ and the stationary distribution is uniform.*

3. *For each c_t, with probability $1 - p_{syn}$, it generates one word w_t with probability proportional to $\exp(\langle c_t, v_{w_t} \rangle)$.*

4. *For each c_t, with probability p_{syn}, it generates a syntactic pair w_t, w'_t with probability proportional to* $\exp(\langle c_t, v_{w_t}\rangle)$ *and* $\exp(\langle c_t, v_{w'_t}\rangle + T(v_{w_t}, v_{w'_t}, c_t))$ *respectively, where T is a $d \times d \times d$ composition tensor.*

3.2.2 Inference in the Model

We now calculate the marginal probabilities of observing pairs and triples of words under the syntactic RAND-WALK model. We will show that these marginal probabilities are closely related to the model parameters (word embeddings and the composition tensor). All proofs for results in this section are given in Section 3.3.

Throughout this section, we consider two adjacent context vectors c_t and c_{t+1}, and condition on the event that c_t generated a single word and c_{t+1} generated a syntactic pair[1]. The main bottleneck in computing the marginal probabilities is that the conditional probailities specified in equations (3.1) and (3.2) are not normalized. Indeed, for these equations to be exact, we would need to divide by the appropriate partition functions, namely $Z_{c_t} := \sum_{w \in V} \exp(\langle v_w, c_t\rangle)$ for the former and $Z_{c_t,a} := \sum_{w \in V} \exp(\langle c_t, v_w\rangle + T(v_a, v_w, c_t))$ for the latter. Fortunately, we show that under mild assumptions these quantities are highly concentrated. To do that we need to control the norm of the composition tensor.

Definition 2. *The composition tensor T is (K, ϵ)-bounded, if for any word embedding v_a, v_b, we have*

$$\|T(v_a, \cdot, \cdot) + I\|^2 \leq \frac{Kd\epsilon^2}{\log^2 n}; \quad \|T(v_a, \cdot, \cdot) + I\|_F^2 \leq Kd; \quad \|T(v_a, v_b, \cdot)\|^2 \leq Kd.$$

To make sure $\exp(\langle c_t, v_w\rangle + T(v_a, v_w, c_t))$ are within reasonable ranges, the value K in this definition should be interpreted as an absolute constant (like 5, similar to

[1] As we will see in Section 5.6, in practice it is easy to identify which words form a syntactic pair, so it is possible to condition on this event in training.

previous constants κ and τ). Intuitively these conditions make sure that the effect of the tensor cannot be too large, while still making sure the tensor component $T(v_a, v_b, c)$ can be comparable (or even larger than) $\langle v_b, c\rangle$. We have not tried to optimize the log factors in the constraint for $\|T(v_a, \cdot, \cdot) + I\|^2$.

Note that if the tensor component $T(v_a, \cdot, \cdot)$ has constant singular values (hence comparable to I), we know these conditions will be satisfied with $K = O(1)$ and $\epsilon = O(\frac{\log n}{\sqrt{d}})$. Later in Section 5.6 we verify that the tensors we learned indeed satisfy this condition. Now we are ready to state the concentration of partition functions:

Lemma 1 (Concentration of partition functions)**.** *For the syntactic RAND-WALK model, there exists a constant Z such that*

$$\Pr_{c\sim\mathcal{C}}[(1-\epsilon_z)Z \le Z_c \le (1+\epsilon_z)Z] \ge 1-\delta,$$

for $\epsilon_z = \tilde{O}(1/\sqrt{n})$ and $\delta = \exp(-\Omega(\log^2 n))$.

Furthermore, if the tensor T is (K, ϵ)-bounded, then for any fixed word $a \in V$, there exists a constant Z_a such that

$$\Pr_{c\sim\mathcal{C}}[(1-\epsilon_{z,a})Z_a \le Z_{c,a} \le (1+\epsilon_{z,a})Z_a] \ge 1-\delta,$$

for $\epsilon_{z,a} = O(\epsilon) + \tilde{O}(1/\sqrt{n})$ and $\delta = \exp(-\Omega(\log^2 n))$.

Using this lemma, we can obtain simple expressions for co-occurrence probabilities. In particular, for any fixed $w, a, b \in V$, we adopt the following notation:

$$p(a) := \Pr[w_{t+1} = a] \quad p(w, a) := \Pr[w_t = w, w_{t+1} = a]$$
$$p([a, b]) := \Pr[w_{t+1} = a, w'_{t+1} = b] \quad p(w, [a, b]) := \Pr[w_t = w, w_{t+1} = a, w'_{t+1} = b].$$

Here in particular we use $[a, b]$ to highlight the fact that a and b form a syntactic

pair. Note $p(w, a)$ is the same as the co-occurrence probability of words w and a if both of them are the only word generated by the discourse vector. Later we will also use $p(w, b)$ to denote $\Pr[w_t = w, w_{t+1} = b]$ (not $\Pr[w_t = w, w'_{t+1} = b]$).

We also require two additional properties of the word embeddings, namely that they are norm-bounded above by some constant times $\sqrt{d}$, and that all partition functions are bounded below by a positive constant. Both of these properties hold with high probability over the word embeddings provided $n \gg d \log d$ and $d \gg \log n$, as shown in the following lemma:

Lemma 2. *Assume that the composition tensor T is (K, ϵ)-bounded, where K is a constant. With probability at least $1 - \delta_1 - \delta_2$ over the word vectors, where $\delta_1 = \exp(\Theta(d \log d) - \Theta(n))$ and $\delta_2 = \exp(\Theta(\log n) - \Theta(d))$, there exist positive absolute constants γ and β such that $\|v_i\| \leq \kappa\gamma$ for each $i \in V$ and $Z_c \geq \beta$ and $Z_{c,a} \geq \beta$ for any unit vector $c \in \mathbb{R}^d$ and any word $a \in V$.*

We can now state the main result.

Theorem 1. *Suppose that the events referred to in Lemma 1 hold. Then*

$$\log p(a) = \frac{\|v_a\|^2}{2d} - \log Z \pm \epsilon_p \tag{3.3}$$

$$\log p(w, a) = \frac{\|v_w + v_a\|^2}{2d} - 2\log Z \pm \epsilon_p \tag{3.4}$$

$$\log p([a, b]) = \frac{\|v_a + v_b + T(v_a, v_b, \cdot)\|^2}{2d} - \log Z - \log Z_a \pm \epsilon_p \tag{3.5}$$

$$\log p(w, [a, b]) = \frac{\|v_w + v_a + v_b + T(v_a, v_b, \cdot)\|^2}{2d} - 2\log Z - \log Z_a \pm \epsilon_p \tag{3.6}$$

Here $\epsilon_p = O(\epsilon + \epsilon_w) + \tilde{O}(1/\sqrt{n} + 1/d)$, where ϵ is from the (K, ϵ)-boundedness of T and ϵ_w is from Definition 1.

3.2.3 Composition

Our model suggests that the latent discourse vectors contain the meaning of the text at each location. It is therefore reasonable to view the discourse vector c corresponding to a syntactic word pair (a, b) as a suitable representation for the phrase as a whole. The posterior distribution of c given (a, b) satisfies

$$\Pr[c_t = c \,|\, w_t = a, w'_t = b] \propto \frac{1}{Z_c Z_{c,a}} \exp\left(\langle v_a + v_b + T(v_a, v_b, \cdot), c\rangle\right) \Pr[c_t = c].$$

Since $\Pr[c_t = c]$ is constant, and since Z_c and $Z_{c,a}$ concentrate on values that don't depend on c, the MAP estimate of c given $[a, b]$, which we denote by $\hat{c}$, satisfies

$$\hat{c} \approx \arg\max_{\|c\|=1} \exp\left(\langle v_a + v_b + T(v_a, v_b, \cdot), c\rangle\right) = \frac{v_a + v_b + T(v_a, v_b, \cdot)}{\|v_a + v_b + T(v_a, v_b, \cdot)\|}.$$

Hence, we arrive at our basic tensor composition: for a syntactic word pair (a, b), the composite embedding for the phrase is $v_a + v_b + T(v_a, v_b, \cdot)$.

Note that our composition involves the traditional additive composition $v_a + v_b$, plus a correction term $T(v_a, v_b, \cdot)$. We can view $T(v_a, v_b, \cdot)$ as a matrix-vector multiplication $[T(v_a, \cdot, \cdot)]^\top v_b$, i.e. the composition tensor allows us to compactly associate a matrix with each word in the same vein as Maillard und Clark (2015). Depending on the actual value of T, the term $T(v_a, v_b, \cdot)$ can also recover any manner of linear or multiplicative interactions between v_a and v_b, such as those proposed in Mitchell und Lapata (2010).

3.3 Proofs for Section 3.2

In this section we will prove the main Theorem 1, which establishes the connection between the model parameters and the correlations of pairs/triples of words. As we explained in Section 3.2, a crucial step is to analyze the partition function of the

model and show that the partition functions are concentrated. We will do that in Section 3.3.1. We then prove the main theorem in Section 3.3.2. More details and some technical lemmas are deferred to Section 3.3.3

3.3.1 Concentration of Partition Function

In this section we will prove concentrations of partition functions (Lemma 1). Recall that we need the tensor to be (K, ϵ)-bounded (where K is a constant) for this to work. Note that K here should be considered as an absolute constant (like 5, in fact in Section 5.6 we show K is less than 4). The first part of this Lemma is exactly Lemma 2.1 in Arora u. a. (2015). Therefore we will focus on the proof of the second part.

For the second part, we know the probability of choosing a word b is proportional to $\exp(T(v_a, v_b, c) + \langle c, v_b \rangle) = \exp(\langle T(v_a, \cdot, c) + c, v_b \rangle)$. If the probability of choosing word w is proportional to $\exp(\langle r, v_w \rangle)$ for some vector r (think of $r = T(v_a, \cdot, c) + c$), then in expectation the partition function should be equal to $n\mathbb{E}_{v \sim \mathcal{D}_V}[\exp(\langle r, v \rangle)]$ (here $\mathcal{D}_V$ is the distribution of the word embeddings). When the number of words is large enough, we hope that with high probability the partition function is close to its expectation. Since the Gaussian distribution is spherical, we also know that the expected partition function $n\mathbb{E}_{v \sim \mathcal{D}_V}[\exp(\langle r, v \rangle)]$ should only depend on the norm of r. Therefore as long as we can prove the norm of $r = T(v_a, \cdot, c) + c$ remain similar for most c, we will be able to prove the desired result in the lemma.

We will first show the norm of $r = T(v_a, \cdot, c) + c$ is concentrated if the tensor T is (K, ϵ)-bounded. Throughout all subsequent proofs, we assume that $\epsilon < 1$ and $d \geq \log^2 n/\epsilon^2$.

Lemma 3. *Let v_a be a fixed word vector, and let c be a random discourse vector. If*

T is (K, ϵ)-bounded with $d \geq \log^2 n/\epsilon^2$, we have

$$\Pr[\|T(v_a, \cdot, c) + c\|^2 \in L \pm O(\epsilon)] \geq 1 - \delta,$$

where $0 \leq L \leq K$ is a constant that depends on v_a, and $\delta = \exp(-\Omega(\log^2 n))$.

Proof. Since c is a uniform random vector on the unit sphere, we can represent c as $c = z/\|z\|$, where $z \sim N(0, I)$ is a standard spherical Gaussian vector. For ease of notation, let $M = T(v_a, \cdot, \cdot) + I$, and write the singular value decomposition of M as $M = U\Sigma V^T$. Note that $\Sigma = \text{diag}(\lambda_1, \ldots, \lambda_d)$ and U and V are orthogonal matrices, so that in particular, the random variable $y = V^T z$ has the same distribution as z, i.e. its entries are i.i.d. standard normal random variables. Further, $\|Ux\|^2 = \|x\|^2$ for any vector x, since U is orthogonal. Hence, we have

$$\|T(v_a, \cdot, c) + c\|^2 = \frac{1}{\|z\|^2}\|Mz\|^2 = \frac{1}{\|z\|^2}\|U\Sigma y\|^2 = \frac{\sum_{i=1}^d \lambda_i^2 y_i^2}{\sum_{i=1}^d z_i^2}.$$

Since both the numerator and denominator of this quantity are generalized χ^2 random variables, we can apply Lemma 6 to get tail bounds on both. Observe that by assumption, we have $\lambda_i^2 \leq Kd\epsilon^2/\log^2 n$ for all i, and $\sum_{i=1}^d \lambda_i^2 \leq Kd$. Set $A = \sum_{i=1}^d \lambda_i^2 y_i^2$ and $B = \sum_{i=1}^d z_i^2$. Let $\lambda_{max}^2 = \max_{1 \leq i \leq d} \lambda_i^2$. Note that $\mathbb{E}[A] = \sum_{i=1}^d \lambda_i^2 \leq Kd$ and $\mathbb{E}[B] = d$.

We will apply Lemma 6 to prove concentration bounds for A, in this case we have

$$\Pr\left[|A - \mathbb{E}[A]| \geq 2\sqrt{\sum_{i=1}^d \lambda_i^4}\sqrt{x} + 2\lambda_{max}^2 x\right] \leq 2\exp(-x).$$

Under our assumptions, we know $\lambda_{max}^2 \leq Kd\epsilon^2/\log^2 n$ and so

$$\sqrt{\sum_{i=1}^{d} \lambda_i^4} \leq \sqrt{\lambda_{max}^2 \sum_{i=1}^{d} \lambda_i^2} \leq Kd\epsilon/\log n.$$

Taking $x = \frac{1}{16}\log^2 n$, we know $2\sqrt{\sum_{i=1}^{d} \lambda_i^4}\sqrt{x} + 2\lambda_{max}^2 x \leq Kd\epsilon$. Therefore

$$\Pr[|A - \mathbb{E}[A]| \geq Kd\epsilon] \leq 2\exp(-\Omega(\log^2 n)).$$

Similarly, we can apply Lemma 6 to B (in fact we can apply simpler concentration bounds for standard χ^2 distribution), and we get

$$\Pr[|B - \mathbb{E}[B]| \geq 2\sqrt{d}\sqrt{x} + 2x] \leq 2\exp(-x).$$

If we take $x = \frac{1}{16}\log^2 n$, we know $2\sqrt{d}\sqrt{x} + 2x \leq \epsilon d$. This implies

$$\Pr[|B - \mathbb{E}[B]| \geq d\epsilon] \leq 2\exp(-\Omega(\log^2 n)).$$

When both events happen we know $|\frac{A}{B} - \frac{\mathbb{E}[A]}{\mathbb{E}[B]}| \leq 4K\epsilon = O(\epsilon)$ (here K is considered as a constant). This finishes the proof.

□

Using this lemma, we will show that $n\mathbb{E}_{v\sim\mathcal{D}_V}[\exp(\langle r, v\rangle)]$, the expected condition number (where $r = T(v_a, \cdot, c) + c$), is concentrated

Lemma 4. *Let v_a be a fixed word vector, and let c be a random discourse vector. If T is (K, ϵ)-bounded, there exists Z_a such that we have*

$$\Pr[n\mathbb{E}_{v\sim\mathcal{D}_V}[\exp(\langle T(v_a, \cdot, c) + c, v\rangle)] \in Z_a(1 \pm O(\epsilon)] \geq 1 - \delta,$$

where $Z_a = \Theta(n)$ *depends on* v_a*, and* $\delta = \exp(-\Omega(\log^2 n))$.

Proof. We know $v = s \cdot \hat{v}$ where $\hat{v} \sim N(0, I)$ and s is a (random) scaling. Let $r = T(v_a, \cdot, c) + c$. Conditioned on s we know $\langle r, v \rangle$ is equivalent to a Gaussian random variable with standard deviation $\sigma = \|r\| s$. For this random variable we know

$$\begin{aligned}
\mathbb{E}[\exp(\langle r, v \rangle)|s] &= \int_x \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{x^2}{2\sigma^2}\right) \exp(x) dx \\
&= \int_x \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\sigma^2)^2}{2\sigma^2} + \sigma^2/2\right) dx \\
&= \exp(\sigma^2/2).
\end{aligned}$$

Hence,

$$\mathbb{E}[\exp(\langle r, v \rangle)|s] = \exp(s^2\|r\|^2/2).$$

Let $g(x) = \mathbb{E}_s[\exp(s^2 x/2)]$, we know $g'(x) = \mathbb{E}_s[\exp(s^2 x/2) \cdot (s^2/2)] \leq \kappa^2/2 \cdot g(x)$. In particular, this implies $g(x + \gamma) \leq \exp(\kappa^2\gamma/2) g(x)$ (for small γ).

By Lemma 3, we know with probability at least $1 - \Omega(\log^2 n)$, $\|r\|^2 \in L \pm O(\epsilon)$. Therefore, when this holds, we have

$$n\mathbb{E}_{v \sim \mathcal{D}_V}[\exp(\langle r, v \rangle)] \in ng(L - O(\epsilon)) \cdot [1, \exp(O(\epsilon\kappa^2/2)].$$

The multiplicative factor on the RHS is bounded by $1 + O(\epsilon)$ when ϵ is small enough (and κ is a constant). This finishes the proof. □

Now we know the expected partition function is concentrated (for almost all discourse vectors c), it remains to show when we have finitely many words the partition function is concentrated around its expectation. This was already proved in Arora u. a. (2015), so we use their lemma below:

Lemma 5. *For any fixed vector r (whose norm is bounded by a constant), with probability at least $1 - \exp(-\Omega(\log^2 n))$ over the choices of the words, we have*

$$\sum_{i=1}^{n} \exp(\langle r, v_i \rangle) \in n\mathbb{E}_{v \sim \mathcal{D}_V}[\exp(\langle r, v \rangle)](1 \pm \epsilon_z),$$

where $\epsilon_z = \tilde{O}(1/\sqrt{n})$.

This is essentially Lemma 2.1 in Arora u. a. (2015) (see Equation A.32). The version we stated is a bit different because we allow r to have an arbitrary constant norm (while in their proof vector r is the discourse vector c and has norm 1). This is a trivial corollary as we can move the norm of r into the distribution of the scaling factor s for the word embedding.

Finally we are ready to prove Lemma 1.

Proof of Lemma 1. The first part is exactly Lemma 2.1 in Arora u. a. (2015).

For the second part, note that the partition function $Z_{c,a} = \sum_{i=1}^{n} \langle T(v_a, \cdot, c) + c, v_i \rangle$. We will use $\mathbb{E}[Z_{c,a}]$ to denote its expectation over the randomness of the word embedding $\{v_i\}$. By Lemma 4, we know for at least $1 - \exp(-\Omega(\log^2 n))$ fraction of discourse vectors c, the expected partition function is concentrated ($\mathbb{E}[Z_{c,a}] \in (1 \pm O(\epsilon))Z_a$). Let $\mathcal{S}$ denote the set of c such that Lemma 4 holds. Now by Lemma 5 we know for any $x \in \mathcal{S}$, with probability at least $1 - \exp(-\Omega(\log^2 n)$ $Z_{c,a} \in (1 \pm \epsilon_z)\mathbb{E}[Z_{c,a}]$.

Therefore we know if we consider both c and the embedding as random variables, $\Pr[Z_{c,a} \in (1 \pm O(\epsilon + \epsilon_z))Z_a] \geq 1 - \delta'$ where $\delta' = \exp(-\Omega(\log^2 n))$. Let S be the set of word embedding such that there is at least $\sqrt{\delta'}$ fraction of c that does not satisfy $Z_{c,a} \in (1 \pm O(\epsilon + \epsilon_z))Z_a$, we must have $\Pr[S] \cdot \sqrt{\delta'} \leq \delta'$. Therefore

$$\Pr[S] \leq \sqrt{\delta'}.$$

That is, with probability at least $1 - \sqrt{\delta'}$ (over the word embeddings), there is at least $1 - \sqrt{\delta'}$ fraction of c such that $Z_{c,a} \in (1 \pm O(\epsilon + \epsilon_z))Z_a$.

□

3.3.2 Estimating the Correlations

In this section we prove Theorem 1. The proof is very similar to the proof of Theorem 2.2 in Arora u. a. (2015). We use several lemmas in that proof, and these lemmas are deferred to Section 3.3.3.

Proof of Theorem 1. Throughout this proof we consider two adjacent discourse vectors c, c', where c generated a single word w and c' generated a syntactic pair (a, b).

The first two results in Theorem 1 are exactly the same as Theorem 2.2 in Arora u. a. (2015). Therefore we only need to prove the result for $p([a, b])$ and $p(w, [a, b])$.

For $p([a, b])$, by definition of the model we know

$$p([a,b]) = \mathbb{E}_{c'}[\frac{1}{Z_{c'}}\frac{1}{Z_{c',a}}\exp(\langle c', v_a\rangle + \langle c', v_b\rangle + T(v_a, v_b, c')].$$

Here $Z_{c'}$ is the partition function $\sum_{i=1}^{n}\exp(\langle c', v_i\rangle)$, and $Z_{c',a}$ is the partition function $\sum_{i=1}^{n}\exp(\langle c', v_i\rangle + T(v_a, v_i, c')$.

Let $\mathcal{F}$ be the event that c' satisfies the equations in Lemma 1. Let $\bar{\mathcal{F}}$ be its negation. By Lemma 1 we know $\Pr[\mathcal{F}] \geq 1 - \exp(-\Omega(\log^2 n))$. Using this event, we can write

$$\begin{aligned} p([a,b]) =& \mathbb{E}_{c'}[\frac{1}{Z_{c'}}\frac{1}{Z_{c',a}}\exp(\langle c', v_a\rangle + \langle c', v_b\rangle + T(v_a, v_b, c'))1_{\mathcal{F}}] \\ &+ \mathbb{E}_{c'}[\frac{1}{Z_{c'}}\frac{1}{Z_{c',a}}\exp(\langle c', v_a\rangle + \langle c', v_b\rangle + T(v_a, v_b, c'))1_{\bar{\mathcal{F}}}]. \end{aligned}$$

The second term can be bounded by Lemma 7 and the fact that $Z_{c'}Z_{c',a} \geq \beta$ from Lemma 2. We know

$$\mathbb{E}_{c'}[\frac{1}{Z_{c'}}\frac{1}{Z_{c',a}}\exp(\langle c', v_a\rangle + \langle c', v_b\rangle + T(v_a, v_b, c')1_{\bar{\mathcal{F}}}] \leq \exp(-\Omega(\log^{1.8} n)).$$

For convenience, let ζ denote $\exp(-\Omega(\log^{1.8} n))$. For the first term, we know by Lemma 1 that there exists Z, Z_a that are close to $Z_{c'}$ and $Z_{c',a}$. Therefore

$$\begin{aligned}
p([a,b]) &= \mathbb{E}_{c'}[\frac{1}{Z_{c'}}\frac{1}{Z_{c',a}}\exp(\langle c', v_a\rangle + \langle c', v_b\rangle + T(v_a, v_b, c')1_{\mathcal{F}}] \\
&\quad + \mathbb{E}_{c'}[\frac{1}{Z_{c'}}\frac{1}{Z_{c',a}}\exp(\langle c', v_a\rangle + \langle c', v_b\rangle + T(v_a, v_b, c')1_{\bar{\mathcal{F}}}]. \\
&\leq (1+\epsilon_z)(1+\epsilon_{z,a})\mathbb{E}_{c'}[\frac{1}{Z}\frac{1}{Z_a}\exp(\langle c', v_a\rangle + \langle c', v_b\rangle + T(v_a, v_b, c')1_{\mathcal{F}}] + \zeta \\
&\leq \frac{(1+\epsilon_z)(1+\epsilon_{z,a})}{ZZ_a}\mathbb{E}_{c'}[\exp(\langle c', v_a\rangle + \langle c', v_b\rangle + T(v_a, v_b, c')] + \zeta \\
&\leq \frac{(1+\epsilon_z)(1+\epsilon_{z,a})(1+\tilde{O}(1/d))}{ZZ_a}\exp(\frac{\|v_a + v_b + T(v_a, v_b, \cdot)\|^2}{2d}) + \zeta.
\end{aligned}$$

Here the last step used Lemma 9. Since both Z and Z_a can be bounded by $O(n)$, and $\frac{\|v_a+v_b+T(v_a,v_b,\cdot)\|^2}{2d}$ is bounded by $(4\kappa + \sqrt{2K})^2$, we know the first term is of order $\Omega(1/n^2)$, and the second term is negligible.

For the lowerbound, we can have

$$\begin{aligned}
p([a,b]) &= \mathbb{E}_{c'}[\frac{1}{Z_{c'}}\frac{1}{Z_{c',a}}\exp(\langle c', v_a\rangle + \langle c', v_b\rangle + T(v_a, v_b, c')1_{\mathcal{F}}] \\
&\quad + \mathbb{E}_{c'}[\frac{1}{Z_{c'}}\frac{1}{Z_{c',a}}\exp(\langle c', v_a\rangle + \langle c', v_b\rangle + T(v_a, v_b, c')1_{\bar{\mathcal{F}}}]. \\
&\geq (1-\epsilon_z)(1-\epsilon_{z,a})\mathbb{E}_{c'}[\frac{1}{Z}\frac{1}{Z_a}\exp(\langle c', v_a\rangle + \langle c', v_b\rangle + T(v_a, v_b, c')1_{\mathcal{F}}] \\
&\geq \frac{(1-\epsilon_z)(1-\epsilon_{z,a})}{ZZ_a}\{\mathbb{E}_{c'}[\exp(\langle c', v_a\rangle + \langle c', v_b\rangle + T(v_a, v_b, c'))] \\
&\qquad\qquad -\mathbb{E}_{c'}[\exp(\langle c', v_a\rangle + \langle c', v_b\rangle + T(v_a, v_b, c'))1_{\bar{\mathcal{F}}}]\} \\
&\geq \frac{(1-\epsilon_z)(1-\epsilon_{z,a})}{ZZ_a}\{\mathbb{E}_{c'}[\exp(\langle c', v_a\rangle + \langle c', v_b\rangle + T(v_a, v_b, c'))] - \zeta\} \\
&\geq \frac{(1-\epsilon_z)(1-\epsilon_{z,a})(1-\tilde{O}(1/d))}{ZZ_a}\left\{\exp(\frac{\|v_a + v_b + T(v_a, v_b, \cdot)\|^2}{2d}) - \zeta\right\}.
\end{aligned}$$

Again the last step is using Lemma 9 and the term ζ is negligible. Combining the upper and lower bound, we know

$$\log p([a,b]) = \frac{\|v_a + v_b + T(v_a, v_b, \cdot)\|^2}{2d} - \log Z - \log Z_a \pm \epsilon_p,$$

where $\epsilon_p = O(\epsilon_z + \epsilon_{z,a}) + \tilde{O}(1/d)$.

Now we turn to the most complicated term $\log p(w, [a,b])$. By definition we know

$$p(w,[a,b]) = \mathbb{E}_{c,c'}[\frac{1}{Z_c}\exp(\langle c, v_w\rangle)\frac{1}{Z_{c'}}\frac{1}{Z_{c',a}}\exp(\langle c', v_a\rangle + \langle c', v_b\rangle + T(v_a, v_b, c')].$$

We will follow similar idea as before. Let $\mathcal{F}$ be the event that both c, c' satisfy the equations in Lemma 1 and $\bar{\mathcal{F}}$ be its negation. By Lemma 1 and union bound we know $\Pr[\mathcal{F}] \geq 1 - \exp(-\Omega(\log^2 n))$.

We again separate the co-occurrence probability based on the event $\mathcal{F}$:

$$\begin{aligned}p(w,[a,b]) =&\mathbb{E}_{c,c'}[\frac{1}{Z_c}\exp(\langle c, v_w\rangle)\frac{1}{Z_{c'}}\frac{1}{Z_{c',a}}\exp(\langle c', v_a\rangle + \langle c', v_b\rangle + T(v_a, v_b, c')1_{\mathcal{F}}]\\ &+\mathbb{E}_{c,c'}[\frac{1}{Z_c}\exp(\langle c, v_w\rangle)\frac{1}{Z_{c'}}\frac{1}{Z_{c',a}}\exp(\langle c', v_a\rangle + \langle c', v_b\rangle + T(v_a, v_b, c')1_{\bar{\mathcal{F}}}].\end{aligned}$$

For the second term, we can again use Lemma 7 to show that it is bounded by $\exp(-\Omega(\log^{1.8} n))$. Now, using techniques similar as before, we can prove

$$p(w,[a,b]) = (1\pm O(\epsilon_z+\epsilon_{z,a}))\frac{1}{Z^2 Z_a}\mathbb{E}_{c,c'}[\exp(\langle c, v_w\rangle)\exp(\langle c', v_a\rangle+\langle c', v_b\rangle+T(v_a, v_b, c'))]. \tag{3.7}$$

Now the final step is to use the fact that c and c' are close to simplify the final formula. Let $A(c') = \mathbb{E}_{c|c'}[\exp(\langle c, v_w\rangle)]$, by Lemma 8 we know $A(c') \in (1 \pm \epsilon_w)\exp(\langle v_w, c'\rangle)$. Therefore

$$\begin{aligned}&\mathbb{E}_{c,c'}[\exp(\langle c, v_w\rangle)\exp(\langle c', v_a\rangle + \langle c', v_b\rangle + T(v_a, v_b, c'))]\\ =&\mathbb{E}_{c'}[\exp(\langle c', v_a\rangle + \langle c', v_b\rangle + T(v_a, v_b, c'))\mathbb{E}_{c|c'}[\exp(\langle c, v_w\rangle)]]\\ =&\mathbb{E}_{c'}[\exp(\langle c', v_a\rangle + \langle c', v_b\rangle + T(v_a, v_b, c'))A(c')]\\ =&(1\pm\epsilon_w)\mathbb{E}_{c'}[\exp(\langle c', v_a\rangle + \langle c', v_b\rangle + T(v_a, v_b, c') + \langle c', v_w\rangle)]\\ =&(1\pm\epsilon_w)(1\pm\tilde{O}(1/d))\exp(\frac{\|v_w + v_a + v_b + T(v_a, v_b, \cdot)\|^2}{2d}).\end{aligned}$$

Here the last step is again by Lemma 9. Combining this with Equation (3.7) gives the result. □

3.3.3 Auxiliary Lemmas

Tail bound for χ^2 distribution We will use the following tail bounds for the generalized χ^2-squared distribution.

Lemma 6. *(Laurent und Massart (2000)) Let $y_1, \ldots, y_d$ be i.i.d. standard normal random variables, and let $a_1, \ldots, a_d$ be nonnegative real numbers. Set $Y = \sum_{i=1}^{d} a_i y_i^2$ and $a = (a_1, a_2, \ldots, a_d)$. Then the following hold for any positive real number x:*

$$P(Y - \mathbb{E}[Y] \geq 2\|a\|_2\sqrt{x} + 2\|a\|_\infty x) \leq \exp(-x)$$
$$P(Y - \mathbb{E}[Y] \leq -2\|a\|_2\sqrt{x}) \leq \exp(-x).$$

Additional Lemmas We will use several tools developed in Arora u. a. (2015). The first lemma allows us to bound the probabilities the discourse vector c does not satisfy the results of Lemma 1.

Lemma 7. *Let $\mathcal{F}$ be any event that depends on the discourse vector c with probability at least $1 - \exp(-\Omega(\log^2 n))$, and $\bar{\mathcal{F}}$ be its negation. Suppose r is a vector of norm $O(\sqrt{d})$, then*

$$\mathbb{E}_c[\exp(\langle r, c\rangle)1_{\bar{\mathcal{F}}}] \leq \exp(-\Omega(\log^{1.8} n)).$$

Further, if we consider two consecutive discourse vectors c, c', redefine $\mathcal{F}$ to be an event that can depend on both discourse vectors, again with probability at least $1 - \exp(-\Omega(\log^2 n))$. If r, r' are two vectors of norm $O(\sqrt{d})$ we have

$$\mathbb{E}_{c,c'}[\exp(\langle r, c\rangle)\exp(\langle r', c'\rangle)1_{\bar{\mathcal{F}}}] \leq \exp(-\Omega(\log^{1.8} n)).$$

Proof. The proof of this lemma appears on page 20 in Arora u. a. (2015), as a step in the proof of their Theorem 2.2. For completeness, we reproduce (and slightly adapt) their argument here.

Observe that

$$\mathbb{E}_c[\exp(\langle r, c\rangle)1_{\bar{\mathcal{F}}}] = \mathbb{E}_c[\exp(\langle r, c\rangle)1_{\langle r,c\rangle>0}1_{\bar{\mathcal{F}}}] + \mathbb{E}_c[\exp(\langle r, c\rangle)1_{\langle r,c\rangle<0}1_{\bar{\mathcal{F}}}].$$

The second term of 3.3.3 is upper bounded by

$$\mathbb{E}_c[1_{\bar{\mathcal{F}}}] \leq \exp(-\Omega(\log^2 n)).$$

Note that the first term of 3.3.3 can be bounded as follows:

$$\mathbb{E}_c[\exp(\langle r, c\rangle)1_{\langle r,c\rangle>0}1_{\bar{\mathcal{F}}}] \leq \mathbb{E}_c[\exp(\langle \alpha r, c\rangle)1_{\langle r,c\rangle>0}1_{\bar{\mathcal{F}}}] \leq \mathbb{E}_c[\exp(\langle \alpha r, c\rangle)1_{\bar{\mathcal{F}}}]$$

for $\alpha > 1$. Therefore, to obtain a bound on $\mathbb{E}_c[\exp(\langle r, c\rangle)1_{\langle r,c\rangle>0}1_{\bar{\mathcal{F}}}]$ is suffices to bound

$$\mathbb{E}_c[\exp(\langle r, c\rangle)1_{\bar{\mathcal{F}}}]$$

when $\|r\| = \Omega(\sqrt{d})$.

Let z denote the random variable $\langle r, c\rangle$, and let $r(z) = 1_{\bar{\mathcal{F}}}$. Using Lemma A.4 in Arora u. a. (2015), we have

$$\mathbb{E}_c[\exp(z)r(z)] \leq \mathbb{E}_c[\exp(z)1_{[t,\infty]}(z)],$$

where t satisfies that $\mathbb{E}_c[1_{[t,\infty]}(z)] = \Pr[z \geq t] = E_c[r(z)] \leq \exp(-\Omega(\log^2 n))$. Then by Lemma A.1 of Arora u. a. (2015), we have that $t \geq \Omega(\log^{.9} n)$. Finally, applying Corollary A.3 of Arora u. a. (2015), we have

$$\mathbb{E}_c[\exp(z)r(z)] \leq E_c[\exp(z)1_{[t,\infty]}(z)] = \exp(-\Omega(\log^{1.8} n)),$$

which completes the proof for the first part of this lemma.

The second part of this lemma can be proved in much the same fashion. By

Cauchy-Schwarz,

$$\begin{aligned}(\mathbb{E}_{c,c'}[\exp(\langle r,c\rangle)\exp(\langle r',c'\rangle)1_{\bar{\mathcal{F}}}])^2 &\leq \left(\mathbb{E}_{c,c'}[\exp(\langle r,c\rangle)^2 1_{\bar{\mathcal{F}}}]\right)\left(\mathbb{E}_{c,c'}[\exp(\langle r',c'\rangle)^2 1_{\bar{\mathcal{F}}}]\right)\\ &\leq \left(\mathbb{E}_c[\exp(\langle 2r,c\rangle)\mathbb{E}_{c'|c}[1_{\bar{\mathcal{F}}}]]\right)\left(\mathbb{E}_{c'}[\exp(\langle 2r',c'\rangle)\mathbb{E}_{c|c'}[1_{\bar{\mathcal{F}}}]]\right).\end{aligned}$$

Now we bound $\mathbb{E}_c[\exp(\langle 2r,c\rangle)\mathbb{E}_{c'|c}[1_{\bar{\mathcal{F}}}]]$ using the same argument as above in the first part of this proof, replacing $1_{\bar{\mathcal{F}}}$ with $\mathbb{E}_{c'|c}[1_{\bar{\mathcal{F}}}]$, r with $2r$, and $r(z) = 1_{\bar{\mathcal{F}}}$ with $r(z) = \mathbb{E}_{c'|z}[1_{\bar{\mathcal{F}}}]$. In particular, we have $\mathbb{E}_c[\exp(\langle 2r,c\rangle)\mathbb{E}_{c'|c}[1_{\bar{\mathcal{F}}}]] \leq \exp(-\Omega(\log^{1.8} n))$. Likewise, we have the same bound for $\mathbb{E}_{c'}[\exp(\langle 2r',c'\rangle)\mathbb{E}_{c|c'}[1_{\bar{\mathcal{F}}}]]$. Putting these two together, we conclude that

$$\begin{aligned}&\mathbb{E}_{c,c'}[\exp(\langle r,c\rangle)\exp(\langle r',c'\rangle)1_{\bar{\mathcal{F}}}]\\ &\qquad\leq \left(\mathbb{E}_c[\exp(\langle 2r,c\rangle)\mathbb{E}_{c'|c}[1_{\bar{\mathcal{F}}}]]\right)^{1/2}\left(\mathbb{E}_{c'}[\exp(\langle 2r',c'\rangle)\mathbb{E}_{c|c'}[1_{\bar{\mathcal{F}}}]]\right)^{1/2}\\ &\qquad\leq \exp(-\Omega(\log^{1.8} n)),\end{aligned}$$

as desired.

□

The next lemma allows us to handle the difference between two consecutive discourse vectors:

Lemma 8. *Let c, c' be two discourse vectors that are adjacent, let v_w be a word embedding satisfying $\|v_w\| \leq K'\sqrt{d}$, and let $A(c) := \mathbb{E}_{c'|c}[\exp(\langle v_w, c'\rangle)]$, then we have*

$$A(c) \in (1 \pm \epsilon_w)\exp(\langle v_w, c\rangle).$$

Proof. The proof of this lemma appears on page 21 in Arora u. a. (2015), again as a step in the proof of their Theorem 2.2. For completeness, we reproduce the argument

here.

Since $\|v_w\| \leq K'\sqrt{d}$ for some constant K', we have that $\langle v_w, c-c'\rangle \leq \|v_w\|\|c-c'\| \leq K'\sqrt{d}\|c-c'\|$. Hence,

$$\begin{aligned} A(c) &= \mathbb{E}_{c'|c}[\exp(\langle v_w, c'\rangle)] \\ &= \exp(\langle v_w, c\rangle)\mathbb{E}_{c'|c}[\exp(\langle v_w, c'-c\rangle)] \\ &\leq \exp(\langle v_w, c\rangle)\mathbb{E}_{c'|c}[K'\sqrt{d}\|c-c'\|)] \\ &\leq (1+\epsilon_w)\exp(\langle v_w, c\rangle), \end{aligned}$$

where the last inequality follows from our model assumptions.

To get the lower bound, observe that

$$\mathbb{E}_{c'|c}[\exp(K'\sqrt{d}\|c-c'\|)] + \mathbb{E}_{c'|c}[\exp(-K'\sqrt{d}\|c-c'\|)] \geq 2.$$

Therefore, the model assumptions imply that

$$\mathbb{E}_{c'|c}[\exp(-K'\sqrt{d}\|c-c'\|)] \geq 1-\epsilon_w.$$

Hence,

$$\begin{aligned} A(c) &= \exp(\langle v_w, c\rangle)\mathbb{E}_{c'|c}[\exp(\langle v_w, c'-c\rangle)] \\ &\geq \exp(\langle v_w, c\rangle)\mathbb{E}_{c'|c}[\exp(K'\sqrt{d}\|c-c'\|)] \\ &\geq (1-\epsilon_w)\exp(\langle v_w, c\rangle). \end{aligned}$$

□

The next lemma we use gives bound on $\mathbb{E}[\exp(\langle v, c\rangle)]$ where c is a uniform vector

on the unit sphere.

Lemma 9. *[Lemma A.5 in Arora u. a. (2015)] Let $v \in \mathbb{R}^d$ be a fixed vector with norm $\|v\| = O(\sqrt{d})$. For random variable c with uniform distribution over the sphere, we have that*

$$\log \mathbb{E}[\exp(\langle v, c\rangle)] = \frac{\|v\|^2}{2d} \pm \epsilon_c,$$

where $\epsilon_c = \tilde{O}(1/d)$.

We end with the proof of Lemma 2.

Proof of Lemma 2. Just for this proof, we use the following notation. Let $I_{d\times d}$ be the d-dimensional identity matrix, and let $x_1, x_2, \ldots, x_n$ be i.i.d. draws from $N(0, I_{d\times d})$. Let $y_i = \|x_i\|_2$, and note that y_i^2 is a standard χ-squared random variable with d degrees of freedom. Let κ be a positive constant, and let $s_1, s_2, \ldots, s_n$ be i.i.d. draws from a distribution supported on $[0, \kappa]$. Let $v_i = s_i \cdot x_i$. Define $Z_c = \sum_{i=1}^n \exp(\langle v_i, c\rangle)$, and define $Z_{c,a} = \sum_{i=1}^n \exp(\langle v_i, c\rangle + T(v_a, v_i, c))$.

We first cover the unit sphere by a finite number of metric balls of small radius. Then we show that with high probability, the partition function at the center of these balls is indeed bounded below by a constant. Finally, we show that the partition function evaluated at an arbitrary point on the unit sphere can't be too far from the partition function at one of the ball centers provided the norms of the v_i are not too large. We finish by appropriately controlling the norms of the v_i.

For $\epsilon > d^{-1}$, cover the unit sphere in $\mathbb{R}^d$ with $N = (\frac{2}{\epsilon} + 1)^d$ balls of radius ϵ. Let $c_1, c_2, \ldots, c_N$ be the centers of these balls (so that each c_i is a unit vector). Let $\alpha \geq 0$ be a constant. Note that $\langle v_j, c_i\rangle = \langle c_j, s_j \cdot c_i\rangle$ and $\langle v_k, c_i\rangle + T(v_l, v_k, c_i) = \langle x_k, s_k(I + T(v_l, \cdot, \cdot))^T c_i\rangle$ are Gaussian random variables with mean 0.

Let $\mathcal{F}_i$ be the event that there exists some $j, k \in [n]$ such that $\langle v_j, c_i\rangle \geq 0$ and

$\langle v_k + T(v_a, v_k, \cdot), c_i \rangle \geq 0$. Note that

$$\begin{aligned}
\Pr[\bar{\mathcal{F}}_i] &\leq \Pr[\forall j \in [n], \langle v_j, c_i \rangle \leq 0] + \Pr[\forall k \in [n], \langle v_k + T(v_a, v_k, \cdot), c_i \rangle \leq 0] \\
&= \prod_{j=1}^{n} \Pr[\langle v_j, c_i \rangle \leq 0] + \prod_{k}^{n} \Pr[\langle v_k + T(v_a, v_k, \cdot), c_i \rangle \leq 0] \\
&\leq \frac{1}{2^n} + \frac{1}{2^n} \\
&\leq \exp(-\Theta(n)).
\end{aligned}$$

Let $\gamma > 0$. Let $\mathcal{G}_i$ be the event that $y_i < \gamma\sqrt{d}$. Set $t = (\frac{1}{\sqrt{2}}\sqrt{\gamma^2 - \frac{1}{2}} - \frac{1}{2})^2 d$, so that $d + 2\sqrt{dt} + 2t = \gamma^2 d$. Then by Lemma 6,

$$\Pr[\bar{\mathcal{G}}_i] \leq \exp(-t).$$

Let $\mathcal{E} = \bigcap_{i=1}^{N} \mathcal{F}_i \bigcap_{i=1}^{n} \mathcal{G}_i$. Assume that the word embeddings satisfy the event $\mathcal{E}$. Let c_i be a center of one of the covering balls such that $\|c - c_i\|_2 < \epsilon$. Let v_j, v_k be vectors that satisfies $\langle x_j, c_i \rangle \geq -\alpha$ and $\langle v_k + T(v_a, v_k, \cdot), c_i \rangle \geq -\alpha$. By Cauchy-Schwarz and the definition of $\mathcal{E}$, we have

$$\begin{aligned}
\langle v_j, c \rangle &= \langle v_j, c_i \rangle + \langle v_j, c - c_i \rangle \\
&\geq -\|v_j\| \|c - c_i\| \\
&\geq -\epsilon\gamma\kappa\sqrt{d} \\
&= -\gamma\kappa d^{-1/2} \\
&\geq \ell
\end{aligned}$$

for some appropriate universal constant ℓ. Likewise, using the boundedness property

of T, we have

$$\begin{aligned}\langle v_k + T(v_a, v_k, \cdot), c\rangle &\geq -\epsilon\sqrt{K}\sqrt{d} \\ &= -\sqrt{K}d^{-1/2} \\ &\geq \ell.\end{aligned}$$

Hence,

$$Z_c = \sum_{i=1}^{n} \exp(\langle v_i, c\rangle) \geq \exp(\langle v_j, c\rangle) \geq \exp(-\ell)$$

and

$$Z_{c,a} = \sum_{i=1}^{n} \exp(\langle v_i, c\rangle + T(v_a, v_i, c)) \geq \exp(\langle v_k + T(v_a, v_k, \cdot), c\rangle) \geq \exp(-\ell).$$

It remains to analyze the probability of $\mathcal{E}$. By the union bound, we have

$$\begin{aligned}\Pr[\mathcal{E}] &\geq 1 - N\exp\left(-\frac{n\alpha^2}{2}\right) - n\exp(-t) \\ &= 1 - \exp(O(d\log d) - \Theta(n)) - \exp(\log n - (\frac{1}{\sqrt{2}}\sqrt{\gamma^2 - \frac{1}{2}} - \frac{1}{2})^2 d) \\ &= 1 - \exp(\Theta(d\log d) - \Theta(n)) - \exp(\Theta(\log n) - \Theta(d)).\end{aligned}$$

Note that this is a high probability if $n \gg d\log d$ and $d \gg \log n$. □

3.4 Learning

In this section we discuss how to learn the parameters of the syntactic RAND-WALK model. Theorem 1 provides key insights into the learning problem, since it relates joint probabilities between words (which can be estimated via co-occurrence counts) to the word embeddings and composition tensor. By examining these equations, we

can derive a particularly simple formula that captures these relationships. To state this equation, we define the PMI for 3 words as

$$PMI3(a, b, w) := \log \frac{p(w, [a, b])p(a)p(b)p(w)}{p(w, a)p(w, b)p([a, b])}. \tag{3.8}$$

We note that this is just one possible generalization of pointwise mutual information (PMI) to several random variables, but in the context of our model, it is a very natural definition as all the partition numbers will be canceled out. Indeed, as an immediate corollary of Theorem 1, we have

Corollary 1. *Suppose that the events referred to in Lemma 1 hold. Then for ϵ_p same as Theorem 1*

$$PMI3(a, b, w) = \frac{1}{d} T(v_a, v_b, v_w) \pm O(\epsilon_p). \tag{3.9}$$

That is, if we consider $PMI3(a, b, w)$ as a $n \times n \times n$ tensor, Equation (3.9) is exactly a Tucker decomposition of this tensor of Tucker rank d. Therefore, all the parameters of the syntactic RAND-WALK model can be obtained by finding the Tucker decomposition of the PMI3 tensor. This equation also provides a theoretical motivation for using third-order pointwise mutual information in learning word embeddings.

The proof of Corollary 1 is just a simple calculation based on Theorem 1:

Proof of Corollary 1. By the definition of PMI3, we know

$$\begin{aligned}
PMI3 &= \log p(w,[a,b]) + \log p(a) + \log p(b) + \log p(w) \\
&\quad - \log p(w,a) - \log p(w,b) - \log p([a,b]) \\
&= (\frac{\|v_w + v_a + v_b + T(v_a,v_b,\cdot)\|^2}{2d} - 2\log Z - \log Z_a) \\
&\quad + (\frac{\|v_a\|^2}{2d} + \frac{\|v_b\|^2}{2d} + \frac{\|v_w\|^2}{2d} - 3\log Z) \\
&\quad - (\frac{\|v_w + v_a\|^2}{2d} + \frac{\|v_w + v_b\|^2}{2d} - 4\log Z) \\
&\quad - (\frac{\|v_a + v_b + T(v_a,v_b,\cdot)\|^2}{2d} - \log Z - \log Z_a) \pm 7\epsilon \\
&= \frac{T(v_a,v_b,v_w)}{d} \pm 7\epsilon.
\end{aligned}$$

□

3.4.1 Implementation

We now discuss concrete details about our implementation of the learning algorithm.

Corpus. We train our model using a February 2018 dump of the English Wikipedia. The text is pre-processed to remove non-textual elements, stopwords, and rare words (words that appear less than 1000 within the corpus), resulting in a vocabulary of size 68,279. We generate a matrix of word-word co-occurrence counts using a window size of 5. To generate the tensors of adjective-noun-word and verb-object-word co-occurrence counts, we first run the Stanford Dependency Parser (Chen und Manning (2014)) on the corpus in order to identify all adjective-noun and verb-object word pairs, and then use context windows that don't cross sentence boundaries to populate the triple co-occurrence counts.

Training. We first train the word embeddings according to the RAND-WALK model, following Arora u. a. (2015). Using the learned word embeddings, we next train the composition tensor T via the following optimization problem

$$\min_{T,\{C_w\},C} \sum_{(a,b),w} f(X_{(a,b),w}) \left(\log(X_{(a,b),w}) - \|v_w + v_a + v_b + T(v_a, v_b, \cdot)\|^2 - C_a - C\right)^2,$$

where $X_{(a,b),w}$ denotes the number of co-occurrences of word w with the syntactic word pair (a, b) (a denotes the noun/object) and $f(x) = \min(x, 100)$. This objective function isn't precisely targeting the Tucker decomposition of the PMI3 tensor, but it is analogous to the training criterion used in Arora u. a. (2015), and can be viewed as a negative log-likelihood for the model. To reduce the number of parameters, we constrain T to have CP rank 1000. We also trained the embeddings and tensor jointly, but found that this approach yields very similar results. In all cases, we utilize the Tensorflow framework (Abadi u. a. (2016)) with the Adam optimizer (Kingma und Ba (2014)) (using default parameters), and train for 1-5 epochs.

3.5 Experimental Verification

In this section, we verify and evaluate our model empirically on select qualitative and quantitative tasks. In all of our experiments, we focus solely on syntactic word pairs formed by adjective-noun phrases, where the noun is considered the root word.

3.5.1 Model Verification

Arora u. a. (2015) empirically verify the model assumptions of RAND-WALK, and since we trained our embeddings in the same way, we don't repeat their verifications here. Instead, we verify two key properties of syntactic RAND-WALK.

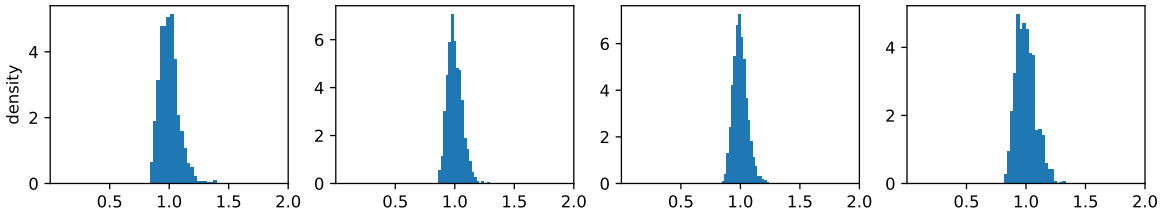


Figure 3.2: Histograms of partition functions $Z_{c,a}$ (x-axis is $Z_{c,a}/\mathbb{E}[Z_{c,a}]$)

Norm of composition tensor We check the assumptions that the tensor T is (K, ϵ)-bounded. Ranging over all adjective-noun pairs in the corpus, we find that $\frac{1}{d}\|T(v_a, \cdot, \cdot) + I\|^2$ has mean 0.052 and maximum 0.248, $\frac{1}{d}\|T(v_a, \cdot, \cdot) + I\|_F^2$ has mean 1.61 and maximum 3.23, and $\frac{1}{d}\|T(v_a, v_b, \cdot)\|^2$ has mean 0.016 and maximum 0.25. Each of these three quantities has a well-bounded mean, but $\|T(v_a, \cdot, \cdot) + I\|^2$ has some larger outliers. If we ignore the log factors (which are likely due to artifacts in the proof) in Definition 2, the tensor is (K, ϵ) bounded for $K = 4$ and $\epsilon = 0.25$.

Concentration of partition functions In addition to Definition 2, we also directly check its implications: our model predicts that the partition functions $Z_{c,a}$ concentrate around their means. To check this, given a noun a, we draw 1000 random vectors c from the unit sphere, and plot the histogram of $Z_{c,a}$.Results for a few randomly selected words a are given in Figure 3.2. All partition functions that we inspected exhibited good concentration.

3.5.2 Qualitative Analysis of Composition

We test the performance of our new composition for adjective-noun and verb-object pairs by looking for the words with closest embedding to the composed vector. For a phrase (a, b), we compute $c = v_a + v_b + T(v_a, v_b, \cdot)$, and then retrieve the words w whose embeddings v_w have the largest cosine similarity to c. We compare our results to the additive composition method. Tables 3.1 and 3.2 show results for three

Table 3.1: Top 10 words relating to adjective-noun phrases

civil war		complex numbers		national park	
additive	tensor	additive	tensor	additive	tensor
war	civil	complex	complex	national	yosemite
civil	somalian	numbers	eigenvalues	park	denali
military	eicher	number	numbers	parks	gunung
army	crimean	function	hermitian	recreation	kenai
conflict	laotian	complexes	quaternions	forest	nps
wars	francoist	functions	marginalia	historic	teton
fought	ulysses	integers	azadi	heritage	refuges
revolutionary	liberian	multiplication	rationals	wildlife	tilden
forces	confederate	algebraic	holomorphic	memorial	snowdonia
outbreak	midst	integer	rhythmically	south	jigme

adjective-noun and verb-object phrases. In each case, the tensor composition is able to retrieve some words that are more specifically related to the phrase. However, the tensor composition also sometimes retrieves words that seem unrelated to either word in the phrase. We conjecture that this might be due to the sparseness of co-occurrence of three words. We also observed cases where the tensor composition method was about on par with or inferior to the additive composition method for retrieving relevant words, particularly in the case of low-frequency phrases.

To focus specifically on high-frequency phrases, in Table 3.3 we show results for the phrases "giving birth", "solve problem", and "changing name". These phrases are all among the top 500 most frequent verb-object phrases appearing in the training corpus. In these examples, the tensor-based phrase embeddings retrieve words that are generally markedly more related to the phrase at hand, and there are no strange false positives. These examples demonstrate how a verb-object phrase can encompass an action that isn't implied simply by the object or verb alone. The additive composition doesn't capture this action as well as the tensor composition.

Table 3.2: Top 10 words relating to verb-object phrases

took place		took part		took lead	
additive	tensor	additive	tensor	additive	tensor
place	occurred	part	participated	took	equalised
took	scheduled	took	participating	lead	halftime
death	commenced	taking	participate	taking	nailing
take	event	take	culminated	take	kenseth
taking	events	taken	organised	went	fumbled
birth	culminated	takes	participation	led	touchdown
taken	thursday	became	hostilities	taken	furlongs
takes	friday	came	culminating	came	trailed
came	postponed	put	invasion	put	keselowski
held	lasted	whole	undertook	wanted	peloton

Table 3.3: Top 10 words relating to high-frequency verb-object phrases

giving birth		solve problem		changing name	
additive	tensor	additive	tensor	additive	tensor
birth	stillborn	problem	analytically	name	rebrand
giving	unborn	solve	creatively	changing	refocus
place	pregnant	problems	solve	change	redevelop
death	fathered	solving	subconsciously	changed	rebranding
give	litters	solved	devising	names	forgo
date	childbirth	solves	devise	referring	divest
gave	remarry	understand	proactively	title	rechristened
summary	newborn	resolve	solvers	word	afresh
gives	gestation	solution	extrapolate	actually	rebranded
given	eloped	question	rationalize	something	opting

3.5.3 Phrase Similarity

We also test our tensor composition method on a adjective-noun phrase similarity task using the dataset introduced by Mitchell und Lapata (2010). The data consists of 108 pairs each of adjective-noun and verb-object phrases that have been given similarity ratings by a group of 54 humans. The task is to use the word embeddings to produce similarity scores that correlate well with the human scores; we use both the Spearman rank correlation and the Pearson correlation as evaluation metrics for this task. We note that the human similarity judgments are somewhat noisy; intersubject agreement for the task is 0.52 as reported in Mitchell und Lapata (2010).

Given a phrase (a, b) with embeddings v_a, v_b, respectively, we found that the tensor composition $v_a + v_b + T(v_a, v_b, \cdot)$ yields worse performance than the simple additive composition $v_a + v_b$. For this reason, we consider a *weighted* tensor composition $v_a + v_b + \alpha T(v_a, v_b, \cdot)$ with $\alpha \geq 0$. Following Mitchell und Lapata (2010), we split the data into a development set of 18 humans and a test set of the remaining 36 humans. We use the development set to select the optimal scalar weight for the weighted tensor composition, and using this fixed parameter, we report the results using the test set. We repeat this three times, rotating over folds of 18 subjects, and report the average results.

As a baseline, we also report the average results using just the additive composition, as well as a weighted additive composition $\beta v_a + v_b$, where $\beta \geq 0$. We select β using the development set ("weighted1") and the test set ("weighted2"). We allow weighted2 to cheat in this way because it provides an upper bound on the best possible weighted additive composition. Additionally, we compare our method to the smoothed inverse frequency ("sif") weighting method that has been demonstrated to be near state-of-the-art for sentence embedding tasks (Arora u. a. (2016)). We also test embeddings of the form $p + \gamma \omega_a \omega_b T(v_a, v_b, \cdot)$ ("sif+tensor"), where p is the sif embedding for (a, b),

ω_a and ω_b are the smoothed inverse frequency weights used in the sif embeddings, and γ is a positive weight selected using the development set. The motivation for this hybrid embedding is to evaluate the extent to which the sif embedding and tensor component can independently improve performance on this task.

We perform these same experiments using two other standard sets of pre-computed word embeddings, namely GloVe[2] and carefully optimized cbow vectors[3] (Mikolov u. a. (2017)). We re-trained the composition tensor using the same corpus and technique as before, but substituting these pre-computed embeddings in place of the RAND-WALK (rw) embeddings. However, a bit of care must be taken here, since our syntactic RAND-WALK model constrains the norm of the word embeddings to be related to the frequency of the words, whereas this is not the case with the pre-computed embeddings. To deal with this, we rescaled the pre-computed embeddings sets to have the same norms as their counterparts in the rw embeddings, and then trained the composition tensor using these rescaled embeddings. At test time, we use the *original* embeddings to compute the additive components of our compositions, but use the *rescaled* versions when computing the tensor components.

The results for adjective-noun phrases are given in Tables 3.4. We observe that the tensor composition outperforms the additive compositions on all embedding sets apart from the Spearman correlation on the cbow vectors, where the weighted additive 2 method has a slight edge. The sif embeddings outperform the additive and tensor methods, but combining the sif embeddings and the tensor components yields the best performance across the board, suggesting that the composition tensor captures additional information beyond the individual word embeddings that is useful for this task. There was high consistency across the folds for the optimal weight parameter α,

[2]obtained from `https://nlp.stanford.edu/projects/glove/`

[3]obtained from `https://fasttext.cc/docs/en/english-vectors.html`

Table 3.4: Correlation measures between human judgments and embedding-based similarity scores (Spearman, Pearson) for adjective-noun phrases

	additive	weighted1	weighted2	tensor	sif	sif+tensor
rw	.446, .438	.444, .448	.452, .453	.460, .465	.482, .477	**.482**, **.481**
glove	.357, .336	.351, .334	.358, .345	.368, .347	.429, .434	**.433**, **.437**
cbow	.471, .452	.469, .451	.476, .456	.474, .471	.489, .482	**.492**, **.484**

Table 3.5: Correlation measures between human judgments and embedding-based similarity scores (Spearman, Pearson) for verb-object phrases

	additive	weighted1	weighted2	tensor	sif	sif+tensor
rw	.379, .370	.391, .385	**.392**, **.387**	.379, .370	.378, .351	.378, .363
glove	.397, .400	.398, .404	.401, .404	.410, **.420**	.387, .380	**.411**, .409
cbow	.423, .414	.423, .410	**.428**, .415	**.428**, **.422**	.404, .404	.420, .417

with $\alpha = 0.4$ for the rw embeddings, $\alpha = .2, .3$ for the glove embeddings, and $\alpha = .3$ for the cbow embeddings. For the sif+tensor embeddings, γ was typically in the range $[.1, .2]$.

The results for verb-object phrases are given in Table 3.5. Predicting phrase similarity appears to be harder in this case. Notably, the sif embeddings perform worse than unweighted vector addition. As before, we can improve the sif embeddings by adding in the tensor component. The tensor composition method achieves the best results for the glove and cbow vectors, but weighted addition works best for the randwalk vectors.

Overall, these results demonstrate that the composition tensor can improve the quality of the phrase embeddings in many cases, and the improvements are at least somewhat orthogonal to improvements resulting from the sif embedding method. This suggests that a well-trained composition tensor used in conjunction with high quality word embeddings and additional embedding composition techniques has the potential to improve performance in downstream NLP tasks.

Table 3.6: Test accuracy for sentiment analysis task (standard deviation reported in parentheses)

Dataset	Additive	Tensor
Pang and Lee	0.741 (0.018)	0.759 (0.025)
Large Movie Review	0.793	0.794

3.5.4 Sentiment Analysis

We finally test the effect of using the composition tensor for a sentiment analysis task. We use the movie review dataset of Pang und Lee (2004) as well as the Large Movie Review dataset (Maas u. a. (2011)), which consist of 2,000 movie reviews and 50,000 movie reviews, respectively. For a fixed review, we identify each adjective-noun pair (a, b) and compute $T(v_a, v_b, \cdot)$. We add these compositions together with the word embeddings for all of the words in the review, and then normalize the resulting sum. This vector is used as the input to a regularized logistic regression classifier, which we train using scikit-learn (Pedregosa u. a. (2011)) with the default parameters. We also consider a baseline method where we simply add together all of the word embeddings in the movie review, and then normalize the sum. We evaluate the test accuracy of each method using 5-fold cross-validation on the smaller dataset and the training-test set split provided in the larger dataset. Results are shown in Table 3.6. Although the tensor method seems to have a slight edge over the baseline, the differences are not significant.

3.6 Conclusion

In this chapter, we proposed a latent variable model for word embeddings that explicitly accounts for syntax and provides a natural composition function for syntactically-related phrases. We further proved that learning the word embeddings and composition

tensor can be reduced to an approximate Tucker decomposition of the PMI3 tensor. We then developed a practical learning algorithm and demonstrated that the composition tensor encodes meaningful information about syntactically-related word pairs that isn't captured by the standard additive composition.

There are several interesting open questions related to this work. Firstly, can we expand our model to account for higher-order grammatical structure, such as a full dependency or constituent parse tree for a sentence? Our model may generalize more naturally to the case of dependency parsing, which expresses the syntactic structure of a sentence as a tree whose nodes are in one-to-one correspondence with the individual words in the sentence. An edge between a parent and child node indicates a certain type of grammatical dependence, such as adjective (child) and noun (parent), or object (child) and verb (parent). The syntactic RAND-WALK model can already account for each such pairwise dependence on its own, but it is not immediately clear how to account for the complete tree structure in an analytically and computationally tractable way.

Another related open question is whether the model can try to *jointly* learn the syntactic dependence and the word embeddings. As it stands, our model is always conditioned on the given syntactic structure, which we extract as a preprocessing step by using an off-the-shelf dependency parser. Is there an effective way to fuse generative models for syntactic dependence with our word embedding model?

A final open question deals with how best to train the model. Although we showed theoretically that the model can be trained through a Tucker decomposition, in practice we formulated the learning problem slightly differently as a nonlinear least-square problem simply targeting the triple co-occurrence counts. Can we obtain better empirical results by attempting to directly solve the Tucker decomposition, and can this be done in a scalable manner? In the next chapter, we study the Tucker

decomposition in its own right, and show that efficient, scalable local search algorithms with provable guarantees may indeed be a viable approach worth exploring.

Chapter 4

Tucker Decomposition via Local Search

In Chapter 3, we proposed the syntactic RAND-WALK model, which included a latent tensor that acted as a bilinear composition function for syntactically-related word embeddings. We showed that this composition tensor together with the latent word embeddings formed a low-rank Tucker decomposition that approximates the 3-way PMI statistic tensor, thus reducing the problem of learning the latent variable to a tensor decomposition problem. Indeed, the Tucker decomposition emerges as an important primitive in the context of other representation learning applications, including image analysis and facial recognition (Vasilescu und Terzopoulos, 2002), data compression (Wang und Ahuja, 2004), and handwritten digits recognition (Savas und Eldén, 2007). In this chapter, we study the Tucker decomposition problem further.

Acknowledgements The results in this chapter are joint work with Rong Ge, and are published in Frandsen und Ge (2020).

4.1 Introduction

Tensor decompositions have been widely applied in data analysis and machine learning. In modern applications, the dimension of the tensor and the amount of data available are often quite large. In practice, simple local search algorithms such as stochastic gradient descent are often used. Even for matrix problems where exact solutions can be computed, local search algorithms are often applied directly to a nonconvex objective (Koren, 2009; Recht und Ré, 2013). Recently, a line of work (Ge u. a., 2015;

Bhojanapalli u. a., 2016; Sun u. a., 2016a; Ge u. a., 2016; Sun u. a., 2016b; Bandeira u. a., 2016) showed that although these problems have nonconvex objectives, they can still be solved by local search algorithms, because they have a simple *optimization landscape*. In particular, for matrix problems such as matrix sensing (Bhojanapalli u. a., 2016; Park u. a., 2016) and matrix completion (Ge u. a., 2016, 2017), it was shown that all local minima are globally optimal. Similar results were also known for special cases of tensor CP decomposition (Ge u. a., 2015).

In this chapter, we prove similar results for Tucker decomposition. Given a tensor $T \in \mathbb{R}^{d\times d\times d}$ with multilinear rank (r, r, r), the Tucker decomposition of the tensor T has the form

$$T = S^*(A^*, B^*, C^*),$$

where $S^* \in \mathbb{R}^{r\times r\times r}$ is a core tensor, $A^*, B^*, C^* \in \mathbb{R}^{r\times d}$ are three components (factor matrices).

To find a Tucker decomposition by local search, the most straight-forward idea is to directly optimize the following nonconvex objective:

$$L(S, A, B, C) = \|T - S(A, B, C)\|_F^2.$$

Clearly, (S^*, A^*, B^*, C^*) is a global minimizer. However, since the optimization problem is nonconvex, it is unclear whether any local search algorithm can efficiently find a globally optimal solution. Our first result (Theorem 2) shows that with an appropriate regularizer (designed in Section 4.2), all local minima of Tucker decomposition are globally optimal.

The main difficulty of analyzing the optimization landscape of Tucker decomposition comes from the existence of *high order saddle points*. For example, when S, A, B, C are all equal to 0, any local movement of norm ϵ will only change the objective by at

most $O(\epsilon^4)$. Characterizing the possible locations of such high order saddle points, and showing that they cannot become local minima is one of the major technical contributions of this chapter.

In general, even if all local minima are globally optimal, a local search algorithm may still fail to find a global optimal solution due to high order saddle points. In the worst case it is known that 3rd order saddle points can be handled efficiently, while 4th order saddle point are hard to escape from (Anandkumar und Ge, 2016). The objective L has 4th order saddle points. However, our next result (Theorem 3) shows that there is a specifically designed local search algorithm that can find an approximate global optimal solution in polynomial time.

4.2 Optimization Problem

For simplicity, we assume $r_1 = r_2 = r_3 = r$, and $d_1 = d_2 = d_3 = d$. It is easy to generalize the result to the case with different r_i's and d_i's. Let $T \in \mathbb{R}^{d \times d \times d}$ be a fixed third order tensor with Tucker rank $r < d$. A simple objective for tensor decomposition can be defined as:

$$L(S, A, B, C) = \|S(A, B, C) - T\|_F^2. \tag{4.1}$$

Suppose $T = S^*(A^*, B^*, C^*)$, then Equation (4.1) has a global minimum at (S^*, A^*, B^*, C^*) with the minimum possible L value 0. In fact, due to symmetry, we know there are many more global minimizers of L: for any invertible matrices $Q_A, Q_B, Q_C \in \mathbb{R}^{r \times r}$, let $S = S^*(Q_A, Q_B, Q_C)$, and $A = {Q_A}^{-1}A^*$, $B = {Q_B}^{-1}B^*$ and $C = {Q_C}^{-1}C^*$, then we also have $T = S(A, B, C)$. Therefore, the loss L has infinitely many global optimal solutions.

The existence of many equivalent global optimal solutions causes problems for local search algorithms, especially simpler ones like gradient descent. The reason is

that if we scale A, B, C with a large constant c, and scale S with $1/c^3$, the tensor $S(A, B, C)$ does not change. However, after this scaling the partial gradient of S is multiplied by c^3, while the partial gradients of A, B, C are multiplied by $1/c$. When c is large one has to choose a very small step size for gradient descent, and this results in very slow convergence.

We address the problem of scaling by introducing a regularizer $l(S, A, B, C)$ given by

$$\|AA^\top - S_{(1)}S_{(1)}^\top\|_F^2 + \|BB^\top - S_{(2)}S_{(2)}^\top\|_F^2 + \|CC^\top - S_{(3)}S_{(3)}^\top\|_F^2. \tag{4.2}$$

Intuitively, the three terms in the regularizer ensure that A and S (similarly, B, C and S) have similar norms. Similar regularizers were used for analyzing the optimization landscape of asymmetric matrix problems(Park u. a., 2016), where the same scaling problem exists. However, to the best of our knowledge we have not seen this regularizer used for Tucker decomposition.

For technical reasons that will become clear in Section 4.3 (especially in Lemma 13), we actually use $R(S, A, B, C) = l(S, A, B, C)^2$ as the regularizer with weight $\lambda > 0$, so the final optimization problem we consider is:

$$\min_{S,A,B,C} L(S, A, B, C) + \lambda R(S, A, B, C). \tag{4.3}$$

Note that even for Equation (4.3), there are still infinitely many global minimizers. In particular, one can rotate A and S (similarly, B, C and S) simultaneously to get equivalent solutions. A priori it is unclear whether there always exists a global minimizer that achieves 0 loss for Equation (4.3). Our proof in Section 4.3 implicitly shows that such a solution must exist.

The space of parameters for our objective function is $\mathbb{R}^{r\times r\times r} \times \mathbb{R}^{r\times d} \times \mathbb{R}^{r\times d} \times \mathbb{R}^{r\times d}$. We write a point in this space as (S, A, B, C), and equip it with inner product

$\langle (S, A, B, C), (S', A', B', C') \rangle = \langle S, S' \rangle + \langle A, A' \rangle + \langle B, B' \rangle + \langle C, C' \rangle$ and associated norm

$$\|(S, A, B, C)\|_F = \sqrt{\|S\|_F^2 + \|A\|_F^2 + \|B\|_F^2 + \|C\|_F^2}.$$

4.3 Characterization of Optimization Landscape

In this section, we analyze the optimization landscape for the objective (4.3) for Tucker decomposition. In particular, we establish the following result.

Theorem 2. *For any fixed $\lambda > 0$, all local minima of the objective function $f = L + \lambda R$ as in Equation* (4.3) *have loss 0.*

Note that the theorem would not hold for $\lambda = 0$ (when there is no regularizer). A counter-example is when $T = a^* \otimes b^* \otimes c^*$ for some unit vectors a^*, b^*, c^*, and $S = 0, A = a^\top, B = b^\top, C = c^\top$ where a, b, c are unit vectors that are orthogonal to a^*, b^*, c^* respectively. A local change will have no effect if the new S is still 0, and will make the objective function larger if the new S is nonzero.

In order to prove this theorem, we demonstrate a direction of improvement for all points (S, A, B, C) that don't achieve the global optimum. A direction of improvement is a tuple $(\Delta S, \Delta A, \Delta B, \Delta C)$ such that

$$f(S + \epsilon \Delta S, A + \epsilon \Delta A, B + \epsilon \Delta B, C + \epsilon \Delta C) < f(S, A, B, C)$$

for all sufficiently small $\epsilon > 0$. Clearly, if a point (S, A, B, C) has a direction of improvement, then it cannot be a local minimum.

Throughout the section, let P_1 (P_2, P_3) be the projection onto the column span of $T_{(1)}$ ($T_{(2)}, T_{(3)}$). Let $A_1 = AP_1$ and $A_2 = A(I - P_1)$ (similarly for B, C). The proof

works in the following 4 steps:

Bounding the regularizer First we show that when $\nabla f = 0$, the regularizer R must be equal to 0 (Lemm 10 in Section 4.3.1). At a high level, this is because the gradient of regularizer R is always orthogonal to the gradient of main term L. Therefore if the gradient of the entire objective is 0, the gradient of R must also be 0. We complete the proof by showing that $\nabla R = 0$ implies $R = 0$.

Removing extraneous directions Next, we show that when $\nabla f = 0$, the projection in the wrong subspaces A_2, B_2, C_2 are all equal to 0. This is because the direction of directly removing the projection in the wrong subspace A_2 is a direction of improvement (see Lemma 14).

Adding missing directions After the previous steps, we know that the rows of A are in the column span of $T_{(1)}$. However, the row span of A might be smaller. In this case, there exist directions a, b, c such that $T(a, b, c) > 0$, and $Aa = 0$. We will show that in this case we can always add the missing directions into A and S. This is the most technical part of our proof, and high order stationary points may appear when Bb or Cc are also 0. See Sections 4.3.3.

Fixing S Finally, we know that the components A, B, C must span the correct subspaces. Our final step shows that in this case, if $L > 0$ then it is easy to find a direction of improvement, see Section 4.3.4.

4.3.1 Points With Nonzero Regularizer

We show any point with nonzero regularizer must also have a nonzero gradient, therefore the (negative) gradient itself is a direction of improvement.

Lemma 10. *For any S, A, B, C, if $R(S, A, B, C) > 0$ then $\|\nabla f\| > 0$.*

To prove this, we first show that if the regularizer is nonzero, then its gradient is nonzero.

Lemma 11. *The function l satisfies*

$$4l(S, A, B, C) = \langle \nabla_A l, A\rangle + \langle \nabla_B l, B\rangle + \langle \nabla_C l, C\rangle + \langle \nabla_S l, S\rangle$$

Proof. Note the following calculations:

$$\begin{aligned}
\langle \nabla_A l, A\rangle &= \langle 4(AA^\top - S_{(1)}S_{(1)}^\top)A, A\rangle \\
&= 4\langle AA^\top - S_{(1)}S_{(1)}^\top, AA^\top\rangle \\
\langle 4S(S_{(1)}S_{(1)}^\top - AA^\top, I, I), S\rangle &= -4\langle AA^\top - S_{(1)}S_{(1)}^\top, S_{(1)}S_{(1)}^\top\rangle
\end{aligned}$$

The left-hand side above is one of the terms in $\nabla_S l$. Doing the same calculation for the other modes and then adding everything together yields the result. □

We next show that the gradient of the regularizer is always orthogonal to the gradient of the main term (i.e. the tensor loss L).

Lemma 12. *For any S, A, B, C, $\langle \nabla L(S, A, B, C), \nabla R(S, A, B, C)\rangle = 0$.*

Proof. We start by calculating the partial gradients for L and r. We have

$$\begin{aligned}
&\nabla_A L = 2S_{(1)}(B \otimes C)(S(A, B, C) - T)_{(1)}^\top \quad \nabla_A l = 4(AA^\top - S_{(1)}S_{(1)}^\top)A \\
&\nabla_B L = 2S_{(2)}(A \otimes C)(S(A, B, C) - T)_{(2)}^\top \quad \nabla_B l = 4(BB^\top - S_{(2)}S_{(2)}^\top)B \\
&\nabla_C L = 2S_{(3)}(A \otimes B)(S(A, B, C) - T)_{(3)}^\top \quad \nabla_C l = 4(CC^\top - S_{(3)}S_{(3)}^\top)C \\
&\nabla_S L = 2(S(A, B, C) - T)(A^\top, B^\top, C^\top) \\
&\nabla_S l = 4S(S_{(1)}S_{(1)}^\top - AA^\top, I, I) + 4S(I, S_{(2)}S_{(2)}^\top - BB^\top, I) \\
&\qquad + 4S(I, I, S_{(3)}S_{(3)}^\top - CC^\top)
\end{aligned}$$

We now compute the following:

$$\begin{aligned}\langle \nabla_A L, \nabla_A l\rangle &= 8\langle S_{(1)}(B\otimes C)(S(A,B,C)-T)_{(1)}^\top, (AA^\top - S_{(1)}S_{(1)}^\top)A\rangle \\ &= 8\langle (S(A,B,C)-T)(A^\top, B^\top, C^\top), S(AA^\top - S_{(1)}S_{(1)}^\top, I, I)\rangle\end{aligned}$$

From here it is easy to see that $\langle \nabla_S L, \nabla_S l\rangle = -\langle \nabla_A L, \nabla_A l\rangle - \langle \nabla_B L, \nabla_B l\rangle - \langle \nabla_C L, \nabla_C l\rangle$, therefore $\langle \nabla L, \nabla l\rangle = 0$. Since $\nabla R = 2l\nabla l$, the result follows. □

Now we are ready to prove Lemma 10:

Proof. By Lemma 12, we know $\|\nabla f\|_F^2 = \|\nabla L\|_F^2 + \|\nabla R\|_F^2$. On the other hand, by Lemma 11 and an application of the Cauchy-Schwarz inequality, we see that

$$\|\nabla l\|_F \|(S,A,B,C)\|_F \geq 4l(S,A,B,C),$$

which means that $\|\nabla l\|_F > 0$ whenever $R = l^2 > 0$. But $\nabla R = 2l\nabla l$, so we have that $\|\nabla R\|_F > 0$, whence $\nabla f \neq 0$. □

To facilitate later proofs, we will also show a fact that if one perturbs a solution with 0 regularizer, then the regularizer remains very small.

Lemma 13. *If $R = 0$, and $\|\Delta A\|_F + \|\Delta B\|_F + \|\Delta C\|_F + \|\Delta S\|_F \leq O(1)$, then $R(S+\epsilon\Delta S, A+\epsilon\Delta A, B+\epsilon\Delta B, C+\epsilon\Delta C) = O(\epsilon^4)$ for sufficiently small ϵ.*

Proof. It suffices to check that the term $\|(A+\epsilon\Delta A)(A+\epsilon\Delta A)^\top - (S+\epsilon\Delta S)_{(1)}(S+\epsilon\Delta S)_{(1)}^\top\|_F = O(\epsilon)$, as other terms are symmetric, and the final R is degree 4 over these terms. This is clear as we know $\|AA^\top - S_{(1)}S_{(1)}^\top\|_F = 0$ because $R = 0$, and all the remaining terms are bounded by $O(\epsilon)$. □

4.3.2 Removing Extraneous Directions

In this section, we show that if A (respectively B, C) has a direction in its row-space that is perpendicular to the column-space of $T_{(1)}$ (respectively $T_{(2)}$, $T_{(3)}$), then we have a direction of improvement. In particular, our goal is to show $A_2 = 0$ for all local minima (symmetric arguments will then show $B_2 = C_2 = 0$). We first show that $S(A_2, B, C) = 0$.

Lemma 14. *Assume that $R(S, A, B, C) = 0$. If $S(A_2, B, C) \neq 0$, then $\Delta A = -A_2$ is a direction of improvement.*

Proof. Set $\Delta A = -A_2$. Then for $\epsilon > 0$

$$\begin{aligned} L(S, A + \epsilon\Delta A, B, C) &= \|S(A, B, C) - T + \epsilon S(\Delta A, B, C)\|_F^2 \\ &= L(S, A, B, C) - 2\epsilon\|S(A_2, B, C)\|_F^2 + O(\epsilon^2), \end{aligned}$$

since $\langle S(A, B, C) - T, S(A_2, B, C)\rangle = \langle S(A_2, B, C), S(A_2, B, C)\rangle$. Hence, for all sufficiently small ϵ, $L(S, A + \epsilon\Delta A, B, C) < L(S, A, B, C)$. By Lemma 13 we know $R(S, A + \epsilon\Delta A, B, C) = O(\epsilon^4)$. Hence, for sufficiently small ϵ, the decrease in L will exceed any increase in R. This shows that ΔA is a direction of improvement. □

We next establish that $R(S, A, B, C) = 0$ and $S(A_2, B, C) = 0$ together imply that $A_2 = 0$.

Lemma 15. *If $R(S, A, B, C) = 0$ and $S(A_2, B, C) = 0$, then $A_2 = 0$.*

Proof. Since $R(S, A, B, C) = 0$, we have $BB^\top = S_{(2)}S_{(2)}^\top$ and $CC^\top = S_{(3)}S_{(3)}^\top$. This means the column span of $S_{(2)}$ ($S_{(3)}$) is the same as column span of B (C). Let B^+ and C^+ denote the pseudoinverses of B and C. Note that the orthogonal projections onto the column-space of B and C are given by $P_B := BB^+$ and $P_C := CC^+$, respectively.

Using these facts along with $S(A_2, B, C) = 0$, we have

$$0 = S(A_2, B, C)(I, B^+, C^+) = S(A_2, P_B, P_C) = S(A_2, I, I).$$

Using the fact that $S_{(1)}S_{(1)}^\top = AA^\top = A_1A_1^\top + A_2A_2^\top$, we have

$$\|A_2A_2^\top\|_F^2 \leq \langle A_2A_2^\top, S_{(1)}S_{(1)}^\top \rangle = \|S(A_2, I, I)\|_F^2 = 0,$$

which, in particular, means that $A_2 = 0$. □

4.3.3 Adding Missing Directions

We now consider the case where the row-spans of A, B, and C are not equal to the column-spans of $T_{(1)}$,$T_{(2)}$, and $T_{(3)}$, respectively. Again by symmetry, we focus on the case when row-span of A is not equal to column-span of $T_{(1)}$.

Lemma 16. *If the row-span of A is a strict subset of the column-span of $T_{(1)}$ and $R = 0$, then there is a direction of improvement.*

Proof. If the row-span of A is a strict subset of column-span of $T_{(1)}$, we must have a vector a that is in the column-span of $T_{(1)}$, but $Aa = 0$. For this vector we know $T(a, I, I) \neq 0$, therefore there must exist vectors b, c such that $T(a, b, c) > 0$. This is true even if we restrict b to be either in the row span of B or to satisfy $Bb = 0$ (and similarly for c), as we can partition the matrix into 4 subspaces based on the projections of its columns to row span of B (and its rows to row span of C). In particular, if we let b_1 be the projection of b onto the row-span of B and c_1 be the projection of c onto the row-span of C, and set $b_2 = b - b_1$ and $c_2 = c - c_1$, then we have $T(a, b, c) = \sum_{i,j \in \{1,2\}} T(a, b_i, c_j)$. Hence, $T(a, b_i, c_j) > 0$ for some choice of

$i, j \in \{1, 2\}$.

One missing direction In this case b and c are in row span of B, C respectively. Choose unit vectors $u, v, w \in \mathbb{R}^r$ such that $A^\top u = 0$, $B^\top v = \alpha_1 b$, and $C^\top w = \alpha_2 c$, where α_1 and α_2 are positive real numbers. Consider the directions $\Delta A = ua^\top$, $\Delta S = u \otimes v \otimes w$. Observe that $\Delta S(A, B, C) = A^\top u \otimes B^\top v \otimes C^\top w = 0$ and $S(\Delta A, B, C) = 0$ since the column-space of $S_{(1)}$ is equal to the column-space of A. Moreover, $\Delta S(\Delta A, B, C) = a \otimes B^\top v \otimes C^\top w = \alpha_1 \alpha_2 a \otimes b \otimes c$. Hence, for $\epsilon > 0$, we have

$$\begin{aligned} L(S + \epsilon \Delta S, A + \epsilon \Delta A, B, C) &= \|S(A, B, C) - T + \epsilon^2 \Delta S(\Delta A, B, C)\|_F^2 \\ &= L(S, A, B, C) - 2\epsilon^2 \alpha_1 \alpha_2 T(a, b, c) + O(\epsilon^4). \end{aligned}$$

On the other hand, by Lemma 13 $R(S+\epsilon\Delta S, A+\epsilon\Delta A, B, C) = O(\epsilon^4)$ since $R(S, A, B, C) = 0$. Hence, for small enough ϵ, the improvement in the tensor loss dominates all other perturbations, so we have a direction of improvement.

Two missing directions Now assume that $Aa = Bb = 0$, and c is in the row span of C. Choose unit vectors $u, v, w \in \mathbb{R}^r$ such that $A^\top u = B^\top v = 0$ and $C^\top w = \alpha c$ where $\alpha > 0$. Consider the directions $\Delta A = ua^\top$, $\Delta B = vb^\top$, $\Delta S = u \otimes v \otimes w$. Through a very similar calculation as in the previous case,

$$L(S + \epsilon \Delta S, A + \epsilon \Delta A, B + \epsilon \Delta B, C) = L(S, A, B, C) - 2\epsilon^3 \alpha T(a, b, c) + \epsilon^6 \alpha^2.$$

As before, by Lemma 13 $R(S + \epsilon\Delta S, A + \epsilon\Delta A, B + \epsilon\Delta B, C) = O(\epsilon^4)$. Hence, the decrease in the tensor loss dominates all other perturbations for sufficiently small ϵ, and so this is a direction of improvement. Note that in this case the amount of improvement is $\Theta(\epsilon^3)$, so the point is a 3rd order saddle point.

The case where $Cc = 0$ and b is in the row-span of B is similar, and likewise yields a direction of improvement.

Three missing directions Now assume that $Aa = Bb = Cc = 0$, and choose unit vectors $u, v, w \in \mathbb{R}^r$ such that $A^\top a = B^\top v = C^\top w = 0$. Consider the directions $\Delta A = ua^\top$, $\Delta B = vb^\top$, $\Delta C = wc^\top$, and $\Delta S = u \otimes v \otimes w$. Once again, most perturbations in the tensor loss vanish, and we have

$$L(S + \epsilon\Delta S, A + \epsilon\Delta A, B + \epsilon\Delta B, C + \epsilon\Delta C) = L(S, A, B, C) - 2\epsilon^4 T(a, b, c) + \epsilon^8.$$

In this case, the regularizer doesn't change at all, since $\Delta S_{(i)} S_{(i)}^\top = 0$ for $i = 1, 2, 3$, $\Delta A A^\top = \Delta B B^\top = \Delta C C^\top = 0$, and $\Delta A \Delta A^\top - \Delta S_{(1)} \Delta S_{(1)}^\top = uu^\top - uu^\top = 0$ (and the two other analogous terms likewise vanish). Hence, for sufficiently small ϵ, the objective function decreases, so this is a direction of improvement. This point is a 4th order saddle point.

□

4.3.4 Improving the Core Tensor

We finally consider the case where the matrices A, B, C have the correct row-spaces but $S(A, B, C) \neq T$. In this situation, we can make progress by changing only S.

Lemma 17. *If $R = 0$, row spans of A, B, C are equal to column span of $T_{(1)}, T_{(2)}, T_{(3)}$ respectively, but $L > 0$, then there exists a direction of improvement.*

Proof. Since the spans of A, B, C are already correct, let A^+ be the pseudoinverse of A, then if we let $S' = T(A^+, B^+, C^+)$, we have $S'(A, B, C) = T$. Consider the

direction $\Delta S = S' - S$.

$$\begin{aligned} L(S + \epsilon\Delta S, A, B, C) &= \|(1-\epsilon)S(A,B,C) - (1-\epsilon)T\|_F^2 \\ &= (1-\epsilon)^2 L(S, A, B, C). \end{aligned}$$

For the regularizer R, again by Lemma 13 we have $R(S + \epsilon\Delta S, A, B, C) = O(\epsilon^4)$. Hence, this is a direction of improvement. □

4.3.5 Proof of Main Theorem

Now with all the lemmas we are ready to prove the main theorem:

Proof of Theorem 2. The Theorem follows immediately from the sequence of lemmas.

First, by Lemma 10, we know any local minima must satisfy $R = 0$. Next, by Lemma 15 and Lemma 14, we know the row spans of A, B, C must be subsets of column spans of $T_{(1)}, T_{(2)}, T_{(3)}$ respectively. In the third step, by Lemma 16, we further show that the row spans of A, B, C must be exactly equal to column spans of $T_{(1)}, T_{(2)}, T_{(3)}$ respectively. Finally, by Lemma 17 we know the loss function must be equal to 0. □

4.4 Escaping from High Order Saddle Points

As we discussed before, since our objective $f = L + \lambda R$ as in (4.3) may have high order saddle points, standard local search algorithms may not be able to find a local minimum. However, in this section we show that the high order saddle points of f are *benign*: there is a polynomial time local search algorithm that can find an approximate local and global minimum of f.

We will first review the guarantees of standard local search algorithms, and then describe how to escape from high order saddle points.

4.4.1 Second Order Stationary Points

For a general function $f(x)$ whose first two derivatives exist, we say a point x is a (τ_1, τ_2)-second order stationary point if

$$\|\nabla f(x)\| \leq \tau_1, \quad \lambda_{min}(\nabla^2 f(x)) \geq -\tau_2.$$

If the function $f(x)$ satisfies the gradient and Hessian Lipschitz conditions

$$\begin{aligned} \forall x, y \quad & \|\nabla f(x) - \nabla f(y)\| \leq \rho_1 \|x - y\|_2, \\ \forall x, y \quad & \|\nabla^2 f(x) - \nabla^2 f(y)\| \leq \rho_2 \|x - y\|_2, \end{aligned}$$

there are many local search algorithms that can find (τ_1, τ_2)-second order stationary points in polynomial time. This includes traditional second order algorithms such as cubic regularization (Nesterov und Polyak, 2006), and more recently first order algorithms such as perturbed gradient descent (Jin u. a., 2017).

Of course, these guarantees are not enough for our objective f, as it has higher order saddle points. The main theorem in this section shows that there is an efficient local search algorithm that can optimize f.

Theorem 3. *Let $\lambda = 1/16r^4$, assume wlog. that $\|T\|_F = 1$ and the initial point satisfies $f = L + \lambda R = O(1)$[1]. Then there is a local search algorithm that in poly$(d, r, 1/\epsilon)$ time finds a point (S, A, B, C) such that $f(S, A, B, C) \leq \epsilon$.*

The algorithm that we will design is just a proof of concept: although its running

[1]This can be achieved by initializing at 0, or any point with norm $O(1)$.

Symbol	Definition	Note
λ	$\frac{1}{16r^4}$	weight for regularizer
K	$O^*(1)$	universal bound for norms of S, A, B, C, T
τ	< 1	bound on $R(S, A, B, C)$
γ	$\Theta^*(\tau^{1/48})$	bound on the norm of A_3, B_3, C_3, introduced in Lemma 23
σ	$\sqrt{\gamma}$	singular value threshold for A_1, B_1, C_1
κ_0	$\sqrt{\gamma}$	max error in $T_{1,1,1}$, introduced in Lemma 26
κ_1	$2K\sigma^{3/4}$	max error in $T_{2,1,1}$, introduced in Lemma 27
κ_2	$2K\sigma^{1/8}$	max error in $T_{2,2,1}$, introduced in Lemma 29
κ_3	$2K\sigma^{1/2}$	max error in $T_{2,2,2}$, introduced in Lemma 30

Table 4.1: Notation and definitions used in Section 4.4

time is polynomial, it is far from practical. We have not attempted to improve the dependencies on $d, r, 1/\epsilon$. Local search algorithms seem to perform much better for Tucker decomposition in practice, and understanding that is an interesting open problem.

To prove Theorem 3, we will first show that sublevel sets of f are all bounded (Section 4.4.2). This allows us to bound the gradient and Hessian Lipschitz constants ρ_1 and ρ_2, so we can use any of the previous local search algorithm to find a (τ_1, τ_2)-second order stationary point.

Next, we follow the steps of Theorem 2, but we do it much more carefully to show that as long as the objective is larger than ϵ, then either the point has a large gradient or a negative eigenvalue in Hessian, or there is a way to construct a direction of improvement. This is captured in our main Lemma 21.

Finally we give a sketch of the algorithm and show that these local improvements are enough to guarantee the convergence in Section 4.4.8.

Throughout the section, we use $O^*(\cdot)$, $\Omega^*(\cdot)$ and $\Theta^*(\cdot)$ to hide polynomial factors of r and d. We will introduce several numerical quantities in the remainder of this section; we list the most important ones in Table 4.1.

4.4.2 Bounded Sublevel Sets

We first establish the boundedness of sublevel sets of the objective function. Our local search algorithm will guarantee the function value decreases in every iteration, so the trajectory of the algorithm will remain in a sublevel set. As a result, we know that the parameters remain bounded in norm at each step by some constant, say K.

Lemma 18. *For all $\Gamma \geq 0$, the set of points (S, A, B, C) with $f(S, A, B, C) \leq \Gamma$ satisfy that $\|S\|_F, \|A\|_F, \|B\|_F, \|C\|_F \leq K$ where $K = O^*((\Gamma + 1)^{1/8})$.*

To prove this lemma, we will first state some tools that we need.

Lemma 19. *For any parameter tuple (S, A, B, C), we have*

$$\|S(A, B, C)\|_F \leq \|S\|_F \|A\|_2 \|B\|_2 \|C\|_2.$$

Proof. This follows from the fact that $\|\cdot\|_F$ is invariant to matricization, and the fact that $\|PQ\|_F \leq \|P\|_2 \|Q\|_F$. Observe that

$$\|S(A, B, C)\|_F = \|A^\top S(I, B, C)_{(1)}\|_F \leq \|A\|_2 \|S(I, B, C)\|_F,$$

and then repeat the argument for the other modes. □

Lemma 20. *For any $S \in \mathbb{R}^{r \times r \times r}$, it holds that*

$$\|S(S_{(1)}, S_{(2)}, S_{(3)})\|_F \geq \frac{1}{r^4} \|S\|_F^4$$

Proof. Let $u, v, w \in \mathbb{R}^r$ be unit vectors such that $S(u, v, w) = \|S\|_2$. Then

$$\begin{aligned} S_{(3)}\text{vec}(u \otimes v) &= S(u, v, I) = \|S\|_2 w, \\ S_{(2)}\text{vec}(u \otimes w) &= S(u, I, w) = \|S\|_2 v, \\ S_{(1)}\text{vec}(v \otimes w) &= S(I, v, w) = \|S\|_2 u. \end{aligned}$$

Then

$$\|S(S_{(1)}, S_{(2)}, S_{(3)})\|_2 \geq S(\|S\|_2 u, \|S\|_2 v, \|S\|_2 w) = \|S\|_2^3 S(u, v, w) = \|S\|_2^4$$

The result then follows from the norm inequality $\|S\|_F \leq r\|S\|_2$. □

Now we are ready to prove Lemma 18:

Proof of Lemma 18. Assume that $\Gamma \geq f(S, A, B, C)$. From L, we have

$$\begin{aligned} \sqrt{\Gamma} &\geq \|S(A, B, C) - T\|_F \\ &\geq \|S(A, B, C)\|_F - \|T\|_F, \end{aligned}$$

so that $\|S(A, B, C)\|_F \leq \sqrt{\Gamma} + \|T\|_F$. Next, define the following for $i = 1, 2, 3$: $d_i(X, S) = XX^\top - S_{(i)}S_{(i)}^\top$. Note that $d_1(A, S), d_2(B, S), d_3(C, S)$ are each bounded above in norm by $\Gamma^{1/4}$. We have

$$\begin{aligned} \|S(A, B, C)\|_F^2 &= \langle S(A, B, C), S(A, B, C)\rangle \\ &= \langle S(AA^\top, BB^\top, CC^\top), S\rangle \\ &= \langle S(S_{(1)}S_{(1)}^\top, S_{(2)}S_{(2)}^\top, S_{(3)}S_{(3)}^\top), S\rangle + g(S, A, B, C) \\ &= \|S(S_{(1)}, S_{(2)}, S_{(3)})\|_F^2 + g(S, A, B, C), \end{aligned}$$

where $g(S, A, B, C)$ is a sum of the seven remainder terms of the form

$$\langle S(d_1(A,S), S_{(2)}S_{(2)}^\top, S_{(3)}S_{(3)}^\top), S\rangle \tag{4.4}$$

$$\langle S(d_1(A,S), d_2(B,S), S_{(3)}S_{(3)}^\top), S\rangle \tag{4.5}$$

$$\langle S(d_1(A,S), d_2(B,S), d_3(C,S)), S\rangle \tag{4.6}$$

There are three terms of type (4.4), and each can be bounded below using Cauchy-Schwarz and Lemma 19 as follows:

$$\langle S(d_1(A,S), S_{(2)}S_{(2)}^\top, S_{(3)}S_{(3)}^\top), S\rangle \geq -\|S\|_F^6 \|d_1(A,S)\|_F \geq -\Gamma^{1/4}\|S\|_F^6.$$

Similarly, we have

$$\langle S(d_1(A,S), d_2(B,S), S_{(3)}S_{(3)}^\top), S\rangle \geq -\Gamma^{1/2}\|S\|_F^4,$$
$$\langle S(d_1(A,S), d_2(B,S), d_3(C,S)), S\rangle \geq -\Gamma^{3/4}\|S\|_F^2.$$

Putting this together and applying Lemma 20, we have that

$$\frac{1}{r^8}\|S\|_F^8 - 3\Gamma^{1/4}\|S\|_F^6 - 3\Gamma^{1/2}\|S\|_F^4 - \Gamma^{3/4}\|S\|_F^2 \leq (\sqrt{\Gamma} + \|T\|_F)^2,$$

which means that $\|S\|_F$ must be bounded by $O^*((\Gamma+1)^{1/8})$. From R, we have

$$\left(\frac{\Gamma}{\lambda}\right)^{1/4} + \|S\|_F^2 \geq \|AA^\top - S_{(1)}S_{(1)}^\top\|_F + \|S_{(1)}S_{(1)}^\top\|_F \geq \|AA^\top\|_F,$$

so $\|A\|_F$ is bounded by $O^*((\Gamma+1)^{1/8})$. We bound B and C similarly. □

4.4.3 Main Step: Making Local Improvements

In order to prove Theorem 3, we rely on the following main lemma:

Lemma 21. *In the same setting as Theorem 3, there exist positive constants* $q_1, q_2, \tau_1 = \Theta^*(\epsilon^{q_1})$, $\tau_2 = \Theta^*(\epsilon^{q_2})$, *such that for any point* S, A, B, C *where* $\epsilon < f(S, A, B, C) < O(1)$, *one of the following is true:*

1. $\|\nabla f(S, A, B, C)\| \geq \tau_1$,
2. $\lambda_{min}(\nabla^2 f(S, A, B, C)) \leq -\tau_2$,
3. *With constant probability, Algorithm 1 constructs a direction of improvement that improves the function value by poly*(ϵ).

Algorithm 1 uses notation that we specify in the paragraphs below. The proof

Algorithm 1 Sampling algorithm for adding missing directions

Require: matrices A, B, C, threshold σ, subspace indicator $(i, j, k) \in \{1, 2\}^3$
Compute the subspaces $U_{1,i}, U_{2,j}, U_{3,k}, V_{1,i}, V_{2,j}, V_{3,k}$
Sample unit vectors a, b, c uniformly from $U_{1,i}, U_{2,j}, U_{3,k}$
if $i = 1$ **then** $u' = (A_1^\top)^+ a$; **else** Randomly sample nonzero $u' \in V_{1,2}$
if $j = 1$ **then** $v' = (B_1^\top)^+ b$; **else** Randomly sample nonzero $v' \in V_{2,2}$
if $k = 1$ **then** $w' = (C_1^\top)^+ c$; **else** Randomly sample nonzero $w' \in V_{3,2}$
Return $a, b, c, u'/\|u'\|_2, v'/\|v'\|_2, w'/\|w'\|_2$

of this lemma has similar steps to the proof of Theorem 2. However, it is more complicated because we are not looking at exact local minima. We give the details of these steps in the following subsections. A key parameter that we will use is a bound τ on the regularizer. We will consider different cases when $R(S, A, B, C) \geq \tau$ and when $R(S, A, B, C) \leq \tau$. All of our other parameters (including τ_1, τ_2, ϵ) will be polynomials in τ.

For the analysis, it is useful to consider $S(A, B, C)$ and T projected onto various subspaces of interest. To this end, we introduce the following notation. Let $\sigma > 0$

be a threshold that we will specify later. For matrix A, we let $V_{1,1}$ and $U_{1,1}$ denote the spaces spanned by the left/right singular vectors of A with singular value greater than σ, and let $V_{1,2} = V_{1,1}^{\perp}$, $U_{1,2} = U_{1,1}^{\perp}$. We can then write $A = A_1 + A_2$, where $A_1 = \text{Proj}_{V_{1,1}} A$ contains the larger singular vectors and $A_2 = \text{Proj}_{V_{1,2}} A$ contains the smaller singular vectors. Let P_1 be the orthogonal projection onto the column-space of $T_{(1)}$ and define $A_3 = A(I - P_1)$, the projection of A onto directions that are unrelated to the true tensor. Similarly, we define $U_{2,1}, U_{2,2}, V_{2,1}, V_{2,2}, P_2, B_1, B_2, B_3$ for matrix B and $U_{3,1}, U_{3,2}, V_{3,1}, V_{3,2}, P_3, C_1, C_2, C_3$ for matrix C. Define $S_{i,j,k} = S(\text{Proj}_{V_{1,i}}, \text{Proj}_{V_{2,j}}, \text{Proj}_{V_{3,k}})$ and $T_{i,j,k} = T(\text{Proj}_{U_{1,i}}, \text{Proj}_{U_{2,j}}, \text{Proj}_{U_{3,k}})$. We can decompose the tensor loss as

$$\|S(A, B, C) - T\|_F^2 = \sum_{i,j,k \in \{1,2\}} \|S_{i,j,k}(A_i, B_j, C_k) - T_{i,j,k}\|_F^2.$$

Our analysis shows how to decrease the objective function if the regularizer or any one of the terms in the right-hand sum is sufficiently large. In particular, after finding a second-order stationary point, the only terms in this sum that may be large are when at least two of i, j, k are equal to 2. In this case, Algorithm 1 can be used to make further progress toward a local minimum.

4.4.4 Decreasing the Regularizer

We first show if the regularizer is large, then the gradient is large. This is very similar to Lemma 10.

Lemma 22. *If $R(S, A, B, C) \geq \tau$, then $\|\nabla f(S, A, B, C)\|_F \geq 4\lambda\tau/K$.*

Proof. By assumption, $l(S, A, B, C) \geq \tau^{1/2}$, and we have$\|\nabla R\|_F = \|2l\nabla l\|_F \geq$

$2\tau^{1/2}\|\nabla l\|_F$. By Lemma 11 and the Cauchy-Schwarz inequality, we have that

$$\|\nabla l\|_F \geq \frac{1}{2K}\|\nabla l\|_F\|(S,A,B,C)\|_F \geq \frac{1}{2K}\langle \nabla l,(S,A,B,C)\rangle = \frac{2l}{K}.$$

Then $\|\nabla R\|_F \geq 4\tau/K$. Since $\|\nabla f\|_F^2 = \|\lambda\nabla R\|_F^2 + \|\nabla L\|_F^2$, we are done. □

4.4.5 Removing Extraneous Directions

We show that if the projection A_3 in the incorrect subspace is large, the gradient must be large so the point cannot be a local minimum.

Lemma 23. *Let $\gamma = \Theta^*(\tau^{1/48})$. If $R(S,A,B,C) < \tau$ and $\|A_3\|_F \geq \gamma$, then*

$$\|\nabla f(S,A,B,C)\|_F = \Omega^*(\tau^{1/6}).$$

Proof. Set $\gamma = C\tau^{1/48}$, where we choose C to be a constant such that

$$\gamma^2 \geq \max\left(r(\tau^{1/24}+\tau^{1/4}), r^4(4K^6\tau^{1/8}+K^4\tau^{3/8})\right).$$

This particular definition allows us to simplify inequality (4.10) below. Consider the direction $\Delta A = -A_3$. When we step in this direction, the first-order perturbation of $L(S,A+\epsilon\Delta A,B,C)$ is $-2\epsilon\|S(A_3,B,C)\|_F^2$ (a simple calculation). For the regularizer, observe that since $A_3 = A(I-P_3)$, we have $(A+\epsilon\Delta A)(A+\epsilon\Delta A)^\top = AA^\top - (2\epsilon - \epsilon^2)A_3A_3^\top$. Hence the first-order perturbation of R is

$$8\epsilon\lambda l(S,A,B,C)\langle AA^\top - S_{(1)}S_{(1)}^\top, A_3A_3^\top\rangle \leq 8\epsilon\lambda\tau^{3/4}\|A_3\|_F^2.$$

Intuitively, we will show that the first-order decrease in L is greater than the first-order increase in R, so that ΔA is aligned negatively with $\nabla_A f$.

First, through very similar arguments to those found in the proof of Lemmas 18 and 20, we have that

$$\|S(A_3, B, C)\|_F^2 \geq \|S(A_3, S_{(2)}, S_{(3)})\|_F^2 - 2\tau^{1/4}K^6 - \tau^{1/2}K^4 \tag{4.7}$$

and if we set u to be the top left singular vector of A_3,

$$\|S(A_3, S_{(2)}, S_{(3)})\|_F \geq \frac{1}{r^2}\|S(u, I, I)\|_F^3 \|A_3\|_F \tag{4.8}$$

$$\begin{aligned} \|S(u, I, I)\|_F^2 &= u^\top S_{(1)} S_{(1)}^\top u \\ &= u^\top AA^\top u + u^\top (S_{(1)}S_{(1)} - AA^\top)u \\ &\geq \frac{1}{r}\|A_3\|_F^2 - \tau^{1/4}. \end{aligned} \tag{4.9}$$

Combining inequalities (4.7), (4.8), and (4.9), we have

$$\|S(A_3, B, C)\|_F^2 \geq \frac{1}{r^4}\|A_3\|_F^2 \left(\frac{1}{r}\|A_3\|_F^2 - \tau^{1/4}\right)^3 - 2\tau^{1/4}K^6 - \tau^{1/2}K^4 \tag{4.10}$$

Using the assumption $\|A_3\|_F \geq \gamma$ and the choice of γ, we can simplify inequality (4.10) to $\|S(A_3, B, C)\|_F^2 \geq \frac{\tau^{1/8}}{2r^4}\|A_3\|_F^2$. Now using the fact that $\lambda = 1/16r^4$ and $\tau^{1/8} > \tau^{3/4}$, we have $\frac{\tau^{1/8}}{2r^4}\|A_3\|_F^2 > 8\lambda\tau^{3/4}\|A_3\|_F^2$. Thus, we see that the first-order decrease in L is greater than the first-order increase in R, and the overall first-order decrease in f is $\Omega^*(\tau^{1/8}\|A_3\|_F^2/2r^4)$. The Taylor expansion of f implies that $|\langle \Delta A, \nabla_A f\rangle| \geq \frac{\tau^{1/8}}{2r^4}\|A_3\|_F^2$, so that

$$\|\nabla f\|_F \geq |\langle \Delta A, \nabla_A f\rangle| / \|\Delta A\|_F = \Omega^*(\tau^{1/8}\gamma) = \Omega^*(\tau^{1/6}),$$

which provides the desired bound on $\|\nabla f\|_F$. □

From this point forward, set $\sigma = \sqrt{\gamma}$. An important consequence of the fact that

A_3 is small is that if $T_{2,1,1}$ is large enough, then A_1 must be rank deficient. This rank deficiency allows us to readily contruct a direction of improvement when we are near a saddle point corresponding to a single missing direction. This is also true when we are near saddle points corresponding to two or three missing directions; see section 4.4.7.

To prove the rank deficiency, we use subspace perturbation bounds. The technical tool we use here is Wedin's Theorem (Wedin, 1972; Stewart, 1998).

Theorem (Wedin's Theorem, adapted from Stewart (1998))**.** *Let $\tilde{A}, A, E \in \mathbb{R}^{d\times r}$ with $d \geq r$ and $\tilde{A} = A + E$. Write the singular value decompositions*

$$A = \begin{pmatrix} U_1 & U_2 \end{pmatrix} \begin{pmatrix} \Sigma \\ 0 \end{pmatrix} V^\top \qquad \tilde{A} = \begin{pmatrix} \tilde{U_1} & \tilde{U_2} \end{pmatrix} \begin{pmatrix} \tilde{\Sigma} \\ 0 \end{pmatrix} \tilde{V}^\top$$

Let Θ and Φ denote the matrices of principal angles between the column spans of $U_1, \tilde{U}_1$ and $V, \tilde{V}$, respectively. If there exists some $\delta > 0$ such that $\min \sigma(\tilde{\Sigma}) \geq \delta$, then

$$\sqrt{\|\sin\Theta\|_F^2 + \|\sin\Phi\|_F^2} \leq \frac{\sqrt{2}\|E\|_F}{\delta}.$$

Lemma 24. *Let $M \in \mathbb{R}^{r\times d}$, and let $M = M_1 + M_2$, where rank(M) = rank$(M_1) = r$. Let $P, P_1 \in \mathbb{R}^{d\times d}$ be the orthogonal projections onto the row spans of M and M_1, respectively. Let σ be the smallest nontrivial singular value of M. Then*

$$\|P - P_1\|_F \leq \frac{2\|M_2\|_F}{\sigma}.$$

Proof. This is a corollary of Wedin's Theorem. Set $A = M_1^\top$, $\tilde{A} = M^\top$, and $E = M_2^\top$. Note that A and $\tilde{A}$ have full row rank, so we have the SVDs

$$A = \begin{pmatrix} U_1 & U_2 \end{pmatrix} \begin{pmatrix} \Sigma \\ 0 \end{pmatrix} V^\top \qquad \tilde{A} = \begin{pmatrix} \tilde{U_1} & \tilde{U_2} \end{pmatrix} \begin{pmatrix} \tilde{\Sigma} \\ 0 \end{pmatrix} \tilde{V}^\top$$

where $V, \tilde{V}$ are $r \times r$ orthogonal matrices, $\Sigma, \tilde{\Sigma}$ are $r \times r$, $U_1, \tilde{U}_1$ are $d \times r$, and $U_2, \tilde{U}_2$ are $d \times (d-r)$. Since V and $\tilde{V}$ have the same column spans, we have that $\sin \Phi = 0$. Further, it is a fact that $\|P - P_1\|_F = \sqrt{2}\|\sin \Theta\|_F$. By assumption, $\min \sigma(\tilde{\Sigma}) = \sigma$. Then Wedin's Theorem states that

$$\sqrt{\|\sin \Theta\|_F^2 + \|\sin \Phi\|_F^2} \leq \frac{\sqrt{2}\|E\|_F}{\sigma},$$

and our result follows immediately. □

Lemma 25. *Let P be the orthogonal projection onto the row-span of A_1. If rank$(A_1) = r$ and $\|A_3\|_F \leq \gamma$, then $\|T(I - P, I, I)\|_F < 2K\sqrt{\gamma}$. In particular, if any of the $T_{1,j,k}$ $(j, k = 1, 2)$ is large, the rank of A_1 must be less than r.*

Proof. Recall we set $\sigma = \sqrt{\gamma}$. Let P_1 be the orthogonal projection onto the column-span of $T_{(1)}$. Write $A_{1,1} = A_1 P_1$, $A_{1,2} = A_1(I - P_1)$. Observe that $\|A_{1,2}\|_F \leq \|A_3\|_F \leq \gamma < \sigma$. Note that $\|A_1 - A_{1,1}\|_F = \|A_{1,2}\|_F < \sigma$, which means that $\text{rank}(A_{1,1}) = r$, since A_1 has distance at least σ to the closest lower-rank matrix.

Since $A_{1,1}$ has rank r, its rows form a basis for the column-span of $T_{(1)}$, and so P_1 is also the orthogonal projection onto the row-span of $A_{1,1}$. Then

$$\begin{aligned}
\|T(I - P, I, I)\|_F &= \|T(P_1 - P, I, I)\|_F \\
&\leq \|T\|_F \|P_1 - P\|_F \\
&\leq K \frac{2\|A_{1,2}\|_F}{\sigma} \\
&< 2K\sqrt{\gamma},
\end{aligned}$$

where the penultimate line follows from Lemma 24. □

4.4.6 Improving S

Unlike the proof of Theorem 2, we will first focus on the simple case of improving the core tensor S. Note that here we only try to make sure we get close to $T_{1,1,1}$ as the components A, B, C may still be missing directions.

Lemma 26. *Set $\kappa_0 = \sqrt{\gamma}$. Assume $R(S, A, B, C) < \tau$. Then*

$$\|T_{1,1,1} - S(A_1, B_1, C_1)\|_F > \kappa_0 \;\Rightarrow\; \|\nabla f(S, A, B, C)\|_F = \Omega(\gamma^{2.5}).$$

Proof. Define $S^* = T(A_1^+, B_1^+, C_1^+)$, so that $S^*(A_1, B_1, C_1) = T_{1,1,1}$. We consider the direction $\Delta S = S^* - S_{1,1,1}$. Observe that $\Delta S(A, B, C) = T_{1,1,1} - S(A_1, B_1, C_1)$. We can write

$$S(A, B, C) - T = \sum_{i,j,k \in \{1,2\}} S(A_i, B_j, C_k) - T_{i,j,k},$$

and this is a sum of mutually orthogonal tensors. Hence, the the first-order perturbation of $L(S + \epsilon\Delta S, A, B, C)$ is

$$2\langle S(A, B, C) - T, \Delta S(A, B, C)\rangle = -2\|\Delta S(A, B, C)\|_F^2.$$

The first-order perturbation in the regularizer $\langle \nabla_S R, \Delta S\rangle$ is bounded by $O(\tau^{3/4}\sigma^{-3}) = o(\sigma)$, since $\|S^*\|_F = O(\sigma^{-3})$. Therefore, the decrease in the tensor loss dominates all other first-order perturbations, so we have a viable direction of improvement. In particular, by moving in direction $\epsilon\Delta S$, we decrease the objective function by

$$\epsilon \cdot \Omega(\|T_{1,1,1} - S(A_1, B_2, C_1)\|_F^2) = \Omega(\epsilon\kappa_0^2).$$

The direction of movement has norm bounded by

$$\|T_{1,1,1}\|_F \|A_1^+\|_2 \|B_1^+\|_2 \|C_1^+\|_2 \leq K\sigma^{-3}.$$

By Cauchy-Schwarz, the gradient has norm at least $\Omega(\kappa_0^2) \times \sigma^3 = \Omega(\gamma^{5/2})$. □

4.4.7 Adding Missing Directions

Finally, we try to add missing directions to A, B, C. As before we separate the cases into missing 1, 2 and 3 directions. This first case (missing one direction) is easy as it is a normal saddle point with negative Hessian.

Lemma 27. *Set $\kappa_1 = 2K\sigma^{3/4}$. Assume $R(S, A, B, C) < \tau$ and $\|A_3\|_F$, $\|B_3\|_F$, and $\|C_3\|_F$ are all less than γ. If $\|T_{2,1,1}\|_2 \geq \kappa_1$, then $\nabla^2 f$ has a negative eigenvalue of at most $-\Omega(\sigma^{15/4})$.*

Proof. Since $\kappa_1 > 2K\sqrt{\gamma}$, by Lemma 25, we have $\text{rank}(A_1) < r$.

By assumption, there exist unit vectors $a \in U_{1,2}$, $b \in U_{2,1}$, and $c \in U_{3,1}$ such that $T(a, b, c) \geq \kappa_1$. Take unit vectors $u, v, w \in \mathbb{R}^r$ such that $A_1^\top u = 0$, $B_1^\top v = \alpha_1 b$, $B_2^\top v = 0$, $C_1^\top w = \alpha_2 c$, and $C_2^\top w = 0$, where $\alpha_i \geq \sigma$ for $i = 1, 2$. In this situation, we are near a second-order saddle point, so we seek to demonstrate a direction with sufficient negative curvature in the objective function. To this end, define

$$\Delta A = \sigma u a^\top \qquad \Delta S = u \otimes v \otimes w.$$

For a step size $\epsilon > 0$, our source of improvement in the tensor loss comes from the second-order perturbation of L in this direction. We aim to compare the second-order decrease in L against the second-order increases in L and R. The second-order

perturbation in the tensor loss $\nabla^2 L$ applied to $(\Delta S, \Delta A, 0, 0)$ is

$$2\langle S(A,B,C) - T, \Delta S(\Delta A, B, C)\rangle + \|\Delta S(A,B,C) + S(\Delta A, B, C)\|^2$$

The magnitude of *decrease* in this perturbation is given by

$$\begin{aligned}\langle T, \Delta S(\Delta A, B, C)\rangle &= \sigma T(a, \alpha_1 b, \alpha_2 c)\\ &= \sigma\alpha_1\alpha_2 T(a,b,c)\\ &\geq \sigma\alpha_1\alpha_2\kappa_1.\end{aligned}$$

To bound the magnitude of the *increase*, observe that

$$\|BB^\top v\|_F = \|\alpha_1 B_1 b\|_F \leq \alpha_1 K; \quad \|CC^\top w\|_F \leq \alpha_2 K$$

Then we have

$$\begin{aligned}\langle S(A,B,C), \Delta S(\Delta A, B, C)\rangle &= \langle S(A,B,C), \sigma a \otimes B^\top v \otimes C^\top w\rangle\\ &= \sigma S(Aa, BB^\top v, CC^\top w)\\ &\leq \sigma^2\alpha_1\alpha_2 K^3 \qquad (4.11)\end{aligned}$$

Additionally,

$$\begin{aligned}
\|\Delta S(A,B,C)\|_F^2 &= \|A_2^\top u \otimes \alpha_1 b \otimes \alpha_2 c\|_F^2 \\
&\leq \sigma^2 \alpha_1^2 \alpha_2^2 \\
\|S_{(1)}^\top u\|_F^2 &= u^\top (S_{(1)} S_{(1)}^\top - AA^\top) u + u^\top AA^\top u \\
&\leq \tau^{1/4} + \sigma^2 \\
\|S(\Delta A,B,C)\|_F^2 &= \sigma^2 \|au^\top S_{(1)}(B \otimes C)\|_F^2 \\
&\leq \sigma^2 \|S_{(1)}^\top u\|_F^2 \|B\|_F^2 \|C\|_F^2 \\
&\leq \sigma^2 K^4 (\tau^{1/4} + \sigma^2)
\end{aligned}$$

Putting this together, we bound $\|\Delta S(A,B,C) + S(\Delta A,B,C)\|_F^2$ above by

$$\left(\sigma\alpha_1\alpha_2 + \sigma K^2 \sqrt{\tau^{1/4} + \sigma^2}\right)^2 \tag{4.12}$$

In light of the definition of κ_1 and inequalities (4.11) and (4.12), the second-order perturbation in L is $-\Omega(\sigma\alpha_1\alpha_2\kappa_1)$, i.e. L decreases to second-order in this direction.

Now we turn our attention to the regularizer. We need to show that the second-order increase in the regularizer doesn't overwhelm the decrease in L. Note that the regularizer is degree 4 with respect to $\|AA^\top - S_{(1)}S_{(1)}^\top\|_F$ (and same terms for B and C), so the second order derivatives have a quadratic term in $\|AA^\top - S_{(1)}S_{(1)}^\top\|_F$, which is $O(\tau^{1/4}) = o(\sigma\alpha_1\alpha_2\kappa_1)$; higher-order terms are negligible in comparison.

We've shown that the loss function decreases by at least $\Omega(\sigma\alpha_1\alpha_2\kappa_1)\cdot\epsilon^2$. Since our direction of improvement has constant norm, this implies that $\nabla^2 f$ has an eigenvalue that is smaller than $-\Omega(\sigma\alpha_1\alpha_2\kappa_1) = -\Omega(\sigma^{15/4})$. □

Next, we need to deal with the high order saddle points. Here our main observation is that if we choose directions randomly in the correct subspace, then the perturbation

is going to have a reasonable correlation with the residual tensor with constant probability. This is captured by the following anti-concentration property:

Lemma 28. *Let $X \in \mathbb{R}^{d_1 \times d_2 \times d_3}$, and let $a \in \mathbb{R}^{d_1}, b \in \mathbb{R}^{d_2}, c \in \mathbb{R}^{d_3}$ be independent, uniformly distributed unit vectors. There exist positive numbers $C_1 = \Omega(1/\sqrt{d_1 d_2 d_3}) = \Omega^*(1), C_2 = \Omega(1)$ such that*

$$Pr[|X(a,b,c)| \geq C_1 \|X\|_F] > C_2.$$

Proof. Although our Algorithm 1 for sampling missing directions only requires uniform unit vectors (from appropriate subspaces), we construct these vectors as normalized Gaussian vectors for this lemma in order to apply a Gaussian polynomial anti-concentration result (Theorem 8 in Carbery und Wright (2001)). As such, let a', b', c' be independent standard Gaussian random vectors (of appropriate dimension) and set $a = a'/\|a'\|_2$, $b = b'/\|b'\|_2$, and $c = c'/\|c'\|_2$. Note that there exists some constant $p > 0$ such that $\|a'\|_2 \leq 2\sqrt{d_1}$, $\|b'\|_2 \leq 2\sqrt{d_2}$, and $\|c'\| \leq 2\sqrt{d_3}$ with probability at least p. Next, note that $\mathbb{E}X(a', b', c') = 0$ and

$$\begin{aligned}
\text{Var}[X(a', b', c')] &= \mathbb{E}(X(a', b', c')^2) \\
&= \mathbb{E}\langle X(a'a'^\top, b'b'^\top, c'c'^\top), X\rangle \\
&= \langle X(I, I, I), X\rangle \\
&= \|X\|_F^2.
\end{aligned}$$

Now $X(a', b', c')/\|X\|_F$ is a degree three polynomial function with unit variance, so the anti-concentration inequality implies that for any $\epsilon > 0$,

$$\Pr[|X(a', b', c')/\|X\|_F| \leq \epsilon] \leq O(1)\epsilon^{1/3}.$$

Simply choosing a constant ϵ and re-arranging terms completes the proof. $\square$

Using this idea, when $T_{2,2,1}$ is large, we show how to get a direction of improvement.

Lemma 29. *Set $\kappa_2 = 2K\sigma^{1/8}$. Assume $R(S, A, B, C) < \tau$ and $\|A_3\|_F$, $\|B_3\|_F$, and $\|C_3\|_F$ are all less than γ. Further assume that $\|T_{2,1,1}\|_2$, $\|T_{1,2,1}\|_2$, and $\|T_{1,1,2}\|_2$ are each less than κ_1. Let a, b, c, u, v, w be the output of Algorithm 1 given the input $A, B, C, \sigma, (2, 2, 1)$. Define the directions $\Delta A = ua^\top$, $\Delta B = vb^\top$, $\Delta S = u \otimes v \otimes w$. If $\|T_{2,2,1}\|_2 \geq \kappa_2$, then with constant probability, a step in these directions decreases the objective function by $\Omega^*(\sigma^{15/8})$.*

Proof. First, observe that $\kappa_2 \geq 2K\sqrt{\gamma}$, which implies that $\text{rank}(A_1) < r$ and $\text{rank}(B_1) < r$ by Lemma 25. Set α such that $\alpha c = C_1^\top w$, and note that $\alpha \geq \sigma$. Per lemma 28, with constant probability we have $|T(a, b, c)|$ is with some constant factor of $\|T_{2,2,1}\|_2$. Therefore, with constant probability, $|T(a, b, c)| \geq C\kappa_2$ for some positive constant C.

Observe that $p(\delta) := f(S + \delta\Delta S, A + \delta\Delta A, B + \delta\Delta B, C)$ defines a degree 8 polynomial in δ. Set $\delta = \sigma^{1/4}$. For convenience, define the following expressions related to the perturbations of L:

$$
\begin{aligned}
L_0 &= S(A, B, C) - T \\
L_1 &= \Delta S(A, B, C) + S(\Delta A, B, C) + S(A, \Delta B, C) \\
L_2 &= \Delta S(\Delta A, B, C) + \Delta S(A, \Delta B, C) + S(\Delta A, \Delta B, C) \\
L_3 &= \Delta S(\Delta A, \Delta B, C)
\end{aligned}
$$

We can upper bound each of these terms in norm, e.g.

$$\begin{aligned}\|L_1\|_F &= \|A^\top u \otimes B^\top v \otimes C^\top w + S(ua^\top, B, C) + S(A, vb^\top, C)\|_F \\ &\leq \sigma^2\alpha + 2K^2\sqrt{\tau^{1/4} + \sigma^2} \\ &= O(\alpha\sigma^2 + \sigma).\end{aligned}$$

Through similar calculations, we have $\|L_2\|_F = O(\alpha\sigma + \sigma)$ and $\|L_3\|_F = O(\alpha)$. The perturbation in the tensor loss is then

$$\delta^3\langle L_0, L_3\rangle + \delta\langle L_0, L_1\rangle + \delta^2\langle L_0, L_2\rangle + \|\delta L_1 + \delta^2 L_2 + \delta^3 L_3\|_F^2. \tag{4.13}$$

Here the first term is responsible for the decrease in tensor loss:

$$\begin{aligned}\delta^3\langle L_0, L_3\rangle &= \delta^3\alpha\langle S(A, B, C) - T, a \otimes b \otimes c\rangle \\ &\leq \delta^3\alpha(K^2\sigma^2 - T(a, b, c)) \\ &= -\delta^3\alpha\Omega(\kappa_2).\end{aligned}$$

For the other terms, we show that they are small enough so they will not cancel this improvement. Observe that

$$\begin{aligned}\langle L_0, L_1\rangle &= \langle L_0, \Delta S(A, B, C)\rangle + \langle L_0, S(ua^\top, B, C) + S(A, vb^\top, C)\rangle \\ &= O(\alpha\sigma^2) + O(\kappa_1\sigma).\end{aligned}$$

The $O(\kappa_1\sigma)$ term appears because $\|S_{2,1,1}(A, B, C) - T_{2,1,1}\|_F = O(\kappa_1)$, $\|S_{1,2,1}(A, B, C) - T_{1,2,1}\|_F = O(\kappa_1)$.

For the next term, we note that $\langle L_0, L_2\rangle$ is a sum of three inner products, any two of which we can make nonpositive by flipping the sign of ΔS and one of ΔA, ΔB

(doing so doesn't change the amount by which the tensor loss decreases). Hence, by design of Algorithm 1, with constant probability we know that $\langle L_0, L_2 \rangle \leq 0$. As a result, we know (4.13) is at most $-\delta^3 \alpha \Omega(\kappa_2)$.

We now consider the perturbations of the regularizer. Define the following terms:

$$
\begin{aligned}
l_{0,1} &= AA^\top - S_{(1)}S_{(1)}^\top, \quad l_{0,2} = BB^\top - S_{(2)}S_{(2)}^\top, \quad l_{0,3} = CC^\top - S_{(3)}S_{(3)}^\top \\
l_{1,1} &= A\Delta A^\top + \Delta A A^\top - S_{(1)}(\Delta S)_{(1)}^\top - (\Delta S)_{(1)}S_{(1)}^\top \\
l_{1,2} &= B\Delta B^\top + \Delta B B^\top - S_{(2)}(\Delta S)_{2)}^\top - (\Delta S)_{(2)}S_{(2)}^\top \\
l_{1,3} &= -S_{(3)}(\Delta S)_{(3)}^\top - (\Delta S)_{(3)}S_{(3)}^\top, \quad l_{2,3} = -(\Delta S)_{(3)}(\Delta S)_{(3)}^\top
\end{aligned}
$$

We bound these terms in norm as follows:

$$
\begin{aligned}
\|l_{1,1}\|_F &= \|Aau^\top + ua^\top A^\top - uS(I,v,w)^\top - S(I,v,w)u^\top\|_F \\
&\leq 2\|A_2\|_F + 2\|S(I,v,w)\|_F \\
&\leq 2\sigma + 2\sqrt{\tau^{1/4} + \sigma^2} \\
&= O(\sigma).
\end{aligned}
$$

Likewise, $\|l_{1,2}\|_F = O(\sigma)$, $\|l_{1,3}\|_F = O(\sigma)$, and $\|l_{2,3}\| = O(1)$. Also note that $\|l_{0,i}\|_F \leq \tau^{1/4}$ for $i = 1, 2, 3$. Using this, we have

$$
\begin{aligned}
\|l_{0,1} + \delta l_{1,1}\|_F &\leq O(\tau^{1/4}) + O(\delta\sigma) \\
\|l_{0,2} + \delta l_{1,2}\|_F &\leq O(\tau^{1/4}) + O(\delta\sigma) \\
\|l_{0,3} + \delta l_{1,3} + \delta^2 l_{2,3}\|_F &\leq O(\tau^{1/4}) + O(\delta\sigma) + O(\delta^2).
\end{aligned}
$$

All of these terms are dominated by $O(\delta^2)$, and so the perturbed regularizer is bounded by $O(\delta^8) = O(\sigma^2) = o(\sigma^{15/8})$. Hence, we see that the decrease in the tensor loss

dominates the increase in the regularizer, as desired. □

We next address the case where $T_{2,2,2}$ is large, which corresponds to A_1, B_1, C_1 being rank deficient.

Lemma 30. *Set $\kappa_3 = 2K\sigma^{1/2}$. Assume $R(S, A, B, C) < \tau$ and $\|A_3\|_F$, $\|B_3\|_F$, and $\|C_3\|_F$ are all less than γ. Further assume that $\|T_{2,1,1}\|_2$, $\|T_{1,2,1}\|_2$, and $\|T_{1,1,2}\|_2$ and each less than κ_1, while $\|T_{2,2,1}\|_2$, $\|T_{2,1,2}\|_2$, and $\|T_{1,2,2}\|_2$ are each less than κ_2. Let a, b, c, u, v, w be the output of Algorithm 1 with input $A, B, C, \sigma, (2, 2, 2)$. Define the directions $\Delta A = ua^\top$, $\Delta B = vb^\top$, $\Delta C = wc^\top$, $\Delta S = u \otimes v \otimes w$. If $\|T_{2,2,2}\|_2 \geq \kappa_3$, then with constant probability, a step in these directions decreases the objective function by $\Omega^*(\sigma^{3/4})$.*

Proof. First observe that $\kappa_3 \geq 2K\sqrt{\gamma}$, which by Lemma 25 means that $\text{rank}(A_1)$, $\text{rank}(B_1)$, and $\text{rank}(C_1)$ are all strictly less than r. Then A_1, B_1, and C_1 are all missing directions from the relevant subspaces of T, and we are near a fourth-order saddle point. By lemma 28, with constant probability, $|T(a, b, c)| > C\kappa_3$ for some positive constant C.

Again let $p(\delta) = f(S + \delta\Delta S, A + \delta\Delta A, B + \delta\Delta B, C + \delta\Delta C)$, and set $\delta = \sigma^{1/8}$. As in the proof of lemma 29, let for $i = 0, \ldots, 4$, let L_i denote the i-th order perturbation term in

$$(S + \Delta S)(A + \Delta A, B + \Delta B, C + \Delta C) - T.$$

We can upper bound each of these terms in norm, e.g.

$$\begin{aligned}\|L_1\|_F &= \|A^\top u \otimes B^\top v \otimes C^\top w + S(ua^\top, B, C) + S(A, vb^\top, C) \\ &\quad + S(A, B, wc^\top)\|_F \\ &\leq 8\sigma^3 + K^2(\|S(u, I, I)\|_F + \|S(I, v, I)\|_F + \|S(I, I, w)\|_F) \\ &\leq 8\sigma^3 + 3K^2\sqrt{\tau^{1/4} + 2\sigma^2} \\ &= O(\sigma).\end{aligned}$$

Through similar calculations, we have $\|L_2\| = O(\sigma)$ and $\|L_3\| = O(\sigma)$. On the other hand, $\|L_4\| \leq 1$ and $\|L_0\| \leq 2K$. The perturbation in the tensor loss is then

$$\sum_{i,j=0}^{4} \langle L_i, L_j \rangle \delta^{i+j} \tag{4.14}$$

The decrease in the tensor loss is due to the following term:

$$\begin{aligned}\delta^4 \langle L_0, L_4 \rangle &= \delta^4 \langle S(A, B, C) - T, a \otimes b \otimes c \rangle \\ &\leq \delta^4 (K\sigma^3 - T(a, b, c)) \\ &= -\delta^4 \Omega(\kappa_3) \\ &= -\Omega(\sigma^{3/4}).\end{aligned}$$

By a simple Cauchy-Schwarz bound, the other perturbation terms in (4.14) are all bounded by $O(\sigma + \delta^8) = O(\sigma) = o(\sigma^{3/4})$.

Now we analyze the perturbations of the regularizer. As before, define the terms

$$l_{0,1} = AA^\top - S_{(1)}S_{(1)}^\top, \quad l_{0,2} = BB^\top - S_{(2)}S_{(2)}^\top, \quad l_{0,3} = CC^\top - S_{(3)}S_{(3)}^\top$$
$$l_{1,1} = A\Delta A^\top + \Delta A A^\top - S_{(1)}(\Delta S)_{(1)}^\top - (\Delta S)_{(1)}S_{(1)}^\top$$
$$l_{1,2} = B\Delta B^\top + \Delta B B^\top - S_{(2)}(\Delta S)_{2)}^\top - (\Delta S)_{(2)}S_{(2)}^\top$$
$$l_{1,3} = C\Delta C^\top + \Delta C C^\top - S_{(3)}(\Delta S)_{(3)}^\top - (\Delta S)_{(3)}S_{(3)}^\top$$

We bound these terms in norm as follows:

$$\begin{aligned}
\|l_{1,1}\|_F &= \|Aau^\top + ua^\top A^\top - uS(I,v,w)^\top - S(I,v,w)u^\top\|_F \\
&\leq 2\|A_2\|_F + 2\|S(I,v,w)\|_F \\
&\leq 2\sigma + 2\sqrt{\tau^{1/4} + \sigma^2} \\
&= O(\sigma).
\end{aligned}$$

Likewise, $\|l_{1,i}\|_F = O(\sigma)$ for $i = 2, 3$, and of course $\|l_{0,i}\|_F \leq \tau^{1/4}$ for $i = 1, 2, 3$. Again,

$$\|l_{0,i} + \delta l_{1,i}\|_F \leq O(\tau^{1/4}) + O(\delta\sigma)$$

and using this, we can bound the perturbed regularizer as $O(\delta^4\sigma^4) = o(\sigma^{3/4})$. Hence, the decrease in the tensor loss dominates all other perturbations, and we improve the objective function by $\Omega(\sigma^{3/4})$. □

4.4.8 Algorithm Description and Proof of Main Theorem

Before sketching the algorithm we will first prove Lemma 21 and explain some of the parameter choices. Refer to Table 4.1 for a list of the numerical quantities that were introduced.

Proof of Lemma 21. Set τ small enough so that κ_0, $d\kappa_1 + K^3\sigma$, $d\kappa_2 + K^2\sigma^2$, $d\kappa_3 + K\sigma^3 < \sqrt{\epsilon}/4$ and $\tau < \epsilon/2$. We then set $\tau_1 = \Theta^*(\tau)$ from Lemma 22 and $\tau_2 = \Theta(\sigma^{15/4})$ from Lemma 27.

Now assume that conditions (1), (2), and (3) from the statement of the Lemma fail to hold. We seek to show that $f(S, A, B, C) < \epsilon$. By Lemma 22 and our choice of τ_1, we have that $R(S, A, B, C) < \tau \leq \epsilon/2$. By Lemma 23, we have that $\|A_3\|_F, \|B_3\|_F, \|C_3\|_F$ are all less than γ. By Lemma 26, we have that $\|T_{1,1,1} - S(A, B, C)\|_F \leq \kappa_0$. By Lemma 27, we have that $\|T_{i,j,k}\|_2 < \kappa_1$ for $(i, j, k) \in \{(2, 1, 1), (1, 2, 1), (1, 1, 2)\}$. By Lemma 29, we have that $\|T_{i,j,k}\|_2 < \kappa_2$ for $(i, j, k) \in \{(2, 2, 1), (2, 1, 2), (1, 2, 2)\}$. By Lemma 30, we have that $\|T_{2,2,2}\|_2 < \kappa_3$.

Combining all of these bounds, we have

$$\begin{aligned} f(S, A, B, C) &= R(S, A, B, C) + \sum_{i,j,k} \|S_{i,j,k}(A_i, B_j, C_k) - T_{i,j,k}\|_F^2 \\ &< \epsilon/2 + \kappa_0^2 + 3(K^3\sigma + d\kappa_1)^2 + 3(K^2\sigma^2 + d\kappa_2)^2 + (K\sigma^3 + d\kappa_3)^2 \\ &< \epsilon/2 + \epsilon/2, \end{aligned}$$

as desired.

□

We now sketch our algorithm in Algorithm 2. The algorithm basically tries to follow the main Lemma 21. If the point has large gradient or negative eigenvalue in Hessian, we can just use any standard local search algorithm. When the point is a higher order saddle point, we use Algorithm 1 as in Lemma 29 or Lemma 30 to generate directions of improvements.

Now we are ready to prove Theorem 3

Proof of Theorem 3. By Lemma 21, for any (τ_1, τ_2)-second order stationary point, if

Algorithm 2 Local search algorithm for Tucker decomposition

Require: tensor T, error threshold ϵ

Choose thresholds τ_1, τ_2 according to Lemma 21.

repeat

Run a local search algorithm to find (τ_1, τ_2)-second order stationary point.

Call Algorithm 1 for $i, j, k = 1, 2$ to generate improvement directions, repeat for $O(\log 1/\epsilon)$ times.

if any of the generated directions improve the function value by at least $\Omega^*(\sigma^{15/8})$ **then**

Move in the direction.

Break.

end if

until no direction of improvement can be found

$f \geq \epsilon$ Lemma 29 and Lemma 30 will be able to generate a direction of improvement that improves the function value by at least $\Omega^*(\sigma^{15/8})$ with constant probability. Since the initial point has constant loss, if a direction of improvement is found for more than $O^*(1/\sigma^{15/8})$ iterations, then the function value must already be smaller than ϵ.

After the repetition, the probability that we find a direction of improvement is at least $1 - o(\sigma)$. By union bound, we know that with high probability for all the iterations we can find a direction of improvement. □

4.5 Conclusion

In this chapter we showed that the standard nonconvex objective for Tucker decomposition with appropriate regularization does not have any spurious local minima. We further gave a local search algorithm that can optimize a regularized version of the objective in polynomial time. There are still many open problems for the optimization of the Tucker decomposition objective. For example, in many applications, the low rank tensor T is not known exactly. We either have significant additive noise $T + E$, or observe only a subset of entries of T (tensor completion). Local search algorithms

on the nonconvex objective are able to handle similar settings for matrices (Chi u. a., 2019). We hope our techniques in this chapter can be extended to give stronger guarantees for noisy Tucker decomposition and tensor completion.

Chapter 5

Learning Linear State Representations

We now consider representation learning in the context of reinforcement learning and control. Control problems center on decision-making in a dynamic environment that responds actively to decisions. It is therefore crucial that data representations preserve these dynamics, so that the consequences of any decision can be predicted. On the other hand, observed data often contain irrelevant or redundant nuisance features, and the key underlying dynamics can be opaque. The data representations should therefore also filter out the unwanted features and simplify the dynamics. In this chapter, we approach this problem by proposing a model for control systems with high-dimensional, nonlinear observations but a latent, low-dimensional linear dynamical system driving the dynamics.

Acknowledgements The results in this chapter are joint work with Rong Ge and Holden Lee.

5.1 Introduction

Reinforcement learning has made tremendous progress recently, achieving strong performance in difficult problems like go (Silver u. a., 2017) and Starcraft (Vinyals u. a., 2019). A common theme in the recent progress is the use of neural networks to handle the cases when the system dynamics and policy are both highly nonlinear. However, theoretical understanding for reinforcement learning is most thoroughly developed in the tabular setting (where the number of state/actions is small) or when

the underlying dynamics of the system is linear (see Section 5.1.1).

Requiring the dynamics to be linear is especially limiting for problems with *rich*, high dimensional output, e.g. manipulating a robot from video frames or playing a game by observing pixel representations. Consider a simple system where we control an object by applying forces to it. The state of the object (position and velocity) can evolve linearly according to physical laws. However, if the observation is a visual rendering of this object in a 3-d environment, the observation contains a lot of redundant information and doesn't have linear dynamics. Such problems can potentially be solved by learning a *state representation mapping* ϕ that maps the complicated observations to states that satisfy simpler dynamics. State representation learning is popular in practice, but theoretical understanding is still nascent; see the survey by Lesort u. a. (2018) and more references in Section 5.1.1. Many approaches either try to learn a *forward model*, which predicts the next state or an *inverse model*, which predicts the action taken given the states. In this paper we show both approaches can provably extract a state representation that encodes linear dynamics.

We first consider a simple theoretical model where the full observation x does not have linear dynamics, but there exists an unknown subspace $\mathcal{V}$ where the projection $y = \Pi_{\mathcal{V}} x$ has linear dynamics. This corresponds to the case when the state representation mapping is a linear projection. We give two provably correct algorithms for identifying $\mathcal{V}$, one based on learning a linear forward model, and one based on learning a linear inverse model.

In more complicated settings one might need a nonlinear mapping in order to extract a latent space representation that has linear dynamics. We extend our algorithms to the nonlinear setting, and show that if we can find solutions to similar nonconvex optimization problems with 0 loss, then the representations have nontrivial linear dynamics.

We discuss related works in Section 5.1.1. We next introduce our model and discuss how one can formalize learning a state representation mapping as optimization problems in Section 5.2. In Sections 5.3 and 5.4 we give algorithms based on forward and inverse models, respectively, and prove that these recover the underlying state representation. We then extend these to nonlinear state representations in Section 5.5. Finally in Section 5.6, we empirically validate our approach on synthetic data and simple RL environments. All detailed proofs are found in the appendix.

5.1.1 Related Work

State Representation Learning with Rich Observations Several recent papers have addressed the problem of state representation learning (SRL) in control problems. Lesort u. a. (2018) survey the recent literature and identify four categories that describe many SRL approaches: reconstructing the observation, learning a forward dynamics model, learning an inverse dynamics model, and using prior knowledge to constrain the state space. Raffin u. a. (2019) evaluate many of these SRL approaches on robotics tasks and show how to combine the strengths of the different methods. Several papers adopt the approach of learning forward or inverse models (Hafner u. a., 2019b; Pathak u. a., 2017; Zhang u. a., 2018) and demonstrate practical effectiveness, but they lack a theoretical analysis of the approach. Our work aims to help fill this gap.

Domains with rich observations include raw images or video frames from video games (Anand u. a., 2019), robotics environments (Higgins u. a., 2017), and renderings of classic control problems (Watter u. a., 2015), and deep learning methods have enabled success in this space. Srinivas u. a. (2020) use a contrastive learning approach to extract state representations from pixels. Ha und Schmidhuber (2018) learn low-dimensional representations and dynamics which simple linear policies to achieve effective control. Hafner u. a. (2019a) utilize latent imagination to learn behaviors

that achieve high performance in terms of reward and sample-efficiency on several visual control tasks.

Theoretical work on state representation learning Du u. a. (2019a) investigate whether good representations lead to sample-efficient RL in the context of MDPs, showing exponential lower bounds in many settings. Our setting circumvents these negative results because the representations that we learn transform the nonlinear problem into a linear (and hence tractable) control problem. Other works (Du u. a., 2019b; Misra u. a., 2019) study the Block MDP model, in which a high-dimensional observation space is generated from a finite set of latent states. Our model, by contrast, considers continuous state and action spaces.

Recent work by Mhammedi u. a. (2020) studies a similar setting to ours, where a latent LQR control problem generates nonlinear observations. That work is interested in finding a near-optimal controller with respect to the quadratic costs, whereas we focus here on finding the ground-truth representation that has linear dynamics. We give both a forward and an inverse approach while Mhammedi u. a. (2020) focuses on learning inverse models. Recent work by Dean u. a. (2020) also consider a somewhat similar setting, although their focus is on robust control guarantees, and their state representations are learned through a supervised technique that requires ground-truth representations.

Linear Dynamical Systems and Control Problems Linear dynamical systems and control problems have been extensively studied for many decades and admit efficient, robust, and provably correct algorithms. For the problem of system identification, Qin (2006) gives a review of subspace identification methods; our inverse model approach, while for a different setting, is somewhat similar to the regression approaches described in the review. Other recent works analyze gradient-based methods

for system identification (Hardt u. a., 2018), policy optimization (Fazel u. a., 2018), and online control (Cohen u. a., 2018) in the setting of linear dynamical systems and quadratic costs.

5.2 Hidden Subspace Model

In this section we introduce a basic model which admits a linear state representation mapping. Later we show that the linear state representation can be learned efficiently via either a forward or inverse modelling approach.

5.2.1 Notation and Preliminaries

We follow the notation of a discrete-time control system, where x_t denotes the (observed) state at the t-th step, u_t denotes the control signal (action) at the t-th step, and f denotes the dynamics function, $x_{t+1} = f(x_t, u_t)$. A *state representation mapping* is a function $\phi : \mathbb{R}^d \to \mathbb{R}^r$ that maps x_t to a different space (usually $r \ll d$) such that the dynamics governing the evolution of $\phi(x_t)$ are simpler, e.g., linear. If $a_1, \ldots, a_k$ are vectors, we let $(a_1, \ldots, a_k)$ denote the concatenation of these vectors.

We adapt the standard notion of controllability from control systems theory to be subspace-dependent.

Definition 3. *Given matrices $A \in \mathbb{R}^{d\times d}$, $B \in \mathbb{R}^{d\times l}$, and an r-dimensional subspace $\mathcal{V} \subset \mathbb{R}^d$, we say that the tuple (A, B) is* $\mathcal{V}$-controllable *if $col(A), col(A^\top), col(B) \subset \mathcal{V}$ and the $d \times rl$ matrix*

$$\begin{bmatrix} B & AB & \cdots & A^{r-1}B \end{bmatrix}$$

has rank r.

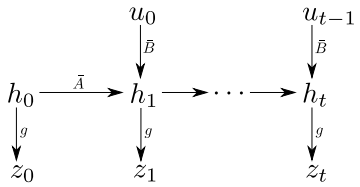


Figure 5.1: The hidden subspace model; latent states h_i evolve according to a linear control system and generate nonlinear features z_i

5.2.2 Hidden Subspace Model

We consider a model with a latent ground truth state $h_t \in \mathbb{R}^r$ and controls $u_t \in \mathbb{R}^l$ that satisfy linear dynamics:

$$h_{t+1} = \bar{A}h_t + \bar{B}u_t.$$

We observe a high-dimensional state $x_t \in \mathbb{R}^d$ with $d > r$ that satisfies $x_t = Vh_t + V^\perp g(h_t)$. Here, $V \in \mathbb{R}^{d\times r}$ and $V^\perp \in \mathbb{R}^{(d-r)\times r}$ are full-rank matrices whose columns respectively form bases for an r-dimensional subspace $\mathcal{V}$ and its orthogonal complement $\mathcal{V}^\perp$, and $g(h_t) \in \mathbb{R}^{d-r}$ is a nonlinear, possibly stochastic function of h_t. We use y_t to denote Vh_t and z_t to denote $V^\perp g(h_t)$, and we call these the *linear* and *nonlinear* parts of x_t, respectively. The model is illustrated in Figure 5.1.

Observe that y_t also satisfies linear dynamics, namely $y_{t+1} = Ay_t + Bu_{t-1}$, where $A = V\bar{A}V^+$ and $B = V\bar{B}$. On the other hand, z_t is conditionally independent of all other variables $\{u_i : i \geq 0\}$, $\{y_i : i \neq t\}$, and $\{z_i : i \neq t\}$ given h_t. Thus, we can write x_t as a sum of two orthogonal components, $x_t = y_t + z_t$, where y_t evolves linearly and z_t contains the nonlinear, redundant features. The constraint that the linear and nonlinear parts of x_t lie in mutually orthogonal subspaces enables one to project away z_t and recover the linear part, if $\mathcal{V}$ is known. Our task is to extract the latent state h_t (or any invertible linear transformation thereof) from x_t, given observed trajectories $x_0, x_1, x_2, \ldots$ and controls $u_0, u_1, \ldots$. To find this mapping it suffices to recover the

hidden subspace $\mathcal{V}$.

Throughout this section and Sections 5.3 and 5.4, we assume that the initial latent state h_0 and the controls u_i are independent standard Gaussian random vectors, that $\mathbb{E}[z_i] = 0$ for each i, and that Σ_{x_i} is full rank for each i. All expectations are taken over the randomness induced by h_0, u_i, and z_i.

5.2.3 Learning Forward and Inverse Models

In order to learn a state representation mapping that induces linear dynamics, a natural approach is to jointly learn the representation and a dynamics model that enforces linearity. Suppose we take random actions $u_0, u_1, \ldots$ from a random initial state x_0, generating a trajectory of observations $x_1, x_2, \ldots$. We could attempt to learn a mapping ϕ and matrices $C \in \mathbb{R}^{r \times r}, D \in \mathbb{R}^{r \times l}$ such that the forward dynamics equation is linear: $\phi(x_{t+1}) = C\phi(x_t) + Bu_t$. Alternatively, we could instead seek to learn ϕ and matrices $P, L \in \mathbb{R}^{l \times d}$ such that the inverse dynamics equation is linear: $u_t = P\phi(x_{t+1}) - L\phi(x_t)$.

We show how both of these ideas can be carefully implemented to identify $\mathcal{V}$ in the hidden subspace model. In particular, we propose the *forward model objective*

$$\min_{P,Q,D} \frac{1}{2}\mathbb{E}\|Px_1 - Qx_0 - Du_0\|_2^2 + \frac{\lambda}{4}\|P\Sigma_{x_1}P^\top - I\|_F^2$$

as well as the *inverse model objective*

$$\min_{P,\{L_i\},\{T_i\}} \frac{1}{2}\mathbb{E}\sum_{i=1}^{r}\|Px_i - L_i x_0 - \sum_{k=1}^{i-1} T_k u_{i-1-k} - u_{i-1}\|_2^2.$$

While both of these approaches are viable in the hidden subspace model, it's worth noting how they differ from each other. The forward model objective is non-

convex, only requires one step of the control system, and it immediately yields the state representation map P. By contrast, the inverse model is a convex optimization problem, it considers trajectories of length r, and the final state representation map is constructed from $P, L_1, \ldots, L_r$. The two approaches also require different assumptions for their theoretical guarantees. In Sections 5.3 and 5.4 we motivate and analyze these two approaches in more detail.

5.3 Forward Model

In this section, we focus on the forward model, which tries to predict the next state given the current state and action. We first motivate the design of forward objective function (5.1) and prove its guarantees. We next explain how the forward model objective is connected to canonical correlation analysis. We finally study the sample complexity of the empirical version of the problem.

Recall the forward model objective

$$\min_{P,Q,D} \frac{1}{2}\mathbb{E}\|Px_1 - Qx_0 - Du_0\|_2^2 + \frac{\lambda}{4}\|P\Sigma_{x_1}P^\top - I\|_F^2 \tag{5.1}$$

where $P, Q \in \mathbb{R}^{r\times d}$, $D \in \mathbb{R}^{r\times l}$, and $\lambda > 0$. To motivate (5.1), note that a first attempt to learn a linear forward model is to find matrices P, C, D that satisfy $Px_1 = CPx_0 + Du_0$. This, however, immediately runs into the problem of trivial solutions – we can choose these matrices to be all zero and the linear dynamics equation holds, but this is clearly not a useful representation and it doesn't recover the subspace $\mathcal{V}$.

A simple way to rule out such trivial solutions is to constrain the state representation Px_1 to have a full-rank covariance matrix. Dealing with non-convex rank constraints directly can be difficult, so we instead introduce a regularizer term $\|\mathbb{E}[Px_1x_1^\top P] - I\|_F^2$

which encourages Px_1 to have spherical covariance. Additionally, we relax the forward dynamics model to a simpler linear model $Px_1 = Qx_0 + Du_0$. This relaxation removes some non-convexity from the objective function and simplifies the analysis. Both of these adjustments lead to (5.1). This objective is still non-convex due to the regularizer term, but its landscape is benign, as explained in Theorem 4.

In order to ensure that solutions to (5.1) recover $\mathcal{V}$, we specify the nonlinearity of z_1 as follows:

Assumption 1 (Forward Nonlinearity)**.** *There exists a constant $\rho \in (0, 1)$ such that $\rho(z_1, (x_0, u_0)) \leq \rho$.*

This assumption simply asserts that z_1 is not linearly dependent on the initial data x_0, u_0. We can now state the theoretical guarantees for the forward model.

Theorem 4. *Set $\lambda \in (0, 1 - \rho^2)$, and let (P, Q, D) be a second-order stationary point of* (5.1)*. Under Assumption 1, col$(P^\top) = \mathcal{V}$.*

A variety of local search algorithms (such as perturbed gradient descent (Jin u. a., 2017)) are proven to efficiently find second-order stationary points, and so Theorem 4 implies that we can efficiently recover $\mathcal{V}$ by optimizing (5.1). Intuitively, Theorem 4 holds because if the rows of P aren't all in $\mathcal{V}$, then Pz_1 is nonzero. However, Assumption 1 implies that z_1 is *not* a linear function of x_0 and u_0, so Pz_1 contributes excess loss in the first term of (5.1), so it should be removed. Moreover, if P is rank-deficient, we can reduce the second term of (5.1) by increasing the rank of P while ensuring its rows stay in $\mathcal{V}$. This argument holds in the more general setting of CCA, as explained in Section . Our proof makes this intuition precise by first showing that Assumption 1 implies a gap in the canonical correlations between x_1 and (x_0, u_0), and then utilizing the theory we develop in the CCA setting, as given in Theorem 5.

We start with a lemma that connects Assumption 1 to the canonical correlations between x_1 and (x_0, u_0).

Lemma 31. *Let w denote (x_0, u_0) and let x denote x_1. Define the matrix $C = \Sigma_x^{-1/2}\Sigma_{xw}\Sigma_w^{-1}\Sigma_{wx}\Sigma_x^{-1/2}$. Let $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d \geq 0$ be the eigenvalues of C (with multiplicity) and let $c_1, \ldots, c_d$ be corresponding orthonormal eigenvectors. Then under Assumption 1, $\lambda_i = 1$ for $i = 1, \ldots, r$ and $\lambda_{r+1} \leq \rho^2$. Moreover, span$\{\Sigma_x^{-1/2}c_i : 1 \leq i \leq r\} = \mathcal{V}$.*

Proof. We first show that $\lambda_{r+1} \leq \rho^2$. Define $\tilde{\mathcal{V}} = \{\Sigma_x^{1/2}v : v \in \mathcal{V}\}$. Noting that $\Pi_{\mathcal{V}}x = z_1$ and $\dim(\tilde{\mathcal{V}}) = 1$, by the variational characterization of eigenvalues of symmetric matrices and Assumption 1, we have that

$$\begin{aligned}
\lambda_{r+1} &= \min_{\mathcal{U}} \left\{ \max_{a \in \mathcal{U}} \frac{a^\top C a}{\|a\|_2^2} \,:\, \dim(\mathcal{U}) = d - r \right\} \\
&\leq \min_{\mathcal{U}} \left\{ \max_{a \in \mathcal{U}, b \in \mathbb{R}^{d+l}} \frac{(a^\top \Sigma_x^{-1/2}\Sigma_{xw}\Sigma_w^{-1/2}b)^2}{\|a\|_2^2\|b\|_2^2} \,:\, \dim(\mathcal{U}) = d - r \right\} \\
&\leq \max_{a \in \tilde{\mathcal{V}}, b \in \mathbb{R}^{d+l}} \frac{(a^\top \Sigma_x^{-1/2}\Sigma_{xw}\Sigma_w^{-1/2}b)^2}{\|a\|_2^2\|b\|_2^2} \\
&= \max_{a \in \mathcal{V}, b \in \mathbb{R}^{d+l}} \frac{(a^\top \Sigma_{xw} b)^2}{(a^\top \Sigma_x a)(b^\top \Sigma_w b)} \\
&= \max_{a \in \mathbb{R}^d, b \in \mathbb{R}^{d+l}} \frac{(a^\top \Sigma_{z_1 w} b)^2}{(a^\top \Sigma_{z_1} a)(b^\top \Sigma_w b)} \\
&= \rho(z_1, w)^2 \\
&\leq \rho^2,
\end{aligned}$$

as desired.

Now let $T : \mathbb{R}^{d+l} \to \mathbb{R}^d$ be the linear transformation satisfying $y_1 = Tw$. Let $v \in \mathcal{V}$.

Note that $v^\top x = v^\top y_1 = v^\top T w$, so we have

$$\begin{aligned} C\Sigma_x^{1/2} v &= \Sigma_x^{-1/2}\Sigma_{xw}\Sigma_w^{-1}\Sigma_{wx} v \\ &= \Sigma_x^{-1/2}\Sigma_{xw}\Sigma_w^{-1}\Sigma_w T^\top v \\ &= \Sigma_x^{-1/2}\Sigma_{xw} T^\top v \\ &= \Sigma_x^{-1/2}\Sigma_x v \\ &= \Sigma_x^{1/2} v. \end{aligned}$$

Hence, $\Sigma_x^{1/2} v$ is an eigenvector of C with eigenvalue equal to 1. Since $\mathcal{V}$ is r-dimensional, we conclude that the top r eigenvalues of C are all equal to 1, and the corresponding r-dimensional eigenspace is precisely $\tilde{\mathcal{V}}$. It then follows that $\text{span}\{\Sigma_x^{-1/2} c_i : 1 \le i \le r\} = \mathcal{V}$. □

We are now ready to prove Theorem 4.

Proof. Let (P, Q, D) be a second-order stationary point of (5.1). Let x, w, C be as in Lemma 31. Let $\mathcal{C}_r$ be as Theorem 5 where u is identified with x and v is identified with w. By Lemma 31, we have that $\mathcal{V} = \Sigma_x^{-1/2}\mathcal{C}_r$. By Theorem 5, we have $\text{rank}(P) = r$ and $\text{col}(\Sigma_x^{1/2} P^\top) = \mathcal{C}_r$. Thus, we conclude that $\text{col}(P^\top) = \Sigma_x^{-1/2}\mathcal{C}_r = \mathcal{V}$, as desired. □

5.3.1 Connection to CCA

We now analyze the forward model objective in a more general setting and draw connections to canonical correlation analysis. Consider two random vectors $u \in \mathbb{R}^n$, $v \in \mathbb{R}^m$ with full-rank covariance matrices Σ_u and Σ_v. Canonical correlation analysis deals with finding the directions of maximal correlation between u and v. To this end,

we propose the optimization problem

$$\min_{P,Q} \frac{1}{2}\mathbb{E}_{u,v}\|Pu - Qv\|_2^2 + \frac{\lambda}{4}\|P\Sigma_u P^\top - I\|_F^2 \tag{5.2}$$

where $P \in \mathbb{R}^{r\times n}$, $Q \in \mathbb{R}^{r\times m}$, and $\lambda \in (0,1)$ is a hyperparameter. Define $C = \Sigma_u^{-1/2}\Sigma_{uv}\Sigma_v^{-1}\Sigma_{vu}\Sigma_u^{-1/2}$ and write its spectral decomposition as $C = \sum_{i=1}^d \rho_i^2 c_i c_i^\top$ where $1 \geq \rho_1 \geq \cdots \geq \rho_d \geq 0$ and $\|c_i\|_2 = 1$ for each i. According to CCA theory (Borga, 2001), ρ_i are the canonical correlations between u and v, and the vectors $\Sigma_u^{-1/2}c_i$ are the corresponding canonical correlation directions for u, i.e., the directions in which u maximally correlates with v.

For each $i \in \{1,\ldots,d\}$, define $\mathcal{C}_i = \text{span}\{c_1,\ldots,c_i\}$. Let $\mathcal{C}_0$ denote the trivial subspace $\{0\}$ and define $\rho_0 = 1, \rho_{d+1} = 0$. The subspaces $\mathcal{C}_i$ are useful because they allow us to project u to a lower-dimensional subspace while maximally preserving its correlation with v. Solving the optimization problem (5.2) recovers these subspaces, depending on the value of λ.

Theorem 5. *Let $i \in \{0,1,\ldots,d\}$ satisfy $1-\rho_i^2 < \lambda < 1-\rho_{i+1}^2$. Let (P,Q) be a second-order stationary point of (5.2). Then col$(\Sigma_u^{1/2}P^\top) \subset \mathcal{C}_i$ and rank$(P) = \min\{r,i\}$.*

Proof. To prove Theorem 5, we first analyze the first-order necessary conditions of (5.2), which are closely connected to the matrix C. In particular, the gradients vanish only if the rows of $P\Sigma_u^{1/2}$ are contained in $\mathcal{C}_i$. Next, we show that the loss function can be additionally minimized if $\text{rank}(P) < \min\{r,i\}$. In particular, by carefully increasing the rank of P we can ensure that the regularizer term decreases more than the linear model term increases.

Define

$$f(P,Q) = \frac{1}{2}\mathbb{E}_{u,v}\|Pu - Qv\|_2^2,\ r(P) = \frac{1}{4}\|P\Sigma_u P^\top - I\|_F^2.$$

Let (P, Q) be a second-order stationary point. We first show that $\mathrm{col}(\Sigma_u^{1/2} P^\top) \subset \mathcal{C}_i$. The gradients of $g := f + \lambda r$ are as follows:

$$\begin{aligned}
\nabla_Q g &= Q\Sigma_v - P\Sigma_{uv} \\
\nabla_P g &= P\Sigma_u - Q\Sigma_{vu} + \lambda(P\Sigma_u P^\top - I)P\Sigma_u
\end{aligned}$$

Since these gradients vanish at (P, Q), we have $Q = P\Sigma_{uv}\Sigma_v^{-1}$. Plugging this into the other gradient expression, we have

$$0 = P\Sigma_u^{1/2}((1-\lambda)I - C)\Sigma_u^{1/2} + \lambda P\Sigma_u P^\top P\Sigma_u \tag{5.3}$$

Set $\tilde{P} = P\Sigma_u^{1/2}$ and $\hat{P} = \tilde{P}(I - \Pi_{\mathcal{C}_i})$. Let $a \in \mathbb{R}^r$ and $\eta > 0$ satisfy $\hat{P}\hat{P}^\top a = \eta a$. Note that

$$((1-\lambda)I - C)(I - \Pi_{\mathcal{C}_i}) = (I - \Pi_{\mathcal{C}_i})((1-\lambda)I - C) = \sum_{j=i+1}^{d} (1 - \lambda - \rho_j^2) c_j c_j^\top$$

is positive definite since $1 - \lambda - \rho_j^2 > 1 - (1 - \rho_{i+1}^2) - \rho_j^2 = \rho_{i+1}^2 - \rho_j^2 \geq 0$ for $j > i$. In particular, this implies that

$$a^\top \tilde{P}((1-\lambda)I - C)\hat{P}^\top a = a^\top \hat{P}((1-\lambda)I - C)\hat{P}^\top a \geq 0.$$

Now by left-multiplying (5.3) with $a^\top$ and right-multiplying with $\Sigma_u^{-1/2}\hat{P}^\top a$ we have

$$\begin{aligned}
0 &= a^\top \tilde{P}((1-\lambda)I - C)\hat{P}^\top a + \lambda a^\top \tilde{P}\tilde{P}^\top \tilde{P}\hat{P}^\top a \\
&\geq \lambda a^\top \tilde{P}\tilde{P}^\top \hat{P}\hat{P}^\top a \\
&= \lambda\eta a^\top \tilde{P}\tilde{P}^\top a \\
&\geq \lambda\eta^2 \|a\|_2^2.
\end{aligned}$$

This means that a must be 0, and so $\hat{P}\hat{P}^\top$ has no nonzero eigenvectors, i.e., $\hat{P} = 0$. Thus, the rows of $\tilde{P}$ are contained in $\mathcal{C}_i$, which is precisely what we wanted to show.

Next, we show that $\text{rank}(P) = \min\{i, r\}$. Assume to the contrary that $\text{rank}(P) < \min\{i, r\}$ and let $b \in \mathbb{R}^d$, $a \in \mathbb{R}^r$ be unit vectors satisfying $b \in \mathcal{C}_i$, $\tilde{P}b = 0$, and $\tilde{P}^\top a = 0$. We claim that for sufficiently small $\epsilon > 0$, the point (P', Q'), where $P' = P + \epsilon a b^\top \Sigma_u^{-1/2}$ and $Q' = Q + \epsilon a b^\top \Sigma_u^{-1/2} \Sigma_{uv} \Sigma_v^{-1}$ yields a strictly smaller loss. First observe that

$$(P + \epsilon a b^\top \Sigma_u^{-1/2}) \Sigma_u (P + \epsilon a b^\top \Sigma_u^{-1/2})^\top = P \Sigma_u P^\top + \epsilon^2 a a^\top.$$

Using this and the fact that $P^\top a = \Sigma_u^{-1/2} \tilde{P}^\top a = 0$, we have

$$\begin{aligned}
4r(P') &= \|P' \Sigma_u P'^\top - I\|_F^2 \\
&= \|P \Sigma_u P^\top - I\|_F^2 + \|\epsilon^2 a a^\top\|_F^2 - 2\epsilon^2 \text{tr}(a a^\top) \\
&= \|P \Sigma_u P^\top - I\|_F^2 + \epsilon^4 - 2\epsilon^2 \\
&< 4r(P)
\end{aligned}$$

for sufficiently small ϵ, so r indeed decreases.

Next we inspect the change in f on this step. We have the following calculation:

$$\begin{aligned}
&\mathbb{E}\|\epsilon a b^\top \Sigma_u^{-1/2} (u - \Sigma_{uv} \Sigma_v^{-1} v)\|_F^2 \\
&\qquad = \epsilon^2 \mathbb{E} \langle a b^\top \Sigma_u^{-1/2} (u - \Sigma_{uv} \Sigma_v^{-1} v), a b^\top \Sigma_u^{-1/2} (u - \Sigma_{uv} \Sigma_v^{-1} v) \rangle \\
&\qquad = \epsilon^2 \langle \Sigma_u^{-1/2} b a^\top a b^\top \Sigma_u^{-1/2}, \Sigma_u - \Sigma_{uv} \Sigma_v^{-1} \Sigma_{vu} \rangle \\
&\qquad = \epsilon^2 \langle b b^\top, I - C \rangle \\
&\qquad \leq \epsilon^2 (1 - \rho_i^2)
\end{aligned}$$

since $b \in \mathcal{C}_i$ and $\|b\|_2 = 1$. Furthermore, we have $\mathbb{E}\langle Pu - Qv, a b^\top \Sigma_u^{-1/2} (u - \Sigma_{uv} \Sigma_v^{-1} v) \rangle =$

0 since $a^\top P = 0$ and $a^\top Q = a^\top P \Sigma_{uv} \Sigma_v^{-1} = 0$. Putting these together, we have

$$\begin{aligned}
2f(P', Q') &= \mathbb{E}\|(P + \epsilon a b^\top \Sigma_u^{-1/2})u - (Q + \epsilon a b^\top \Sigma_u^{-1/2} \Sigma_{uv} \Sigma_v^{-1})v\|_2^2 \\
&= \mathbb{E}\|Pu - Qv + \epsilon a b^\top \Sigma_u^{-1/2}(u - \Sigma_{uv}\Sigma_v^{-1} v)\|_2^2 \\
&= 2f(P, Q) + 2\epsilon \mathbb{E}\langle Pu - Qv, a b^\top \Sigma_u^{-1/2}(u - \Sigma_{uv}\Sigma_v^{-1} v)\rangle \\
&\quad + 2\mathbb{E}\|\epsilon a b^\top \Sigma_u^{-1/2}(u - \Sigma_{uv}\Sigma_v^{-1} v)\|_2^2 \\
&\leq 2f(P, Q) + \epsilon^2(1 - \rho_i^2).
\end{aligned}$$

Therefore, we conclude that

$$\begin{aligned}
(f(P, Q) + \lambda r(P)) - (f(P', Q') + \lambda r(P')) &\geq \frac{\lambda(2\epsilon^2 - \epsilon^4)}{4} - \frac{\epsilon^2(1 - \rho_i^2)}{2} \\
&= \frac{(\lambda - (1 - \rho_i^2))\epsilon^2}{2} - \frac{\lambda \epsilon^4}{4}.
\end{aligned}$$

Since $\lambda > 1 - \rho_i^2$, this implies that the Hessian has a negative eigenvalue at (P, Q). This contradicts the fact that (P, Q) is a second-order stationary point, so we conclude that $\text{rank}(P) = \min\{i, r\}$.

□

5.3.2 Sample Complexity

Since (5.1) involves an expectation, we can't solve it exactly. Instead, we optimize the empirical objective function

$$\min_\theta \frac{1}{2n}\|PX_1 - QX_0 - DU_0\|_F^2 + \frac{\lambda}{4}\|P\Sigma_{X_1}P^\top - I\|_F^2 \tag{5.4}$$

where the columns of $X_i \in \mathbb{R}^{d \times n}$ and $U_i \in \mathbb{R}^{l \times n}$ are i.i.d. copies of x_i and u_i, respectively. If n is sufficiently large, the solution to this problem recovers $\mathcal{V}$. We

introduce the following assumption that allows us to utilize quantitative concentration results, and then we state the theoretical guarantee for (5.4).

Assumption 2 (Sub-Gaussianity)**.** *There exists a constant $C > 0$ such that for any unit vector q, $P(|\langle q, \Sigma_{\xi_i\xi_i}^{-1/2}\xi_i\rangle| > t) \leq \exp(-Ct^2)$, where we define $\xi_i := (z_1, x_0, u_0)$.*

Theorem 6. *Set $\lambda \in (0, (1-\rho^2)/4)$, and let (P, Q, D) be a second-order stationary point of (5.4). Under Assumptions 1 and 2, there exists a constant C_0 such that if $n \geq C_0 \log^2(2d+l)/(1-\rho)^2$, then with probability at least 0.99, col$(P^\top) = \mathcal{V}$.*

To prove this theorem, we first use a concentration of measure result (Lemma 32) to establish that the empirical canonical correlations between X_1 and (X_0, U_0) have a sufficient gap (Lemma), and then the arguments from the proof of Theorem 5 do most of the rest of the work.

Define empirical canonical correlation in the natural way: for random vectors y and z, let Y and Z be the corresponding sample matrices and define

$$\rho(Y, Z) = \max_{a,b} \frac{a^\top \Sigma_{YZ} b}{\sqrt{a^\top \Sigma_Y a}\sqrt{b^\top \Sigma_Z b}}.$$

We need to control the difference between $\rho(Y, Z)$ and $\rho(y, z)$ when the number of samples is large enough. We utilize a concentration result stated in (Gao u. a., 2019) that quantifies this.

Lemma 32 (Adapted from Corollary 7 of Gao u. a. (2019))**.** *Assume that $y \in \mathbb{R}^{k_1}$ and $z \in \mathbb{R}^{k_2}$ are sub-Gaussian, set $k = k_1 + k_2$, and let $\epsilon \in (0, 1)$. There exists a constant C such that for any $t \geq 1$, if $n \geq Ct^2 k \log^2 k/\epsilon^2$ then $|\rho(Y, Z) - \rho(y, z)| \leq \epsilon$ with probability at least $1 - \exp(-t^2 k)$.*

Note that the statement of this result in Gao u. a. (2019) is slightly different since they don't specify the dependence of the sample complexity on the failure probability

parameter t. Our version here is easily obtained by using Corollary 5.50 from Vershynin (2010) to include the parameter t.

Now we establish that (X_0, U_0) and X_1 have a sufficient gap in their canonical correlations, with high probability.

Lemma 33. *Let w denote (x_0, u_0). Let Z and W be the sample matrices of z_1 and w, respectively. Let $\mathcal{E}$ denote the event that $|\rho(Z, W) - \rho(z, w)| \leq (1-\rho)/2$. Under Assumption 2, there exists a constant C_0 such that if $n \geq C_0 \log(2d+l)/(1-\rho)^2$, then $P(\mathcal{E}) \geq 0.99$.*

Proof. Set $t = \sqrt{2\log(10)/(2d+l)}$ and note that $1 - \exp(-t^2(2d+l)) = 0.99$. Let $\epsilon = (1-\rho)/2$ and note that

$$t^2(2d+l)\log^2(2d+l)/\epsilon^2 = 8\log 10 \log^2(2d+l)/(1-\rho)^2.$$

Then by Lemma 32 applied to z_1 and w with the above ϵ, there exists a constant C such that if $n \geq (8\log 10)C\log^2(2d+l)/(1-\rho)^2$, then $|\rho(Z,W) - \rho(z_1, w)| \leq (1-\rho)/2$ with probability at least 0.99. □

We can now prove Theorem 6.

Proof. As before, let x denote x_1, and let X denote the corresponding sample matrix. We also have w, W, and Z as before. Condition on the event that $|\rho(Z,W) - \rho(z_1, w)| \leq (1-\rho)/2$, which has probability at least 0.99 by Lemma 33. Then $\rho(Z,W) \leq \rho(z_1, w) + (1-\rho)/2 \leq (1+\rho)/2$.

Define $C = \Sigma_X^{-1/2}\Sigma_{XW}\Sigma_W^{-1}\Sigma_{WX}\Sigma_X^{-1/2}$, and let $\lambda_1 \geq \cdots \geq \lambda_d$ be its eigenvalues with corresponding orthonormal eigenvectors $c_1, \ldots, c_d$. Just as in the population case, the eigenvalues of C are the squared empirical canonical correlations between X and W, and the eigenvectors give the corresponding canonical correlation directions. Note that

$\Pi_{\mathcal{V}} X$ has perfect linear correlation with W. Using the same arguments from Lemma 31, we have that the top r eigenvalues of C are equal to 1 and $\lambda_{r+1} \leq (1+\rho)^2/4$. Moreover, $\text{span}\{\Sigma_X^{-1/2} c_i : 1 \leq i \leq r\} = \mathcal{V}$.

Note that $(1-\rho^2)/4 \leq (3-2\rho-\rho^2)/4 = 1-(1+\rho)^2/4$. Hence, we have $\lambda \leq 1-(1+\rho)^2/4$. Now let (P, Q, D) be a second-order stationary point of (5.4). We claim that the proof of Theorem 5 carries through exactly the same if we replace the population objective function with the corresponding finite sample version (the argument is identical, we just need to replace every covariance and cross-covariance matrix with the appropriate empirical version – there are no spurious correlations to deal with). Hence, we can make use of that result here, and conclude that P has rank r and $\text{col}(\Sigma_X^{1/2} P^\top) = \text{span}\{c_1, \ldots, c_r\}$. Combining this with the fact that $\text{span}\{\Sigma_X^{-1/2} c_i : 1 \leq i \leq r\} = \mathcal{V}$ completes the proof. □

5.4 The Inverse Model

In this section, we focus on the inverse model, whose goal is to predict action based on the state representations. We show that this approach efficiently learns the linear state representation in our hidden subspace model when certain assumptions are satisfied. We also study the sample complexity of this problem when we only have i.i.d. samples from the model. In the appendix we study a simplified version of the model where there is noise.

Recall the inverse model objective

$$\min_{\theta} \frac{1}{2} \mathbb{E} \sum_{i=1}^{r} \|P x_i - L_i x_0 - \sum_{k=1}^{i-1} T_k u_{i-1-k} - u_{i-1}\|_2^2 \tag{5.5}$$

Here, θ is the tuple of parameters $(P, L_1, \ldots, L_r, T_1, \ldots, T_{r-1})$ with $P, L_i \in \mathbb{R}^{l \times d}$ and $T_i \in \mathbb{R}^{l \times l}$, and the expectation is taken over the randomness of $x_0, u_0, \ldots, u_{r-1}$. To

motivate (5.5), we start by considering one step of the dynamics:

$$x_1 = y_1 + z_1 = Ay_0 + Bu_0 + z_1 = Ax_0 + Bu_0 + z_1.$$

If B has full column rank, then we have $B^+B = I$ and $B^+z_1 = 0$ since the rows of B^+ are in $\mathcal{V}$. Hence, we have $u_0 = B^+x_1 - B^+Ax_0$. This expression suggests that if we fit a linear model to predict u_0 given x_0 and x_1, the solution may allow us to recover B^+ and B^+A, both of which reveal part of the latent subspace $\mathcal{V}$. Advancing the system up to timestep i, we have a similar relationship:

$$u_{i-1} = B^+x_i - B^+A^ix_0 - \sum_{k=1}^{i-1} B^+A^kBu_{i-1-k}.$$

Once again, if we fit a linear model to predict u_{i-1} from $x_i, x_0, u_0, \ldots, u_{i-2}$, then we can recover more of $\mathcal{V}$.

Trying to solve for A and B directly by minimizing a squared error loss based on the above expression is problematic, given the presence of high powers of A and products between A, B, and B^+. The optimization landscape corresponding to such an objective function is non-convex and ill-conditioned. To circumvent this issue, we propose the *convex relaxation*:

$$u_{i-1} = Px_i - L_ix_0 - \sum_{k=1}^{i-1} T_ku_{i-1-k}$$

Here, P corresponds to B^+, L_i to B^+A^i, and T_k to B^+A^kB. We arrive at (5.5) by fitting this inverse model over a trajectory of length r, which is chosen so that we can recover the entirety of $\mathcal{V}$.

In order to state our theoretical guarantees for this approach, we introduce a few assumptions.

Assumption 3 (No Linear Dependence)**.** *Let h_0 and $u_0, \ldots, u_{i-1}$ be independent standard Gaussian vectors. Then $\mathbb{E}[z_i] = 0$ and there is a constant $0 \leq \rho < 1$ such that for each $i = 1, \ldots, r$, $\rho((z_i, z_0), (h_i, h_0)) \leq \rho$.*

Remark 1. *Assumption 3 concretely specifies the nonlinearity of z_i, as it precludes any linear dependence between z_i and the controls. Without this assumption, the inverse model that we learn may use information from $\mathcal{V}^\perp$ to predict the controls, and it is impossible to uniquely recover $\mathcal{V}$.*

Assumption 4 (Controllability)**.** *The tuple $(A^\top, (B^+)^\top)$ is $\mathcal{V}$-controllable.*

Remark 2. *Assumption 4 is related to the standard controllability property of linear control systems. Instead of assuming (A, B) controllability, we need the property to hold for $(A^\top, (B^+)^\top)$ since we are learning an inverse model which is related to the matrices $B^+, B^+A, \ldots, B^+A^r$.*

Assumption 5 (Non-degeneracy)**.** *,The matrix B has linearly independent columns, i.e. rank$(B) = l$.*

Remark 3. *Assumption 5 allows us to learn the inverse model. If B is rank-deficient, we could not hope to predict even u_0 from x_0 and x_1, since it is non-identifiable. One interpretation of this assumption is that the control inputs u_i are well-specified, i.e., not redundant.*

Observe that (5.5) is a convex optimization problem, but there may not be a unique global minimizer due to redundancies in the parametrization. While the set of global optimizers is in general a linear subspace, by imposing certain norm preferences we can still recover the intended solutions B^+, B^+A_i and B^+A^kB. We now state the theoretical guarantee for our algorithm.

Theorem 7. *Let f be the objective function in* (5.5)*, and let*

$$\Theta_0^* = \{\theta = (P, \{L_i\}_{i=1}^r, \{T_i\}_{i=1}^{r-1}) \in f^{-1}(0) \mid \|P\|_F \text{ is minimal}\}$$

be the set of optimal solutions to (5.5) *that have minimal norm for P. Let $\theta^* = (P^*, \{L_i^*\}, \{T_i^*\}) \in \Theta_0^*$ be the solution in this set that minimizes $\sum_{i=1}^r \|L_i\|_F^2$. Then under assumptions 3, 4, and 5, $P = B^+$ and $L_i = B^+A^i$ for $i = 1, \ldots, r$. Moreover, $\mathcal{V} = col(P^\top) + col(L_1^\top) + \cdots + col(L_r^\top)$.*

Remark 4. *To find the desired solution, we can first find the set $\Theta^* = f^{-1}(0)$ of global minimizers. For such linear systems, Θ^* is a subspace, so Θ_0^* can be obtained by optimizing for the norm of P within this subspace.*

Intuitively, Theorem 7 is correct because by Assumption 3, any direction in $\mathcal{V}^\perp$ will not have a perfect linear correlation with the control signal u_i that we are trying to predict. This does not mean that every optimal solution to Equation (5.5) has components only in $\mathcal{V}$ – it is still possible that components in $\mathcal{V}^\perp$ cancel each other. However, if any of the matrices P, L_i have components in $\mathcal{V}^\perp$, removing those components will reduce the norm of the matrices while not changing the predictive accuracy. Therefore the minimum norm solution must lie in the correct subspace. Finally, the fact that we recover the entirety of $\mathcal{V}$ follows from Assumption 4.

Our proof makes this intuition precise by analyzing the first-order optimality conditions of (5.5) and making use of a spectral characterization of Assumption 3. Observe that there is a mismatch between Assumption 3 and our objective function (5.5): in the objective (5.5), we try to enforce a linear relationship between $x_i, x_0, u_1, u_2, ..., u_{i-1}$, while Assumption 3 is about (h_i, h_0) and (z_i, z_0). The following lemma helps relate the two.

Lemma 34. *Let $i \in \{1, \ldots, r\}$. Let $\tilde{h}_i = (h_i, h_0, u_0, \ldots, u_{i-2})$ and let $\tilde{z}_i = (z_i, z_0)$. Then*

$$\rho(\tilde{h}_i, \tilde{z}_i) \leq \rho((h_i, h_0), (z_i, z_0)).$$

Proof. Note that the definition of $\tilde{h}_i$ doesn't make sense for $i = 1$. In that case, define $\tilde{h}_1 = (h_1, h_0)$. Let $u = (u_0, u_1, \ldots, u_{i-1})$. Observe that the coordinates of $\tilde{h}_i$ are a subset of the coordinates of (h_i, h_0, u), so $\rho(\tilde{h}_i, \tilde{z}_i) \leq \rho((h_i, h_0, u), \tilde{z}_i)$. Note that there exist matrices P and Q such that $h_i = Ph_0 + Qu$. Let $a_1, a_2 \in \mathbb{R}^r$, $b_1, b_2 \in \mathbb{R}^d$, and $a_3 \in \mathbb{R}^{il}$. Write $a_3 = Q^\top v_1 + v_2$, where $Qv_2 = 0$. Note that u is independent of h_0 and $\langle v_2, u \rangle$ is independent of each coordinate of h_i (as these are Gaussian random vectors). Then we have

$$\begin{aligned}
\mathbb{E}[(\langle a_1, h_i \rangle + \langle a_2, h_0 \rangle + \langle a_3, u \rangle)^2] &= \mathbb{E}[(\langle a_1, h_i \rangle + \langle a_2, h_0 \rangle + \langle Q^\top v_1, u \rangle \\
&\qquad + \langle v_2, u \rangle + \langle P^\top v_1, h_0 \rangle - \langle P^\top v_1, h_0 \rangle)^2] \\
&= \mathbb{E}[(\langle a_1 + v_1, h_i \rangle + \langle a_2 - P^\top v_1, h_0 \rangle + \langle v_2, u \rangle)^2] \\
&= \mathbb{E}[(\langle a_1 + v_1, h_i \rangle + \langle a_2 - P^\top v_1, h_0 \rangle)^2] + \mathbb{E}[\langle v_2, u \rangle^2]
\end{aligned}$$

Now u is independent of z_0, so $\mathbb{E}[\langle v_2, u \rangle \langle b_2, z_0 \rangle] = 0$. Moreover, u and z_i are conditionally independent given h_i, so we have

$$\begin{aligned}
\mathbb{E}[\langle v_2, u \rangle (\langle b_1, z_i \rangle + \langle b_2, z_0 \rangle)] &= \mathbb{E}[\langle v_2, u \rangle \langle b_1, z_i \rangle] \\
&= \mathbb{E}[\mathbb{E}[\langle v_2, u \rangle \langle b_1, z_i \rangle | h_i]] \\
&= \mathbb{E}[\mathbb{E}[\langle v_2, u \rangle | h_i] \mathbb{E}[\langle b_1, z_i \rangle | h_i]] \\
&= \mathbb{E}[\mathbb{E}[\langle v_2, u \rangle] \mathbb{E}[\langle b_1, z_i \rangle | h_i]] \\
&= 0.
\end{aligned}$$

Then we have

$$\begin{aligned}
&\frac{\mathbb{E}[(\langle a_1, h_i\rangle + \langle a_2, h_0\rangle + \langle a_3, u\rangle)(\langle b_1, z_i\rangle + \langle b_2, z_0\rangle)]}{\sqrt{\mathbb{E}[(\langle a_1, h_i\rangle + \langle a_2, h_0\rangle + \langle a_3, u\rangle)^2]\mathbb{E}[(\langle b_1, z_i\rangle + \langle b_2, z_0\rangle)^2]}} \\
&\quad \leq \frac{\mathbb{E}[(\langle a_1 + v_1, h_i\rangle + \langle a_2 - P^\top v_1, h_0\rangle)(\langle b_1, z_i\rangle + \langle b_2, z_0\rangle)}{\sqrt{\mathbb{E}[(\langle a_1 + v_1, h_i\rangle + \langle a_2 - P^\top v_1, h_0\rangle)^2]\mathbb{E}[(\langle b_1, z_i\rangle + \langle b_2, z_0\rangle)^2]}} \\
&\quad \leq \rho((h_i, h_0), (z_i, z_0)).
\end{aligned}$$

□

We are ready to prove Theorem 7:

Proof. The main idea of the proof is to derive conditions for the variables based on first-order optimality conditions. We first prove that the optimal variables have support only on the linearizing subspace. As a consequence, we can then show that these variables equal the true model parameters.

To start, fix $i \in \{1, \ldots, r\}$, define $\theta_i = [P\ L_i\ T_1\ \cdots\ T_{i-1}]$, and let

$$\begin{aligned}
\tilde{y}_i &= (y_i, -y_0, -u_{i-2}, \ldots, -u_0) \\
\tilde{h}_i &= (h_i, -h_0, -u_{i-2}, \ldots, -u_0) \\
\tilde{z}_i &= (z_i, -z_0).
\end{aligned}$$

Define $K = [I\ 0]^\top$ to be the block matrix that satisfies $\theta_i K = [P\ L_i]$. Define $\tilde{V} = \mathrm{diag}(V, V, I, \ldots, I)$ to be the block diagonal matrix that satisfies $\tilde{y}_i = \tilde{V}\tilde{h}_i$, and note that $\tilde{V}$ has full column rank. Observe that there exists a matrix M such that $u_{i-1} = M\tilde{h}_i$.

We can now express the objective function as $f(\theta) = \sum_{i=1}^{\tau} f_i(\theta_i)$, where $f_i(\theta_i) = \frac{1}{2}\mathbb{E}\|\theta_i(K\tilde{z}_i + \tilde{V}\tilde{h}_i) - u_{i-1}\|_2^2$. Since each f_i has minimal value 0, any optimal point for f must simultaneously optimize each f_i. Hence, $\nabla f(\theta_i) = 0$ is a necessary condition

for optimality. To this end, we compute the gradient of f_i as

$$\nabla f_i(\theta_i) = \theta_i(K\Sigma_{\tilde{z}_i\tilde{z}_i}K^\top + \tilde{V}\Sigma_{\tilde{h}_i\tilde{h}_i}\tilde{V}^\top + \tilde{V}\Sigma_{\tilde{h}_i\tilde{z}_i}K^\top + K\Sigma_{\tilde{z}_i\tilde{h}_i}\tilde{V}^\top) - \Sigma_{u_{i-1}\tilde{z}_i}K^\top - \Sigma_{u_{i-1}\tilde{h}_i}\tilde{V}^\top$$

We split the optimality condition according to orthogonal subspaces V and $V^\perp$ to obtain

$$0 = \theta_i(\tilde{V}\Sigma_{\tilde{h}_i\tilde{h}_i}\tilde{V}^\top + K\Sigma_{\tilde{z}_i\tilde{h}_i}\tilde{V}^\top) - \Sigma_{u_{i-1}\tilde{h}_i}\tilde{V}^\top \tag{5.6}$$

$$0 = \theta_i(K\Sigma_{\tilde{z}_i\tilde{z}_i}K^\top + \tilde{V}\Sigma_{\tilde{h}_i\tilde{z}_i}K^\top) - \Sigma_{u_{i-1}\tilde{z}_i}K^\top \tag{5.7}$$

From (5.6), we have $\theta_i\tilde{V}\Sigma_{\tilde{h}_i\tilde{h}_i} = \Sigma_{u_{i-1}\tilde{h}_i} - \theta_i K\Sigma_{\tilde{z}_i\tilde{h}_i}$, and plugging this into (5.7) (while also clearing $K^\top$ by right-multiplying the equation by K) gives

$$\begin{aligned}
0 &= \theta_i K\Sigma_{\tilde{z}_i\tilde{z}_i} + \theta_i\tilde{V}\Sigma_{\tilde{h}_i\tilde{h}_i}\Sigma^+_{\tilde{h}_i\tilde{h}_i}\Sigma_{\tilde{h}_i\tilde{z}_i} - \Sigma_{u_{i-1}\tilde{z}_i} \\
&= \theta_i K\Sigma_{\tilde{z}_i\tilde{z}_i} - \theta_i K\Sigma_{\tilde{z}_i\tilde{h}_i}\Sigma^+_{\tilde{h}_i\tilde{h}_i}\Sigma_{\tilde{h}_i\tilde{z}_i} - \Sigma_{u_{i-1}\tilde{z}_i} + \Sigma_{u_{i-1}\tilde{h}_i}\Sigma^+_{\tilde{h}_i\tilde{h}_i}\Sigma_{\tilde{h}_i\tilde{z}_i} \\
&= \theta_i K(\Sigma_{\tilde{z}_i\tilde{z}_i} - \Sigma_{\tilde{z}_i\tilde{h}_i}\Sigma^+_{\tilde{h}_i\tilde{h}_i}\Sigma_{\tilde{h}_i\tilde{z}_i}) - M\Sigma_{\tilde{h}_i\tilde{z}_i} + M\Sigma_{\tilde{h}_i\tilde{h}_i}\Sigma^+_{\tilde{h}_i\tilde{h}_i}\Sigma_{\tilde{h}_i\tilde{z}_i} \\
&= \theta_i K\Sigma_{\tilde{z}_i\tilde{z}_i}(I - \Sigma^+_{\tilde{z}_i\tilde{z}_i}\Sigma_{\tilde{z}_i\tilde{h}_i}\Sigma^+_{\tilde{h}_i\tilde{h}_i}\Sigma_{\tilde{h}_i\tilde{z}_i}).
\end{aligned}$$

By Lemma 34 and Assumption 3, we have that $I - \Sigma^+_{\tilde{z}_i\tilde{z}_i}\Sigma_{\tilde{z}_i\tilde{h}_i}\Sigma^+_{\tilde{h}_i\tilde{h}_i}\Sigma_{\tilde{h}_i\tilde{z}_i}$ is nonsingular, so we conclude that $\theta_i K\Sigma_{\tilde{z}_i\tilde{z}_i} = 0$. In particular, this implies that $\theta_i K\Sigma_{\tilde{z}_i\tilde{y}_i} = 0$.

We can now simplify (5.6) as $0 = \theta_i\tilde{V}\Sigma_{\tilde{h}_i\tilde{h}_i}\tilde{V}^\top - \Sigma_{u_{i-1}\tilde{h}_i}\tilde{V}^\top = \theta_i\Sigma_{\tilde{y}_i\tilde{y}_i} - \Sigma_{u_{i-1}\tilde{y}_i}$. This matrix equation can be naturally partitioned into blocks according to the block partition of θ_i and $\tilde{y}_i$. Reading out the second block column gives $0 = -PA^i + L_iVV^\top$. Reading out the $(k+1)$-st block column (for $k \geq 1$) gives $0 = -PA^kB + T_k$. The first

block column gives

$$\begin{aligned}
0 &= P\Sigma_{y_i y_i} - L_i \Sigma_{y_0 y_i} - \sum_{k=1}^{i-1} T_k \Sigma_{u_{i-1-k} y_i} - \Sigma_{u_{i-1} y_i} \\
&= P(A^i (A^i)^\top + \sum_{k=1}^{i-1} A^k B (A^k B)^\top + BB^\top) - L_i (A^i)^\top - \sum_{k=1}^{i-1} T_k (A^k B)^\top - B^\top \\
&= (PB - I)B^\top + (PA^i - L_i)(A^i)^\top + \sum_{k=1}^{i-1} (PA^k B - T_k)(A^k B)^\top \\
&= (PB - I)B^\top.
\end{aligned}$$

Using Assumption 5, we right-multiply by $(B^+)^\top$ to obtain $PB = I$. Since P is the minimal-norm optimal solution, we conclude that $P = B^+$. Then $L_i V V^\top = B^+ A^i$ and $T_k = B^+ A^k B$. Since we are also minimizing the norm of L_i, we see that L_i must vanish on $V^\perp$, so that $L_i = L_i V V^\top$, and $L_i = B^+ A^i$. That we recover all of V is a consequence of Assumption 4. □

5.4.1 Sample Complexity

As with the forward model, in practice we can only solve the empirical inverse model objective

$$\min_\theta \frac{1}{2n} \sum_{i=1}^{r} \|PX_i - L_i X_0 - \sum_{k=1}^{i-1} T_k U_{i-1-k} - U_{i-1}\|_F^2 \tag{5.8}$$

Here, the columns of $X_i \in \mathbb{R}^{d \times n}, U_i \in \mathbb{R}^{l \times n}$ are i.i.d. copies of x_i and u_i, respectively. We again introduce an assumption that allows us to utilize quantitative concentration results, and then we state the sample complexity result.

Assumption 6 (Sub-Gaussianity)**.** *There exists a constant $C > 0$ such that for each $i \in \{1, \ldots, r\}$, $P(|\langle q, \Sigma_{\xi_i \xi_i}^{-1/2} \xi_i \rangle| > t) \leq \exp(-Ct^2)$ for any unit vector q, where we define $\xi_i := (z_i, z_0, h_i, h_0)$.*

Theorem 8. *Let f be the objective function in* (5.8)*, and let*

$$\Theta_0^* = \{\theta = (P, \{L_i\}_{i=1}^r, \{T_i\}_{i=1}^{r-1}) \in f^{-1}(0) \mid \|P\|_F \text{ is minimal}\}$$

be the set of optimal solutions to (5.5) *that have minimal norm for P. Let $\theta^* = (P^*, \{L_i^*\}, \{T_i^*\}) \in \Theta_0^*$ be the solution in this set that minimizes $\sum_{i=1}^r \|L_i\|_F^2$. Under assumptions 3, 4, 5, and 6, there exists a constant C_0 such that if $n \geq C_0(d + rl)\log r \log^2(d+rl)/(1-\rho)^2$, then with probability at least* 0.99*, $P = B^+$ and $L_i = B^+A^i$ for $i = 1, \ldots, r$.*

Our proof for Theorem 8 is similar to that of Theorem 7 but requires somewhat more care. We use additional concentration of measure results in order to ensure that the empirical canonical correlation $\rho((Z_i, Z_0), (H_i, H_0))$ is close to its population value. The first-order optimality conditions of (5.8) also contain additional empirical cross-covariance terms that must be handled.

We will again make use of Lemma 32, as well as the following standard matrix concentration inequality.

Lemma 35 (From Corollary 5.35 of Vershynin (2010))**.** *Let $Y \in \mathbb{R}^{k \times n}$ be a matrix whose entries are independent standard Gaussian random variables. Then for every $t \geq 0$, with probability at least $1 - 2\exp(-t^2/2)$ it holds that*

$$\sqrt{n} - \sqrt{k} - t \leq \sigma_{min}(Y).$$

We now establish that the certain empirical canonical correlations and covariance matrices are sufficiently close to their population counterparts.

Lemma 36. *Let $\tilde{Z}_i$ and $\tilde{H}_i$ be the sample matrices of $\tilde{z}_i$ and $\tilde{h}_i$, respectively (from the proof of Theorem 7). Further define $\hat{H}_i$ to be the sample matrix for the random vector*

$\hat{h}_i := (h_0, u_{i-1}, u_{i-2}, \ldots, u_0)$. *Let* $\mathcal{E}_i$ *denote the event that* $|\rho(\tilde{H}_i, \tilde{Z}_i) - \rho(\tilde{h}_i, \tilde{z}_i)| \leq (1-\rho)/2$. *Let* $\mathcal{F}_i$ *denote the event that* $\sigma_{min}(\hat{H}_i) \geq 1/2$. *There exists a constant* C_0 *such that if* $n = C_0(d+rl)\log r \log^2(d+rl)/(1-\rho)^2$, *then*

$$P\left(\bigcap_{i=1}^{r} \mathcal{E}_i \cap \mathcal{F}_i\right) \geq 0.99.$$

Proof. Set the failure probability parameter $t = C'\sqrt{\log r}$, where C' is a large enough constant such that

$$r(\exp(-t^2(2d+2r)) + 2\exp(-t^2/2)) \leq 0.01.$$

Let C be the constant from Lemma 32 applied to $\tilde{h}_i$ and $\tilde{z}_i$ with $\epsilon = (1-\rho)/2$ – we can take the same C for each i since we assume each (h_i, z_i) satisfy the same sub-Gaussian property. Set C_0 large enough so that when $n = C_0(d+rl)\log r \log^2(d+rl)/(1-\rho)^2$, the following hold for $i = 1, \ldots, r$:

$$\begin{aligned} n &\geq 4Ct^2(2d+2r+(i-2)l)\log^2(2d+2r+(i-2)l)/(1-\rho)^2, \\ \sqrt{n} &\geq 1/2 + \sqrt{r+(i-1)l} + t \end{aligned}$$

We first analyze $P(\mathcal{E}_i)$. Apply Lemma 32 to $\tilde{h}_i \in \mathbb{R}^{2r+(i-2)l}$ and $\tilde{z}_i \in \mathbb{R}^{2d}$ with $\epsilon = (1-\rho)/2$ and the specified value of t. Then we see that n is large enough to ensure that $P(\mathcal{E}_i) \geq 1 - \exp(-t^2(2d+2r+(i-2)l)) \geq 1 - \exp(-t^2(2d+2r))$.

Next, consider $P(\mathcal{F}_i)$. Apply Lemma 35 to $\hat{H}_i$ with the specified value of t. Again it is clear that n is large enough to ensure that $P(\mathcal{F}_i) \geq 1 - 2\exp(-t^2/2)$.

Finally, by the union bound,

$$\begin{aligned} P\left(\bigcap_{i=1}^{r} \mathcal{E}_i \cap \mathcal{F}_i\right) &\geq 1 - \sum_{i=1}^{r} (2 - P(\mathcal{E}_i) + P(\mathcal{F}_i)) \\ &\geq 1 - r(\exp(-t^2(2d+2r)) + 2\exp(-t^2/2)) \\ &\geq 0.99. \end{aligned}$$

□

We now prove Theorem 8.

Proof. Lemma 36 provides the sample complexity and success probability – all that's left is to analyze the empirical loss assuming that the conclusion of Lemma 36 holds. Our analysis of the empirical loss is close to that of the population loss. We use the same notation as in the proof of Theorem 7, e.g. $\tilde{Y}_i, \tilde{H}_i, \tilde{U}_i, \tilde{Z}_i$ are the sample matrices of $\tilde{y}_i, \tilde{h}_i, \tilde{u}_i, \tilde{z}_i$, respectively. Likewise, define θ_i, K, and $\tilde{V}$ as before. We additionally define $\hat{H}_i$ to be the sample matrix for $(h_0, u_{i-1}, u_{i-2}, \ldots, u_0)$.

By the same argument as in the proof of Theorem 7, we have that

$$0 = \theta_i K \Sigma_{\tilde{Z}_i \tilde{Z}_i} (I - \Sigma^{+}_{\tilde{Z}_i \tilde{Z}_i} \Sigma_{\tilde{Z}_i \tilde{H}_i} \Sigma^{+}_{\tilde{H}_i \tilde{H}_i} \Sigma_{\tilde{H}_i \tilde{Z}_i}).$$

The spectral norm of $-\Sigma^{+}_{\tilde{Z}_i \tilde{Z}_i} \Sigma_{\tilde{Z}_i \tilde{H}_i} \Sigma^{+}_{\tilde{H}_i \tilde{H}_i} \Sigma_{\tilde{H}_i \tilde{Z}_i}$ is $\rho(\tilde{H}_i, \tilde{Z}_i)$, and by assumption and Lemma 36, we have

$$\rho(\tilde{H}_i, \tilde{Z}_i) \leq \rho(\tilde{h}_i, \tilde{z}_i) + (1-\rho)/2 \leq (1+\rho)/2 < 1.$$

Hence, $(I - \Sigma^{+}_{\tilde{Z}_i \tilde{Z}_i} \Sigma_{\tilde{Z}_i \tilde{H}_i} \Sigma^{+}_{\tilde{H}_i \tilde{H}_i} \Sigma_{\tilde{H}_i \tilde{Z}_i})$ is robustly nonsingular, so we conclude that $\theta_i K \Sigma_{\tilde{Z}_i \tilde{Z}_i} = 0$ and likewise $\theta_i K \Sigma_{\tilde{Z}_i \tilde{Y}_i} = 0$.

Using this fact, we can continue to follow the proof of Theorem 7 to obtain

$$0 = \theta_i \tilde{V} \Sigma_{\tilde{H}_i \tilde{H}_i} \tilde{V}^\top - \Sigma_{U_{i-1} \tilde{H}_i} \tilde{V}^\top.$$

Analyzing this equation is slightly more complicated now due to the fact that sample cross-covariance terms like $\Sigma_{H_0 U_j}$ are nonzero (whereas the corresponding population covariances vanish due to independence). By splitting the equation into block columns, grouping terms, and simplifying the terms that cancel, it is straightforward to see that

$$0 = [(PA^i - L_i)\ (PB - I)\ (PAB - T_1)\ \cdots\ (PA^{i-1}B - T_{i-1})] \tilde{V} \Sigma_{\hat{H}_i \hat{H}_i} \tilde{V}^\top.$$

By assumption, $\sigma_{min}(\hat{H}_i) \geq 1/2$, so $\Sigma_{\hat{H}_i \hat{H}_i}$ is robustly nonsingular. Hence, we have that

$$0 = [(PA^i - L_i)\ (PB - I)\ (PAB - T_1)\ \cdots\ (PA^{i-1}B - T_{i-1})] \tilde{V},$$

which implies that $PB = I$ and $(PA^i - L_i)V = 0$ for all i. Since we assume P has minimal norm, we conclude that $P = B^+$. Thus, $L_i V = B^+ A^i V$, i.e. $L_i = B^+ A^i$ on the subspace V. By the construction of our minimal norm solution, we know that L_i must vanish on $V^\perp$, and this completes the proof. □

5.4.2 Handling Noise in the Model

We now consider a simple version of our model with noise, and show that our algorithm identifies the correct subspace (up to an error proportional to the noise) in this setting as well. We consider a one-step trajectory where the initial state $x_0 = 0$, and we assume that our observation is corrupted by independent centered noise. In particular, we can write the state observation as $x = Bu + z + \xi$, where ξ is a random vector in $\mathbb{R}^d$ that is independent of both u and z. Assume the noise covariance matrix $\Sigma_{\xi\xi}$

splits orthogonally along the subspace V and $V^\perp$, that is, we can write $\Sigma_{\xi\xi} = \Sigma_1 + \Sigma_2$, where Σ_1 is the covariance of the noise projected onto V and Σ_2 is the covariance of the noise projected onto the column-span of $V^\perp$. This orthogonal splitting is satisfied when ξ is a spherical Gaussian random vector, for example.

Given this noisy state observation x and control input u, the task is to recover the column-span of B by learning a linear inverse model:

$$\min_P \frac{1}{2}\mathbb{E}_{u,\xi}\|Px - u\|_2^2 \tag{5.9}$$

Due to the noise term, this linear model will not achieve zero error. However, we can bound the error of our solution as a function of the noise magnitude and correlation bound.

Theorem 9. *Let $u \in \mathbb{R}^l$ and $\xi \in \mathbb{R}^d$ be independent spherical Gaussian random vectors, with $\Sigma_{\xi\xi} = \sigma^2 I$. Let P be the minimal norm optimal solution to the optimization problem (5.9). Write $P = P_1 + P_2$, where P_1 is the projection of P onto V, and P_2 is its projection onto $V^\perp$. In the noisy setting described above, we have $P_1 = B^+$ and*

$$\|P_2\|_2 \leq \frac{\sigma\rho}{2\sqrt{1-\rho^2}}\|B^+\|_2\|P_1\|_2$$

where $\sigma = \lambda_{max}(\Sigma_{\xi\xi})$ and $\rho := \rho(u, z)$.

Note that ideally we want $P_2 = 0$, since its rows are in $V^\perp$. This theorem says that the spectral norm of P_2 is small compared to P_1, which allows us to approximately recover B^+.

Proof. In this setting, the optimality conditions of (5.9) take the form

$$0 = B\Sigma_{uu}(B^\top P_1 - I) + B\Sigma_{uz}P_2 + \sigma^2 P_1 \tag{5.10}$$

$$0 = \Sigma_{zu}(B^\top P_1 - I) + \Sigma_{zz}P_2 + \sigma^2 P_2, \tag{5.11}$$

Multiplying (5.10) by $\Sigma_{zu}(B\Sigma_{uu})^+$ and subtracting (5.11) yields the following identity (after simplification):

$$(\sigma^2 I + \Sigma_{zz} - \Sigma_{zu}\Sigma_{uu}^{-1}\Sigma_{uz})P_2 = \sigma^2\Sigma_{zu}(B\Sigma_{uu})^+ P_1. \tag{5.12}$$

Let Q_z be the (orthogonal) projection onto the column-span of Σ_{zz}, and note that we can write $Q_z = (\Sigma_{zz}^{1/2})^+\Sigma_{zz}^{1/2}$. Define $C = Q_z - (\Sigma_{zz}^{1/2})^+\Sigma_{zu}\Sigma_{uu}^{-1}\Sigma_{uz}(\Sigma_{zz}^{1/2})^+$. Note that $(\Sigma_{zz}^{1/2})^+\Sigma_{zu}\Sigma_{uu}^{-1}\Sigma_{uz}(\Sigma_{zz}^{1/2})^+$ has maximal eigenvalue ρ^2. Then C has column-span equal to that of Σ_{zz}, with minimal nonzero singular value equal to $1-\rho^2$.

Set $\Gamma = (\sigma^{-1}\Sigma_{zz}^{1/2}C^{1/2})^+ + (\sigma^{-1}\Sigma_{zz}^{1/2}C^{1/2})^\top$. Based on the properties of C that we established, it is evident that Γ has column-span equal to that of Σ_{zz}, and it has minimal singular value bounded below by 2 by Lemma 37. We have

$$\begin{aligned}
P_2 &= (C^{1/2}\Gamma)^+ C^{1/2}\Gamma P_2 \\
&= (C^{1/2}\Gamma)^+\sigma^{-1}(\Sigma_{zz}^{1/2})^+\left(\sigma^2 I + \Sigma_{zz} - \Sigma_{zu}\Sigma_{uu}\Sigma_{uz}\right)P_2 \\
&= (C^{1/2}\Gamma)^+\sigma^{-1}(\Sigma_{zz}^{1/2})^+\sigma^2\Sigma_{zu}\Sigma_{uu}^{-1}B^+P_1 \\
&= \sigma\Gamma^+(C^{1/2})^+((\Sigma_{zz}^{1/2})^+\Sigma_{zu}\Sigma_{uu}^{-1/2})\Sigma_{uu}^{-1/2}B^+P_1.
\end{aligned}$$

Now $(\Sigma_{zz}^{1/2})^+\Sigma_{zu}\Sigma_{uu}^{-1/2}$ must have maximal singular value equal to ρ, since it gives a symmetric low-rank factorization of $(\Sigma_{zz}^{1/2})^+\Sigma_{zu}\Sigma_{uu}^{-1}\Sigma_{uz}(\Sigma_{zz}^{1/2})^+$. Hence, we finally have

the bound

$$\|P_2\|_2 \leq \frac{\sigma\rho}{2\sqrt{1-\rho^2}}\|\Sigma_{uu}^{-1/2}B^+\|_2\|P_1\|_2.$$

□

Lemma 37. *For any matrix A, the minimal nonzero singular value of $A^+ + A^\top$ is at least 2.*

Proof. Write the compressed SVD of A^+ as $U\Sigma V^\top$, and note that we can write $A^\top = U\Sigma^{-1}V^\top$. It is then evident that the non-zero singular values of $A^+ + A^\top$ are of the form $x + x^{-1}$ for $x > 0$. But $x + x^{-1} \geq 2$ for all $x > 0$. □

5.5 Nonlinear State Representation Learning

In this section, we extend the forward model and inverse model objectives to the setting in which there are no latent linear dynamics in the original state observations. In this case, we try to learn a nonlinear state representation ϕ under which the dynamics are nearly linear. An example of this setting is when the state observations are raw pixels from a camera and ϕ is a convolutional neural network.

For the forward model objective, we introduce an intermediate feature map $\psi : \mathbb{R}^d \to \mathbb{R}^{d'}$ and fit the forward model to the transformed states $\psi(x_0), \psi(x_1)$. The resulting optimization problem is

$$\min_\theta \mathbb{E}\|P\psi(x_1) - Q\psi(x_0) - Du_0\|_2^2 + \lambda\|P\Sigma_{\psi(x_1)}P^\top - I\|_F^2 \tag{5.13}$$

where θ is the tuple of parameters (ψ, P, Q, D). The final state representation map ϕ is given by $\phi(x) = P\psi(x)$.

For the inverse model objective, we again simply fit the inverse model to the

transformed states $\psi(x_i)$:

$$\min_{\theta} \frac{1}{2}\mathbb{E} \sum_{i=1}^{\tau} \|P\psi(x_i) - L_i\psi(x_0) - \sum_{k=1}^{i-1} T_k u_{i-1-k} - u_{i-1}\|_2^2 \tag{5.14}$$

Although unlikely to be obtained in practice, we verify that if the 0 loss is achieved, then we can extract a nontrivial linear control system.

Theorem 10. *Let $\psi, P, \{L_i, T_i\}, i = 1, \ldots, \tau$ be optimal solutions to the optimization problem* (5.14)*, and assume that these parameters incur zero loss. Define $\mathcal{V} = col(P^\top) + col(L_1^\top) + \cdots + col(L_{\tau-1}^\top)$, and assume that $col(L_\tau^\top) \subset \mathcal{V}$. Let $\phi(x) = \Pi_{\mathcal{V}}\psi(x)$. Then there exist matrices $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times l}$ such that for each x and u,*

$$\phi(x_{t+1}) = A\phi(x_t) + Bu_t.$$

As the theorem indicates, to get the final representation ϕ, we apply ψ followed by the projection $\Pi_{\mathcal{V}}$. The dynamics of ϕ are nontrivial because, as before in the linear case, the control input can be predicted given the initial and current state representations and previous control inputs.

Intuitively, as τ increases, by result of Theorem 7 we can expect that one learns a larger and larger subspace with linear dynamics. Theorem 10 shows that as soon as the dimension of this linear subspace stops increasing at a certain length τ, the dynamics of $\psi(x)$ are linear on this subspace. To prove this, we use the fact that the loss is 0 to obtain the identity

$$P\psi(x_i) = L_i\psi(x_0) + \sum_{k=1}^{i-1} T_k u_{i-1-k} + u_{i-1}$$

for $i = 1, \ldots, \tau$. Notice that if we view the trajectory as starting at x_1, we have

$$P\psi(x_i) = L_{i-1}\psi(x_1) + \sum_{k=1}^{i-2} T_k u_{i-1-k} + u_{i-1}.$$

Combining these identities and simplifying yields $L_{i-1}\psi(x_1) = L_i\psi(x_0) + T_{i-1}u_0$. This shows roughly that ψ has linear dynamics in the directions of L_{i-1} and L_i. We use these facts together with condition $\text{col}(L_\tau^\top) \subset \mathcal{V}$ to show that ψ has (invariant) linear dynamics on all of $\mathcal{V}$.

Proof. For this proof, instead of writing $\psi(x)$ to denote the state representation of x, we simply drop explicit reference to ψ and agree that any system state we discuss has already been mapped to its representation via ψ. This will simplify notation but doesn't change any of the analysis.

Zero loss in the objective function implies that

$$Pf(x, \{u_0, \ldots, u_{i-1}) = L_i x + \sum_{k=0}^{i-2} T_{i-1-k} u_k + u_{i-1}$$

for all $x \in \phi(\mathbb{R}^d)$ and $u_j \in \mathbb{R}^l$, $j = 0, \ldots, i-1$.

Fix $2 \leq i \leq \tau + 1$. By assumption,

$$Pf(x, \{u_0, \ldots, u_{i-1}\}) = L_i x + \sum_{k=0}^{i-2} T_{i-1-k} u_k + u_{i-1}.$$

But we can also express this as follows:

$$\begin{aligned} Pf(x, \{u_0, \ldots, u_{i-1}\}) &= Pf(f(x, u_0), \{u_1, \ldots, u_{i-1}\}) \\ &= L_{i-1} f(x, u_0) + \sum_{k=0}^{i-3} T_{i-2-k} u_{k+1} + u_{i-1} \end{aligned}$$

Equating these two expressions and eliminating like terms gives

$$L_{i-1}f(x,u_0) = L_i x + T_{i-1}u_0.$$

Note that here it is crucial that we couple the T_i matrices. Without the coupling we would not be able to eliminate the terms relating u_i for $i > 0$.

Next, let $\{v_1, \ldots, v_r\}$ be an orthonormal basis for V. Then we can write $Q = \sum_{j=1}^{r} v_j v_j^\top$. Furthermore, by construction, for each v_j, there exist vectors $y_{j,0}, y_{j,1}, \ldots, y_{j,\tau}$ such that $v_j = P^\top y_{j,0} + \sum_{i=1}^{\tau} L_i^\top y_{j,i}$. Notice that for $i = 1, \ldots, \tau+1$, since $\mathrm{col}(L_i^\top) \subset V$, it holds that $L_i = L_i Q$. Then we have

$$\begin{aligned}
Qf(x,u) &= \sum_{j=1}^{r} v_j v_j^\top f(x,u) \\
&= \sum_{j=1}^{r} v_j \left(y_{j,0}^\top P f(x,u) + \sum_{i=1}^{\tau} y_{j,i}^\top L_i f(x,u) \right) \\
&= \sum_{j=1}^{r} v_j \left(y_{j,0}^\top (L_1 x + u) + \sum_{i=1}^{\tau} y_{j,i}^\top (L_{i+1} x + T_i u) \right) \\
&= \left(\sum_{j=1}^{r} \sum_{i=0}^{\tau} v_j y_{j,i}^\top L_{i+1} \right) Qx + \left(\sum_{j=1}^{r} \sum_{i=0}^{\tau} v_j y_{j,i}^\top T_i \right) u
\end{aligned}$$

where we let $T_0 = I$. Now set $A = \sum_{j=1}^{r} \sum_{i=0}^{\tau} v_j y_{j,i}^\top L_{i+1}$ and $B = \sum_{j=1}^{r} \sum_{i=0}^{\tau} v_j y_{j,i}^\top T_i$, and we have our result. □

To solve either (5.13) or (5.14) in practice, we constrain ψ to be in some parametric function class and minimize the empirical version of the objective function induced by a finite sample. Since these problems now involve optimizing the parameters of ψ, they are non-convex and much more difficult to analyze explicitly. In general, we can only hope to obtain small loss rather than 0 loss. It is natural to ask whether we can get any guarantees when we have small but nonzero loss for (5.14). This is a

challenging question to answer theoretically, but empirically we observe that achieving moderately small loss yields reasonable state representations on two simple nonlinear control environments; see Section 5.6.

5.6 Experiments

We conduct simple experiments to numerically validate our theory. We first discuss experiments with synthetic data generated according to our hidden subspace model, and then experiments using standard RL environments with nonlinear dynamics and high-dimensional observations.

5.6.1 Synthetic Experiments

CCA Objective We generate 1000 i.i.d. samples from the model $u \in \mathbb{R}^5 \sim N(0, I)$ and $v = u + \epsilon$, where $\epsilon \sim N(0, \text{diag}(0, .25, .5, .75, 1))$. We compute the empirical canonical correlations $\rho_1, \ldots, \rho_5$ and the corresponding subspaces $\mathcal{C}_1, \ldots, \mathcal{C}_5$, and then set regularizer weights $\lambda_i = (2 - \rho_i^2 - \rho_{i+1}^2)/2$ for $i = 1, \ldots, 4$. For each λ_i, we optimize the finite-sample version of (5.2) using gradient descent on the parameters $P, Q \in \mathbb{R}^{5\times 5}$ with a learning rate of 0.1 for 20000 steps. We measure the error in P given by $\|P\Sigma_u^{1/2}\Pi_{\mathcal{C}_i^\perp}\|_F$ and the threshold rank of P (the number of singular values of P greater than 10^{-5}). In every case, the error in P converges to 0 and the threshold rank converges to i, thus confirming the conclusion of Theorem 5. In Figure 5.2, we plot of these quantities for $i = 3$ as a function of the gradient step.

Forward Model We create synthetic data from the hidden subspace model by first drawing random matrices $\bar{A} \in \mathbb{R}^{3\times 3}$ and $\bar{B} \in \mathbb{R}^{3\times 2}$ with i.i.d. standard Gaussian entries, and then generating 1000 i.i.d. samples from the model $h_0 \in \mathbb{R}^3 \sim N(0, I)$,

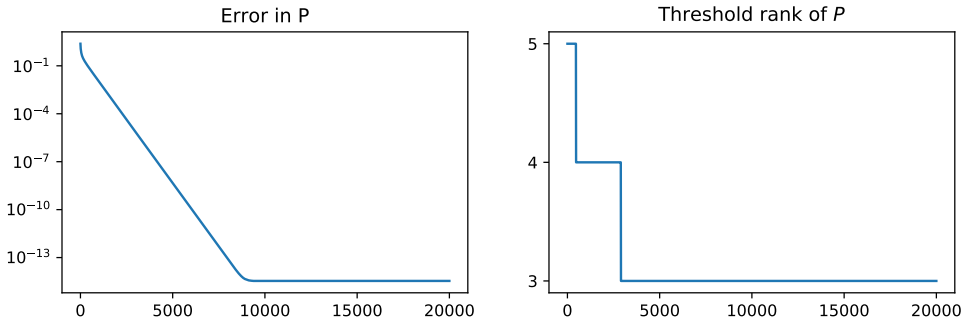


Figure 5.2: Error and threshold rank of P during training

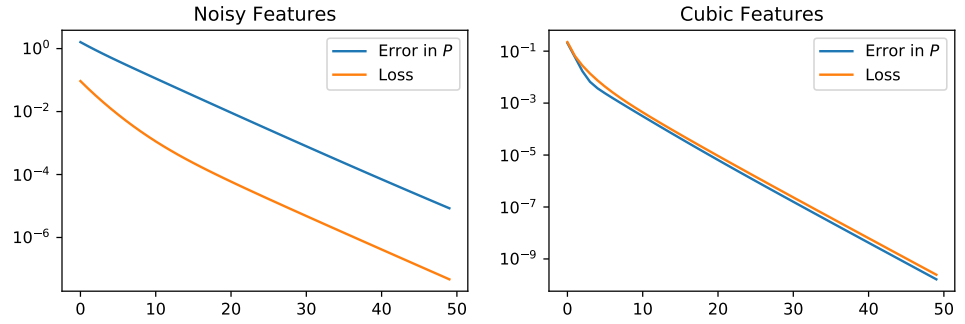


Figure 5.3: Error in P and loss for the forward model objective

$u_0 \in \mathbb{R}^2 \sim N(0, I)$, $h_1 = \bar{A}h_0 + \bar{B}u_0$. We consider two methods for generating nonlinear features: first by adding white noise where $z_i = h_i + \epsilon_i$, with $\epsilon_i \sim N(0, \sigma^2 I)$ and $\sigma = .1$, and second by taking cubic features $z_i = h_i^3$ (where the cubic power is performed entry-wise, and then truncation is applied to prevent numerical blowup). The observations are then constructed by concatenation: $x_i = (h_i, z_i)$. For both the noisy and cubic features, we optimize (5.4) using gradient descent with $\lambda = .75, \lambda = .5$, learning rates $.001, .0005$, and number of steps $2.5 \times 10^6, 5 \times 10^5$, respectively. In both cases, the correct subspace is successfully identified. In Figure 5.3 we plot the loss function and error in P for both experiments. We also conduct similar experiments for the CCA objective – details are in the appendix.

Inverse Model We generate data according to our hidden subspace model as follows. Set the system matrices $\bar{A}, \bar{B}$ at random (with i.i.d. Guassian entries) and multiply $\bar{A}$ by a constant to ensure it is well-conditioned (to avoid numerical issues). Sample the initial latent states $h_0 \sim N(0, I)$ and actions $u_i \sim N(0, I)$. The nonlinear components z_i are created either as independent Gaussian noise or low-degree polynomials of h_i. We collect $5(d + rl)$ samples for each run (this is lower than the sample complexity we give in the Theorem 8, but it sufficed for our experiments).

To construct the particular minimal-norm solutions in Theorem 8, we optimize (5.8) in a two-stage linear least squares process using a standard least-square solver ("lstsq" function in SciPy), as explained below. We then check that our constructed solution matches the solution guaranteed by Theorem 8. In all of our runs, whenever the computations were numerically stable, we indeed recover the expected solution.

To explain this process in detail, it simplifies things to consider the least squares problem

$$\min_{x,y} \|Ax + By - c\|_2^2,$$

where A and B are arbitrary matrices and c is an arbitrary vector. Assume the space of solutions $\{x, y\}$ that have zero error is nonempty (i.e. it is an entire linear space of solutions). We want to select the optimal solution (x^*, y^*) such that for any other optimal solution (x', y'), we have $\|x^*\|_2 \leq \|x'\|_2$ and if $\|x^*\|_2 = \|x'\|_2$ then $\|y^*\|_2 \leq \|y'\|_2$.

We can obtain such a solution by splitting the problem into two stages. First, let x^* be the minimal norm solution of

$$\min_{x} \|(I - P_B)Ax - (I - P_B)c\|_2^2,$$

where P_B is the orthogonal projection onto the column-span of B. We can compute x^*

using standard least squares techniques such as using the singular value decomposition. Then, let y^* be the minimal norm solution to

$$\min_y \|By - P_B c + P_B A x^*\|_2^2.$$

Let us verify that (x^*, y^*) has the desired properties. Let (x', y') be any solution, i.e. $Ax' + By' = c$. Left-multiplying the equation by $I - P_B$, we see that $(I - P_B)Ax' = (I - P_B)c$. By construction, we have that $\|x^*\|_2 \leq \|x'\|_2$. Now assume that $\|x^*\|_2 = \|x'\|_2$. This implies that $x^* = x'$ (the minimum-norm solution is unique). Then we have $By' = P_B c - P_B Ax' = P_B c - P_B Ax^*$. Again, by construction we have that $\|y^*\|_2 \leq \|y'\|_2$, as desired.

5.6.2 Nonlinear RL Environments

While our theory doesn't provide guarantees for the setting in which the learned nonlinear state representations incur nonzero loss, we can empirically investigate whether optimizing (5.13) and (5.14) lead to reasonable representations. We examine the learned representations visually and explore whether they admit effective control policies. We focus on two standard continuous control tasks from OpenAI Gym (Brockman u. a., 2016): 'Pendulum-v0' and 'MountainCarContinuous-v0'.

We implement our learning algorithm in PyTorch (Paszke u. a., 2017), and our policy search algorithms use the Stable Baselines library (Hill u. a., 2018). We follow the basic approach taken by Lillicrap u. a. (2015) in working with pixel observations: modify the environments so that each action is repeated over three consecutive timesteps in the original environment, and concatenate the resultant observations. More specifically, for each environment, we re-implement the scene renderings to reduce render time and make it compatible with our computing environment. We also

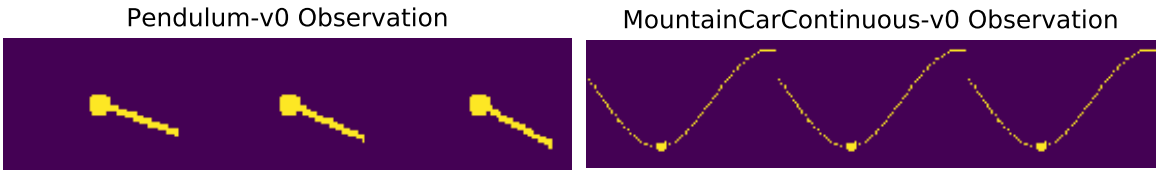


Figure 5.4: Pixel observations for the environments tested.

add to our environments the ability to reset the system in a desired state (rather than a random state). This allows us to sample initial states from a desired distribution on the state space. After repeating the action three times, the resulting concatenated pixel observations have sizes $(64, 192)$ and $(80, 360)$ for the pendulum and mountain car environments, respectively. Figure 5.4 displays examples of these state observations.

The state representation map ψ is a basic neural network with two convolutional layers (each with 16 output channels, the first layer with kernel size 8 and stride 4, the second layer with kernel size 4 and stride 2) followed by two fully connected layers each of width 50 for the inverse model, and two fully connected layers of width 1000 and 8 for the forward model. All layers use ReLu activation with no other nonlinearities. For the inverse model, after the final layer, we project to the top 4 right singular directions of the matrix $[P^\top \; L_1^\top \; \cdots \; L_\tau^\top]^\top$, so that in the end, we have a 4-dimensional representation.

To train the representations, we solve (5.13) and (5.14) using the Adam optimizer using minibatches of data. We observed that the loss function converged to a nonzero value, which means there may be room to better learn the forward and inverse models if we explore different architecture or training options.

Visual Analysis The pendulum environment has two underlying state variables: the angle of the pendulum from vertically upwards, and its angular velocity. Hence, the slice of the state space corresponding to 0 angular velocity can be viewed as a

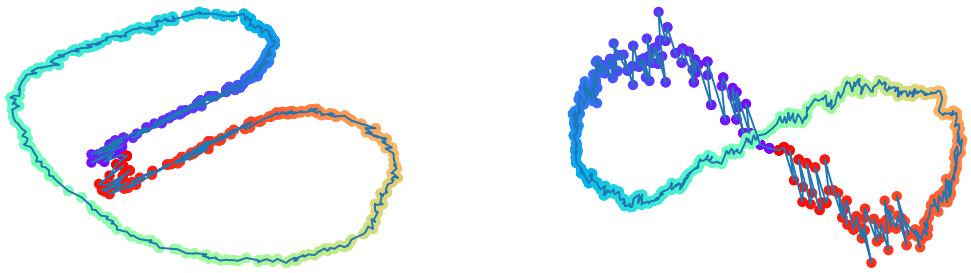


Figure 5.5: Visualizations of learned pendulum state representations for the forward model (left) and inverse model (right)

circle, with the pendulum angle ranging cyclically from 0 to 2π. Given a trained representation, we evenly sample this slice of the state space and compute the state representation at each of these points. We then project onto the top two principal components and plot the result in Figure 5.5, where the color-coding indicates the angle in radians of the pendulum from vertical (red and violet correspond to fully vertical). Both representations capture in distinct ways the symmetry of the state space when reflecting the angle about 0.

Policy Learning After training the state representations, we next explore whether they admit effective policies. The forward and inverse model representations were trained on a fixed set of batches of trajectories. The inverse model naturally trains over longer trajectories. With the forward model we found it necessary to not just collect many length-1 trajectories, but instead collect longer trajectories and then extract the 1-step trajectories embedded within. For the inverse model, we restrict to linear policies, as these are simpler to optimize and work well for our representation. For the forward model, nonlinear neural network policies are trained, as the linear policies weren't as effective.

We use the Stable Baselines implementation of TRPO to train the policies. For the

linear policies, we use all of the default parameters except for the stepsize parameter "vf_stepsize", which we tested over a range of values in $[0.00005, 0.5]$. For the nonlinear policies, we tested over a larger set of hyperparameters and again reported the best results.

We also include two baselines that rely on standard RL algorithms implemented in Stable Baselines. Baseline 1 trains nonlinear policies directly from the raw pixel observations (we tune the learning rates and report the best results), and baseline 2 trains policies directly on the low-dimensional state variables using tuned hyperparameters provided by RL Baselines Zoo (Raffin, 2018). The learning curves for all approaches are shown in Figure 5.6.

Note that the forward model representation is *far less sample efficient* than the other methods (its x-axis is scaled up by an order of magnitude compared to the others). The discrepancy between the forward and inverse model representations in this respect may be due in part to the fact that a neural network policy has many more parameters than a linear policy, and hence will naturally train slower. It may also point to an intrinsic difference in the quality of representations produced by the forward model and the inverse model. Perhaps the inverse model representation benefits from being trained over longer trajectories, whereas the forward model just uses a single time step.

For both the pendulum and mountain car environments, each representation admits performant policies that solve the tasks, albeit with less total reward than the tuned Baseline 2 policies. These results show that our state representation learning algorithms capture the relevant structure and dynamics necessary to enable reasonable control policies.

5.7 Conclusion and Future Work

State representation learning is a promising way to bridge nonlinear, high-dimensional reinforcement learning problems and the simple linear models that have theoretical guarantees. In this paper we study a basic model for state representation learning and show that effective, low-dimensional state representations can be learned efficiently from rich observations using either forward or inverse models. The algorithm inspired by our theory can indeed recover reasonable state representations for simple tasks in OpenAI gym.

There are still many open problems: the nonconvex objectives (5.13) and (5.14) can be hard to optimize for the network architectures we tried; is there a way to design the architecture to make the loss go to 0? Additionally, the our algorithms rely on the initial state distribution, which may not sufficiently cover all parts of the state space; can we complement our algorithm with an exploration strategy? Are there more realistic models for state representation learning that can also be learned efficiently? We hope our paper serves as a starting point towards these questions.

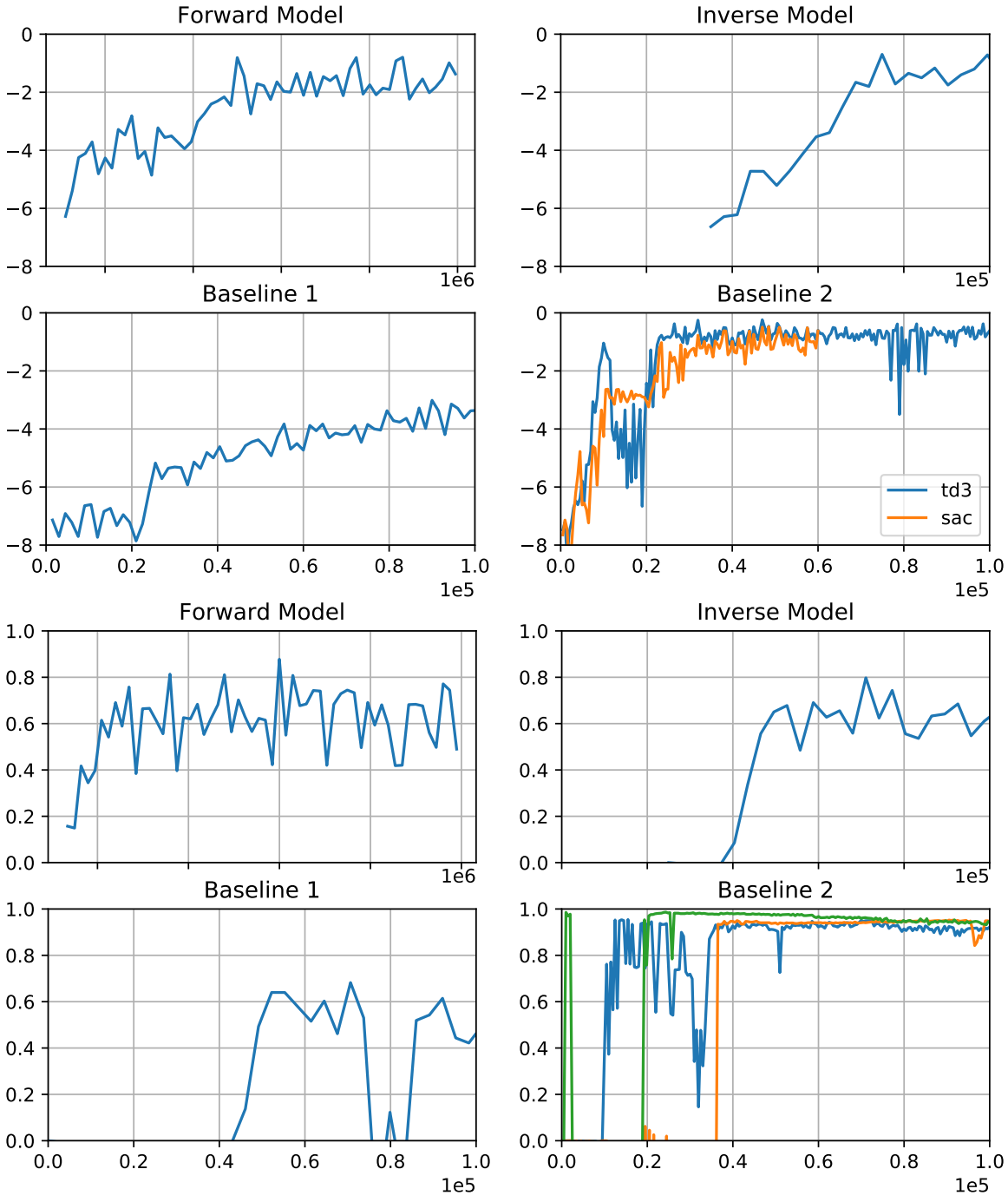


Figure 5.6: Learning curves for 'Pendulum-v0' (top four) and 'MountainCarContinuous-v0' (bottom four).

Chapter 6

Conclusion

In this thesis, we studied representation learning in the context of latent variable models with latent linear-algebraic structure. In the area of natural language processing, we proposed a model for word embeddings that builds on a previous latent variable model but introduces additional structure to model syntactic relationships. We demonstrated that the latent multilinear structure in our model can be efficiently learned and in practice produces word embeddings that better represent the rich semantic and syntactic structure of human language. In the area of reinforcement learning and control, we introduced a simple but general model for state representation learning in which the true latent dynamics are linear, and we developed algorithms based on forward and inverse modeling that provably recover the ground-truth representations. Our work provides a principled foundation for similar state representation learning algorithms used in practice. In both of these applications, we connected the learning problem with fundamental linear-algebraic problems, namely Tucker decomposition and canonical correlation analysis. We gave nonconvex optimization formulations for these problems and proved that they can be efficiently and globally optimized, building on results and techniques developed over a series of works on nonconvex optimization.

One natural continuation of the work in this thesis is to apply our approach of latent variable modeling for representation learning to additional areas and problems of importance in current machine learning practice. In natural language processing, for example, transformer-based models have become overwhelmingly popular. Can we propose a reasonable latent variable model that captures the kinds of structure

that transformers are designed for, and can we thereby rigorously analyze these algorithmic techniques that have been so successful in practice? In the area of reinforcement learning and control, can we extend our model to incorporate reward signals, constraints, or more complex dynamics such as switching linear systems? Other areas such as computer vision, computational biology, or graph-structured data also provide fertile ground for studying representation learning. By proposing and studying latent variable models that capture the core elements of current machine learning algorithms, we can build understanding and insight into why these algorithms work in practice and how they might be improved. It is a key step toward sounder theoretical grounding of machine learning.

Of course, the approach we take in this thesis has important limitations. For one, latent variable models rarely reflect reality completely. Real-word data are usually noiser and more complex than our models allow. We can adapt our approach to this setting by allowing for some model misspecification. If we can make reasonable assumptions about how the real-world data deviate from the latent variable model, it may be possible to adapt the representation learning algorithm such that the learned representations still have provably beneficial properties. Proving any guarantees about representation learning algorithms in this regime is more challenging, but ultimately the goal is to obtain a rigorous understanding of the real-world behavior of machine learning.

Another limitation of our specific approach is the focus on modeling with linear-algebraic structure. This choice has many benefits, but it isn't reasonable to expect that good representations should *always* exhibit linear structure – some problems may simply be better described through nonlinear structure. Although nonlinearity is present in all of our models, is it possible and beneficial to consider latent variable models where the latent representations capture nonlinear structure? Can we still

design provably correct algorithms in this case? An interesting theoretical question is whether there are certain natural problems in representation learning where allowing for nonlinear structure is strictly better than representations capturing only linear-algebraic structure.

Bibliography

[Abadi u. a. 2016] Abadi, Martín ; Barham, Paul ; Chen, Jianmin ; Chen, Zhifeng ; Davis, Andy ; Dean, Jeffrey ; Devin, Matthieu ; Ghemawat, Sanjay ; Irving, Geoffrey ; Isard, Michael u. a.: TensorFlow: A System for Large-Scale Machine Learning. In: *OSDI* Bd. 16, 2016, S. 265–283

[Anand u. a. 2019] Anand, Ankesh ; Racah, Evan ; Ozair, Sherjil ; Bengio, Yoshua ; Côté, Marc-Alexandre ; Hjelm, R D.: Unsupervised state representation learning in atari. In: *Advances in Neural Information Processing Systems*, 2019, S. 8766–8779

[Anandkumar u. a. 2012a] Anandkumar, Anima ; Foster, Dean P. ; Hsu, Daniel J. ; Kakade, Sham M. ; Liu, Yi-Kai: A spectral algorithm for latent dirichlet allocation. In: *Advances in neural information processing systems* 25 (2012)

[Anandkumar und Ge 2016] Anandkumar, Animashree ; Ge, Rong: Efficient approaches for escaping higher order saddle points in non-convex optimization. In: *Conference on learning theory*, 2016, S. 81–102

[Anandkumar u. a. 2013] Anandkumar, Animashree ; Ge, Rong ; Hsu, Daniel ; Kakade, Sham: A tensor spectral approach to learning mixed membership community models. In: *Conference on Learning Theory* PMLR (Veranst.), 2013, S. 867–881

[Anandkumar u. a. 2014] Anandkumar, Animashree ; Ge, Rong ; Hsu, Daniel ; Kakade, Sham M. ; Telgarsky, Matus: Tensor decompositions for learning latent variable models. In: *Journal of machine learning research* 15 (2014), S. 2773–2832

[Anandkumar u. a. 2012b] Anandkumar, Animashree ; Hsu, Daniel ; Kakade, Sham M.: A method of moments for mixture models and hidden Markov models. In: *Conference on Learning Theory* JMLR Workshop and Conference Proceedings (Veranst.), 2012, S. 33–1

[Andreas und Ghahramani 2013] Andreas, Jacob ; Ghahramani, Zoubin: A generative model of vector space semantics. In: *Proceedings of the Workshop on Continuous Vector Space Models and their Compositionality*, 2013, S. 91–99

[Andreas und Klein 2014] Andreas, Jacob ; Klein, Dan: How much do word embeddings encode about syntax? In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)* Bd. 2, 2014, S. 822–827

[Arora u. a. 2018] Arora, Sanjeev ; Hazan, Elad ; Lee, Holden ; Singh, Karan ; Zhang, Cyril ; Zhang, Yi: Towards provable control for unknown linear dynamical systems. (2018)

[Arora u. a. 2015] Arora, Sanjeev ; Li, Yuanzhi ; Liang, Yingyu ; Ma, Tengyu ; Risteski, Andrej: Rand-walk: A latent variable model approach to word embeddings. In: *arXiv preprint arXiv:1502.03520* (2015)

[Arora u. a. 2016] Arora, Sanjeev ; Liang, Yingyu ; Ma, Tengyu: A simple but tough-to-beat baseline for sentence embeddings. (2016)

[Arora und Risteski 2017] Arora, Sanjeev ; Risteski, Andrej: Provable benefits of representation learning. In: *arXiv preprint arXiv:1706.04601* (2017)

[Bailey und Aeron 2017] Bailey, Eric ; Aeron, Shuchin: Word Embeddings via Tensor Factorization. In: *arXiv preprint arXiv:1704.02686* (2017)

[Bandeira u. a. 2016] Bandeira, Afonso S. ; Boumal, Nicolas ; Voroninski, Vladislav: On the low-rank approach for semidefinite programs arising in synchronization and community detection. In: *arXiv preprint arXiv:1602.04426* (2016)

[Baroni und Zamparelli 2010] Baroni, Marco ; Zamparelli, Roberto: Nouns are vectors, adjectives are matrices: Representing adjective-noun constructions in semantic space. In: *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing* Association for Computational Linguistics (Veranst.), 2010, S. 1183–1193

[Bengio u. a. 2013] Bengio, Yoshua ; Courville, Aaron ; Vincent, Pascal: Representation learning: A review and new perspectives. In: *IEEE transactions on pattern analysis and machine intelligence* 35 (2013), Nr. 8, S. 1798–1828

[Bhojanapalli u. a. 2016] Bhojanapalli, Srinadh ; Neyshabur, Behnam ; Srebro, Nati: Global optimality of local search for low rank matrix recovery. In: *Advances in Neural Information Processing Systems*, 2016, S. 3873–3881

[Blei u. a. 2003] Blei, David M. ; Ng, Andrew Y. ; Jordan, Michael I.: Latent dirichlet allocation. In: *Journal of machine Learning research* 3 (2003), Nr. Jan, S. 993–1022

[Boffi u. a. 2020] Boffi, Nicholas M. ; Tu, Stephen ; Slotine, Jean-Jacques E.: Regret Bounds for Adaptive Nonlinear Control. In: *arXiv preprint arXiv:2011.13101* (2020)

[Borga 2001] Borga, Magnus: Canonical correlation: a tutorial. In: *On line tutorial http://people. imt. liu. se/magnus/cca* 4 (2001), Nr. 5

[Brockman u. a. 2016] Brockman, Greg ; Cheung, Vicki ; Pettersson, Ludwig ; Schneider, Jonas ; Schulman, John ; Tang, Jie ; Zaremba, Wojciech: Openai gym. In: *arXiv preprint arXiv:1606.01540* (2016)

[Bullins u. a. 2019] Bullins, Brian ; Hazan, Elad ; Kalai, Adam ; Livni, Roi: Generalize across tasks: Efficient algorithms for linear representation learning. In: *Algorithmic Learning Theory* PMLR (Veranst.), 2019, S. 235–246

[Carbery und Wright 2001] Carbery, Anthony ; Wright, James: Distributional and L^ q norm inequalities for polynomials over convex bodies in R^ n. In: *Mathematical research letters* 8 (2001), Nr. 3, S. 233–248

[Carroll und Chang 1970] Carroll, J D. ; Chang, Jih-Jie: Analysis of individual differences in multidimensional scaling via an N-way generalization of "Eckart-Young" decomposition. In: *Psychometrika* 35 (1970), Nr. 3, S. 283–319

[Chen und Manning 2014] Chen, Danqi ; Manning, Christopher: A fast and accurate dependency parser using neural networks. In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, S. 740–750

[Chen und Goodman 1996] Chen, Stanley F. ; Goodman, Joshua: An empirical study of smoothing techniques for language modeling. In: *Proceedings of the 34th annual meeting on Association for Computational Linguistics* Association for Computational Linguistics (Veranst.), 1996, S. 310–318

[Cheng und Kartsaklis 2015] Cheng, Jianpeng ; Kartsaklis, Dimitri: Syntax-aware multi-sense word embeddings for deep compositional models of meaning. In: *arXiv preprint arXiv:1508.02354* (2015)

[Chi u. a. 2019] Chi, Yuejie ; Lu, Yue M. ; Chen, Yuxin: Nonconvex optimization meets low-rank matrix factorization: An overview. In: *IEEE Transactions on Signal Processing* 67 (2019), Nr. 20, S. 5239–5269

[Coecke u. a. 2010] Coecke, Bob ; Sadrzadeh, Mehrnoosh ; Clark, Stephen: Mathematical foundations for a compositional distributional model of meaning. In: *arXiv preprint arXiv:1003.4394* (2010)

[Cohen u. a. 2018] Cohen, Alon ; Hassidim, Avinatan ; Koren, Tomer ; Lazic, Nevena ; Mansour, Yishay ; Talwar, Kunal: Online linear quadratic control. In: *arXiv preprint arXiv:1806.07104* (2018)

[Van de Cruys 2011] Cruys, Tim Van de: Two multivariate generalizations of pointwise mutual information. In: *Proceedings of the Workshop on Distributional Semantics and Compositionality* Association for Computational Linguistics (Veranst.), 2011, S. 16–20

[Davenport und Romberg 2016] Davenport, Mark A. ; Romberg, Justin: An overview of low-rank matrix recovery from incomplete observations. In: *IEEE Journal of Selected Topics in Signal Processing* 10 (2016), Nr. 4, S. 608–622

[De Lathauwer u. a. 2000a] De Lathauwer, Lieven ; De Moor, Bart ; Vandewalle, Joos: A multilinear singular value decomposition. In: *SIAM journal on Matrix Analysis and Applications* 21 (2000), Nr. 4, S. 1253–1278

[De Lathauwer u. a. 2000b] De Lathauwer, Lieven ; De Moor, Bart ; Vandewalle, Joos: On the best rank-1 and rank-(r 1, r 2,..., rn) approximation of higher-order tensors. In: *SIAM journal on Matrix Analysis and Applications* 21 (2000), Nr. 4, S. 1324–1342

[Dean u. a. 2020] Dean, Sarah ; Matni, Nikolai ; Recht, Benjamin ; Ye, Vickie: Robust guarantees for perception-based control. In: *Learning for Dynamics and Control* PMLR (Veranst.), 2020, S. 350–360

[Dhillon u. a. 2015] Dhillon, Paramveer S. ; Foster, Dean P. ; Ungar, Lyle H.: Eigenwords: Spectral word embeddings. In: *Journal of Machine Learning Research* 16 (2015), S. 3035–3078

[Ding u. a. 2006] Ding, Chris ; Li, Tao ; Peng, Wei: Nonnegative matrix factorization and probabilistic latent semantic indexing: Equivalence chi-square statistic, and a hybrid method. In: *AAAI* Bd. 42, 2006, S. 137–43

[Donahue und Simonyan 2019] Donahue, Jeff ; Simonyan, Karen: Large scale adversarial representation learning. In: *Advances in Neural Information Processing Systems* 32 (2019)

[Du u. a. 2019a] Du, Simon S. ; Kakade, Sham M. ; Wang, Ruosong ; Yang, Lin F.: Is a Good Representation Sufficient for Sample Efficient Reinforcement Learning? In: *arXiv preprint arXiv:1910.03016* (2019)

[Du u. a. 2019b] Du, Simon S. ; Krishnamurthy, Akshay ; Jiang, Nan ; Agarwal, Alekh ; Dudík, Miroslav ; Langford, John: Provably efficient rl with rich observations via latent state decoding. In: *arXiv preprint arXiv:1901.09018* (2019)

[Eldén und Savas 2009] Eldén, Lars ; Savas, Berkant: A Newton–Grassmann Method for Computing the Best Multilinear Rank-(r_1, r_2, r_3) Approximation of a Tensor. In: *SIAM Journal on Matrix Analysis and applications* 31 (2009), Nr. 2, S. 248–271

[Ericsson u. a. 2021] Ericsson, Linus ; Gouk, Henry ; Loy, Chen C. ; Hospedales, Timothy M.: Self-Supervised Representation Learning: Introduction, Advances and Challenges. In: *arXiv preprint arXiv:2110.09327* (2021)

[Fazel u. a. 2018] Fazel, Maryam ; Ge, Rong ; Kakade, Sham M. ; Mesbahi, Mehran: Global convergence of policy gradient methods for the linear quadratic regulator. In: *arXiv preprint arXiv:1801.05039* (2018)

[Finn u. a. 2016] Finn, Chelsea ; Tan, Xin Y. ; Duan, Yan ; Darrell, Trevor ; Levine, Sergey ; Abbeel, Pieter: Deep spatial autoencoders for visuomotor learning. In: *2016 IEEE International Conference on Robotics and Automation (ICRA)* IEEE (Veranst.), 2016, S. 512–519

[Folkestad u. a. 2019] Folkestad, Carl ; Pastor, Daniel ; Mezic, Igor ; Mohr, Ryan ; Fonoberova, Maria ; Burdick, Joel: Extended dynamic mode decomposition with learned koopman eigenfunctions for prediction and control. In: *arXiv preprint arXiv:1911.08751* (2019)

[Frandsen und Ge 2019] Frandsen, Abraham ; Ge, Rong: Understanding Composition of Word Embeddings via Tensor Decomposition. In: *International Conference on Learning Representations*, 2019

[Frandsen und Ge 2020] Frandsen, Abraham ; Ge, Rong: Optimization landscape of Tucker decomposition. In: *Mathematical Programming* (2020)

[Gao u. a. 2019] Gao, Chao ; Garber, Dan ; Srebro, Nathan ; Wang, Jialei ; Wang, Weiran: Stochastic canonical correlation analysis. In: *Journal of Machine Learning Research* 20 (2019), Nr. 167, S. 1–46

[Ge 2013] Ge, Rong: *Provable algorithms for machine learning problems*, Princeton University, Dissertation, 2013

[Ge u. a. 2015] Ge, Rong ; Huang, Furong ; Jin, Chi ; Yuan, Yang: Escaping from saddle points—online stochastic gradient for tensor decomposition. In: *Conference on Learning Theory*, 2015, S. 797–842

[Ge u. a. 2017] Ge, Rong ; Jin, Chi ; Zheng, Yi: No spurious local minima in nonconvex low rank problems: A unified geometric analysis. In: *International Conference on Machine Learning*, 2017

[Ge u. a. 2016] Ge, Rong ; Lee, Jason D. ; Ma, Tengyu: Matrix completion has no spurious local minimum. In: *Advances in Neural Information Processing Systems*, 2016, S. 2973–2981

[Ge u. a. 2018] Ge, Rong ; Lee, Jason D. ; Ma, Tengyu: Learning one-hidden-layer neural networks with landscape design. In: *International Conference on Learning Representations*, 2018

[Gittens u. a. 2017] Gittens, Alex ; Achlioptas, Dimitris ; Mahoney, Michael W.: Skip-gram-zipf+ uniform= vector additivity. In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* Bd. 1, 2017, S. 69–76

[Guevara 2010] Guevara, Emiliano: A regression model of adjective-noun compositionality in distributional semantics. In: *Proceedings of the 2010 Workshop on GEometrical Models of Natural Language Semantics* Association for Computational Linguistics (Veranst.), 2010, S. 33–37

[Ha und Schmidhuber 2018] Ha, David ; Schmidhuber, Jürgen: World models. In: *arXiv preprint arXiv:1803.10122* (2018)

[Hafner u. a. 2019a] Hafner, Danijar ; Lillicrap, Timothy ; Ba, Jimmy ; Norouzi, Mohammad: Dream to control: Learning behaviors by latent imagination. In: *arXiv preprint arXiv:1912.01603* (2019)

[Hafner u. a. 2019b] Hafner, Danijar ; Lillicrap, Timothy ; Fischer, Ian ; Villegas, Ruben ; Ha, David ; Lee, Honglak ; Davidson, James: Learning latent dynamics for planning from pixels. In: *International Conference on Machine Learning* PMLR (Veranst.), 2019, S. 2555–2565

[Hardt u. a. 2018] Hardt, Moritz ; Ma, Tengyu ; Recht, Benjamin: Gradient descent learns linear dynamical systems. In: *The Journal of Machine Learning Research* 19 (2018), Nr. 1, S. 1025–1068

[Harshman u. a. 1970] Harshman, Richard A. u. a.: Foundations of the PARAFAC procedure: Models and conditions for an" explanatory" multimodal factor analysis. (1970)

[Hartung u. a. 2017] Hartung, Matthias ; Kaupmann, Fabian ; Jebbara, Soufian ; Cimiano, Philipp: Learning Compositionality Functions on Word Embeddings for Modelling Attribute Meaning in Adjective-Noun Phrases. In: *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers* Bd. 1, 2017, S. 54–64

[Hashimoto u. a. 2016] Hashimoto, Tatsunori B. ; Alvarez-Melis, David ; Jaakkola, Tommi S.: Word embeddings as metric recovery in semantic spaces. In: *Transactions of the Association for Computational Linguistics* 4 (2016), S. 273–286

[Håstad 1990] Håstad, Johan: Tensor rank is NP-complete. In: *Journal of Algorithms* 11 (1990), Nr. 4, S. 644–654

[Heideman u. a. 1984] Heideman, Michael ; Johnson, Don ; Burrus, Charles: Gauss and the history of the fast Fourier transform. In: *IEEE ASSP Magazine* 1 (1984), Nr. 4, S. 14–21

[Higgins u. a. 2017] Higgins, Irina ; Pal, Arka ; Rusu, Andrei ; Matthey, Loic ; Burgess, Christopher ; Pritzel, Alexander ; Botvinick, Matthew ; Blundell, Charles ; Lerchner, Alexander: Darla: Improving zero-shot transfer in reinforcement learning. In: *Proceedings of the 34th International Conference on Machine Learning-Volume 70* JMLR. org (Veranst.), 2017, S. 1480–1490

[Hill u. a. 2018] Hill, Ashley ; Raffin, Antonin ; Ernestus, Maximilian ; Gleave, Adam ; Kanervisto, Anssi ; Traore, Rene ; Dhariwal, Prafulla ; Hesse, Christopher ; Klimov, Oleg ; Nichol, Alex ; Plappert, Matthias ; Radford, Alec ; Schulman, John ; Sidor, Szymon ; Wu, Yuhuai: *Stable Baselines*. `https://github.com/hill-a/stable-baselines`. 2018

[Hillar und Lim 2013] Hillar, Christopher J. ; Lim, Lek-Heng: Most tensor problems are NP-hard. In: *Journal of the ACM (JACM)* 60 (2013), Nr. 6, S. 45

[Hitchcock 1927] Hitchcock, Frank L.: The expression of a tensor or a polyadic as a sum of products. In: *Journal of Mathematics and Physics* 6 (1927), Nr. 1-4, S. 164–189

[Hofmann 2013] Hofmann, Thomas: Probabilistic latent semantic analysis. In: *arXiv preprint arXiv:1301.6705* (2013)

[Hsu und Kakade 2013] Hsu, Daniel ; Kakade, Sham M.: Learning mixtures of spherical gaussians: moment methods and spectral decompositions. In: *Proceedings of the 4th conference on Innovations in Theoretical Computer Science*, 2013, S. 11–20

[Hsu u. a. 2012] Hsu, Daniel ; Kakade, Sham M. ; Zhang, Tong: A spectral algorithm for learning hidden Markov models. In: *Journal of Computer and System Sciences* 78 (2012), Nr. 5, S. 1460–1480

[Hu u. a. 2021] Hu, Jiachen ; Chen, Xiaoyu ; Jin, Chi ; Li, Lihong ; Wang, Liwei: Near-optimal representation learning for linear bandits and linear rl. In: *International Conference on Machine Learning* PMLR (Veranst.), 2021, S. 4349–4358

[Jin u. a. 2017] Jin, Chi ; Ge, Rong ; Netrapalli, Praneeth ; Kakade, Sham M. ; Jordan, Michael I.: How to escape saddle points efficiently. In: *International Conference on Machine Learning* PMLR (Veranst.), 2017, S. 1724–1732

[Jumper u. a. 2021] Jumper, John ; Evans, Richard ; Pritzel, Alexander ; Green, Tim ; Figurnov, Michael ; Ronneberger, Olaf ; Tunyasuvunakool, Kathryn ; Bates, Russ ; Žídek, Augustin ; Potapenko, Anna u. a.: Highly accurate protein structure prediction with AlphaFold. In: *Nature* 596 (2021), Nr. 7873, S. 583–589

[Kakade u. a. 2020] Kakade, Sham ; Krishnamurthy, Akshay ; Lowrey, Kendall ; Ohnishi, Motoya ; Sun, Wen: Information theoretic regret bounds for online nonlinear control. In: *arXiv preprint arXiv:2006.12466* (2020)

[Kawahara 2016] Kawahara, Yoshinobu: Dynamic mode decomposition with reproducing kernels for Koopman spectral analysis. In: *Advances in neural information processing systems*, 2016, S. 911–919

[Kim und Stern 2016] Kim, Chanwoo ; Stern, Richard M.: Power-normalized cepstral coefficients (PNCC) for robust speech recognition. In: *IEEE/ACM Transactions on audio, speech, and language processing* 24 (2016), Nr. 7, S. 1315–1329

[Kingma und Ba 2014] Kingma, Diederik P. ; Ba, Jimmy: Adam: A method for stochastic optimization. In: *arXiv preprint arXiv:1412.6980* (2014)

[Kintsch 2001] Kintsch, Walter: Predication. In: *Cognitive science* 25 (2001), Nr. 2, S. 173–202

[Kiros u. a. 2014] Kiros, Ryan ; Zemel, Richard ; Salakhutdinov, Ruslan R.: A multiplicative model for learning distributed text-based attribute representations. In: *Advances in neural information processing systems*, 2014, S. 2348–2356

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

[Koren 2009] Koren, Yehuda: The bellkor solution to the netflix grand prize. In: *Netflix prize documentation* 81 (2009), Nr. 2009, S. 1–10

[Kusner u. a. 2015] Kusner, Matt ; Sun, Yu ; Kolkin, Nicholas ; Weinberger, Kilian: From word embeddings to document distances. In: *International Conference on Machine Learning*, 2015, S. 957–966

[Landauer u. a. 1998] Landauer, Thomas K. ; Foltz, Peter W. ; Laham, Darrell: An introduction to latent semantic analysis. In: *Discourse processes* 25 (1998), Nr. 2-3, S. 259–284

[Laurent und Massart 2000] Laurent, Beatrice ; Massart, Pascal: Adaptive estimation of a quadratic functional by model selection. In: *Annals of Statistics* (2000), S. 1302–1338

[Le-Khac u. a. 2020] Le-Khac, Phuc H. ; Healy, Graham ; Smeaton, Alan F.: Contrastive representation learning: A framework and review. In: *IEEE Access* 8 (2020), S. 193907–193934

[Lee und Seung 2000] Lee, Daniel ; Seung, H S.: Algorithms for non-negative matrix factorization. In: *Advances in neural information processing systems* 13 (2000)

[Lesort u. a. 2018] Lesort, Timothée ; Díaz-Rodríguez, Natalia ; Goudou, Jean-Franois ; Filliat, David: State representation learning for control: An overview. In: *Neural Networks* 108 (2018), S. 379–392

[Levy und Goldberg 2014a] Levy, Omer ; Goldberg, Yoav: Dependency-based word embeddings. In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)* Bd. 2, 2014, S. 302–308

[Levy und Goldberg 2014b] Levy, Omer ; Goldberg, Yoav: Neural word embedding as implicit matrix factorization. In: *Advances in neural information processing systems*, 2014, S. 2177–2185

[Li u. a. 2017] Li, Qianxiao ; Dietrich, Felix ; Bollt, Erik M. ; Kevrekidis, Ioannis G.: Extended dynamic mode decomposition with dictionary learning: A data-driven adaptive spectral decomposition of the Koopman operator. In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 27 (2017), Nr. 10, S. 103111

[Li u. a. 2015] Li, Yitan ; Xu, Linli ; Tian, Fei ; Jiang, Liang ; Zhong, Xiaowei ; Chen, Enhong: Word Embedding Revisited: A New Representation Learning and Explicit Matrix Factorization Perspective. In: *IJCAI*, 2015, S. 3650–3656

[Lillicrap u. a. 2015] Lillicrap, Timothy P. ; Hunt, Jonathan J. ; Pritzel, Alexander ; Heess, Nicolas ; Erez, Tom ; Tassa, Yuval ; Silver, David ; Wierstra, Daan: Continuous control with deep reinforcement learning. In: *arXiv preprint arXiv:1509.02971* (2015)

[Lim 2005] Lim, Lek-Heng: Singular values and eigenvalues of tensors: a variational approach. In: *1st IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing, 2005.* IEEE (Veranst.), 2005, S. 129–132

[Logan u. a. 2000] Logan, Beth u. a.: Mel Frequency Cepstral Coefficients for Music Modeling. In: *ISMIR*, 2000

[Lowe 2004] Lowe, David G.: Distinctive image features from scale-invariant keypoints. In: *International journal of computer vision* 60 (2004), Nr. 2, S. 91–110

[Lusch u. a. 2018] Lusch, Bethany ; Kutz, J N. ; Brunton, Steven L.: Deep learning for universal linear embeddings of nonlinear dynamics. In: *Nature communications* 9 (2018), Nr. 1, S. 1–10

[Maas u. a. 2011] Maas, Andrew L. ; Daly, Raymond E. ; Pham, Peter T. ; Huang, Dan ; Ng, Andrew Y. ; Potts, Christopher: Learning Word Vectors for Sentiment Analysis. In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Portland, Oregon, USA : Association for Computational Linguistics, June 2011, S. 142–150. – URL `http://www.aclweb.org/anthology/P11-1015`

[Maillard und Clark 2015] Maillard, Jean ; Clark, Stephen: Learning adjective meanings with a tensor-based skip-gram model. In: *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*, 2015, S. 327–331

[Mania u. a. 2018] Mania, Horia ; Guy, Aurelia ; Recht, Benjamin: Simple random search provides a competitive approach to reinforcement learning. In: *arXiv preprint arXiv:1803.07055* (2018)

[Mania u. a. 2020] Mania, Horia ; Jordan, Michael I. ; Recht, Benjamin: Active learning for nonlinear system identification with guarantees. In: *arXiv preprint arXiv:2006.10277* (2020)

[Mhammedi u. a. 2020] Mhammedi, Zakaria ; Foster, Dylan J. ; Simchowitz, Max ; Misra, Dipendra ; Sun, Wen ; Krishnamurthy, Akshay ; Rakhlin, Alexander ; Langford, John: Learning the Linear Quadratic Regulator from Nonlinear Observations. In: *arXiv preprint arXiv:2010.03799* (2020)

[Mikolov u. a. 2017] Mikolov, Tomas ; Grave, Edouard ; Bojanowski, Piotr ; Puhrsch, Christian ; Joulin, Armand: Advances in pre-training distributed word representations. In: *arXiv preprint arXiv:1712.09405* (2017)

[Mikolov u. a. 2013] Mikolov, Tomas ; Sutskever, Ilya ; Chen, Kai ; Corrado, Greg S. ; Dean, Jeff: Distributed representations of words and phrases and their compositionality. In: *Advances in neural information processing systems*, 2013, S. 3111–3119

[Misra u. a. 2019] Misra, Dipendra ; Henaff, Mikael ; Krishnamurthy, Akshay ; Langford, John: Kinematic State Abstraction and Provably Efficient Rich-Observation Reinforcement Learning. In: *arXiv preprint arXiv:1911.05815* (2019)

[Mitchell und Lapata 2008] Mitchell, Jeff ; Lapata, Mirella: Vector-based models of semantic composition. In: *proceedings of ACL-08: HLT* (2008), S. 236–244

[Mitchell und Lapata 2010] Mitchell, Jeff ; Lapata, Mirella: Composition in distributional models of semantics. In: *Cognitive science* 34 (2010), Nr. 8, S. 1388–1429

[Nesterov und Polyak 2006] Nesterov, Yurii ; Polyak, Boris T.: Cubic regularization of Newton method and its global performance. In: *Mathematical Programming* 108 (2006), Nr. 1, S. 177–205

[Nguyen u. a. 2019] Nguyen, Luong T. ; Kim, Junhan ; Shim, Byonghyo: Low-rank matrix completion: A contemporary survey. In: *IEEE Access* 7 (2019), S. 94215–94237

[Pang und Lee 2004] Pang, Bo ; Lee, Lillian: A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts. In: *"Proceedings of the ACL"*, 2004

[Park u. a. 2016] Park, Dohyung ; Kyrillidis, Anastasios ; Caramanis, Constantine ; Sanghavi, Sujay: Non-square matrix sensing without spurious local minima via the Burer-Monteiro approach. In: *arXiv preprint arXiv:1609.03240* (2016)

[Paszke u. a. 2017] Paszke, Adam ; Gross, Sam ; Chintala, Soumith ; Chanan, Gregory ; Yang, Edward ; DeVito, Zachary ; Lin, Zeming ; Desmaison, Alban ; Antiga, Luca ; Lerer, Adam: Automatic differentiation in pytorch. (2017)

[Pathak u. a. 2017] Pathak, Deepak ; Agrawal, Pulkit ; Efros, Alexei A. ; Darrell, Trevor: Curiosity-driven exploration by self-supervised prediction. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, S. 16–17

[Pedregosa u. a. 2011] Pedregosa, F. ; Varoquaux, G. ; Gramfort, A. ; Michel, V. ; Thirion, B. ; Grisel, O. ; Blondel, M. ; Prettenhofer, P. ; Weiss, R. ; Dubourg, V. ; Vanderplas, J. ; Passos, A. ; Cournapeau, D. ; Brucher, M. ; Perrot, M. ; Duchesnay, E.: Scikit-learn: Machine Learning in Python. In: *Journal of Machine Learning Research* 12 (2011), S. 2825–2830

[Pennington u. a. 2014] Pennington, Jeffrey ; Socher, Richard ; Manning, Christopher D.: Glove: Global Vectors for Word Representation. In: *EMNLP* Bd. 14, 2014, S. 1532–1543

[Phan u. a. 2014] Phan, Anh-Huy ; Cichocki, Andrzej ; Tichavskỳ, Petr: On fast algorithms for orthogonal Tucker decomposition. In: *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* IEEE (Veranst.), 2014, S. 6766–6770

[Pittner und Kamarthi 1999] Pittner, Stefan ; Kamarthi, Sagar V.: Feature extraction from wavelet coefficients for pattern recognition tasks. In: *IEEE Transactions on pattern analysis and machine intelligence* 21 (1999), Nr. 1, S. 83–88

[Qin 2006] Qin, S J.: An overview of subspace identification. In: *Computers & chemical engineering* 30 (2006), Nr. 10-12, S. 1502–1513

[Raffin 2018] Raffin, Antonin: *RL Baselines Zoo*. `https://github.com/araffin/rl-baselines-zoo`. 2018

[Raffin u. a. 2019] Raffin, Antonin ; Hill, Ashley ; Traoré, Kalifou R. ; Lesort, Timothée ; Díaz-Rodríguez, Natalia ; Filliat, David: Decoupling feature extraction from policy learning: assessing benefits of state representation learning in goal based robotics. In: *arXiv preprint arXiv:1901.08651* (2019)

[Rajeswaran u. a. 2017] Rajeswaran, Aravind ; Lowrey, Kendall ; Todorov, Emanuel V. ; Kakade, Sham M.: Towards generalization and simplicity in continuous control. In: *Advances in Neural Information Processing Systems*, 2017, S. 6550–6561

[Recht und Ré 2013] Recht, Benjamin ; Ré, Christopher: Parallel stochastic gradient algorithms for large-scale matrix completion. In: *Mathematical Programming Computation* 5 (2013), Nr. 2, S. 201–226

[Savas und Eldén 2007] Savas, Berkant ; Eldén, Lars: Handwritten digit classification using higher order singular value decomposition. In: *Pattern recognition* 40 (2007), Nr. 3, S. 993–1003

[Schölkopf u. a. 2021] Schölkopf, Bernhard ; Locatello, Francesco ; Bauer, Stefan ; Ke, Nan R. ; Kalchbrenner, Nal ; Goyal, Anirudh ; Bengio, Yoshua: Toward causal representation learning. In: *Proceedings of the IEEE* 109 (2021), Nr. 5, S. 612–634

[Schulman u. a. 2015] Schulman, John ; Levine, Sergey ; Abbeel, Pieter ; Jordan, Michael ; Moritz, Philipp: Trust region policy optimization. In: *International conference on machine learning*, 2015, S. 1889–1897

[Sharan und Valiant 2017] Sharan, Vatsal ; Valiant, Gregory: Orthogonalized ALS: A Theoretically Principled Tensor Decomposition Algorithm for Practical Use. In: *arXiv preprint arXiv:1703.01804* (2017)

[Shelhamer u. a. 2016] Shelhamer, Evan ; Mahmoudieh, Parsa ; Argus, Max ; Darrell, Trevor: Loss is its own reward: Self-supervision for reinforcement learning. In: *arXiv preprint arXiv:1612.07307* (2016)

[Silver u. a. 2017] Silver, David ; Schrittwieser, Julian ; Simonyan, Karen ; Antonoglou, Ioannis ; Huang, Aja ; Guez, Arthur ; Hubert, Thomas ; Baker, Lucas ; Lai, Matthew ; Bolton, Adrian u. a.: Mastering the game of go without human knowledge. In: *Nature* 550 (2017), Nr. 7676, S. 354–359

[Socher u. a. 2012] Socher, Richard ; Huval, Brody ; Manning, Christopher D. ; Ng, Andrew Y.: Semantic compositionality through recursive matrix-vector spaces. In: *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning* Association for Computational Linguistics (Veranst.), 2012, S. 1201–1211

[Srinivas u. a. 2020] Srinivas, Aravind ; Laskin, Michael ; Abbeel, Pieter: Curl: Contrastive unsupervised representations for reinforcement learning. In: *arXiv preprint arXiv:2004.04136* (2020)

[Stewart 1998] Stewart, Gilbert W.: Perturbation theory for the singular value decomposition. 1998. – Forschungsbericht

[Sun u. a. 2016a] Sun, Ju ; Qu, Qing ; Wright, John: Complete dictionary recovery over the sphere I: Overview and the geometric picture. In: *IEEE Transactions on Information Theory* (2016)

[Sun u. a. 2016b] Sun, Ju ; Qu, Qing ; Wright, John: A geometric analysis of phase retrieval. In: *Information Theory (ISIT), 2016 IEEE International Symposium on* IEEE (Veranst.), 2016, S. 2379–2383

[Tang u. a. 2014] Tang, Duyu ; Wei, Furu ; Yang, Nan ; Zhou, Ming ; Liu, Ting ; Qin, Bing: Learning Sentiment-Specific Word Embedding for Twitter Sentiment Classification. In: *ACL (1)*, 2014, S. 1555–1565

[Thekumparampil u. a. 2021] Thekumparampil, Kiran K. ; Jain, Prateek ; Netrapalli, Praneeth ; Oh, Sewoong: Statistically and Computationally Efficient Linear Meta-representation Learning. In: *Advances in Neural Information Processing Systems* 34 (2021)

[Toutanova u. a. 2003] Toutanova, Kristina ; Klein, Dan ; Manning, Christopher D. ; Singer, Yoram: Feature-rich part-of-speech tagging with a cyclic dependency network. In: *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1* Association for Computational Linguistics (Veranst.), 2003, S. 173–180

[Tschannen u. a. 2018] Tschannen, Michael ; Bachem, Olivier ; Lucic, Mario: Recent advances in autoencoder-based representation learning. In: *arXiv preprint arXiv:1812.05069* (2018)

[Tucker 1966] Tucker, Ledyard R.: Some mathematical notes on three-mode factor analysis. In: *Psychometrika* 31 (1966), Nr. 3, S. 279–311

[Turian u. a. 2010] Turian, Joseph ; Ratinov, Lev ; Bengio, Yoshua: Word representations: a simple and general method for semi-supervised learning. In: *Proceedings of the 48th annual meeting of the association for computational linguistics* Association for Computational Linguistics (Veranst.), 2010, S. 384–394

[Van Hoof u. a. 2016] Van Hoof, Herke ; Chen, Nutan ; Karl, Maximilian ; Smagt, Patrick van der ; Peters, Jan: Stable reinforcement learning with autoencoders for tactile and visual data. In: *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* IEEE (Veranst.), 2016, S. 3928–3934

[Vasilescu und Terzopoulos 2002] Vasilescu, M Alex O. ; Terzopoulos, Demetri: Multilinear analysis of image ensembles: Tensorfaces. In: *European Conference on Computer Vision* Springer (Veranst.), 2002, S. 447–460

[Vershynin 2010] Vershynin, Roman: Introduction to the non-asymptotic analysis of random matrices. In: *arXiv preprint arXiv:1011.3027* (2010)

[Vinyals u. a. 2019] Vinyals, Oriol ; Babuschkin, Igor ; Chung, Junyoung ; Mathieu, Michael ; Jaderberg, Max ; Czarnecki, Wojciech M. ; Dudzik, Andrew ; Huang, Aja ; Georgiev, Petko ; Powell, Richard u. a.: Alphastar: Mastering the real-time strategy game starcraft ii. In: *DeepMind blog* (2019), S. 2

[Wang und Ahuja 2004] Wang, Hongcheng ; Ahuja, Narendra: Compact representation of multidimensional data using tensor rank-one decomposition. In: *vectors* 1 (2004), S. 5

[Wang u. a. 2017] Wang, Miaoyan ; Duc, Khanh D. ; Fischer, Jonathan ; Song, Yun S.: Operator norm inequalities between tensor unfoldings on the partition lattice. In: *Linear algebra and its applications* 520 (2017), S. 44–66

[Wang und Zhang 2012] Wang, Yu-Xiong ; Zhang, Yu-Jin: Nonnegative matrix factorization: A comprehensive review. In: *IEEE Transactions on knowledge and data engineering* 25 (2012), Nr. 6, S. 1336–1353

[Watter u. a. 2015] Watter, Manuel ; Springenberg, Jost ; Boedecker, Joschka ; Riedmiller, Martin: Embed to control: A locally linear latent dynamics model for control from raw images. In: *Advances in neural information processing systems*, 2015, S. 2746–2754

[Wedin 1972] Wedin, Per-Åke: Perturbation bounds in connection with singular value decomposition. In: *BIT Numerical Mathematics* 12 (1972), Nr. 1, S. 99–111

[Williams u. a. 2015] Williams, Matthew O. ; Kevrekidis, Ioannis G. ; Rowley, Clarence W.: A data–driven approximation of the koopman operator: Extending dynamic mode decomposition. In: *Journal of Nonlinear Science* 25 (2015), Nr. 6, S. 1307–1346

[Yeung u. a. 2019] Yeung, Enoch ; Kundu, Soumya ; Hodas, Nathan: Learning deep neural network representations for Koopman operators of nonlinear dynamical systems. In: *2019 American Control Conference (ACC)* IEEE (Veranst.), 2019, S. 4832–4839

[Zhang u. a. 2018] Zhang, Amy ; Satija, Harsh ; Pineau, Joelle: Decoupling dynamics and reward for transfer learning. In: *arXiv preprint arXiv:1804.10689* (2018)

[Zhang u. a. 2013] Zhang, Yangmuzi ; Jiang, Zhuolin ; Davis, Larry S.: Learning structured low-rank representations for image classification. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013, S. 676–683