Machine learning and data decompositions for complex networked dynamical systems

Bethany Lusch

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2016

Reading Committee:

Jose Nathan Kutz, Chair

Eric Shea-Brown

Steven Brunton

Program Authorized to Offer Degree:
Applied Mathematics

University of Washington

Abstract

Machine learning and data decompositions for complex networked dynamical systems

Bethany Lusch

Chair of the Supervisory Committee:
Professor Jose Nathan Kutz
Department of Applied Mathematics

Machine learning has become part of our daily lives. Its applications include personalized advertisements, stock price predictions, and self-driving cars. The goal of this thesis is to study ways to apply machine learning to complex dynamical systems in science. Each of the methods we discuss involves an optimization problem to fit a model to data. First, we study pairwise-conditional Granger causality, a popular statistical method in fields such as economics and neuroscience for inferring causal connections from time series data. We systematically test this method on data generated by a nonlinear model with known network structure. We find significant discrepancies between the original and inferred networks, unless the true structure is extremely sparse or dense. This work illustrates that network inference is a fundamentally challenging task which needs further innovative developments to be accurate. Second, we develop a specialized tensor decomposition to extract important spatial modes from a data set and sparsely fit time dynamics from an over-complete library to each spatial mode. This decomposition is more readily interpretable than others because the output includes the analytic forms of the time dimension. It is especially intended for data sets that other methods struggle with due to transient and intermittent phenomena. We demonstrate its usage on real crime and climate data. Finally, we simulate damage from traumatic brain injury and neurodegenerative disease on artificial neural networks used in deep learning. We use well-established biophysical data on focal axonal swellings to quantitatively study

the progress of impairments on our model of cognition. Our model provides intuitively appealing results about the manner in which cognitive impairments arise. Together, these methods demonstrate ways to use the frameworks of machine learning and optimization to study complex systems and advance science.

TABLE OF CONTENTS

LIST OF FIGURES

Figure Number Page

LIST OF TABLES

ACKNOWLEDGMENTS

First of all, I would like to thank my advisor, Nathan Kutz. He has been endlessly encouraging, and his enthusiasm is contagious. He truly cares about his students and makes sure he is helping us become better versions of ourselves.

I am much obliged to my collaborators. Thanks to Pedro Maia for being a dedicated mentor. He always finds time to offer advice to his many students. Thank you to Jake Weholt—it was great fun to discuss our research together. I am also grateful for Eric Chi. He is a wonderful example of patience and kindness.

Many thanks to my committee, Eric Shea-Brown, Steve Brunton, and Zelda Zabinsky. They always had thoughtful questions and suggestions.

There are many teachers and professors who prepared me for graduate school, especially in the University of Notre Dame math department. I would like to particularly thank Cindy Nagis and Bev Cange, who were my math teachers and math team coaches at Rosary High School. I was inspired to pursue math when they showed me that it could be about creative problem solving. I would also like to acknowledge Carl Meyer at North Carolina State University. His summer research program was instrumental in encouraging me to do a PhD.

I am grateful for my friends and family, especially for encouragement and advice from Susie Sargsyan and Jenny Taylor. Thank you to my parents, Mark and Jan Herwaldt, for sacrificing to make sure I had an excellent education and for listening to me talk about my studies endlessly. Finally, I would like to thank my husband Adam. I hope to someday reach his levels of patience and compassion.

This research was facilitated in part by a National Physical Science Consortium Fellowship and by stipend support from the National Security Agency.

DEDICATION

to Adam

Chapter 1

INTRODUCTION

According to Jim Gray, data-intensive science is a fourth paradigm of scientific exploration, complementing experimental, theoretical, and computational science [71]. Scientific approaches have broadened from describing natural phenomena with experiments to generalizing with theoretical models to simulating complex behavior. Now we have a wealth of data from experiments and simulations, and we need new approaches and technologies in order to take advantage of it.

Although there is much debate about the exact definition of "data science," it is certainly a field focused on taking advantage of large quantities of data. Many of the concrete successes of data science are happening inside tech companies, such as Netflix recommending movies, Facebook suggesting who to tag in photos, and Apple's Siri answering questions. On the surface, these product features may seem far removed from problems of scientific exploration, but some core approaches translate.

A major component of data science is machine learning, which itself is also difficult to define. In 1959, Arthur Samuel defined machine learning as a "field of study that gives computers the ability to learn without being explicitly programmed." [140] In 1997, Tom M Mitchell gave a more formal definition: "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E." [120] In the case of Facebook tagging, a program learns from tagged photos how to tag other photos, and its accuracy can be measured. Similar methods are being applied to assist doctors in accurately diagnosing disease from medical imaging. [58]

Statistical inference learns from data in the sense of drawing a more general conclusion about how the data was generated or the population that it was sampled from. Professor Larry Wasserman, who holds a joint appointment in the Statistics and Machine Learning departments at Carnegie Mellon, argues that there is essentially no difference between statistics and machine learning, but statistics tends to emphasize statistical inference in low-dimensional settings while machine learning emphasizes high-dimensional prediction. [165]

In Chapter 2, we study pairwise-conditional Granger causality, a causal inference method that can be used to infer networks from time series data. This method began in economics and is now also popular in neuroscience and other domains. We generate data from a nonlinear networked dynamical system so that we know the true underlying causal network. This enables us to test the accuracy of the method and show that it consistently produces results that have little relationship to the true network. We propose that our framework can be used to test other network inference methods.

In Chapter 3, we develop a tensor decomposition method to extract patterns from data. This method finds low-dimensional structure. We constrain a dimension (such as time) to be a sparse combination of candidate functions from an over-complete library. Since we built the library, we know the analytic form of the time dynamics, which makes the results more interpretable. We demonstrate the method on two real-world data sets from complex systems—85,622 crimes in Houston occurring over a period of eight months and six years of weekly sea surface temperatures over the Pacific Ocean.

In Chapter 4, we study convolutional neural networks (CNNs), a subfield of machine learning. This general method for prediction was originally inspired from knowledge of neuroscience. We use these networks as a model for understanding how traumatic brain injury and neurodegeneration may affect the brain's ability to perform classification tasks.

Chapter 2

INFERRING CONNECTIVITY IN NETWORKED DYNAMICAL SYSTEMS: CHALLENGES USING GRANGER CAUSALITY

This chapter is based on joint work with Pedro D. Maia and J. Nathan Kutz.

Determining the interactions and causal relationships between nodes in an unknown networked dynamical system from measurement data alone is a challenging, contemporary task across the physical, biological and engineering sciences. Statistical methods, such as the increasingly popular Granger causality, are being broadly applied for data-driven discovery of connectivity in fields from economics to neuroscience. A common version of the algorithm is called pairwise-conditional Granger causality, which we systematically test on data generated from a nonlinear model with known causal network structure. Specifically, we simulate networked systems of Kuramoto oscillators and use the Multivariate Granger Causality Toolbox to discover the underlying coupling structure of the system. We compare the inferred results to the original connectivity for a wide range of parameters such as initial conditions, connection strengths, community structures and natural frequencies. Our results show a significant systematic disparity between the original and inferred network, unless the true structure is extremely sparse or dense. Specifically, the inferred networks have significant discrepancies in the number of edges and the eigenvalues of the connectivity matrix, demonstrating that they typically generate dynamics which are inconsistent with the ground truth. We provide a detailed account of the dynamics for the Erdős-Rényi network model due to its importance in random graph theory and network science. We conclude that Granger causal methods for inferring network structure are highly suspect and should always be checked against a ground truth model. The results also advocate the need to perform such comparisons with

any network inference method since the inferred connectivity results appear to have very little to do with the ground truth system.

2.1 Introduction

In 1956, Norbert Wiener proposed a statistical notion of causality [167]: Y causes X if knowing the past of Y improves the prediction of X (as compared to using the past of X alone). In 1969, the Nobel Prize winning econometrician Clive Granger formalized this concept in the context of linear autoregressive modeling [55]. The resulting method is now commonly referred to as *Granger causality* (GC). The importance of understanding causal relationships in complex, dynamical networks from time-series measurements alone is clear: it becomes a fundamental tool for data-driven scientific discovery [82, 121, 132]. Methods to infer causality are the source of much debate and require entirely different statistical models from those used in associational inference [74]. Complicating the methodology is the fact that correlation does not imply causation. So, despite numerous methods for computing correlation, they only serve a limited role in understanding if there is an underlying causal relationship. In this manuscript, we consider a popular and commonly used form of GC to infer the connectivity in a known, networked system of Kuramoto oscillators. Our goal is to evaluate GC as a tool for data-driven scientific discovery. We demonstrate that the method is highly suspect, inferring connectivity and dynamics that are significantly different than the known ground truth model. With the ever-increasing demand to understand connectivity in dynamic networks, we hope that the results from this study will serve as a strong cautionary note to the broader scientific community using such statistical techniques for data-driven network inference.

Following Wiener's statistical innovations, the seminal work of Granger was originally defined in terms of two variables X and Y. However, it was quickly generalized to larger sets of variables where *pairwise-conditional Granger causality* could be computed among the variables. By checking for causal links between each pair of variables, the aim was to infer the most probable directed graph structure. Figure 2.1 illustrates this idea: each node in

the dynamical network generates its own time series data that is influenced by interactions with other nodes. In practice, we are usually limited to individual noisy recordings without knowledge of the underlying network connectivity—which is precisely what GC attempts to determine. This mathematical framework became popular in the economics community [76] for determining how nodes of a financial network might be influencing each other. For example, Hamilton [62] used GC as evidence that oil shocks were a contributing factor to recessions. More recently, it has risen in popularity in neuroscience [21] where Bressler et al. [22] used it to justify that activity in certain areas of the frontal and parietal lobes can predict visual processing activity before an anticipated visual stimulus. More broadly, pairwise-conditional GC is currently being used to infer networks of connectivity in many applications [5, 27, 174, 173, 26, 176].

The method is highly attractive in such systems due to the fact that there may be no other way to understand the underlying network of causal relationships. Attempts to infer causality have also led to numerous other statistical innovations for determining causality [82, 121, 132], including those leveraging independent component analysis [146] and network structure [134], for instance. A seminal recent contribution by Sugihara et al. [152] called convergent cross mapping (CCM) tests for causation by measuring the extent to which the historical record of Y values can reliably estimate states of X. The CCM method looks for the signature of X in Y's time series by seeing whether there is a correspondence between the attractor manifold built from Y and points in the X manifold, where the two manifolds are constructed from lagged (time-delay) coordinates of the time-series variables. This is a promising avenue especially for systems displaying a dynamical attractor. More recently, a formulation by Wahl et al. [162] has employed local linear models in the GC framework to resolve causal relationships in distinct regions of state space, leading to a promising technique for resolving overall GC structure.

As is still the case today, Granger's definition was met with controversy. Concerns have ranged from philosophical matters [56] to conceptual limitations [128] to analytical and practical implementation issues [152, 151]. Granger responded to criticism in 1980 [56] by

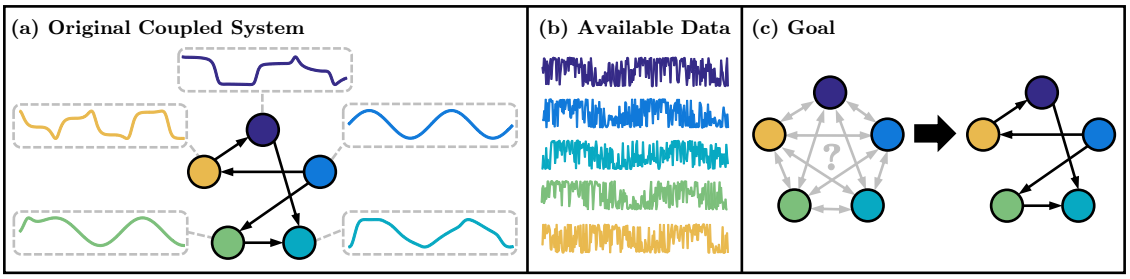

Figure 2.1: Inferring network connectivity via Granger causality analysis. (a) Schematics of a coupled dynamical system where the directed network architecture plays an important role. The time series generated by each node is influenced by its connectivity to other nodes. (b) In several applications, the connectivity structure is unknown, but noisy measurements from each node are available. (c) We use Granger causality to infer the original network structure from the noisy data.

arguing that although there is no consensus for the concept of causality, it is still worth choosing a specific and operational definition for the context of a written work or lecture. He suggested that GC should be viewed merely as evidence in a Bayesian sense. In 2003, he acknowledged in his Nobel Lecture that because his definition was pragmatic and easy to apply, "of course, many ridiculous papers appeared" (see [57]). Several concerns have led to variations in the methodology which we describe in Section 2.2. We will primarily consider the version called *pairwise-conditional* GC. We do not address theoretical or philosophical concerns with Granger causality. Instead, we accept it as a technical definition and evaluate its efficacy in inferring network structure. We use data generated from a known network of nonlinear Kuramoto coupled oscillators [98]. This is a canonical choice for studying synchronizable systems, such as power grids, pacemaker cells in the heart, pedestrian crowds, and coupled cortical neurons (see [41] and references therein). We generate random networks to reconstruct, sampling from the Erdős-Rényi family [44]. This is a well-studied network model [124] and provides a practical way to generate random networks with a large range of edge densities. We calculate the GC structure primarily using the Multivariate Granger

Causality (MVGC) Matlab Toolbox [14]. MVGC is a popular implementation of pairwise-conditional GC written with neuroscience data in mind [174, 5, 175, 172, 126, 27], but we also consider other numerical implementations of GC in order to cross-validate the results.

The outline of this chapter is as follows: Sections 2.2 and 2.3 provide all necessary background information for the GC framework and Kuramoto systems respectively. We describe our methodology in Section 2.4 and present a comprehensive list of results in Section 2.5. We summarize our conclusions in Section 2.6.

2.2 Background: Granger Causality

Granger causality (GC) is defined in the context of linear auto-regressive modeling, which computes the relationship of a time series with its own past. One important model that is used for multivariate stochastic processes is called the Vector Auto-Regressive (VAR) model. Let $\boldsymbol{X_t}$ be a vector-valued stochastic process with mean zero (averaging at each time t over the realizations). A VAR model for $\boldsymbol{X_t}$ is a sequence of $n \times n$ real matrices $\mathbf{A}_k$ and an n-dimensional white noise process (independently and identically distributed and serially uncorrelated) $\boldsymbol{\epsilon}_t$ such that

$$\boldsymbol{X_t} = \sum_{k=1}^{p} \mathbf{A}_k \boldsymbol{X_{t-k}} + \boldsymbol{\epsilon}_t. \tag{2.1}$$

The $\mathbf{A}_k$ matrices (called the *regression coefficients*) describe how $\boldsymbol{X_t}$ depends on its past and represent the predictable behavior of the process. The $\boldsymbol{\epsilon}_t$ process (called the *residuals*) represent the unpredictable behavior. We call p the *model order*. Note that fitting a VAR model to data does not imply that the data was generated by a VAR process.

The above formulation is often written as a first-order VAR model of the form $\tilde{\boldsymbol{X}}_{\boldsymbol{t}} = \boldsymbol{A}\tilde{\boldsymbol{X}}_{\boldsymbol{t-1}} + \tilde{\boldsymbol{\epsilon}}_t$ where

$$\tilde{\boldsymbol{X}}_{\boldsymbol{t}} = \begin{bmatrix} \boldsymbol{X_t} \\ \boldsymbol{X_{t-1}} \\ \vdots \\ \boldsymbol{X_{t-(p-1)}} \end{bmatrix}, \quad \mathbf{A} = \begin{bmatrix} \mathbf{A}_1 & \mathbf{A}_2 & \cdots & \mathbf{A}_p \\ \mathbf{I}_n & 0 & \cdots & 0 \\ 0 & \ddots & 0 & \vdots \\ 0 & 0 & \mathbf{I}_n & 0 \end{bmatrix},$$

and $\tilde{\boldsymbol{\epsilon}_t} = [\boldsymbol{\epsilon}_t, 0, \ldots, 0]^T$ with $\mathbf{I}_n$ being an $n \times n$ identity matrix. The spectral radius of a VAR model is defined to be the spectral radius of $\mathbf{A}$, $\rho(\mathbf{A})$. Recall that the spectral radius of a matrix $\mathbf{A}$ is defined as $\rho(\mathbf{A}) = \max\{|\lambda_1|, \ldots, |\lambda_n|\}$, where $\{\lambda_i\}$ are the eigenvalues of $\mathbf{A}$. The stability criteria for a VAR model is analogous to those for difference equations: $x^{(k+1)} = Tx^{(k)}$ is stable if and only if $\rho(T) < 1$. Thus a VAR model is stable if and only if $\rho(\mathbf{A}) < 1$.

The statistical basis of GC can be stated as follows: Y causes X if the past of Y improves the prediction of X as compared to only using the past of X. Specifically, if a stochastic process $\mathbf{Y}_t$ is used to predict $\mathbf{X}_t$, this can be written as

$$\mathbf{X}_t = \sum_{k=1}^{p} \mathbf{A}'_k \mathbf{X}_{t-k} + \sum_{k=1}^{p} \mathbf{B}_k \mathbf{Y}_{t-k} + \boldsymbol{\epsilon}'_t. \tag{2.2}$$

Then we say that Y *Granger-causes* X if Eq. (2.2) is a "better" prediction of X than Eq. (2.1). In particular, Y Granger-causes X if the variance of $\boldsymbol{\epsilon}'_t$ is statistically significantly lower than the variance of $\boldsymbol{\epsilon}_t$.

There are many variations on the original definition. Most formulations rely on representing data as a VAR model, although some differ significantly. Extensions include blockwise GC [164], partial GC [60], and piecewise GC [171]. Improvements for nonlinear time series are studied in [31, 169, 47, 115].

We focus on *pairwise-conditional* GC, specifically as implemented in the MVGC Toolbox [14]. GC might wrongly infer that Y Granger-causes X if there is a third, latent variable Z that influences both X and Y. To minimize this effect, we can "condition out" Z. We do this by changing Eqs. (2.1) and (2.2) to:

$$\boldsymbol{X_t} = \sum_{k=1}^{p} \mathbf{A}_k \boldsymbol{X_{t-k}} + \sum_{k=1}^{p} \mathbf{B}_k \boldsymbol{Y_{t-k}} + \sum_{k=1}^{p} \mathbf{C}_k \boldsymbol{Z_{t-k}} + \boldsymbol{\epsilon}_t \tag{2.3}$$

$$\boldsymbol{X_t} = \sum_{k=1}^{p} \mathbf{A}'_k \boldsymbol{X_{t-k}} + \sum_{k=1}^{p} \mathbf{B}'_k \boldsymbol{Z_{t-k}} + \boldsymbol{\epsilon}'_t. \tag{2.4}$$

We are considering the null hypothesis that $\mathbf{B}_1 = \mathbf{B}_2 = \cdots = \mathbf{B}_p = 0$. We calculate the

G-causality by considering the log-likelihood ratio

$$\mathcal{F}_{Y\to X|Z} := \ln \frac{|\Sigma'|}{|\Sigma|},$$

where $\Sigma = \mathrm{Cov}(\boldsymbol{\epsilon}_t)$ and $\Sigma' = \mathrm{Cov}(\boldsymbol{\epsilon}'_t)$. Thus, to check the causality between a pair of variables, we can condition out the other $n-2$ variables. In particular, if $\boldsymbol{U}$ is composed of n processes $U_{1t}, \ldots, U_{nt}$, we can compute *pairwise-conditional* causalities $\mathcal{G}_{i,j}(\boldsymbol{U}) := \mathcal{F}_{U_j\to U_i|\boldsymbol{U}_{[ij]}}$, where $\boldsymbol{U}_{[ij]}$ denotes omitting U_i and U_j, and perform a statistical test to determine which values $\mathcal{G}_{i,j}$ are large enough represent a causal relationship between U_j and U_i. In our network context, this is a directed edge from node j to node i. In the MVGC Toolbox [14], this is calculated using multiple representations of a VAR model. It computes causality both in temporal and frequency domains and returns an error message if the results do not match. See [14] for details.

Not all datasets lend themselves to GC analysis. The coefficients $\mathbf{A}_k$ of the fitted VAR model, for instance, must be square summable and stable [14]. Square summability implies $\sum_{k=1}^{p} \|A_k\|^2 < \infty$, which is trivially true for finite p. However, some stochastic processes may only be fit by a VAR with $p = \infty$. The MVGC Toolbox [14] does not provide a practical way to check this criterion, but mentions that violations may occur if the data contains a strong, slow moving average component. This may trigger a warning or an error.

According to [14], there are five likely reasons for problems with using GC on time series data: (i) *Colinearity*: If there are linear or nearly linear relationships between time series, the VAR representation will be ambiguous. This is likely to be detected by the toolbox and reported, stopping with an error. (ii) *Stationarity*: The data must be covariance-stationary. If the spectral radius of the estimated VAR model is larger than one, the GC analysis stops with an error. (iii) *Long-term memory*: If the autocorrelation does not decay exponentially, the data is unsuited to VAR modeling since it may silently yield spurious results. This may be detected when computing the autocovariance sequence where long-term memory typically manifests itself as power-law behavior. The sequence should decay exponentially when the process has a spectral radius less than one. However, there is a limit to how far the sequence

is calculated, and if the spectral radius is close to one, it may not decay below a specified tolerance within that length. In that case, the results may be inaccurate and a warning may be issued. (iv) *Moving average*: If the data contains a strong, slow moving average component, the coefficients might not be square-summable, the analysis may be invalid, and the toolbox will typically report warnings or errors. (v) *Heteroscedasticity*: If the variance of the residual terms depends on the values of the process, then the statistical inference is likely to suffer. It can invalidate standard statistical significance tests or confound G-causal inference. The toolbox does not offer any way to test or counteract this effect. All the results in this chapter were attained after running all of the diagnostic tests recommended in [14]. The toolbox did not return any errors in our runs. The only warnings given were from the autocovariance sequence not decaying sufficiently quickly, which we carefully annotated.

2.3 Background: Kuramoto Oscillators

Coupled oscillators have been of long-standing interest in the scientific community due to their ability to describe canonical phenomena such as synchronization. Yoshiki Kuramoto proposed one of the most well-studied systems modeling nonlinear coupled oscillators, the *Kuramoto oscillators*:

$$\dot{\theta}_i = \omega_i + \frac{K}{n}\sum_{j=1}^{n} A_{ij}\sin(\theta_j - \theta_i), \quad i = 1, \ldots, n. \tag{2.5}$$

In this model, the dynamics of the ith oscillator is governed by θ_i, which has a natural frequency ω_i. The n oscillators are coupled in a network with adjacency matrix $\mathbf{A}$ and coupling strength K. Depending on the parameters of the model, the oscillators may synchronize or exhibit chaotic dynamics. Kuramoto defined an order parameter to describe these different potential dynamics:

$$r(t) = \frac{1}{n}\left|\sum_{j=1}^{n} e^{i\theta_j(t)}\right| \tag{2.6}$$

where $r(t)$ varies from $O(1/\sqrt{n})$ to unity when synchronization occurs. When K increases, so does the average order parameter r. Figure 2.2 depicts the synchronization as a function

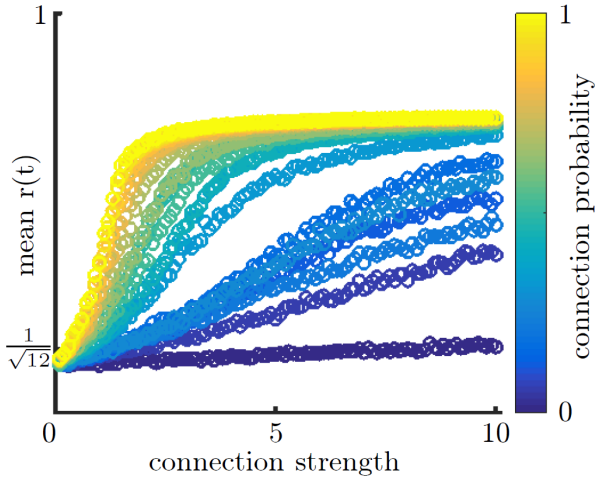

Figure 2.2: Increase in synchrony as connection strength increases. We generate random 12-node Erdős-Rényi networks with a range of connection probabilities. We then solve the Kuramoto model on each network for varying connection strengths. For each network and connection strength pair, we calculate the average order parameter $r(t)$ (Eq. 2.6). We see that as the connection strength increases, the synchrony also increases. However, for sparse networks, the network remains unsynchronized $\left(r(t) \approx \frac{1}{\sqrt{n}}\right)$ and for dense networks, the network synchronizes for moderate connection strength. This data was generated in Experiment C1 (See Section 2.5).

of strength and probability of connection in a 12-node Erdős-Rényi network.

Figure 2.3 depicts several two-oscillator examples. Synchronization occurs if both oscillators converge to the same frequency. Depending on the network structure, they may converge to one oscillator's natural frequency or an average of the two. Notice how the cases with exactly one edge appear to match Granger's definition of causality; the dominating oscillator predicts itself, but the other oscillator is strongly influenced by it. Figure 2.4 exemplifies a Kuramoto system with twelve nodes connected in two disjoint communities. The blue community synchronizes to a slow frequency ω_A and the green community synchronizes to a fast frequency ω_B. The order parameter $r(t)$ considers synchronization across the whole

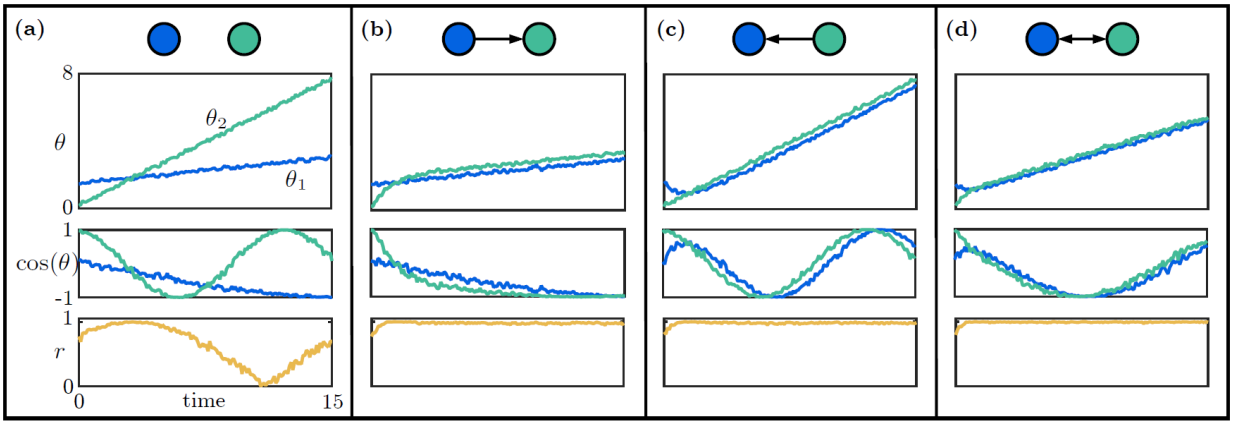

Figure 2.3: A pair of coupled Kuramoto oscillators with distinct natural frequencies. We show the four possible network architectures in panels (a)–(d). We first plot θ_1 and θ_2, the solution of the differential equations in Eq. (2.5). We then plot $\cos(\theta_1)$ and $\cos(\theta_2)$, the more natural way to view oscillators. In panel (a), the oscillators are uncoupled, so they merely oscillate with their natural frequency. However, in panels (b)–(d), we see cases leading to synchronization. The overall synchronization of the network can be summarized by the parameter $r(t)$ with full synchronization achieved when $r(t) = 1$ (see Eq. (2.6)).

network, making it difficult to interpret (black line). If we evaluate $r(t)$ on each community separately (the blue and green lines) we see that each community synchronizes with itself.

In this chapter, we simulate the Kuramoto model and use Granger causality (GC) to infer the adjacency matrix $\mathbf{A}$. As demonstrated in Figs. 2.3–2.4, the network structure influences the system dynamics. We expect the dynamics to preserve signatures of the network architecture and for GC to potentially discover these connections. Because we know the ground-truth data, our model guarantees that there are no external or hidden variables influencing the system. However, as we will see in Section 2.5, GC will consistently fail to recover the known connectivity. We are not the first to apply GC methods to Kuramoto systems. Angelini et al. [7, 6] develop a version of GC that does not use VAR modeling and is specialized for circular variables, using Kuramoto oscillators as an example. Wu et al.

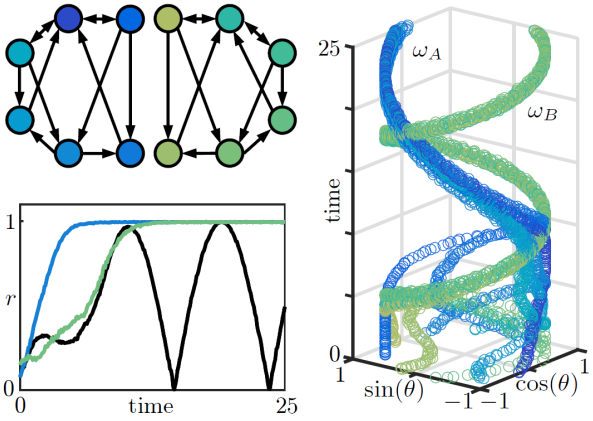

Figure 2.4: Example of synchronicity in structured Kuramoto networks. We have two disjoint subnetworks. The blue oscillators have frequencies with average -0.2 while the green oscillators have average frequency 0.5. As we see in the right panel, the individual trajectories collapse. The blue nodes synchronize to frequency ω_A and the green nodes synchronize to frequency ω_B. In the lower left, we see that the measure of synchronicity for each community, $r(t)$ (Eq. (2.6)), approaches one but at different synchronization times. However, when $r(t)$ is evaluated on the entire network, we do not achieve total synchronicity because the two communities are not connected.

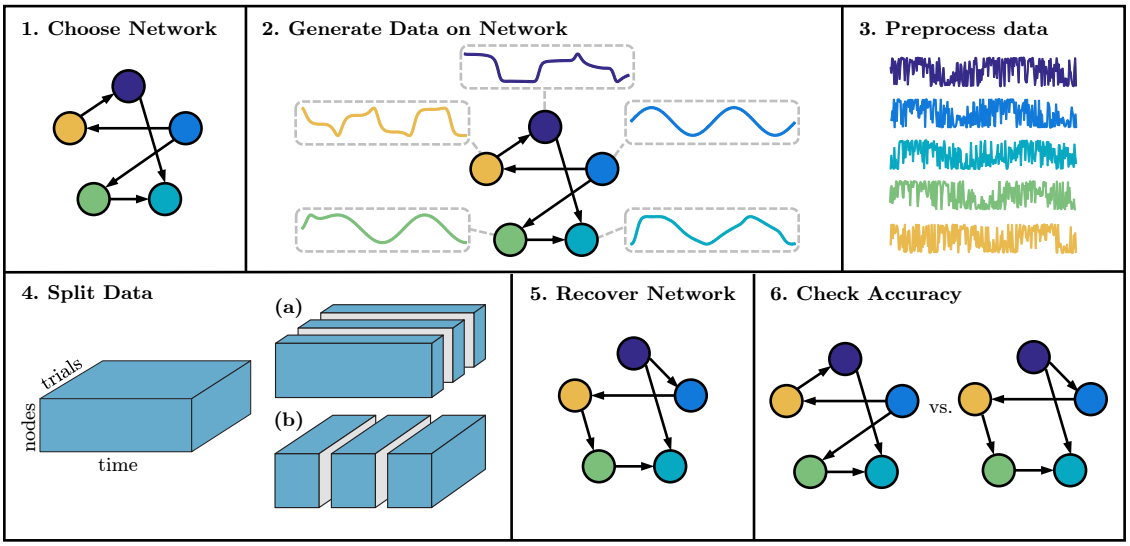

Figure 2.5: Overview of steps in our methodology. We will experiment with varying the decisions made in each step—see Section 2.4.

[170] develop an algorithm for inferring a network of Kuramoto oscillators using piecewise GC [171] followed by a pruning of edges.

2.4 Numerical Experiments

We test Granger causality (GC) by applying it to data generated from the Kuramoto model Eq. (2.5) with the goal of reconstructing the network adjacency matrix A. We split our methodology into six steps; see Fig. 2.5 for a schematic overview. We explore several options at each step to avoid limiting ourselves to the best or worst cases for GC performance. However, as we will show in Section 2.5, network reconstruction is consistently poor, usually without warnings from the toolbox. The following specific steps are taken in our evaluation algorithm.

1. *Choose Network.* We set up a network with n nodes that we wish to reconstruct. Our default value ($n = 12$) yields a sizable network while performing simulations in a timely

manner. We tried other values for comparison ($n = 2, 6$ and 24). See Exp. A1–A2 and C2–C3 for details.

In most experiments, we generate Erdős-Rényi networks; each potential edge is included with constant probability p [44]. We vary $p = 0.05, 0.1, \ldots, 1$ to address how the density of the network affects GC results.

2. *Generate Data on Network.* We simulate several Kuramoto systems with a variety of parameters:

   - Connection strength K. By default, we consider $K = 0.5, 1, 2, 4, 8$ to span GC reconstructions ranging from underestimation to overestimation of edges (see Fig. 2.9). Experiments A1–A2 and C1 display a wider range of K values.
   - Initial conditions θ^0: randomly sampled from uniform distributions. We reset them for each trial. This is reasonable for real data and additionally helps the data have a constant mean when averaging over trials (a requirement for being covariance-stationary). Our default distribution is $[0, 2\pi]$ since we will apply cosine to the data, which has a period of 2π. In Exp. C4–C5, we shift this distribution for comparison.
   - Natural frequencies ω: randomly sampled from uniform distributions. We reset them for each trial. A uniform distribution of $[-1, 1]$ is used in some studies of Kuramoto oscillators [29, 20]. However, when we used that range of natural frequencies, the toolbox gave many warnings (see Experiment C7). We, therefore, shifted the distribution to $[0, 2]$ for most experiments. See Exp. C6–C7 for comparisons to other distributions.
   - Number of trials N (each from solving the Kuramoto model once with random initial conditions and natural frequencies). Our default is $N = 100$, but we consider other values in Exp. C8–C9.

- Data sampling rate: $[0, T]$ with time step Δt, giving $m = T/\Delta t$ time points. Our default values are $\Delta t = 0.1, T = 25$, and $m = 250$. These were chosen by searching the parameter space for cases with no warnings and low error. We compare to other values in Exp. C10–C20.

3. *Preprocess Data.* We add noise to our simulations since measurement errors are expected in most applications, and it helps the data be more covariance-stationary. Specifically, we add white Gaussian noise of strength s, i.e., a constant power spectral density of s^2. Each one of the N random trials will have different noise realizations. Our default value of s is 2.5, based on experiments to minimize the error in the results, but other values are compared in Exp. C21–C22.

   Next, we usually apply cosine: instead of using $\theta_1, \ldots, \theta_n$, we use $\cos(\theta_1), \ldots, \cos(\theta_n)$. This is a natural way to view oscillations (see Fig. 2.3) and remove linear trends. We explore alternatives in Exp. C23–C24.

4. *Split Data.* At this point we already generated a "cube" of data with N random trials, each with a time series of length m for each of the n oscillators. As pictured in Step 4 of Fig. 2.5, we could apply GC to the whole cube of data at once. However, we have the option to split the data cube into smaller cubes by (a) splitting trials into smaller sets or (b) splitting the time into smaller time intervals. We apply GC to each one of the smaller cubes, letting them "vote" for edges. We include a directed edge if at least half of the voting networks include it.

   Barnett and Seth [14] suggest splitting the data into smaller time intervals for making it covariance-stationary. This is also the idea behind piecewise GC [171]. Splitting trials may reduce error if the subsets are each sufficiently large for reasonable inference. Then the rationale is that the process would become more robust when considering each "vote."

   We experiment with splitting data and voting in Exp. C25–C30.

5. *Recover Network with Granger Causality.* We use the MVGC Toolbox to recover a network from our data. Alternative implementations of GC are explored in Section 2.5.3.

6. *Check Accuracy.* Finally, we compare the GC estimated network with the ground truth. Our standard error metric is the percentage of wrong edges. For n nodes, there are $n^2 - n$ potential directed edges. We add the number of false-positive and false-negative edges and divide by $n^2 - n$. We consider other error metrics in Section 2.5.3.

We list all parameter choices of the exhaustive computational exploration in Table 2.17.

2.5 Network Reconstruction Results

We summarize our four classes of numerical experiments in Table 2.1. In Section 2.5.1 we consider a pair of oscillators, as pictured in Figure 2.3. In Section 2.5.2, we consider the network structure with two independent communities from Fig. 2.4. Finally, in Section 2.5.3, we generate random Erdős-Rényi networks. For each experiment, we make choices for all six steps described in Section 2.4, which are detailed in Table 2.17. For purposes of reproducibility, all MATLAB codes constructed are available online at `github.com/BethanyL/gc`.

2.5.1 Two-Node Networks

Experiments A1–A2 investigate a simple, two oscillator system. This could be an example in economics, such as the relationship between oil shocks and recessions. We try all possible 2-node networks (see Fig. 2.3) and vary the parameters of the system Eq. (2.5) with $n = 2$. The parameter choices for our experiments are summarized in Table 2.17. We present the results from Experiment A1 as a confusion matrix in Table 2.2. Each row shows the distribution of output networks for a given true network. If the method perfectly recovers the connectivity of all of the networks, this matrix would have entries of 100% along the diagonal. Instead we see that networks with one edge are rarely recovered correctly. The method has a tendency to overestimate the number of edges. We will see in later experiments that this pattern

Ref.	**Class**	**Figures**	**Tables**
A1–A2	Two-node networks	2.3, 2.6	2.2, 2.17
B1–B3	Two independent communities	2.4, 2.7, 2.8	2.17
C1–C30	Erdős-Rényi $(*)$	2.2, 2.9, 2.10, 2.11, 2.12, 2.13, 2.16	2.4, 2.5, 2.17
D1–D2	Erdős-Rényi $(**)$	2.14, 2.15	2.17

Table 2.1: Summary of our four classes of numerical experiments. Full details are in Table 2.17.

* changing parameters

** changing implementation

Truth \ Estimate				
	93%	4%	2%	1%
	9%	2%	0%	89%
	7%	0%	2%	91%
	4%	0%	2%	94%

Table 2.2: Performance (%) of Granger causality in two-node Kuramoto oscillator example. Confusion matrix. The table summarizes reconstruction results for Experiment A1: four different true networks and 100 values of connection strength. As we will continue to see in larger networks, the performance is weakest when the network is not extremely sparse or dense. The specific parameter choices for this experiment are in Table 2.17.

continues as the size of the network increases; performance is weakest when the number of edges is not extremely low or extremely high.

It may be argued that there was too much noise on the data for accurate connectivity reconstruction. We decrease the noise strength to 0.5 for Experiment A2, since in this case, the low noise does not cause warnings in the MVGC Toolbox. Figure 2.6 compares the results from these two experiments. We find that, perhaps counter-intuitively, the error is higher with lower noise. In particular, the lower noise results in even more overestimation of edges. Another pattern that will persist for larger networks is that as the connection strength increases, so does the number of edges inferred.

2.5.2 Two-Community Example

For Experiments B1–B3, we return to the Two-Community example in Fig. 2.4. In Experiment B1, we try to reconstruct a 12-node network (Eq. (2.5) with $n = 12$) using Granger causality on the same data plotted in the right panel of Fig. 2.4. The full parameter choices

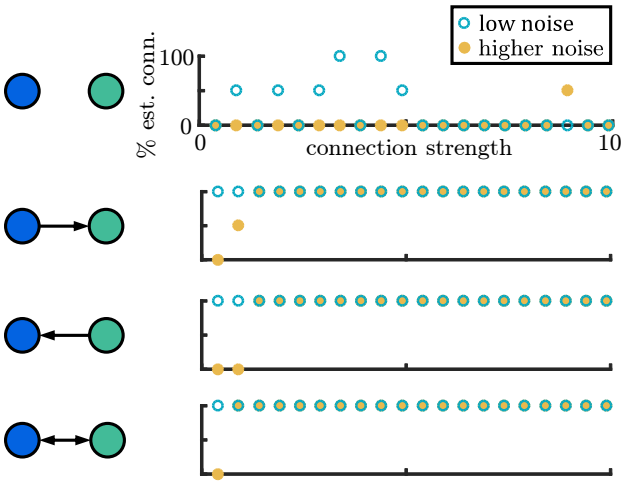

Figure 2.6: Percentage of inferred edges as the connection strength varies. We try all four possible 2-node networks (0, 1, or 2 edges), and we vary the connection strength across the horizontal axis. We also try two amounts of noise. The teal circles are for Experiment A1 (low noise) and the orange closed circles are for Experiment A2 (higher noise). The specific parameter choices for these experiments are in Table 2.17. The error is higher for low noise. As the connection strength increases, so does the number of inferred edges, a pattern that will continue for larger networks.

for this experiment are given in Table 2.17. The resulting network is shown in Fig. 2.7 (b). There are many extra edges and some missing edges, resulting in an error of 25%. Note that the community structure is lost even though it is clear from the plot of the data in Fig. 2.4 that the blue and green nodes synchronize separately. An error of 25% may sound reasonable, but visually comparing the two networks suggests that the error is significant. Similar error percentages are used as evidence of a Granger causality variation working well in papers such as [170].

Addressing Warnings. The MVGC Toolbox did produce warnings for Experiment B1, so for Experiment B2, we increased the noise to a strength of $s = 2.5$, leaving the rest of the parameters the same. The new data did not cause any warnings, and the resulting network is shown in Fig. 2.7 (c). This network had an error of 22%. It is missing many edges but also added some, including connecting the two communities.

Varying Time Sampling. In Experiment B3, we tried solving the Kuramoto model again but after halving the step size Δt. Generally, we hope that algorithms are stable, i.e., that small changes in the input will lead to small changes in the output. However, changing the time sampling led to a vastly different estimated network. Again, the data did not cause any warnings, but this time, the number of edges were vastly overestimated, as shown in Fig. 2.7 (d). This network has an error of 30%.

Checking Autocovariance Decay. If the autocovariance sequence does not decay exponentially, the data is not suitable for VAR modeling. This should be detected by the toolbox, but as a verification, we plot the required exponential decay in Fig. 2.8. The parameter choices for Experiments B1–B3 are summarized in Table 2.17.

2.5.3 Erdős-Rényi Networks

In our remaining experiments (Experiments C1–C30 and D1–D2) we consider random Erdős-Rényi networks. For each experiment, we vary p, the probability that a directed edge exists, and we vary K, the connection strength. See Table 2.17 for all of the parameter choices. Experiments C2–C30 are small variations on Experiment C1. A sampling of the results for

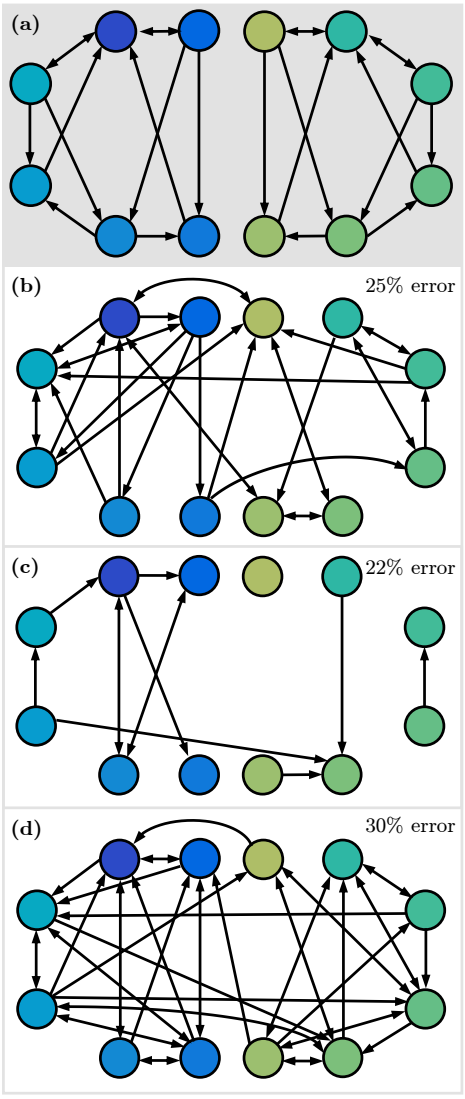

Figure 2.7: Results of Granger causality inference on the Two-Community network. Panel (a) depicts the true network. The resulting network from Experiment B1 in panel (b) has many extra connections and even connects the two separate communities, but the MVGC Toolbox [14] provides warnings. In Experiment B2, we increase the noise and try again, producing the network in panel (c) without warnings. This network is missing many edges but also connects the two communities. In Experiment B3, we keep the higher level of noise but halve the time step, resulting in the network in panel (d) without warnings. We again have vast overestimation of edges and the community structure is lost. The specific parameter choices for these experiments are in Table 2.17.

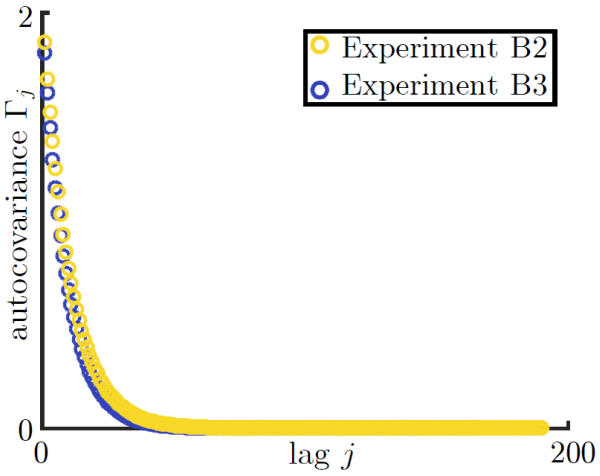

Figure 2.8: Autocovariance decay for Experiments B2 and B3. In these two experiments, the toolbox does not provide warnings, but there are significant errors (Fig. 2.7). Here we plot the autocovariance sequence for each experiment to demonstrate that it decays exponentially, as required.

Experiment C1 are shown in Fig. 2.9. In panel (a), we plot the percentage of true edges against the percentage of estimated edges. If the density of edges was inferred correctly, the results should match the identity line (the diagonal dashed line). The next assessment is whether or not the edges inferred were actually the correct ones. However, we generally do not even estimate the correct number of edges. Just as we saw with the two-node case in Fig. 2.6 and Table 2.2, the number of edges is most accurate for the extremes—very sparse or very dense. Another general pattern persists: for lower connection strength, pairwise-conditional GC underestimates the number of edges, and for higher connection strength, pairwise-conditional GC overestimates the number of edges.

The sparse and dense limits of connectivity are the only two regions where the inferred number of connections is somewhat consistent with the ground truth. In panels 2.9(b) and 2.9(c), we consider these two limiting network cases more closely. In particular, they are marked by two vertical dashed lines in panel 2.9(a). Here we plot the connection strength

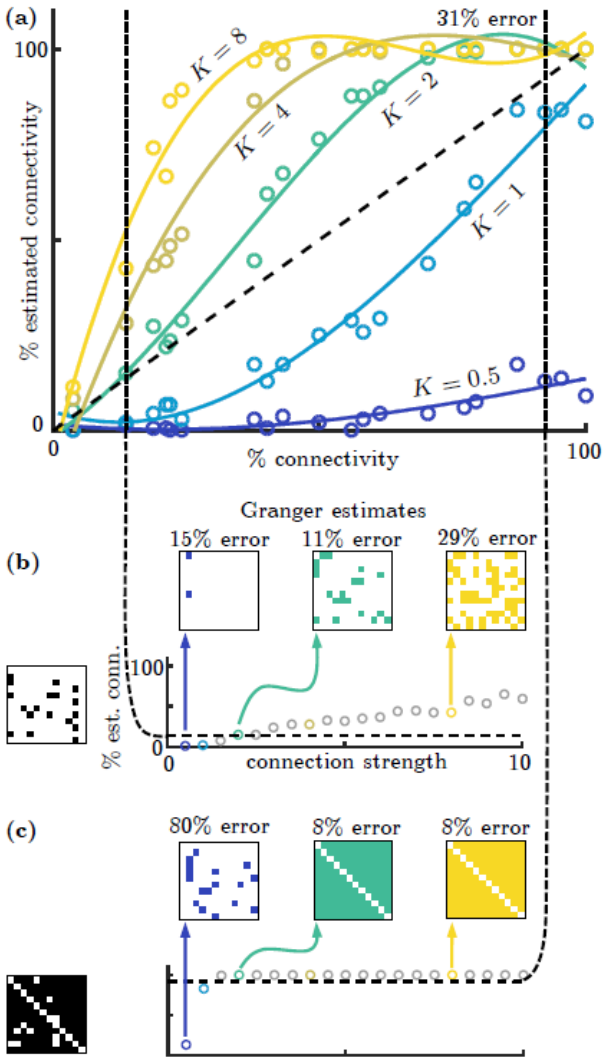

Figure 2.9: Results from Experiment C1. Here $n = 12$ and twenty different Erdős-Rényi networks are generated while varying the percentage of connections. We also vary the connection strength K. In panel (a), we plot the true percentage of connections versus the estimated percentage of connections for five values of K. If the percentage of connections was correct, our points would be on the dashed diagonal line. However, they may still have the wrong edges even if the correct number are inferred. For the sparse and dense cases, a varying connection strength K is considered in panels (b) and (c). The correct percentage of connections is plotted as a horizontal dashed line for reference. In the inset plots, we see some examples of the inferred networks. These are colored visualizations of the adjacency matrices. White squares denote zeros (no edge) and colored squares denote ones (an edge).

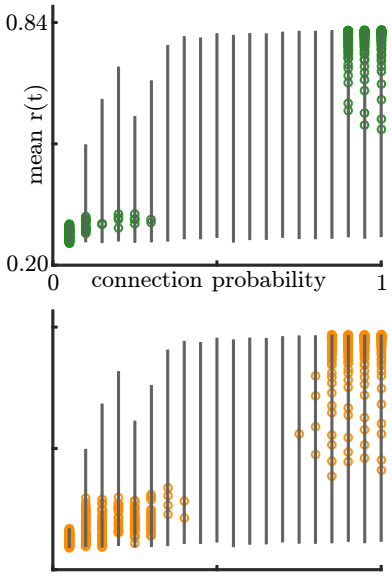

Figure 2.10: Optimal bands of synchrony. We consider the results of Experiment C1 in terms of average r, the synchrony measure in Eq. (2.6). For each connection probability p across the horizontal axis, we plot a gray line showing the range of average synchrony r attained as we varied the connection strength K. We then plot green circles in panel (a) for the values of r for which the error was less than 10%. We also plot orange circles in panel (b) for the values of r for which the error was less than 20%.

against the percentage estimated connectivity for a broader range of K values. The correct percentage connectivity is marked with horizontal dashed lines. We see again that for low connection strengths, the number of edges is underestimated. As K increases, our density estimates go from underestimating to overestimating the connectivity. We can visualize the exact networks inferred for three values of K, $K = 0.5, 2$, and 8. We see that even when the density of edges is approximately correct, the actual chosen edges do not match.

The Erdős-Rényi networks can also be analyzed from the viewpoint of the synchrony metric. We consider how our results relate to the strength of connection and the average order parameter $r(t)$ (Eq. (2.6)) for each data set. In the top plot of Fig. 2.10, we consider

each of the 20 networks, plotted by the connection probability p used to generate them. For each network, we generated the data for 100 values of connection strength K. In general, higher connection strength K means that the network is more likely to synchronize—a higher average $r(t)$ is produced (See Fig. 2.2). The full range of average $r(t)$ values attained for each network is plotted as a gray line segment. We see that for sparse networks, high synchronization was not attained for any value of K in our range. This makes sense, since $r(t)$ measures synchronization over the entire network, and a sparse network will not even be fully connected. On the other hand, we see that dense networks attain a wide range of synchronization as K is varied. We then checked for the cases where the percentage wrong was under ten percent and plotted them as green circles. We see that for sparse networks, the error is best when the network does not synchronize. For dense networks, the error is best when the network does synchronize strongly. For medium-density networks, the error is never below 10%. In the bottom panel of Fig. 2.10, we check a looser standard—plotting all cases with the error below 20%. We see that the general pattern continues.

Many variations to the experiment can be performed, including varying the number of nodes, percentage of connectivity in the Erdős-Rényi network and the strength of connections. These variations are summarized in Table 2.17. Figure 2.11 demonstrates the connectivity results as a function of network size. The computations show that the qualitative behavior does not change with n. We can also vary the θ^0 (initial condition) distribution, the ω (natural frequency) distribution, and the number of trials N, the third dimension of our data cube. These results are shown in Fig. 2.12 We note that having both positive and negative natural frequencies seems to cause many warnings, perhaps due to different synchronization effects. The number of trials has a large impact on the number of edges inferred—as N increases, so does the number of edges. The other variations in the experiments do not change the qualitative shape of the results. We also modify how we sample in time. In Fig. 2.13, we vary the time step Δt down the rows and vary the end time T across the columns. This means that the number of observations m is different in each plot.

Extensive computational experiments also considered varying the noise added to the data

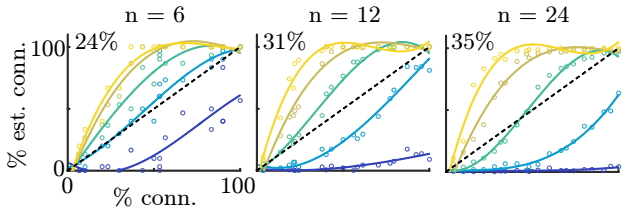

Figure 2.11: Varying number of oscillators. Here we plot the percent connectivity vs. estimated percent connectivity and five values of connection strength K for three numbers of oscillators. Left to right, we compare six, twelve, and twenty-four oscillators. The general pattern is consistent, but the average error seems to grow with the number of oscillators.

(including adding it before or after applying cosine), differencing and detrending of the data instead of applying a cosine (recommended by [14] for making data covariance-stationary), splitting the data and weighting multiple inferences of network structure. In all these cases, the same trends as shown in the preceding figures hold, i.e. as the connection strength increases, so does the number of edges inferred. In no case does the GC method produce accurate results.

The MVGC Toolbox is a successor to the Granger Causal Connectivity Analysis (GCCA) toolbox [144]. The newer toolbox adds more diagnostic warnings and errors and is intended to be more accurate. For Experiment D1, we rerun Experiment C1 with the GCCA toolbox and obtain very similar results (Fig. 2.14). We also try three other implementations of Granger causality. First, we consider the implementation of the classic Granger causality test (GCT) [109] provided with the [139] paper. This implementation only accepts one trial at a time ($N = 1$). They compare it to other network inference procedures, including the MVGC toolbox [14], on data generated by three models. The first two models are simply VARs. The third adds latent and exogenous variables. Although this is not explicitly stated, it seems that all implementations correctly infer the first two network models but sometimes make mistakes on the third. The focus of the paper is on whether the methods are consistent over repeated trials. They state that the MVGC toolbox [14] is *anomalous* in its lack of compliance

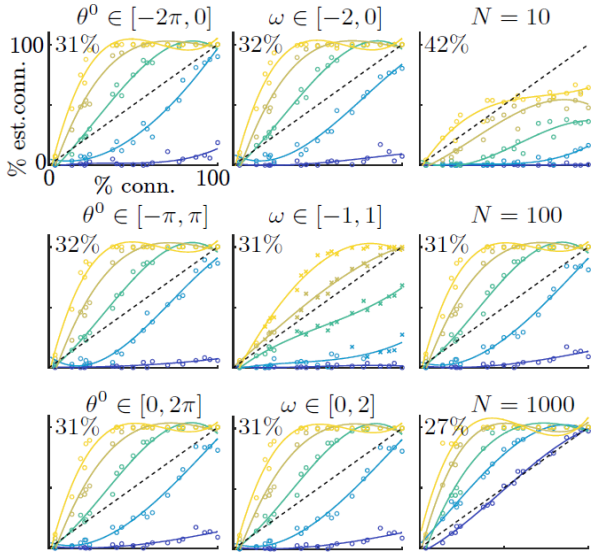

Figure 2.12: Varying system parameters. We plot the percent connectivity vs. estimated percent connectivity and five values of connection strength K while changing the distribution of initial conditions, the distribution of natural frequencies, and the number of trials. In the first column, we vary the distributions of random initial conditions, and in the second column, we vary the distributions of random natural frequencies. In the third column, we change the number of trials. The general pattern is consistent except when the number of trials is varied; the number of edges inferred grows as the number of trials grows. Results accompanied by a warning are marked with an "x" instead of a circle.

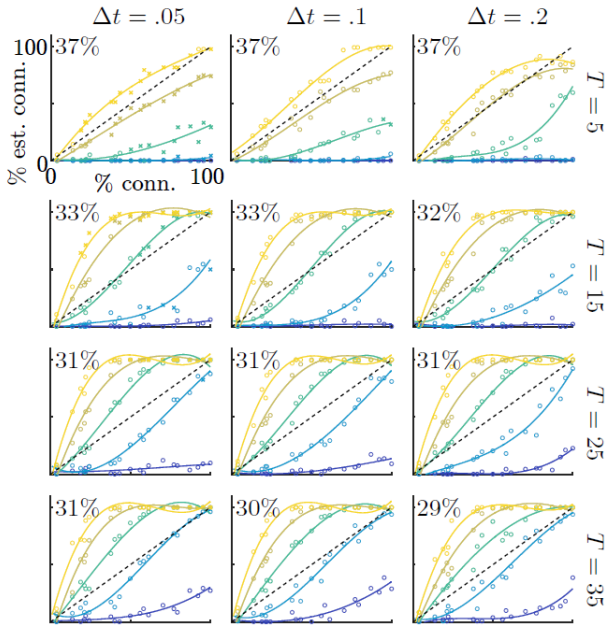

Figure 2.13: Varying data sampling in time. We plot the percent connectivity vs. estimated percent connectivity and five values of connection strength K while changing the data sampling in time. We use data from time 0 to T, where T varies down the rows. We use a time step of Δt where Δt varies across the columns. The general pattern is consistent except when the end time T is small. Results accompanied by a warning are marked with an "x" instead of a circle.

Method	Acronym	Reference
Multivariate Granger Causality	MVGC	[14]
Granger Causal Connectivity Analysis	GCCA	[144]
Granger Causality Test	GCT	[139]
Extended Granger Causality	eGC	[142]
eGC toolbox for standard GC		

Table 2.3: Network inference toolboxes and methods compared in our simulations.

to Neyman-Pearson criteria. We additionally consider a version of Granger causality called *extended Granger causality* that allows instantaneous causal relationships (zero-lag) [142]. The paper is accompanied by two implementations, one that includes zero-lag relationships (which we refer to as *Schiatti eGC*) and one that does not (which we refer to as *Schiatti GC*). These implementations also only accept one trial at a time ($N = 1$). They are tested on data generated by an extended VAR model that allows for zero-lag relationships. The methods are then compared on real data where the true network structure is not known. Table 2.3 shows the methods compared along with their commonly used acronym and initial source reference.

In order to test implementations that only accept one trial at a time, in Experiment D2, we generate 50 trials, separate them into sets of $N = 1$, and then vote over the 50 estimates. We compare the MVGC [14], GCCA [144], GCT [139], Schiatti eGC and Schiatti GC [142] implementations. We see that in all five implementations, almost no edges are kept after voting (Fig. 2.15). Although individual estimates contain some correct edges, the methods are not sufficiently consistent to estimate the same edge at least half of the time. This suggests that it is not sufficient to test a new version of Granger causality on data generated by a VAR model.

Thus far, we have evaluated the accuracy of our results in three ways: visually comparing

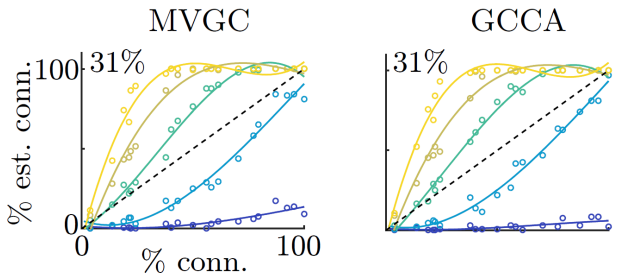

Figure 2.14: Comparing implementations. Here we plot the percent connectivity vs. estimated percent connectivity and five values of connection strength K. In Experiment D1, we repeat Experiment C1 with the GCCA implementation and observe very little difference.

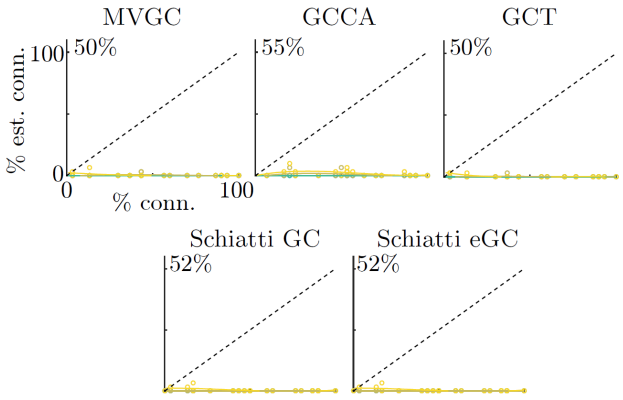

Figure 2.15: Comparing implementations. Here we plot the percent connectivity vs. estimated percent connectivity and five values of connection strength K. In Experiment D2, we compare the MVGC Toolbox to other implementations. We observe that each method infers very few edges when only considering only one trial at a time and voting over 50 sets.

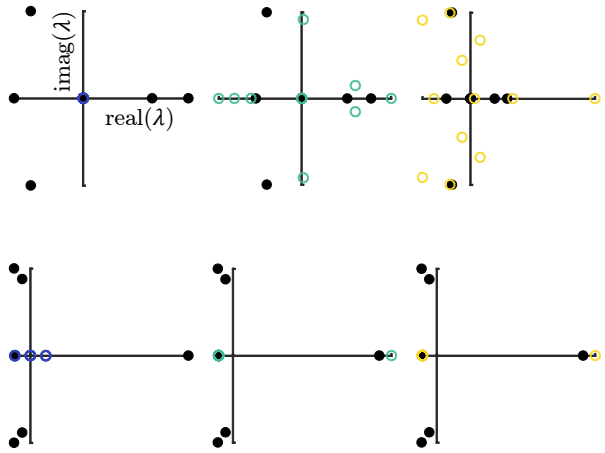

Figure 2.16: Eigenvalue comparison. One measure of accuracy is how well the estimated network $\tilde{A}$ would recreate the same dynamics as the true network A. We therefore compare the eigenvalues of $\tilde{A}$ (open colored circles) and A (closed black circles) for the six estimated networks in Fig. 2.9. We see many cases of eigenvalues being significantly wrong. For example, in the first plot, the inferred network has only eigenvalues of about 0, missing eigenvalues for significant growth.

the original network to the estimated network (as in Figures 2.7 and 2.9), comparing the percentage connnectivity to the percentage estimated connectivity, and calculating an error—the percentage of potential edges that are correctly labeled as an edge or not an edge. Perhaps in some applications, what is important is reconstructing a network that would produce similar dynamics. Thus, we may be concerned with comparing the eigenvalues of the estimated network $\tilde{\mathbf{A}}$ to the original network $\mathbf{A}$. We return to the six estimated networks in Fig. 2.9: two true networks and the corresponding estimates when $K = 0.5, 2$, and 8. We plot the eigenvalues of the true network with the eigenvalues of the estimated network in Fig. 2.16. We see that even when the densities are relatively correct, the dynamics produced by the connectivity matrix would be significantly wrong.

In [16], Billio et al. use a form of Granger causality to infer a network of financial institutions. Then they propose various econometric measures of connectedness to assign ranks to institutions and predict financial loss. One metric used to rank is closeness: the average distance from node j to the remaining nodes, where unconnected nodes are defined to have the maximum distance, $n-1$. In Tables 2.4 and 2.5, we use this metric to assign ranks to the twelve nodes in our examples from Fig. 2.9. We see that a ranking formed from the estimated networks has little relationship with the true ranking. This once again shows that the GC method fails to capture meaningful results concerning the ground truth network.

2.6 Conclusions

The inference of causal structure from time series measurements remains one of the most challenging tasks in data-driven discovery across the sciences. It has become especially important in the emerging area of network science for understanding how different dynamical nodes of a system interact to produce overall network functionality. A variety of statistical methods have been instrumental in developing mathematical architectures for inferring connections between nodes. These methods often make assumptions about the physical processes generating the data and the form of the connections (e.g. linear). Foremost among these methods is pairwise-conditional Granger causality as it has been used extensively across a variety of disciplines [76, 62, 21, 22, 5, 27, 174, 173, 26, 176]. We consider a nonlinear, networked dynamical system of Kuramoto oscillators as a ground-truth test model for inferring network connectivity and demonstrate that without exception, Granger causality gives highly inaccurate results for the inferred causal relations and the eigenvalues of the connectivity matrix. This is consistent with an additional study of the quantitative accuracy of the GC method [128]. This is an important assessment of the statistical efficacy of the GC method, and it further suggests that it should be carefully validated before use with any networked time series data.

The Kuramoto oscillator model is chosen for consideration for this study as it has become

node	true rank	estimated, $K = 0.5$	estimated, $K = 2$	estimated, $K = 8$
1	12	1	8	3
2	11	3	9	12
3	3	3	4	3
4	10	3	5	3
5	1	3	5	7
6	6	3	10	10
7	4	1	7	8
8	1	3	2	1
9	5	3	10	11
10	6	3	2	3
11	9	3	1	2
12	6	3	10	8

Table 2.4: Closeness Ranking, Part I. Closeness has been used to rank institutions on a network. We return to Fig. 2.9 and compare the closeness ranking on the first true network to the closeness rankings on three estimates. We see that the rankings formed from the estimated networks have little relationship with the true ranking.

node	true rank	estimated, $K = 0.5$	estimated, $K = 2$	estimated, $K = 8$
1	6	11	1	1
2	1	9	1	1
3	1	9	1	1
4	6	3	1	1
5	1	5	1	1
6	1	6	1	1
7	6	2	1	1
8	6	1	1	1
9	11	11	1	1
10	6	6	1	1
11	12	3	1	1
12	1	8	1	1

Table 2.5: Closeness Ranking, Part II. Closeness has been used to rank institutions on a network. We return to Fig. 2.9 and compare the closeness ranking on the second true network to the closeness rankings on three estimates. We see that the rankings formed from the estimated networks have little relationship with the true ranking.

a canonical model in networked dynamical systems. It has simple oscillatory behavior that is influenced by its interaction structure. Both synchronization, partial and complete, and chaotic behavior is possible in the network. Given that we can specify a ground truth connectivity structure, the GC method can be used to test the efficacy of the inference method. We observe that as the connection strength or number of trials increases, we transition from underestimating the number of edges to overestimating the number of edges, quickly surpassing the correct number. This pattern is consistent over variations in parameters, and individual networks inferred are not consistent with the ground truth (See Fig. 2.7). This suggests that the algorithm is not stable; small changes in the input data lead to large changes in the estimated network. Accuracy is best on very sparse or very dense networks, although arguably still not sufficient. Perhaps the errors are controlled in very sparse networks because the network overall does not synchronize even for high connection strength, mitigating confusion from synchronized but unconnected nodes. (See Fig. 2.10.) Similarly, perhaps the errors are limited in dense networks because the networks become fairly synchronized overall, thus implying many connections.

It is possible that a property of the data generated by Kuramoto oscillators makes it unsuitable for Granger causality computations. However, we used all provided tools for checking for problems. We suggest that further study is required to understand the conditions under which the results can be trusted and to provide practical ways to check those conditions. Unfortunately, of the myriad of uses made of GC in practice [62, 22, 5, 27, 174, 173, 26, 176], none of the authors validate the technique against a ground truth example. There may be another version of Granger causality that can correctly infer networks from our time series data. However this remains an open challenge to the community at large. Our code is available online at `github.com/BethanyL/gc` so that our experiments can be repeated with other network inference methods. In particular, we have shown that it is important to test methods on data that is not simply generated from a VAR model and that a range of networks should be considered. It may be possible that other statistical innovations for determining causality can be used to infer network structure [82, 121, 132], including new di-

rections leveraging independent component analysis [146], the phase slope index (PSI) [128], and/or network structure [134]. More recent innovations have considered the construction of local models of GC to infer the broader inference network [162] and finding time-delay embeddings in systems displaying attractor structures [152]. Regardless of technique, this is a fundamentally difficult problem [8] requiring new ideas, innovations and methods from the broader mathematical sciences community. Network science is here to stay and inference models will only increase in importance to the physical sciences community.

	n	A	K	θ^0	ω	N	Δt	T	m	s	prep	voting
A1	2	all four 2-node networks	$0.1, 0.2, \ldots, 10$	$[0, 2\pi]$	$[-1, 1]$	100	0.1	25	250	2.5	$\cos(\theta)$	none
A2	2	all four 2-node networks	$0.1, 0.2, \ldots, 10$	$[0, 2\pi]$	$[-1, 1]$	100	0.1	25	250	0.5	$\cos(\theta)$	none
B1	12	Fig. 4	5	*	**	1	0.1	25	250	0.1	$\cos(\theta)$	none
B2	12	Fig. 4	5	*	**	1	0.1	25	250	2.5	$\cos(\theta)$	none
B3	12	Fig. 4	5	*	**	1	0.05	25	500	2.5	$\cos(\theta)$	none
C1	12	E-R, $p = 0.05, 0.1, \ldots, 1$	$0.1, 0.2, \ldots, 10$	$[0, 2\pi]$	$[0, 2]$	100	0.1	25	250	2.5	$\cos(\theta)$	none
C2	6	E-R, $p = 0.05, 0.1, \ldots, 1$	$0.5, 1, 2, 4, 8$	$[0, 2\pi]$	$[0, 2]$	100	0.1	25	250	2.5	$\cos(\theta)$	none
C3	24	E-R, $p = 0.05, 0.1, \ldots, 1$	$0.5, 1, 2, 4, 8$	$[0, 2\pi]$	$[0, 2]$	100	0.1	25	250	2.5	$\cos(\theta)$	none
C4	12	E-R, $p = 0.05, 0.1, \ldots, 1$	$0.5, 1, 2, 4, 8$	$[-2\pi, 0]$	$[0, 2]$	100	0.1	25	250	2.5	$\cos(\theta)$	none
C5	12	E-R, $p = 0.05, 0.1, \ldots, 1$	$0.5, 1, 2, 4, 8$	$[-\pi, \pi]$	$[0, 2]$	100	0.1	25	250	2.5	$\cos(\theta)$	none
C6	12	E-R, $p = 0.05, 0.1, \ldots, 1$	$0.5, 1, 2, 4, 8$	$[0, 2\pi]$	$[-2, 0]$	100	0.1	25	250	2.5	$\cos(\theta)$	none
C7	12	E-R, $p = 0.05, 0.1, \ldots, 1$	$0.5, 1, 2, 4, 8$	$[0, 2\pi]$	$[-1, 1]$	100	0.1	25	250	2.5	$\cos(\theta)$	none
C8	12	E-R, $p = 0.05, 0.1, \ldots, 1$	$0.5, 1, 2, 4, 8$	$[0, 2\pi]$	$[0, 2]$	10	0.1	25	250	2.5	$\cos(\theta)$	none
C9	12	E-R, $p = 0.05, 0.1, \ldots, 1$	$0.5, 1, 2, 4, 8$	$[0, 2\pi]$	$[0, 2]$	1000	0.1	25	250	2.5	$\cos(\theta)$	none
C10	12	E-R, $p = 0.05, 0.1, \ldots, 1$	$0.5, 1, 2, 4, 8$	$[0, 2\pi]$	$[0, 2]$	100	0.05	5	25	2.5	$\cos(\theta)$	none
C11	12	E-R, $p = 0.05, 0.1, \ldots, 1$	$0.5, 1, 2, 4, 8$	$[0, 2\pi]$	$[0, 2]$	100	0.05	15	75	2.5	$\cos(\theta)$	none
C12	12	E-R, $p = 0.05, 0.1, \ldots, 1$	$0.5, 1, 2, 4, 8$	$[0, 2\pi]$	$[0, 2]$	100	0.05	25	125	2.5	$\cos(\theta)$	none
C13	12	E-R, $p = 0.05, 0.1, \ldots, 1$	$0.5, 1, 2, 4, 8$	$[0, 2\pi]$	$[0, 2]$	100	0.05	35	175	2.5	$\cos(\theta)$	none
C14	12	E-R, $p = 0.05, 0.1, \ldots, 1$	$0.5, 1, 2, 4, 8$	$[0, 2\pi]$	$[0, 2]$	100	0.1	5	50	2.5	$\cos(\theta)$	none
C15	12	E-R, $p = 0.05, 0.1, \ldots, 1$	$0.5, 1, 2, 4, 8$	$[0, 2\pi]$	$[0, 2]$	100	0.1	15	150	2.5	$\cos(\theta)$	none
C16	12	E-R, $p = 0.05, 0.1, \ldots, 1$	$0.5, 1, 2, 4, 8$	$[0, 2\pi]$	$[0, 2]$	100	0.1	35	350	2.5	$\cos(\theta)$	none
C17	12	E-R, $p = 0.05, 0.1, \ldots, 1$	$0.5, 1, 2, 4, 8$	$[0, 2\pi]$	$[0, 2]$	100	0.2	5	50	2.5	$\cos(\theta)$	none
C18	12	E-R, $p = 0.05, 0.1, \ldots, 1$	$0.5, 1, 2, 4, 8$	$[0, 2\pi]$	$[0, 2]$	100	0.2	15	300	2.5	$\cos(\theta)$	none
C19	12	E-R, $p = 0.05, 0.1, \ldots, 1$	$0.5, 1, 2, 4, 8$	$[0, 2\pi]$	$[0, 2]$	100	0.2	25	500	2.5	$\cos(\theta)$	none
C20	12	E-R, $p = 0.05, 0.1, \ldots, 1$	$0.5, 1, 2, 4, 8$	$[0, 2\pi]$	$[0, 2]$	100	0.2	35	700	2.5	$\cos(\theta)$	none
C21	12	E-R, $p = 0.05, 0.1, \ldots, 1$	$0.5, 1, 2, 4, 8$	$[0, 2\pi]$	$[0, 2]$	100	0.1	25	250	1.5	$\cos(\theta)$	none
C22	12	E-R, $p = 0.05, 0.1, \ldots, 1$	$0.5, 1, 2, 4, 8$	$[0, 2\pi]$	$[0, 2]$	100	0.1	25	250	3.5	$\cos(\theta)$	none
C23	12	E-R, $p = 0.05, 0.1, \ldots, 1$	$0.5, 1, 2, 4, 8$	$[0, 2\pi]$	$[0, 2]$	100	0.1	25	250	2.5	$\theta_{i+1} - \theta_i$	none
C24	12	E-R, $p = 0.05, 0.1, \ldots, 1$	$0.5, 1, 2, 4, 8$	$[0, 2\pi]$	$[0, 2]$	100	0.1	25	250	2.5	detrend	tenths in time
C25	12	E-R, $p = 0.05, 0.1, \ldots, 1$	$0.5, 1, 2, 4, 8$	$[0, 2\pi]$	$[0, 2]$	1	0.1	25	250	2.5	$\cos(\theta)$	1000 sets
C26	12	E-R, $p = 0.05, 0.1, \ldots, 1$	$0.5, 1, 2, 4, 8$	$[0, 2\pi]$	$[0, 2]$	10	0.1	25	250	2.5	$\cos(\theta)$	100 sets
C27	12	E-R, $p = 0.05, 0.1, \ldots, 1$	$0.5, 1, 2, 4, 8$	$[0, 2\pi]$	$[0, 2]$	100	0.1	25	250	2.5	$\cos(\theta)$	10 sets
C28	12	E-R, $p = 0.05, 0.1, \ldots, 1$	$0.5, 1, 2, 4, 8$	$[0, 2\pi]$	$[0, 2]$	100	0.1	25	250	2.5	$\cos(\theta)$	halves in time
C29	12	E-R, $p = 0.05, 0.1, \ldots, 1$	$0.5, 1, 2, 4, 8$	$[0, 2\pi]$	$[0, 2]$	100	0.1	25	250	2.5	$\cos(\theta)$	fourths in time
C30	12	E-R, $p = 0.05, 0.1, \ldots, 1$	$0.5, 1, 2, 4, 8$	$[0, 2\pi]$	$[0, 2]$	100	0.1	25	250	2.5	$\cos(\theta)$	eighths in time
D1	12	E-R, $p = 0.05, 0.1, \ldots, 1$	$0.5, 1, 2, 4, 8$	$[0, 2\pi]$	$[0, 2]$	100	0.1	25	250	2.5	$\cos(\theta)$	none
D2	12	E-R, $p = 0.05, 0.1, \ldots, 1$	$0.5, 1, 2, 4, 8$	$[0, 2\pi]$	$[0, 2]$	1	0.1	25	250	2.5	$\cos(\theta)$	50 sets

Figure 2.17: Summary of experiments. For Experiments C1–C30, gray boxes highlight any change from the usual parameters.

* $\theta^0 = [10, 11, 6, 9, 5, 3, 8, 4, 0, 2, 7, 1]\frac{2\pi}{11}$

** $\omega = [.6, .4, .65, .35, .55, .45, -.1, -.3, -.05, -.35, -.15, -.25]$

Chapter 3

SHAPE CONSTRAINED TENSOR DECOMPOSITIONS USING SPARSE REPRESENTATIONS IN OVER-COMPLETE LIBRARIES

This chapter is based on joint work with Eric C. Chi and J. Nathan Kutz.

We consider N-way data arrays and low-rank tensor factorizations where the time mode is coded as a sparse linear combination of temporal elements from an over-complete library. Our method, Shape Constrained Tensor Decomposition (SCTD) is based upon the CANDECOMP/PARAFAC (CP) decomposition which produces r-rank approximations of data tensors via outer products of vectors in each dimension of the data. By constraining the vector in the temporal dimension to known analytic forms which are selected from a large set of candidate functions, more readily interpretable decompositions are achieved and analytic time dependencies discovered. The SCTD method circumvents traditional *flattening* techniques where an N-way array is reshaped into a matrix in order to perform a singular value decomposition. A clear advantage of the SCTD algorithm is its ability to extract transient and intermittent phenomena which is often difficult for SVD-based methods. We motivate the SCTD method using several intuitively appealing results before applying it on a number of high-dimensional, real-world data sets in order to illustrate the efficiency of the algorithm in extracting interpretable spatio-temporal modes. With the rise of data-driven discovery methods, the decomposition proposed provides a viable technique for analyzing multitudes of data in a more comprehensible fashion.

3.1 Introduction

Matrix decompositions are critically enabling algorithms for scientific computing and data analysis applications across every field of the engineering, social, biological, and physical sciences. Of particular importance is the singular value decomposition (SVD), which provides a principled method for dimensionality reduction and computation of interpretable subspaces within which the data reside. So widespread is the usage of the algorithm, and minor modifications thereof, that it has generated a myriad of names across various communities, including Principal Component Analysis (PCA) [133], the Karhunen-Loève (KL) decomposition, Hotelling transform [78, 79], Empirical Orthogonal Functions (EOFs) [106] and Proper Orthogonal Decomposition (POD) [108, 75]. However, in order to use the SVD, data, which generally may be of N distinct dimensions, must be *flattened* into a matrix form, potentially compromising the statistical accuracy of the subspaces computed. Tensor decompositions are a generalization of the SVD concept to higher dimensions, allowing for N-way arrays ($N \geq 3$) of data to be decomposed into their constitutive, low-rank subspaces without flattening, which is especially advantageous for categorical data types. It is often the case that one of the dimensions considered in the tensor is the time variable. In this chapter, we develop a version of a tensor decomposition algorithm that restricts the time dynamics to analytically tractable solutions sparsely selected from a large, over-complete library of candidate functions. In so doing, we provide a more interpretable framework for the tensor modes in the decomposition process and analytic expressions for their associated time dynamics.

With the rise of data science and data-driven discovery, tensor decompositions are of increasing value and importance for characterizing underlying structure and dimensionality of data [95]. Indeed, finding low-rank structure in high-dimensional data is at the core of many machine learning architectures [18, 123]. In applications, one of the important dimensions of the data set is a time variable which measures how the other quantities of interest evolve over a prescribed time course. A tensor decomposition produces the low-rank time variable

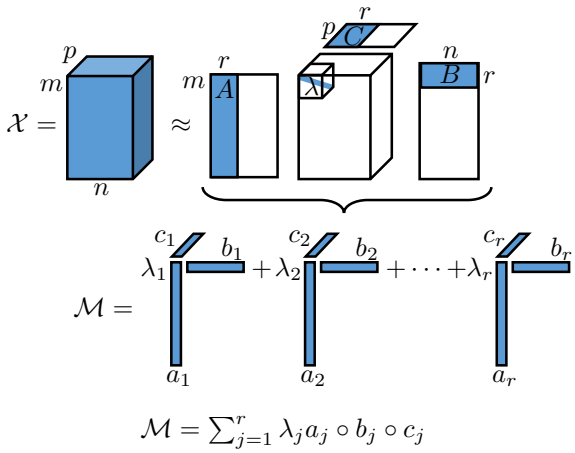

Figure 3.1: CP tensor decompositions. This type of decomposition approximates a data set $\mathcal{X}$ with a tensor $\mathcal{M}$ consisting of r components. Each component is an outer product of three vectors and is of the form $\lambda_j a_j \circ b_j \circ c_j$.

evolution. However, the low-rank time modes often are complicated and noisy due to the form of the data itself. In contrast, we often expect simple and highly structured temporal signatures, whether it be oscillations of a prescribed frequency or exponential growth/decay of a signal, for instance. The natural remedy is to constrain the form of the temporal modes extracted from the tensor decomposition. By specifying an over-complete library of temporal functions, we are able to extract analytic forms for the best fit temporal evolution of the data. The appropriate time behavior in our over-complete library is selected through sparse ℓ_1 regression techniques so as to select a minimal, but most informative, set of time dynamics. This is highly advantageous for characterizing the structure of the data and for data-driven discovery of underlying processes responsible for producing the dynamics observed.

Unlike the SVD for matrices, tensor decompositions are not unique, and there are a variety of decompositions that can be applied to N-way arrays ($N \geq 3$) of data. We consider the CANDECOMP/PARAFAC (CP) decomposition [25, 64] illustrated in Fig. 3.1, which arranges r-rank data into a series of outer products of N vectors. There are other

decompositions available, including the Tucker tensor decomposition [95] and the recently developed tensor-based method [94] for Dynamic Mode Decomposition (DMD) [160]. The former method is widely used in the tensor community while the latter method provides a regression that enforces Fourier mode behavior in the time mode. All three methods fall short of our primary goal, which is to provide a tensor decomposition with analytically tractable time dynamics capable of modeling transient phenomena. The DMD algorithm solves the first part of this objective but fails in modeling transient phenomena. Although a multi-resolution DMD method has been proposed to handle transients [99], it has a multiple pass architecture that sometimes struggles to extract spatio-temporal structures in a completely unsupervised manner. In this work, we demonstrate that the CP tensor decomposition can be modified to constrain the time dynamic mode to a broad range of analytic solutions that are selected from a large and over-complete library of candidate functions. By using sparse regression techniques, the best candidate functions are selected for representing the temporal dynamics. We call this technique Shape Constrained Tensor Decomposition (SCTD). The clear advantage of the SCTD over standard CP decompositions is that it gives analytic results which are readily interpretable. We demonstrate the method on a number of examples, including high-dimensional data generated from Houston crime data and global temperature measurements.

The rest of the chapter is organized as follows: Sec. II develops the basic mathematical architecture of the CP tensor decomposition and our refinement, the SCTD algorithm. This is followed by Sec. III in which we discuss practical details such as how to select tuning parameters and how to construct an appropriate over-complete library. Sec. IV tests the algorithm on simulated data, and Sec. V provides examples demonstrating the effectiveness of the algorithm on real-world data sets. Conclusions and an outlook for the SCTD algorithm are discussed in Sec. VI. For full details, all MATLAB and R codes used for this chapter are available online at `github.com/BethanyL/SCTD`.

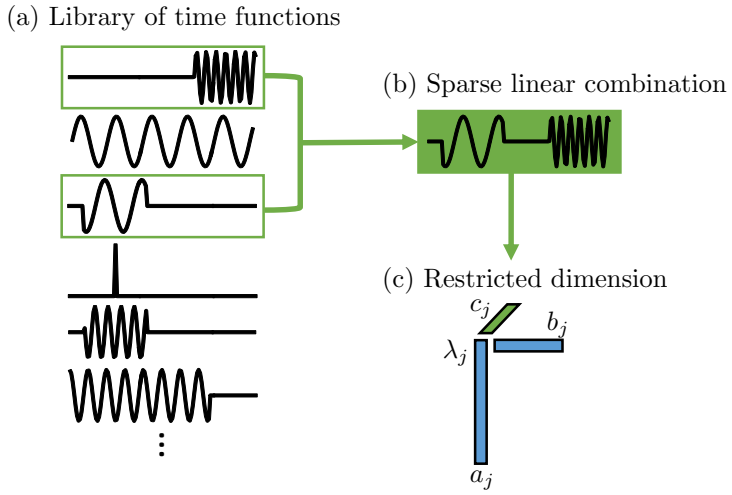

Figure 3.2: Sparse selection from an over-complete library. We restrict the third dimension of our CP tensor decomposition (the set of c_j vectors) to be a sparse linear combination of time dynamics functions from a library that we create. (a) We create a library with a variety of functions in time. (b) The algorithm then selects a small (sparse) subset of the library to linearly combine into a c_j vector fitting the time dynamics in the data. (c) This is a restriction on the third dimension of each component.

3.2 Methodology

In this manuscript, we present a number of modifications to the standard CP tensor decomposition that are intuitively appealing and improve interpretability of the low-rank modes extracted from data. Specifically, we introduce an over-complete library of temporal responses that constrains the time mode dynamics. A sparsity-promoting algorithm further selects a small number of these modes to represent the data. Thus the procedure can be thought of as a sparsity-promoting, constrained optimization problem.

The SCTD method is illustrated at a high level in Figs. 3.2 and 3.3. Fig. 3.2 shows the selection process whereby a small number of modes from an over-complete library of temporal functions are selected to best represent the temporal evolution of the data. We rely on an ℓ_1 optimization procedure so as to obtain a sparse representation of the temporal

dynamics. The algorithm thus restricts the temporal mode in the decomposition. While the temporal functions populating the library may be arbitrary, the power of the SCTD relies on the fact that temporal dynamics are typically far from arbitrary. Fig. 3.3 illustrates some temporal functions that serve as prototypes that characterize real-world temporal dynamics. Note that some of these functions are ideally suited for handling transient dynamics. Indeed, the success of the method is directly related to the temporal library functions included in the regression procedure.

A specific demonstration of the SCTD is shown in Fig. 3.4. In this example, three different spatial mode structures are combined with three specific time dynamics. The imposed time dynamics are representative of simple functional forms that are often difficult for the standard CP or DMD methods to model or resolve, i.e. temporal responses that have finite time windows of activity. In this example, the sequence of data snapshots are gathered into a 3-way data tensor $\boldsymbol{\mathcal{M}}$. Different snapshots of the dynamics depict the spatial structure arising from the combination of the different modal structures. The objective of the SCTD is to solve the inverse problem: Given the data tensor $\boldsymbol{\mathcal{M}}$, find the low-rank decomposition that correctly reconstructs the spatial modes and their time dynamics. The algorithm proposed here, which is based on the CP decomposition, can indeed recover the three modes and their time dynamics as shown in Fig. 3.4.

In the subsections that follow, the technical details of the CP tensor decomposition algorithm are considered along with strategies for building an over-complete library and enforcing a parsimonious combination of temporal dynamics prototypes.

3.2.1 CP Tensor Decompositions

We begin by first reviewing some useful notation. We denote the rth column of a matrix $\mathbf{A}$ by $\mathbf{a}_r$. Given matrices $\mathbf{A} \in \mathbb{R}^{I\times K}$ and $\mathbf{B} \in \mathbb{R}^{J\times K}$, their Khatri-Rao product is denoted by $\mathbf{A} \odot \mathbf{B}$ and is defined to be the $IJ \times K$ matrix of column-wise Kronecker products, namely

$$\mathbf{A} \odot \mathbf{B} \;=\; \Big(\mathbf{a}_1 \otimes \mathbf{b}_1 \quad \cdots \quad \mathbf{a}_K \otimes \mathbf{b}_K\Big).$$

For an N-way tensor $\boldsymbol{\mathcal{A}}$ of size $I_1 \times I_2 \times \cdots \times I_N$, we denote its $\mathbf{i} = (i_1, i_2, \ldots, i_N)$ entry by $a_{\mathbf{i}}$. The inner product between two N-way tensors $\boldsymbol{\mathcal{A}}$ and $\boldsymbol{\mathcal{B}}$ of compatible dimensions is given by

$$\langle \boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{B}} \rangle \;=\; \sum_{\mathbf{i}} a_{\mathbf{i}} b_{\mathbf{i}}.$$

The Frobenius norm of a tensor $\boldsymbol{\mathcal{A}}$, denoted by $\|\boldsymbol{\mathcal{A}}\|_{\mathrm{F}}$, is the square root of the inner product of $\boldsymbol{\mathcal{A}}$ with itself, namely $\|\boldsymbol{\mathcal{A}}\|_{\mathrm{F}} = \sqrt{\langle \boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{A}} \rangle}$. Finally, the mode-$n$ matricization or unfolding of a tensor $\boldsymbol{\mathcal{A}}$ is denoted by $\mathbf{A}_{(n)}$.

Let $\boldsymbol{\mathcal{M}}$ represent an N-way data tensor of size $I_1 \times I_2 \times \cdots \times I_N$. We are interested in an R-component CANDECOMP/PARAFAC (CP) [25, 64] factor model

$$\boldsymbol{\mathcal{M}} \;=\; \sum_{r=1}^{R} \lambda_r \, \mathbf{a}_r^{(1)} \circ \cdots \circ \mathbf{a}_r^{(N)}, \tag{3.1}$$

where $\circ$ represents outer product and $\mathbf{a}_r^{(n)}$ represents the rth column of the *factor matrix* $\mathbf{A}^{(n)}$ of size $I_n \times R$. We refer to each summand as a *component*. Assuming each factor matrix has been column-normalized to have unit Euclidean length, we refer to the λ_r's as *weights*. We will use the shorthand notation $\boldsymbol{\mathcal{M}} = [\![\boldsymbol{\lambda}; \mathbf{A}^{(1)}, \ldots, \mathbf{A}^{(N)}]\!]$, where $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_R)^{\mathsf{T}}$ [10]. A tensor that has a CP decomposition is sometimes referred to as a Kruskal tensor.

For the rest of this article, we consider a 3-way tensor where two modes index state variation and the third mode indexes time variation.

$$\boldsymbol{\mathcal{M}} \;=\; \sum_{r=1}^{R} \lambda_r \, \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r.$$

Let $\mathbf{A} \in \mathbb{R}^{I_1 \times R}$ and $\mathbf{B} \in \mathbb{R}^{I_2 \times R}$ denote the factor matrices corresponding to the two state modes and $\mathbf{C} \in \mathbb{R}^{I_3 \times R}$ denote the factor matrix corresponding to the time mode. This 3-way decomposition is illustrated in Fig. 3.1.

3.2.2 Sparse Representations in Over-Complete Libraries

We can further impose structure on the factors of the low-rank decomposition. For example, we could impose sparsity and smoothness on factors [168, 4]. Here we assume that the

time mode can be coded as a sparse linear combination from a known over-complete library $\mathbf{D} \in \mathbb{R}^{I_3 \times P}$, namely

$$\mathbf{C} \;=\; \mathbf{DZ},$$

where the elements of $\mathbf{Z} \in \mathbb{R}^{P \times R}$ are predominantly zero, i.e. $\text{nnz}(\mathbf{Z}) \ll PR$, and $I_3 \ll P$. This set up can be thought of as a sparse version of CANDELINC (canonical decomposition with linear constraints) [42]. Figs. 3.2 and 3.3 show both the constrained decomposition and some example library functions used.

We seek the CP model that maximizes a penalized correlation with the data tensor $\boldsymbol{\mathcal{X}}$

$$\hat{\boldsymbol{\mathcal{M}}} = \arg\max_{\boldsymbol{\mathcal{M}}} f(\boldsymbol{\mathcal{M}}) \equiv \langle \boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{M}} \rangle - \tau \|\mathbf{Z}\|_1 \tag{3.2}$$

such that

$$\begin{aligned} \boldsymbol{\mathcal{M}} &= [\![\boldsymbol{\lambda}; \mathbf{A}, \mathbf{B}, \mathbf{C}]\!], \\ \mathbf{C} &= \mathbf{DZ}, \\ \|\boldsymbol{\lambda}\|_2 &\leq 1, \\ \|\mathbf{a}_r\|_2, \|\mathbf{b}_r\|_2, \|\mathbf{z}_r\|_2 &\leq 1 \quad \text{for } r = 1, \ldots, R \quad . \end{aligned}$$

The matrix norm $\|\mathbf{Z}\|_1$ is the sum of all the absolute values of $\mathbf{Z}$ and not the induced matrix 1-norm. Thus, the non-negative parameter τ trades off the degree of correlation of the model to the data and the sparsity level in the loadings $\mathbf{Z}$. The inequality constraints are added to ensure that the feasible set of the optimization problem is compact. The Bolzano-Weierstrass theorem ensures that a solution to the problem exists. Without these constraints, the optimization problem is not well posed as there is no global maximum.

We pause to clarify the relationship between the above problem and the sparse coding or dictionary learning problem [129]. Note that we can rewrite the optimization problem in (3.2) as

$$\text{maximize } \left\langle \mathbf{X}_{(3)}, (\mathbf{B} \odot \mathbf{A})\boldsymbol{\Lambda}\mathbf{D}^{\mathsf{T}}\mathbf{Z}^{\mathsf{T}} \right\rangle - \tau \|\mathbf{Z}\|_1$$

such that

$$\begin{aligned} \|\boldsymbol{\lambda}\|_2 &\leq 1, \\ \|\mathbf{a}_r\|_2, \|\mathbf{b}_r\|_2, \|\mathbf{z}_r\|_2 &\leq 1 \quad \text{for } r = 1, \ldots, R. \end{aligned}$$

If we add the additional constraint $\|(\mathbf{B} \odot \mathbf{A})\boldsymbol{\Lambda}\mathbf{D}^\mathsf{T}\mathbf{Z}^\mathsf{T}\|_\mathrm{F} = c$ for some constant c, then maximizing the penalized correlation is equivalent to minimizing the penalized squared error

$$\frac{1}{2}\|\mathbf{X}_{(3)} - \mathbf{W}\mathbf{Z}^\mathsf{T}\|_\mathrm{F}^2 + \tau\|\mathbf{Z}\|_1,$$

where $\mathbf{W} = (\mathbf{B} \odot \mathbf{A})\boldsymbol{\Lambda}\mathbf{D}^\mathsf{T}$. Thus, we see that the optimization problem given in (3.2) is closely related to a special case of the sparse coding problem where we seek to learn sparse coefficients $\mathbf{Z}$ as well as a dictionary matrix $\mathbf{W}$ which must obey rather strong structural constraints.

3.2.3 Algorithm

We now describe an algorithm for computing our structured low-rank approximation $\boldsymbol{\mathcal{M}}$. Note that the CP constraint $\boldsymbol{\mathcal{M}} = [\![\boldsymbol{\lambda}; \mathbf{A}, \mathbf{B}, \mathbf{C}]\!]$ renders the optimization problem in (3.2) non-convex. Note, however, that if we fix all but one of the block variables $\mathbf{A}, \mathbf{B}, \mathbf{Z}$, or $\boldsymbol{\lambda}$, the optimization problem involves a straightforward concave optimization whose solutions can be written in closed form. Thus, we propose a block coordinate ascent (BCA) algorithm.

One complication of adopting a BCA algorithm is that the updates for $\mathbf{A}, \mathbf{B}$, and $\mathbf{Z}$ each separate into R identical optimization problems. To be explicit, consider the problem of updating $\mathbf{Z}$ when $\boldsymbol{\lambda}, \mathbf{A}$, and $\mathbf{B}$ are fixed.

$$\max \sum_{r=1}^{R} \left[\lambda_r \left\langle \boldsymbol{\mathcal{X}}, \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{D}\mathbf{z}_r \right\rangle - \tau\|\mathbf{z}_r\|_1\right] \tag{3.3}$$

such that

$$\|\mathbf{z}_r\|_2 \leq 1 \quad \text{for } r = 1, \ldots, R.$$

Consequently, solving for the entire factor matrix $\mathbf{Z}$ at once will yield identical columns, namely $\mathbf{z}_1 = \mathbf{z}_2 = \ldots = \mathbf{z}_R$. To deal with this degeneracy, we construct $\mathcal{M}$ via deflation. The idea is to find the most correlated rank-1 tensor and then subtract it from the data tensor. We then repeat the procedure on the residual tensor.

We are now ready to summarize at a high level the BCA algorithm. Suppose we have completed $r-1$ rounds so far and let $\mathcal{Y}_r$ denote the residual, namely $\mathcal{X} - \sum_{r'=1}^{r-1} \lambda_r \mathbf{a}_{r'} \circ \mathbf{b}_{r'} \circ \mathbf{D}\mathbf{z}_{r'}$. At the rth round, we solve the following optimization problem.

$$\hat{\mathcal{M}} = \arg\max_{\mathcal{M}} f(\mathcal{M}) \equiv \langle \mathcal{Y}_r, \mathcal{M} \rangle - \tau \|\mathbf{z}_r\|_1 \tag{3.4}$$

such that

$$\begin{aligned} \mathcal{M} &= \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r, \\ \mathbf{c}_r &= \mathbf{D}\mathbf{z}_r, \\ \|\mathbf{a}_r\|_2, \|\mathbf{b}_r\|_2, \|\mathbf{z}_r\|_2 &\leq 1. \end{aligned}$$

We again solve the above maximization problem with block coordinate ascent. Once BCA has converged, we determine λ_r by solving the problem:

$$\lambda_r = \arg\min_{\lambda} \|\mathcal{Y}_r - \lambda \mathcal{M}\|_2.$$

The solution to the above scalar optimization problem is given by

$$\lambda_r = \frac{\langle \mathcal{Y}_r, \mathcal{M} \rangle}{\|\mathcal{M}\|_{\mathrm{F}}^2}.$$

We now detail the updates for the factors $\mathbf{a}_r, \mathbf{b}_r$, and $\mathbf{z}_r$.

Updating $\mathbf{a}_r$:

$$\mathbf{a}_r^{(n+1)} = \underset{\|\mathbf{a}\|_2 \leq 1}{\arg\max} \langle \mathbf{u}_r, \mathbf{a} \rangle , \tag{3.5}$$

where $\mathbf{u}_r = \mathbf{Y}_{(1)}(\mathbf{c}_r^{(n)} \otimes \mathbf{b}_r^{(n)})$. The update simply requires normalizing $\mathbf{u}_r$.

$$\mathbf{a}_r^{(n+1)} = \frac{\mathbf{u}_r}{\|\mathbf{u}_r\|_2}.$$

Updating $\mathbf{b}_r$:

$$\mathbf{b}_r^{(n+1)} = \underset{\|\mathbf{b}\|_2 \leq 1}{\arg\max} \langle \mathbf{v}_r, \mathbf{b} \rangle , \tag{3.6}$$

where $\mathbf{v}_r = \mathbf{Y}_{(2)}(\mathbf{c}_r^{(n)} \otimes \mathbf{a}_r^{(n+1)})$. The update simply requires normalizing $\mathbf{v}_r$.

$$\mathbf{b}_r^{(n+1)} = \frac{\mathbf{v}_r}{\|\mathbf{v}_r\|_2}.$$

Updating $\mathbf{z}_r$:

$$\mathbf{z}_r^{(n+1)} = \underset{\|\mathbf{z}\|_2 \leq 1}{\arg\max} \langle \mathbf{f}_r, \mathbf{z} \rangle - \tau \|\mathbf{z}\|_1, \tag{3.7}$$

where $\mathbf{f}_r = \mathbf{D}^\mathsf{T}\mathbf{Y}_{(3)}(\mathbf{b}_r^{(n+1)} \otimes \mathbf{a}_r^{(n+1)})$.

The optimization problem posed in (3.7) is a modified lasso problem and has appeared in similar settings [4, 168, 32]. The update is given by

$$\begin{aligned}
\tilde{\mathbf{z}}_r &= \underset{\mathbf{z}}{\arg\min} \frac{1}{2}\|\mathbf{f}_r - \mathbf{z}\|_2^2 + \tau\|\mathbf{z}\|_1 \\
[\tilde{\mathbf{z}}_r]_i &= \operatorname{sign}([\mathbf{f}_r]_i) \max\{|[\mathbf{f}_r]_i| - \tau, 0\} \\
\mathbf{z}_r^{(n+1)} &= \begin{cases} \frac{\tilde{\mathbf{z}}_r}{\|\tilde{\mathbf{z}}_r\|_2} & \text{if } \|\tilde{\mathbf{z}}_r\|_2 \neq 0 \\ \mathbf{0} & \text{otherwise.} \end{cases}
\end{aligned}$$

The derivation of this update rule is given in the Appendix. We then update λ_r, followed by calculating the next residual $\mathcal{Y}_{r+1} = \mathcal{Y}_r - \lambda_r \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{D}\mathbf{z}_r$.

3.3 Details of SCTD Algorithm

Now that we are familiar with the basic procedure, we next discuss important details in the SCTD. We first describe how the sparsity-inducing parameter τ is chosen. We then describe how the over-complete library is constructed.

3.3.1 Picking The Regularization Parameter

At each iteration, we use the Bayesian Information Criterion (BIC) [143] to pick regularization parameter τ_r. The BIC is a quantitative score that balances how well the model fits the data against how complicated the model is. In the context of the SCTD, a constrained rank-1 Kruskal tensor with low BIC corresponds to a rank-1 Kruskal tensor which fits the data well in light of how many free parameters were used in fitting it. As defined in [4], for this problem, the BIC criterion is

$$\text{BIC}(\tau_r) = \log\left[\frac{\|\mathcal{Y}_r - \lambda_r \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{D}\mathbf{z}_r\|_F^2}{I_1 I_2 I_3}\right] + \frac{\log(I_1 I_2 I_3)}{I_1 I_2 I_3}|\{\mathbf{z}_r\}|,$$

where $|\{\mathbf{z}_r\}|$ is the number of non-zero elements of $\mathbf{z}_r$. This can be derived from each update being an ℓ_1-norm penalized regularization problem.

We use the BIC criterion to pick the best τ_r from a range of options. We further refine the value of τ_r by checking the neighborhood of the current best option until the neighborhood is sufficiently small or the BIC curve is sufficiently constant on that neighborhood. An upper limit of τ_r is the point at which all entries of $\mathbf{z_r}$ are zero. Since we accumulate small amounts of error on each iteration, we wish to encourage increasing levels of sparsity as r increases. We thus use τ_{r-1} as a lower bound for τ_r, unless τ_{r-1} is greater than the current upper bound.

3.3.2 Constructing the Over-Complete Library

We can choose an over-complete library based on knowledge of the application area or data set. Some natural candidates are displayed in Fig. 3.3. If we expect periodic but transient dynamics, we may choose to populate the library with windowed sines and cosines, varying

the frequencies of the sines and cosines and the widths and shifts of the windows. If we anticipate transient phenomena that are non-periodic, it may be appropriate to include Gaussians with a range of means and variances. If the time domain itself is periodic, such as hour of the day, then we might improve the results by including dynamics that have this period. For example, to allow a Gaussian-like mode to vary smoothly through the night, we could generate cosines with varying frequencies and shifts, but only include one period of the cosine (see "wrapped cosines" in Fig. 3.3).

3.4 Simulation Experiments

We begin by testing the SCTD on a simulated data set similar to Fig. 3.4. Recall that this data set is composed of three spatio-temporal modes (specifically, it is a Kruskal tensor with rank three). We can think of this data set as a video or sequence of frames. Our goal is to decompose it into three modes (a rank-three Kruskal tensor) with an analytical description for the temporal dimension. Although the SCTD is exceptional for the data in Fig. 3.4, the example is limited since no noise was included in the data.

We next consider a more realistic example shown in Fig. 3.5. In this experiment, we added white Gaussian noise with standard deviation σ in the frequency domain to the data ($u_n(t) = \mathcal{F}^{-1}[\hat{u}(\omega) + \sigma\mathcal{N}(0,1)]$). In this case, we used $\sigma = 3$, resulting in a signal-to-noise ratio of 0.1374, where signal-to-noise ratio is defined as the ratio of the summed squared magnitude of the signal to the summed squared magnitude of the noise. The algorithm outlined in Sec. II can now be applied to the data and a direct comparison can be made to a CP decomposition and a DMD reduction. In particular, for the CP decomposition, we use the CP_ALS function in the Matlab Tensor Toolbox [11], [9], which uses an alternating least squares algorithm. Fig. 3.5 shows that despite the inclusion of noise, the modes and temporal dynamics can be cleanly extracted using the SCTD. Indeed, analytic forms for the time dynamics can be discovered. In comparison, the CP algorithm gives a decomposition with noisy time modes which lack analytic description. The DMD algorithm (using data flattening) can give analytic expressions for the time dynamics, but the temporal expressions

are significantly flawed due to the fact that DMD cannot handle such transient and/or intermittent time dynamics, i.e. only time dynamics of the form $\exp(\omega t)$ are allowed.

The SCTD also provides a diagnostic for performing an r-rank truncation. For an SVD decomposition, the singular values provide the requisite metric for truncation. Similarly, Fig. 3.6 shows the decay of reconstruction error as a function of the number of tensor modes. We can choose the rank, or number of components to keep in the SCTD, by considering the trade-off between error and complexity. Here we see diminishing returns in reconstruction error after the inclusion of the first three components, suggesting that a rank-3 approximation sufficiently captures the majority of the systematic variation in the data.

To further explore the example shown in Fig. 3.5, we consider a number of different cases which highlight the use of the algorithm and the choice of library prototypes. Thus we consider the following:

- *Case (a): Library contains true modes.* We start with an easy case. We construct a library with 3000 prototypes, including the true temporal modes, and we do not add noise to the data. We see in Fig. 3.7 and Table 3.1 that in two iterations, the SCTD picks exactly one prototype, and in one iteration, the SCTD picks 299 from the 3000 and accumulates a small amount of error.

- *Case (b): Library does not contain true modes.* Next, we want to assess how robust the SCTD is to "model misspecification": We construct another library of 3000 prototypes but do not "cheat" by including the true time dynamics. As we can see in Fig. 3.7 and Tab. 3.1, in this experiment, the method uses extra prototypes (about 10% of the library) and accumulates more error. However, the factor accuracy only reduces from 0.989 to 0.948.

- *Case (c): Library does not contain true modes and the data is noisy.* Finally, we increase the difficulty by adding white Gaussian noise to the data ($\sigma = 1$). The results are very similar to Case (b) without noise (see Fig. 3.7 and Tab. 3.1). Note that

Table 3.1: Details to accompany Fig. 3.7

case	# prototypes chosen	relative error	factor accuracy	frequency of top mode
(a)	301 (3.6%)	.1144	.988924	.0982 .3927 .7854
(b)	1132 (13.5%)	.2329	.948130	.1026 .3846 .7692
(c)	857 (10.2%)	.6947	.937874	.1026 .3846 .7692

although the resulting analytic expression of hundreds of modes is not simple, if you want a simple analytical expression, you can pick the mode with the largest coefficient and still maintain accuracy. The top mode is plotted in green on top of the linear combination (blue) and the true mode (black) in Fig. 3.7.

Next, we consider Case (c) in more detail. So far, we have seen two examples of this case. In Fig. 3.5, the library contains 40,000 prototypes and the white noise has standard deviation $\sigma = 3$. In Fig. 3.7 (and Tab. 3.1), the library contains 3,000 prototypes and the white noise has standard deviation $\sigma = 1$. We now consider the effect of σ (Fig. 3.8) and the effect of the library size (Fig. 3.9).

In Fig. 3.8, we fix the library size to 50,000 and vary σ. As the magnitude of the noise increases, so does the error. However, this growth in error is slow when the error is measured against the original (noiseless) data. For context, see Fig. 3.5 for a visualization of data with

$\sigma = 3$.

In Fig. 3.9, we consider data that has no noise and vary the library size. Once the library is sufficiently large, the relative error does not improve. However, the number of prototypes chosen grows.

3.5 Real Data Examples

We now apply the SCTD to two real-world data sets exhibiting complex spatio-temporal dynamics with intermittency. These examples illustrate the power of the SCTD to produce interpretable results, especially in the constrained time dynamics. Figures were rendered with the ggmap and ggplot2 R packages [90, 166].

3.5.1 Houston Crime

Data mining is beginning to be applied to a myriad of law enforcement problems [117]. These techniques can be used to help agencies deploy their employees more efficiently, predict the outcomes of new initiatives, and identify trends in crime in order to take preventative measures.

We apply the SCTD to a data set, collected by the Houston Police Department, with 85,622 crimes occurring in Houston from January to August 2010. We use a preprocessed version of the data included in the ggmap R package [90]. We create a 3-way tensor of counts of these crimes. The dimensions are type of crime (aggravated assault, auto theft, burglary, robbery, or theft), crime beat (118 options), and hour of day (0–23). We then apply the SCTD to this data set using a mix of the three types of functions displayed in Fig. 3.3—windowed sines and cosines, Gaussians, and wrapped cosines.

The first three components of the SCTD are displayed in Fig. 3.10. In the first component, most beats are at least lightly included, although some are more intense. Theft in the evening is especially emphasized. The second component adds non-theft crimes to a different set of hot spots and subtracts non-theft crime from some of the beats that were important in the first component. The third component re-emphasizes some of the same beats, this

time adding burglary and subtracting theft in the morning and conversely adding theft and subtracting burglary in the evening.

In Fig. 3.11, we compare our results to the first three components from CP_APR (the non-negative CP tensor decomposition with alternating Poisson regression [33] as implemented in the Matlab Tensor Toolbox [11]). The results using the SCTD are much smoother and more interpretable in the time dimension, but many of the same beats are considered important.

3.5.2 El Niño

Sensor and imaging technologies (oceanic, terrestial and satelite) have led to a significant increases in climate data and a limited but growing understanding of how to extract meaningful information from it. Interest in this interdisciplinary field has spawned, for example, the annual International Workshop on Climate Informatics [100].

We demonstrate the SCTD on a data set of sea surface temperatures. The data are freely available from the NOAA/OAR/ESRL PSD, Boulder, Colorado, USA. We used the weekly sea surface temperature from the NOAA_OI_SST_V2 data set, which can be downloaded from `http://www.esrl.noaa.gov/psd/`. In particular, we consider the Pacific Ocean from 1995 through the end of 2000. We subtracted the background from the data using DMD [59], and then we created a library with a combination of Gaussians and windowed sines and cosines. Two of the first twelve modes that the SCTD extracted are shown in Fig. 3.12. The second component finds one-time phenomena related to the El Niño event of 1997–1998. In particular, we see unusually warm temperatures in the eastern Pacific ocean, especially near Peru, but almost stretching to New Guinea. By mid-to-late 1997, unusually cool waters occurred near the coast of Australia. The third component finds annual variation in temperature, split over the equator.

These results could not be obtained with standard DMD because the El Niño event is not a Fourier mode. However, recent innovations around multi-resolution analysis and DMD (the multi-resolution DMD algorithm [99]) does allow for a significantly improved description. Likewise, a traditional CP tensor decomposition might extract similar patterns, they would

not be accompanied with a sparse analytic description. We show that the SCTD choses a sparse linear combination of our over-complete library in Fig. 3.13.

3.6 Conclusion

Data-driven discovery has become ubiquitous across the sciences, leading to the rise of the fourth paradigm of scientific discovery [71]. Critical in meeting the challenges of this emerging paradigm is the development of algorithms that are capable of extracting meaningful and interpretable low-dimensional features from data that is high-dimensional and includes many distinct dimensions. The success of machine learning is largely due to its ability to represent data in low-dimensional feature spaces where data can be more effectively analyzed, classified and clustered. Matrix decomposition techniques, which project to low-rank subspaces via some underlying optimization algorithm, are the workhorses of the data science industry. For instance, Principal Component Analysis is now standard across almost every field of the engineering, social, biological and physical sciences. This SVD-based method provides a least-square fitting algorithm for data, thus providing low-rank subspaces that best represent the features of the data.

The success of the SVD is difficult to overestimate. It is simply the most dominant and successful matrix decomposition method being used today. The SVD requires, however, that multi-dimensional data first be *flattened* before being processed through the decomposition. This can lead to less parsimonious fitting than if the data was preserved in its original N-way data tensor. Tensor decompositions, on the other hand, allow the data to be preserved in its original multi-dimensional context, which is especially advantageous for categorical data. Although tensors have been the subject of active research for the past four decades, it has been difficult for tensor decompositions to displace standard SVD with flattening decompositions. This is in part due to the multitude of potential tensor decompositions available to the practitioner, i.e. it is not unique. Moreover, the SVD has numerous enhancements for handling high-dimensional data, such as the randomized SVD [61, 46], which enables efficient computation of the matrix decomposition even with extraordinarily large data.

In this manuscript, we have developed what we think is a highly useful innovation to the standard CP tensor decomposition. By constraining the time dimension of the tensor decomposition, a more intuitively appealing and interpretable decomposition can be achieved. Indeed, analytic solution forms for the time dependency of the data decomposition can be extracted. This is done by using an over-complete library of potential temporal functions in order to select the best candidate functions via sparse regression. This work merges three distinct mathematical methods: tensor decompositions, sparse regression, and over-complete libraries. The success of the SCTD method is demonstrated on a number of simulated problems and two real-world applications where preserving the tensor nature of the data is highly desirable and advantageous. The SCTD method provides a viable data-discovery algorithm that can be used in a host of settings where low-rank features of an N-way data tensor need to be analyzed. It should also be noted that one can easily envision also constraining other dimensions of the data, not just the time dimension.

Ultimately, the most useful data analysis techniques developed allow for interpretable diagnostics which are also predictive in nature. The SCTD advances a theoretical framework for tensor decompositions that provides an intuitively appealing framework for understanding the rich time dynamics of low-rank decompositions without requiring data-flattening. With the emergence of many categorical data structures, this can be especially appealing. Thus, we render a tensor decomposition package that is user-friendly and aids in identifying important dynamics structures in data, including intermittent phenomena, which are very difficult for standard tensor, DMD and PCA-like methods to deduce.

Appendix

Notation Details

Matricization of a tensor: The mode-n matricization or unfolding of a tensor $\boldsymbol{\mathcal{A}}$ is denoted by $\mathbf{A}_{(n)}$ and is of size $I_n \times J_n$ where $J_n \equiv \prod_{m \neq n} I_m$. In this case, the tensor element with

index $\mathbf{i}$ maps to matrix element (i, j) where

$$i = i_n \quad \text{and} \quad j = 1 + \sum_{\substack{k=1 \\ k \neq n}}^{N} (i_k - 1) \left(\prod_{\substack{m=1 \\ m \neq n}}^{k-1} I_m \right).$$

Derivation of $\mathbf{z}$ *update:*

We prove that the update for $\mathbf{z}$ is the solution to the optimization problem given in (3.7).

Proof. The negative of the Lagrangian for (3.7) (to express the optimization as a minimization) is given by

$$\mathcal{L}(\mathbf{z}, \gamma) \;=\; -\langle \mathbf{f}, \mathbf{z} \rangle + \tau \|\mathbf{z}\|_1 + \gamma(\|\mathbf{z}\|_2^2 - 1).$$

The KKT conditions are given by

$$\begin{aligned} \mathbf{f} - 2\gamma\mathbf{z} &\in \tau \partial \|\mathbf{z}\|_1 \\ \|\mathbf{z}\|_2^2 &\leq 1 \\ \gamma &\geq 0 \\ \gamma(\|\mathbf{z}\|_2^2 - 1) &= 0. \end{aligned}$$

There are two cases to consider. If $\tilde{\mathbf{z}} = \mathbf{0}$, then the pair $(\mathbf{z}, \gamma) = (\mathbf{0}, 0)$ satisfies the KKT conditions. The stationarity condition is satisfied since $\mathbf{f} \in \tau \partial \|\mathbf{0}\|_1$ since $\tilde{\mathbf{z}} = \mathbf{0}$ solves the lasso problem. The other conditions are easily verified.

If $\tilde{\mathbf{z}} \neq \mathbf{0}$, then the pair $(\mathbf{z}, \gamma) = (\tilde{\mathbf{z}}/\|\tilde{\mathbf{z}}\|_2, 0)$ satisfies the KKT conditions. The only tricky condition to verify is the stationarity condition. The other conditions are easy to verify. Since $\tilde{\mathbf{z}}$ is the solution to the lasso problem we have that

$$\begin{aligned} \mathbf{f} - \tilde{\mathbf{z}} &\in \tau \partial \|\tilde{\mathbf{z}}\|_1 \\ \mathbf{f} - 2\left(\frac{\|\tilde{\mathbf{z}}\|_2}{2}\right) \frac{\tilde{\mathbf{z}}}{\|\tilde{\mathbf{z}}\|_2} &\in \tau \partial \left\| \|\tilde{\mathbf{z}}\| \frac{\tilde{\mathbf{z}}}{\|\tilde{\mathbf{z}}\|_2} \right\|_1. \end{aligned}$$

Note that we have used the fact that $\partial \|\mathbf{z}\|_1 = \partial \|c\mathbf{z}\|_1$ for all $c > 0$. Therefore, the pair $(\mathbf{z}, \gamma) = (\tilde{\mathbf{z}}/\|\tilde{\mathbf{z}}\|_2, \|\tilde{\mathbf{z}}\|_2/2)$ satisfies the KKT conditions. □

Acknowledgment

J. N. Kutz would like to acknowledge support from the Air Force Office of Scientific Research (FA9550-15-1-0385).

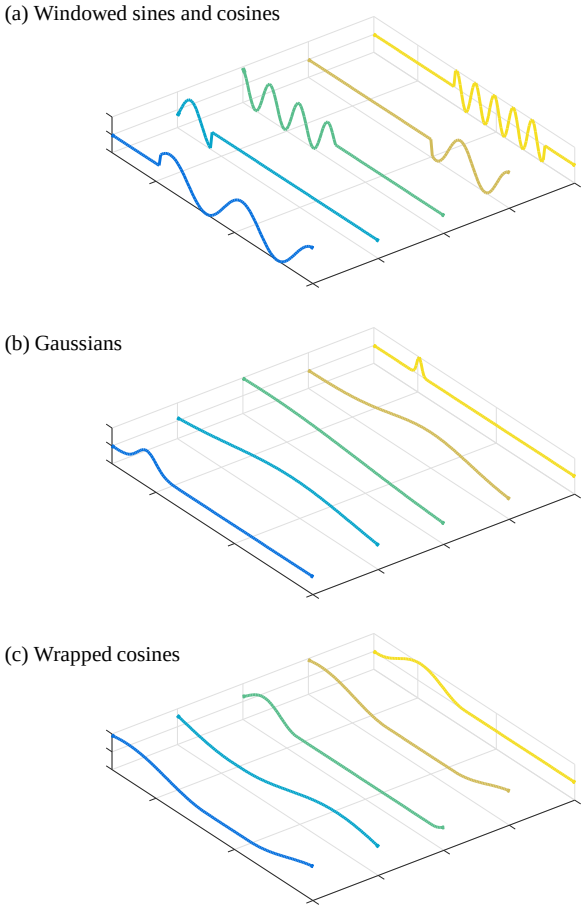

Figure 3.3: Constructing a library. Based on the application, we choose a library of possible time dynamics functions. Options include: (a) Windowed Sines and Cosines. We generate a range of sines and cosines, varying the frequency, width of the window, and center of the window. (b) Gaussians. We fill the library with Gaussian functions, varying the μ and σ parameters. (c) Wrapped Cosines. One way to generate a library that is Gaussian-like but has a period that is the length of the interval is to use one period of a shifted cosine. The frequency and shift can be varied.

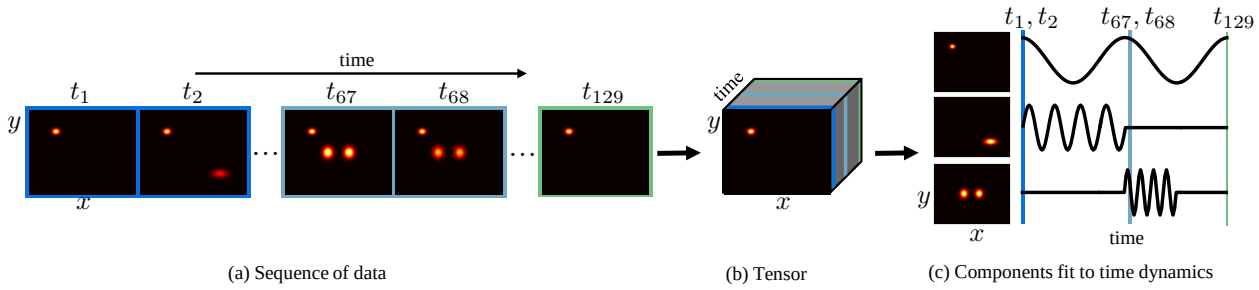

Figure 3.4: Extracting patterns from spatio-temporal data. (a) We begin with a data set where spatial information is collected over time. If we collect two-dimensional data at each time step, we may informally think of the data as a sequence of "frames." (b) The sequence of frames can be saved as a tensor (one data cube) where the third dimension is time. (c) Our goal is to decompose that tensor into a sum of important frame components where each frame component has its own time dynamics. In this example, we see the three components coming in and out of the frames as time passes. The color coding demonstrates how the sample frames in part (a) are combinations of the components shown in part (c).

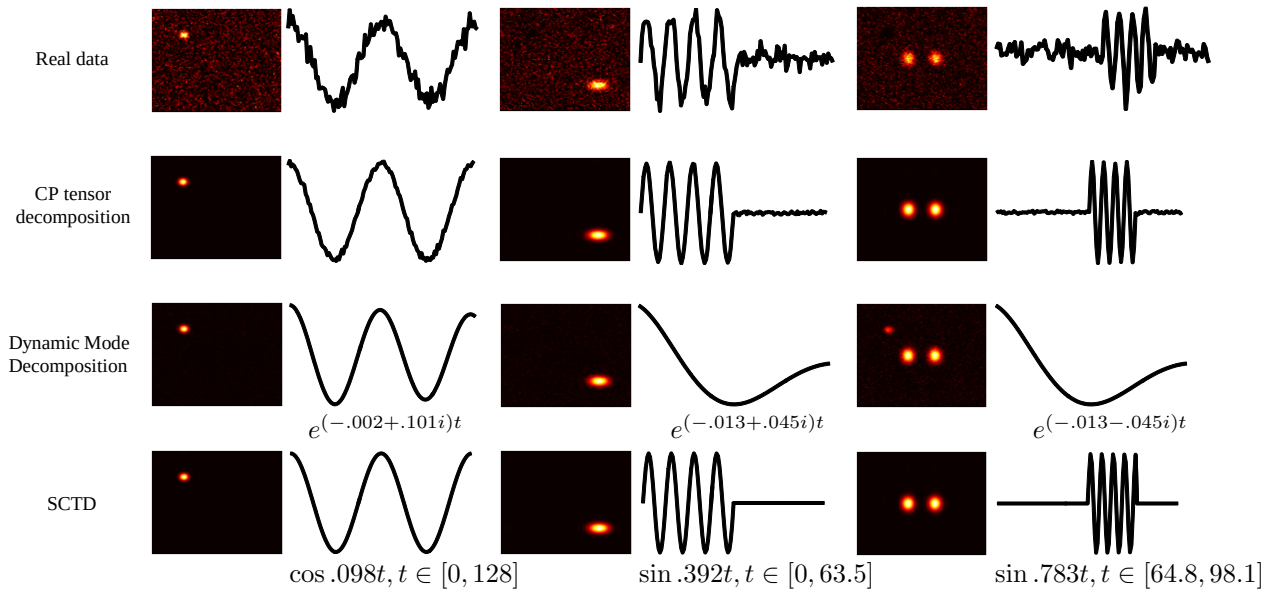

Figure 3.5: Comparing methods on a simulated data set. The data set, a 3-way tensor, is generated as described in Fig. 3.4, except that noise is added. We hope that a method can decompose the tensor into its three noiseless components. A traditional CP tensor decomposition sometimes falls into a good local minimum and decomposes the data correctly. Clean spatial modes are found, but some noise in the time dynamics is maintained. The time dynamics are not fit to analytic expressions. The Dynamic Mode Decomposition tries to fit clean time dynamics functions to the spatial modes. However, it is restricted to Fourier modes and cannot handle the windowed behavior in this data set. It also does not correctly separate the third spatial mode. The SCTD finds clean spatial modes and fits smooth time dynamics to each component. The output includes the exact functions that were fit to the time dynamics.

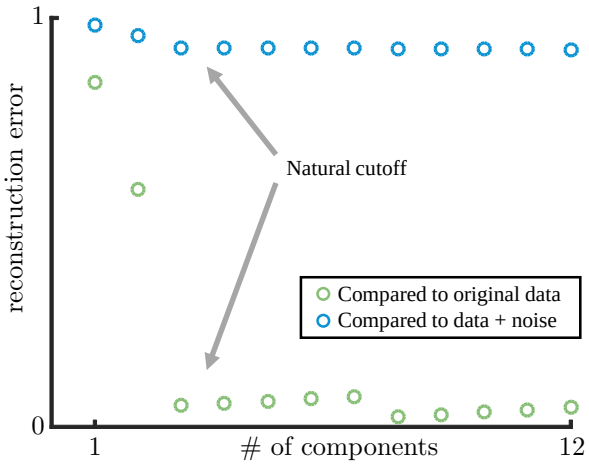

Figure 3.6: Reconstruction error curve. We can choose the number of components to keep in the SCTD by considering the trade-off between error and complexity. Here we see diminishing returns in reconstruction error after the inclusion of the first three components, suggesting that a rank-3 approximation sufficiently captures the majority of systematic variation in the data. We calculate the error in two ways—by comparing the reconstruction to the original clean data and to the noisy data.

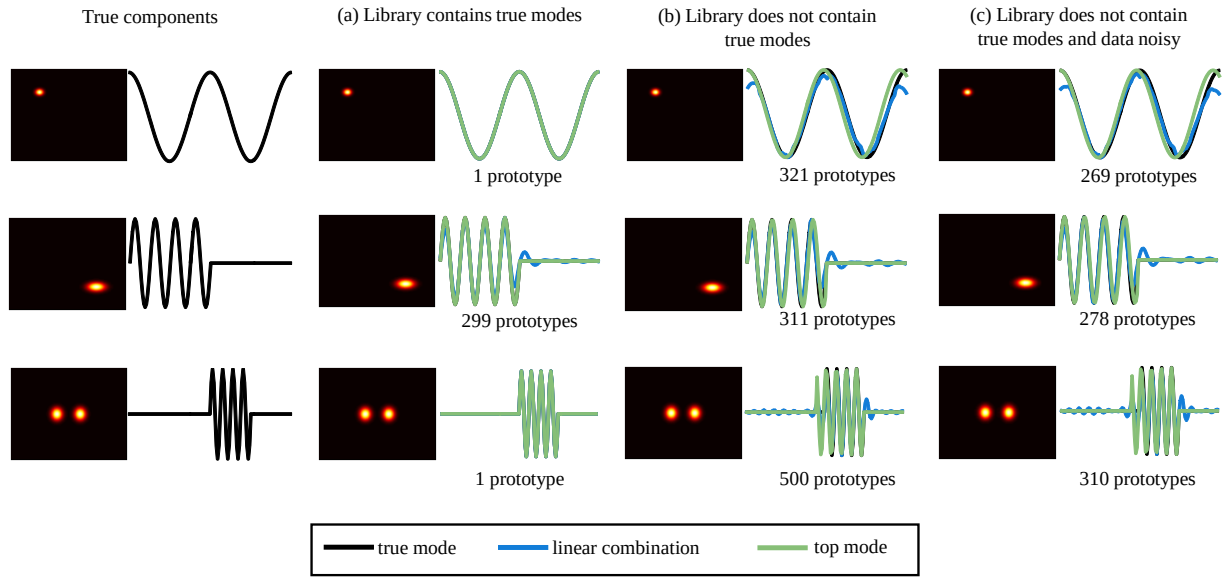

Figure 3.7: Results on simulated data set. We repeat for reference the three true components that compose the data set. (a) When the library contains the correct time dynamics functions, the SCTD does a good job of recovering them. (b) When the library does not contain the exact right modes, the SCTD uses more prototypes to fit the data, but still chooses a sparse number. (c) When we additionally make the data noisy, the SCTD is robust. It chooses more prototypes, but if an especially simple output is desired, using just the prototype with the highest coefficient is accurate. See more detail in Tab. 3.1.

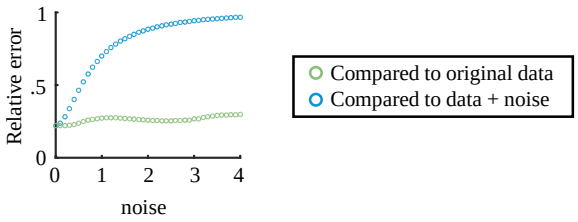

Figure 3.8: Varying the noise. In Figs. 3.5 and 3.7 and Tab. 3.1, we displayed results on noisy data. Here we vary the amount of noise to display the robustness of the SCTD. The value of σ ranges 0.1–4 while the SNR ranges 123.8–0.101. As the noise increases, the error in the reconstruction of the original data increases. Note that the cases of $\sigma = 3$ and $\sigma = 1$ are displayed in Figs. 3.5 and 3.7, respectively. The increase in error is slow when the error is in terms of the noiseless data.

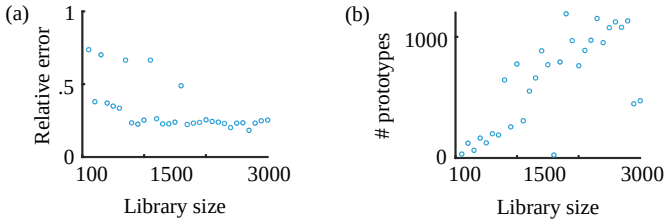

Figure 3.9: Varying the library size. In Fig. 3.7 and Tab. 3.1, we displayed results on a library with 3,000 prototypes. Here we vary the size of the library to consider the tradeoffs. Once we have a reasonably large library, the relative error is consistent. However, the number of selected prototypes roughly grows with the library size. Thus to limit complexity, we may wish to pick a library size that is sufficient for low error reconstructions but is not larger than necessary.

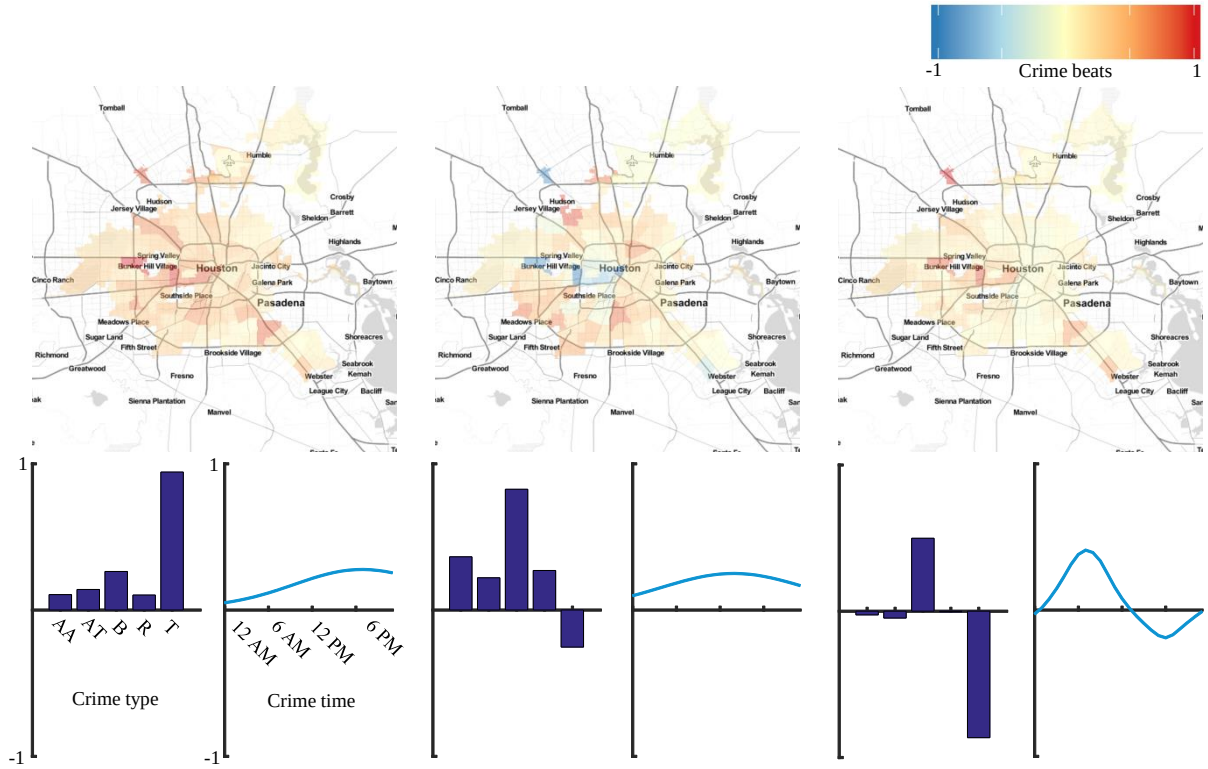

Figure 3.10: Results on Houston crime data set using SCTD. We start with a data set of Houston crime where the first dimension is type of crime, the second is crime beat, and the third is hour of the day (0–23). The five crimes considered are aggravated assault (AA), auto theft (AT), burglary (B), robbery (R), and theft (T). We decompose the data set with the SCTD and display the first three modes here. Our method finds sets of beats behaving similarly and assigns smooth, interpretable time dynamics.

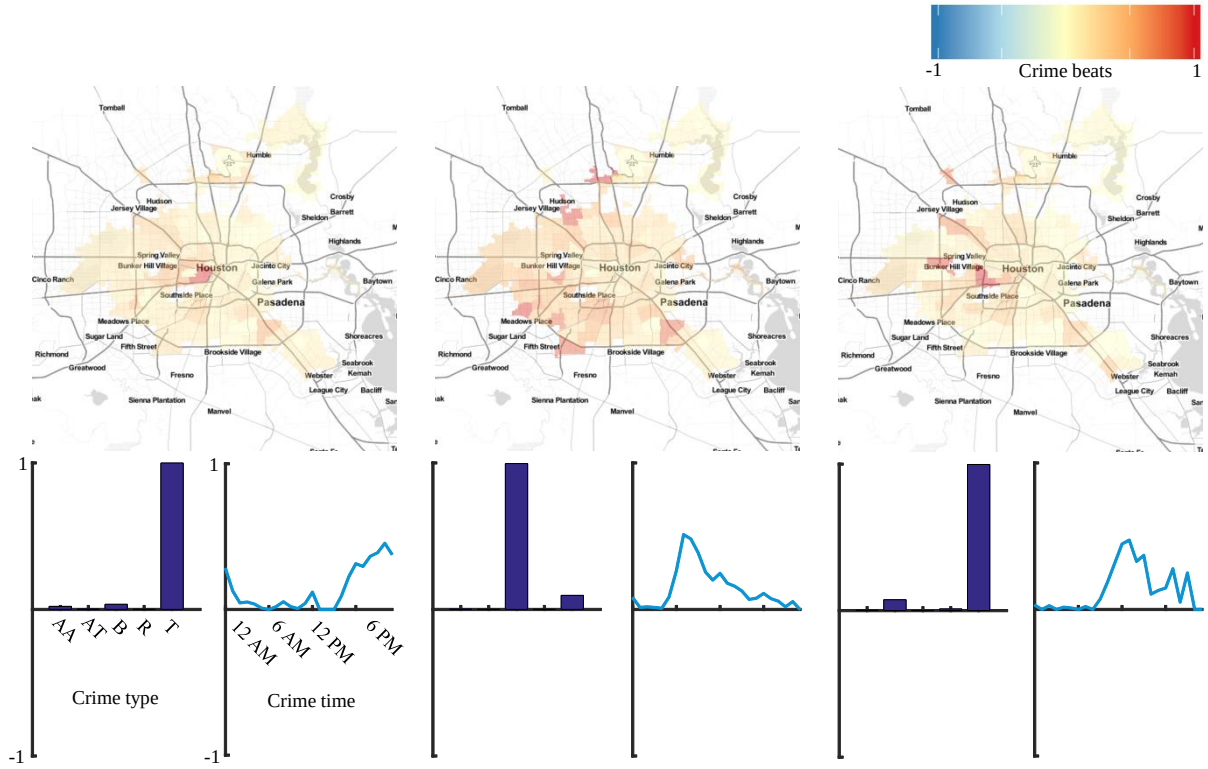

Figure 3.11: Results on Houston crime data set using CP_APR. We decompose the Houston crime data set again, but this time with the with the CP_APR tensor decomposition for comparison. We display the first three modes here. Note that the time dynamics are noisy.

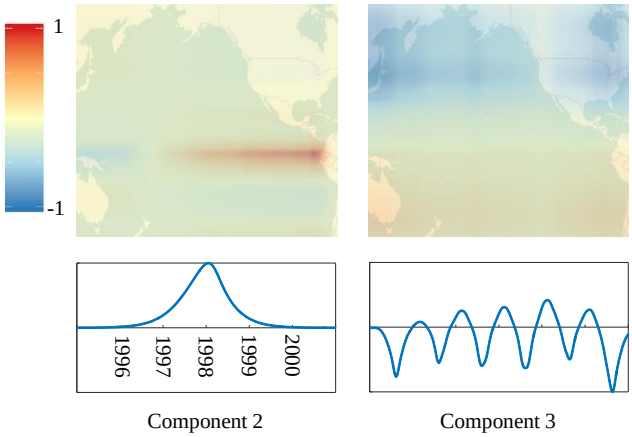

Figure 3.12: Results on ocean surface temperature data set. We start with a data set of ocean surface temperature over time. The dimensions are longitude, latitude, and time. Here we display a sample of the components found by the SCTD. The second component finds the El Niño event of 1997–1998, a warm band in the central and east-central equatorial Pacific. The third contain annual variation, split over the equator.

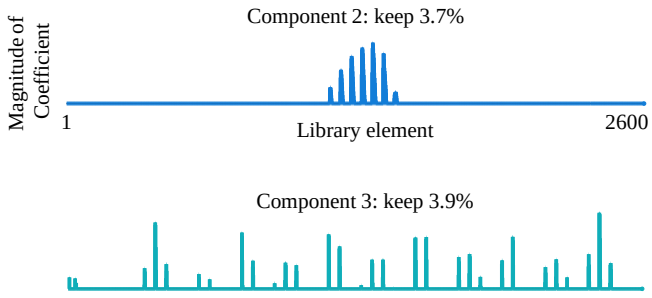

Figure 3.13: Results on ocean surface temperature data set, continued. This figure gives further information about the results in Fig. 3.12. We demonstrate the sparsity of the time dynamics by plotting the magnitudes of the coefficients in each $\mathbf{z}_r$.

Chapter 4

MODELING COGNITIVE DEFICITS FOLLOWING NEURODEGENERATIVE DISEASES AND TRAUMATIC BRAIN INJURIES WITH DEEP CONVOLUTIONAL NEURAL NETWORKS

Neurodegenerative disease and traumatic brain injuries (TBI) present grave challenges to doctors. Both cause cognitive deficits due to focal axonal swellings (FAS), but it is difficult to deliver a prognosis due to our limited ability to diagnose damage at a cellular level in vivo. In this chapter, we simulate neurodegenerative disease and TBI. We use convolutional neural networks (CNNs) as our model of cognition, since they were originally inspired by neuroscience, and they mimic important features of brains. We start with CNNs pre-trained to perform classification, then utilize data on FAS to damage the connections in a biophysically relevant way. In order to improve the model, we incorporate the idea that brains operate under energy constraints by pruning the CNNs to be less over-engineered. Qualitatively, we demonstrate that damage to the connections leads to human-like mistakes. Our experiments also provide quantitative assessments of how accuracy is affected by various types and levels of damage. The deficit resulting from a fixed amount of damage greatly depends on which connections are randomly injured, providing intuition for why it is difficult to predict the impairments resulting from an injury. There is a large degree of subjectivity when it comes to interpreting cognitive deficits from dynamically evolving complex systems such as the human brain. However, we provide important insight and a quantitative framework for several disorders in which FAS are implicated, such as TBI, Alzheimer's, Parkinson's, and Multiple Sclerosis.

This chapter is based on joint work with Jake Weholt, Pedro D. Maia, and J. Nathan Kutz.

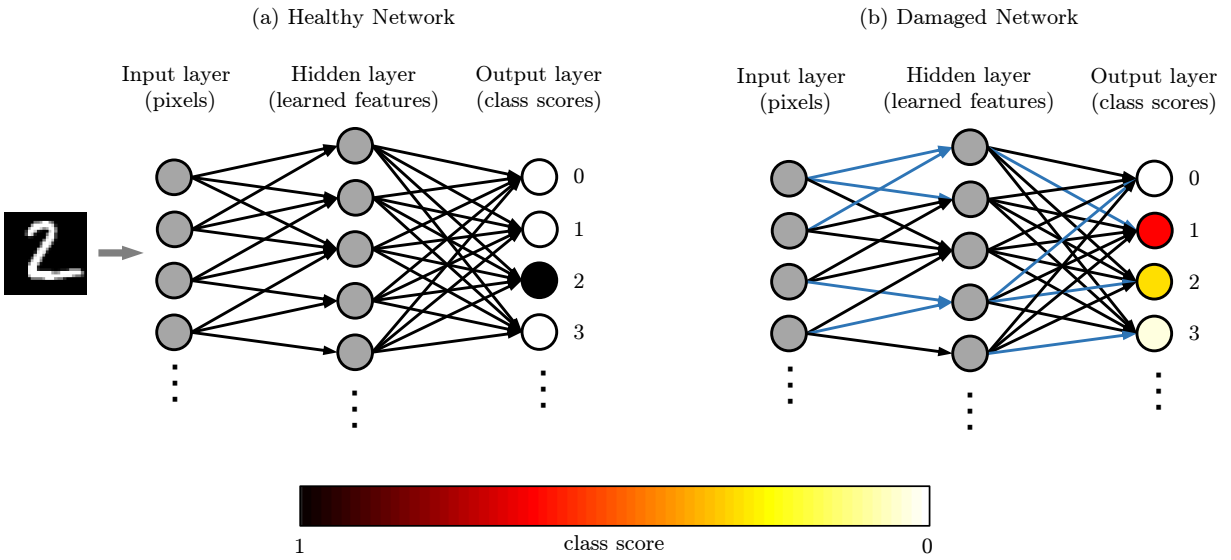

Figure 4.1: Damaging a Convolutional Neural Network (CNN). (a) We start with a "healthy" CNN that accepts an image of a handwritten digit as an input and outputs scores for each possible digit, 0-9. We classify the image as the digit with the highest score. (b) We then damage the weights on the network in a biophysically-relevant way. In this figure, the healthy network correctly classifies the image as a 2, but the damaged network classifies it as a 1.

4.1 Introduction

The mathematical architecture of convolutional neural networks (CNNs) was originally inspired by the Nobel prize-winning work of Hubel and Wiesel on the primary visual cortex of cats [81]. Their seminal experiments were the first to suggest that neurons in the visual system organize themselves in hierarchical layers of cells for processing visual stimulus. The first quantitative model of the CNN, termed the Neocognitron by Fukushima in 1980 [52], already displayed many of the characteristic features of today's deep CNNs, including a multi-layer structure, convolution, max pooling and nonlinear dynamical nodes. The connection between neuroscience and CNN theory, although clearly a conceptional abstraction [135], has since

been instrumental to improve quantitative models of how the brain integrates neuro-sensory information for stimulus classification and decision making. Given that CNNs mimic many of the important cognitive features of the brain, we use it as a model for understanding how neurodegenerative diseases and traumatic brain injuries (TBI) can compromise an array of recognition tasks. Specifically, by using well-established biophysical data on the statistics (distribution and size) of focal axonal swellings (FAS), which are the primary symptoms of neurodegeneration and TBI, we evaluate the progress of impairments on a CNN-based model of cognition. Our model provides quantitative metrics for understanding how cognitive deficits are accumulated as a function of FAS development, allowing for potentially new diagnostics for the evaluation of brain disorders due to neurodegenerative diseases and/or TBI.

Understanding how neurodegenerative diseases and traumatic brain injuries affect cognitive function remains a critically important challenge for societal mental health. TBI alone is one of the major causes of disability and mortality worldwide, which in turn, dramatically jeopardizes society in several socioeconomic ways [118]. Not only is it the signature injury of the wars in Afghanistan and Iraq [87], but also the leading cause of death to young people [49]. While many survive the events that induce TBI, persistent cognitive, psychiatric, and physiological dysfunction often follows from the mechanical impact (see Sec. 2). Likewise, neurodegenerative diseases are responsible for an overwhelming variety of functional deficits, with common symptoms including memory loss or behavioral/cognitive impairments which are related to an inability to correctly process multi-modal information for decision-making tasks. The majority of brain disorders have a complex cascade of pathological effects spanning multiple spatial scales: from cellular or network levels to tissues or entire brain areas. Unfortunately, our limited ability to diagnose cerebral malfunctions in vivo cannot detect several anomalies that occur on smaller scales. FAS, however, are ubiquitous to TBI and most leading and incurable disorders that dramatically affect signaling properties of neurons, such as Multiple Sclerosis, Alzheimer's and Parkinson's diseases.

Given the currently available wealth of data on FAS morphology from TBI studies and

from almost every leading neurodegenerative disease, significant progress can be made towards understanding qualitatively how FAS impacts cognitive function. In this work, we consider a set of deep CNN models as an *abstraction* for functioning brains. Our focus is to understand how the processing of input data (classification) is compromised as a function of increasing injury and/or disease progression. Of course, it is obvious that the system's performance will be compromised as the CNN is injured, but the *manner* in which the cognitive impairments arise is quite illustrative and informative, providing intuitively appealing results about how cognitive deficits can develop and evolve as a neurodegenerative disease progresses.

Figure 4.1 illustrates our approach. We begin with the original (healthy) CNN, which is trained to perform a classification task. In Figure 4.1, the specific task is to label a handwritten digit. We then expose the CNN to different injury protocols based upon biophysical observations of FAS statistics and morphological parameters. In particular, we use statistical distributions of FAS from a recent experiment consisting of TBI-induced damage in the visual cortex of rats [163]. To impose these injury statistics on the original CNN, we assume that each neuronal connection has a biophysically plausible probability to malfunction; while mild axonal injury may simply weaken a connection, severe cases may break it permanently (i.e., set the connection strength to zero). Ultimately, the severity of the injury and re-weighting of connections is also determined by biophysical data and the statistical distribution of the size of the FAS. We can then progressively monitor the deleterious effects of the injury on the functionality of the CNN, providing metrics for cognitive deficits that arise.

The chapter is outlined as follows: In Sec. 4.2 we provide key background material on the two primary fields integrated into this work: convolutional neural networks and neural disorders in which FAS are implicated. We describe our methodology in Sec. 4.3 and present results in Sec. 4.4. We summarize our conclusions in Sec. 4.5.

4.2 Background

4.2.1 Convolutional Neural Networks

Deep convolutional neural networks (DCNNs) are transforming almost every field of science involving big data. The success of the method has been enabled by two critical components: (i) the continued growth of computational power (e.g. GPU and networked computing), and (ii) exceptionally large labeled data sets capable of taking advantage of the full power of a multi-layer architecture. Indeed, although the theoretical inception of CNNs has an almost four-decade history, the analysis [96] of the ImageNet data set [39] in 2012 provided a watershed moment for CNNs and Deep Learning [104]. Prior to this data set, there were a number of data sets available with approximately tens of thousands of labeled images. ImageNet provided over 15 million labeled, high-resolution images with over 22,000 categories. DCNNs have since transformed the field of computer vision by dominating the performance metrics in almost every meaningful computer vision task intended for classification and identification (see, for example, the International Conference on Computer Vision 2015).

ImageNet has been a critically enabling data set for the evolution of the field. However, CNNs were a topic of intensive research long before. Indeed, they were highly successful in a wide range of applications and machine learning architectures. By the early 1990s, the neural networks were studied as standard textbook material [17], with the focus typically on a small number of layers. Critical machine learning tasks such as PCA decompositions were shown to be intimately connected with networks which included back propagation [12, 141]. Importantly, there were a number of critical innovations which established multilayer feedforward networks as a class of universal approximators [77]. Specifically, Hornik et al. rigorously established that standard multilayer feedforward networks with as few as one hidden layer using arbitrary squashing functions were capable of approximating any Borel measurable function from one finite dimensional space to another to any desired degree of accuracy, provided sufficiently many hidden units were available. Thus, multilayer feedforward networks could be thought of as a class of universal approximators [77].

The past five years have seen tremendous advances in the DCNN architecture. Innovations have come from algorithmic tricks and modifications that have led to significant performance gains in a variety of fields. These innovations include pretraining [73, 15, 45], dropout [150], max pooling [96], inception modules [153], data augmentation (virtual examples) [127], batch normalization [83] and/or residual learning [65]. This is only a partial list of potential algorithmic innovations available for improving the performance of classification and labeling. Our goal is not to provide a complete review of the DCNN field, but rather to highlight the continuing and rapid pace of progress in the field. Integrating the state-of-the-art in DCNNs is the open source software called TensorFlow (tensorflow.org). TensorFlow was originally developed by researchers and engineers working on the Google Brain Team within Google's Machine Intelligence research organization. The system is designed to facilitate research in machine learning and to make it quick and easy to transition from research prototype to production system. TensorFlow has allowed for the test-bedding of new algorithmic structures in a reproducible and verifiable manner, which is a significant and important advancement in the field. Indeed, the DCNN architecture used here relies on the TensorFlow architecture, helping us understand how state-of-the-art DCNNs relate to cognitive abilities.

4.2.2 Focal Axonal Swellings

Concussions and Traumatic Brain Injuries (TBI) are more than ever a concern for contact sport practitioners [48], for veteran soldiers exposed to blast injuries [38, 87], and for society as a whole [49, 118, 138]. In fact, TBI contributes to one-third of all injury-related deaths and is one of the major drivers of functional impairments. TBI pathologies affect several spatial scales [145], but a ubiquitous development at the neuronal microenvironment level is the presence of axonal injury [72, 84, 147]. As reviewed in [72], rapid axonal stretch injury triggers secondary axonal changes that can vary in extent and severity [43, 63, 69], but most often culminate in Focal Axonal Swellings (FAS).

FAS are monitored whenever possible in *in-vitro* studies [30, 50, 66, 67, 68, 110, 122, 148],

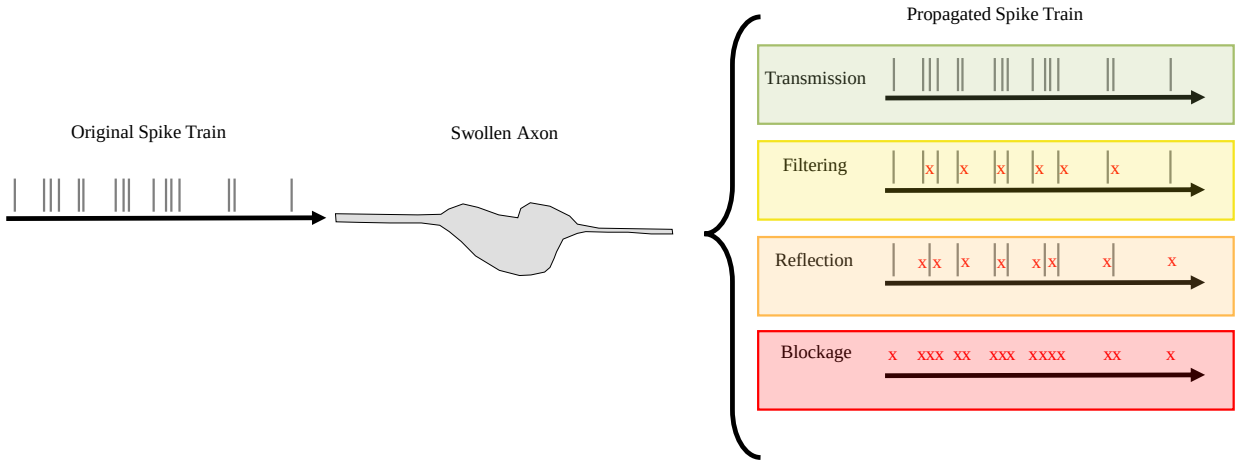

Figure 4.2: Four Types of Damaged Axons. A spike train passes through a swollen axon. Depending on the way that the axon is swollen, there are four ways that the information can be transmitted. In transmission, the spike train is propagated correctly despite the damage. In filtering, the spike train goes through a low-pass filter. Regions of the spike train with high frequency are especially likely to lose spikes. In reflection, pairs of spikes combine and only half of the spikes are transmitted. In blockage, none of the spikes are transmitted.

in *in-vivo* experiments [23, 40, 116, 163], and in human patients [3, 19, 35, 54, 87, 93, 136]. In many cases, FAS critically affect the axonal morphology [155, 156] and consequently, the information content encoded in spike trains propagating throughout them.

Recent computational studies distinguished geometrical axonal enlargements that lead to minor changes in propagation from those that result in critical phenomena such as reflection or blockage of the original traveling pulse [113], or filtering of action potentials [112]. This led to a diagnostic toolbox that extracts meaningful geometrical parameters from sequential images of injured axon segments [111]. These algorithms provide a principled approach to deal with imaging distortions caused by experimental artifacts in order to extract the cross-section of an axon by detecting local symmetries, turning points and turning regions. More importantly, they provide the first description of biologically plausible injurious effects due to FAS that can be incorporated into neuronal network simulations. Figure 4.2 reviews these different effects; in the transmission regime, the spike train propagates through the FAS without significant modifications. In the filtering regime, pulses that are too close to each other get deleted by a mechanism named pile-up collision [112]. As the FAS geometrical parameters worsen, a single spike will split into two components, one propagating forward and the other propagating backward. The reflected, back-propagating pulse will collide with the next spike in the train and they will mutually annihilate each other. Thus, the reflection regime effectively halves the firing rate of the neuron. Finally, in the worst-case scenario, the FAS will block all spikes and transmit no information whatsoever.

In what follows, we will introduce FAS in a biologically plausible way to a few examples of deep-learning convolutional neural networks and evaluate the extent to which cognitive deficits develop.

Table 4.1: Summary of CNNs Used

Task	Training Set	# conv. layers	# fully connected layers
handwritten digit classification	MNIST	2	1
object classification	ImageNet	5	3
face verification	VGG-Face	13	2

4.3 Materials and Methods

4.3.1 CNNs original calibration, training and performance

We simulate the development of FAS damage in three different convolutional neural networks. Each network has its own properties and was trained with different data sets for separate tasks (see Table 4.1).

The MNIST network classifies a handwritten number as a digit from 0 to 9 and could be used, for example, by a post office machine to read zip codes from envelopes. The training data set consists of a series of black and white images that are 28 by 28 pixels. We used the TensorFlow framework [1] to train a CNN with two convolutional layers and a fully connected layer, as advised by a TensorFlow tutorial. We use a subset of the standard MNIST test set for our testing purposes so that our set contains the same number of examples for each digit. In particular, we picked the first 852 images for each digit. Our trained network has an accuracy of 98.74% on this test set.

The ImageNet network classifies images from the ILSVRC 2012 challenge among one thousand possible classes. The CNN-F network was pre-trained by the Visual Geometry Group at Oxford [28] and made available through the MatConvNet Matlab Toolbox [161], where it is referred to as imagenet-vgg-f. The network contains five convolutional layers

and three fully connected layers. For our experiments, we use a subset with two examples randomly chosen from each class. The network is 54.6% accurate on this test set.

The VGG-Face network is trained to classify faces as one of one thousand people. However, if you remove the last classification layer and normalize the output vector, the network can instead be used to output feature vectors for face verification. If the Euclidean norm of the difference between the feature vectors for two images is under a threshold τ, the pair of images is classified as being the same person. This network was also trained by the Visual Geometry Group [130] and made available through the MatConvNet toolbox [161], where it is called vgg-face. For our experiments, we randomly chose five pictures each of fifty randomly chosen celebrities from the Labeled Faces in the Wild (LFW) data set [80]. We also needed to choose a threshold τ. We chose $\tau = 1.2$ based on Linear Discriminant Analysis on a training set of 5700 examples from the LFW data set. Each of the 250 images in our test set is then compared to the four other images of the same person and four images of other people. We thus test one thousand pairs of images, half of which are of the same person and half of which are not. The network is 81.6% accurate with $\tau = 1.2$ on our test set of one thousand pairs of images.

4.3.2 Network impairments following FAS injuries

To simulate the effects of traumatic brain injury on a CNN, we randomly "damage" p percent of the weights in the convolutional and fully connected layers. For consistency with the TBI analogy, we only target the connections between neurons and not bias weights. For simplicity, we first assume that all axonal injuries lead to the total blockage of spikes, which effectively sets p percent of weights to zero. We consider damage examples for each one of the previously described networks to develop intuition about possible functional impairments.

In Figure 4.3, we choose a handwritten "2" as the input to the MNIST network. The network assigns a score to each of the ten possible digits and then classifies the image as the digit with the highest score. The original network gives scores of .999987 to 2, .000013 to 1, and 0 to the rest of the digits and thus correctly classifies the image as a 2. We then randomly

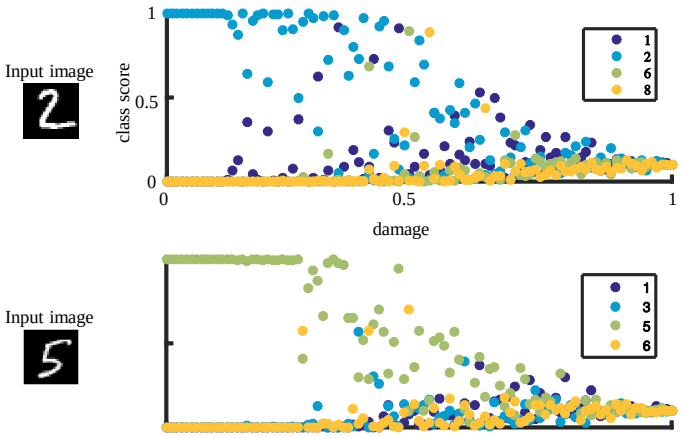

Figure 4.3: Change in Class Scores on Damaged Networks. This CNN accepts an image of a handwritten digit as an input and outputs scores for each possible digit, 0-9. In these two examples, the original network correctly and confidently classifies the digit. As we increase the damage level, confidence drops and the classes eventually become confused. For high levels of damage, all classes have similar scores.

damage the network in 100 separate experiments, setting $p = .01, .02, \ldots, 1$. Since we are simulating TBI, the damage happens all at once and is not accumulated across experiments. Thus, the set of damaged neurons with $p = .01$ may have little overlap with the targeted neurons in the $p = .02$ case. At around 12% damage, the network becomes noticeably less confident, but still correct. The network makes its first mistake at 30% damage by labeling the image as a 1. At higher levels of damage, it frequently confuses classes 1 and 2. After 90% damage, the ordering of the class scores continues to change, but their values become quite similar. We see an analogous pattern in the second part of Figure 4.3 where we input a "5" as an example, although the damaged networks make fewer mistakes.

Next, we consider two examples from the image classification problem. In Figure 4.4, we use an image of a green bell pepper as the input to the ImageNet network. Here the network assigns a score to each of one thousand possible classes before matching it with

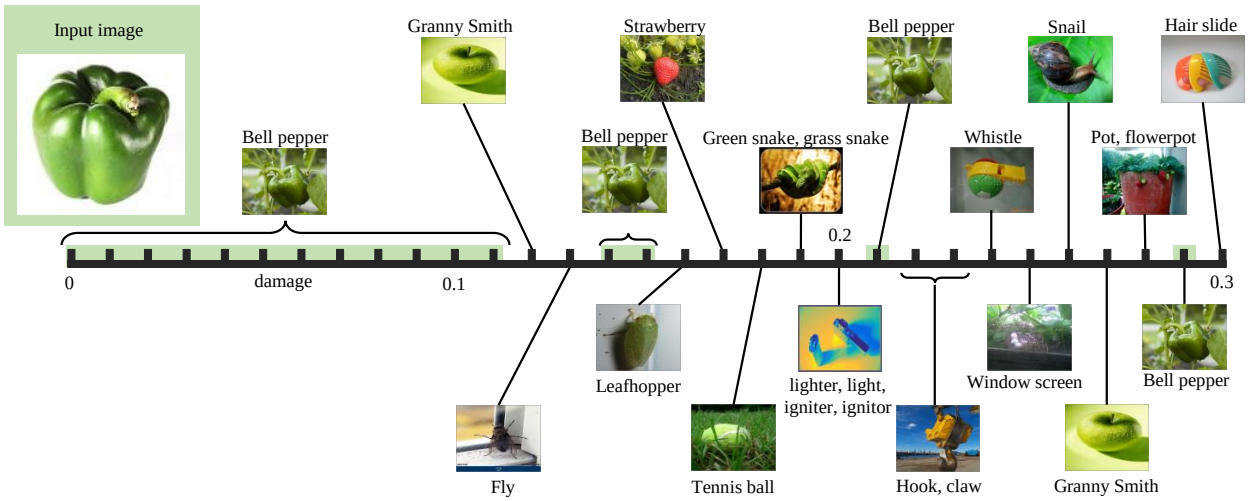

Figure 4.4: Classification Mistakes as Damage Increases, Example 1. We start with a healthy network trained to classify images. The original network correctly classifies this image as a green pepper, but with enough damage, the network makes mistakes. For moderate amounts of damage, the wrong classifications make some intuitive sense.

the class with highest score. We visualize how the classification changes as the damage increases ($p = 0, 0.01, 0.02, \ldots, 0.3$). The network makes its first mistake at 12% damage but sometimes returns to classifying the image as a bell pepper. Some of the mistakes are relatively sensible, such as "granny smith" or "tennis ball," and share similar colors, textures and/or shapes with the original image. Some of the later mistakes seem less understandable, such as "hair slide." Note that this network was trained on about 1.2 million images of the one thousand classes, encompassing a wide range of examples for each class. For illustration purposes in this figure, we show an example image from the test set for each class. However, the input image is downloaded from Flickr [89].

In Figure 4.5, we give a more difficult input image to the ImageNet network—a group of vegetables composed predominately by peppers with a variety of colors but also containing garlic. This image accompanies the Image Processing Toolbox for Matlab as peppers.png and is used as a demo for this network in the MatConvNet Toolbox. The network successfully

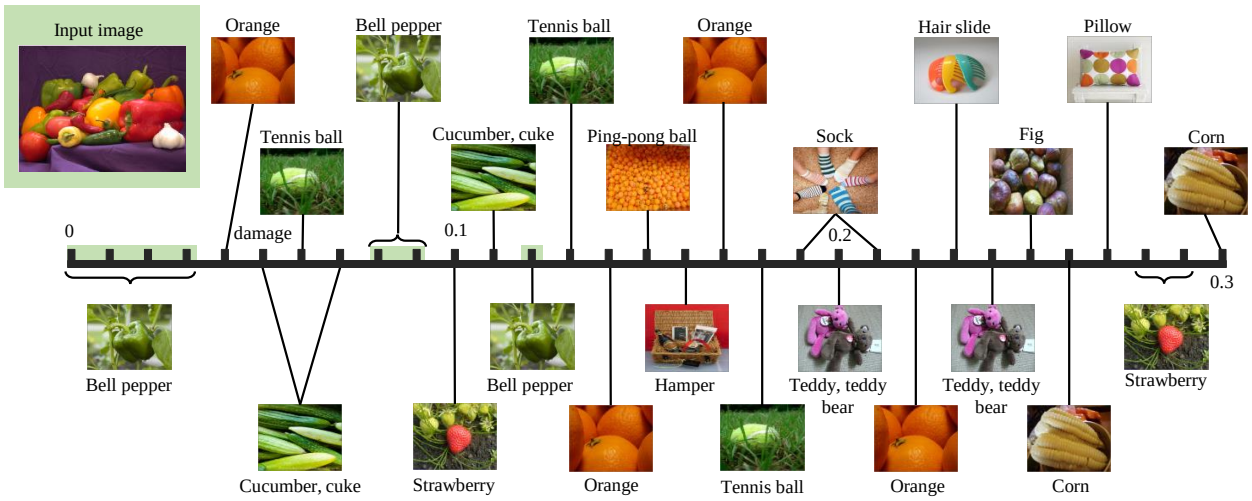

Figure 4.5: Classification Mistakes as Damage Increases, Example 2. We increase the difficulty by using an image of a group of vegetables, primarily bell peppers. The network does not maintain the "bell pepper" classification as long, but the early mistakes are also produce or also round items.

chooses the bell pepper class among one thousand possibilities, but it is not as robust to injury as the one in the previous, easier example. Misclassifications begin at 4% injury. Again, some errors are reasonable, such as a "cucumber" or "orange" (which is not that different from an orange-colored pepper). Others are quite surprising, such as "socks" or "teddy bear".

In Figure 4.6, we show analogous deficits for the facial recognition network (VGG-Face). We input an image of George W. Bush to the network and have it compare the image with three other pictures—one of his father, one of Bill Clinton, and another one of himself. All images come from the LFW data set. The healthy network correctly identifies that both pictures of George W. Bush are of the same person and that the pictures of his father and Bill Clinton are of different people. We also see that George W. Bush is closer to his father than to Bill Clinton. As the damage increases, the network occasionally classifies George H. W. Bush as being the same as his son and, eventually, cannot even distinguish

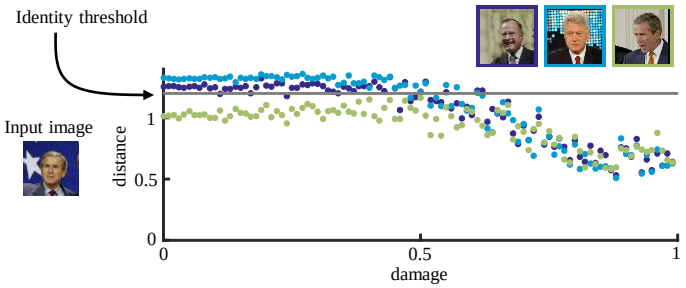

Figure 4.6: Change in Distance Between Images. This network outputs a feature vector for each image and can be used to find the distance between two images. If the distance is below our threshold τ, the pair is labeled as being of the same person. The network originally correctly identifies the second image of George W. Bush as being the same person while labeling the images of George H. W. Bush and Bill Clinton as being different people. After sufficient damage, the distances between the images all shrink and it is not possible to determine whether or not a pair of images are of the same person.

Bill Clinton. After about 70% damage, all images start looking alike, and the network continuously exchanges the ordering of the distances as the damage increases. This pattern continues in the broader experiments and, with enough damage, all pairs of images are labeled as being of the same person. Note that adjusting the threshold as damage increases would not improve the accuracy since the second picture of George W. Bush does not remain closer than the pictures of other people.

4.4 Results

In this section, we move from qualitative descriptions of single network errors to a more broad, statistical account of mistakes within the test sets. We also consider a few variations of FAS injury protocols, network settings, and their dynamics to model biologically relevant phenomena such as aging and the development of neurodegenerative effects across CNNs.

4.4.1 Overall network impairments

In Figure 4.7, we return to the MNIST handwritten digit classification task and plot confusion matrices $M_{i,j}$ for the ten digits as the damage percentage p increases. At $p = 0\%$, the network has 98.75% accuracy, so the matrix is concentrated on the diagonal ($i = j$). At $p = 20\%$, we begin to see substantial errors ($i \neq j$), especially by over-classifying digits 0, 4, and 9. As the damage increases, the confusion matrices become even more distributed, but the types of errors change. For example, at 40% damage, some especially common labels are 1 and 6, while at 60% damage, the disproportionately common labels become 0, 2, 4, and 7. However, recall that in our TBI analogy, the damage is not accumulated—in each experiment, we return to the original network and randomly choose a new set of weights to damage. At $p = 100\%$, there is no randomness; all weights are set 0 and all images are labeled as a 1. Overall, confusion matrices provide a straightforward visualization for misclassification within the CNN data set that could be advantageous for diagnosing cognitive deficits.

In Figure 4.8, we summarize our TBI experiments for (i) the MNIST network, (ii) the ImageNet network, and (iii) the Facial Recognition network. All targeted neurons are assumed to malfunction the same way, fully blocking the signal transmission to their neighbors. For each damage level p (%), we average the accuracy across all random trials on that network. Again, there is no randomness when $p = 0\%$ or 100%. At $p = 100\%$, all weights are set to 0 and all examples are placed in the same class. As expected, the accuracy of the network decays asymptotically to $1/n$, where n is the number of classes. Note that the MNIST network and the ImageNet network have qualitatively similar trends, and display some accuracy deficit even at low injury levels. In other hand, the Facial Recognition network is able to maintain its original accuracy level past $p = 50\%$ before decaying abruptly.

4.4.2 Relevance of connections and biological constraints

In all three examples of network dysfunction, there is a considerable amount of variability across trials even for the same injury levels. We found that the deficits greatly depend on

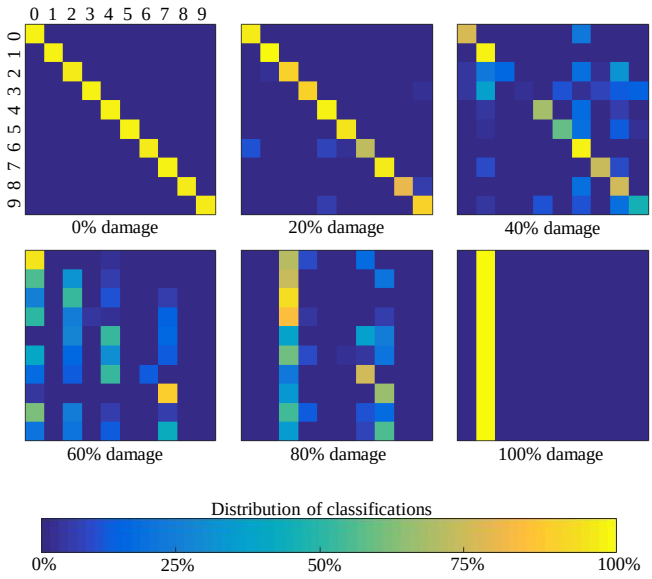

Figure 4.7: Confusion matrices as damage increases. We depict the classification results of the handwritten digit classification network for varying amounts of damage. If the images are perfectly classified, only the diagonal is colored. As the damage increases, most images are mapped to the same few digits. Eventually, all images are classified as a one.

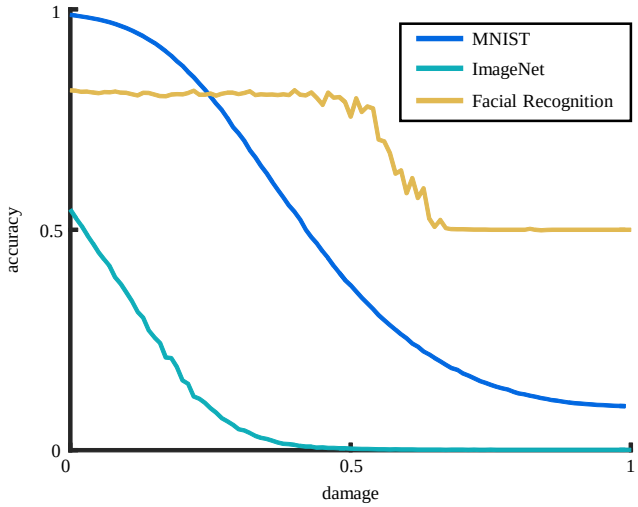

Figure 4.8: Accuracy decay as damage increases. We randomly damage edges of the network by setting their weights to zero. We plot the percentage of edges that are damaged against the average accuracy of the network for three problems. We see that damage initially has little effect, but then there's a steep drop off until the accuracy levels off around the level of random guessing.

which weights were randomly selected. In other words, neuronal connections in CNNs do not contribute equally to a task, and damaging weights with large magnitude typically impacts the accuracy more than targeting weaker links—although magnitude alone cannot explain all cases.

We illustrate some of these issues in Figure 4.9 for the the MNIST network. We repeat the average decay in accuracy in blue but add error bars. We found that roughly the worst case is to damage the weights in decreasing order of magnitude instead of randomly. The resulting steep accuracy drop off is plotted in teal. Conversely, the approximate best case is to damage the weights in increasing order of magnitude (plotted in gold). These three damage strategies are visualized in terms of their effect on the distribution of weights. The purple histogram displays the distribution of the original weights. In general, we randomly choose weights to damage, so the effect is distributed across the distribution of weights. However, damaging the weights in order of decreasing magnitude is equivalent to progressively removing the tails of the histogram, and choosing the weights in increasing order of magnitude is equivalent to removing the middle of the histogram. These experiments may provide intuition into why the outcomes of TBI are so difficult to predict. We hypothesize that randomness in the location of FAS could explain, for instance, why two soldiers near the same explosion site may develop significantly different post-traumatic outcomes.

One of the most striking differences between artificial CNNs and biological neuronal networks is that the latter must operate under geometric, biophysical and energy constraints. As reviewed in [101], brains have evolved to operate efficiently since economy and proficiency are guiding principles in physiology. In fact, nervous systems are a major drain on an animal's energy budget and many aspects of the brain's anatomy seem to limit wiring costs [34, 91, 149]. Brain networks can therefore be said to negotiate an economical trade-off between minimizing inter-neuron connection cost & maximizing topological value and capacity for information processing. See Bullmore and Sporns [24] for a recent review on the topic.

The MNIST network could be more biophysically plausible if it was not as over-engineered. With its original topology and settings, the CNN becomes artificially resistant to damage.

In what follows, we will first *sparsify* the CNN by picking a point on the accuracy-efficiency trade off curve (see Figure 4.9). There are multiple ways to choose the best trade-off point. A reasonable choice is to remove the weakest 69.4% of the links, which decreases the accuracy from 98.74% to 91.47%.

4.4.3 Different types of FAS dysfunctions

As described in Section 4.2.2, Focal Axonal Swellings (FAS) affect spike trains in four qualitatively different regimes: transmission, filtering, reflection, and blockage. So far we have only considered the worst case, blockage, which we model by setting a weight to zero. Now we also consider the other types of neuronal malfunctions. We model transmission as not damaging a weight, reflection as halving a weight, and filtering as applying a low-pass filter on each weight. We choose an example filtering function of $f(x) = -.2774x^2 + .9094x - .0192$ plus Gaussian noise $\sim N(0, 0.05)$ by fitting a confusion matrix from experimental results [112]. We believe that these additions to CNNs contain, in a tractable way, the key features of the jeopardizing effects caused by FAS described in [113, 112, 111].

Recent experimental results provide detailed morphological descriptions of the FAS developing after traumatic brain injuries. Wang et al [163] damaged the optic nerve of adult rats with a central fluid percussions injury. The optic nerve is a relatively organized bundle of axons and allowed for monitoring of FAS development 12h, 24h and 48h after the impact. They divided the nerve in 12 serial grids and reported for each of them the number of axonal swellings per unit area, the total area of axonal swellings, and the individual sizes of swellings. It is possible to infer the fraction of FAS in each dysfunctional regime from these statistical distributions [114]. Based on these results, we conduct numerical experiments with 30% blockage, 45% reflection, 20% filtering, and 5% transmission. In Figure 4.10, we show results for the sparsified MNIST network, comparing its average accuracy for heterogeneous and homogeneous FAS distributions. As expected, the worst deficits occurred when all of the swellings were in the blockage regime. The best case is when all of the FAS are in the filtering regime, closely followed by the related reflection case. When we combine these

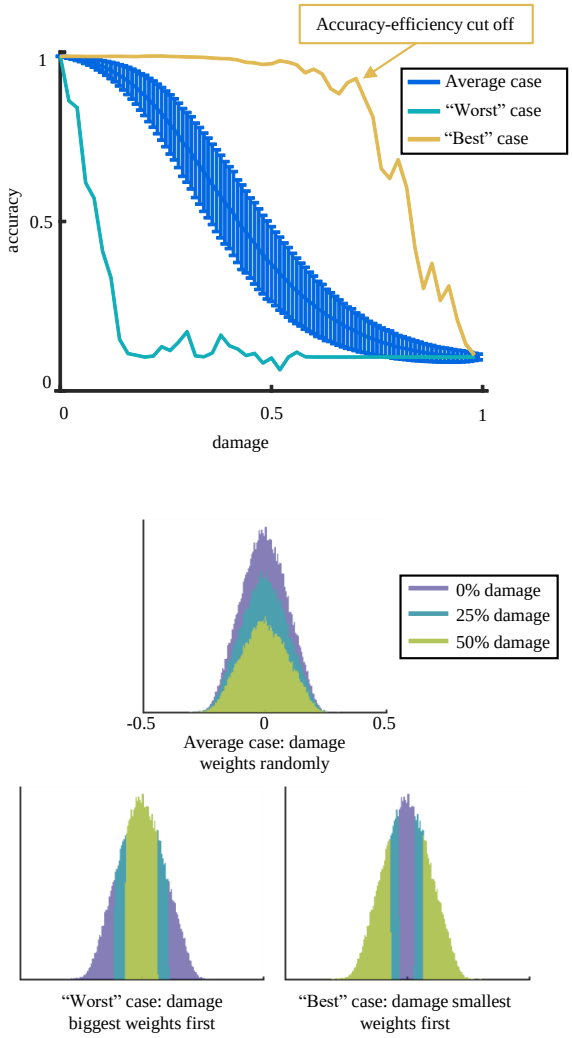

Figure 4.9: Range of possible outcomes. The change in accuracy as weights are damaged varies depending on which weights were randomly chosen. In blue, we plot the average accuracy plus error bars for each level of damage. We also add curves in teal and yellow for approximations of best and worst-case accuracies, respectively. The approximate worst-case was found by damaging the weights in decreasing order of their absolute value. Similarly, the approximate best-case was found by damaging the weights in increasing order. We give a visualization in terms of a histogram of what it means to damage the weights in a random order ("average case"), in decreasing order ("worst case"), and in increasing order ("best case"). The yellow "best case" provides an accuracy-efficiency trade off. We choose a turning point in the curve: if we remove the smallest 69.4% of the weights, the accuracy only decreases from 98.74% to 91.47%.

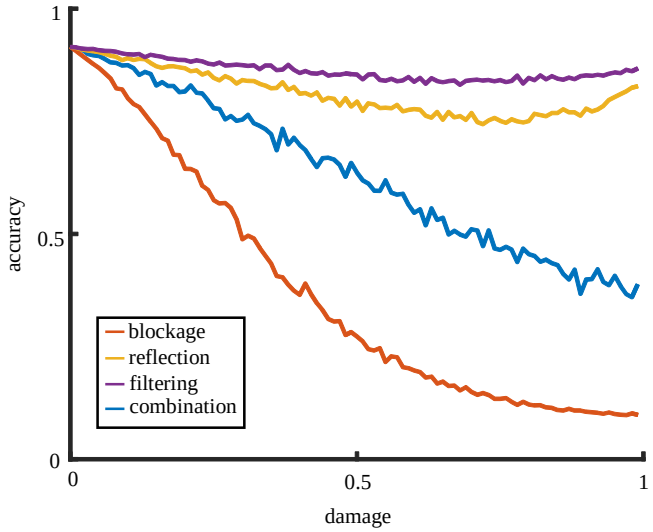

Figure 4.10: Comparing types of damage. In these experiments, we begin with a "sparsified" network with the smallest 69.4% of the weights removed. Then we compare the types of FAS (blockage, reflection, and filtering) and a combination of all types based on experimental evidence. As expected, blockage causes the most damage, and reflection is a strong form of filtering.

regimes (30% blockage, 45% reflection, 20% filtering, and 5% transmission), the accuracy is between these more extreme cases.

4.4.4 Aging and Neurodegenerative Diseases

Alzheimer's Disease (AD) is the most commonly found type of dementia, which is an umbrella term for a variety of brain disorders and pathologies [88]. Aging is the single greatest risk factor for AD [131], and most public health systems across the developed world are expected to face huge challenges due to the growing elderly population [137]. Recent estimates suggest that more than 5.2 million people have AD in the United States alone and that a new case occurs every 68 seconds [157]. The most typical symptom of the disease is an

increasing difficulty in recalling new information, although it sometimes occurs in conjunction with challenges in completing familiar tasks, confusion with time or place, and trouble understanding visual images and spatial relationships. Find more information about AD symptoms here.

W. Thies and L. Bleiler [157] report that in many cases, AD diagnostics are accompanied by cognitive tests, since individuals with mild cognitive impairments have changes in thinking abilities that are noticeable to family members and friends. We believe, however, that there is still a large degree of subjectivity when it comes to interpreting cognitive deficits from dynamically evolving complex systems such as the human brain. Thus, simulations with convolutional neural networks that incorporate biophysically plausible neural malfunctions may provide a window of opportunity to better diagnose, for instance, confusion in visual image classification. On this account, focal axonal swelling pathologies are present in AD [2, 37, 97, 159] and in other neurodegenerative diseases such as Parkinson's disease [154, 107, 53], Multiple Sclerosis [51, 125, 158], and others [70, 92, 102, 103]. In many cases, FAS arise by the agglomeration of specific proteins over time [36, 119], and again, the computational modeling of focal axonal swellings and their effects to spike propagation from [111] provide a platform to investigate network dysfunction.

In all of the previous experiments, we simulated TBI by abruptly applying axonal injuries. Here we instead simulate aging and its neurodegenerative effects by gradually accumulating random damage. We continue to use the sparsified network and the heterogeneous FAS distribution. Figure 4.11 shows that if damage is applied at a constant rate (targeting 1% of the connections at each step), the results will look similar to a sequence of TBI experiments with $p = .01, .02, \ldots$ (Figure 4.10, in dark blue) except that each trial will have a smoother trajectory. This is translationally relevant since traumatic brain injuries dramatically increase the risk of dementia later in life [13, 85, 86, 105]. Perhaps a more plausible and biophysically relevant case occurs when the FAS accumulation rate increases linearly with time (in cyan). There, the young brain accumulates very little damage, but the older brain rapidly acquires new swellings.

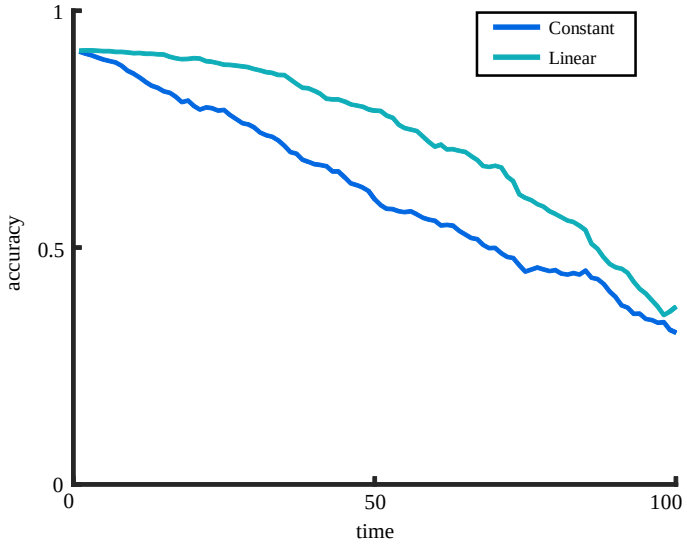

Figure 4.11: Accumulating damage over time. In aging or neurodegenerative disease, damage to axons is accumulated over time, in contrast to a one-time injury. We compare the accuracy curves for a constant number of connections damaged for each time step to the case where the number of connections damaged increases with time. When damage increases over time, the initial loss in accuracy is slow and the later loss is faster.

4.5 Conclusions

Assessing levels of cognitive deficits in patients is largely a subjective task, with indicators such as whether or not the patient and those close to them have noticed difficulties with memory. There are some tools available, including the Mini-Mental State Examination (MMSE). The MMSE assigns a score after testing performance on a brief series of tasks such as identifying objects and following written instructions. This score can be used to quantitatively track changes in a person's cognitive function. Similarly, in this chapter, we calculate the change in accuracy on related tasks, such as reading handwritten numbers and labeling objects. Since we can conduct extensive experiments with any level of injury, we believe that simulating FAS on our model of cognition can lead to insight into the complex processes underlying TBI and neurodegeneration.

Non-invasive diagnostic tools cannot detect anomalies in vivo such as FAS that occur at the cellular level. In fact, this has motivated a large body of in vitro experiments to replicate these injuries in a controlled setting [30, 50, 66, 67, 68, 110, 122, 148]. However, in the latter case, the cognitive effects of these injuries cannot be assessed. Simulations provide an opportunity to connect understanding of FAS to measures of cognitive performance.

Both CNNs and brains operate somewhere on an accuracy-efficiency trade-off curve. However, arguably brains are more highly constrained than CNNs due to the high energy costs of nervous systems [101]. In contrast, many CNNs are trained with a high focus on small gains in accuracy, especially those trained for competitions such as ImageNet. In addition, all three of the CNNs studied here utilized dropout, which encourages redundancy in the weights [150]. A key step in our methodology was to prune the CNNs to be less over-engineered and thus more biologically plausible. Remarkably, the networks performed very well even if many weak connections were removed.

Our simulations of damage on our model of cognition result in interpretable and human-like mistakes, such as confusing a handwritten 5 with a 6 (Figure 4.3), labeling peppers as an apple or a cucumber (Figures 4.4 and 4.5) and confusing George W. Bush with his father

(Figure 4.6). We are able to quantify how accuracy changes as damage increases (Figures 4.8, 4.9, 4.10, and 4.11) as well exactly which kinds of mistakes are being made (Figure 4.7). We demonstrate that the effect on accuracy is highly variable and depends on which connections are randomly selected (Figure 4.9), providing intuition for why impairments are difficult to predict.

As with any model, using CNNs as an abstraction for the brain comes with limitations. Biological neural networks have many complex features and constraints that are not factored into our model. One important difference between convolutional neural networks and human subjects is the latter's ability to infer significantly more information from the *context* of an image. For instance, a patient classifying all objects depicted in Fig. 4.5 might, due to some form of meta-analysis, readily interpret them as a collection of many-colored peppers. Consequently, he could discard extraneous objects from a list of candidates (like ping-pong/tennis balls) even if their shape and color alone do not provide sufficient evidence for such dismissal. We would encourage the usage of images with non-sensical pairings of objects to circumvent this difficulty in diagnostic tests for cognitive deficits.

In summary, we provide a platform for quantitatively and qualitatively studying the progression of focal axonal swellings in a neural network. We can provide insight into disorders which feature FAS, such as TBI, Alzheimer's, Parkinson's, and Multiple Sclerosis, linking damage at the cellular level to changes in cognitive behavior.

BIBLIOGRAPHY

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[2] Robert Adalbert, Antal Nogradi, Elisabetta Babetto, Lucie Janeckova, Simon A. Walker, Martin Kerschensteiner, Thomas Misgeld, and Michael P. Coleman. Severely dystrophic axons at amyloid plaques remain continuous and connected to viable cell bodies. *BRAIN*, 132:402–416, 2009.

[3] J. Hume Adams, Bryan Jennett, Lilian S. Murray, Graham M. Teasdale, Thomas A. Gennarelli, and David I. Graham. Neuropathological findings in disabled survivors of a head injury. *Journal of Neurotrauma*, 28:701–709, 2011.

[4] Genevera Allen. Sparse higher-order principal components analysis. In *Proceedings of the 15th International Conference on Artificial Intelligence and Statistics*, 2012.

[5] Joan Francesc Alonso, Sergio Romero, Miquel Àngel Mañanas, and Jordi Riba. Serotonergic psychedelics temporarily modify information transfer in humans. *International Journal of Neuropsychopharmacology*, 18(8), 2015.

[6] L Angelini, M De Tommaso, D Marinazzo, L Nitti, M Pellicoro, and S Stramaglia. Redundant variables and granger causality. *Physical Review E*, 81(3):037201, 2010.

[7] Leonardo Angelini, Mario Pellicoro, and Sebastiano Stramaglia. Granger causality for circular variables. *Physics Letters A*, 373(29):2467–2470, 2009.

[8] Marco Tulio Angulo, Jaime A Moreno, Albert-László Barabási, and Yang-Yu Liu. Fundamental limitations of network reconstruction. *arXiv preprint arXiv:1508.03559*, 2015.

[9] Brett W. Bader and Tamara G. Kolda. Algorithm 862: MATLAB tensor classes for fast algorithm prototyping. *ACM Transactions on Mathematical Software*, 32(4):635–653, December 2006.

[10] Brett W. Bader and Tamara G. Kolda. Efficient MATLAB computations with sparse and factored tensors. *SIAM Journal on Scientific Computing*, 30(1):205–231, December 2007.

[11] Brett W. Bader, Tamara G. Kolda, et al. Matlab tensor toolbox version 2.6. Available online, February 2015.

[12] Pierre Baldi and Kurt Hornik. Neural networks and principal component analysis: Learning from examples without local minima. *Neural networks*, 2:53–58, 1989.

[13] D. E. Barnes, A. Kaup, K.A. Kirby, A. L. Byers, R.Diaz-Arrastia, and K. Yaffe. Traumatic brain injury and risk of dementia in older veterans. *Neurology*, 83:312–319, 2014.

[14] Lionel Barnett and Anil K Seth. The mvgc multivariate granger causality toolbox: a new approach to granger-causal inference. *Journal of neuroscience methods*, 223:50–68, 2014.

[15] Yoshua Bengio, Pascal Lamblin, Dan Popovici, Hugo Larochelle, et al. Greedy layer-wise training of deep networks. *Advances in neural information processing systems*, 19:153, 2007.

[16] Monica Billio, Mila Getmansky, Andrew W Lo, and Loriana Pelizzon. Econometric measures of connectedness and systemic risk in the finance and insurance sectors. *Journal of Financial Economics*, 104(3):535–559, 2012.

[17] Christopher M Bishop. *Neural networks for pattern recognition*. Oxford university press, 1995.

[18] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[19] P.C. Blumbergs, G. Scott, J. Manavis, H. Wainwright, D.A. Simpson, and A.J. McLean. Topography of axonal injury as defined by amyloid precursor protein and the sector scoring method in mild and severe closed head injury. *Journal of Neurotrauma*, 12:565–572, 1995.

[20] Markus Brede. Synchrony-optimized networks of non-identical kuramoto oscillators. *Physics Letters A*, 372(15):2618–2622, 2008.

[21] Steven L Bressler and Anil K Seth. Wiener–granger causality: a well established methodology. *Neuroimage*, 58(2):323–329, 2011.

[22] Steven L Bressler, Wei Tang, Chad M Sylvester, Gordon L Shulman, and Maurizio Corbetta. Top-down control of human visual cortex by frontal and parietal cortex in anticipatory visual spatial attention. *The Journal of Neuroscience*, 28(40):10056–10061, 2008.

[23] K. D. Browne, X. H. Chen, D. F. Meaney, and D. H. Smith. Mild traumatic brain injury and diffuse axonal injury in swine. *Journal of Neurotrauma*, 28(9):1747–1755, 2011.

[24] Ed Bullmore and Olaf Sporns. The economy of brain network organization. *Nature Reviews Neuroscience*, 13:336–349, 2012.

[25] J. Douglas Carroll and Jih-Jie Chang. Analysis of individual differences in multidimensional scaling via an N-way generalization of "Eckart-Young" decomposition. *Psychometrika*, 35:283–319, 1970.

[26] Giorgio Castagneto-Gissey, Mario Chavez, and F De Vico Fallani. Dynamic granger-causal networks of electricity spot prices: A novel approach to market integration. *Energy Economics*, 44:422–432, 2014.

[27] AK Charakopoulos, TE Karakasidis, and A Liakopoulos. Spatiotemporal analysis of seawatch buoy meteorological observations. *Environmental Processes*, 2(1):23–39, 2015.

[28] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In *British Machine Vision Conference*, 2014.

[29] Maoyin Chen, Yun Shang, Yong Zou, and Jürgen Kurths. Synchronization in the kuramoto model: a dynamical gradient network approach. *Physical Review E*, 77(2):027101, 2008.

[30] Y. C. Chen, D. H. Smith, and D.F. Meaney. In-vitro approaches for studying blast-induced traumatic brain injury. *Journal of Neurotrauma*, 26(6):861–876, 2009.

[31] Yonghong Chen, Govindan Rangarajan, Jianfeng Feng, and Mingzhou Ding. Analyzing multiple nonlinear time series with extended granger causality. *Physics Letters A*, 324(1):26–35, 2004.

[32] Eric C. Chi, Genevera I. Allen, Hua Zhou, Omid Kohannim, Kenneth Lange, and Paul M. Thompson. Imaging genetics via sparse canonical correlation analysis. In *Biomedical Imaging (ISBI), 2013 IEEE 10th International Symposium on*, pages 740–743, 2013.

[33] Eric C. Chi and Tamara G. Kolda. On tensors, sparsity, and nonnegative factorizations. *SIAM Journal on Matrix Analysis and Applications*, 33(4):1272–1299, December 2012.

[34] D. B. Chklovskii and A. A. Koulakov. Maps in the brain: what can we learn from them? *Annual Reviews in Neuroscience*, 27:369–392, 2004.

[35] C.W. Christman, M.S. Grady, S.A. Walker, K.L. Hol-Loway, and J.T. Povlishock. Ultra-structural studies of diffuse axonal injury in humans. *Journal of Neurotrauma*, 11:173–186, 1994.

[36] M. Coleman. Axon degeneration mechanisms: commonality amid diversity. *Nature Reviews Neuroscience*, 6(11):889–898, 2005.

[37] Madelaine Daianu, Russell E. Jacobs, Terrence Town, and Paul M. Thompson. Axonal diameter and density estimated with 7-tesla hybrid diffusion imaging in transgenic alzheimer rats. *SPIE Proceedings*, 9784:1–6, 2016.

[38] M. J. del Razo, Y. Morofuji, J. S. Meabon, B. R. Huber, E. R. Peskind, W. A. Banks, P. D. Mourad, R. J. LeVeque, and D. G. Cook. Computational and in vitro studies of blast-induced blood-brain barrier disruption. *SIAM Journal on Scientific Computing*, 38(3):347–374, 2016.

[39] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009.

[40] K. Dikranian, R. Cohen, C. Mac Donald, Y. Pan, D. Brakefield, P. Bayly, and A. Parsadanian. Mild traumatic brain injury to the infant mouse causes robust white matter axonal degeneration which precedes apoptotic death of cortical and thalamic neurons. *Experimental Neurology*, 211:551–560, 2008.

[41] Florian Dörfler and Francesco Bullo. Synchronization in complex networks of phase oscillators: A survey. *Automatica*, 50(6):1539–1564, 2014.

[42] J. Douglas Carroll, Sandra Pruzansky, and Joseph B. Kruskal. Candelinc: A general approach to multidimensional analysis of many-way arrays with linear constraints on parameters. *Psychometrika*, 45(1):3–24, 1980.

[43] Brian L. Edlow, William A. Copen, Saef Izzy, Andre van der Kouwe, Mel B. Glenn, Steven M. Greenberg, David M. Greer, and Ona Wu. Longitudinal diffusion tensor imaging detects recovery of fractional anisotropy within traumatic axonal injury lesions. *Neurocritical Care*, 24(3):342–352, 2016.

[44] Paul Erdős and Alfréd Rényi. On random graphs. *Publicationes Mathematicae Debrecen*, 6:290–297, 1959.

[45] Dumitru Erhan, Yoshua Bengio, Aaron Courville, Pierre-Antoine Manzagol, Pascal Vincent, and Samy Bengio. Why does unsupervised pre-training help deep learning? *Journal of Machine Learning Research*, 11(Feb):625–660, 2010.

[46] N. B. Erichson, J. N. Kutz, S. L. Brunton, and S. Voronin. Randomized matrix decompositions using r. *arXiv preprint arXiv:1608.02148*, 2016.

[47] Luca Faes, Giandomenico Nollo, and Alberto Porta. Information-based detection of nonlinear granger causality in multivariate processes via a nonuniform embedding technique. *Physical Review E*, 83(5):051112, 2011.

[48] M. Fainaru-Wada and S. Fainaru. League of denial: The nfl, concussions, and the battle for truth. *Crown Archetype*, 2013.

[49] M. Faul, L. Xu, M. M. Wald, and V. G. Coronado. Traumatic brain injury in the united states: emergency department visits, hospitalizations, and deaths. *Atlanta (GA): Centers for Disease Control and Prevention, National Center for Injury Prevention and Control*, 2010.

[50] Imran Fayanz and Charles H. Tator. Modeling axonal injury in vitro: injury and regeneration following acute neuritic trauma. *Journal of Neuroscience Methods*, 102:69–79, 2000.

[51] Manuel A. Friese, Benjamin Schattling, and Lars Fugger. Mechanisms of neurodegeneration and axonal dysfunction in multiple sclerosis. *Nature Reviews Neurology*, 10:225–238, 2014.

[52] F. Fukushima. A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetic*, 36:193–202, 1980.

[53] J. E. Galvin, K. Uryu, V. M. Lee, and J. Q. Trojanowski. Axon pathology in parkinson's disease and lewy body dementia hippocampus contains α-, β-, and γ -synuclein. *Proceedings of National Academy of Science*, 96:13450–13455, 1999.

[54] M.S. Grady, M.R. Mclaughlin, C.W. Christman, A.B. Valadaka, C.L. Flinger, and J.T. Povlishock. The use of antibodies against neurofilament subunits for the detection of diffuse axonal injury in humans. *Journal of Neuropathology and Experimental Neurology*, 52:143–152, 1993.

[55] Clive WJ Granger. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica: Journal of the Econometric Society*, 37(3):424–438, 1969.

[56] Clive WJ Granger. Testing for causality: a personal viewpoint. *Journal of Economic Dynamics and control*, 2:329–352, 1980.

[57] Clive WJ Granger. Time series analysis, cointegration, and applications. *American Economic Review*, 94(3):421–425, 2004.

[58] Hayit Greenspan, Bram van Ginneken, and Ronald M Summers. Guest editorial deep learning in medical imaging: Overview and future promise of an exciting new technique. *IEEE Transactions on Medical Imaging*, 35(5):1153–1159, 2016.

[59] Jacob Grosek and J Nathan Kutz. Dynamic mode decomposition for real-time background/foreground separation in video. *arXiv preprint arXiv:1404.7592*, 2014.

[60] Shuixia Guo, Anil K Seth, Keith M Kendrick, Cong Zhou, and Jianfeng Feng. Partial granger causalityeliminating exogenous inputs and latent variables. *Journal of neuroscience methods*, 172(1):79–93, 2008.

[61] Nathan Halko, Per-Gunnar Martinsson, and Joel A Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review*, 53(2):217–288, 2011.

[62] James D Hamilton. Oil and the macroeconomy since world war ii. *The Journal of Political Economy*, 91(2):228–248, 1983.

[63] Anders Hanell, John E. Greer, Melissa J. McGinn, and John T. Povlishock. Traumatic brain injury induced axonal phenotypes react differently to treatment. *Acta Neuropathologica*, 129:317–332, 2015.

[64] Richard A. Harshman. Foundations of the PARAFAC procedure: Models and conditions for an "explanatory" multi-modal factor analysis. *UCLA working papers in phonetics*, 16:1–84, 1970. Available at `http://www.psychology.uwo.ca/faculty/harshman/wpppfac0.pdf`.

[65] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.

[66] Amy N. Hellman, Behrad Vahidi, Hyung Joon Kim, Wael Mismar, Oswald Steward, Noo Li Jeonde, and Vasan Venugopalan. Examination of axonal injury and regeneration in micropatterned neuronal culture using pulsed laser microbeam dissection. *Lab on a Chip*, 16:20832092, 2010.

[67] M.A. Hemphill, B.E. Dabiri, S. Gabriele, L. Kerscher, C. Franck, J.A. Goss, P.W. Alford, and K.K. Parker. A possible role for integrin signaling in diffuse axonal injury. *PLos ONE*, 6(7):e22899, 2011.

[68] M.A. Hemphill, S. Dauth, C. J. Yu, B.E. Dabiri, and K.K. Parker. Traumatic brain injury and the neuronal microenvironment: A potential role for neuropathological mechanotransduction. *Neuron*, 86(6):1177–1192, 2015.

[69] Nils Henninger, James Bouley, Elif M. Sikoglu, Jiyan An, Constance M. Moore, Jean A. King, Robert Bowser, Marc R. Freeman, and Robert H. Brown Jr. Attenuated traumatic axonal injury and improved functional outcome after traumatic brain injury in mice lacking sarm1. *BRAIN*, pages 1–12, 2016.

[70] Marina Herwerth, Sudhakar Reddy Kalluri, Rajneesh Srivastava, Tatjana Kleele, Selin Kenet, Zsolt Illes, Doron Merkler, Jeffrey L. Bennett, Thomas Misgeld, and Bernhard Hemmer. In vivo imaging reveals rapid astrocyte depletion and axon damage in a model of neuromyelitis optica-related pathology. *Annals of Neurology*, 79:794–805, 2016.

[71] Tony Hey, Stewart Tansley, and Kristin M. Tolle, editors. *The Fourth Paradigm: Data-Intensive Scientific Discovery*. Microsoft Research, 2009.

[72] Ciaran S. Hill, Michael P. Coleman, and David K. Menon. Traumatic axonal injury: mechanisms and translational opportunities. *Trends in Neuroscience*, 39(5):311–324, 2016.

[73] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006.

[74] Paul W Holland. Statistics and causal inference. *Journal of the American Statistical Association*, 81(396):945–960, 1986.

[75] Philip Holmes, John L. Lumley, and Gal Berkooz. *Turbulence, coherent structures, dynamical systems, and symmetry*. Cambridge monographs on mechanics. Cambridge University Press, Cambridge, England, 2nd edition, 2012.

[76] Kevin D Hoover. Causality in economics and econometrics. *The new Palgrave dictionary of economics*, 2, 2008.

[77] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

[78] Harold Hotelling. Analysis of a complex of statistical variables with principal components. *Journal of Educational Psychology*, 24:417–441, September 1933.

[79] Harold Hotelling. Analysis of a complex of statistical variables with principal components. *Journal of Educational Psychology*, 24:498–520, October 1933.

[80] Gary B Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical report, Technical Report 07-49, University of Massachusetts, Amherst, 2007.

[81] D. H. Hubel and T. N. Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *Journal of Physiology*, 160:106–154, 1962.

[82] Guido W. Imbens and Donald B. Rubin. *Causal Inference for Statistics, Social, and Biomedical Sciences: An Introduction*. Cambridge University Press, 2015.

[83] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[84] V. E. Johnson, W. Stewart, and D. H. Smith. Axonal pathology in traumatic brain injury. *Experimental Neurology*, 246:35–43, 2013.

[85] Victoria E. Johnson, William Stewart, and Douglas H. Smith. Traumatic brain injury and amyloid-β pathology: a link to alzheimer's disease? *Nature Reviews Neuroscience*, 11:361–370, 2010.

[86] Victoria E. Johnson, William Stewart, and Douglas H. Smith. Widespread tau and amyloid-beta pathology many years after a single traumatic brain injury in humans. *Brain Pathology*, 22:142–149, 2012.

[87] R. E. Jorge, L. Acion, T. White, D. Tordesillas-Gutierrez, R. Pierson, B. Crespo-Facorro, and V.A. Magnotta. White matter abnormalities in veterans with mild traumatic brain injury. *American Journal of Psychiatry*, 169(12):1284–1291, 2012.

[88] A. F. Jorm and D. Jolley. The incidence of dementia: a meta analysis. *Neurology*, 51:728–733, 1998.

[89] Sharunas Jurevic. Green pepper, 2011. [Online; accessed June 1, 2016].

[90] David Kahle and Hadley Wickham. ggmap: Spatial visualization with ggplot2. *The R Journal*, 5(1):144–161, 2013.

[91] M Kaiser and C. C. Hilgetag. Nonoptimal component placement, but short processing paths, due to long- distance projections in neural systems. *PLoS Computational Biology*, 2(e95), 2006.

[92] Pall Karlsson, Simon Haroutounian, Michael Polydefkis, Jens R. Nyengaard, and Troels S. Jensen. Structural and functional characterization of nerve fibres in polyneuropathy and healthy subjects. *Scandinavian Journal of Pain*, 10:28–35, 2016.

[93] Kirsi Maria Kinnunen, Richard Greenwood, Jane Hilary Powell, Robert Leech, Peter Charlie Hawkins, Valerie Bonnelle, Maneesh Chandrakant Patel, Serena Jane Counsell, and David James Sharp. White matter damage and cognitive impairment after traumatic brain injury. *Brain*, pages 1–15, 2010.

[94] Stefan Klus, Patrick Gelß, Sebastian Peitz, and Christof Schütte. Tensor based dynamic mode decomposition. arXiv:1512.06527 [math.NA], 2016.

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

[96] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[97] D. Krstic and I. Knuesel. Deciphering the mechanism underlying late-onset alzheimer disease. *Nature Reviews Neuroscience*, 9(1):25–34, 2012.

[98] Yoshiki Kuramoto. Self-entrainment of a population of coupled non-linear oscillators. In *International symposium on mathematical problems in theoretical physics*, pages 420–422. Springer, 1975.

[99] J. Nathan Kutz, Xing Fu, and Steven L. Brunton. Multiresolution dynamic mode decomposition. *SIAM Journal on Applied Dynamical Systems*, 15(2):713–735, 2016.

[100] Valliappa Lakshmanan, Eric Gilleland, Amy McGovern, and Martin Tingley, editors. *Machine Learning and Data Mining Approaches to Climate Science*, Proceedings of the 4th International Workshop on Climate Informatics. Springer, 2015.

[101] Simon B. Laughlin and Terrence Sejnowski. Communication in neuronal networks. *Science*, 301:1870–1874, 2003.

[102] Jeremy J. Laukka, John Kamholz, and Denise Bessert. Novel pathologic findings in patients with pelizaeus-merzbacher disease. *Neuroscience Letters*, 2016.

[103] G. Lauria, M. Morbin, R. Lombardi, M. Borgna, G. Mazzoleni, A. Sghirlanzoni, and D. Pareyson. Axonal swellings predict the degeneration of epidermal nerve fibers in painful neuropathies. *Neurology*, 61:631–636, 2003.

[104] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[105] Christian LoBue, David Denney, Linda S. Hynan, Heidi C. Rossetti, Laura H. Lacritz, John Hart Jr., Kyle B. Womack, Fu L. Woon, and C. Munro Cullum. Self-reported traumatic brain injury and mild cognitive impairment: increased risk and earlier age of diagnosis. *Journal of Alzheimer's Disease*, 51:727–736, 2016.

[106] Edward N. Lorenz. Empirical orthogonal functions and statistical weather prediction. Technical report, Massachusetts Institute of Technology, December 1956.

[107] Elan D. Louis, Phyllis L. Faust, J.P.G. Vonsattel, Lawrence S. Honig, Alex Rajput, Ali Rajput, Rajesh Pahwa, Kelly E Lyons, G. Webster Ross, Rodger J. Elble, Cordelia Erickson-Davis, Carol B. Moskowitz, and Arlene Lawton. Torpedoes in parkinson's disease, alzheimer's disease, essential tremor, and control brains. *Movement Disorders*, 24(11):1600–1605, 2009.

[108] John L. Lumley. *Stochastic tools in turbulence*. Applied mathematics and mechanics. Academic press, New York, London, 1970.

[109] Helmut Lütkepohl. *New introduction to multiple time series analysis*. Springer Science & Business Media, 2005.

[110] M. H. Magdesian, F.S. Sanchez, M. Lopez, P. Thostrup, N. Durisic, W. Belkaid, D. Liazoghli, P. Grütter, and R. Colman. Atomic force microscopy reveals important differences in axonal resistance to injury. *Biophysical Journal*, 103(3):405–414, 2012.

[111] Pedro D. Maia, Matthew A. Hemphill, Brendan Zehnder, Chenfei Zhang, Kevin K. Parker, and J. Nathan Kutz. Diagnostic tools for evaluating the impact of focal axonal swellings arising in neurodegenerative diseases and/or traumatic brain injury. *Journal of Neuroscience Methods*, 253:233–243, 2015.

[112] Pedro D. Maia and J. Nathan Kutz. Compromised axonal functionality after neurodegeneration, concussion and/or traumatic brain injury. *Journal of Computational Neuroscience*, 27:317–332, 2014.

[113] Pedro D. Maia and J. Nathan Kutz. Identifying critical regions for spike propagation in axon segments. *Journal of Computational Neuroscience*, 36(2):141–155, 2014.

[114] Pedro Doria Maia. *Mathematical modeling of focal axonal swellings arising in traumatic brain injuries and neurodegenerative diseases.* PhD thesis, University of Washington, 2014.

[115] Daniele Marinazzo, Mario Pellicoro, and Sebastiano Stramaglia. Kernel method for nonlinear granger causality. *Physical Review Letters*, 100(14):144103, 2008.

[116] W. L. Maxwell, J. T. Povlishock, and D. L. Graham. A mechanistic analysis of nondisruptive axonal injury: A review. *Journal of Neurotrauma*, 17(7):419–440, 1997.

[117] Colleen McCue. *Data mining and predictive analysis: Intelligence gathering and crime analysis*. Butterworth-Heinemann, 2014.

[118] David K. Menon and Andrew I. R. Maas. Progress, failures and new approaches for tbi research. *Nature Reviews Neuroloy*, 11:71–72, 2015.

[119] S. Millecamps and J.P. Julien. Axonal transport deficits and neurodegenerative diseases. *Nature Reviews Neuroscience*, 14(161):161–176, 2013.

[120] Tom M Mitchell. *Machine learning*. McGraw Hill, 1997.

[121] Stephen L. Morgan and Christopher Winship. *Counterfactuals and Causal Inference: Methods and Principles for Social Research, 2nd Ed.* Cambridge University Press, 2015.

[122] Barclay Morrison, Benjamin S. Elkin, Jean Pierre Dolle, and Martin L. Yarmush. In vitro models of traumatic brain injury. *Annual Reviews in Biomedical Engineering*, 13(1):91–126, 2011.

[123] Kevin P. Murphy. *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012.

[124] Mark EJ Newman. The structure and function of complex networks. *SIAM review*, 45(2):167–256, 2003.

[125] Ivana Nikic, Doron Merkler, Catherine Sorbara, Mary Brinkoetter, Mario Kreutzfeld, Florence Bareyre, Wolfgang Bruck, Derron Bishop, Thomas Misgeld, and Martin Kerschensteiner. A reversible form of axon damage in experimental autoimmune encephalomyelitis and multiple sclerosis. *Nature Medicine*, 17(4):495–499, 2011.

[126] Xiaoke Niu, Li Shi, Hong Wan, Zhizhong Wang, Zhigang Shang, and Zhihui Li. Dynamic functional connectivity among neuronal population during modulation of extraclassical receptive field in primary visual cortex. *Brain research bulletin*, 117:45–53, 2015.

[127] Partha Niyogi, Federico Girosi, and Tomaso Poggio. Incorporating prior information in machine learning by creating virtual examples. *Proceedings of the IEEE*, 86(11):2196–2209, 1998.

[128] Guido Nolte, Andreas Ziehe, Nicole Krämer, Florin Popescu, and Klaus-Robert Müller. Comparison of granger causality and phase slope index. In *NIPS Causality: Objectives and Assessment*, pages 267–276. Citeseer, 2010.

[129] Bruno A. Olshausen and David J. Field. Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision Research*, 37(23):3311 – 3325, 1997.

[130] Omkar M. Parkhi, Andrea Vedaldi, and Andrew Zisserman. Deep face recognition. In Mark W. Jones Xianghua Xie and Gary K. L. Tam, editors, *Proceedings of the British Machine Vision Conference (BMVC)*, pages 41.1–41.12. BMVA Press, September 2015.

[131] Bruce W. Patterson, Donald L. Elbert, Kwasi G. Mawuenyega, Tom Kasten, Vitaliy Ovod, Shengmei Ma, Chengjie Xiong, Robert Chott, Kevin Yarasheski, Wendy Sigurdson, Lily Zhang, Alison Goate, Tammie Benzinger, John C. Morris, David Holtzman, and Randall J. Bateman. Age and amyloid effects on human central nervous system amyloid-beta kinetics. *American Neurological Association*, 78(3):439–453, 2015.

[132] Judea Pearl. *Causality: Models, Reasoning and Inference, 2nd Ed.* Cambridge University Press, 2009.

[133] Karl Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine Series 6*, 2(11):559–572, 1901.

[134] V. Pernice, B. Staude, S. Cardanobile, and S. Rotter. How structure determines correlations in neuronal networks. *PLOS Comp. Bio.*, 7(5):e1002059, 2011.

[135] T. Poggio. Deep learning: mathematics and neuroscience. *Views & Reviews, McGovern Center for Brains, Minds and Machines*, pages 1–7, 2016.

[136] John T. Povlishock and Douglas I. Katz. Update of neuropathology and neurological recovery after traumatic brain injury. *Journal of Head Trauma Rehabilitation*, 20(1):76–94, 2005.

[137] Chengxuan Qiu, Miia Kivipelto, and Eva von Strauss. Epidemiology of alzheimer's disease: occurrence, determinants, and strategies toward intervention. *Dialogues in Clinical Neuroscience*, 11(2):111–128, 2009.

[138] Bob Roozenbeek, Andrew I. R. Maas, and David K. Menon. Changing patterns in the epidemiology of traumatic brain injury. *Nature Reviews Neurology*, 9:231–236, 2013.

[139] Koichi Sameshima, Daniel Y Takahashi, and Luiz A Baccalá. On the statistical performance of granger-causal connectivity estimators. *Brain Informatics*, pages 1–15, 2015.

[140] Arthur L Samuel. Some studies in machine learning using the game of checkers. *IBM Journal of research and development*, 3(3):210–229, 1959.

[141] Terence D Sanger. Optimal unsupervised learning in a single-layer linear feedforward neural network. *Neural networks*, 2(6):459–473, 1989.

[142] L Schiatti, G Nollo, G Rossato, and L Faes. Extended granger causality: a new tool to identify the structure of physiological networks. *Physiological measurement*, 36(4):827, 2015.

[143] Gideon Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6(2):461–464, 03 1978.

[144] Anil K Seth. A matlab toolbox for granger causal connectivity analysis. *Journal of neuroscience methods*, 186(2):262–273, 2010.

[145] David J. Sharp, Gregory Scott, and Robert Leech. Network dysfunction after traumatic brain injury. *Nature Reviews Neurology*, 10:156–166, 2014.

[146] S. Shimizu, P. Hoyer, A. Hyvärinen, and A. Kerminen. A linear non-gaussian acyclic model for causal discovery. *J. Machine Learning Research*, 7:2003–2030, 2006.

[147] Toril Skandsen, Kjell Arne Kvistad, Ole Solheim, Ingrid Haavde Strand, Mari Folvik, and Anne Vik. Prevalence and impact of diffuse axonal injury in patients with moderate and severe head injury: a cohort study of early magnetic resonance imaging findings and 1-year outcome. *Journal of Neurosurgery*, 113(3):556–563, 2010.

[148] D.H. Smith, J.W. Wolf, T.A. Lusardi, V.M.Y. Lee, and D.F. Meaney. High tolerance and delayed elastic response of cultured axons to dynamic stretch injury. *The Journal of Neuroscience*, 19(11):4263–4269, 1999.

[149] Olaf Sporn. *Networks of the brain*. MIT Press, 2011.

[150] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[151] Patrick A Stokes. *Fundamental problems in Granger causality analysis of neuroscience data*. PhD thesis, Massachusetts Institute of Technology, 2015.

[152] G. Sugihara, R. May, H. Ye, C. Hsieh, E. Deyle, M. Fogarty, and S. Munch. Detecting causality in complex ecosystems. *Science*, 338:496–500, 2012.

[153] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.

[154] Patricia Tagliaferro and Robert E. Burke. Retrograde axonal degeneration in parkinson disease. *Journal of Parkinson's Disease*, 6:1–15, 2016.

[155] M. D. Tang-Schomer, V. E. Johnson, P. W. Baas, W. Stewart, and D. H. Smith. Partial interruption of axonal transport due to microtubule breakage accounts for the formation of periodic varicosities after traumatic axonal injury. *Experimental Neurology*, 233:364–372, 2012.

[156] M. D. Tang-Schomer, A.R. Patel, P. W. Bass, and D. H. Smith. Mechanical breaking of microtubules in axons during dynamic stretch injury underlies delayed elasticity, microtubule disassembly, and axon degeneration. *The FASEB Journal*, 24(5):1401–1410, 2010.

[157] W. Thies and L. Bleiler. Alzheimer's disease facts and figures. *Alzheimer's & Dementia*, 9(2):208–245, 2013.

[158] Bruce D Trapp and Klaus-Armin Nave. Multiple sclerosis: An immune or neurodegenerative disorder? *Annual Review Neuroscience*, 31(1):247–269, 2008.

[159] J. Tsai, J. Grutzendler, K. Duff, and W. B. Gan. Fibrillar amyloid deposition leads to local synaptic abnormalities and breakage of neuronal branches. *Nature Neuroscience*, 7:1181–1183, 2004.

[160] Jonathan H. Tu, Clarence W. Rowley, Dirk M. Luchtenburg, Steven L. Brunton, and J. Nathan Kutz. On dynamic mode decomposition: Theory and applications. *Journal of Computational Dynamics*, 1(2):391–421, 2014.

[161] A. Vedaldi and K. Lenc. Matconvnet – convolutional neural networks for matlab. In *Proceeding of the ACM Int. Conf. on Multimedia*, 2015.

[162] B. Wahl, U. Feudel, J. Hlinka, M. Wächter, J. Peinke, and J. Freund. Granger-causality maps of diffusion processes. *Phys. Rev. E*, 93:022213, 2016.

[163] J. Wang, R. J. Hamm, and J. T. Povlishock. Traumatic axonal injury in the optic nerve: evidence for axonal swelling, disconnection, dieback and reorganization. *Journal of Neurotrauma,*, 28(7):1185–1198, 2011.

[164] Xue Wang, Yonghong Chen, Steven L Bressler, and Mingzhou Ding. Granger causality between multiple interdependent neurobiological time series: blockwise versus pairwise methods. *International journal of neural systems*, 17(02):71–78, 2007.

[165] Larry Wasserman. Statistics versus machine learning. `https://normaldeviate.wordpress.com/2012/06/12/statistics-versus-machine-learning-5-2/`, 2012.

[166] Hadley Wickham. *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York, 2009.

[167] Norbert Wiener. The theory of prediction. *Modern mathematics for engineers*, 1:125–139, 1956.

[168] Daniela M. Witten, Robert Tibshirani, and Trevor Hastie. A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis. *Biostatistics*, 10(3):515–534, 2009.

[169] Guorong Wu, Xujun Duan, Wei Liao, Qing Gao, and Huafu Chen. Kernel canonical-correlation granger causality for multiple time series. *Physical Review E*, 83(4):041921, 2011.

[170] Xiaoqun Wu, Weihan Wang, and Wei Xing Zheng. Inferring topologies of complex networks with hidden variables. *Physical Review E*, 86(4):046106, 2012.

[171] Xiaoqun Wu, Changsong Zhou, Guanrong Chen, and Jun-an Lu. Detecting the topologies of complex networks with stochastic perturbations. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 21(4):043129, 2011.

[172] He Xu, Eleni Kroupi, and Touradj Ebrahimi. Functional connectivity from eeg signals during perceiving pleasant and unpleasant odors. In *Affective Computing and Intelligent Interaction (ACII), 2015 International Conference on*, pages 911–916. IEEE, 2015.

[173] Fan Yang, Ping Duan, Sirish L Shah, and Tongwen Chen. *Capturing connectivity and causality in complex industrial processes*. Springer Science & Business Media, 2014.

[174] Dov Yellin, Aviva Berkovich-Ohana, and Rafael Malach. Coupling between pupil fluctuations and resting-state fmri uncovers a slow build-up of antagonistic responses in the human cortex. *NeuroImage*, 106:414–427, 2015.

[175] Edward Zagha, Xinxin Ge, and David A McCormick. Competing neural ensembles in motor cortex gate goal-directed motor output. *Neuron*, 88(3):565–577, 2015.

[176] Bo Zong, Yinghui Wu, Jie Song, Ambuj K Singh, Hasan Cam, Jiawei Han, and Xifeng Yan. Towards scalable critical alert mining. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1057–1066. ACM, 2014.