LEARNING SPATIO-TEMPORAL DYNAMICS

Nonparametric Methods for Optimal Forecasting and Automated Pattern Discovery

A dissertation submitted in partial fulfillment of

the requirements for the degree of

DOCTOR OF PHILOSOPHY

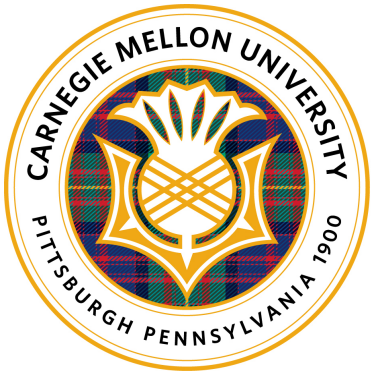
IN

STATISTICS

by

GEORG MATTHIAS GOERG

Department of Statistics
Carnegie Mellon University
Pittsburgh, PA 15213



December 2012

ADVISORS

Cosma Shalizi

Larry Wasserman

COMMITTEE

Christopher Genovese

Cristopher Moore

Chad Schafer

ABSTRACT

Many important scientific and data-driven problems involve quantities that vary over space and time. Examples include functional magnetic resonance imaging (fMRI), climate data, or experimental studies in physics, chemistry, and biology.

Principal goals of many methods in statistics, machine learning, and signal processing are to use this data and i) extract informative structures and remove noisy, uninformative parts; ii) understand and reconstruct underlying spatio-temporal dynamics that govern these systems; and iii) forecast the data, i.e., describe the system in the future.

Being data-driven problems, it is important to have methods and algorithms that work well in practice for a wide range of spatio-temporal processes as well as various data types. In this thesis I present such generally applicable statistical methods that address all three problems in a unifying manner.

I introduce two new techniques for optimal nonparametric forecasting of spatio-temporal data: *hard* and *mixed LICORS* (Light Cone Reconstruction of States). Hard LICORS is a consistent predictive state estimator and extends previous work from Shalizi (2003); Shalizi, Haslinger, Rouquier, Klinkner, and Moore (2006); Shalizi, Klinkner, and Haslinger (2004) to continuous-valued spatio-temporal fields. Mixed LICORS builds on a new, fully probabilistic model of light cones and predictive states mappings, and is an EM-like version of hard LICORS. Simulations show that it has much better finite sample properties than hard LICORS. I also propose a sparse variant of mixed LICORS, which improves out-of-sample forecasts even further.

Both methods can then be used to estimate local statistical complexity (LSC) (Shalizi, 2003), a fully automatic technique for pattern discovery in dynamical systems. Simulations and applications to fMRI data demonstrate that the proposed methods work well and give useful results in very general scientific settings.

Lastly, I made most methods publicly available as R (R Development Core Team, 2010) or Python (Van Rossum, 2003) packages, so researchers can use these methods and better understand, forecast, and discover patterns in the data they study.

RELATED PUBLICATIONS

Some ideas and figures have appeared in:

Goerg, Merriam, Genovese, and Shalizi (2012)

Goerg and Shalizi (2012a)

Goerg and Shalizi (2012b)

Software implementations of the methods and estimators presented in this thesis are publicly available (see Appendix B for details). Manuals are available at:

Goerg (2012a)

Goerg (2012b)

Goerg (2012c)

ACKNOWLEDGMENTS

First and foremost, I want to thank Cosma Shalizi and Larry Wasserman for being great advisors. Their excellent guidance, challenging questions, many suggestions, and continuous support provided the basis for the successful completion of my thesis. Not only did they give me ample freedom to pursue the thesis research, but also to do completely independent work. Special thanks go to Cosma for suggesting a research topic that perfectly fitted my interests and was always exciting and fun to work on.

During my graduate studies many faculty members have been inspiring and helping me in research and life: either by giving advice, teaching excellent courses, or by simply making a comment that changed my views on statistics, machine learning, or research in general. I especially want to thank Jay Kadane, Jiashun Jin, Aarti Singh, and Steve Fienberg. I also thank Chris Genovese, Cris Moore, and Chad Schafer for their interest and questions about my work, and for serving on my committee. I am grateful to Kathryn Roeder, who recommended me to Google for an internship – a great experience that I would not have had without her kind words about me. I also want to thank Maria Kurnikova and Igor Kurnikov for collaborating on the ADA project, for pushing me to step outside my math and statistics comfort zone, and thus showing me challenges and rewards of working on interdisciplinary research.

Of course the last couple of years would not have been the same without the friends I made here. Big thanks go to the six strangers I met on day one of the

PhD – Li, Luba, Nathan, Wanjie, Xiaolin, and Zach – and who made it through the first year with me; to Stefa for hosting me during my first weeks in Pittsburgh and always helping out when I – bluntly – asked for it; to Shuhei and Felipe for their advice when difficult decisions had to be made; to Bruno, Jessica, Mauricio, Bassi, Rafael and Lea, who can still take me seriously after all the laughs and good times we spent being silly; and to Salwa for making the time here especially worthwhile.

Finally, I want to thank my family for their support, advice, and encouragement – even more so when being away on the other side of the Atlantic was not always easy.

Pittsburgh, Pennsylvania
December 2012

CONTENTS

LIST OF FIGURES

ACRONYMS AND NOTATION

FLC	Future Light Cone
KDE	Kernel Density Estimation / Estimator
LC	Light Cone
LICORS	LIght COne Reconstruction of States
LSC	Local Statistical Complexity
MLE	Maximum Likelihood Estimation / Estimator
MSE	Mean Squared Error
pdf	probability density function
PLC	Past Light Cone
RV	Random Variable
wKDE	weighted KDE

$\mathbf{S}$	set of space coordinates; usually an N-dimensional lattice
$\mathbb{T}$	set of time coordinates; usually $1, \ldots, T$
$(\mathbf{r}, t)$	space-time coordinate $\mathbf{r} \in \mathbf{S}$, $t \in \mathbb{T}$
$X(\mathbf{r}, t)$	stochastic process X at $(\mathbf{r}, t)$
$\ell^+(\mathbf{r}, t)$	FLC configuration at $(\mathbf{r}, t)$
$\ell^-(\mathbf{r}, t)$	PLC configuration at $(\mathbf{r}, t)$
$\mathcal{C}(\mathbf{r}, t)$	LSC at $(\mathbf{r}, t)$
$S(\mathbf{r}, t)$	predictive state at $(\mathbf{r}, t)$
$\epsilon(\ell^-(\mathbf{r}, t))$	predictive state mapping of PLC at $(\mathbf{r}, t)$
$\mathcal{S}$	set of predictive states $\mathcal{S} = \{S_1, \ldots, S_K\}$

Für Wilhelm

Part I

INTRODUCTION

1

MOTIVATION

My interest is in the future because I am going to spend the rest of my life there.

Charles F. Kettering

Many important scientific and data-driven problems involve quantities which vary over both space and time. Examples include functional magnetic resonance imaging (fMRI), climate data, or experimental studies in physics, chemistry, and biology.

Figure 1.1 displays three examples of spatio-temporal data: (a) is a simulated one-dimensional field (space extends vertically) observed over time (left to right), which I use as a running example at several points throughout the thesis; (b) is a snapshot of an fMRI scan - data that I use to demonstrate the usefulness of the presented methods for experimental research, in particular neuro-science; and (c) is an easy to understand toy-example, yet it highlights the main challenges in the understanding and modeling of spatio-temporal data.

While those exemplary datasets come from three entirely different systems, they all exhibit non-trivial spatial as well as temporal dependence. That is, events that

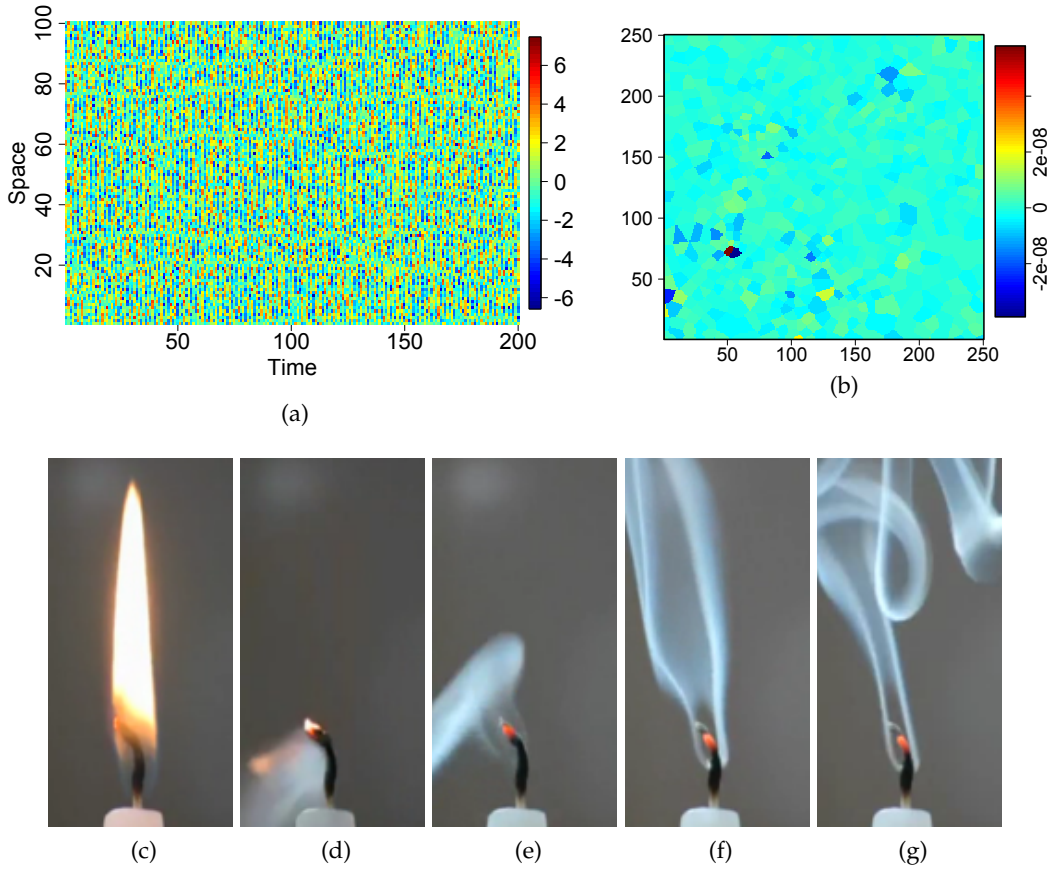


Figure 1.1: Examples of spatio-temporal fields: (a) simulated field where the one-dimensional space coordinate is vertical and time runs from left to right (Section 5.4); (b) fMRI snapshot of part of the visual cortex (Section 8.2); (c) - (g) video snapshots of a candle in the wind (Section 8.1).

happen at location A at time t_1 will most likely have an influence on what happens at location B at $t_2 > t_1$. For example, the field in Fig. 1.1a has green horizontal traces that propagate over time from left to right. Similarly the burning/smoking candle in Fig. 1.1c – 1.1g has clear temporal and spatial structure. These patterns and how one follows the other, illustrate what we mean by spatio-temporal dependence.

It is this dependence that many methods in statistics, machine learning, and signal processing try to use to

1) extract informative structures and remove noisy, uninformative parts;

2) understand and reconstruct underlying spatio-temporal dynamics that govern these systems; and

3) forecast the data, i.e., describe the system in the future.

Being data-driven problems, it is important to have methods and algorithms that work well for a wide range of spatio-temporal problems as well as different data types. It would be very useful to have an automated method that can extract informative patterns, learn important dynamics, and also produce optimal forecasts for any type of dynamical system - without prior knowledge about it.

In this thesis, I present such a generally applicable, statistical methodology and estimators that address all three challenges in a unifying approach.

1.1 PATTERN DISCOVERY

In pattern recognition or anomaly detection one wants to detect "interesting" or unusual features (Chandola, Banerjee, and Kumar, 2009), e.g., activity bursts in fMRI scans (Merriam, Genovese, and Colby, 2007).

A large body of literature in statistics, machine learning, and signal processing (see Section 2.3) is dedicated to developing new methods and algorithms to find such patterns. While they work extremely well in a wide-range of applications, they often do so because they already *know* what to look for: someone has decided beforehand what is interesting and what not. When properties of the data do not align with conditions on the algorithm anymore, they often break down and a new algorithm has to be developed for the new interesting situation. For example, in many brain imaging studies, the experimentalist has full control over the shape and intensity of the stimulus; one can then often use a "matched filter" (or "template matching") technique to detect the response to the stimulus in fMRI data (Kruggel, Cramon, and Descombes, 1999; Marchini and Ripley, 2000). For a different stimulus

such hand-crafted methods must be modified to match the new stimulus structure or distribution. This is not only time consuming, but also limited to situations where the stimulus is known. If the shape of the stimulus is unknown - as often is the case in real-world settings - it is impossible to design such a filter.

For (upcoming) applications it would be very beneficial to have an algorithm that tells us automatically - without any user input - what we should concentrate on and what we can ignore, no matter if we analyze a sound recording or an image, study weather patterns in satellite images, or brain activity in fMRI data.

Optimally, a measure of *interestingness* should reflect our intuition of how these processes evolve over time and in space. If we have such a measure then we can focus on parts of space-time where this measure suddenly increases (decreases), indicating that something interesting has happened (or stopped to happen) there. For example, watching an entire video of the burning candle in Fig. 1.1c will become boring after a while; however, it becomes much more interesting when someone blows it out (Fig. 1.1d) and smoke starts to rise and forms swirls (Fig. 1.1e – 1.1g). An appropriate *interestingness* measure should reflect this increase in information.

In Chapter 4, I present such a statistical measure of complexity and illustrate it on the simulated dataset in Fig. 1.1a. In Chapter 8 I apply it to the candle video and to various fMRI datasets in order to detect active brain regions.

1.2 LEARNING SPATIO-TEMPORAL DYNAMICS FROM DATA

Another important goal is to understand the underlying dynamics that give rise to the patterns we see in the data. For example, the weather is a spatio-temporal system that we would like to understand better - in particular to improve forecasts. Using a physical approach one would formulate a model based on our scientific understanding of climate and weather, describe all relationship between the involved

physical processes using high-dimensional differential equations, and then analyze this model.

A purely statistical approach - like the one we put forward here - would use the immense amount of weather data we have gathered over the last centuries - and especially decades-, and then learn these dynamics from the data. For example, an algorithm can learn that dark clouds will likely cause rain, or that smoke (heat) rises. For weather or a flickering candle, such a statistical approach is rather complimentary. However, for not well-understood systems, e.g., information processing in the brain or disease spread on a social network, a statistical approach that discovers physical, bio-chemical, or socio-economic "laws" from observed data can be an invaluable tool in scientific research.

As a particularly valuable consequence of being able to learn generative dynamics, consider the challenge of simulating new data with the same spatio-temporal properties as observed data. Put in other words: what if we only had experimental data without knowing the underlying mechanisms that generated it; could we simulate another realization of the same process using *only* the observed data?

For example, researchers often run many experiments with different starting conditions to gain a better understanding of the processes they study. It would be very useful to simulate - on a computer - how different starting conditions influence the system, *without* having to know its exact mathematical model. Optimally one would run a small number of experiments, let the algorithm learn the spatio-temporal dynamics, and then use these estimates to simulate new experiments on a computer without having to set up and run expensive, time-consuming and labor-intensive experiments for each single starting condition.

The answer to this question raised above is yes, since our predictive state estimates and optimal forecasts are distribution-based and can therefore also be used

for simulations. In Section 6.5 I show how to simulate from estimated dynamics and Fig. 7.6 illustrates the accuracy of these observation-based simulations.

1.3 FORECASTING

If a system cannot be influenced directly, then we would at least like to predict what its future will look like based on the data we have gathered. This is especially important for systems that do not follow clear, well-understood mechanism, e.g., socio-economic dynamics, where it is necessary to make forecasts based on historical data.

But even for well-understood physical processes, probabilistic forecasts can offer advantages. For example, to obtain weather forecasts based on a physical climate and weather model, it is typically necessary to numerically solve the model forward in time for slightly different starting positions and then make a forecast based on an aggregated output of the model. Solving these high-dimensional models is very time-consuming and must be repeated for every single starting condition. That is, if we want to forecast the weather for tomorrow based on today, we have to run the same lengthy computations as we did to forecast the weather today from yesterday.

On the contrary, estimation and forecasting in a statistical model are separate. While it might take a long time to train the model, once we have an estimate we can almost instantaneously obtain probabilistic forecasts for a great number of different starting conditions.

In Chapter 3, I show how to construct provably optimal predictors in spatio-temporal processes. In Chapters 5 & 7, I then propose two methods to estimate these optimal forecasts from continuous-valued datasets, such as the ones presented in Fig. 1.1.

2

OVERVIEW

Wie dieses Buch zu lesen sei, um möglicherweise verstanden werden zu können, habe ich hier anzugeben mir vorgesetzt. Was durch dasselbe mitgetheilt werden soll, ist ein einziger Gedanke. Dennoch konnte ich, aller Bemühungen ungeachtet, keinen kürzeren Weg ihn mitzutheilen finden, als dieses ganze Buch.

Arthur Schopenhauer

2.1 THESIS OUTLINE

The whole is more than the sum of its parts.

Aristotle, Metaphysica

This thesis is composed of five parts.

Part I motivates the type of problems we study (Chapter 1), outlines the structure of the thesis, and gives an overview of previous work (Chapter 2).

Part II formally defines the prediction problem and presents light cones and predictive states (Shalizi, 2003) as an optimal way to do forecasting (Chapter 3). Using these optimal forecasts, I then describe local statistical complexity (LSC) (Shalizi, 2003; Shalizi et al., 2004), a measure of complexity in spatio-temporal fields (Chapter 4).

Part III contains the main results:

CHAPTER 5: LICORS, a consistent predictive state estimator for continuous-valued data.

CHAPTER 6: A statistical model for light cones and predictive states, which embeds previous work by Shalizi et al. in a fully probabilistic setting. This model enables us to obtain probabilistic forecasts and to make observation-based simulations of spatio-temporal processes.

CHAPTER 7: Mixed LICORS, a nonparametric EM(-like) version of hard LICORS with largely improved out-of-sample prediction performance.

CHAPTER 8: Applications of LSC to functional magnetic resonance imaging (fMRI) data, where we automatically detect active brain regions in very different experiments and thus demonstrate the generality of the proposed methods.

In Part IV, I discuss directions for future research and summarize the contributions of the thesis (Chapter 9 & 10).

Finally, Part V presents proofs of main results (Appendix A), references to publicly available software[1] implementations of the presented methods (Appendix B), and references to data sources (Appendix C).

2.2 MAIN RESULTS AND CONTRIBUTIONS

2.2.1 *LICORS: Light Cone Reconstruction of States*

In Chapter 5, I introduce a new, nonparametric forecasting method for data where continuous values are observed discretely in space and time. The method, *light-cone reconstruction of states* (LICORS), uses physical principles to identify predictive states which are local properties of the system, both in space and time. LICORS discovers the number of predictive states and their predictive distributions automatically, and consistently, under mild assumptions on the data source. We provide an algorithm to implement our method, along with a cross-validation scheme to pick control settings. Simulations show that CV-tuned LICORS outperforms standard methods in forecasting challenging spatio-temporal dynamics. This estimator provides applied researchers with a new, highly automatic method to analyze and forecast spatio-temporal data.

2.2.2 *A Statistical Predictive State Model for Spatio-temporal Processes*

Previous work by Shalizi *et al.* was mostly motivated by prediction and pattern recognition problems in discrete mathematics, physics, and information theory. While forecasting in itself is important, many real-world spatio-temporal systems pose interesting questions, which are not directly related to forecasting. For ex-

1 All computations were either done in R (R Development Core Team, 2010) or Python (Van Rossum, 2003). See Appendix B for more details.

ample, pattern discovery in experimental data, classification of spatio-temporal dynamics, estimating dependence between spatio-temporal systems in sociology, public health, biology, economics, etc. In Chapter 6, I embed the predictive state space framework into a fully probabilistic setting, and thus provide the basis for optimal statistical inference on spatio-temporal data.

2.2.3 *Mixed LICORS*

Based on this statistical model in Chapter 7 I introduce *mixed LICORS*, a nonparametric EM-like predictive state space estimator. Mixed LICORS is a soft-thresholding generalization of (hard) LICORS from Chapter 5. I also propose a generally applicable penalization technique to obtain sparse mixture weights in EM algorithms. Simulations show that mixed LICORS has much better finite sample properties than hard LICORS, and that sparsification avoids overfitting and even further improves the out-of-sample predictive power. Furthermore, mixed LICORS estimates can be used for observation-based simulations.

2.2.4 *Pattern Discovery in fMRI data*

Shalizi (2003) introduced local statistical complexity (LSC) as a measure to detect interesting and important events in a discrete-valued spatio-temporal field. The underlying idea is that statistically optimal predictors not only predict well but - for this very reason - also reveal inherent dynamic structure in the data. The advantage of pattern *discovery* by LSC compared to traditional pattern *recognition* techniques is that researchers do not have to know what is interesting beforehand; LSC detects informative areas in space-time automatically.

Using the nonparametric LICORS estimator, I extend the methods and algorithms to continuous-valued processes, which allows researchers to apply LSC to experimental data. In Chapter 8, we apply LSC to high-resolution fMRI data, which gives an "interestingness" score for each voxel at each moment in time. Traditional techniques often rely on prior knowledge of the stimulus to hand-craft algorithms that look for pre-defined structures in the observed data ("template matching"). Applications to various fMRI datasets demonstrate that LSC can automatically detect brain activity of highly irregular spatio-temporal patterns.

LSC can therefore become an unparalleled tool for applied researchers - in any field working with spatio-temporal data - to detect important structures in, yet unknown, dynamical systems.

2.3 LITERATURE REVIEW (AKA PAST WORK CONE)

Spatio-temporal data being increasingly easy to acquire, manipulate, and visualize, statisticians have developed methods for statistical inference, reviewed in works like Cressie and Wikle (2011); Finkenstädt, Held, and Isham (2007). The usual tools are a combination of ways of describing the distribution of the random field (e.g., various dependency measures), and stochastic modeling, focusing primarily on parametric inference, and secondarily on parameter-conditional predictions.

For plain forecasting, classic time series approaches focus on second order stationary processes, with the particularly popular (vector) auto-regressive models or - more general - state space models to produce optimal forecasts (Brockwell and Davis, 1991; Hamilton, 1994). In the machine learning / signal processing literature, hidden Markov models (HMM) are commonly used (Cappé, Moulines, and Rydén, 2005). While these approaches are valuable, there is a complementary role for direct, nonparametric prediction of spatio-temporal data, just as with time series (Bosq, 1998; Fan and Yao, 2003).

For the spatio-temporal setting we will use predictive state recovery as our basis methodology.

2.3.1 *Predictive State Space Reconstruction, Forecasting*

Predictive state reconstruction estimates the prediction processes introduced by Knight (1975). Knight's construction is for stochastic processes X with a single, continuous time index; but since X_t can take values in infinite-dimensional spaces, most useful spatial models can implicitly be handled in this way, and by considering discrete time we avoid many measure-theoretic complications. After Knight, the same basic construction of the prediction process was independently rediscovered in nonlinear dynamics and physics (Crutchfield and Young, 1989; Shalizi and Crutchfield, 2001), in machine learning (Jaeger, 2000; Langford, Salakhutdinov, and Zhang, 2009; Littman, Sutton, and Singh, 2002), and in the philosophy of science (Salmon, 1984).

Spatio-temporally local prediction processes were introduced in Shalizi (2003); Shalizi et al. (2004) to study self-organization and system complexity, along lines suggested by Crutchfield and Young (1989); Grassberger (1986). A related proposal was made by Parlitz and Merkwirth (2000), and light cones have been used in stochastic models of crystallization (Capasso and Micheletti, 2002), going back to Kolmogorov (1937).

While the prediction-process formalism allows for continuous-valued observable fields, prior work by Shalizi *et al.* only gave algorithms for discrete-valued fields. Jänicke *et al.* used those procedures for continuous-valued fields by discretizing the data (Jänicke, 2009; Jänicke and Scheuermann, 2010; Jänicke, Wiebel, Scheuermann, and Kollmann, 2007). It is important to point out that this discretization is not necessary for the theory or methodology to work, but is only done for statistical convenience since estimation for discrete-valued random variables often reduces to simple counting.

One of the main contributions of this thesis is the development of nonparametric statistical methods that can accurately estimate the predictive states from continuous-valued data without such a prior discretization step.

2.3.2 *Pattern Discovery And Complexity*

Local statistical complexity was introduced only recently (Shalizi, 2003) and thus prior literature is fairly sparse. Shalizi et al. (2006, 2004) use the LSC methodology and present algorithms to estimate predictive states for discrete valued data. To the best of my knowledge the only related work independent of Shalizi is (Jänicke, 2009; Jänicke and Scheuermann, 2010; Jänicke et al., 2007) who worked on computational improvements of existing algorithms as well as new visualization techniques of LSC for high dimensional data. They do not, however, statistically improve existing or propose new algorithms for predictive state space recovery.

Part II

OPTIMAL FORECASTS AND MEASURES OF COMPLEXITY IN SPATIO-TEMPORAL SYSTEMS

3

LOCAL PREDICTION OF SPATIO-TEMPORAL FIELDS

When one admits that nothing is certain one must, I think,
also add that some things are more nearly certain than others.

Bertrand Russell

Many important scientific and data-analytic problems involve fields which vary over both space and time, e.g., functional magnetic resonance imaging, meteorological observations, or experimental studies in physics, chemistry, and biology. An outstanding objective in studying such data is prediction, where we want to describe the field in the future.

Spatio-temporal data being increasingly easy to acquire, manipulate and visualize, statisticians have developed corresponding methods for statistical inference, reviewed in works like Cressie and Wikle (2011); Finkenstädt et al. (2007). The usual tools are a combination of ways of describing the distribution of the random field (e.g., various dependency measures), and stochastic modeling, focusing primarily

on parametric inference, and secondarily on parameter-conditional predictions.

In this chapter, I present optimality results from Shalizi (2003), which show how to construct provably optimal forecasts for spatio-temporal fields. The idea behind this construction is simply that it takes time for influences to propagate across space, so we can constrain the search for predictors to a spatio-temporally local neighborhood at each point.

Notation and concepts introduced here provide the basis for the statistical methodology I develop in the remainder of the thesis.

3.1 SETTING AND NOTATION

We observe a random field $(X(\mathbf{r}, t))_{\mathbf{r}\in\mathbf{S}, t\in\mathbb{T}}$ in discrete space and time. The field takes values in a set $\mathcal{X}$, which may be discrete or continuous. Space $\mathbf{S}$ is a regular lattice, equipped with norm $\|\mathbf{r}\|$. Time $\mathbb{T}$ is taken to be the positive integers up to T.

We restrict the setting to the regular lattice and regularly sampled times for the sake of computational convenience, easier visualization, and much simpler notation. We want to stress though, that the general concepts of light cones, predictive states, and the optimality results we outline below also apply to stochastic processes on arbitrarily shaped spatial domains (e.g., networks) and irregularly sampled time steps.

3.2 LIGHT CONES

Suppose that disturbances or influences in the system have a maximum speed of propagation, c. Then the only events which could affect what happens at a given $(\mathbf{r}, t)$ are those where $s \leqslant t$ and $\|\mathbf{r} - \mathbf{u}\| \leqslant c(t - s)$. Since this set grows as s recedes into the past, we call this the *past light cone* (PLC) of $(\mathbf{r}, t)$. The *future light cone* (FLC)

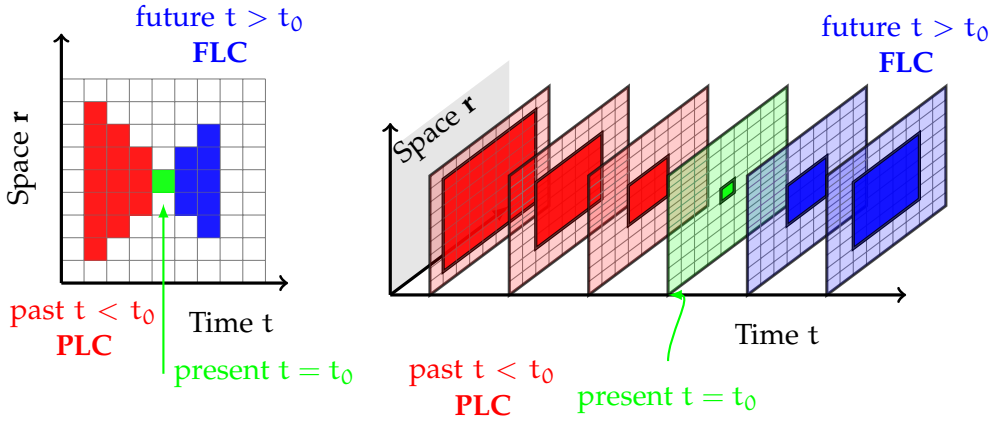


Figure 3.1: Past (red) and future (blue) light cones in a $(1+1)$D (a) and $(2+1)$D (b) field. Here c, the velocity of signal propagation, is set to 1. The past cone is truncated at a horizon of $h_p = 3$ steps, while the future cone's horizon is only $h_f = 2$. The present (green) is included in the future cone (see also Chapter 6, Proposition 6.1.2).

are all events which could be affected by the present moment $(\mathbf{r}, t)$; it thus consists of all those $(\mathbf{u}, s)$, where $s > t$ and $\|\mathbf{r} - \mathbf{u}\| \leqslant c(s-t)$. Light cones look like triangles in $(1+1)$D fields, and in $(2+1)$D, pyramids (Fig. 3.1). Denote the configuration in the past cone of $(\mathbf{r}, t)$ by $L^-(\mathbf{r}, t)$:

$$L^-(\mathbf{r}, t) = \{X(\mathbf{u}, s) \mid s \leqslant t, \|\mathbf{r} - \mathbf{u}\| \leqslant c(t-s)\} \tag{3.1}$$

$L^+(\mathbf{r}, t)$ is, similarly, the configuration in the future cone.

The spatio-temporal prediction problem is thus: use the configuration of the past cone, $L^-(\mathbf{r}, t)$, to forecast the configuration of the future cone, $L^+(\mathbf{r}, t)$. Light-cone prediction compromises between capturing global patterns and needing only local information. We will construct optimal predictors for light cones presently. Light cones can be defined for spatial extended patches of points. (When the "patch" becomes the whole spatial lattice, we are back to global prediction.) This leads to a parallel theory of prediction, but it turns out that the predictive state of a patch is

determined by the predictive states of its points (Shalizi, 2003, §3.3, Lemma 2 and Theorem 3), so we lose no information, and gain tractability, by not considering cones for patches.

Computationally, we need to truncate the cones at a finite number of time steps — we will call these the *past horizon* h_p of L^-, and likewise the *future horizon* h_f of L^+. Doing this reduces L^+ and L^- to finite-dimensional random vectors. (For instance, in Fig. 3.1, with $h_p = 3$ and $c = 1$, $\ell^-(\mathbf{r}, t)$ has 15 degrees of freedom.) The horizons are control settings, and may be tuned through (for example) cross-validation (§5.4.2). Similarly, when the maximum speed of propagation c is not given from background knowledge, it is also a control setting.

3.3 PREDICTIVE STATES

There is no present or future, only the past,
happening over and over again, now.

Eugene O'Neill

To predict the future $L^+(\mathbf{r}, t)$ from a particular past configuration, say ℓ^-, requires knowing the conditional distribution

$$\mathbb{P}\left(L^+(\mathbf{r}, t) \mid L^-(\mathbf{r}, t) = \ell^-\right) \tag{3.2}$$

for all ℓ^-. (Subsequently $(\mathbf{r}, t)$ may be omitted for readability.) Since treating this conditional distribution as an arbitrary function of ℓ^- is not feasible statistically or computationally, we try to find a *sufficient statistic* η of past configurations that keeps the predictive information:

$$\mathbb{P}\left(L^+(\mathbf{r}, t) \mid H(\mathbf{r}, t) = \eta(\ell^-)\right) = \mathbb{P}\left(L^+(\mathbf{r}, t) \mid L^-(\mathbf{r}, t) = \ell^-\right) . \tag{3.3}$$

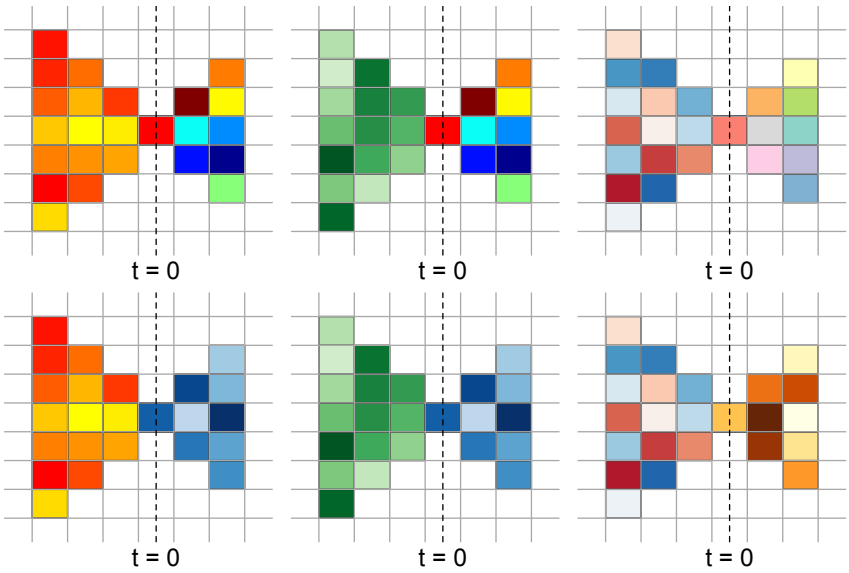


Figure 3.2: PLC configurations and their influence on FLCs: $\mathbb{P}\left(L^+ \mid \ell^-_{\text{left}}\right) = \mathbb{P}\left(L^+ \mid \ell^-_{\text{center}}\right) \neq \mathbb{P}\left(L^+ \mid \ell^-_{\text{right}}\right)$. Thus by Definition 3.3.1 $\ell^-_{\text{left}} \sim \ell^-_{\text{center}}$, but $\ell^-_{\text{left}} \nsim \ell^-_{\text{right}}$.

There are usually many sufficient statistics $\eta, \eta', \ldots$. When η and η' are both sufficient, but $\eta(\ell^-) = f(\eta'(\ell^-))$ for some f, then η is a smaller, more compressed, summary of the data than η', and so the former is preferred by Occam's Razor. The minimal sufficient statistic ϵ compresses the data as much as can be done without losing any predictive power, retaining only what is needed for optimal predictions.

We now construct the minimal sufficient statistic, following Shalizi (2003), to which we refer for some mathematical details.

Definition 3.3.1 (Equivalent configurations). *The past configurations ℓ^-_i at $(\mathbf{r}, t)$ and ℓ^-_j at $(\mathbf{u}, s)$ are* predictively equivalent, *$(\ell^-_i, (\mathbf{r}, t)) \sim (\ell^-_j, (\mathbf{u}, s))$, if they predict the same future with equal probabilities, i.e., if*

$$\mathbb{P}\left(L^+(\mathbf{r}, t) \mid L^-(\mathbf{r}, t) = \ell^-_i\right) = \mathbb{P}\left(L^+(\mathbf{u}, s) \mid L^-(\mathbf{u}, s) = \ell^-_j\right) \tag{3.4}$$

Figure 3.2 shows three past configurations - ℓ^-_{left}, ℓ^-_{center}, and ℓ^-_{right} - each one with exactly two equally probable FLCs. Even though all PLCs have pairwise different configurations, the left and center PLC have the same distribution over FLCs, whereas the right PLC leads to different FLCs. Thus $\ell^-_{\text{left}} \sim \ell^-_{\text{center}}$, but $\ell^-_{\text{left}} \nsim \ell^-_{\text{right}}$ - and since "$\sim$" is an equivalence class also $\ell^-_{\text{center}} \nsim \ell^-_{\text{right}}$.

Let $[(\ell^-, (\mathbf{r}, t))]$ be the equivalence class of $(\ell^-, (\mathbf{r}, t))$, i.e., the set of all past configurations and coordinates that predict the same future as ℓ^- does at $(\mathbf{r}, t)$. Let

$$\epsilon : (\ell^-, (\mathbf{r}, t)) \mapsto \left[\ell^-\right] \tag{3.5}$$

be the function mapping each $(\ell^-, (\mathbf{r}, t))$ to its predictive equivalence class.

The values ϵ can take are the *predictive states*; they are the minimal statistics which are sufficient for predicting L^+ from L^- (Shalizi, 2003). That means $\epsilon(\ell^-, (\mathbf{r}, t))$ has the same predictive information as $(\ell^-, (\mathbf{r}, t))$, i.e.,

$$\mathbb{P}\left(L^+(\mathbf{r}, t) \mid (\ell^-, (\mathbf{r}, t))\right) = \mathbb{P}\left(L^+(\mathbf{r}, t) \mid \epsilon(\ell^-, (\mathbf{r}, t))\right). \tag{3.6}$$

Furthermore, the future is conditional independent of the past given the predictive state

$$\mathbb{P}\left(L^+(\mathbf{r}, t) \mid (\ell^-, (\mathbf{r}, t)), \epsilon(\ell^-, (\mathbf{r}, t))\right) = \mathbb{P}\left(L^+(\mathbf{r}, t) \mid \epsilon(\ell^-, (\mathbf{r}, t))\right). \tag{3.7}$$

Eqs. (3.6) and (3.7) are the two main properties I use throughout this work.[1] In particular, in Chapter 6 I embed predictive states and the mapping ϵ in a statistical model so these optimal forecasts can be used for more general statistical inference.

1 For additional useful properties of predictive states see Shalizi (2003).

3.4 PREDICTIVE STATE PROCESS

Each predictive state has a unique predictive distribution and vice versa. We will thus slightly abuse notation to denote by $\mathcal{E}$ both the set of equivalence classes and the set of predictive distributions, whose elements we will write ϵ_j. We will further abuse notation by writing the mapping from past cone configurations to predictive distributions as $\epsilon(\cdot)$, leading to the measure-valued random field

$$S(\mathbf{r}, t) := \epsilon\left(L^{-}(\mathbf{r}, t)\right) \ . \tag{3.8}$$

One can show that $S(\mathbf{r}, t)$ is Markov even if $X(\mathbf{r}, t)$ is not (Shalizi, 2003). However, X is not an ordinary hidden Markov random field, since there is an unusual *deterministic* dependence between transitions in S and the realization of X, analogous to that of a chain with complete connections (Fernández and Maillard, 2005).

In the next chapter we use properties of this predictive state process to obtain a general measure of "interestingness" for spatio-temporal field.

4

FROM PREDICTIONS TO PATTERN DISCOVERY

And so from that, I've always been fascinated
with the idea that complexity can come out of such simplicity.

Will Wright

Many algorithms in statistics, machine learning, and signal processing try to express data in a better, more interesting coordinates, where *interesting* can have various interpretations: principal component analysis (PCA) finds uncorrelated projections of the data with highest variance (Jolliffe, 2002); independent component analysis (ICA) tries to decompose the signal into underlying independent sources (Hyvärinen and Oja, 2000); slow feature analysis (SFA) looks for slowly varying signals (Wiskott and Sejnowski, 2002); Laplacian graphs and diffusion maps find connected points on non-linear manifolds in high dimensional space (Lee and Wasserman, 2010; von Luxburg, 2006).

Although they work extremely well in a wide-range of real world applications, many algorithms must know what to look for, i.e., someone has to decide beforehand what is *interesting*. This works for well-defined tasks such as face/smile recognition in photos, but it can become very difficult to provide these features for problems that allow for a great variety of alternatives or features that are highly non-linear or high dimensional.

As for (upcoming) applications, it would be very beneficial to have an algorithm that tells us automatically – without any user input – what we should concentrate on and what to ignore, no matter if we analyze satellite images, recordings of microscopic heart muscle activity, or fMRI data.

Shalizi (2003); Shalizi et al. (2004) introduce *local statistical complexity* (LSC), a statistical method for automatic pattern discovery in spatio-temporal data. The underlying idea of LSC is that statistically optimal predictors do not only predict well but - for this very reason - reveal important dynamics. Shalizi et al. (2006) estimated LSC from discrete-valued fields and demonstrated that it can automatically reveal important spatio-temporal structures. An important contribution of this thesis is to estimate LSC from continuous-valued processes and thus being able to apply it to experimental data.

In this chapter, I review definitions and properties of LSC, and illustrate it on the simulated dataset from Chapter 1 - for convenience replicated in Fig. 4.2a.

4.1 LOCAL STATISTICAL COMPLEXITY

Following Crutchfield and Young (1989); Grassberger (1986), the *local statistical complexity* of the field $X(\mathbf{r}, t)$ was defined in Shalizi et al. (2004) as

$$\mathcal{C}(\mathbf{r}, t) = -\log_2 \mathbb{P}\left(S(\mathbf{r}, t) = \epsilon(\ell^-(\mathbf{r}, t))\right), \tag{4.1}$$

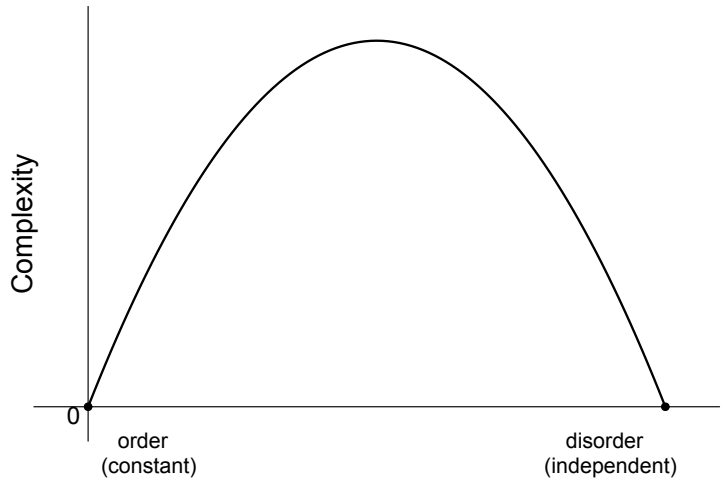


Figure 4.1: Complexity lies between order and disorder: completely ordered $X(\mathbf{r}, t)$ (constant over space and time) as well as completely unordered $X(\mathbf{r}, t)$ (independent) give $\mathcal{C}(\mathbf{r}, t) \equiv 0$, otherwise $\mathcal{C}(\mathbf{r}, t) > 0$ between these two extremes.

i.e., the minimum number of bits needed to describe the state of the prediction process at $(\mathbf{r}, t)$ (see also Fig. 4.1). Equivalently, this is the number of bits of information about the configuration of the PLC which are relevant to predicting the future (Shalizi et al., 2004). $\mathcal{C}(\mathbf{r}, t)$ is a non-negative, real-valued random field, where regions of high local complexity are ones where fine details of history matter to future evolution, suggesting them as targets for additional measurement and/or intervention.

In Section 6.6, I give yet another interpretation of LSC in the context of mixture / hidden-variable models.

4.1.1 *Average Complexity*

For biological organisms, it is often interesting to know if they have organized over time. The average complexity at time t

$$\overline{\mathcal{C}}(t) = \int_{\mathbf{S}} \mathcal{C}(\mathbf{r}, t)\, d\mathbf{r} \tag{4.2}$$

gives a one-dimensional summary of how interesting the field $X(\mathbf{r}, t)$ is at time t (see e.g., Fig. 8.1h). A system has organized between $t_1 < t_2$ if the average complexity has increased, i.e., $\overline{\mathcal{C}}(t_2) - \overline{\mathcal{C}}(t_1) =: \Delta\overline{\mathcal{C}} > 0$ (Shalizi et al., 2004). Thus jumps in $\overline{\mathcal{C}}(t)$ immediately show when an interesting event has (stopped to) happened.

Complimentary, a temporal average

$$\overline{\mathcal{C}}(\mathbf{r}) = \int_{\mathbb{T}} \mathcal{C}(\mathbf{r}, t)\, dt \tag{4.3}$$

shows a spatial "interestingness map" for each $\mathbf{r}$ (see e.g., Fig. 8.2).

4.2 ESTIMATING LSC

Shalizi et al. (2004) estimate (4.1) as follows: i) determine the set of different predictive states $\mathcal{E}$ (either known from simulations or estimated from data), ii) for each state, count the total number of PLCs in the state and divide by N to obtain a frequency estimate for the probability of being in a particular state, iii) compute the information-theoretic entropy of this event, and iv) assign it to location $(\mathbf{r}, t)$.

Formally this is,

$$\widehat{\mathcal{C}}(\mathbf{r}, t) = -\log_2 \widehat{\mathbb{P}}\left(S(\mathbf{r}, t) = \epsilon(\ell^-(\mathbf{r}, t))\right) \tag{4.4}$$

$$= -\log_2 \frac{\sum_{(\mathbf{s},u)\in\mathbf{S}\times\mathbb{T}} \mathbf{1}\left(S(\mathbf{s}, u) = \widehat{\epsilon}(\ell^-(\mathbf{r}, t))\right)}{N}, \tag{4.5}$$

where $S(\mathbf{r}, t)$ is the predictive state process from (3.8).

4.3 ILLUSTRATION

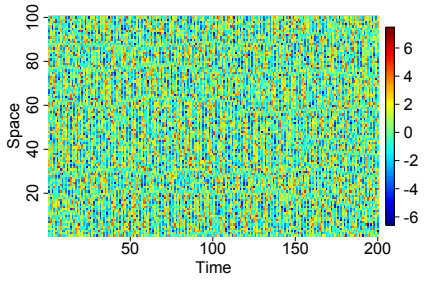
Simplicity does not precede complexity, but follows it.

Alan Perlis

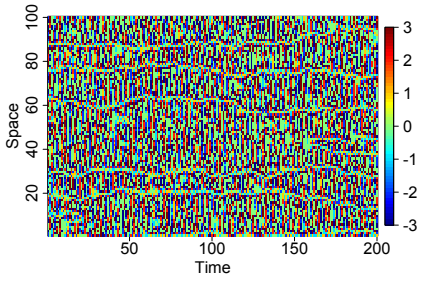
For the purpose of illustration, we compute $\mathcal{C}(\mathbf{r}, t)$ for the field in Fig. 4.2a (for easier comparison this is a replicate of Fig. 1.1a). Since this data was simulated we *know* the predictive state space (Fig. 4.2b).

Figure 4.2c shows the complexity at each space-time coordinate, $\mathcal{C}(\mathbf{r}, t)$, along with its spatial (top) and temporal (right) average complexity. High complexity is dark-red, low complexity is yellow. It confirms the first impression of Fig. 1.1a that the green traces (left to right) are especially important for the evolution of this field; less complex, but still visible are color switches between neighboring columns, and the low-complexity background is least important. The spatial average is essentially constant since the field does not show any variation in spatial complexity over time. It has thus neither organized nor disorganized over time, $\Delta\overline{\widehat{\mathcal{C}}} \approx 0$. The temporal average also shows that the most interesting areas in space are the horizontal traces, which act as barriers for information to pass from one horizontal strip to another.

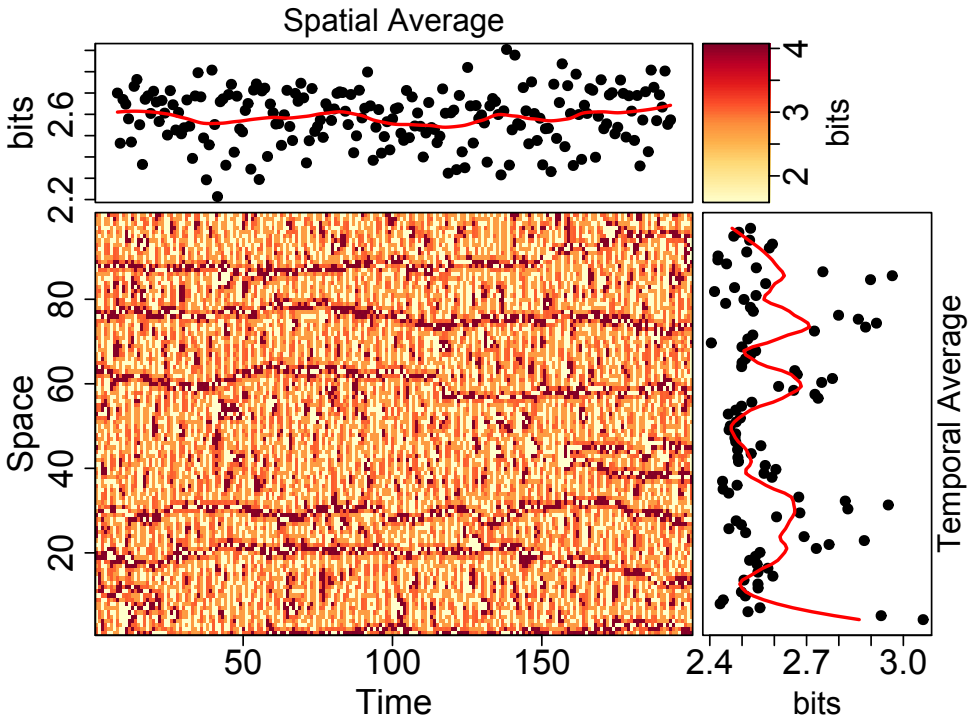
While being a simulated toy-example with a *known* state space, Fig. 4.2 shows that LSC is a useful measure to find interesting patterns. In Part III, I show how to obtain consistent estimates of $S(\mathbf{r}, t)$, and - by Slutsky's theorem (Ash and Doléans-Dade, 2000) - we thus obtain a consistent estimator $\widehat{\mathcal{C}}(\mathbf{r}, t)$ (see also Antos and Kontoyiannis, 2001). I then use these estimators on fMRI data to detect interesting brain activity.



(a) observed field



(b) true predictive states



(c) $\mathcal{C}(\mathbf{r}, t)$ computed via (4.4)

Figure 4.2: Local statistical complexity for simulated $(1+1)$D field.

Part III

METHODS, RESULTS, AND APPLICATIONS

5

LICORS: LIGHT CONE RECONSTRUCTION OF STATES

Suam habet fortuna rationem.

(Chance has its reasons.)

Petronius

Our aim here is to blend modern methods of nonparametric prediction with insights from nonlinear physics on the organization of spatial dynamics, yielding predictors of spatio-temporal evolution that are computationally efficient and make minimal assumptions on the data source, but are still accurate and even interpretable.

We achieve this by combining the optimality results from Chapter 3 with a novel form of nonparametric smoothing, which infers the prediction (regression or conditional probability) function by averaging together similar observations, where "similarity" is defined in terms of predictive consequences, effectively replacing the original geometry of the predictor variables with a new one, optimized for forecasting. This new geometry lets us discover underlying structures, as well make fast and accurate predictions.

Section 5.1 presents our nonparametric statistical methods to estimate the predictive state mapping ϵ (Eq. (3.5)) from continuous-valued fields. Section 5.2 shows, under weak conditions on the data-generating process, that our method consistently estimates the predictive state and corresponding distributions. Section 5.3 gives implementation details. Section 5.4 compares the predictive accuracy to standard time series techniques via extensive simulations and also proposes a cross-validation scheme to choose the control settings. Proofs of the main results can be found in Appendix A.

5.1 OPTIMAL NONPARAMETRIC FORECASTS FOR SPATIO-TEMPORAL DATA

We extend the work of Shalizi (2003); Shalizi et al. (2004) to continuous-valued fields, introducing statistical methods to estimate and predict non-linear dynamics accurately and efficiently, while still obtaining insight into the spatio-temporal structure.

See Figure 5.1 for an overview of the method.

To be able to draw useful inferences from a single realization of the process, we must assume some form of homogeneity or invariance of the conditional distributions.

Assumption 5.1.1 (Conditional invariance). *The predictive distribution of a PLC configuration ℓ^- does not change over time or space. That is, for all $\mathbf{r}, t$, all $\mathbf{u}, s$, and all past light-cone configurations ℓ^-,*

$$(\ell^-, (\mathbf{r}, t)) \sim (\ell^-, (\mathbf{u}, s)) \tag{5.1}$$

We may thus regard $\sim$ as an equivalence relation among PLC configurations, and ϵ as a function over ℓ^- alone.

This is just *conditional* invariance, like the conditional stationarity for time series used in Caires and Ferreira (2005). It would be implied by the field being a Markov random field with homogeneous transitions, or of course by full stationarity and spatial invariance, but it is weaker. Assumption 5.1.1 lets us talk about *the* predictive distribution of a PLC configuration, regardless of when or where it was observed, and to draw inferences by pooling such observations. If this assumption fails, we could in principle still learn a different set of predictive states for each moment of time and/or each point of space (as in Shalizi (2003)), but this would need data from multiple realizations of the same process.

Assume we have T consecutive measurements of the field $X(\mathbf{r}, t)$, observed over the lattice $\mathbf{S}$, with $N = |\mathbf{S}| \cdot T$ space-time coordinates $(\mathbf{r}, t)$ in all. Each one of these N point-instants has a past and a future light-cone configuration, $\ell^-(\mathbf{r}, t)$ and $\ell^+(\mathbf{r}, t)$, represented as, respectively, n_p and n_f dimensional vectors. Since predictive states are sets of PLC configurations with the same predictive distribution, we need to test this sameness, based on conditional samples $\{\ell^+ \mid \ell_i^-\}_{i=1}^N$ from the observed

1. Collect the PLC and FLC configurations, $\ell^-(\mathbf{r}, t)$ and $\ell^+(\mathbf{r}, t)$, for each $(\mathbf{r}, t)$ in the observed data $\mathcal{D} = \{X(\mathbf{r}, 1), \ldots, X(\mathbf{r}, T)\}_{\mathbf{r} \in \mathbf{S}}$.
2. To cluster or not to cluster:
   a) Assign each point to its own cluster. Only for small N this is computationally feasible.
   b) Perform an initial clustering (e.g., K-means++ (Arthur and Vassilvitskii, 2007)) in the PLC configuration space (Section 5.1.1).
3. For each pair of clusters, test whether the estimated conditional FLC distributions are significantly different, at some fixed level α (Section 5.1.2). If not, merge them and go on. Stop when no more merges are possible.
4. Treat the remaining clusters as predictive states, and estimate the conditional distributions over FLC configurations.
5. Return the partition of PLC configurations into predictive states, and the associated predictive distributions.

Figure 5.1: Estimating predictive states from continuous-valued data: in 2a conditional distributions are tested for each ℓ_i^-, $i = 1, \ldots, N$, using a δ-neighborhood (or k nearest neighbors) of ℓ_i^- (see Section 5.2.1.1 for details); 2b uses an initial clustering to reduce complexity of the testing problem from $\mathcal{O}(N^2)$ to $\mathcal{O}(K^2)$ (see also Section 5.1.1).

field. We will apply nonparametric two-sample tests for $H_0 : \mathbb{P}\left(L^+ \mid L^- = \ell_i^j\right) = \mathbb{P}\left(L^+ \mid L^- = \ell_i^-\right)$ pairwise for all i and j. Because there are typically a great many past light cones (one for each point-instant), and light-cone configurations are themselves high-dimensional objects, we generally must do this step-wise.

5.1.1 *Partitioning PLC Configurations: Similar Pasts Have Similar Futures*

It is often reasonable to assume that the mapping from the past to predictive distributions is regular, so that if two historical configurations are close (in some suitable metric), then their predictive distributions are also close. This lets us avoid having to do some pairwise tests, as their results can be deduced from others.

Assumption 5.1.2 (Continuous histories). *For every* $\rho > 0$*, there exists a* $\delta > 0$ *such that*

$$\|\ell_i^- - \ell_j^-\| < \delta \Rightarrow \mathcal{D}_{KL}\left(\mathbb{P}\left(L^+ \mid \ell_i^-\right) \| \mathbb{P}\left(L^+ \mid \ell_j^-\right)\right) < \rho, \tag{5.2}$$

where $\mathcal{D}_{KL}(p \| q)$ *is the Kullback-Leibler divergence between distributions* p *and* q *(Kullback, 1968).*

Assumption 5.1.2 requires that sufficiently small changes ($< \delta$) in the local past make only negligible ($< \rho$) changes to the distribution of local future outcomes. Statistically, such smoothness-in-distribution lets us pool observations from highly similar PLC configurations, enhancing efficiency; physically, it reflects the smoothness of reasonable dynamical mechanisms. Chaotic systems, where the exact trajectory depends sensitively on initial conditions, do not present difficulties, since Assumption 5.1.2 is about the conditional *distribution* of the future given partial information on the past, and chaos has long been recognized as a way to stabilize such distributions, forming the basis for prediction and control of chaos (Kantz and Schreiber, 2004).

We use Assumption 5.1.2 to justify an initial "pre-clustering" of the PLC configuration space, greatly reducing computational cost with little damage to predictions. We first divide the PLC configuration space using fast clustering algorithms into $K \ll N$ clusters, and then test equality of distributions between clusters ($\mathcal{O}(K^2)$), rather than light cones ($\mathcal{O}(N^2)$).

When N is small enough, we can skip this initial pre-clustering. To simplify exposition, we treat this as assigning each distinct past cone to its own cluster.

5.1.2 *Partitioning Clusters into Predictive States*

Each cluster P_k contains a set of similar PLC configurations, and also defines a sample of conditional FLCs, $\mathbf{F}_k(\delta) = \{\ell_j^+ \mid \ell_j^- \in P_k\} \in \mathbb{R}^{N_k \times n_f}$, $k = 1, \ldots, K$. Since all $\ell_j^- \in P_k$ have very similar distribution, $\mathbf{F}_k \sim Q$ is an approximate sample from the predictive distribution $\mathbb{P}(L^+(\mathbf{r}, t) \mid \ell^- \in P_k)$. Lemma 5.2.7, below, shows that for sufficiently small δ, $\mathbf{F}_{k_i}(\delta)$ is an exact sample of $p(\epsilon(P_{k_i}))$. Thus, to simplify the exposition, we ignore the ρ difference in this section.

Finding equivalent clusters reduces to testing hypotheses of the form $H_0 : p_{k_i} = p_{k_j}$ based on the two samples $\mathbf{F}_{k_i}(\delta)$ and $\mathbf{F}_{k_j}(\delta)$. For $h_f = 0$ and $c = 1$, FLCs are one-dimensional and we can use a Kolmogorov-Smirnov test (or any other two-sample univariate test). In general, however, $\mathbf{F}_k$ are samples from a very high-dimensional distribution, and we use nonparametric, multivariate, two-sample tests (see e.g. Gretton, Borgwardt, Rasch, Schölkopf, and Smola, 2007; Rizzo and Székely, 2010; Rosenbaum, 2005). Any test satisfying Assumption 5.2.14 could be used.

To estimate the predictive states from an initial partitioning of PLC configurations, we iterate through the list of configurations, recursively testing equality of distributions. To initialize the algorithm, create the first predictive state ϵ_1, containing the first configuration ℓ_1^-. Then take ℓ_2^- and test if its distribution is equal to that of ϵ_1. If it is (at the level α), then put ℓ_2^- in ϵ_1; otherwise generate a new predictive state ϵ_2 with ℓ_2^-. Then test the next configuration against all previously established predictive states and proceed as before. This continues until all configurations have been assigned to a predictive state.

The predictive distribution of each predictive state can be found by applying any consistent nonparametric density estimator to the future cone samples belonging to that state. If we only want point forecasts, we can skip estimating the whole predictive distribution and just get (e.g.) the mean of the samples.

5.2 CONSISTENCY

LICORS consistently recovers the correct assignment of past cone configurations to predictive states, and the predictive distributions, under weak assumptions on the data-generating process. These allow for the number of predictive states to grow slowly with the sample size, so that we have nonparametric consistency. Proofs of the main results can be found in Appendix A.

5.2.1 *Assumptions*

Let $N = |\mathbf{S} \times \mathbb{T}|$ be the total number of space-time points at which we observe both the past and future light cone. We presume that $N \to \infty$, without caring whether $|\mathbf{S}| \to \infty$, $|\mathbb{T}| \to \infty$, or both.

Assumption 5.2.1 (Slowly growing number of predictive states)**.** *The number of predictive states,* $|\mathcal{E}| = m(N) = o(N)$*, and always* $\leqslant N$*.*

Assumption 5.2.1 only guarantees that *at least* one of the predictive states grows in size. To bound testing error probabilities, the number of light cones seen in *every* state must grow as N grows.

Assumption 5.2.2 (Increasing number of light cones in each state)**.** *The number of light cones in each state,* $N_j := |\epsilon_j|$*, grows with* N*: for all* $\epsilon_j \in \mathcal{E}$*,*

$$\lim_{N \to \infty} N_j(N) = \infty \tag{5.3}$$

Let $N_{\min} = \min_j N_j$ be the number of samples in the smallest predictive state; thus also $N_{\min} \to \infty$ for $N \to \infty$. Assumption 5.2.2 means that the process re-visits each predictive state as it evolves, i.e., all states are recurrent. This lets us learn the predictive distribution of each state from a growing sample of its behavior.

Assumption 5.2.3 (Bounded conditional distributions)**.** *All predictive distributions* $\epsilon_j \in \mathcal{E}$ *have densities with respect to a common reference measure* ν*, and* $0 < \iota < d\epsilon_j/d\nu < \kappa < \infty$*, for some constants* ι *and* κ*.*

This merely technical assumption guarantees bounded likelihood ratios.

Assumption 5.2.4 (Distinguishable predictive states)**.** *The KL divergence between states is bounded from below:* $\forall i \neq j$*,*

$$0 < d_{min} \leqslant \mathcal{D}_{KL}\left(\epsilon_i \,\|\, \epsilon_j\right) =: d_{i,j} \tag{5.4}$$

We do not need $d_{i,j} < \infty$. (In fact, $\mathcal{D}_{KL}\left(\epsilon_i \,\|\, \epsilon_j\right) = \infty$ is helpful.) Eq. (5.4) is automatically satisfied for any fixed number of states. For an increasing state space, $m = m(N)$, assume

$$\inf_{i,j \in m(N)} d_{i,j} = d_{min} > 0 \text{ for } N \to \infty. \tag{5.5}$$

Lemma 5.2.5 (Conditionally independent FLCs)**.** *If the cones* $L^+(\mathbf{r}, t)$ *and* $L^+(\mathbf{u}, s)$ *do not overlap, then*

$$L^+(\mathbf{r}, t) \perp\!\!\!\perp L^+(\mathbf{u}, s) \mid S(\mathbf{r}, t), S(\mathbf{u}, s)\,. \tag{5.6}$$

In particular,

$$\mathbb{P}\left(L^+(\mathbf{r}, t), L^+(\mathbf{u}, s) \mid S(\mathbf{r}, t), S(\mathbf{u}, s)\right) = \mathbb{P}\left(L^+(\mathbf{r}, t) \mid S(\mathbf{r}, t)\right) \mathbb{P}\left(L^+(\mathbf{u}, s) \mid S(\mathbf{u}, s)\right)\,. \tag{5.7}$$

Corollary 5.2.6. *If* $h_f = 0$*, then FLCs are conditionally independent given their predictive state.*

5.2.1.1 *Getting samples from* ϵ_i

We get a sample of FLCs from the predictive distribution of ℓ_i by first taking all PLCs in a δ-neighborhood around ℓ_i,

$$I_i(\delta) = \{j \mid \|\ell_i^- - \ell_j^-\| < \delta\}. \tag{5.8}$$

For later use, we denote the number of such light cones by $S_i(N, \delta) = |I_i(\delta)|$. By Assumption 5.1.2, we get our sample from ϵ_i by collecting the corresponding future cone configurations:

$$\mathbf{F}_i(\delta) = \{\ell_j^+ \mid j \in I_i(\delta)\}, \tag{5.9}$$

Lemma 5.2.7. *For sufficiently small $\delta > 0$, all past configurations in $I_i(\delta)$ are predictively equivalent: $\forall j, k \in I_i(\delta), \ell_j^- \sim \ell_k^-$. Consequently, all ℓ_j^+, $j \in I_i(\delta)$, are drawn from the same distribution $\epsilon(\ell_i^-)$.*

For finite N, it may not be possible in practice to find and use a sufficiently small δ. With pre-clustering, for instance, some of the clusters may have diameters greater than the δ which guarantees equality of distribution. Then the samples $\mathbf{F}_i(\delta)$ are actually from multiple states. One could circumvent this by using more clusters, which generally shrinks cluster diameters, but this would also reduce the number of samples per neighborhood, increasing the error rate of our two-sample tests. In practice, one must trade off decreasing δ to discover all predictive states and keeping a low testing error.

Corollary 5.2.8. *For sufficiently small $\delta > 0$, and non-overlapping FLCs, all the future configurations in $\mathbf{F}_i(\delta)$ are IID samples from $\epsilon(\ell_i^-)$.*

In general, for $h_f > 0$, the FLCs in $\mathbf{F}_i(\delta)$ can be overlapping and the conditional likelihood does not factorize. Yet, without loss of generality, we can consider only non-overlapping FLCs. This is because we can explicitly exclude overlapping FLCs

from $\mathbf{F}_i(\delta)$, at the cost of reducing the sample size to $\tilde{S}_i(N,\delta) \leqslant S_i(N,\delta)$. For each ℓ_i, the maximum number of FLCs which we must thereby exclude, say w, is fixed geometrically, by c, h_f, and the dimension of the space $\mathbf{S}$, and does not grow with N. The exclusion is thus asymptotically irrelevant, since $\frac{S_i(N,\delta)}{w} \leqslant \tilde{S}_i(N,\delta) \leqslant S_i(N,\delta)$.

Furthermore, at least formally, it is enough to analyze the univariate, zero-horizon FLC distributions, which rules out overlaps. This is because longer-horizon FLC distributions must be consistent with the one-step ahead distributions and the transition relations of the underlying predictive states. Thus we could get the n_f-dimensional FLC distribution by iteratively combining the univariate FLC distributions and the predictive state transitions, i.e., by chaining together one-step-ahead predictions, as in Shalizi and Crutchfield (2001, Corollary 2).

Assumption 5.2.9 (Number of samples from each cone)**.** *For each fixed $\delta > 0$, and each past light cone ℓ_i, $S_i(N,\delta) \xrightarrow[N\to\infty]{} \infty$.*

For each δ, $S_i(N,\delta)$ is a random variable, and to establish consistency we need some regularity conditions on how S_i grows with N. Let $S_{\min}(N,\delta) = \min_j S_j(N,\delta)$ be the smallest number of samples per δ-neighborhood for each N and δ.

Assumption 5.2.10. *For some $\tilde{c} > 0$,*

$$N \cdot m(N) \cdot \mathbb{E}e^{-\tilde{c}\, d_{\min}^2 S_{\min}(N,\delta)} \xrightarrow[N\to\infty]{} 0\,. \tag{5.10}$$

Since $\mathbb{E}e^{tS_{\min}(N,\delta)}$ is the moment generating function of $S_{\min}$, this amounts to asserting that the number of samples concentrates around its mean while growing, ruling out pathological cases where $S_i(N,\delta)$ grows to infinity, but concentrates around small values.

5.2.2 *Consistency in the Oracle Setting*

First we consider the setting where an oracle gives us the predictive distributions.

Assumption 5.2.11 (Oracle: $\mathcal{P}$ known). *The distributions $p_1, \dots, p_{m(N)} \in \mathcal{P}$ of each of the $m(N)$ predictive states are known.*

Under Assumption 5.2.11, recovering the predictive state means assigning each ℓ_i^-, $i = 1, \dots, N$ to one of the $m(N)$ predictive distributions. We represent the correct assignment by an $N \times m(N)$ binary matrix $\mathbf{B}$, where $B_{ij} = 1$ iff $\epsilon_j = \epsilon(\ell_i^-)$. In the oracle setting, we can consistently estimate $\mathbf{B}$ by maximizing the likelihood.

From Lemma 5.2.7, there is a $\delta > 0$ such that the FLCs we consider, $\mathbf{F}_i(\delta) = \{\ell^+_{i,1}, \dots, \ell^+_{i,S_i(N,\delta)}\}$, are actually conditioned on the same (still unknown) predictive state $\epsilon(\ell_i^-)$. The log-likelihood is thus, by Corollary 5.2.8,

$$\log L(p_k; \mathbf{F}_i(\delta)) = \prod_{s=1}^{S_i(N,\delta)} p_k\left(\ell^+_{i,s}\right), \tag{5.11}$$

The maximum likelihood estimator (MLE), $\widehat{\theta}_{MLE}(i)$, assigns PLC ℓ_i^- to the state with the largest likelihood. This leads to a binary matrix $\widehat{\mathbf{B}}$, where $\widehat{B}_{ij} = 1$ iff $\widehat{\theta}_{MLE}(i) = j$.

Theorem 5.2.12 (Consistency). *Under Assumptions 5.1.2, 5.1.1, 5.2.1, 5.2.2, 5.2.3, 5.2.4, 5.2.9, 5.2.10, and 5.2.11, if*

$$S_{\min}(N, \delta) \cdot d^2_{\min} = \Omega(\log N + \log m(N)) \tag{5.12}$$

then, asymptotically, predictive states can be recovered perfectly,

$$\mathbb{P}\left(\widehat{\mathbf{B}}_{MLE} \neq \mathbf{B}\right) \xrightarrow[N\to\infty]{} 0\,. \tag{5.13}$$

Corollary 5.2.13. *If (5.12) holds, then predictive states can be estimated consistently even if $d_{i,j} \to 0$ as long as the mgf of $S_{\min}$ satisfies Assumption 5.2.10 also for $d^2_{\min} \to 0$.*

5.2.3 *Unknown Predictive States: Two-sample Problem*

With a finite number of observations, N, recovering the states is the same as determining which past cone configurations are predictively equivalent. We represent this with an $N \times N$ binary matrix $\mathbf{A}$, where $A_{ij} = 1$ if and only if $\ell_i \sim \ell_j$. LICORS gives us an estimate of this matrix, $\widehat{\mathbf{A}}$, and we will say that the predictive states can be recovered consistently when

$$\mathbb{P}\left(\widehat{\mathbf{A}} \neq \mathbf{A}\right) \xrightarrow[N\to\infty]{} 0 \,. \tag{5.14}$$

Since the predictive distributions are unknown, we use nonparametric two-sample tests to determine whether two past cone configurations are predictively equivalent. While simulations can always be used to approximate the power of particular tests against particular alternatives, there do not (yet) seem to be any general expressions for the power of such tests, analogous to the bounds on likelihood tests in terms of KL divergence (Kullback, 1968). Nonetheless, we expect that for $N \to \infty$, the probability of error approaches zero, as long as the true distributions are far enough apart. We thus make the following assumption.

Assumption 5.2.14. *Suppose we have* n *samples from distribution* p*, and* n' *samples from distribution* q*, all IID. Then there exist a positive constants* $d_{n,n'}$ *tending to* 0 *as* $n, n' \to \infty$*, and a sequence of tests* $T_{n,n'}$ *of* $H_0 : p = q$ *vs.* $H_1 : p \neq q$ *with size* $\alpha = o\big(\min(n, n')^{-2}\big)$*, and type II error rate* $\beta(\alpha, n, n') = o\big(\min(n, n')^{-2}\big)$ *so long as* p *and* q *are mutually absolutely continuous and* $\mathcal{D}_{KL}(p \,\|\, q) \geqslant d_{n,n'}$*.*

If the number of predictive states is constant in N, we can weaken the assumption to just a sequence of tests whose type I and type II error probabilities both go to zero supra-quadratically when $\mathcal{D}_{KL}(p \,\|\, q) \geqslant d_{\min}$.

Theorem 5.2.15 (Consistent predictive state estimation)**.** *Under Assumptions 5.1.1, 5.1.2, 5.2.1, 5.2.2, 5.2.3, 5.2.4, 5.2.9, 5.2.10, and 5.2.14,*

$$\mathbb{P}\left(\widehat{\mathbf{A}} \neq \mathbf{A}\right) \xrightarrow[N\to\infty]{} 0. \tag{5.15}$$

5.3 DETAILS OF IMPLEMENTATION AND ALGORITHMS

We partition the observed PLCs $\{\ell_i^-\}_{i=1}^N \subset \mathbb{R}^{n_P}$ into $K = K(\delta)$ disjoint groups $\{P_k\}_{k=1}^K$, choosing the number of groups so that all have diameters less than δ. This choice of $K(\delta)$ guarantees (Assumption 5.1.2) that all $\ell^- \in P_k$ have predictive distributions that are at most ρ apart. Thus all PLCs within a group P_k are (nearly) equivalent by Definition 3.3.1. This in turn means we only need to compare predictive distributions between clusters.

5.3.1 *Further performance enhancements for testing*

While it is better to do $\mathcal{O}(K^2)$ high-dimensional tests than $\mathcal{O}(N^2)$, it would be better still to speed up each test. Since two distributions are the same only if their moments are, we can start by testing simply for equality of means, which is fast and powerful, and do a full distributional test only if we cannot reject on that basis. For multivariate mean tests we can use the Hotelling test (Abello, Buchsbaum, and Westbrook, 1998) and its randomized generalization (Lopes, Jacob, and Wainwright, 2011). Yet another strategy to reduce the number of costly high-dimensional, non-parametric tests is to test various functions $f(\cdot)$ of the samples. If the distributions of $\mathbf{F}_{k_i}(\delta)$ and $\mathbf{F}_{k_j}(\delta)$ are the same, then also $\mathbb{P}\left(f\left(\mathbf{F}_{k_i}(\delta)\right)\right) = \mathbb{P}\left(f\left(\mathbf{F}_{k_j}(\delta)\right)\right)$ for any measurable f. Particularly, we can apply random projections (Lopes et al., 2011) to $\mathbf{F}_{k_i}$ to go from the high-dimensional $\mathbb{R}^{n_f}$ down to the one-dimensional $\mathbb{R}$, followed

Algorithm 1 Test equality of conditional predictive FLC distributions $\mathbb{P}(L^+ \mid \text{clusterID} = k)$

Input:

$\mathbf{F} = \{\ell_i^+\}_{i=1}^N$	...	$N \times n_f$ array with FLC samples
clusterID	...	labels of PLC partitioning (step 2a or 2b in Fig. 5.1)
$\alpha \in [0,1]$	...	significance level α for testing $H_0 : \mathbb{P}\left(L^+ \mid \ell_i^-\right) = \mathbb{P}\left(L^+ \mid \ell_j^-\right)$

Output: predictive state labels

$k_{max} = \max$ clusterID
for $k = 1, \ldots, k_{max}$ **do**
  fetch FLC samples given partition P_k: $\mathbf{F}_k = \{\ell_i^+\}_{\{i|\texttt{clusterID}[i]==k\}}$
  j = k
  lasttested = 0; pvalue = 1
  while pvalue $> \alpha$ or $j \leqslant k_{max}$ **do**
    j = j+ 1
    lasttested = j
    fetch FLC samples given partition P_j: $\mathbf{F}_j = \{\ell_i^+\}_{\{i|\texttt{clusterID}[i]==j\}}$
    pvalue $\leftarrow$ test$(\mathbb{P}(L^+ \mid P_k) = \mathbb{P}\left(L^+ \mid P_j\right) \mid \mathbf{F}_k, \mathbf{F}_j)$
    if pvalue $< \alpha$ **then**
      merge cluster j with cluster k: clusterID[clusterID == j] = k
return clusterID

by a Kolmogorov-Smirnov test. Only if these tests can not reject equality for several projections, one uses multivariate nonparametric tests.

5.4 SIMULATIONS

To evaluate the non-asymptotic predictive ability of LICORS and to compare it to more conventional methods, we use the following simulation, designed to be challenging, but not impossible. $X(\mathbf{r}, t)$ is a continuous-valued field in $(1+1)$D, with a discrete latent state $d(\mathbf{r}, t)$. We use "wrap-around" boundary conditions,

so sites 0 and $|S|-1$ are adjacent, and the one spatial dimension is a torus. The observable field $X(\mathbf{r},t)$ is conditionally Gaussian,

$$\mathbb{P}\left(X(\mathbf{r},t) \mid d(\mathbf{r},t)\right) = \begin{cases} \mathcal{N}(d(\mathbf{r},t),1), & \text{if } |d(\mathbf{r},t)| < 4, \\ \mathcal{N}(0,1), & \text{otherwise,} \end{cases} \tag{5.16}$$

$$\text{and initial conditions: } X(\cdot,1) = X(\cdot,2) = \mathbf{0} \in \mathbb{R}^{|S|}. \tag{5.17}$$

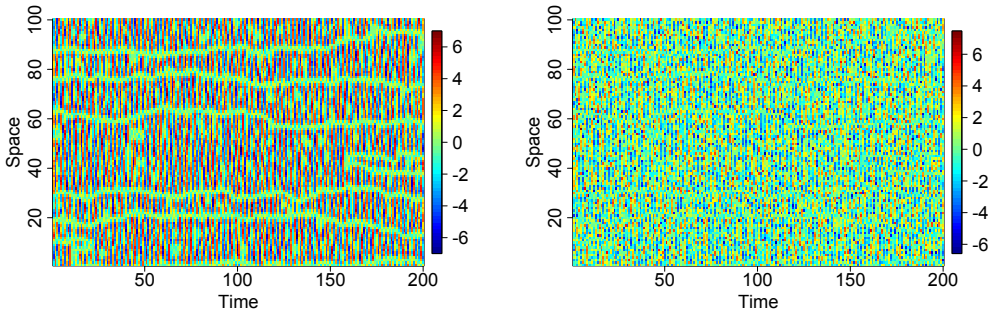
The state space $d(\mathbf{r},t)$ evolves with the observable field,

$$d(\mathbf{r},t) = \left[\frac{\sum_{i=-2}^{2} X((\mathbf{r}+\mathbf{i}) \bmod |\mathbf{S}|, t-2)}{5} - \frac{\sum_{i=-1}^{1} X((\mathbf{r}+\mathbf{i}) \bmod |\mathbf{S}|, t-1)}{3}\right], \tag{5.18}$$

where $[x]$ is the closest integer to x. In words, Eq. (5.18) says that the latent state $d(\mathbf{r},t)$ is the rounded difference between the sample average of the 5 nearest sites at $t-2$ and the sample average of the 3 nearest sites at $t-1$. Thus $h_p = 2$ and $c = 1$.

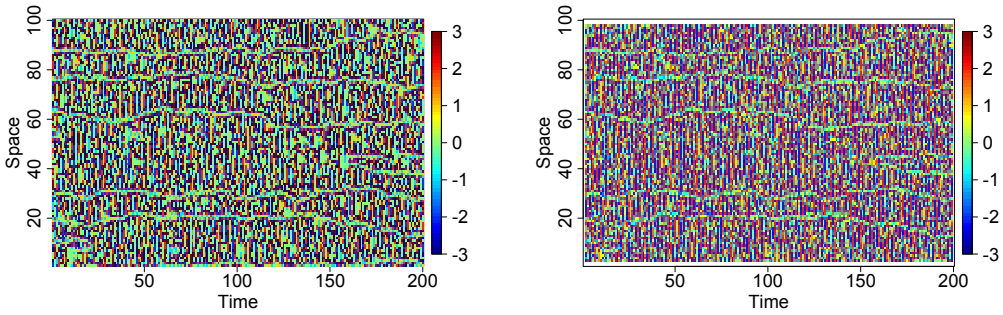
If we include the present in the FLC, (5.16) gives $h_f = 0$, making FLC distributions one-dimensional and letting us use the Kolmogorov-Smirnov test. As $d(\mathbf{r},t)$ is integer-valued, a little calculation shows there are 7 predictive states, which we label with their conditional means as $\{\epsilon_{-3}, \epsilon_{-2}, \ldots, \epsilon_2, \epsilon_3\}$. Thus $X(\mathbf{r},t) \mid \epsilon_k \sim \mathcal{N}(k,1)$.

Figure 5.2 shows one realization of (5.16)–(5.18). The latent states have clear spatial structures, which is obscured in the observed field. Figure 5.3a shows the true predictive state space $S(\mathbf{r},t)$ (expected value at each $(\mathbf{r},t)$); the LICORS estimate $\widehat{S}(\mathbf{r},t)$ is shown in Fig. 5.3b. LICORS not only accurately estimates $S(\mathbf{r},t)$, but also learns the prediction rule (5.16) from the observed field $X(\mathbf{r},t)$.



(a) (b)

Figure 5.2: Simulation of (5.16)–(5.18): (a) state-space $d(\mathbf{r}, t)$, (b) observed field $X(\mathbf{r}, t)$. Space (100 cells) runs vertically, time (200 steps, first 100 discarded for burn-in) runs from left to right.



(a) (b)

Figure 5.3: Comparison of true and estimated predictive distributions. (a) true predictive state $S(\mathbf{r}, t)$, with points colored by conditional expectations; (b) LICORS predictions, with states and distributions reconstructed using $k = 50$ nearest neighbors (fixed) and $h_p = 2, \alpha = 0.2$ (chosen by cross-validation).

5.4.1 *Forecasting Competition: AR, VAR, and LICORS*

A brute-force approach to spatio-temporal prediction would treat the whole spatial configuration at any one time as a single high-dimensional vector, and then use ordinary, parametric time-series methods such as vector auto-regressions (VAR) (Lütkepohl, 2007), or non- or semi-nonparametric models (Bosq, 1998; Fan and Yao, 2003). Such global approaches suffer under the curse of dimensionality: real data sets may contain millions of space-time points. Hence, fitting global models becomes im-

practical, even with strong regularization (Bosq and Blanke, 2007). Moreover, such global models will not be good representations of complex spatial dynamics.

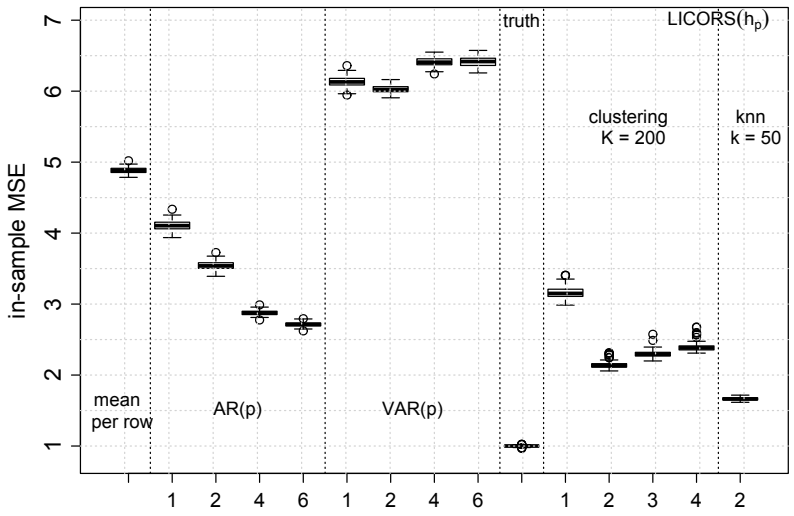
On the other hand, space can be broken up into small patches (in the limit, single points), followed by fitting standard time series model to each low-dimensional patch. Such local strategies (partially) lift the curse of dimensionality and hence make VAR or nonparametric time-series prediction practical, but creates the problem of selecting good sizes and shapes for these patches, and ignores spatial dependence across patches.

To show how LICORS escapes this dilemma, we compare it to other forecasting techniques in a simulation. Using 100 replications of (5.16) – (5.18), with $n = 100$ points in space, and $T = 200$ steps in time, we compared LICORS, with and without pre-clustering, to (a) the empirical time-average of each spatial point; (b) a separate, univariate $AR(p)$ model for each point; a (c) separate $VAR(p)$ for each non-overlapping spatial patch of 5 points; and the true conditional expectation function.[1]

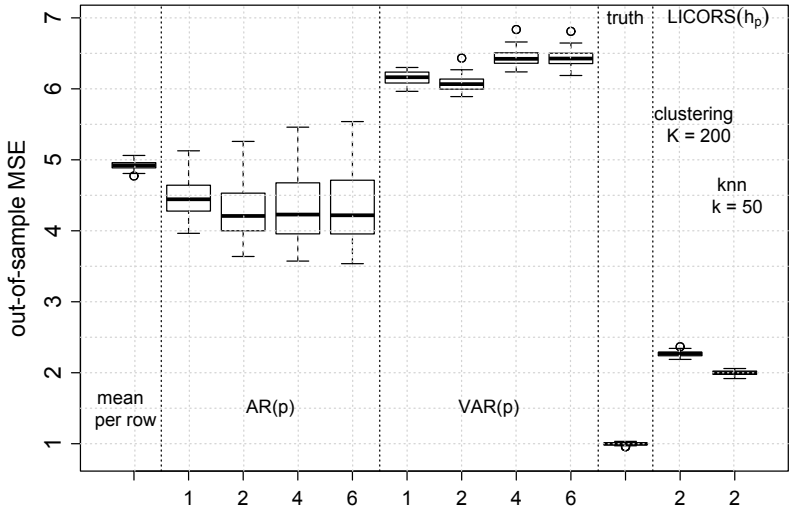
Figure 5.4 shows for each predictor the estimated mean squared error (MSE) for the in-sample (Fig. 5.4a) as well as out-of-sample (Fig. 5.4b) one-step ahead prediction error. Splitting up space while using standard methods appears not to help and may even hurt. LICORS performs best among all methods, once $h_p \geqslant 2$. While pre-clustering performs worse than direct estimation, it still predicts much better than the other methods.

Overall, LICORS with $h_p = 2$ gives the best forecasts, where $\alpha = 0.05$ was set in advance. At no point did we make an assumption about the number of predictive states or the shape of the conditional distribution. Even though the true predictive distributions are Gaussian, LICORS out-performed the parametric Gaussian models. Thus we expect to do even better on non-Gaussian fields.

1 The local VAR models were fit with Lasso regularization (Song and Bickel, 2011), as implemented in the `fastVAR` package (Wong, 2012). We also tried un-regularized VAR models, but they performed even worse.



(a) In-sample



(b) Out-of-sample

Figure 5.4: MSEs for LICORS and parametric competitors on (5.16)–(5.18). LICORS with pre-clustering used $K = 200$ clusters and varying past horizons; LICORS without pre-clustering use $k = 50$ neighbors and $h_p = 2$; both variants fixed $\alpha = 0.05$.

Even though we know the true light cone size in simulations, the "true" α can not be obtained directly. It controls the number of estimated predictive states: larger α implies less merging of clusters, and thus more number of predictive states; smaller α leads to more merging and hence less states.

1. Split dataset at its middle in time: $\mathcal{D}_{train} = \{X(\mathbf{r}, t)\}_{t=1}^{T/2}$ and $\mathcal{D}_{test} = \{X(\mathbf{r}, t)\}_{t=T/2+1}^{T}$
2. For each combination of control settings, do:
   a) Training: estimate predictive states from $\mathcal{D}_{train}$
   b) Test-set prediction: find predictive state of each PLC $\in \mathcal{D}_{test}$ and predict its FLC.
   c) Error: compare to the observed FLCs $\in \mathcal{D}_{test}$ and evaluate the loss.
3. Choose the control settings with the smallest test-set loss.

Figure 5.5: Cross-validation to choose control settings given data $\mathcal{D} = \{X(\mathbf{r}, t)\}_{t=1}^{T}$.

In practice, one does not know the true light cone size nor the true number of states; they are rather control settings which affect the predictive performance. As we can accurately measure predictive performance by out-of-sample MSE, we propose a cross-validation (CV) procedure to tune h_p and α.

5.4.2 *Cross-validation to Choose Optimal Control Settings*

A good method should learn invariant predictive structures, avoiding over-fitting to the accidents of the observed sample. Ideally, the method should estimate nearly the same predictive states from (almost) any two realizations of the same system, while still being sensitive to differences between distinct systems.

Cross-validation is the classic way to handle this sensitivity-stability trade-off, and we use a data-set splitting version of it here. We simply divide the data set at its mid-point in time, use its earlier half to find predictive states, and evaluate the states' performance on the data's later half; see Figure 5.5. (Assumption 5.1.1, of conditional stationarity, is important here.) While quite basic, simulations show that it does indeed find good control settings.

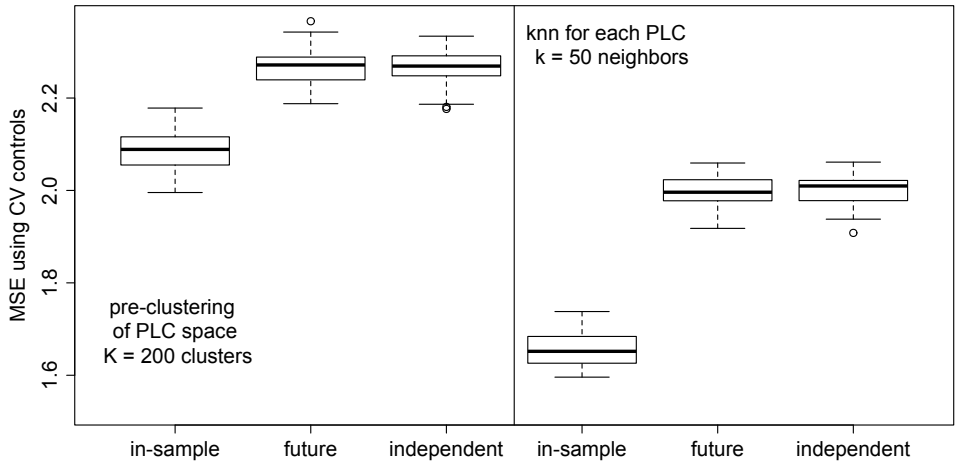


Figure 5.6: Cross-validation for LICORS: MSE, using the CV-picked control settings, on the first half of each realization ("in-sample"), on the second half ("future"), and on all of an independent realization ("independent").

Using the same realizations of the model system as in the forecasting competition, we tried all combinations of $h_p \in \{1, 2, 3\}$ and $\alpha \in \{0.3, 0.2, 0.15, 0.1, 0.05, 0.01, 0.001\}$. We picked the control settings to do well on the continuation of the sample realization, but since this is a simulation, we can also check that these settings perform well on an *independent* realization of the same process. Figure 5.6 compares, for the selected control settings, the in-sample MSE on the first half of each realization, the MSE on the second half, and the MSE on all of a completely independent realization, for both the direct and the pre-clustered versions of LICORS. (As before, direct estimation does a bit better than pre-clustering.) There is little difference between the MSEs on the continuation of the training data and on independent data, indicating little over-fitting to accidents of particular sample paths. (See §5.4.3 in the supplemental information for further details.) Notably, CV picked the optimal h_p, namely 2, on all 100 trials.

As expected, the smaller the value of α picked by CV, the more merging between clusters, and the smaller the number of states (see Supplemental Figure 5.7). Here, the true number of states $m = 7$, but both pre-clustering and direct estimation give much higher $\widehat{m}$ (10–30 with pre-clustering, 30–90 without). The gap appears to

be due to cross-validating pushing (in this context) for lower approximation error and more states, rather than fewer states and lower estimation error (§5.4.3 in the supplemental information). Having $\widehat{m}$ be substantially larger than m thus does not degrade out-of-sample predictions.

5.4.3 *Excess Risk, Test Size, and Number of Estimated States*

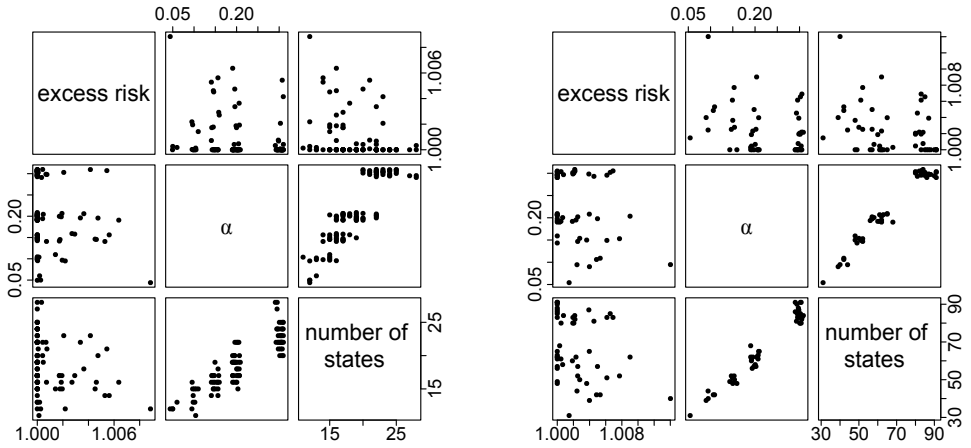
Figs. 5.7a and 5.7b show the expected relationship between α and the number of predictive states recovered $\widehat{m}$: smaller α leads to more merging, and fewer states. Here the true number of states $m = 7$, but both pre-clustering and direct estimation give much higher $\widehat{m}$. Thus LICORS pushes for more states and lower approximation error, rather than fewer states and lower estimation error. We can check this explanation by considering the ratio

$$\text{excess risk} := \frac{\text{MSE(sample } i+1\text{) using } (h_p, \alpha)_{i,CV_i}}{\text{MSE(sample } i+1\text{) using } (h_p, \alpha)_{i+1,\min}} \geqslant 1. \tag{5.19}$$

Recall that $(h_p, \alpha)_{i,CV_i}$ is chosen using only sample i, while $(h_p, \alpha)_{i+1,\min}$ is the minimizing pair after having evaluated the MSE on sample $i+1$. The best that any data-driven procedure could do would be to guess $(h_p, \alpha)_{i+1,\min}$ from sample i, so the excess risk is $\geqslant 1$, with equality only if CV picked the optimal control settings. The scatter-plots show that our CV procedure has an excess risk on the order of 10^{-2} compared to the oracle pair. Hence, even though $\widehat{m}$ is substantially larger than m, the difference is practically irrelevant for predictions.

5.4.4 *Discussion of the Simulations*

The simulations showed that LICORS outperforms standard forecasting techniques by a large margin, even though it presumes very little about the data source. Especially note that the out-of-sample MSE in Fig. 5.6 is still much lower than the best



(a) (b)

Figure 5.7: Relations between excess risk, test size, and the number of reconstructed states for LICORS: (a) selected α, number of estimated states, and excess risk (Eq. (5.19)) for pre-clustered LICORS; (b) the same for direct-estimation LICORS. α values in are jittered.

parametric in-sample MSE in Fig. 5.4a — even though it uses only half the sample size. The good performance of the CV procedure (Fig. 5.5) suggests that using it to find control settings in applications will avoid over-fitting.

In real applications N would typically on the order of millions (rather than merely 2×10^4), making pre-clustering essential computationally — at least until $\mathcal{O}(N^2)$ comparisons for millions of data points become tractable. Pre-clustering usually leads to a performance loss as it hides fine structures in the predictive distribution space (see also the remark below Lemma 5.2.7). However, the in-sample and out-of-sample MSE comparison showed that this performance loss is small compared to the gain over standard parametric methods, and further attenuated with CV.

5.5 SUMMARY

We present a new nonparametric forecasting method for data where continuous values are observed on a regular spatial grid at regular time-intervals. Our method,

light-cone reconstruction of causal states (LICORS), uses physical principles to identify predictive states which are local properties of the system, both in space and time. LICORS is completely nonparametric, discovering the number of predictive states and their predictive distributions automatically, and consistently under mild assumptions on the data-generating process. We provide an algorithm to implement our method, along with a cross-validation scheme to pick control settings. Simulations show that CV-tuned LICORS outperforms standard time series methods in forecasting challenging spatio-temporal dynamics.

6

THE STATISTICS OF LIGHT CONES AND PREDICTIVE STATES

You do not understand anything until you learn it more than one way.

Marvin Minsky

In this chapter I introduce a fully statistical model of light cones and predictive states, which bridges a gap between their origins in physics and purely data-driven, computational machine learning approaches for state-space recovery, thus making it more accessible to a wider (statistical) audience. Such a statistical model is more apt for solving general inference problems involving spatio-temporal data, since in a probabilistic, generative framework one can easily incorporate other statistical methodology. For example, one can perform probabilistic classification based on samples from spatio-temporal data.

This statistical model also yields a mixture model as well as hidden state interpretation of predictive states. I also show details on probabilistic forecasting given

new data (Section 6.4), and how to simulate a new realization given only observed data (Section 6.5).

6.1 A STATISTICAL PREDICTIVE STATES MODEL FOR SPATIO-TEMPORAL PROCESSES

Previous work on predictive states and pattern discovery started immediately with characterizing or estimating the conditional distributions $\mathbb{P}(L^+ \mid \ell^-)$. While this is obviously important for forecasting, a lot of scientific questions are not directly related to predicting the future.

For general statistical inference from a realization of a stochastic process $X_1, \ldots, X_{\tilde{N}}$, one is first and foremost interested in its joint distribution

$$\mathbb{P}(X_1, \ldots, X_{\tilde{N}}) . \tag{6.1}$$

Conditional predictive distributions of any sort can then be derived as needed.

Most results in previous chapters (in particular, Chapter 5) were derived from the product of the predictive PLC distributions

$$\prod_{i=1}^{N} \mathbb{P}\left(X_i \mid \ell_i^-\right) . \tag{6.2}$$

It is therefore natural to ask if previously described results and estimators only work for forecasting, or if they can also be used to solve general statistical inference problems.

In this section, I show that the limitation to the conditional predictive distribution is not restrictive since (6.2) is proportional to (6.1).

Remark 6.1.1 (Analogy to joint pdf of a Markov process)**.** *The argument is analogous to the factorization of the joint pdf of a Markov process of order one,*

$$\mathbb{P}(X_1, \dots, X_T) = \mathbb{P}(X_T \mid X_{T-1}, \dots, X_1)\, \mathbb{P}(X_{T-1}, \dots, X_1) \tag{6.3}$$

$$= \mathbb{P}(X_T \mid X_{T-1})\, \mathbb{P}(X_{T-1}, \dots, X_1), \tag{6.4}$$

where the second equality follows by the Markov property. By induction,

$$\mathbb{P}(X_1, \dots, X_T) = \mathbb{P}(X_1) \prod_{t=2}^{T} \mathbb{P}(X_t \mid X_{t-1}) \tag{6.5}$$

The joint pdf factorizes to a product of $T-1$ *conditional probabilities (which are typically easy to compute) times the initial marginal pdf of* X_1 *(which is typically more difficult to obtain).*

Let θ *be the parameter specifying the transition probabilities from* t *to* $t+1$*. For large* T *the marginal distribution becomes negligible, and we approximate the average log-likelihood with*

$$\frac{1}{T}\ell(\theta; X_1, \dots, X_T) = \frac{1}{T}\sum_{t=2}^{T} \log \mathbb{P}(X_t \mid X_{t-1}; \theta) + \frac{1}{T}\log \mathbb{P}(X_1; \theta) \tag{6.6}$$

$$= \frac{1}{T}\sum_{t=2}^{T} \log \mathbb{P}(X_t \mid X_{t-1}; \theta) + \mathcal{O}\big(1/T\big) \tag{6.7}$$

$$\approx \frac{1}{T}\sum_{t=2}^{T} \log \mathbb{P}(X_t \mid X_{t-1}; \theta) \tag{6.8}$$

We now show that an analogous result holds for the spatio-temporal process $X(\mathbf{r}, t)$*.*

For simplicity of notation, we choose the index set $i = 1, \dots, N$ in such a relation to the spatio-temporal grid $(\mathbf{s}, t)$, that the PLC of i_1 cannot contain X_{i_2} if $i_2 > i_1$.

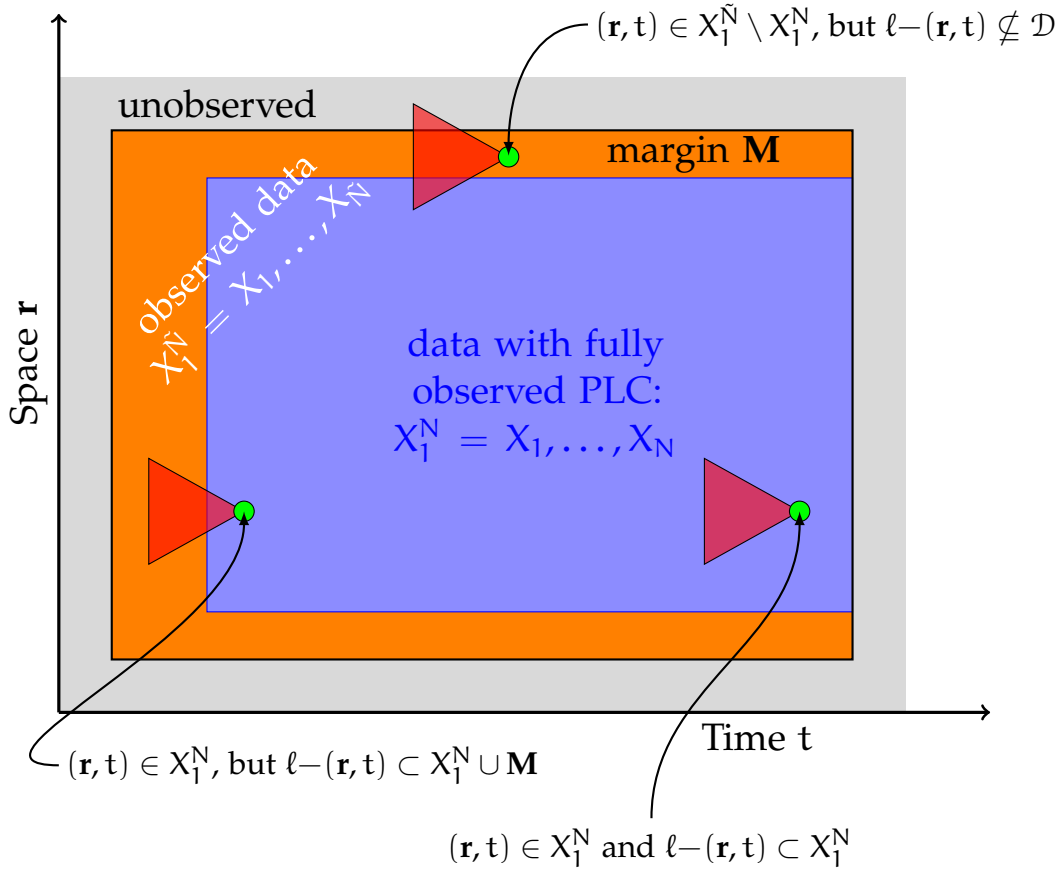


Figure 6.1: Margin of a spatio-temporal field in $(1+1)$D.

This can be guaranteed by iterating through space-time in increasing order over time (for fixed time the order in space does not matter). Formally,

$$(\mathbf{s},t)\,,\mathbf{s} \in \mathbf{S}, t \in \mathbb{T} \to \left(i_{(t-1)\cdot|\mathbf{S}|+1}, \ldots, i_{(t-1)\cdot|\mathbf{S}|+|\mathbf{S}|)}\right) = (t-1)\cdot|S| + (1,\ldots,|S|). \tag{6.9}$$

For consistent notation with the rest of this work, assume that we observed the process on an extended grid $\tilde{\mathbf{S}} \times \tilde{\mathbb{T}}$, where $\tilde{\mathbf{S}} \supset \mathbf{S}$ and $\tilde{\mathbb{T}} = \{-(h_p-1), \ldots, 0\} \cup \mathbb{T}$.

Let the extended field have $\tilde{N} > N$ observations, $X_1, \dots, X_{\tilde{N}}$. The margin $\mathbf{M}$ are all $X(\mathbf{s}, u), (\mathbf{s}, u) \in \tilde{\mathbf{S}} \times \tilde{\mathbb{T}}$, that do not have a fully observed PLC. Formally,

$$\mathbf{M} = \{X(\mathbf{s}, u), (\mathbf{s}, u) \in \tilde{\mathbf{S}} \times \tilde{\mathbb{T}} \mid \ell^-(\mathbf{s}, u) \notin X(\mathbf{r}, t), (\mathbf{r}, t) \in \mathbf{S} \times \mathbb{T}\}. \tag{6.10}$$

The size of $\mathbf{M}$ depends on the past horizon h_p as well as the speed of propagation c, $\mathbf{M} = \mathbf{M}(h_p, c)$.

In Figure 6.1, the extended field $X_1, \dots, X_{\tilde{N}}$ "lives" on the red *and* gray area, all points with a fully observed PLC, $X_1, \dots, X_N$, are on the red grid. The PLCs of points in the margin (gray) extend into the unobserved (blue) area; points in the red area have a PLC that lies fully in the red or partially in the gray, but never in the blue area. As can be seen in Fig. 6.1, the margin at each t is a constant fraction of space, thus overall $\mathbf{M}$ grows linearly with T; it does not grow with an increasing $\mathbf{S}$, but stays constant.

Proposition 6.1.2. *The joint pdf of the observable field* $\mathbb{P}(X_1, \dots, X_{\tilde{N}})$ *satisfies*

$$\mathbb{P}(X_1, \dots, X_{\tilde{N}}) = \mathbb{P}(\mathbf{M}) \prod_{i=1}^{N} \mathbb{P}\left(X_i \mid \ell_i^-\right). \tag{6.11}$$

Proof. For simplicity of notation, assume that $X_1, \dots, X_N$ are from the truncated (red) field, such that all their PLCs are observed (they may lie in $\mathbf{M}$), and the remaining $\tilde{N} - N$ X_js lie in $\mathbf{M}$ (with a PLC that is only partially observed). Furthermore, let $X_1^k := \{X_1, \dots, X_k\}$. Thus,

$$\begin{aligned}
\mathbb{P}\left(\{X(\mathbf{s}, t) \mid (\mathbf{s}, t) \in \tilde{\mathbf{S}} \times \tilde{\mathbb{T}}\}\right) &= \mathbb{P}\left(X_1^{\tilde{N}}\right) && (6.12)\\
&= \mathbb{P}\left(X_1^N, \mathbf{M}\right) && (6.13)\\
&= \mathbb{P}\left(X_1^N \mid \mathbf{M}\right) \mathbb{P}(\mathbf{M}) && (6.14)
\end{aligned}$$

The first term factorizes as

$$\mathbb{P}\left(X_1^N \mid \mathbf{M}\right) = \mathbb{P}\left(X_N \mid X_1^{N-1}, \mathbf{M}\right) \mathbb{P}\left(X_1^{N-1} \mid \mathbf{M}\right) \tag{6.15}$$

$$= \mathbb{P}\left(X_N \mid \ell_N^- \cup \{X_1^{N-1}, \mathbf{M}\} \setminus \{\ell_N^-\}\right) \mathbb{P}\left(X_1^{N-1} \mid \mathbf{M}\right) \tag{6.16}$$

$$= \mathbb{P}\left(X_N \mid \ell_N^-\right) \mathbb{P}\left(X_1^{N-1} \mid \mathbf{M}\right) \tag{6.17}$$

where the second-to-last equality follows since by (6.9), $\ell_N^- \subset \{X_k \mid 1 \leqslant k < N\} \cup \mathbf{M}$, and the last equality holds since X_i is conditional independent of the rest given its light cone (due to limits in information propagation over space-time).

By induction,

$$\mathbb{P}\left(X_1, \ldots, X_N \mid \mathbf{M}\right) = \prod_{j=0}^{N-1} \mathbb{P}\left(X_{N-j} \mid \ell_{N-j}^-\right) = \prod_{i=1}^{N} \mathbb{P}\left(X_i \mid \ell_i^-\right). \tag{6.18}$$

□

This shows that the conditional log-likelihood maximization we use for our estimators is equivalent (up to a constant $\mathbb{P}(\mathbf{M})$) to full joint maximum likelihood estimation (MLE). However, as $\mathbf{M}$ grows linearly with T the approximation does not converge to the full joint for $T \to \infty$. Rather we can think of the conditional log-likelihood approximation as a reduction in sample size (by a factor of $N/\tilde{N} \lessapprox 1$).

Proposition 6.1.2 shows that the focus on conditional predictive distributions does not restrict the applicability of the light cone model to forecasts only, but is in fact a generative model for any spatio-temporal process. Thus decomposition (6.11) can be used as a starting point for other statistical analysis of spatio-temporal data.

6.2 PREDICTIVE STATES AS OPTIMAL PARAMETERS IN A MIXTURE MODEL

Another way to understand predictive states is as the extremal distributions of an optimal mixture model (Lauritzen, 1974, 1984).

To predict any variable L^+, we have to know its distribution $\mathbb{P}(L^+)$. If, as often happens, that distribution is very complicated, we may try to decompose it into a mixture of simpler "base" or "extremal" distributions, $\mathbb{P}(L^+ \mid \theta)$, with mixing weights $\pi(\theta)$,

$$\mathbb{P}\left(L^+\right) = \int \pi(\theta)\mathbb{P}\left(L^+ \mid \theta\right) d\theta \; . \tag{6.19}$$

The familiar Gaussian mixture model, for instance, makes the extremal distributions to be Gaussians (with θ indexing both expectations and variances), and makes the mixing weights $\pi(\theta)$ a combination of delta functions, so that $\mathbb{P}(L^+)$ becomes a weighted sum of finitely-many Gaussians.

The conditional predictive distribution of $L^+ \mid \ell^-$ in (6.19) is a weighted average over the extremal conditional distributions $\mathbb{P}(L^+ \mid \theta, \ell^-)$,

$$\mathbb{P}\left(L^+ \mid \ell^-\right) = \int \pi(\theta|\ell^-)\mathbb{P}\left(L^+ \mid \theta, \ell^-\right) d\theta \tag{6.20}$$

This only makes the forecasting problem harder, unless

$$\mathbb{P}\left(L^+ \mid \theta, \ell^-\right)\pi(\theta \mid \ell^-) = \mathbb{P}\left(L^+ \mid \hat{\theta}(\ell^-)\right)\delta(\theta - \hat{\theta}(\ell^-)), \tag{6.21}$$

that is, unless $\hat{\theta}(\ell^-)$ is a predictively sufficient statistic for L^+. The most parsimonious mixture model is the one with the minimal sufficient statistic, $\theta = \epsilon(\ell^-)$. This shows that predictive states are the best "parameters" in (6.19) for optimal forecasting. Using them,

$$\mathbb{P}\left(L^+\right) = \sum_{j=1}^{K} \mathbb{P}\left(\epsilon(\ell^-) = s_j\right)\mathbb{P}\left(L^+ \mid \epsilon(\ell^-) = s_j\right) \tag{6.22}$$

$$= \sum_{j=1}^{K} \pi_j(\ell^-) \cdot p_j\left(L^+\right) \; , \tag{6.23}$$

where $\pi_j(\ell^-)$ is the probability that the predictive state of ℓ^- is s_j, and $p_j(L^+) := \mathbb{P}\left(L^+ \mid S = s_j\right)$. Since each light cone has a unique predictive state,

$$\pi_j(\ell^-) = \begin{cases} 1, & \text{if } \epsilon(\ell^-) = s_j, \\ 0 & \text{otherwise.} \end{cases} \tag{6.24}$$

The predictive distribution given the PLC ℓ_i^- is just

$$\mathbb{P}\left(L^+ \mid \ell_i^-\right) = \sum_{j=1}^{K} \pi_j(\ell_i^-) \cdot p_j\left(L^+ \mid \ell_i^-\right) = p_{\epsilon(\ell_i^-)}\left(L^+\right). \tag{6.25}$$

Now the forecasting problem simplifies to mapping ℓ_i^- to its predictive state, $\epsilon(\ell_i^-) = s_j$; the appropriate distribution-valued forecast is $p_j(L^+)$, and point forecasts are derived from it as needed.

This mixture-model point of view highlights how prediction benefits from grouping points by their predictive consequences, rather than by spatial proximity (as a Gaussian mixture would do). For us, this means clustering PLC configurations according to the similarity of their predictive distributions, not according to (say) the Euclidean geometry. We thus learn a new geometry for the system, which is optimized for forecasting.

6.3 PREDICTIVE STATES AS HIDDEN VARIABLES

Causa latet, vis est notissima.
(The cause is hidden, but the result is well known.)

Ovidius

Recall that we are ultimately interested in predicting X_i from a given PLC configuration ℓ_i^-. To do this efficiently we assume there exists a deterministic mapping,

$\epsilon(\ell_i^-)$, with properties (3.6) and (3.7). Thus, knowing ϵ, the joint predictive distribution of $X(\mathbf{r}, t)$ conditioned on the margin simplifies to

$$\mathbb{P}(X_1, \ldots, X_N \mid \mathbf{M}; \epsilon) = \prod_{i=1}^{N} \mathbb{P}\left(X_i \mid \ell_i^-; \epsilon(\ell_i^-)\right) \tag{6.26}$$

$$= \prod_{i=1}^{N} \mathbb{P}\left(X_i \mid \epsilon(\ell_i^-)\right), \tag{6.27}$$

where the second equality follows by Definition 3.3.1 and Eq. (3.7). Any particular ϵ implicitly specifies the number of predictive states K and all K predictive distributions $\mathbb{P}\left(X_i \mid \epsilon(\ell_i^-)\right)$. However, in practice only X_i and ℓ_i^- are observed; the mapping ϵ is exactly what we are trying to estimate.

Since the mapping ϵ and the equivalence classes/predictive states $\epsilon(\ell^-)$ are in a one-to-one relation, Shalizi (2003) and follow-up literature do not formally distinguish between them. Also the hard LICORS estimator from Chapter 5 built directly on these equivalence classes. For a proper statistical modeling, however, it is important to keep the distinction between the predictive state space $\mathcal{S}$ and the mapping $\epsilon : \ell_i^- \to \mathcal{S}$. Here we make the state variable S_i explicit and by this means naturally obtain a hidden variable model for predictive states.

Let $\mathcal{S} = \{s_1, \ldots, s_K\}$ be the predictive state space, and let S_i be the predictive state at coordinate i; hence

$$\epsilon : \ell_i^- \mapsto S_i \in \mathcal{S} = \{s_1, \ldots, s_K\}. \tag{6.28}$$

Since ϵ is unknown, S_i is a hidden (random) variable.

Using this latent variable approach (6.27) can be equivalently written as

$$\prod_{i=1}^{N} \mathbb{P}(X_i \mid S_i) = \prod_{i=1}^{N} \sum_{j=1}^{K} \mathbf{1}(S_i = s_j)\, \mathbb{P}(X_i \mid S_i = s_j) \tag{6.29}$$

The key insight is that (6.29) is the probability density function (pdf) of a K component mixture model with *complete data*, and $\mathbf{1}\left(S_i = s_j\right)$ is a randomized version of the unknown mapping $\epsilon : \ell_i^- \mapsto S_i$.

The observable pdf can be obtained by integrating over the predictive state variable

$$\mathbb{P}\left(X_1, \ldots, X_N \mid \mathbf{M}\right) = \prod_{i=1}^{N} \mathbb{P}\left(X_i \mid \ell_i^-\right) \tag{6.30}$$

$$= \prod_{i=1}^{N} \sum_{j=1}^{K} \mathbb{P}\left(S_i = s_j \mid \ell_i^-\right) \mathbb{P}\left(X_i \mid \epsilon(\ell_i^-) = s_j\right). \tag{6.31}$$

In Chapter 7, I present a nonparametric EM algorithm for predictive state recovery and use soft- and hard thresholding of $\mathbb{E}\left(\mathbf{1}\left(S_i = s_j\right) \mid \mathcal{D}\right)$ as a probabilistic estimate of ϵ.

6.4 DISTRIBUTION FORECASTS GIVEN NEW DATA

Suppose that after fitting a model $\widehat{\epsilon}^*$ to data $\mathcal{D}$ we are asked to predict $\tilde{X}$ based on a new $\tilde{\ell}^-$.[1] Integrating out S_i yields a mixture distribution

$$\mathbb{P}\left(\tilde{X} = x \mid \tilde{\ell}^-\right) = \sum_{j=1}^{K} \mathbb{P}\left(\tilde{S} = s_j \mid \tilde{\ell}^-\right) \cdot \mathbb{P}\left(\tilde{X} = x \mid \tilde{S} = s_j\right). \tag{6.32}$$

As $\mathbb{P}\left(\tilde{X} = x \mid \tilde{S} = s_j\right)$ does not depend on $\tilde{\ell}^-$, we do not have to re-estimate them for each $\tilde{\ell}^-$, but can use density estimates from the training data. The mixture weights $\tilde{w}_j := \mathbb{P}\left(\tilde{S} = s_j \mid \tilde{\ell}^-\right)$ are in general different for each PLC and can again

1 In Section 7.4 we propose one particular way to estimate the quantities presented below. In this section I just outline the general methodology without giving estimation details.

be estimated using Bayes' rule (with the important difference that now we only condition on $\tilde{\ell}^-$, not on $\tilde{X}$):

$$\widehat{\tilde{w}}_j^* \propto \widehat{\mathbb{P}}\left(\tilde{\ell}^- \mid \tilde{S} = s_j; \widehat{\epsilon}^*\right) \times \widehat{\mathbb{P}}\left(\tilde{S} = s_j; \widehat{\epsilon}^*\right) \tag{6.33}$$

After re-normalization of $\widehat{\tilde{\mathbf{w}}} = (\widehat{\tilde{w}}_1^*, \ldots, \widehat{\tilde{w}}_K^*)$, the predictive distribution (6.32) can be estimated via

$$\widehat{\mathbb{P}}\left(\tilde{X} = x \mid \tilde{\ell}^-\right) = \sum_{j=1}^{K} \widehat{\tilde{w}}_j^* \cdot \widehat{\mathbb{P}}\left(\tilde{X} = x \mid \widehat{\epsilon}^*\right), \tag{6.34}$$

where $\widehat{\mathbb{P}}\left(\tilde{X} = x \mid \widehat{\epsilon}^*\right)$ are state-dependent pdf estimates, e.g., a kernel density estimate (KDE) as we use in (7.15).

A point forecast can then be obtained by a weighted combination of point estimates in each component (e.g., weighted mean), or by the mode of the full distribution. In the simulations we use the weighted average from each component as the prediction of $\tilde{X}$.

6.5 SIMULATING SPATIO-TEMPORAL DATA

Recall that all we need to simulate a spatio-temporal field is i) the predictive state space $S(\mathbf{r}, t)$ with its conditional FLC distributions, ii) and the mapping from PLCs to the state space. For example, (5.16) – (5.18) fully specify i) and ii) and we can therefore simulate a field as in Fig. 1.1a.

In most real-world problems, however, researchers typically encounter the opposite: they only observe one (or a few) realization of the system, but the underlying dynamics are (yet) unknown. Thus one cannot simply simulate new realizations, but has to perform more experiments.

Set t_{max} for the maximum time steps of the simulation:

0. Set initial conditions $\{X(\cdot, -(h_p - 1)), \ldots, X(\cdot, 0)\}$. Set $t = 1$.
1. Fetch PLC configurations of field at time t: $P_t = \{\ell^-(\mathbf{r}, t)\}_{\mathbf{r}\in\mathbf{S}}$
2. For each $\ell^- \in P_t$:
   a) Draw the state s_j from the multinomial $S \sim \mathbb{P}\left(S = s_j \mid \ell^-\right)$ using the estimates from (6.33).
   b) Draw one sample from $X \sim \mathbb{P}\left(x \mid S = s_j\right)$ using (7.15) and assign this draw to the $X(\mathbf{r}, t)$ corresponding to PLC $\ell^- = \ell^-(\mathbf{r}, t)$.
3. If $t < t_{max}$, set $t = t + 1$ and go to step 1. Otherwise return simulated field $\{X(\cdot, 1), \ldots, X(\cdot, t_{max})\}$.

Figure 6.2: Simulate new observation from spatio-temporal field. True distributions and mappings can be replaced by (LICORS) estimates and then one can simulate new data from estimated dynamics (see also Section 7.5).

Since the statistical model of predictive states (and their estimates) specify the entire distribution, and not only - like many other methods - the conditional mean, it is possible to simulate a field from the estimated model. Figure 6.2 outlines this simulation procedure.

Being able to estimate these dynamics and the predictive state space can in principle make a lot of expensive, time- and labor-intensive experimental studies, if not obsolete, then at least much more manageable and easier to plan. Depending on the variety of dynamics, a statistical model can learn spatio-temporal dynamics already after a few experiments. Once learned, researchers can use the estimates to simulate their system from different starting conditions within a couple of minutes rather than waiting for their experiments to finish in hours, days, or months.

In Chapter 5, I presented a consistent estimator to learn the dynamics. In Chapter 7. I introduce a fully probabilistic algorithm for more general and accurate estimation of predictive states; in Section 7.5. I demonstrate the accuracy of such an observation-based simulation.

6.6 LOCAL STATISTICAL COMPLEXITY REVISITED

The prediction problem simplifies because once we *know* the predictive state of a PLC, we can do optimal prediction. For pattern discovery however, predictions per se are not that important; rather it is the landscape of different predictors that reveal interesting dynamics. In other words, for pattern discovery the particular predictive state of a PLC is not critical, but the probability of landing in a given state is. Pattern discovery should therefore not depend on the conditional distribution *of each* state, $\mathbb{P}\left(L^+ \mid s_j\right)$, but on the distribution *over* states, $\mathbb{P}\left(S_i = s_j\right)$.

This is exactly what LSC does, since for a discrete state space (4.1) is equivalent to

$$\mathcal{C}(\mathbf{r}, t) = -\log_2 \mathbb{P}\left(S_i = s_j\right), \tag{6.35}$$

the entropy of the hidden state variable S_i. Following the usual interpretation of entropy (Cox, 2001; Shannon, 1948), $\mathcal{C}(\mathbf{r}, t)$ informs us about the interesting events in the predictive state space: states with low probability show interesting patterns; high-probability states are less interesting.

Since $\mathcal{C}(\mathbf{r}, t)$ is the entropy of the event that the predictive state process $S(\mathbf{r}, t)$ takes on the observed state at $(\mathbf{r}, t)$, another interpretation of $\widehat{\mathcal{C}}(\mathbf{r}, t)$ is the surprise of seeing the particular dynamics at $(\mathbf{r}, t)$ after having seen the data.

7

MIXED LICORS: A NONPARAMETRIC EM ALGORITHM FOR PREDICTIVE STATE RECONSTRUCTION

It is very certain that, when it is not in our power to determine what is true, we ought to act according to what is most probable.

René Descartes, *Discours de la Méthode*

Equipped with the statistical model of predictive states, I now introduce a probabilistic version of LICORS (Chapter 5). Recall that LICORS uses a combination of initial pre-clustering with K-means followed by agglomerative clustering. Thus overall it remains a hard-clustering estimator (hard LICORS). However, experience with other clustering problems shows that soft threshold often predicts much better than hard threshold. Famously, while k-means (Lloyd, 1982) is very fast and robust, the expectation maximization (EM) algorithm (Dempster, Laird, and Rubin, 1977) in Gaussian mixture models gives better clustering results. Moreover, the mixture model framework admits of a probabilistic interpretation of clustering, and the assignment of novel observations to clusters.

With this inspiration, we introduce *mixed LICORS*, a soft-thresholding version of (hard) LICORS. Mixed LICORS uses the mixture-model setting from Chapter 6, where the predictive states correspond to the optimal mixing weights on extremal distributions, which are themselves optimized for forecasting. Our proposed nonparametric EM-like algorithm then follows naturally.

In Section 7.1, I present the nonparametric EM algorithm. Section 7.2 shows an automatic selection of the number of components (= predictive states) as well as penalization strategies to obtain sparse mixture weights. While the former is restricted to our particular forecasting setting, the latter is generally applicable. In Section 7.4, we show how to forecast based on an EM estimate. We then evaluate its performance on simulated data and demonstrate the large improvements in out-of-sample prediction compared to hard LICORS (Section 7.6).

7.1 EM ALGORITHM FOR PREDICTIVE STATE ESTIMATION

The EM algorithm we propose is based on the idea that the predictive state is a hidden variable, S_i, taking values in the finite state space $\mathcal{S} = \{s_1, \ldots, s_K\}$, and the

mixture weights of PLC ℓ_i^- are the soft-threshold version of the mapping $\epsilon(\ell_i^-)$. It is this hidden variable interpretation of predictive states that we use to estimate the minimal sufficient statistic ϵ, the hidden state space S_i, and the predictive distributions $\mathbb{P}(X_i \mid S_i)$. In this light cone and predictive state setting the hidden state variable and the "parameter" play the same role, since ϵ maps to $\mathcal{S}$. As we will see below this results in very similar E and M steps. Figure 7.1 gives an overview of the proposed algorithm.

The complete data log-likelihood can be obtained from (6.29) as (we omit the additive constant $\log \mathbb{P}(\mathbf{M})$)

$$\ell(\epsilon; \mathcal{D}, S_1^N) = \sum_{i=1}^{N} \log \left(\sum_{j=1}^{K} \mathbf{1}\left(S_i = s_j\right) \mathbb{P}\left(X_i \mid \epsilon(\ell_i^-) = s_j\right) \right) \tag{7.1}$$

$$= \sum_{i=1}^{N} \sum_{j=1}^{K} \mathbf{1}\left(S_i = s_j\right) \log \mathbb{P}\left(X_i \mid \epsilon(\ell_i^-) = s_j\right), \tag{7.2}$$

where $S_1^N := \{S_1, \ldots, S_N\}$ and the second equality follows since $\mathbf{1}\left(S_i = s_j\right) = 1$ for one and only one j, and 0 otherwise.

The "parameters" in (7.2) are ϵ and K; X_i and ℓ_i^- are observed, and S_i is a hidden variable. The optimal mapping $\epsilon : L^- \to \mathcal{S}$ is the one that maximizes (7.2):

$$\epsilon^* = \arg\max_{\epsilon} \ell(\epsilon; \mathcal{D}, S_1^N). \tag{7.3}$$

Without any constraints on K or ϵ the maximum is obtained for $K = N$ and $\epsilon(\ell_i^-) = \ell_i^-$; "the most faithful description of the data is the data".[1] As this tells us nothing about the underlying dynamics, we must put some constraints on K and/or ϵ to get a useful solution. For now, assume that $K \ll N$ is fixed and we only have to estimate ϵ; in Section 7.2.1, we will give a data-driven procedure to choose K.

1 On the other extreme is a field with only $K = 1$ predictive state, i.e., the iid case.

Input:

$\mathcal{D}$: data $\{X_i, \ell_i^-\}_{i=1}^N$

$K \in \mathbb{N}$: maximum number of states (starting value)

0. **Initialization:** Set $n = 0$. Split data in training and test set, $\mathcal{D}_{train}$ and $\mathcal{D}_{test}$. Assign each PLC from $\mathcal{D}_{train}$ to one state uniformly at random from $\{s_1, \ldots, s_K\} \to \widehat{\epsilon}^{(0)}$. Set $\widehat{\mathbf{W}}^{(0)}$ to a 0/1 matrix, with 1 in row i and column j, if $\widehat{\epsilon}^{(0)}(\ell_i^-) = s_j$, and 0 otherwise.

1. **E-step:** Update weights $\widehat{\mathbf{W}}^{(n+1)}$ in (7.14) using the conditional distribution of S given $\mathcal{D}$ and current estimate $\widehat{\mathbf{W}}^{(n)}$. Evaluate expected log-likelihood (7.9).

2. **Approximate M-step:** Update mixture components $\mathbb{P}\left(x_i \mid S_i = s_j\right)$ with $\widehat{\mathbf{W}}^{(n+1)}$ using (7.15).

3. **Out-of-sample Prediction:** Evaluate out-of-sample MSE for $\widehat{\epsilon}^{(n+1)}$ and $\widehat{\mathbf{W}}^{(n+1)}$ by predicting FLCs from PLCs in $\mathcal{D}_{test}$. Set $n = n + 1$.

4. **Temporary convergence:** Iterate 1 - 3 until

$$\|\widehat{\mathbf{W}}^{(n)} - \widehat{\mathbf{W}}^{(n-1)}\| < \delta \quad \text{or} \quad \|\ell(\widehat{\mathbf{W}}^{(n)}; \mathcal{D}) - \ell(\widehat{\mathbf{W}}^{(n-1)}; \mathcal{D})\| < \delta^*. \tag{7.4}$$

Figure 7.1: Mixed LICORS: nonparametric EM algorithm for predictive state recovery in spatio-temporal data.

7.1.1 *Nonparametric Likelihood Approximation*

To solve (7.3) with $K \ll N$ we need to evaluate (7.2) for candidate solutions ϵ. Doing this directly is not possible since

i) the right hand side (RHS) of (7.2) depends on the unknown S_i, and

ii) the component distributions $\mathbb{P}\left(X_i \mid \epsilon(\ell_i^-) = s_j\right)$ can not be evaluated directly unless we use a parametric model. Since predictive distributions can have arbitrary shapes, we do not want to put restrictions on $\mathbb{P}\left(X_i \mid \epsilon(\ell_i^-) = s_j\right)$ but use nonparametric methods.

We solve i) by using a nonparametric variant of the expectation maximization (EM) algorithm (Dempster et al., 1977); for ii) we follow the nonparametric EM literature (Benaglia and Hunter, 2009a; Bordes, Chauveau, and Vandekerkhove, 2007; Hall, Neeman, Pakyari, and Elmore, 2005) and approximate $\mathbb{P}\left(X_i \mid \epsilon(\ell_i^-) = s_j\right)$ in the log-likelihood with kernel density estimators (KDEs) using a previous estimate $\widehat{\epsilon}^{(n)}$. That is we approximate (7.2) with

$$\widehat{\ell}^{(n)}(\epsilon; \mathcal{D}, S_1^N) := \widehat{\ell}(\epsilon; \mathcal{D}, S_1^N, \epsilon^{(n)}) = \sum_{i=1}^{N} \sum_{j=1}^{K} \mathbf{1}\left(S_i = s_j\right) \log \widehat{f}^{(n)}\left(X_i \mid \epsilon(\ell_i^-) = s_j\right), \tag{7.5}$$

where the expression for $\widehat{f}^{(n)}\left(X_i \mid \epsilon(\ell_i^-) = s_j\right)$ is given below in (7.15).

7.1.2 *Expectation Step*

The E-step requires the expected log-likelihood

$$Q(\epsilon \mid \epsilon^{(n)}) = \mathbb{E}_{S \mid \mathcal{D}; \epsilon^{(n)}} \ell(\epsilon; \mathcal{D}, S_1^N), \tag{7.6}$$

where expectation is taken with respect to $\mathbb{P}\left(S_i = s_j \mid \mathcal{D}; \epsilon^{(n)}\right)$, the conditional distribution of the hidden variable S_i given the data $\mathcal{D}$ and the current estimate $\epsilon^{(n)}$. Using (7.2) we obtain

$$Q(\epsilon \mid \epsilon^{(n)}) = \sum_{i=1}^{N} \sum_{j=1}^{K} \mathbb{P}\left(S_i = s_j \mid X_i, \ell_i^-; \epsilon^{(n)}(\ell_i^-)\right) \times \log \mathbb{P}\left(X_i \mid \epsilon^{(n)}(\ell_i^-) = s_j\right). \tag{7.7}$$

As for $\ell(\epsilon; \mathcal{D})$, we also replace the component distributions with the nonparametric KDEs and get an approximate expected log-likelihood

$$\widehat{Q}^{(n)}(\epsilon \mid \epsilon^{(n)}) = \mathbb{E}_{S|\mathcal{D},\epsilon^{(n)}} \widehat{\ell}^{(n)}(\epsilon; \mathcal{D}, S_1^N) \tag{7.8}$$

$$= \sum_{i=1}^{N} \sum_{j=1}^{K} \mathbb{P}\left(S_i = s_j \mid X_i, \ell_i^-; \epsilon^{(n)}(\ell_i^-)\right) \times \log \widehat{f}^{(n)}\left(X_i \mid \epsilon^{(n)}(\ell_i^-) = s_j\right) \tag{7.9}$$

The conditional distribution of S_i given its FLC and PLC, $\{X_i, \ell_i^-\}$, comes from Bayes's rule,

$$\mathbb{P}\left(S_i = s_j \mid X_i, \ell_i^-\right) \propto \mathbb{P}\left(X_i, \ell_i^- \mid S_i = s_j\right) \mathbb{P}\left(S_i = s_j\right) \tag{7.10}$$

$$= \mathbb{P}\left(X_i \mid S_i = s_j\right) \mathbb{P}\left(\ell_i^- \mid S_i = s_j\right) \mathbb{P}\left(S_i = s_j\right), \tag{7.11}$$

where (7.11) holds by conditional independence of X_i and ℓ_i^- given the state S_i.

For brevity, let $w_{ij} := \mathbb{P}\left(S_i = s_j \mid X_i, \ell_i^-\right)$, forming an $N \times K$ weight matrix $\mathbf{W}$, whose rows are probability distributions over states. This $\mathbf{w}_i$ is the soft-thresholding version of $\epsilon(\ell_i^-)$, so we can write the expected log-likelihood in terms of $\mathbf{W}$,

$$\widehat{Q}^{(n)}(\mathbf{W} \mid \widehat{\mathbf{W}}^{(n)}) = \sum_{i=1}^{N} \sum_{j=1}^{K} w_{ij} \times \log \mathbb{P}\left(X_i \mid \widehat{\mathbf{W}}_j^{(n)}\right). \tag{7.12}$$

The current $\widehat{\mathbf{W}}^{(n)}$ can be used to update (conditional) probabilities in (7.11) by

$$\widehat{w}_{ij}^{(n+1)} \propto \widehat{\mathbb{P}}\left(X_i \mid S_i = s_j; \widehat{\mathbf{W}}^{(n)}\right) \times \widehat{\mathbb{P}}\left(\ell_i^- \mid S_i = s_j; \widehat{\mathbf{W}}^{(n)}\right) \times \widehat{\mathbb{P}}\left(S_i = s_j; \widehat{\mathbf{W}}^{(n)}\right) \tag{7.13}$$

$$= \widehat{f}(x_i \mid S_i = s_j; \widehat{\mathbf{W}}^{(n)}) \times \mathcal{N}\left(\ell_i^- \mid \widehat{\mu}_j^{(n)}, \widehat{\Sigma}_j^{(n)}; \widehat{\mathbf{W}}^{(n)}\right) \times \frac{\widehat{N}_j^{(n)}}{N}, \tag{7.14}$$

where i) $\widehat{N}_j^{(n)} = \sum_{i=1}^{N} \widehat{\mathbf{W}}_{ij}^{(n)}$ is the effective sample size of state s_j, ii) $\widehat{\mu}_j^{(n)}$ and $\widehat{\mathbf{\Sigma}}_j^{(n)}$ are weighted mean and covariance matrix estimators of the PLCs using the jth column of $\widehat{\mathbf{W}}^{(n)}$, and iii) the FLC distribution is estimated with a weighted[2] KDE (wKDE)

$$\widehat{f}(x \mid S_i = s_j; \widehat{\mathbf{W}}^{(n)}) = \frac{1}{\widehat{N}_j^{(n)}} \sum_{i=1}^{N} \widehat{\mathbf{W}}_{ij}^{(n)} K_{h_j}(\|x_i - x\|), \tag{7.15}$$

where the weights are again the jth column of $\widehat{\mathbf{W}}^{(n)}$, and $K_{h_j}(\|x_i - x\|)$ is a kernel function with a state-dependent bandwidth h_j. For all our numerical calculations we use a Gaussian kernel in the R function `density()`. To obtain a good, cluster-adaptive bandwidth h_j we only use those x_i with $j = \arg\max_k w_{ik}$ in `bw.ndr0()` (hard-thresholding of weights; see also Benaglia and Hunter (2009b)). After estimation, each $\widehat{\mathbf{w}}_i$ in (7.14) must be normalized, $\widehat{w}_{ij}^{(n+1)} \leftarrow \frac{\widehat{w}_{ij}^{(n+1)}}{\sum_{j=1}^{K} \widehat{w}_{ij}^{(n+1)}}$.

Ideally, we would use a nonparametric estimate for the PLC distribution, e.g., forest density estimators (Chow and Liu, 1968; Liu, Xu, Gu, Gupta, Lafferty, and Wasserman, 2011). Currently, however, such estimators are too slow to handle many iterations at large N, so we model state-conditional PLC distributions as multivariate Gaussians. Simulations suggest that this is often adequate in practice.

7.1.3 *Approximate Maximization Step*

In a parametric model the M-step would solve

$$\epsilon^{(n+1)} = \arg\max_{\epsilon} \widehat{Q}^{(n)}(\epsilon \mid \epsilon^{(n)}), \tag{7.16}$$

to improve the estimate. Starting from an initial guess $\epsilon^{(0)}$, the EM algorithm iterates (7.6) and (7.16) until convergence.

2 We also tried a hard-threshold estimator, but we found that the soft-threshold KDE performed better.

In nonparametric problems, finding an $\epsilon^{(n+1)}$ that increases $\widehat{Q}^{(n)}(\epsilon \mid \epsilon^{(n)})$ is difficult, since wKDEs with non-zero bandwidth are not maximizing the likelihood; they are not even guaranteed to increase it. Optimizing (7.9) by brute force is not computationally feasible either, as it would mean searching K^N state assignments (see also Bordes et al., 2007).

However, in our particular setting the parameter space and the expectation of the hidden variable are the same, in the sense that $\widehat{\mathbf{w}}_i$ is a soft-thresholding version of $\epsilon(\ell_i^-)$. Furthermore, none of the estimates above requires a deterministic ϵ mapping, but they are all weighted MLEs or KDEs. Thus, like Benaglia and Hunter (2009a), we take the weights from the E-step, $\widehat{\mathbf{W}}^{(n+1)}$, to update each component distribution using (7.15). This in turn can then be plugged into (7.5) to update the likelihood function, and in (7.14) for the next E-step.

The wKDE update does not solve (7.16) nor does it provably increase the log-likelihood (although in simulations it often does so). We thus use cross-validation (CV) to select the best $\widehat{\mathbf{W}}^*$, and henceforth do not rely on an ever increasing log-likelihood as the usual stopping rule in EM algorithms (see Section 7.6 for details).

7.2 EXTENSIONS OF THE EM

We propose two additional modifications from standard EM algorithms that simplify interpretation of the results and also improve predictive performance: a) after convergence to a (local) optimum we merge the two closest components, where we measure closeness in distribution space (either by distance metric or by a two sample test); b) we impose sparsity on the mixture weights.

With step a) we incorporate an automatic selection of K, which is a key challenge in fitting mixture models (Biernacki, Céleux, and Govaert, 1999; Biernacki and Govaert, 1998; Tibshirani, Walther, and Hastie, 2000). Step b) facilitates interpretability of the results, as ideal cluster assignments are usually preferred over mixtures of multiple clusters.

7.2.1 *Data-driven Choice of* K*: Merge Predictive States To Obtain Minimal Sufficiency*

The advantage of the mixture model in (6.22) is that predictive states have by definition similar conditional distributions. Since conditional densities can be tested for equality by a nonparametric two sample test (or using a distribution metric), we can merge classes if they are close. We propose a data-driven automatic selection of K, which solves this key challenge in fitting mixture models: 1) start with a sufficiently large number of clusters, $K_{max} < N$; 2) test for equality of distribution each time the EM reaches a (local) optimum; 3) merge until $K = 1$ (iid case) – step 6 in Fig. 7.1; 4) choose the best model $\widehat{\mathbf{W}}^*$ by CV.

The stopping criterion can be set in two ways: i) either set a significance level $0 < \alpha < 1$ (or minimum distance $d_{min} > 0$) and stop merging once equality of distributions can be rejected (or distance is larger than d_{min}), ii) or merge until $K = 1$ (iid case) and then use cross-validation (CV) to choose an optimal K. Both approaches eventually stop the algorithm and give an optimal number of clusters. For the simulation we use the second approach, as it does not require the additional nuisance parameter α (or d_{min}).

7.2.2 *Induce Sparsity to Mixture Weights for Identifiability*

Even though none of the estimates and forecasts we derive requires a unique state space assignment, it is often desirable to have a unique label for each observation - if only for easier interpretation and visualization. We say that mixture weights are *sparse* if most of the K entries are 0, leaving only a couple of clusters in the weights. Optimally, we would like one entry to equal one, the remaining $K-1$ entries should equal zero. If this is the case, then the weight vector $\mathbf{w}_i$ uniquely maps sample i

Input:

$\widehat{\mathbf{W}}^{*}_{EM}$: EM solution from step 4 in Fig. 7.1.

$\lambda \geqslant 0$: sparsity inducing penalty parameter

5. **Sparsity:** If $\lambda > 0$, move weights to sparser solution using (7.26). Iterate until $R(\widehat{\mathbf{W}}^{(n+1)}_{EM,sparse}) > R(\widehat{\mathbf{W}}^{(n)}_{EM,sparse})$.
6. **Merging-step:** Estimate pairwise distances

$$\widehat{d}_{jk} = \text{dist}\left(\widehat{f}^{(n)}(x \mid S = s_j), \widehat{f}^{(n)}(x \mid S = s_k)\right) \text{ for all } j, k = 1, \ldots, K, \quad (7.18)$$

where $\text{dist}(f, g)$ is a distance measure (or a two sample test) for distributions f and g.

a) If $K > 1$, choose $(j^{(\min)}, k^{(\min)}) = \arg\min_{j \neq k} \widehat{d}_{jk}$ and merge corresponding columns of $\mathbf{W}^{(n)}$

$$\mathbf{W}^{(n)}_{j^{(\min)}} \leftarrow \mathbf{W}^{(n)}_{j^{(\min)}} + \mathbf{W}^{(n)}_{k^{(\min)}} \quad (7.19)$$

Omit the $k^{(\min)}$th column from $\mathbf{W}^{(n)}_{k^{(\min)}}$, set $K = K - 1$, and start iterations again at 1 in Fig. 7.1

b) If $K = 1$, return $\widehat{\mathbf{W}}^{*}$ and $\widehat{\epsilon}^{*}$ with the lowest out-of-sample MSE.

Figure 7.2: Sparse extension and automatic selection of K for Mixed LICORS; use after iterations in Fig. 7.1.

to one and only one cluster (the position of the 1 in $\mathbf{w}_i$). Such a vector is usually denoted as a (canonical) basis vector $\mathbf{e}_j$ of $\mathbb{R}^K$.[3]

While in simulations many weight vectors converge to one of the K basis vectors, we want to actively enforce sparser weights. Unique cluster assignments are usually obtained by assigning sample i the maximum posterior probability state, i.e.,

$$\widehat{\epsilon}(\ell_i^{-}) = \arg\max_j \widehat{w}_{ij}. \quad (7.17)$$

3 The j-th basis vector of $\mathbb{R}^K$, $\mathbf{e}_j = (0, \ldots, 0, 1, 0, \ldots, 0)$, is a K-dimensional vector with all zeros, but a one at the j-th position. The dimension K is usually implicitly clear from the context.

As the EM algorithm uses weighted estimation, it is not necessary to find this deterministic mapping in each step, $\epsilon^{(n)}(\ell_i^-)$, but we only need to hard-threshold the optimal solution $\widehat{\mathbf{W}}^*_{EM}$ to get $\widehat{\epsilon}^*_{EM}$.

Instead of using (7.17) on each row of $\widehat{\mathbf{W}}^*_{EM}$, we propose adding a penalty $R(\mathbf{W}) \geqslant 0$ to the log-likelihood

$$\ell(\mathbf{W}; \mathcal{D}) - \lambda R(\mathbf{W}), \tag{7.20}$$

where $\lambda \geqslant 0$ is the regularization parameter. The penalty function should satisfy $R(\mathbf{W}) \geqslant 0$ for all $\mathbf{W}$ with equality if and only if for all i, $\mathbf{w}_i = \mathbf{e}_j$ for some j.

The functions $\ell(\mathbf{W}; \mathcal{D})$ and $R(\mathbf{W})$ are continuous and differentiable with respect to $\mathbf{W}$, and $F_\lambda(\mathbf{W})$ is a continuous function of λ. Furthermore, for $\lambda = 0$ the EM algorithm presented above solves (7.23) (at least locally). For $\lambda \to \infty$, the penalty must tend to 0, which can only occur for deterministic cluster assignments, e.g., the arg max assignment.

However, the arg max assignment not necessarily solve the optimization problem for $\lambda \to \infty$, but is only a candidate solution. In fact, simulations show that the sparse EM solution we propose has better out-of-sample prediction than the arg max version of the unrestricted $\widehat{\mathbf{W}}^*_{EM}$.

For $\lambda \in (0, \infty)$ the optimal solution

$$\mathbf{W}^*_\lambda = \arg \max_{\mathbf{w}_i \in \Delta^{(K-1)}} \ell(\mathbf{W}; \mathcal{D}) - \lambda R(\mathbf{W}), \tag{7.21}$$

is a hybrid between the unique cluster and the unrestricted EM solution.

We thus now sparsify the EM algorithm using gradient descent and thus obtain a more sparse optimal solution $\widehat{\mathbf{W}}^*_{EM,sparse}$.

7.2.2.1 *Gradient Descent Algorithm to Induce Sparsity*

Stated as a minimization problem

$$F_\lambda(\mathbf{W}) = -\ell(\mathbf{W}; \mathcal{D}) + \lambda R(\mathbf{W}), \tag{7.22}$$

we have to solve

$$\mathbf{W}^*_\lambda = \arg \min_{\mathbf{w}_i \in \Delta^{(K-1)}} F_\lambda(\mathbf{W}). \tag{7.23}$$

A gradient descent algorithm obtains a solution to (7.23) by iteratively calculating

$$\mathbf{W}^{(k+1)} = \mathbf{W}^{(k)} - t_k \nabla F_\lambda \left(\mathbf{W}^{(k)}\right) \tag{7.24}$$

$$= \mathbf{W}^{(k)} - t_k \left(-\nabla \ell \left(\mathbf{W}^{(k)}\right) + \lambda \nabla R \left(\mathbf{W}^{(k)}\right)\right), \tag{7.25}$$

where t_k is the step size, and using a starting point $\mathbf{W}^{(0)}$. We use $\mathbf{W}^{(0)} = \widehat{\mathbf{W}}^{(*)}_{EM}$ after convergence (for a fixed K) for initialization.

Since the gradient of the log-likelihood is in general complicated, we use the EM algorithm to approximate part of the gradient step. We can view the updated weights in EM algorithm as an implicit gradient update: $\widehat{\mathbf{W}}^{(n+1)}_{EM} \approx \mathbf{W}^{(k)} + t_k \nabla \ell \left(\mathbf{W}^{(k)}\right)$. Thus we can make the temporary EM update $\widehat{\mathbf{W}}^{(n+1)}_{EM}$ more sparse, by subtracting a fraction of the gradient (see Bach, Jenatton, Mairal, and Obozinski (2012, Chapter 7))

$$\mathbf{W}^{(n+1)}_{EM,sparse} \leftarrow \mathbf{W}^{(n+1)}_{EM} - t_n \lambda \nabla R \left(\mathbf{W}^{(n+1)}\right). \tag{7.26}$$

Apart from choosing λ we also have to choose t_n. For good convergence properties the step size should usually go to 0 at an appropriate rate as the algorithm approaches the optimum. Since the EM update counteracts the sparsity inducing gradient step, a diminishing step size would lead the iterations $\mathbf{W}^{(n+1)}_{EM,sparse}$ ulti-

mately back to the non-sparse solution $\widehat{\mathbf{W}}_{EM}^{(n+1)}$. We thus set $t_n = 1$ for all n, choose a λ, and then stop updating via (7.26) once [4]

$$R(\mathbf{W}_{EM,sparse}^{(n+1)}) > R(\mathbf{W}_{EM,sparse}^{(n)}). \tag{7.27}$$

7.2.2.2 *Entropy Penalty*

We use an entropy penalty

$$R(\mathbf{W}) = \frac{1}{N} \sum_{i=1}^{N} \mathcal{H}(\mathbf{w}_i), \tag{7.28}$$

where $\mathcal{H}(\mathbf{p}) = -\sum_{j=1}^{K} p_j \log_2 p_j$ is the entropy (log base 2) of the discrete probability distribution $(p_1, \ldots, p_K) = \mathbf{p} \in \Delta^{(K-1)}$. The entropy penalty is a continuous and twice differentiable function in $\mathbf{W}$ with gradient

$$\nabla R = -\frac{1}{N}(1 + \log \mathbf{W}) \in \mathbb{R}^{N \cdot K \times 1}. \tag{7.29}$$

In the simulations we use the entropy penalization times $\log_2 K$,

$$F_\lambda(\mathbf{W}) = -\ell(\mathbf{W}; \mathcal{D}) + \lambda R(\mathbf{W}) \log_2 K, \tag{7.30}$$

which not only favors unique assignments, but also "small K" mixtures. Thus, Eq. (7.26) becomes

$$\widehat{\mathbf{W}}^{(n+1)} \leftarrow \widehat{\mathbf{W}}_{EM}^{(n+1)} - t_n \lambda \left(1 + \log_K \left(\widehat{\mathbf{W}}_{EM}^{(n+1)}\right)\right). \tag{7.31}$$

This updating mechanism follows our intuition that weights above the uniform threshold $1/K$ should get pushed towards 1, weights below towards 0.[5] The log-

4 As an alternative approach one could increase λ by a factor $a > 1$ once (7.27) holds. This leads to a sequence of $\lambda_n \to \infty$ and thus a unique cluster assignment.

5 Since update (7.26) does not guarantee $\widehat{\mathbf{W}}_i^{(n+1)} \in \Delta^{(K-1)}$, it is necessary to project each row back onto the probability simplex: we set negative entries in $\widehat{\mathbf{W}}^{(n+1)}$ to 0, and then re-normalize each row to sum to 1.

arithm to base K ensures that at least one entry per row will be increased, since among K classes at least one weight is larger than $1/K$ with probability one.

This entropy penalization is identical to the negative entropy criterion (NEC) to choose a good K in mixture models (Biernacki et al., 1999). Here we use the entropy to guide the algorithm to a more sparse solution.

7.2.2.3 *Why LASSO and Ridge Penalties Fail*

The entropy penalization is not a standard regularization, rather L_2 (ridge) and L_1 (LASSO) (Tibshirani, 1996) norms are typically used. Even though the latter have been used in the context of sparse mixture models - see Mallapragada, Jin, and Jain (2010) for L_2 and Bunea, Tsybakov, and Wegkamp (2009) for L_1 penalization - we want to point out that they do not (!) induce sparsity. The LASSO (Bunea et al., 2009)"penalizes" the log-likelihood by the same constant for every weight vectors, since $\|\mathbf{w}_i\|_1 = 1$ for all $\mathbf{w}_i \in \Delta^{(K-1)}$. The L_2 norm (Mallapragada et al., 2010) actually favors non-trivial mixture weights since $\|\mathbf{w}_i\|_2 \leqslant \|\mathbf{e}_j\|_2 = 1$ with equality if and only if $\mathbf{w}_i = \mathbf{e}_j$; thus basis vectors - which are as sparse as possible - get larger penalty. Mallapragada et al. (2010) use this property of the Gaussian prior to make transitions between iterations more smooth - and the EM algorithm more robust.

Remark 7.2.1 (Sparse Probability Measures)**.** *Very recent work by Pilanci, El Ghaoui, and Chandrasekaran (2012) seems to go in similar directions. However, the full article has not been made publicly available at the moment of completion of this work.*

Remark 7.2.2 (Penalized Fuzzy C Means (PFCM))**.** *During completion of this work it came to our attention that such an entropy-based penalization approach has been proposed in the signal and image processing literature, where it is known as* penalized fuzzy c means *(PFCM) (Yang, 1993).*

7.3 DISCUSSION OF THE EM ALGORITHM

Mixed LICORS is an iterative estimator for the predictive state space assignment and for the predictive distributions in each state. While it is not a true EM algorithm, since the M-step is only approximate using KDEs, the estimator arises naturally from a mixture model and is very similar to a standard, parametric EM. Similarly to Benaglia and Hunter (2009a), we use deterministic, weighted KDE for the updating step; we have not implemented the stochastic EM from Bordes et al. (2007) – but in principle, mixed LICORS can be modified to such a stochastic procedure.

We also propose a sparsity inducing penalty to obtain unique state space assignments. Contrary Bunea et al. (2009) and Mallapragada et al. (2010), our penalty does not avoid, but induces sparsity. We derive analytic updating rules to obtain more sparse mixture weights and simulations show (Section 7.6) that sparse mixed LICORS performs even slightly better than the non-sparse EM.

7.4 FORECASTING GIVEN NEW DATA

The estimate $\widehat{\mathbf{W}}^*$ can be used to forecast $\tilde{X}$ given a new $\tilde{\ell}^-$ as described in Section 6.4. The component distributions $\mathbb{P}\left(\tilde{X} = x \mid \tilde{S} = s_j\right)$ can be estimated independently from $\tilde{\ell}^-$ using (7.15). The mapping from PLC to predictive state is represented by the weight vector $\mathbf{w}(\tilde{\ell}^-)$. They are in general different for each $\tilde{\ell}^-$ and can be obtained by (6.33). We estimate element j of $\mathbf{w}(\tilde{\ell}^-)$ by evaluating the Gaussian distribution at the new PLC times the frequency of state s_j:

$$\begin{aligned}\widehat{\mathbf{w}}_j(\tilde{\ell}^-; \widehat{\mathbf{W}}^*) = \widehat{\mathbb{P}}\left(\tilde{S} = s_j \mid \tilde{\ell}^-; \widehat{\mathbf{W}}^*\right) &\propto \widehat{\mathbb{P}}\left(\tilde{\ell}^- \mid \tilde{S} = s_j; \widehat{\mathbf{W}}^*\right) \times \widehat{\mathbb{P}}\left(\tilde{S} = s_j; \widehat{\mathbf{W}}^*\right) \\ &= \mathcal{N}\left(\tilde{\ell}^-; \widehat{\boldsymbol{\mu}}^*_{(j)}, \widehat{\boldsymbol{\Sigma}}^*_{(j)}; \widehat{\mathbf{W}}^*\right) \times \frac{\widehat{N}^*_j}{N},\end{aligned} \tag{7.32}$$

where i) $\widehat{N}_j^{(*)} = \sum_{i=1}^{N} \widehat{W}_{ij}^{(*)}$ is the effective sample size of state s_j, and ii) $\widehat{\mu}_j^{(*)}$ and $\widehat{\Sigma}_j^{(*)}$ are weighted mean and covariance matrix estimates of the PLCs using the jth column of $\widehat{W}^{(*)}$.

Distribution forecasts for $\tilde{X} \mid \tilde{\ell}^-$ can then be obtained by evaluating (6.34) (after re-normalizing $\widehat{w}(\tilde{\ell}^-; \widehat{W}^*)$). Point forecasts can be obtained as needed, e.g., mean, median, or mode of (6.34). In the simulations we use the weighted average from each component as the prediction of $\tilde{X}$.

7.5 SIMULATING NEW DATA FROM EM ESTIMATES

Recall from Section 6.5 that a fully probabilistic model allows us to simulate a system based on an estimate $\widehat{\epsilon}^*$, rather than having to *know* the underlying dynamics. Since mixed LICORS estimates distributions and system dynamics at the same time, we can use $\widehat{W}^*$ and $\widehat{f}^*\left(x \mid \tilde{S} = s_j\right)$ to simulate another realization of the same spatio-temporal dynamics as outlined in Section 6.5, Fig. 6.2. For the multinomial distribution in step 2a, we use re-normalized estimated weights from (7.32). For step 2b, we can simulate from the weighted KDE in (7.15).

7.6 SIMULATIONS

Here we show that mixed LICORS largely improves in out-of-sample forecasts compared to hard LICORS, demonstrating the predictive advantages of using a probabilistic model of predictive states. We also compare sparse to non-sparse mixed LICORS and find that sparsification even further improves predictive power.

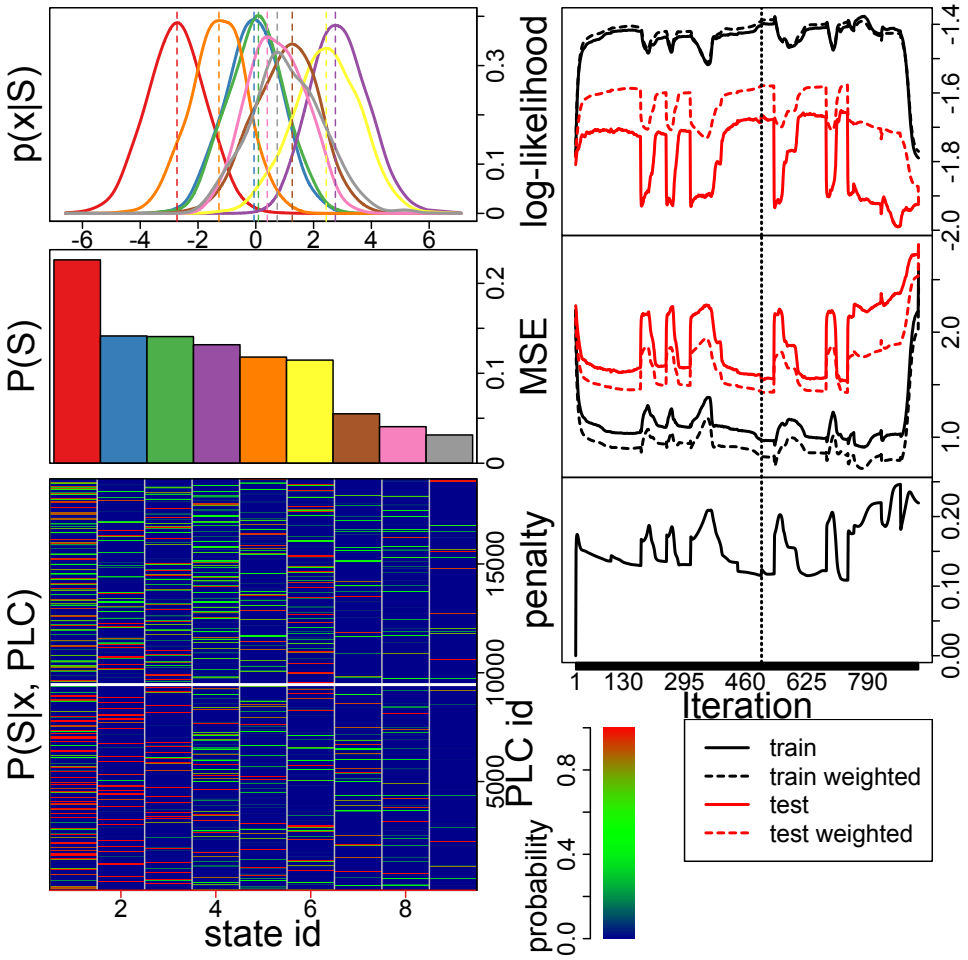
We use the same 100 simulations as in Section 5.4, Eqs. (5.16) and (5.18). Recall that the observable $X(\mathbf{r}, t)$ is a continuous-valued $(1+1)$D field with 7 discrete latent states $\{\epsilon_{-3}, \epsilon_{-2}, \ldots, \epsilon_2, \epsilon_3\}$. Control parameters are the past horizon $h_p = 2$ and speed of propagation $c = 1$. The FLC distributions are univariate and are con-

ditionally Gaussian given the state, $X(\mathbf{r}, t) \mid \epsilon_k = \mathcal{N}(k, 1)$, $k = -3, 2, \ldots, 2, 3$. One realization of these dynamics for $\mathbf{S} = \{1, \ldots, 100\}$ (vertical) and $t = 1, \ldots, T = 200$ (left to right) is shown in Chapter 5, Fig. 5.2b; the corresponding predictive state space is shown in Fig. 5.3a. While the (usually unobserved) state space has distinctive green temporal traces and also alternating red and blue patches, the observed field is too noisy to clearly see any of these patterns.

Figures 7.3 and 7.4 summarize two runs of mixed LICORS EM with $K = 15$ initial clusters and $h_p = 2$: (a) has no sparsity penalty ($\lambda = 0$) and (b) uses $\lambda = 1$ with the entropy penalty. The first 100 time steps were used as training data, and the second half as test data. The optimal $\widehat{\mathbf{W}}^*_{EM} = \widehat{\mathbf{W}}^*_{\lambda=0}$ which minimized the out-of-sample weighted MSE occurred after 502 iteration, with 9 estimated predictive states. All trace plots (log-likelihood, MSE, penalty) show large temporary drops (or increases respectively) when the EM reaches a local optimum and merges two states. After merging the forecasting performance and log-likelihood quickly return to - or even surpass - previous optima. The weight matrix $\widehat{\mathbf{W}}^*_{EM}$ in the lower-left shows that many weight vectors converged to one of the basis vectors (one red, all other blue per row). The horizontal white line separates training (lower half) from test data (upper half). The entropy penalty of $\widehat{\mathbf{W}}^{(n+1)}$ quickly iterates to an almost single state assignment, but still $R(\widehat{\mathbf{W}}^*_{EM}) \approx 0.11$.

The penalty plot of the sparse EM in Fig. 7.4 shows that only very few iterations of (7.26) are needed to move the non-sparse EM solution ($R(\widehat{\mathbf{W}}^{(*)}_{EM}) \approx 0.18$) to an extremely sparse estimate ($R(\widehat{\mathbf{W}}^{(*)}_{EM,sparse}) \approx 0.03$.). This can also be seen in the weight matrix, where almost every row has only one active (red) entry. As this enforced sparsification only affects the weights for the training data, the test data still has several true mixtures over states.

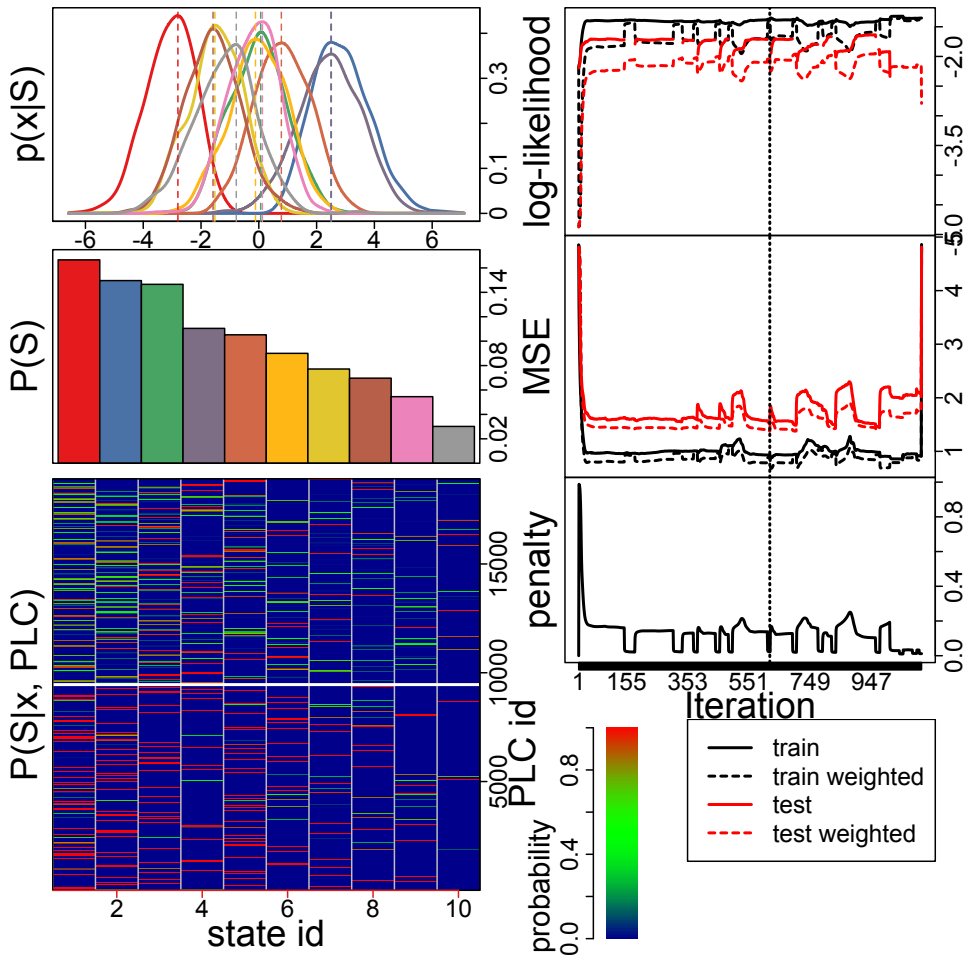
The predictions from $\widehat{\mathbf{W}}^*$ in Fig. 7.5b show that mixed LICORS is practically unbiased – compare to the visually indistinguishable true state space in Fig. 7.5a.



(a) No sparsity $\lambda = 0$.

Figure 7.3: Mixed LICORS with $h_p = 2$ and $K = 15$ starting states. (left) nonparametric estimates of (top) conditional predictive distributions $\mathbb{P}(X = x \mid S = s_j)$, (center) marginal state probabilities $\mathbb{P}(S = s_j)$, and (bottom) conditional state probabilities $\mathbb{P}(S_i = s_j \mid X_i, \ell_i^-)$; (right) trace plots for test and training data & weighted and non-weighted (top) log-likelihood, (center) MSE, and (bottom) entropy penalty $R(\widehat{\mathbf{W}}^{(n)})$.

The residuals in Fig. 7.5c show no obvious patterns except for a larger variance in the right half (training vs. test data).

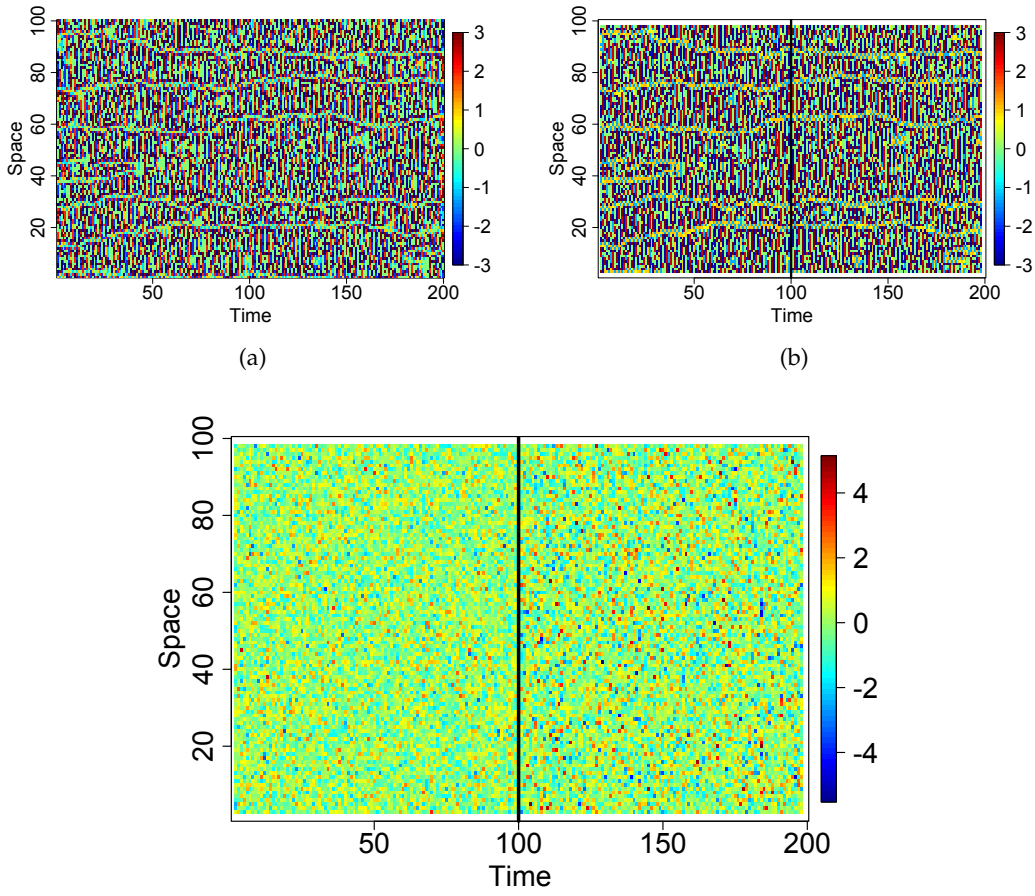


(a) Sparsity $\lambda = 1$.

Figure 7.4: Sparse mixed LICORS with $\lambda = 1$ after initial convergence with non-sparse EM. See caption of Fig. 7.3 for details.

7.6.1 *Simulating System From Different Initial Conditions*

Recall that all simulations use $X(\cdot, 1) = X(\cdot, 2) = \mathbf{0} \in \mathbb{R}^{|S|}$ as initial conditions (see (5.17)). If we want to know the effect of different starting conditions, then we can simply use Eqs. (5.16) & (5.18) to simulate that system, since they fully specify the evolution of the stochastic process. In experimental studies, however, researchers



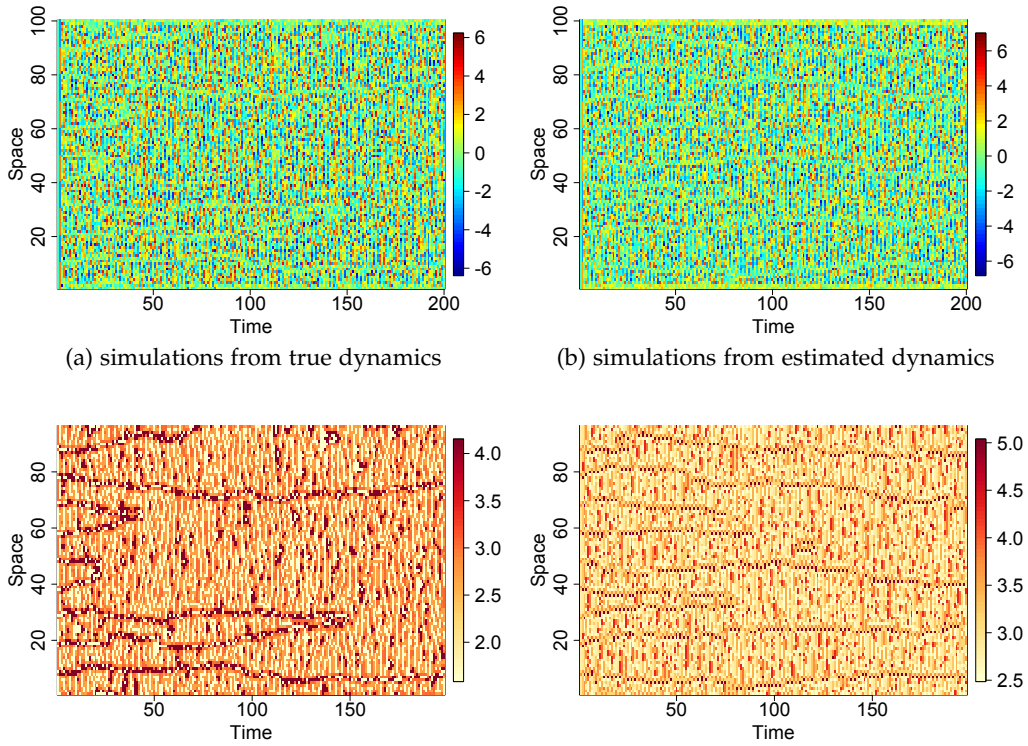
(a) (b)

(c)

Figure 7.5: Mixed LICORS model check: (a) true predictive state space $S(\mathbf{r}, t)$; (b) estimated predictive state space using sample average of the estimated predictive state distribution at each $(\mathbf{r}, t)$; (c) residuals.

usually do not have these mechanisms available, but finding them is one of the main purposes of the experiment in the first place.

Since mixed LICORS estimates joint and conditional predictive distributions, and not only the conditional mean, it is possible to simulate a new realization from an estimated model. Figure 6.2 outlines this simulation procedure. We will now demonstrate that mixed LICORS can be used instead to simulate from different initial conditions *without* knowing Eqs. (5.16) & (5.18).



(a) simulations from true dynamics

(b) simulations from estimated dynamics

(c) LSC of known predictive state space

(d) LSC of known predictive state space

Figure 7.6: Simulating another realization of (5.16) – (5.18) with different initial conditions.

Figure 7.6a shows a simulation using the true mechanisms in (5.16) & (5.18) with starting conditions $X(\cdot, 1) = -\mathbf{1}$ and $X(\cdot, 2) = \pm\mathbf{3} \in \mathbb{R}^{|\mathbf{S}|}$ in alternating patches of ten times 3, ten times -3, ten times 3, etc. (total of 10 patches since $|\mathbf{S}| = 100$). The first couple of columns (on the left) are influenced by different starting conditions, but the initial effect dies out soon (since $h_p = 2$) and similar structures (left to right traces) as in simulations with (5.17) emerge (Fig. 5.2).

Figure 7.6b shows simulations solely using the sparse mixed LICORS estimates in Fig. 7.4. While the patterns are quantitatively different (due to random sampling), the qualitative structures are strikingly similar (compare also Figures 7.6c and 7.6d). The overall higher complexity is due to the larger estimated state space in the sparse EM (9 estimated states versus 7 true states); hence probabilities per state are overall smaller, and hence complexity higher. Mixed LICORS can not only

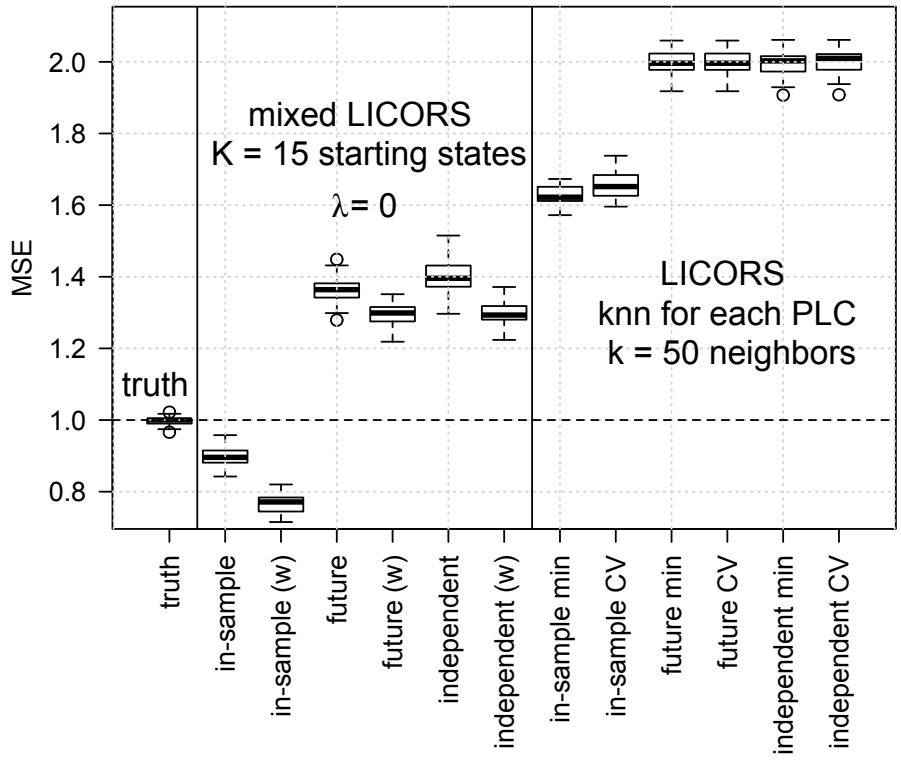


Figure 7.7: MSE comparison between (non-sparse) mixed and hard LICORS.

accurately estimate $S(\mathbf{r}, t)$, but also learn the optimal prediction rule (5.16) solely from the observed data $X(\mathbf{r}, t)$.

This shows that in principle mixed LICORS can learn state-conditional distributions and predictive state mappings successfully from observed data, and then regenerate fields with similar spatio-temporal dynamics. This can become very useful in experimental setups, where running another experiment might be very costly and time-intensive, while running mixed LICORS and simulating on a computer is very fast and cheap.

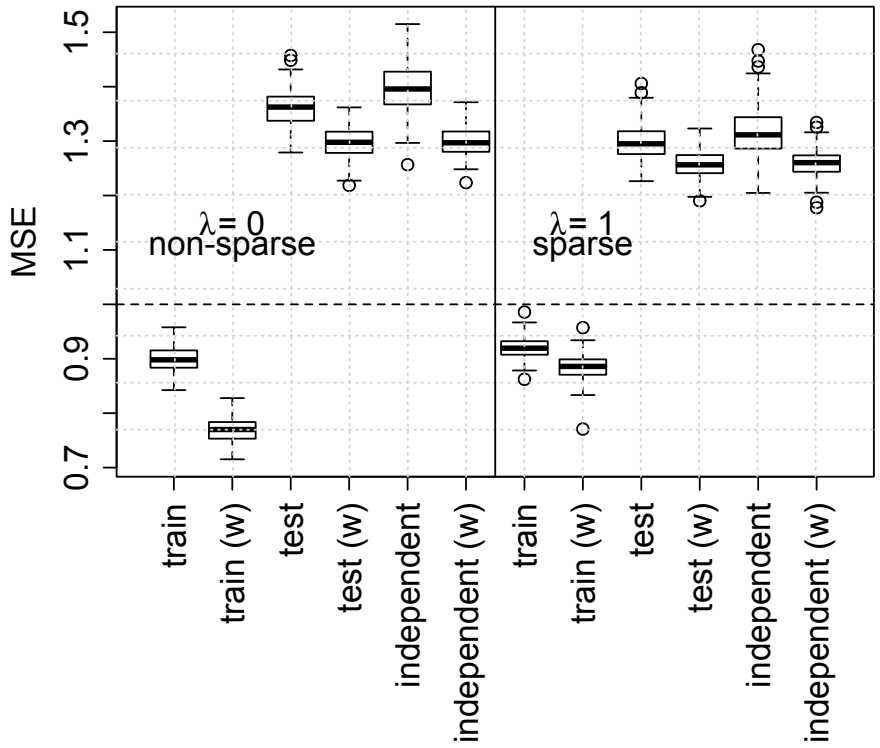


Figure 7.8: MSE comparison between sparse and non-sparse mixed LICORS.

7.6.2 *Mixed versus Hard LICORS; Sparse versus Non-Sparse*

Here we show that mixed LICORS indeed outperforms hard LICORS over multiple runs. We use 100 independent realizations of (5.16) – (5.18) and for each one we train the model on the first, and test it on the second half (future). The optimal model is the one with lowest out-of-sample future MSE.

We use the EM algorithm as outlined in Fig. 7.1 with $K_{max} = 15$ states, 1000 maximum number of iterations, two sparsity levels $\lambda \in \{0, 1\}$, and we keep the estimate with the lowest out-of-sample MSE out of 10 independent runs. The first run is initialized with a K-means clustering on the PLC space; the remaining state initializations are assigned uniformly at random from $\{s_1, \ldots, s_K\}$.

To test whether mixed LICORS accurately estimates the mapping $\epsilon : \ell^- \mapsto \mathcal{S}$, we also predict FLCs of an independently generated realization of the same underlying process. If the out-of-sample MSE for the independent field is the same as the out-of-sample MSE for the future realization of the field it was trained on, then mixed LICORS is not biased towards random fluctuations in any given realization, but estimates characteristics of the underlying system. For comparison, we use weighted forecasting as well as unique-state (arg max rule) predictions.

Figures 7.7 and 7.8 summarize the results for $\lambda = 0$ and $\lambda = 1$, respectively. Both the standard EM as well as the sparse EM solutions greatly improve upon the optimal hard LICORS estimates, with up to 33% reductions for out-of-sample MSE. Similarly to hard LICORS, the MSE for future and independent realizations are essentially the same, which shows that mixed LICORS can also learn characteristic of the system, and is not biased towards random fluctuations in the training data. Since the optimal weights are often already canonical basis vectors, weighted prediction does only slightly better than hard-thresholded state predictions.

It is noteworthy that the sparse EM solutions do not overfit as much as the non-sparse solutions, and also perform better on the test data even compared to the arg max solutions from non-sparse mixed LICORS (Fig. 7.8). This suggests that penalty induced EM algorithms might, in general, provide better unique cluster assignments than the commonly used arg max rule.

7.7 MIXED LICORS VERSUS HARD LICORS

Based on the probabilistic framework from Chapter 6, I introduce *mixed LICORS*, a nonparametric EM-like algorithm for estimating the predictive state space of a spatio-temporal process. Mixed LICORS is a probabilistic generalization of hard LICORS and can thus be easily adapted to other statistical settings such as classification or regression. Simulations show that it greatly outperforms its hard-clustering predecessor. We also introduce sparsity inducing penalties to obtain unique cluster

assignments in EM algorithms. Simulations show that this penalty imposed sparsity performs better than weighted mixtures, and also better than the commonly used arg max rule.

However, similarly to other state-of-the-art nonparametric EM algorithms (Bordes et al., 2007; Hall et al., 2005; Mallapragada et al., 2010), theoretical properties of our EM are not yet well understood. In particular, the nonparametric estimation of mixture models can pose identifiability problems (see Benaglia and Hunter, 2009a, Section 2, and references therin). We demonstrated empirically that it does not suffer from identifiability problems, and outperforms hard-clustering (identifiable) methods.

We also demonstrate that mixed LICORS can learn spatio-temporal dynamics from data, which can then be used for simulating new experiments. Thus mixed LICORS can in principle make a lot of expensive, time- and labor-intensive experimental studies much more manageable and easier to plan. Such an optimal statistical model can potentially learn spatio-temporal dynamics already after a few experiments. Once learned, researchers can use the estimates to simulate their system from different starting conditions within a couple of minutes rather than waiting for their empirical experiments to finish in hours, days, or months.

8

APPLICATIONS OF LOCAL STATISTICAL COMPLEXITY

Es ist nicht genug zu wissen - man muss auch anwenden.
Es ist nicht genug zu wollen - man muss auch tun.
(Knowing is not enough, we must apply.
Willing is not enough, we must do.)

Johann Wolfgang von Goethe

Since local statistical complexity (LSC) can be directly obtained from a LICORS estimate, we can readily apply LSC to real-world data for automated pattern discovery. This provides researchers with a purely data-driven algorithm that automatically shows informative areas in space-time without any specification of what to look for.

To illustrate the interpretation as well as show the accuracy of LSC in Section 8.1, we revisit the candle example from Chapter 1. In Section 8.2, we apply LSC to two fMRI datasets and show that it automatically detects irregular spatio-temporal activity in the brain; activity that traditional techniques fail to find.

8.1 CANDLE IN THE WIND

Reality is not always probable, or likely.

Jorge Luis Borges

For the sake of illustration, we first consider the video of a burning flame that gets extinguished (recall that a video is a $(2+1)$D spatio-temporal system). We use this toy-example to demonstrate our approach since it is intuitively clear what the spatially and temporally interesting events are.

With a resolution of $\mathbf{S} = 100 \times 200$ pixels and a duration of $T = 225$ frames, this video has a total of $N = 225 \times 100 \times 200 = 4.5 \times 10^6$ points in space-time, each one associated with a high-dimensional PLC and FLC. The seven representative frames (top row of Fig. 8.1) feature a wide range of dynamics from a stable burning candle to chaotic evolution of smoke. Temporally interesting are the extinction of the flame in the beginning and the rising smoke that follows. Spatially, we can differentiate between generally uninteresting background versus interesting foreground (candle, flame, and smoke); moreover, smoke becomes most interesting whenever it forms complicated swirls.

Specifying these interesting features by hand and implementing an algorithm to find them is not only difficult, but also time-consuming. The LSC methodology presented in Section 4.1 can learn and detect such features automatically from the data. Here we set $h_p = 2$, $h_f = 2$, and $c = 2$. Thus the PLC and FLC are 106 and

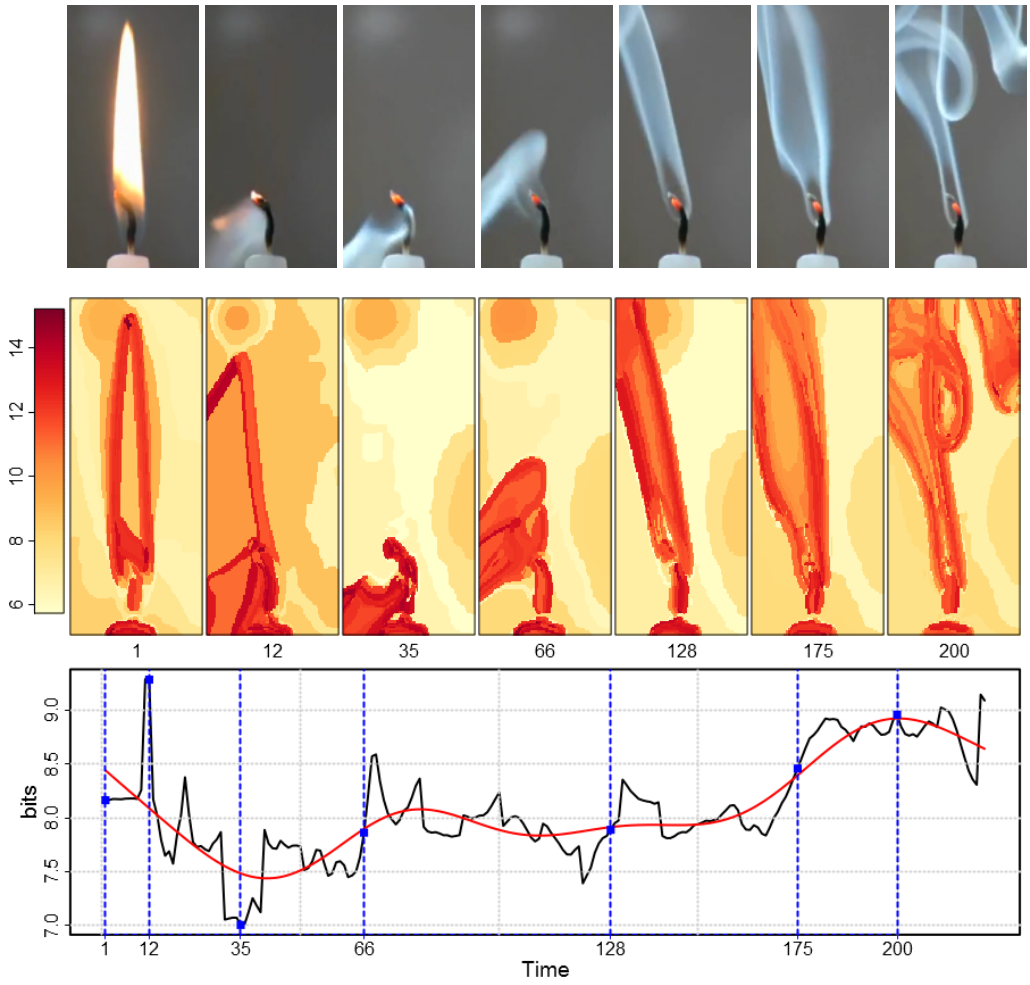


Figure 8.1: Candle LSC - average spatial complexity evolving over time: (top) snapshots of original data $X(\mathbf{r}, t)$ at selected t; (middle) $\widehat{\mathcal{C}}(\mathbf{r}, t)$ (dark values $\leftrightarrow$ high complexity); (bottom) spatial average at each t plus smoothed regression line (red). Peaks and valleys in average complexity identify interesting/uninteresting dynamics: $t \approx 1-10$ flame is burning, $t \approx 12$ candle gets blown out, $t \approx 66$ smoke starts to rise, $t \approx 175$ smoke starts to move to the right, $t \approx 200$ smoke forms complicated whirls.

107-dimensional vectors, respectively.[1] The significance level was set to $\alpha = 0.001$, using a starting partition by K-means++ of $K = 1\,000$.

1 We put the present $(\mathbf{r}, t)$ in the FLC.

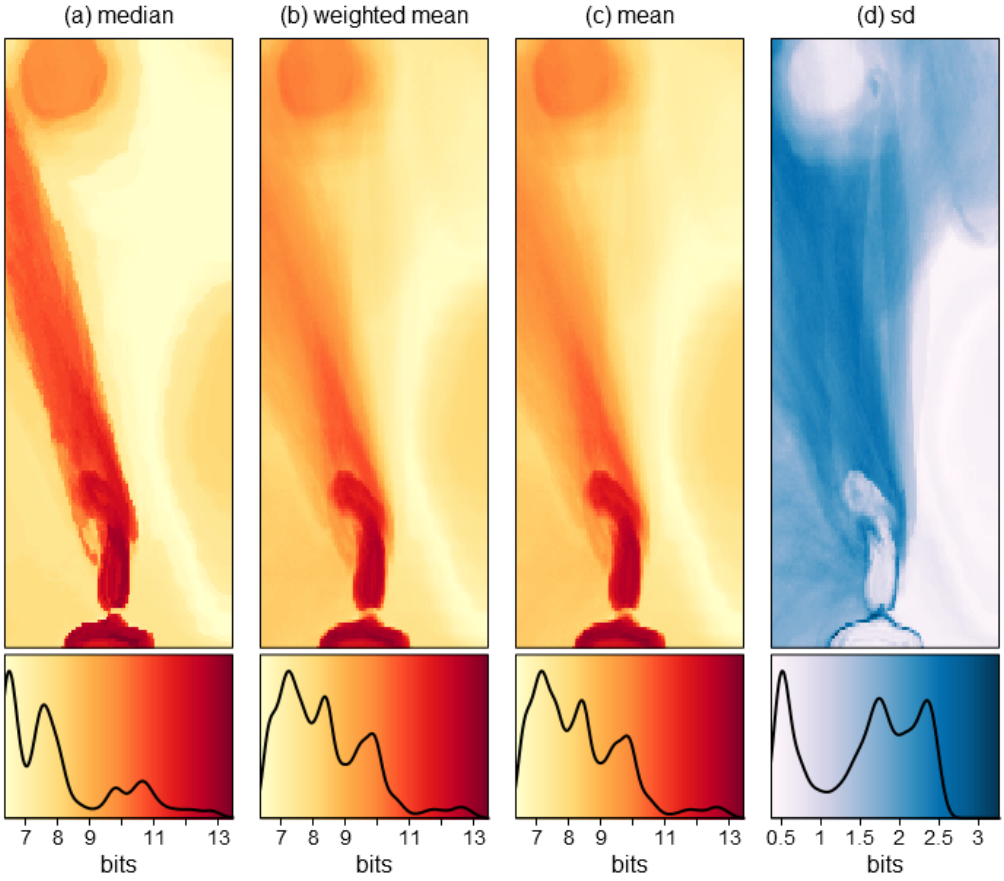


Figure 8.2: Candle LSC: average temporal complexity distributed over space. Pixel-wise summary statistics of $\widehat{\mathcal{C}}(\mathbf{r}, t)$: (a) median, (b) weighted mean, (c) mean, and (c) standard deviation. Mean and median distinguish between candle/smoke and background; also the circular spot on the top is clearly visible. Standard deviation shows that temporal changes occur only on the left half, but not on the right or the circular spot on top. Thus we can infer that the candle got extinguished from the right hand side, and smoke was rising on the left side.

The middle row of Fig. 8.1 shows that $\widehat{\mathcal{C}}(\mathbf{r}, t)$ matches our intuitive notion of interesting events in space and time very closely (darker colors correspond to higher complexity). The candle, the boundary between flame and background, and especially the glowing candle wick have high complexity; once smoke evolves LSC identifies it as the most interesting feature in the video.

The spatial average complexity $\overline{\overline{\mathcal{C}}}(t)$ in the bottom row of Fig. 8.1 correctly identifies interesting moments in the video: candle gets extinguished ($t \approx 12$), smoke starts to rise ($t \approx 66$), and smoke starts to swirl at the top ($t \approx 175$ and 200).

The temporal average complexity in Fig. 8.2 also provide valuable information. All three location estimates (median, weighted mean, and median) identify the candle and wick as the most interesting parts, and other interesting events in the upper-left part of video (especially note the spot in the top-left corner); the right half is overall uninteresting. The pixel-wise standard deviation shows that also most variation happens in the left part, whereas the candle, the upper-left spot, and the entire right half do not vary a lot over time. Thus LSC also tells us that the flame was blown out from the right, and smoke only evolved in the left half of the image.

This toy-example shows that LSC does indeed reflect and - most importantly - quantify our intuition of interesting spatio-temporal patterns. We will now apply LSC to fMRI datasets, where it is not immediately clear what the interesting structures look like. LSC can therefore inform neuro-scientists about the location and timing of important spatio-temporal brain activity.

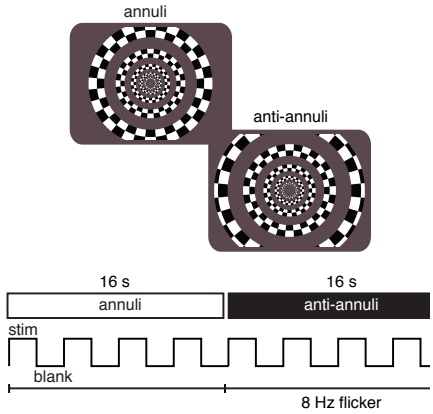
8.2 FUNCTIONAL MAGNETIC RESONANCE IMAGING

An idea which can be used once is a trick.
If it can be used more than once it becomes a method.

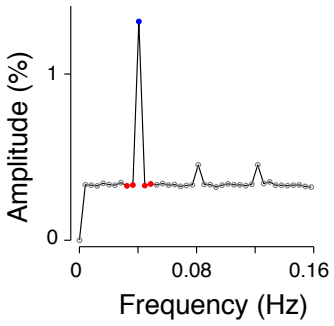
George Polya and Gabor Szego

We apply LSC to two datasets from high-resolution fMRI experiments (voxel size $0.75 \times 0.75 \times 0.75\ \text{mm}^3$) to illustrate its advantages over traditional pattern discovery methods.

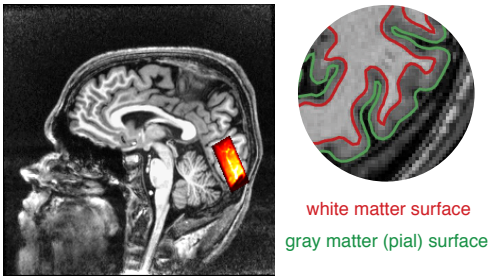
HARMONIC STIMULUS: In this experiment the stimulus consisted of concentric annuli alternating with anti-annuli (Fig. 8.3a), which evoked alternating, evenly-



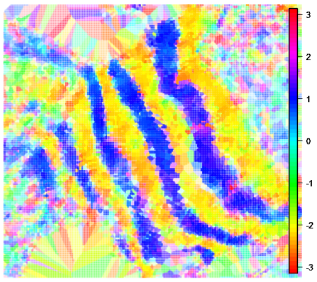
(a) Stimulus and experimental design.



(b) Fourier amplitude spectrum averaged over all voxels.



(c) Zoomed field of view of primary visual cortex.



(d) Activity bands detected by Fourier methods.

Figure 8.3: Experimental protocol (top), measurements & ring-pattern identified by using the modulus of the Fourier transform per voxel averaged over time (bottom).

spaced bands of activity in visual cortex (Fig. 8.3d). The ring size was particularly selected to produce small ($\approx$ 3 mm) activity bands. For further details on the experimental setup see Schlupeck, Merriam, Sanchez-Panchuelo, Francis, Bowtell, Velasco, Inati, and Heeger (2010).

For LSC estimation, we use 80 frames with a resolution of 282×263 pixels; thus $N \approx 6 \times 10^6$.

NON-HARMONIC STIMULUS: See Freeman, Brouwer, Heeger, and Merriam (2011) for details on the experimental protocol.

Here we analyze 164 frames with 210×210 pixels each; thus $N \approx 1.2 \times 10^7$.

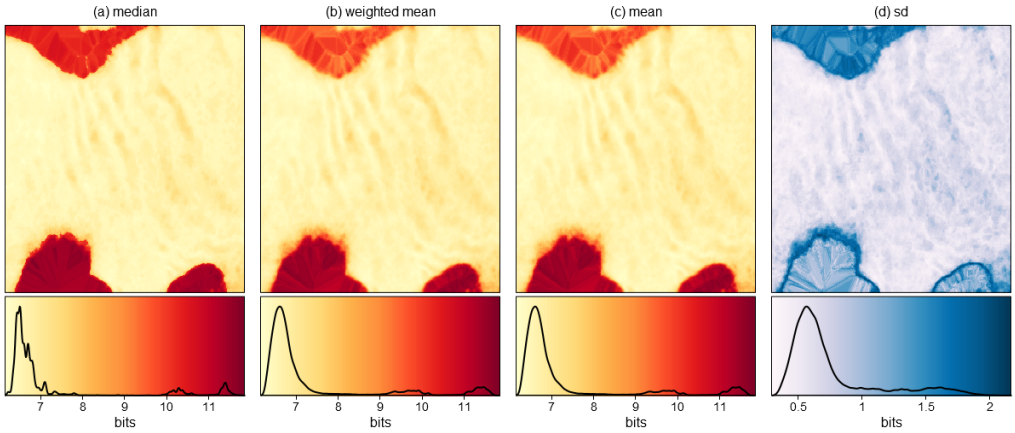
In both analysis, we applied hard LICORS with $h_p = 4$, $h_f = 2$, and $c = 1$; control parameters were set to $K = 226$ ($K = 367$) initial clusters, and significance level $\alpha = 10^{-4}$ ($\alpha = 10^{-3}$). Predictive states, state probabilities, and LSC were then estimated automatically from the data.

8.2.1 *Harmonic Stimulus*

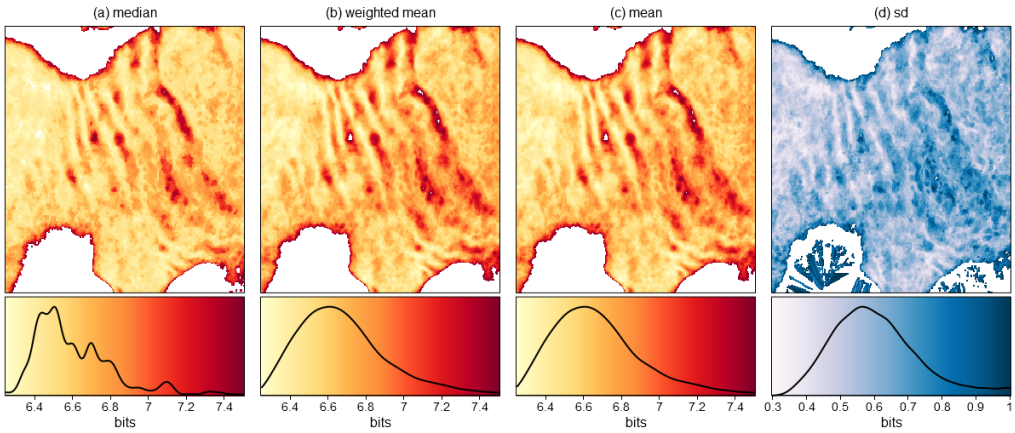
As the stimulus is periodic with a known frequency we use a Fourier-based method as the comparative baseline technique. The activity bands in Fig. 8.3d were obtained by computing the amplitude of the Fourier transform of each voxel time series at the *known* frequency of the stimulus. This is an example of a "template matching" technique, since the experimentalist controls the stimulus and can particularly search for voxels that respond at the same frequency. This particular Fourier-based technique has to use the time dimension to compute relevant statistics. It can therefore only give an average response of each voxel over time. On the contrary, LSC $\mathcal{C}(\mathbf{r}, t)$ provides estimates across the full spatio-temporal resolution.

This is an ideal test situation since the nonparametric LSC is a complete opposite to template matching: it does not use any prior knowledge about the stimulus.

Figures 8.4 and 8.5 summarize the LSC estimates. Even though LSC does not make any assumption on the shape of the stimulus, it detects the same activity bands (Fig. 8.4), just with more noise. However, we believe that this slightly lower signal to noise ratio is by far compensated by the higher spatio-temporal resolution yielding a much better understanding of the dynamics of brain activity. For example, contrary to the Fourier technique, $\mathcal{C}(\mathbf{r}, t)$ can be analyzed over time to see when those ring patterns turn on and off (upper row of Fig. 8.5).



(a) Color scale on full range



(b) Color scale on truncated levels

Figure 8.4: LSC results on fMRI of harmonic stimulus: average temporal LSC distributed over space. Pixel-wise summary statistics of $\widehat{\mathcal{C}}(\mathbf{r}, t)$: (a) median, (b) weighted mean, (c) mean, and (c) standard deviation.

8.2.2 *Non-harmonic Stimulus*

As the stimulus is not harmonic, Fourier methods are only of limited use, and one would have to design a new method for this particular stimulus structure. This is not only very time-consuming but also requires knowledge about the stimulus and its effect on the information processing in the brain; often this is exactly what researchers want to infer from their experiments. Here LSC demonstrates its full

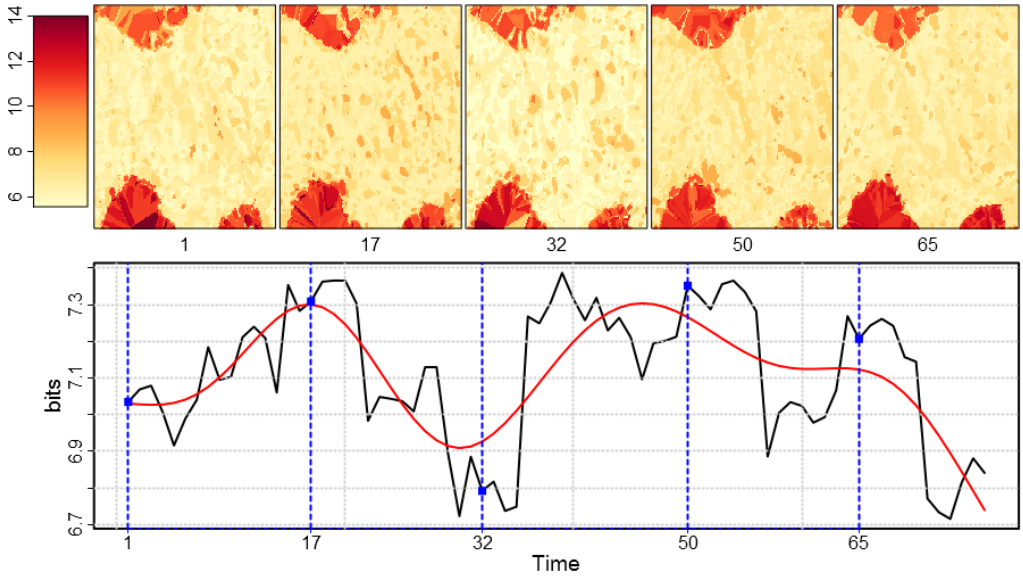


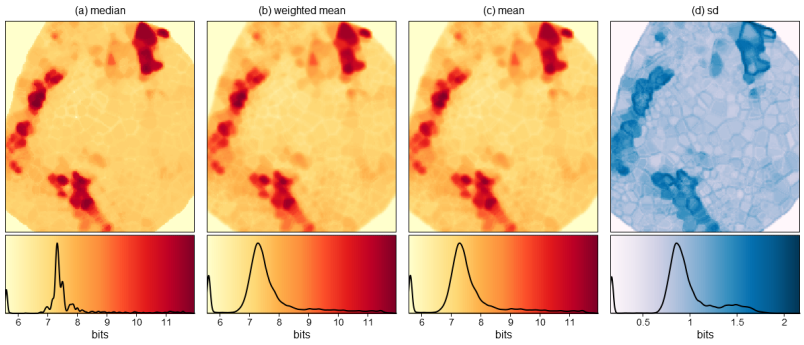
Figure 8.5: fMRI LSC: average spatial complexity evolving over time.

strength as it detects patterns in a completely automatic, nonparametric fashion and greatly outperforms matched filter techniques.

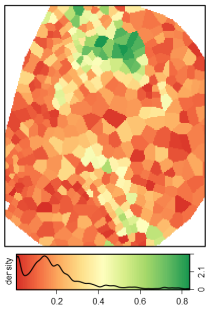
Using the same Fourier method as for the harmonic stimulus highlights large activity at the top and some activity at the bottom (Fig. 8.6b). LSC discovers an additional activity region (stripe on the left in Fig. 8.6a and 8.7). The high spatio-temporal resolution is another distinctive feature of LSC, and researchers can use it to very specifically target brain regions of interest. Figure 8.7 shows, for example, that brain regions respond at different times to the stimulus. Finally, we find that LSC estimates for this dataset are much less noisy than Fourier-based estimates.

8.2.3 *Multiple Experiments*

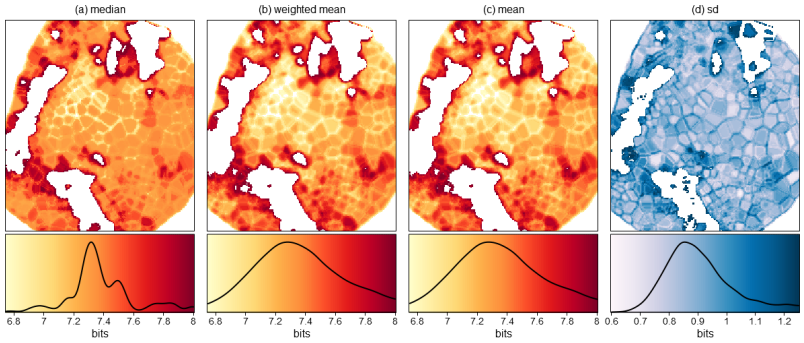
For the non-harmonic stimulus, we have fMRI data from 11 repeated experiments. This puts us in the rare position to have 11 realization of the same system. By applying LSC on each one separately, we can obtain a notion of uncertainty in the estimates.



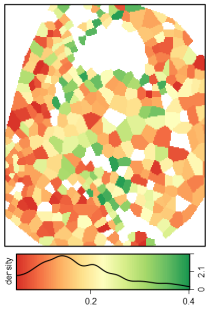
(a) Color scale on full range



(b) Voxel-wise Fourier methods.



(c) Color scale on truncated levels



(d) Voxel-wise Fourier methods.

Figure 8.6: LSC results on fMRI of non-harmonic stimulus: average temporal LSC distributed over space. Pixel-wise summary statistics of $\widehat{\mathcal{C}}(\mathbf{r}, t)$: (a) median, (b) weighted mean, (c) mean, and (c) standard deviation.

We first generate an average fMRI scan of all 11 runs – thus reducing noise in the data–, and then apply LSC to this average dataset ($h_p = 4$, $h_f = 2$, $c = 1$, and $K = 367$, $\alpha = 10^{-3}$). Results in Fig. 8.4a identify three main activity regions in the center part plus two spots on the edge (top right and bottom left). The temporal standard-deviation also identifies those areas with a larger standard deviation. It is not clear if we can use this pixel-wise standard deviation as a good uncertainty measure of the spatio-temporal estimate $\widehat{\mathcal{C}}(\mathbf{r}, t)$. However, we can apply LSC to all 11 runs separately, with the same settings as for the average dataset, and use

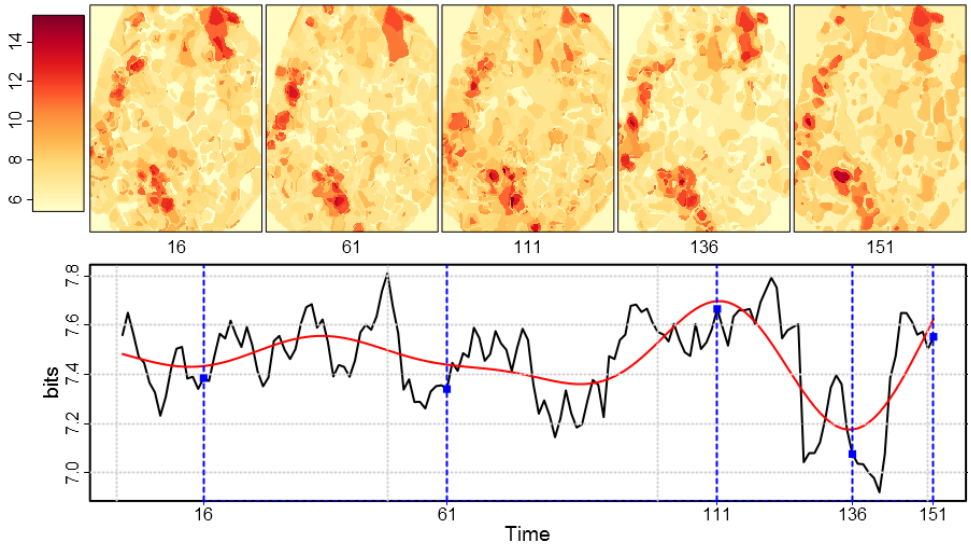


Figure 8.7: LSC results on non-harmonic stimulus: average spatial LSC evolving over time.

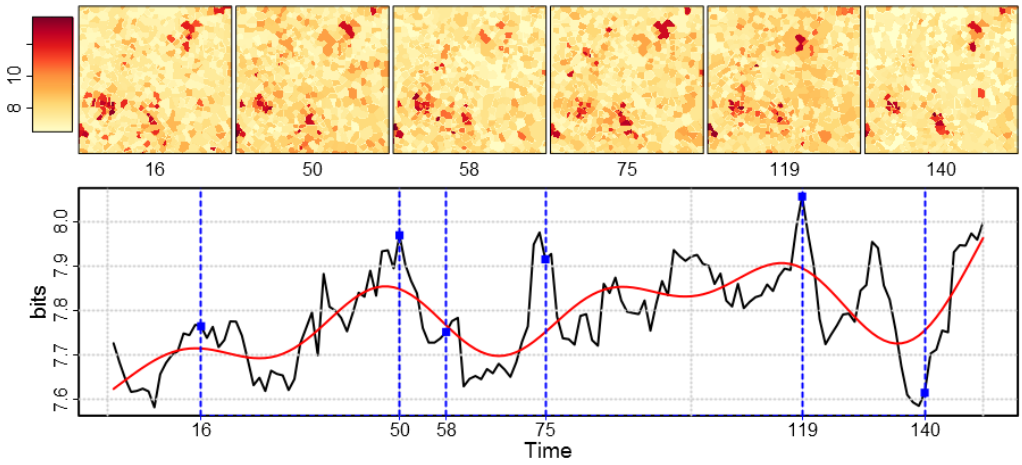
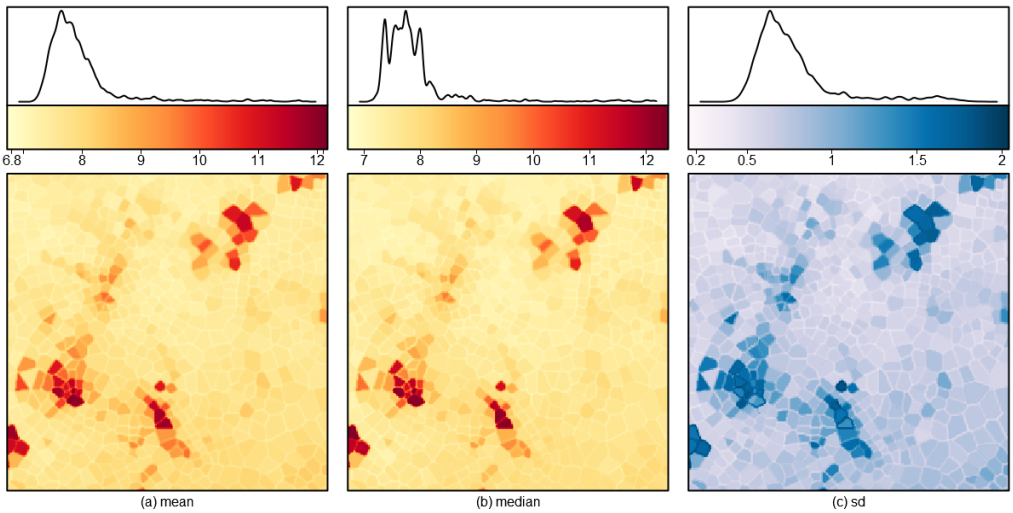


Figure 8.8: LSC results on subset of average fMRI dataset: snapshots of spatio-temporal complexity and average spatial LSC evolving over time.

cross-sectional averages and standard deviations to quantify the uncertainty in the estimates.

Figure 8.10a shows the temporal average LSC of all 11 runs plus cross-experiment pixel-wise average and standard deviation. As a comparison, it also shows the temporal average and standard deviation of LSC plus the Fourier estimate from the average dataset. Except for some deviations in experiment 1 (top right), and



(a) Color scale on full range

Figure 8.9: LSC results on average fMRI dataset: average temporal complexity distributed over space. Pixel-wise summary statistics of $\widehat{\mathcal{C}}(\mathbf{r}, t)$: (a) median, (b) weighted mean, (c) mean, and (c) standard deviation.

8 & 9 (stronger activity in top right), LSC consistently estimates the same structures throughout all experiments. Furthermore, they are the same structures as in the average dataset, suggesting that LSC estimates from a single realization are proper.

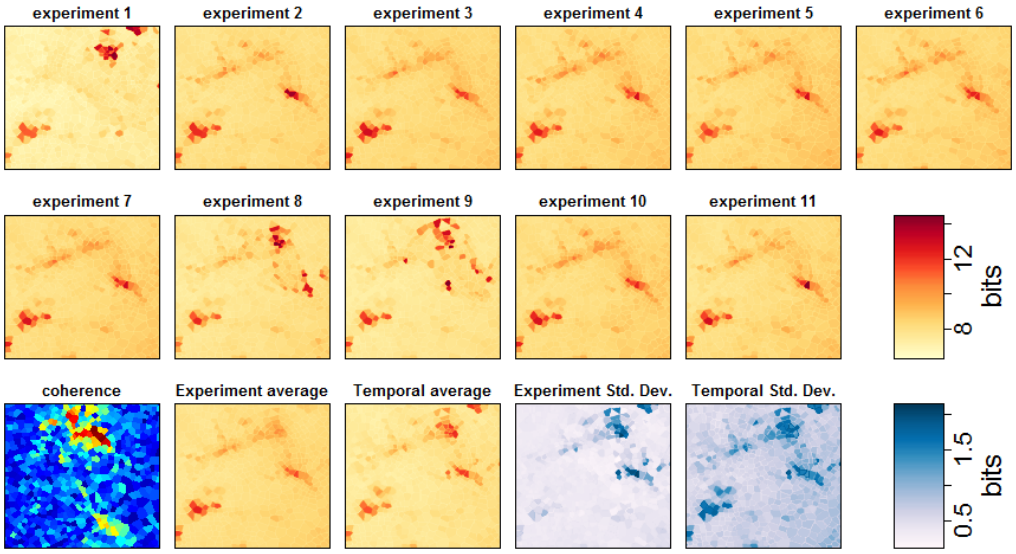
The Fourier estimates, on the other hand, have several shortcomings (Fig. 8.10b):

i) Estimates are much more noisy than LSC estimates and sometimes even lack any structure (e.g., experiment 2 and 6)

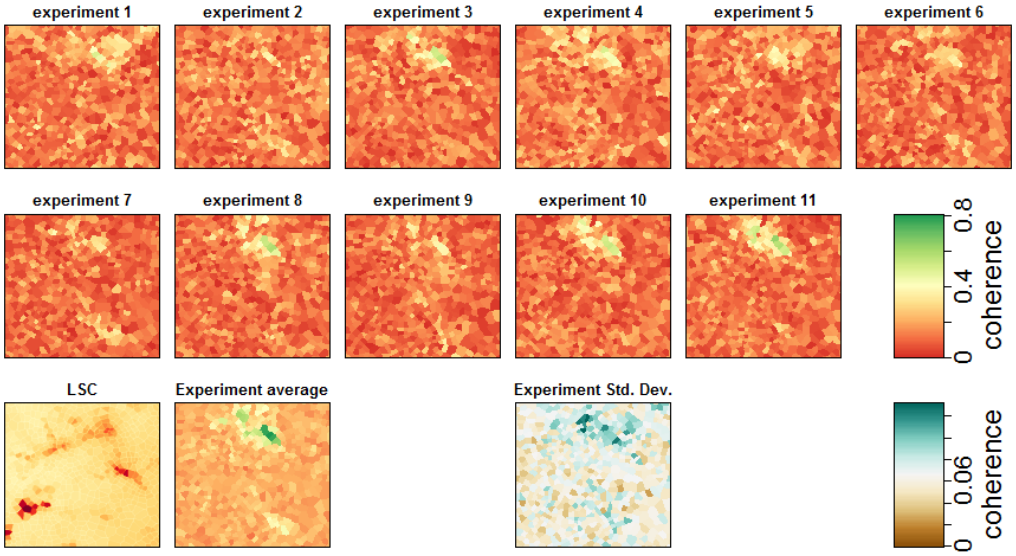
ii) The activity in the bottom half only appears in about half of the experiments, and is also barely detectable in the average of Fourier methods.

iii) The cross-experiment averages are smaller and have less variation than those of the average dataset. This indicates that the Fourier method can overestimate brain activity when using a single realization.

This comparison shows that LSC from one realization is close to the estimates from multiple realization of the same process. This suggests that LSC estimator based on a single realization have good frequentist properties.



(a) LSC with comparison to Fourier (bottom-left)



(b) Fourier with comparison to LSC (bottom-left)

Figure 8.10: Cross-experiment pixel-wise averaged estimates versus estimate on averaged experimental data based on 11 observations of the same experiment. (top) LSC estimates; (bottom) Fourier method.

8.3 SUMMARY

Local statistical complexity (LSC) is a method for fully automated pattern discovery in spatio-temporal data - such as functional magnetic resonance imaging (fMRI) - by means of optimal local prediction. The underlying idea is that statistically optimal predictors not only predict well, but - for this very reason - also reveal informative structure inherent in the system. A major advantage of pattern *discovery* by LSC over pattern *recognition* techniques is that it is not necessary to know what is interesting beforehand; LSC detects informative areas in space-time automatically.

Simulations on one dimensional space-time fields as well as applications to fMRI data show that LSC is a very valuable exploratory tool for analyzing spatio-temporal data. In particular, we show that it can detect highly irregular spatio-temporal brain activity in fMRI data from very different experimental setups.

LSC can therefore become an unparalleled tool for applied researchers to detect important structures in, yet unknown, dynamical systems.

Part IV

DISCUSSION

9

FUTURE WORK CONE

I never think of the future - it comes soon enough.

Albert Einstein

In Part III, I presented the main results of this thesis. Apart from methodology, theoretical results, and applied work, one main contribution is the embedding of light cones, predictive states, and optimal forecasts in a probabilistic setting, which naturally leads to a statistical model for general statistical inference. This in turn opens up a wide range of topics for future work and refinements of LICORS and LSC using machinery and results from statistics and machine learning. Here I will briefly mention some selected topics.

Such a list can never be complete; the topics I discuss are the ones I find most prevalent and interesting.

9.1 SCALING LICORS UP TO "BIG DATA" AND ONLINE ALGORITHMS

While spatio-temporal systems are ubiquitous in scientific research, detailed spatio-temporal measurements started to become available only recently. One distinctive feature of such data is its large dimensionality. For "small N" – meaning $N \approx 10^5 - 10^6$ – the implementations I propose for hard and mixed LICORS work on the order of minutes on a standard office laptop (2.5 GHz, dual core, 4 GB RAM). Most spatio-temporal datasets, however, easily exceed this small sample size. For example, even the low-resolution (100×200 pixels) and very short (200 frames) candle clip already has about 10^6 samples. In scientific experimental setups or in plain HD video data this can easily grow to 10^{10} or more data points. It is therefore important to have fast algorithms for LICORS and LSC to also work on "big data".

Computational speed-ups can be achieved by using a parallel implementation whenever possible. For example, while the EM algorithm is by construction sequential, one can trivially parallelize KDE estimation of all states in the E-step or the update of the posterior weights $\mathbf{w}_i$. Similarly, forecasting and simulating can be accelerated by evaluating the forecast weights $\tilde{\mathbf{w}}$ for all new PLCs in parallel.

Possible statistical improvements include online algorithms for kernel density algorithms (Kristan, Leonardis, and Skočaj, 2011; Lambert, Harrington, Harvey, and Glodjo, 1999) for faster updating, as well as entirely new methods to estimate the predictive state space.

9.1.1 *Nonparametric High-Dimensional Density Estimation*

In the updating step of posterior probabilities (Eq. (7.14)), it is necessary to estimate $\mathbb{P}\left(\ell_i^- \mid S_i = s_j\right)$; a possibly very high-dimensional distribution. Since we require

estimation and evaluation in every iteration of the EM algorithm we have to do this efficiently. For statistical as well as computational reasons we therefore deviate slightly from the nonparametric framework and use multivariate Gaussian distributions to approximate those densities.

Optimally we would like to have a fully nonparametric estimator for this update. However, such a generalization to a fully nonparametric estimator must be fast enough to handle both high-dimensional samples as well as large N cases. One approach could use forest density estimators (Chow and Liu, 1968; Liu et al., 2011), but other efficient estimators for $\mathbb{P}\left(\ell_i^- \mid S_i = s_j\right)$ can be used.

9.2 CONTINUOUS STATE SPACE MODELS

Both theory and proposed estimators work for continuous-valued data, but they still require a discretized state space. For some problems a continuous state space might be a better fit for the question we are trying to answer.

Based on the optimal mixture model interpretation of predictive states, one could use the connection of finite mixture models with mixed membership models. Whereas in finite mixture models a sample comes with a certain probability from one and only one cluster, mixed membership models allow one sample to be a true mix of several clusters. Thus one natural extension of the current work to a continuous state space could use Dirichlet distributions (or any other continuous distribution on the probability simplex) on the weights $\mathbf{w}_i$. It must be pointed out though, that especially in our nonparametric setting care must be taken with respect to identifiability of such a model.

9.3 MEASURES OF UNCERTAINTY

We currently show that LICORS is a consistent estimator of the predictive state space, but we do not provide theoretical measures uncertainty of these estimates based on a single realization. The comparison over multiple datasets in the fMRI experiments suggests that it is well-behaved, but the theory still lacks of proper uncertainty bounds (bias, standard errors, confidence intervals, rates of convergence, etc.). Currently, we overcome this in our application of LSC to fMRI data by using multiple replications and then estimate sample mean and standard error using the cross-section. With the fully probabilistic model in Chapter 6, I made this light cones and predictive states setting more widely accessible to a statistical and machine learning audience, thus opening the door for further theoretical research on estimator and algorithm properties.

9.4 APPLICATIONS

Many scientific fields study quantities that vary over space and time. For example, climatology, organizational biology, computational chemistry, or material sciences - just to name a few. In this thesis I apply LICORS and LSC in the context of neuroscience and fMRI experiments. But the methods I develop and the software I provide are generally applicable: they automatically estimate optimal forecasts and discover patterns for a very large class of spatio-temporal processes. Thus researchers can use LICORS and LSC without much modification to optimally forecast and find interesting patterns in their spatio-temporal data. To facilitate such an analysis, I made most methods publicly available (see Appendix B).

10

CONCLUSION

Willst Du Dich am Ganzen erquicken,
so musst Du das Ganze im Kleinsten erblicken.

Johann Wolfgang von Goethe

In this thesis, I develop a generally applicable statistical methodology to address three challenging problems in the analysis of spatio-temporal data: i) pattern discovery, ii) learning of the underlying spatio-temporal dynamics, and iii) optimal forecasting.

Previous work in physics and discrete math focused on discrete-valued fields which limited its use to a small subset of real-world problems. Here I embed predictive state estimation in a statistical framework and obtain fully probabilistic models and well-behaved estimators for continuous-valued fields. Most results and applications were motivated on $(1+1)$D and $(2+1)$D fields, but both theory and practice extend without modification to higher-dimensional and arbitrarily shaped fields.

Based on previous work from Shalizi *et al.*, I present two new nonparametric predictive state estimators, hard and mixed LICORS, which can be used for optimal prediction of continuous-valued spatio-temporal processes. Hard LICORS is a provably consistent estimator and simulations show that it also has good finite sample predictive power. Mixed LICORS is a probabilistic generalization of previously

proposed algorithms for predictive state recovery; simulations show that it has excellent finite sample properties. I also apply those methods to fMRI data where the optimal predictors provide the basis for automated pattern discovery using local statistical complexity (LSC). Contrary to many algorithms in the statistics, machine learning, and signal processing literature, LSC can detect interesting dynamical patterns automatically from the data without any prior user input. Last but not least, I make most methods publicly available in R and Python packages.

This thesis provides researchers with powerful, principled, and highly automatic methods to analyze and optimally forecast complex spatio-temporal data.

Part V

APPENDIX

A

PROOFS

This is a one line proof ... if we start sufficiently far to the left.

Anonymous

A.1 LICORS CONSISTENCY IN THE ORACLE CASE

Proof of Theorem 5.2.12. Recall that

$$\widehat{\theta}_{\mathrm{MLE}}(i) = \arg\max_{p_k \in \mathcal{P}} \log L(p_k; \mathbf{F}_i(\delta)). \tag{A.1}$$

By the union bound

$$\mathbb{P}\left(\widehat{\mathbf{B}}_{\mathrm{MLE}} \neq \mathbf{B}\right) = \mathbb{P}\left(\bigcup_{i=1}^{N} \{\text{row } i \text{ is incorrect}\}\right) \tag{A.2}$$

$$\leqslant \sum_{i=1}^{N} \mathbb{P}\left(\text{row } i \text{ is incorrect}\right) \tag{A.3}$$

$$= \sum_{i=1}^{N} \mathbb{P}\left(\widehat{\theta}_{\mathrm{MLE}}(i) \neq k \mid \epsilon(\ell_i^-) = p_k\right). \tag{A.4}$$

Thus it remains to show that for all $i = 1, \ldots, m(N)$,

$$\mathbb{P}\left(\widehat{\theta}_{\mathrm{MLE}}(i) \neq k \mid \epsilon(\ell_i^-) = p_k\right) \xrightarrow[N\to\infty]{} 0\,, \tag{A.5}$$

sufficiently fast.

Let

$$\widehat{\Lambda}_i^{(j,k)} = \log L(p_j; \mathbf{F}_i(\delta)) - \log L(p_k; \mathbf{F}_i(\delta)) = \sum_{x \in I_i(\delta)} \log \frac{p_j(\ell_x^+)}{p_k(\ell_x^+)} \tag{A.6}$$

be the log-likelihood ratio between state j and k for the δ-sample of PLC ℓ_i^-. Since $\widehat{\theta}_{MLE}(i)$ maximizes (5.11)

$$\widehat{\Lambda}_i^{(\widehat{\theta}_{MLE}(i),k)} \geqslant 0 \text{ for all } k \neq \widehat{\theta}_{MLE}(i). \tag{A.7}$$

For a finite set of alternatives and continuous random variables, Eq. (A.7) has equality if and only if $k = \widehat{\theta}_{MLE}(i)$ with probability one. The event $\{\widehat{\theta}_{MLE}(i) \neq k \mid p_k\}$ is equivalent to the existence of at least one j with $\widehat{\Lambda}_i^{(j,k)} > 0$, which can be bounded by

$$\mathbb{P}\left(\widehat{\theta}_{MLE}(i) \neq k \mid p_k\right) = \mathbb{P}\left(\exists j : \widehat{\Lambda}_i^{(j,k)} > 0 \mid \epsilon(\ell_i^-) = p_k\right) \tag{A.8}$$

$$\leqslant \sum_{k=1}^{m(N)} \mathbb{P}\left(\widehat{\Lambda}_i^{(j,k)} > 0 \mid \epsilon(\ell_i^-) = p_k\right). \tag{A.9}$$

Remark A.1.1 (Different support)**.** *We have not made any assumptions on the support of the predictive state distributions, and so allow for infinite divergence between them. This is not a difficulty, since infinite KL divergence simply means that the power of likelihood-ratio tests grows super-exponentially, which only improves our analysis. Therefore, if we only explicitly treat the case where all KL divergences are finite, we are being conservative.*

By Assumption 5.2.3 the log-likelihood ratio for any single FLC is bounded

$$\log \frac{\iota}{\kappa} < \log \frac{p_j(\ell^+)}{p_k(\ell^+)} < \log \frac{\kappa}{\iota}. \tag{A.10}$$

$\widehat{\Lambda}_i^{(j,k)}$ is a sum of bounded, IID random variables, so we could use Hoeffding's bound (Hoeffding, 1963) on each term in (A.9). However, the number of terms in

(A.6) is random: first, we condition on $S_i(N,\delta)$, bound the error probabilities, and then take an expectation over S_i.

Given $\{S_i(N,\delta) = s_i(N,\delta)\}$, Hoeffding's inequality says

$$\begin{aligned}\mathbb{P}\left(\widehat{\Lambda}_i^{(j,k)} > 0 \mid \epsilon(\ell_i^-) = p_k, S_i(N,\delta) = s_i(N,\delta)\right) &\leqslant e^{-2s_i(N,\delta)\mathcal{D}_{KL}\left(p_j \| p_k\right)^2/a^2} \\ &= e^{-\tilde{c}s_i(N,\delta)d_{j,k}^2} \end{aligned} \tag{A.11}$$

where $a = \left(\log\frac{\kappa}{\iota} - \log\frac{\iota}{\kappa}\right) > 0$ and $\tilde{c} = 2c^{-2}$. Lower-bounding in the exponent with the minimum distance $d_{min} = \min_{j,k} d_{j,k}$, the upper bound becomes independent of j and k:

$$\begin{aligned}\sum_{k=1}^{m(N)} \mathbb{P}\left(\widehat{\Lambda}_i^{(j,k)} > 0 \mid \epsilon(\ell_i^-) = p_k, S_i(N,\delta) = s_i(N,\delta)\right) &\leqslant \sum_{k=1}^{m(N)} e^{-\tilde{c}s_i(N,\delta)d_{j,k}^2} \\ &\leqslant m(N)e^{-\tilde{c}s_i(N,\delta)d_{min}^2}. \end{aligned} \tag{A.12}$$

Using the union bound again,

$$\begin{aligned}\mathbb{P}\left(\widehat{\mathbf{B}}_{MLE} \neq \mathbf{B} \mid S_i(N,\delta) = s_i(N,\delta)\right) &\leqslant m(N)\sum_{i=1}^{N} e^{-\tilde{c}s_i(N,\delta)d_{min}^2} \\ &\leqslant Nm(N)e^{-\tilde{c}s_{min}(N,\delta)d_{min}^2}, \end{aligned} \tag{A.13}$$

where $s_{min}(N,\delta) = \min_i s_i(N,\delta)$. Taking expectation over $S_{min}(N,\delta)$ gives

$$\begin{aligned}\mathbb{E}_S\left[Nm(N)e^{-\tilde{c}s_{min}(N,\delta)d_{min}^2}\right] &= Nm(N)\sum_{s=1}^{\infty}\mathbb{P}\left(S_{min}(N,\delta) = s\right)e^{-\tilde{c}sd_{min}^2} \\ &= Nm(N)\mathbb{E}e^{-\tilde{c}d_{min}^2 S_{min}(N,\delta)} \\ &\xrightarrow[N\to\infty]{} 0, \end{aligned}$$

using Assumption 5.2.10 in the last line. □

Proof of Corollary 5.2.13. The divergence between states, $d_{i,j}$, can tend to zero as long as $d_{\min}^2$ decays more slowly than the minimum number of samples in each neighborhood of the FLC, $S_{\min}(N, \delta)$, grow. Analogous to the proof of Lemma 5.2.7, choosing $\delta = o(1)$ such that ρ goes faster to zero than $d_{\min}^2$ guarantees that $\mathbf{F}_i(\delta)$ only contains samples from the predictive state of PLC ℓ_i^-. □

A.2 LICORS CONSISTENCY IN THE NON-ORACLE CASE

Proof of Lemma 5.2.5.

$$
\begin{aligned}
&\mathbb{P}\left(L^+(\mathbf{r},t), L^+(\mathbf{u},s) \mid S(\mathbf{r},t), S(\mathbf{u},s)\right) \\
&= \mathbb{P}\left(L^+(\mathbf{r},t) \mid L^+(\mathbf{u},s), S(\mathbf{r},t), S(\mathbf{u},s)\right) \mathbb{P}\left(L^+(\mathbf{u},s) \mid S(\mathbf{r},t), S(\mathbf{u},s)\right) \\
&= \mathbb{P}\left(L^+(\mathbf{r},t) \mid L^+(\mathbf{u},s), S(\mathbf{r},t), S(\mathbf{u},s)\right) \mathbb{P}\left(L^+(\mathbf{u},s) \mid S(\mathbf{u},s)\right) \\
&= \mathbb{P}\left(L^+(\mathbf{r},t) \mid S(\mathbf{r},t)\right) \mathbb{P}\left(L^+(\mathbf{u},s) \mid S(\mathbf{u},s)\right) \ .
\end{aligned}
$$

The first equality is simple conditioning, the second equality holds since given the predictive state at $(\mathbf{u}, s)$ the distribution of L^+ is independent of the predictive state at another $(\mathbf{r}, t)$, and the last equality holds for the same reason as the second plus the non-overlap of the FLCs at $(\mathbf{r}, t)$ and $(\mathbf{u}, s)$. □

Proof of Corollary 5.2.6. The FLC of $(\mathbf{r}, t)$ with $h_f = 0$ is just the single point $X(\mathbf{r}, t)$. Since two univariate FLCs cannot overlap unless they are equal, the result follows immediately from Lemma 5.2.5. □

Proof of Lemma 5.2.7. By contradiction. Assume that ℓ_j^- and ℓ_k^-, with $j, k \in I_i(\delta)$, have different predictive states, without loss of generality ϵ_1 and ϵ_2. By Assumption 5.2.4, then, $\mathcal{D}_{KL}(\epsilon_1 \parallel \epsilon_2)$ and $\mathcal{D}_{KL}(\epsilon_2 \parallel \epsilon_1)$ are both at least $d_{\min}$. By definition of $I_i(\delta)$, $\|\ell_j^- - \ell_k^-\| < 2\delta$. By Assumption 5.1.2, then, $\mathcal{D}_{KL}(\epsilon_1 \parallel \epsilon_2)$ and $\mathcal{D}_{KL}(\epsilon_2 \parallel \epsilon_1)$ are both at most $\rho(2\delta)$. But by making δ sufficiently small, $\rho(2\delta)$ can be made as

small as desired, and in particular, can be made less than $d_{\min}$. This is a contradiction, so all past cone configurations in $I_i(\delta)$ must be predictively equivalent. □

Proof of Corollary 5.2.8. Immediate from combining Lemmas 5.2.7 and 5.2.5. □

Proof of Theorem 5.2.15. Before going into the formal proof, we make an observation regarding nonparametric two-sample tests. Most of these, to have good operating characteristics, require independent samples. Since we will be applying the tests to $\mathbf{F}_i(\delta)$ and $\mathbf{F}_j(\delta)$,

Properties A.2.1 (Pairwise independent samples)**.** *If*

$$I_i(\delta) \cap I_j(\delta) = \emptyset. \tag{A.14}$$

then the samples $\mathbf{F}_i(\delta)$ *are independent of* $\mathbf{F}_j(\delta)$, $j \neq i$ *(see* (5.8)*).*

Let $\Delta_{ij} := \|\ell_i^- - \ell_j^-\|$. If $\Delta_{ij} > 2\delta$, then (A.14) is satisfied. If $\Delta_{ij} < 2\delta$, then a sample in $\mathbf{F}_i(\delta)$ might also appear in $\mathbf{F}_j(\delta)$ and therefore violate the independence assumption for two sample tests.

For these rare cases redefine the index set $I_i(\delta)$ and $I_j(\delta)$ such that (A.14) holds. We can achieve this by excluding the intersection, split it in half (± 1 sample), and then re-assign these halves to each index set. For all pairs $i \neq j$, determine $I_i(\delta) \cap I_j(\delta) =: I_{i\cap j}(\delta)$. Then let

$$I_i := I_i \setminus I_{i\cap j} \cup \{i_1, \ldots, i_{|I_{i\cap j}|/2} \mid i_k \in I_{i\cap j}\} \tag{A.15}$$

$$\text{and } I_j := I_j \setminus I_{i\cap j} \cup \{i_{|I_{i\cap j}|/2}, \ldots, i_{|I_{i\cap j}|} \mid i_k \in I_{i\cap j}\}. \tag{A.16}$$

If $I_{i\cap j} = \emptyset$, (A.15)–(A.16) does not change the index set; if $I_{i\cap j} \neq \emptyset$, then (A.15)–(A.16) guarantees an empty intersection.

The proof of consistency relies crucially on a growing index set I_i. The redefinition in (A.15)–(A.16) does not change the rate at which $S_i(N, \delta)$ grows, because in the worst case (for close PLCs) it just divides $s_i(N, \delta)$ and $s_j(N, \delta)$ in half.

PROOF: We first bound the error for each row $\widehat{\mathbf{A}}_i$, and then use a union bound for the probability of error for $\widehat{\mathbf{A}}$.

BOUND ERROR PER ROW For each row $T_{n,m}$ tests $H_0 : \ell_i \sim \ell_j$, $j > i$ (due to symmetry the cases $j < i$ have already been tested before) based on the sample $\mathbf{F}_i(\delta) \sim \epsilon_i$ and $\mathbf{F}_j(\delta) \sim \epsilon_j$. The worst-case distance d for the nonparametric test in Assumption 5.2.14 is $d = d_{\min}$. For simplicity, consider the first row: here we have to make $N-1$ tests, of which $N_1 - 1$ should correctly accept, and $N - N_1$ should correctly reject equality of distributions.

$$\begin{aligned}
\mathbb{P}\left(\widehat{\mathbf{A}}_j \neq \mathbf{A}_j\right) &\leqslant (N_j - 1)\mathbb{P}\left((\text{type I}) + (N - N_j)\right)\mathbb{P}\left(\text{type II}\right) && \text{(A.17)}\\
&\leqslant (N_j - 1)\alpha + (N - N_j)\beta\left(\alpha, S_{\min}(N,\delta), S_{\min}(N,\delta)\right) && \text{(A.18)}\\
&\leqslant N_j\alpha + (N - N_j)\beta\left(\alpha, S_{\min}(N,\delta), S_{\min}(N,\delta)\right) && \text{(A.19)}
\end{aligned}$$

since the worst case, for type II error, is that both samples are as small as possible.

BOUND ERROR FOR ENTIRE MATRIX The probability of error for the entire predictive state clustering can again be bounded using the union bound:

$$\begin{aligned}
\mathbb{P}\left(\widehat{\mathbf{A}} \neq \mathbf{A}\right) = \mathbb{P}\left(\bigcup_{j=1}^{N}\{\widehat{\mathbf{A}}_j \neq \mathbf{A}_j\}\right) &\leqslant \sum_{j=1}^{N}\mathbb{P}\left(\widehat{\mathbf{A}}_j \neq \mathbf{A}_j\right) && \text{(A.20)}\\
&\leqslant N\left(N_{\max}\alpha + (N - N_{\min})\beta\left(\alpha, S_{\min}(N,\delta), S_{\min}(N,\delta)\right)\right) && \text{(A.21)}\\
&= NN_{\max}\alpha + (N^2 - NN_{\min})\beta\left(\alpha, S_{\min}(N,\delta), S_{\min}(N,\delta)\right), && \text{(A.22)}
\end{aligned}$$

where $N_{\max} = \max_j N_j$ is the number of light cones in the largest predictive state.

Under Assumption 5.2.14, α and β are both $o(NN_{\max})$, so the over-all error probability tends to zero. □

B

ALGORITHMS & CODE

Everyone knows that debugging is twice as hard as writing a program in the first place.
So if you are as clever as you can be when you write it,
how will you ever debug it?

Brian Kernighan

All computations and simulations were done in R (R Development Core Team, 2010) and Python (Van Rossum, 2003). For detailed references on third party libraries see the manuals of the software packages below. For the Python implementation, I want to highlight the OpenCV library (Bradski, 2000) as a major component in the implementation of hard LICORS.

B.1 PYLICORS: A PYTHON LIBRARY FOR PREDICTIVE STATE ESTIMATION

pyLICORS provides a Python interface for (hard) LICORS estimation and is publicly available at `http://pypi.python.org/pypi/pyLICORS/`. For implementation details, current status, and future developments check the online manuals.

B.2 LICORS & LSC: R PACKAGES FOR ESTIMATION AND VISUALIZATION OF PREDICTIVE STATES AND LOCAL STATISTICAL COMPLEXITY

The R packages LICORS and LSC accompany this thesis. At the moment of finishing this manuscript the current versions are 0.1.1 and 0.1, respectively. **LICORS** implements mixed LICORS (EM-like algorithm) along with some auxiliary functions. **LSC** estimates local statistical complexity (LSC) from a given state space, and also has a wrapper function `LICORS2LSC` to handle output from **LICORS**. For details on the implementation, current status, and future developments check the online package manuals available at the Comprehensive R Archive Network (CRAN):

- `cran.r-project.org/web/packages/LICORS/index.html` and
- `cran.r-project.org/web/packages/LSC/index.html`.

C

DATA

"Data! Data! Data!" he cried impatiently.
"I can't make bricks without clay."

Sherlock Holmes in Sir Arthur Conan Doyle's
The Adventure Of The Copper Beeches

I use three main datasets in this thesis:

SIMULATIONS: The $(1+1)$D fields in the simulations are fully specified by Eqs. (5.16) – (5.18) and can be easily implemented in standard software packages.

For the sake of reproducibility, I made the running example dataset (state space in Fig. 5.2a and observations in Fig. 5.2b) available in the R package LICORS, dataset `contCA00`.

CANDLE VIDEO: The candle dataset is a cropped version of publicly available video "Candle Being Extinguished" (`www.youtube.com/watch?v=ucNQhsBOs54`).

FMRI: The datasets we use in Section 8.2 were courteously shared by Dr. Elisha P. Merriam. The experimental protocol for the fMRI experiments can be found in Schlupeck et al. (2010) (harmonic stimulus) and Freeman et al. (2011) (non-harmonic stimulus).

BIBLIOGRAPHY

Abello, J., A. Buchsbaum, and J. Westbrook (1998). A functional approach to external graph algorithms. In *Proceedings of the 6th European Symposium on Algorithms*, Berlin. Springer.

Antos, A. and I. Kontoyiannis (2001). Estimating the entropy of discrete distributions. In *Information Theory, 2001. Proceedings. 2001 IEEE International Symposium on*, pp. 45.

Arthur, D. and S. Vassilvitskii (2007). `k-means++`: The advantages of careful seeding. In H. Gabow (Ed.), *Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms [SODA07]*, Philadelphia, pp. 1027–1035. Society for Industrial and Applied Mathematics.

Ash, R. B. and C. Doléans-Dade (2000). *Probability and measure theory*. San Diego, CA: Academic Press.

Bach, F., R. Jenatton, J. Mairal, and G. Obozinski (2012). Optimization with sparsity-inducing penalties. *Found. Trends Mach. Learn. 4*(1), 1–106.

Benaglia, T., C. D. and D. R. Hunter (2009a). An EM-like algorithm for semi- and non-parametric estimation in multivariate mixtures. *Journal of Computational and Graphical Statistics 18*(2), 505–526.

Benaglia, T., C. D. and D. R. Hunter (2009b). Bandwidth Selection in an EM-like algorithm for nonparametric multivariate mixtures. http://hal.archives-ouvertes.fr/docs/00/35/32/97/PDF/npEM_bandwidth.pdf.

Biernacki, C., G. Céleux, and G. Govaert (1999). An improvement of the NEC criterion for assessing the number of clusters in a mixture model. *Non-Linear Anal. 20*, 267–272.

Biernacki, C. and G. Govaert (1998). Choosing Models in Model-Based Clustering and Discriminant Analysis. Technical report, Institut National de Recherche en Informatique et en Automatique.

Bordes, L., D. Chauveau, and P. Vandekerkhove (2007). A stochastic EM algorithm for a semiparametric mixture model. *Comput. Stat. Data Anal. 51*(11), 5429–5443.

Bosq, D. (1998). *Nonparametric Statistics for Stochastic Processes: Estimation and Prediction* (Second ed.). Berlin: Springer-Verlag.

Bosq, D. and D. Blanke (2007). *Inference and Prediction in Large Dimensions*. New York: Wiley.

Bradski, G. (2000). The OpenCV Library.

Brockwell, P. J. and R. A. Davis (1991). *Time Series: Theory and Methods* (2 ed.). New York, NY: Springer Series in Statistics.

Bunea, F., A. B. Tsybakov, and M. H. Wegkamp (2009). SPADES and Mixture Models. *Annals of Statistics 38*, 2525–2558.

Caires, S. and J. A. Ferreira (2005). On the non-parametric prediction of conditionally stationary sequences. *Statistical Inference for Stochastic Processes 8*, 151–184. Correction, vol. 9 (2006), pp. 109–110.

Capasso, V. and A. Micheletti (2002). Stochastic geometry of spatially structured birth and growth processes: Application to crystallization processes. In E. Merzbach (Ed.), *Topics in Spatial Stochastic Processes*, Berlin, pp. 1–39. Springer-Verlag.

Cappé, O., E. Moulines, and T. Rydén (2005). *Inference in Hidden Markov Models*. New York: Springer.

Chandola, V., A. Banerjee, and V. Kumar (2009). Anomaly detection: A survey. *ACM Comput. Surv. 41*, 15:1–15:58.

Chow, C. and C. Liu (1968). Approximating discrete probability distributions with dependence trees. *Information Theory, IEEE Transactions on 14*(3), 462 – 467.

Cox, R. T. (2001). *Algebra of Probable Inference*. Johns Hopkins University Press.

Cressie, N. A. C. and C. K. Wikle (2011). *Statistics for Spatio-Temporal Data*. New York: Wiley.

Crutchfield, J. P. and K. Young (1989). Inferring statistical complexity. *Physical Review Letters 63*, 105–108.

Dempster, A. P., N. M. Laird, and D. B. Rubin (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society Series B Methodological 39*(1), 1–38.

Fan, J. and Q. Yao (2003). *Nonlinear Time Series: Nonparametric and Parametric Methods*. Berlin: Springer-Verlag.

Fernández, R. and G. Maillard (2005). Chains with complete connections: General theory, uniqueness, loss of memory and mixing properties. *Journal of Statistical Physics 118*, 555–588.

Finkenstädt, B., L. Held, and V. Isham (Eds.) (2007). *Statistical Methods for Spatio-Temporal Systems*. Boca Raton, Florida: Chapman and Hall/CRC.

Freeman, J., G. J. Brouwer, D. J. Heeger, and E. P. Merriam (2011). Orientation decoding depends on maps, not columns. *The Journal of Neuroscience: The Official Journal of the Society for Neuroscience 31*(13), 4792–4804.

Goerg, G. M. (2012a). *LICORS: Light Cone Reconstruction of States - Predictive State Estimation From Spatio-Temporal Data*. R package and manual available at cran.r-project.org/web/packages/LICORS/index.html.

Goerg, G. M. (2012b). *LSC: Local Statistical Complexity - Automatic Pattern Discovery in Spatio-Temporal Data*. R package and manual available at cran.r-project.org/web/packages/LSC/index.html.

Goerg, G. M. (2012c). *pyLICORS: A Python interface for predictive state estimation from spatio-temporal data*. Available at pypi.python.org/pypi/pyLICORS.

Goerg, G. M., E. P. Merriam, C. R. Genovese, and C. R. Shalizi (2012). Local Statistical Complexity: A New Method for Spatio-temporal Pattern Discovery in fMRI Data. Technical report, Carnegie Mellon University, New York University.

Goerg, G. M. and C. R. Shalizi (2012a). LICORS: Light Cone Reconstruction of States for Non-parametric Forecasting of Spatio-Temporal Systems. Technical report, Carnegie Mellon Univsersity, Department of Statistics. In preparation for submission. Available at arxiv.org/abs/1206.2398.

Goerg, G. M. and C. R. Shalizi (2012b). Mixed LICORS: A Nonparametric EM Algorithm for Predictive State Reconstruction. Technical report, Carnegie Mellon University. Submitted for publication. Available at arxiv.org/abs/1211.3760.

Grassberger, P. (1986). Toward a quantitative theory of self-generated complexity. *International Journal of Theoretical Physics 25*, 907–938.

Gretton, A., K. Borgwardt, M. Rasch, B. Schölkopf, and A. Smola (2007). A kernel method for the two sample problem. In *Advances in neural information processing systems 19*, pp. 513–520. MIT Press.

Hall, P., A. Neeman, R. Pakyari, and R. Elmore (2005). Nonparametric inference in multivariate mixtures. *Biometrika 92*(3), 667–678.

Hamilton, J. (1994). *Time Series Analysis*. Princeton University Press.

Hoeffding, W. (1963). Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association 58*, 13–30.

Hyvärinen, A. and E. Oja (2000). Independent Component Analysis: Algorithms and Applications. *Neural Networks 13*, 411–430.

Jaeger, H. (2000). Observable operator models for discrete stochastic time series. *Neural Computation 12*, 1371–1398.

Jänicke, H. (2009). *Information Theoretic Methods for the Visual Analysis of Climate nd Flow Data*. Ph. D. thesis, Universität Leipzig.

Jänicke, H. and G. Scheuermann (2010). Towards automatic feature-based visualization. In H. Hagen (Ed.), *Scientific Visualization: Advanced Concepts*, Volume 1 of *Dagstuhl Follow-Ups*, pp. 62–77. Dagstuhl, Germany: Schloss Dagstuhl—Leibniz-Zentrum fuer Informatik.

Jänicke, H., A. Wiebel, G. Scheuermann, and W. Kollmann (2007). Multifield visualization using local statistical complexity. *IEEE Transactions on Visualization and Computer Graphics 13*, 1384–1391.

Jolliffe, I. T. (2002). *Principal Component Analysis* (2 ed.). New York, NY: Springer.

Kantz, H. and T. Schreiber (2004). *Nonlinear Time Series Analysis* (Second ed.). Cambridge, England: Cambridge University Press.

Knight, F. B. (1975). A predictive view of continuous time processes. *Annals of Probability 3*, 573–596.

Kolmogorov, A. N. (1937). A statistical theory for the recrystallization of metals. *Bulletin of the Academy of Sciences, USSR, Physical Series 1*, 355–359. In Russian.

Kristan, M., A. Leonardis, and D. Skočaj (2011). Multivariate Online Kernel Density Estimation with Gaussian Kernels. *Pattern Recognition 44*(10-11), 2630–2642.

Kruggel, F., D. Y. V. Cramon, and X. Descombes (1999). Comparison of filtering methods for fmri datasets. *NeuroImage 10*, 530–543.

Kullback, S. (1968). *Information Theory and Statistics* (2nd ed.). New York: Dover Books.

Lambert, C. G., S. E. Harrington, C. R. Harvey, and A. Glodjo (1999). Efficient on-line nonparametric kernel density estimation. *Algorithmica 25*, 37–57.

Langford, J., R. Salakhutdinov, and T. Zhang (2009). Learning nonlinear dynamic models. Electronic preprint.

Lauritzen, S. L. (1974). Sufficiency, prediction and extreme models. *Scandinavian Journal of Statistics 1*, 128–134.

Lauritzen, S. L. (1984). Extreme point models in statistics. *Scandinavian Journal of Statistics 11*, 65–91. With discussion and response.

Lee, A. B. and L. Wasserman (2010). Spectral connectivity analysis. *Journal of the American Statistical Association 105*(491), 1241–1255.

Littman, M. L., R. S. Sutton, and S. Singh (2002). Predictive representations of state. In T. G. Dietterich, S. Becker, and Z. Ghahramani (Eds.), *Advances in Neural Information Processing Systems 14 (NIPS 2001)*, Cambridge, Massachusetts, pp. 1555–1561. MIT Press.

Liu, H., M. Xu, H. Gu, A. Gupta, J. Lafferty, and L. Wasserman (2011). Forest density estimation. *J. Mach. Learn. Res. 12*, 907–951.

Lloyd, S. (1982). Least squares quantization in PCM. *Information Theory, IEEE Transactions on 28*(2), 129 – 137.

Lopes, M., L. Jacob, and M. J. Wainwright (2011). A More Powerful Two-Sample Test in High Dimensions using Random Projection. In J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. C. N. Pereira, and K. Q. Weinberger (Eds.), *NIPS*, pp. 1206–1214.

Lütkepohl, H. (2007). *New Introduction to Multiple Time Series Analysis*. Heidelberg: Springer.

Mallapragada, P. K., R. Jin, and A. K. Jain (2010). Non-parametric Mixture Models for Clustering. In E. R. Hancock, R. C. Wilson, T. Windeatt, I. Ulusoy, and F. Escolano (Eds.), *SSPR/SPR*, Volume 6218 of *Lecture Notes in Computer Science*, pp. 334–343. Springer.

Marchini, J. L. and B. D. Ripley (2000). A New Statistical Approach to Detecting Significant Activation in Functional MRI. *NeuroImage 12*, 366–380.

Merriam, E. P., C. R. Genovese, and C. L. Colby (2007). Remapping in human visual cortex. *Journal of Neurophysiology 97*(2), 1738–1755.

Parlitz, U. and C. Merkwirth (2000). Prediction of spatiotemporal time series based on reconstructed local states. *Physical Review Letters 84*, 1890–1893.

Pilanci, M., L. El Ghaoui, and V. Chandrasekaran (2012). Recovery of Sparse Probability Measures via Convex Programming. In *NIPS 2012*.

R Development Core Team (2010). *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. ISBN 3-900051-07-0.

Rizzo, M. L. and G. J. Székely (2010). DISCO analysis: A nonparametric extension of analysis of variance. *Annals of Applied Statistics 4*, 1034–1055.

Rosenbaum, P. R. (2005). An exact distribution-free test comparing two multivariate distributions based on adjacency. *Journal of the Royal Statistical Society Series B 67*, 515–530.

Salmon, W. C. (1984). *Scientific Explanation and the Causal Structure of the World*. Princeton: Princeton University Press.

Schlupeck, D., E. Merriam, R. Sanchez-Panchuelo, S. Francis, R. Bowtell, P. Velasco, S. Inati, and D. Heeger (2010). The spatial precision of high-resolution functional MRI. In *Society for Neuroscience Abstracts*, Volume 37.

Shalizi, C. R. (2003). Optimal nonlinear prediction of random fields on networks. *Discrete Mathematics and Theoretical Computer Science AB(DMCS)*, 11–30.

Shalizi, C. R. and J. P. Crutchfield (2001). Computational mechanics: Pattern and prediction, structure and simplicity. *Journal of Statistical Physics 104*, 817–879.

Shalizi, C. R., R. Haslinger, J.-B. Rouquier, K. L. Klinkner, and C. Moore (2006). Automatic filters for the detection of coherent structure in spatiotemporal systems. *Physical Review E 73*, 036104.

Shalizi, C. R., K. L. Klinkner, and R. Haslinger (2004). Quantifying self-organization with optimal predictors. *Physical Review Letters 93*, 118701.

Shannon, C. E. (1948). A Mathematical Theory of Communication. *Bell System Technical Journal 27*, 379–23, 623–656.

Song, S. and P. J. Bickel (2011). Large Vector Auto Regressions. Available at http://arxiv.org/pdf/1106.3915.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society (Series B) 58*, 267–288.

Tibshirani, R., G. Walther, and T. Hastie (2000). Estimating the Number of Clusters in a Dataset via the Gap Statistic. *Journal of the Royal Statistical Society, Series B 63*, 411–423.

Van Rossum, G. (2003). *The Python Language Reference Manual*. Network Theory Ltd.

von Luxburg, U. (2006). A tutorial on spectral clustering. Technical Report 149, Max Planck Institute for Biological Cybernetics.

Wiskott, L. and T. J. Sejnowski (2002). Slow Feature Analysis: Unsupervised Learning of Invariances. *Neural computation 14*(4), 715–770.

Wong, J. (2012). *fastVAR*. R package version 1.2.1.

Yang, M.-S. (1993). On a class of fuzzy classification maximum likelihood procedures. *Fuzzy Sets Syst. 57*(3), 365–375.