

# Optimal Nonparametric Prediction and Automated Pattern Recognition in Dynamical Space-Time Systems

Thesis Proposal

Georg M. Goerg

[gmg@stat.cmu.edu](mailto:gmg@stat.cmu.edu)

Department of Statistics  
Carnegie Mellon University

Advisors: Cosma Shalizi and Larry Wasserman

April 8, 2011

## Abstract

Many methods in statistics, machine learning, and signal processing, such as speech analysis or pattern recognition in images and videos, try to extract informative structures from a dynamic system and remove noisy uninformative parts. Although such methods and algorithms work well in practice, they often do so because they have been specifically tuned to work in a very particular setting, and thus may break down when conditions and properties of the data do not hold anymore.

It would be very useful to have an automated pattern recognition method for dynamic system, which does not rely on any particular model or data structure, but gives informative patterns for any kind of system. [Shalizi \(2003\)](#) showed for discrete fields that an automated pattern discovery can be constructed by a characterization and classification of local conditional predictive distributions. The underlying idea is that statistically optimal predictors not only predict well but - for this very reason - also describe the data well, and therefore reveal informative structure inherent in the system.

In this thesis I extend previous work from [Shalizi, Shalizi, and Haslinger \(2004\)](#) to obtain a fully automated pattern recognition for continuous-valued space-time systems - such as videos - by means of optimal local prediction of the space-time field. Applications to simulated one-dimensional spatial dynamics and a real-world image pattern recognition demonstrate the usefulness and generality of the presented methods.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Discrete-valued fields: Cellular Automata . . . . .	2
<b>2</b>	<b>Local Statistical Complexity</b>	<b>3</b>
2.1	Causal States . . . . .	5
2.2	Estimating causal states from data . . . . .	7
<b>3</b>	<b>Comparing Probability Distributions</b>	<b>7</b>
3.1	Testing equality of distributions . . . . .	7
3.1.1	Estimating the null distribution . . . . .	8
<b>4</b>	<b>Applications</b>	<b>8</b>
<b>5</b>	<b>Continuous-valued systems: Gaussian random fields</b>	<b>9</b>
5.1	Simulations . . . . .	10
<b>6</b>	<b>Proposed Work</b>	<b>12</b>
6.1	Continuous valued fields . . . . .	12
6.2	Statistical formulation and theory of causal states . . . . .	12
6.2.1	Non-parametric estimation and testing in high dimensions . . . . .	12
6.2.2	(Spectral) clustering in distribution space . . . . .	12
6.2.3	Prediction . . . . .	13
6.2.4	Cross-validation . . . . .	13
6.2.5	Hypothesis testing . . . . .	14
6.3	Properties of the LSC estimator . . . . .	15
6.4	Applications . . . . .	15
	<b>Bibliography</b>	<b>16</b>
<b>A</b>	<b>Information Theory</b>	<b>19</b>
A.1	Kullback-Leibler divergence and variants . . . . .	21
A.1.1	J-divergence . . . . .	21
A.1.2	Resistor divergence . . . . .	21
<b>B</b>	<b>Cross Validation</b>	<b>21</b>
<b>C</b>	<b>Algorithms and Implementation</b>	<b>22</b>
<b>D</b>	<b>Cellular Automata in Detail</b>	<b>23</b>
<b>E</b>	<b>Local Sensitivity</b>	<b>24</b>

# 1 Introduction

Many methods in signal processing try to find patterns and reduce noise in data. For example, in image recognition it is common practice to define features that identify e.g. a face: eyebrows, eyes, hair. In anomaly detection in videos the user defines a feature as a region of high density of such-and-such colored values for example. On a more general level, many algorithms try to express the data in a better/more interesting coordinate system 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) defines interesting to mean slowly varying (Wiskott and Sejnowski, 2002); Laplacian graphs and diffusion maps find connected points in a high dimensional space (Lee and Wasserman, 2010; von Luxburg, 2006), for a brief list of common methods in the statistics and machine learning literature.

Although they work extremely well in a wide-range of real world applications, algorithms must know what to look for, i.e. someone has to decide what is *interesting*. This works fine for well-defined tasks such as face/smile recognition in images or the identification of specific objects in videos, 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 and/or high dimensional.

Going even one step further, it is impossible to supply useful features to a pattern recognition algorithm if even we do not know what to look for. For example, we cannot tell an algorithm to scan a video for interesting parts, without knowing its content: interesting events of a highway surveillance video are clearly different to interesting features in a recording of microscopic heart muscle activity. Yet, the moment we watch the video it is often immediately clear to us which parts are interesting and which not.

Hence, for (upcoming) real world 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, watch a surveillance video or recordings of microscopic heart muscle activity, or study an object evolving over time in Euclidean space.

Shalizi (2003) showed for discrete-valued fields that an automated pattern discovery can be constructed by a characterization and classification of local conditional predictive distributions. The underlying idea is that statistically optimal predictors not only predict well, but for this very reason also describe the data in an informative way, and therefore finding these optimal predictors should reveal structure inherent in the system. While the fundamental concepts and most theoretical results are also valid for continuous random variables (RVs) - except for minor modifications-, Shalizi (2003); Shalizi et al. (2004) only give an algorithm to obtain the optimal predictors for the discrete case and apply them to discrete-valued data. Additionally, only some statistical properties of the estimators are studied in detail.

Thus in this thesis I want to put their methods on a statistical basis, extend previous work to accommodate continuous-valued space-time systems, and apply these automated pattern recognition algorithms to real world data from 1D (images) and 2D (videos) random fields.<sup>1</sup>

**Definition 1.1** ((Space-time) Random Field). *A collection of random variables (RVs)  $\{X(\mathbf{r}, t) \mid \mathbf{r} \in \mathbf{S}, t \in \mathbb{T}\}$ , where  $t$  is time and  $\mathbf{r}$  is a spatial index varying in  $\mathbf{S}$ , is called a space-time random field (RF).*<sup>2</sup>

---

<sup>1</sup>See Section 4 for a list of possible applications.

<sup>2</sup>Unless stated otherwise, a random field will always denote a space-time random field.

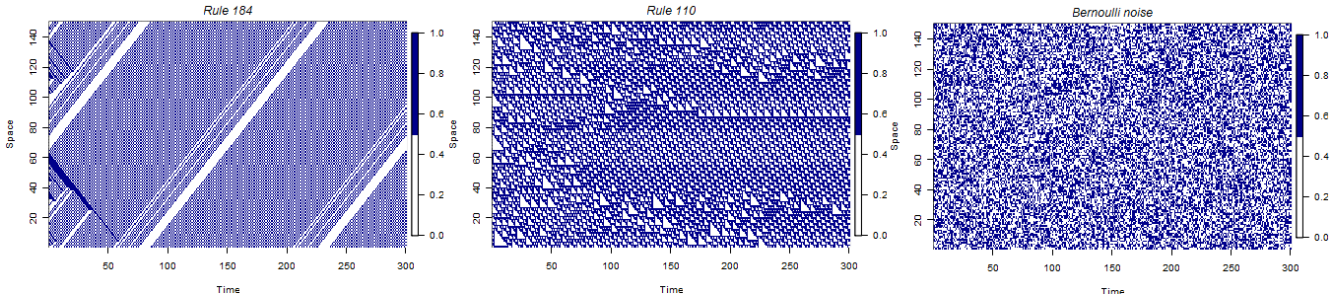


Figure 1: Dynamic structure in Cellular Automata. Space is 1 dimensional (vertical), time goes from left to right. At each time  $t_i$  all pixels  $x(\cdot, t_i)$  are updated simultaneously according to their local neighborhood. See Section 1.1 for details.

In this work, space  $\mathbf{S}$  and time  $\mathbb{T}$  are both discrete, but  $X(\mathbf{r}, t)$  can be real-valued. A typical example of a continuous RF is a video, where one frame can be thought of as a  $2D$  stochastic process on a discrete lattice and we observe it over discrete time  $\mathbb{T} = \{t_1, \dots, t_n\}$ . For example, a 10 second video with 25 frames per second (fps) with a  $800 \times 600$  pixel resolution can be modeled as a  $(2 + 1)D$  RF with  $\mathbb{T} = \{1, \dots, 250\}$  and  $\mathbf{S} = \{(x, y) \mid x \in \{1, \dots, 600\}, y \in \{1, \dots, 800\}\}$ .

For the purpose of demonstration I will first consider a particular example of  $(1 + 1)D$  dynamical systems: one-dimensional cellular automata (CA).

## 1.1 Discrete-valued fields: Cellular Automata

Cellular Automata (CA) are dynamic systems, which consist of cells in a spatial domain that update themselves depending on their local past neighborhood. Although the updating rules can be very simple, the global structure and organization of the evolving system can be surprisingly complex. For so called “elementary” one-dimensional binary CAs - see upper row of Fig. 1 - a rule describes how each pixel  $x(\mathbf{r}, t)$  changes according to its three immediate adjacent past neighbors  $x(\mathbf{r} - \mathbf{1}, t - 1)$ ,  $x(\mathbf{r}, t - 1)$ , and  $x(\mathbf{r} + \mathbf{1}, t - 1)$  (Wolfram, 1983). See Appendix D for a detailed explanation of how Cellular Automata work.

Figure 1 shows simulations<sup>3</sup> of rule 110 and 184, which are good examples to illustrate the ideas motivating the thesis. For a baseline comparison Fig. 1 also shows a collection of independent identically distributed (iid) Bernoulli noise. All images are  $150 \times 300$  meaning that the system extends 150 units in space (vertically) and runs for  $T = 300$  iterations (from left to right). Although all three systems start from the same random (Bernoulli) initial configurations  $X(\mathbf{r}, 1)$ , just in a couple of steps different structure evolves according to the individual rules.

By construction the Bernoulli iid field does not show any patterns neither in space nor in time; thus as far as pattern recognition goes it is very uninteresting. Rule 184 is not independent over space and time, but quickly settles to a very predictable behavior: a trace of white space going from the lower left to upper right with a perfectly repetitive checkerboard-like background. The deterministic patterns, which are only broken due to the image boundaries, make it only a little more interesting than the iid image. In contrast, rule 110 generates much more complex dynamics with several different local patterns (lower - middle - upper part of the image) propagating over time. It is clearly more structured than the Bernoulli

<sup>3</sup>Package `CellularAutomaton` in R, (R Development Core Team, 2010).

field, yet there are no visible deterministic patterns, although we might have some idea of its evolution for few time steps into the future. In that sense, rule 110 is much less predictable than rule 184, and at the same time much more structured than Bernoulli noise, which makes it more interesting than any of these two.

Optimally, a measure of *interestingness* should reflect our intuition of how these systems evolve over time. 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, rule 184 generates interesting structure at the very beginning but quickly shows very uninteresting patterns for the rest of the time; an *interestingness* measure should reflect this drop.

## 2 Local Statistical Complexity

One measure to identify interesting behavior in a space-time system can be complexity or self-organization. In the literature there are various notions of complexity and self-organization (Feldman and Crutchfield, 1997); I will use *local statistical complexity* (LSC), a concept introduced by Shalizi (2003); Shalizi et al. (2004). Here complexity is related to the uncertainty of future predictive distributions of a system at a given point in time  $t$ .<sup>4</sup> The authors lay out their method in detail for finite discrete valued systems (e.g. a binary field as in Fig. 1) and present empirical results based on simulations of CA. Figure 2 shows the application of their methods with slight adaptations regarding the classification of predictive distributions (for details see Section 6). LSC captures exactly what we would expect as it shows us the most interesting - in the sense of informative for the instant future - parts of the image (see row iv) and v) of each sub-figure).

It is often important to get a good prediction of a dynamical system, for example for optimal decision planning. Let  $X(\mathbf{r}, t)$  be an  $(N + 1)$  dimensional field (deterministic or stochastic) and the interactions in this system propagate with velocity  $v \geq 0$ .<sup>5</sup> We could just consider  $X(\mathbf{r}, t)$  as a very high-dimensional time series, where all space points are arranged in a vector time series. This might work for small systems, but for real-world applications this becomes intractable very quickly. For example, a  $800 \times 600$  video requires an approximately 500,000 dimensional time series model. If we model it jointly as a multivariate process then already estimating a simple vector autoregressive (VAR) process of order 1 is impossible - since the lag 1 matrix has size  $500,000 \times 500,000$  - let alone more realistic models that can actually capture the dynamics of the observed system. If we model the time series individually, parameter estimation is in general not a problem, but we will miss all the interrelations between space-time points. It is therefore necessary to make a compromise between global and individual modeling, in order to balance the trade-off between statistical/physical accuracy and statistical/computational complexity.

**Definition 2.1** (Past and future light cones). *The past light cone (PLC) of  $(\mathbf{r}, t)$  are all space-time points that can influence  $(\mathbf{r}, t)$  (which depends on the velocity  $v$ ), i.e. all space-time points  $(\mathbf{q}, s)$ ,  $s < t$  such that  $\|\mathbf{q} - \mathbf{r}\| \leq v(t - s)$  where  $\|\cdot\|$  is an appropriate vector norm (e.g.  $\mathbb{L}^2$  for Euclidean space). The future light cone (FLC) of  $(\mathbf{r}, t)$  is defined equivalently as all points that can be influenced by what happens at  $(\mathbf{r}, t)$ , i.e. all points  $(\mathbf{q}, u)$ ,  $u > t$  such that  $\|\mathbf{r} - \mathbf{q}\| < v(u - t)$ .*

<sup>4</sup>Another approach is *local sensitivity analysis*, which infers structure by perturbing the system at each point and measuring how much the system changes in the future by this perturbation. If the change is large, then this point is important; otherwise it's not (see Shalizi, Haslinger, Rouquier, Klinkner, and Moore, 2006, for details). Appendix shows how local statistical complexity and the methods derived here can be used in practice to approximate local sensitivity.

<sup>5</sup>Velocity could also be a function of space and time; for simplicity assume  $v$  is constant. Even if in real world applications  $v$  varies over space-time, then set  $v = \sup_{\mathbf{r}, t} v(\mathbf{r}, t)$  to be on the safe side. Such an upper bound  $v$  leads to efficiency loss in computation (more data points than necessary have to be processed), but at least we do not miss important influences of the past on the future.

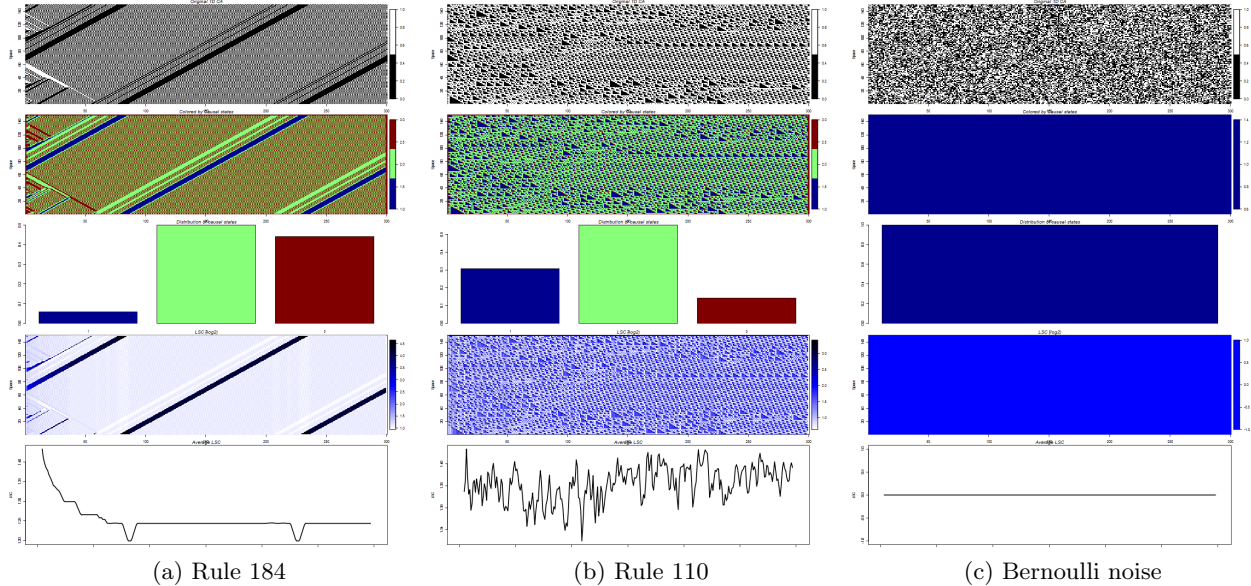


Figure 2: Local statistical complexity (LSC) applied to binary RFs of Fig. 1. In each subfigure (a)-(c), from top to bottom: i) original field  $X(\mathbf{r}, t)$ ; ii) each pixel colored by the causal state it belongs to; iii) distribution over causal states; iv) estimated local statistical complexity  $\widehat{C}(\mathbf{r}, t)$ ; v) average complexity  $\overline{\widehat{C}}(t) = \int_{\mathbf{S}} \widehat{C}(\mathbf{r}, t) d\mathbf{r}$ . Color versions of these figures are shown in the pdf.

Light cones are a good compromise between capturing the global patterns of a system, while at the same time only incorporating local structure.<sup>6</sup> See Fig. 3 for an illustration in the  $(1+1)D$  setting.

Let  $\ell^-(\mathbf{r}, t)$  be the configuration of the system  $X(\mathbf{r}, t)$  in the PLC; analogously let  $\ell^+(\mathbf{r}, t)$  denote the FLC configuration. Both  $\ell^+(\mathbf{r}, t)$  and  $\ell^-(\mathbf{r}, t)$  are collections of  $n_f$  and  $n_p$  points, which are the values of  $X(\mathbf{r}, t)$  in the future or past light cone, respectively. In general, light cones are space-time objects; for statistical/computational purposes, however, we can think of a light cone as an  $n_f$  (or  $n_p$ ) -dimensional vector which contains the values of the configuration in a certain way (natural space-time ordering stacked together into a vector). Exactly these local summaries of the system lie at the heart of finding interesting patterns.

For successful prediction of the future  $\ell^+(\mathbf{r}, t)$  given the past configuration  $\ell^-(\mathbf{r}, t)$  it is necessary to characterize (and in practice estimate) the conditional distribution  $\mathbb{P}(\ell^+(\mathbf{r}, t) | \ell^-(\mathbf{r}, t))$ .<sup>7</sup> Using the entire collection of past configurations for prediction is very likely unnecessarily complex and also becomes computationally intractable very fast. Thus we try to find a function  $\eta(\ell^-)$  of past configurations that keeps all relevant predictive information

$$\mathbb{P}(\ell^+ | \eta(\ell^-)) = \mathbb{P}(\ell^+ | \ell^-). \quad (1)$$

<sup>6</sup>One might argue that given the spatial dependence of random fields it would be more appropriate to consider light cones of entire patches of present pixels and not just one pixel. While this is a valid alternative and defining light cones for patches is analogous to the one pixel case, one reason why we consider one pixel only is that the predictive state of the patches is always a function of the predictive state of each pixel in the patch (Shalizi, 2003). However, for computational or statistical purposes it might be beneficial to consider entire patches.

<sup>7</sup>Subsequently the space-time arguments  $(\mathbf{r}, t)$  are omitted for better readability; however it should be kept in mind that all statements about  $\ell^-$  and  $\ell^+$  (and functions of them) in general depend on the particular  $(\mathbf{r}, t) \in \mathbf{S} \times \mathbb{T}$ .

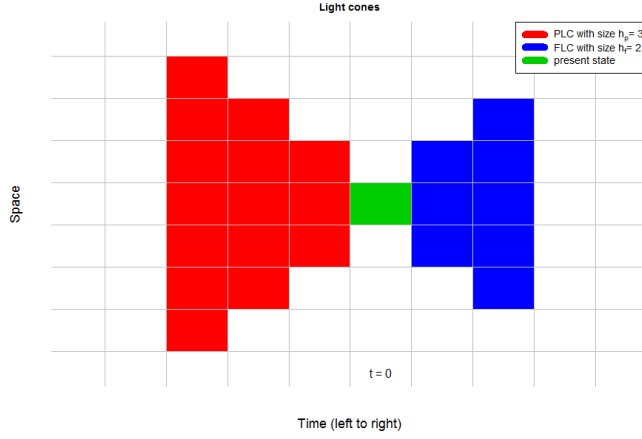


Figure 3: PLC (red) and FLC (blue) for  $(1 + 1)D$  space-time systems. Where to put the present  $X(\mathbf{r}, 0)$  (green) is a matter of convention. In Definition 2.1 it is part of past (red). In applied settings it can be beneficial to consider the present as part of the FLC; in particular, for testing univariate conditional FLC distributions, as in Section 5, I consider the present (green) as the one-dimensional future to predict from the past.

In statistics such a summary of data is known as a *sufficient statistics* (see Lehmann and Casella, 1998). In general, there are various sufficient statistics  $\eta_1(\ell^-), \dots, \eta_k(\ell^-)$ , each one with a generally different degree of data summary.

It is natural in terms of memory and data storage that we want to use that statistic  $\epsilon(\ell^-)$  which summarizes the data as much as possible without losing predictive power, i.e. it should not tell us more than absolutely necessary to get optimal predictions.

**Definition 2.2** (Minimal sufficient statistics). *A statistic  $\epsilon(\ell^-)$  is minimal sufficient if it can be computed from any other sufficient statistics  $\eta(\ell^-)$ . See Lehmann and Casella (1998, p. 37).*

## 2.1 Causal States

Shalizi et al. (2006) construct such a minimal sufficient statistics for the predictive distributions based on a particular collection of PLC configurations.

**Definition 2.3** (Equivalent PLC configurations). *Two PLCs  $\ell_1^-$  and  $\ell_2^-$  are equivalent,  $\ell_1^- \sim \ell_2^-$ , if they predict the same future with equal probabilities, i.e. if*

$$\mathbb{P}(\ell^+(\mathbf{r}, t) \mid \ell_1^-(\mathbf{r}, t)) = \mathbb{P}(\ell^+(\mathbf{r}, t) \mid \ell_2^-(\mathbf{r}, t)) \quad (2)$$

for all  $(\mathbf{r}, t) \in \mathbf{S} \times \mathbb{T}$ .

If  $X(\mathbf{r}, t)$  is not stationary over time, then the search for PLCs  $\lambda$  which are equivalent to  $\ell_1^-(\mathbf{r}_0, t_0)$  can be restricted to PLCs at time  $t_0$ . If the field is stationary, then it might be useful to look at all PLCs in the entire space-time. From a computational perspective, however, this increase makes the search for similar PLCs in  $\mathbb{R}^{n_p}$  much more time-consuming ( $|\mathbf{S}|$  versus  $|\mathbb{T}| \cdot |\mathbf{S}|$ , where  $|A|$  is the cardinality of  $A$ ). The results for discrete CA are based on equivalent PLCs for all  $t$ , for continuous fields I restrict the search to a fixed  $t$  (unless stated otherwise).

Let

$$[\ell^-] = \{\lambda \mid \lambda \sim \ell^-\} = \{\lambda \mid \mathbb{P}(\ell^+ \mid \ell^-) = \mathbb{P}(\ell^+ \mid \lambda)\} \quad (3)$$

be the equivalence class of  $\ell^-$ , i.e. all PLCs  $\lambda$  that predict the same future as  $\ell^-$ , and let  $\epsilon(\ell^-) : \ell^- \mapsto [\ell^-]$  be the function that maps each  $\ell^-$  to its equivalence class  $[\ell^-]$ . The values  $\epsilon$  can take are so-called *causal states* (Shalizi, 2003; Shalizi et al., 2006); they are the coarsest set of predictive sufficient statistics. Due to this functional relation, it is possible to identify  $\epsilon(\ell^-)$  with the equivalence class  $[\ell^-]$ ,  $\epsilon(\ell^-) \equiv [\ell^-]$ . Thus below  $\epsilon(\ell^-)$  can refer to both the function as well as the equivalence class, depending on the context.

For any sufficient statistics  $\eta$ ,  $\mathbb{P}(\ell^+ | \ell^-) = \mathbb{P}(\ell^+ | \eta(\ell^-))$ . Thus if  $\eta(\ell_1^-) = \eta(\ell_2^-)$ , then the two PLC configurations belong to the same causal state. Since we can compute  $\epsilon(\ell^-)$  from any other  $\eta(\ell^-)$ ,  $\epsilon(\ell^-)$  is indeed a minimal sufficient statistics by Definition 2.2.

How minimal is it? It can be encoded in  $\mathcal{I}(\epsilon(\ell^-); \ell^-)$  bits<sup>8</sup>, which is an objective measure as it is only based on  $X(\mathbf{r}, t)$ , not on any particular choice of model. Thus let

$$\mathcal{C}(\mathbf{r}, t) := \mathcal{I}(\epsilon(\ell^-(\mathbf{r}, t)); \ell^-(\mathbf{r}, t)) \quad (4)$$

be the *local statistical complexity* (LSC) of  $X(\mathbf{r}, t)$  at  $(\mathbf{r}, t)$ . It is measured in *bits* and equals the  $\log_2$  effective number of causal states, which correspond to different predictive distributions. Omitting the space-time index I (will usually) write  $\mathcal{C} := \mathcal{I}(\epsilon(\ell^-); \ell^-)$ .

Complexity lies between order and disorder: completely ordered  $X(\mathbf{r}, t)$  (constant over space and time) as well as completely disordered  $X(\mathbf{r}, t)$  (independent RVs) give  $\mathcal{C} = 0$ , otherwise  $\mathcal{C}$  is a positive value between these two extremes.

For discrete fields,  $\mathcal{C} = \mathcal{H}(\epsilon(\ell^-))$ , since

$$\mathcal{I}(\epsilon(\ell^-); \ell^-) = \mathcal{H}(\epsilon(\ell^-)) - \underbrace{\mathcal{H}(\epsilon(\ell^-) | \ell^-)}_{=0} = \mathcal{H}(\epsilon(\ell^-)), \quad (5)$$

where the second equality holds as  $\epsilon$  is a function of  $\ell^-$ , thus given  $\ell^-$  no uncertainty remains about  $\epsilon(\ell^-)$  anymore. This holds only for discrete fields; for continuous fields  $\mathcal{H}(\epsilon(\ell^-) | \ell^-) = 0$  need not hold, since differential entropy<sup>9</sup> does not necessarily equal zero. However, as  $\mathcal{I}(X; Y) = \mathcal{D}_{KL}(p(X, Y) || p(Y) \times p(X))$ , we can estimate  $\mathcal{I}(\epsilon(\ell^-); \ell^-)$  for continuous fields by using appropriate KL divergence estimators (Perez-Cruz, 2008).

In the discrete case (7) can be rewritten to

$$\bar{\mathcal{C}}(t_0) = \frac{1}{|\mathbf{S}|} \sum_{\mathbf{r} \in \mathbf{S}} \mathcal{C}(\mathbf{r}, t_0) = -\frac{1}{|\mathbf{S}|} \sum_{i=1}^n |\epsilon_i| \cdot \log_2 p_i = -\sum_{i=1}^n \frac{|\epsilon_i|}{|\mathbf{S}|} \log_2 p_i = -\sum_{i=1}^n p_i \log_2 p_i = \mathcal{H}(\epsilon). \quad (6)$$

Thus average statistical complexity equals the entropy of the causal state RV at  $t_0$ . If only one causal state exists, then there is no complexity:  $\mathcal{H}(p_1 = 1) = 0$ ; if every point of space constitutes a different causal state, in the sense that each one has a different conditional predictive distribution ( $p_i = \frac{1}{|\mathbf{S}|}$ ,  $i = 1, \dots, |\mathbf{S}|$ ), then this is the most complex possible setting at  $X(\mathbf{r}, t_0)$ , which reflects in the maximality of entropy for the uniform distribution.

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

$$\bar{\mathcal{C}}(t) := \int_{\mathbf{S}} \mathcal{C}(\mathbf{r}, t) d\mathbf{r} \quad (7)$$

<sup>8</sup>See Appendix A for definitions and details of information theory.

<sup>9</sup>See Definition A.7.

gives a one-dimensional summary of how interesting the field  $X(\mathbf{r}, t)$  is at time  $t$ . The lowest row in each subfigure of Fig. 2 shows the trajectory  $\bar{\mathcal{C}}(t)$ . A system has organized between  $t_1$  and  $t_2$  if complexity has increased, i.e.  $\bar{\mathcal{C}}(t_1) - \bar{\mathcal{C}}(t_2) =: \Delta\bar{\mathcal{C}} > 0$  (Shalizi et al., 2004).

## 2.2 Estimating causal states from data

In practice, it is necessary to estimate  $\mathcal{C}(\mathbf{r}, t)$  and  $\bar{\mathcal{C}}(t)$  from data  $x(\mathbf{r}, t)$ , to show us informative parts of an image or to give a one-dimensional time series  $\{\bar{\mathcal{C}}(t_1), \dots, \bar{\mathcal{C}}(t_n)\}$  indicating when something interesting is happening in a video.

Several parts play an important role in estimating  $\mathcal{C}(\mathbf{r}, t)$ :

0. Set velocity  $v$  and light cone horizon  $h_f$  and  $h_p$  to initial values. An optimal choice for these parameters can be obtained by cross validation - see Section 6.2.4.

The following steps must be repeated for each  $t_j \in \{t_1, \dots, t_T\}$ :

1. Iterate over all  $\mathbf{r}$  of  $(\mathbf{r}, t_j)$  and put all PLCs of the data in a list  $\{\ell_i^-\}_{i=1}^{|\mathbf{S}|}$ .
2. estimate conditional predictive distributions  $\mathbb{P}(\ell^+ | \ell_i^-)$  for each  $\ell_i^-$  in  $X(\mathbf{r}, t_j)$
3. cluster PLCs in equivalence classes according to equal conditional predictive distributions  $\rightarrow$  these are the causal states  $\epsilon(\ell^-)$  - see Section 3 and 6.2.2.
4. compute  $\mathcal{C}(\mathbf{r}, t_j)$  and  $\bar{\mathcal{C}}(t_j)$ .

For details about the algorithm, implementations, etc. see Section 6.

## 3 Comparing Probability Distributions

To characterize the causal states it is necessary to compare a collection of conditional RV  $\{\ell^+ | \ell_i^-\}_{i=1}^{|\mathbf{S}|}$ , which “live” in the configuration space  $\mathbb{R}^{n_f}$ . For example, for a  $(2+1)D$  field and a future horizon  $h_f = 2$  with velocity  $v = 1$ , there are  $n_f = 9 + 25 = 34$  points in the configuration space; the general formula for  $(2+1)D$  with  $v = 1$  is  $n_f = \frac{h_f(2h_f-1)(2h_f+1)}{3} - 1$ .

The causal states can be constructed by i) measuring the distance/similarity between the multivariate probability distributions  $\mathbb{P}(\ell^+ | \ell_i^-)$  and then use a spectral/hierarchical clustering algorithm based on the pairwise distance/similarity matrix, or ii) explicitly test the null hypothesis  $H_0 : \mathbb{P}(\ell^+ | \ell_i^-) = \mathbb{P}(\ell^+ | \ell_j^-)$  for all pairwise  $i$  and  $j$ .

A popular distance measure is the Kullback-Leibler (KL) divergence (Cover and Thomas, 1991), which itself is a particular case of the larger class of Ali-Silvey distance measures (Ali and Silvey, 1966) - see Appendix A.1 for details.

### 3.1 Testing equality of distributions

Here I group PLCs into causal states by directly testing for equality of distributions, which reduces to the two-sample problem in high dimensions. Although many non-parametric tests for  $H_0 : F = G$  based on two samples  $\mathbf{X}_n := \{X_1, \dots, X_n\} \sim F$  and  $\mathbf{Y}_m := \{Y_1, \dots, Y_m\} \sim G$  exist, they suffer from at least one of the following deficiencies:

**Unknown null:** Many tests rely on a comparison between the inner- and cross (Euclidean) distances between the  $\mathbf{X}_n$  and  $\mathbf{Y}_m$ , for example [Li, Maasoumi, and Racine \(2009\)](#); [Rizzo and Székely \(2010\)](#). Although standard distance measures can be computed very fast, critical values and p-values for these tests rely on permutation or bootstrapping, because the test statistic distribution under the null is unknown or only holds for large  $n$ .

**Computationally complex:** [Rosenbaum \(2005\)](#) introduces a test statistic, which relies on the number of matching pairs in a graph between two samples. Although the test is distribution free, i.e. under the null it has a standard distribution independent of the underlying unknown distribution  $F = G$ , it can only be computed by a bipartite matching algorithm which scales  $\mathcal{O}(n^3)$  in computational complexity.

Since one has to test all pairwise combinations of  $|\mathbf{S}|$  PLCs, both type of tests are not feasible for the light cone setting due to the quadratic increase in the number of tests. For example, for the  $200 \times 300$  fields in [Fig. 2](#), at each  $t = t_i$  one has to test  $\frac{200 \cdot 199}{2} - 200 = 19,700$  pairs. It is practically not feasible to do a permutation test with, say, 1,000 permutations each time.

However, one can use these 19,700 test-statistics to estimate the null distribution, since some proportion of them ( $\varepsilon > 0$ ) are presumably equal conditional FLC distributions, and others not ( $1 - \varepsilon$ ) ([Efron, 2004](#); [Jin and Cai, 2007](#)).

### 3.1.1 Estimating the null distribution

The non-parametric tests using pairwise Euclidean distances, presumably have some scaled (unknown)  $\chi^2$  distribution under the null hypothesis. Thus fitting the density of a  $\Gamma(\alpha, \beta)$  distribution to the small values of the test-statistics should give a good estimate of the null distribution (for details see [Schwartzman, Dougherty, Lee, Ghahremani, and Taylor, 2009](#)). The critical value for rejection of the null can then be calculated by the  $1 - \alpha$  quantile of the estimated  $\Gamma$  distribution.

## 4 Applications

Since LSC is a fully automated pattern recognition technique for any type of dynamical systems, applications to real world data are vast. [Shalizi et al. \(2006\)](#) show empirical results simulated 1D and 2D discrete CA; [Jänicke \(2009\)](#); [Jänicke and Scheuermann \(2010\)](#) present results on climate and flow data, with emphasis on visual representation. Below I list just a few of the many possible applications.

**Vibrothermography:** Vibrothermography is a recent non-destructive testing technique to detect tiny cracks in material such as turbine engine blades or steel bars ([Ibarra-castanedo, Susa, Klein, Grenier, Piau, Larby, Bendada, and Maldague, 2008](#); [Renshaw, Holland, and Thompson, 2008](#)).

It can be best explained by an example: suppose we have to test if blades of a plane turbine are safe to fly with. Due to the extreme forces exhibited on a blade during the flight, already tiny - invisible to the human eye - cracks can lead to disastrous consequences. The underlying idea of vibrothermography is that if we expose the material to ultrasonic waves these waves will result in friction - and thus to higher temperature - in the local neighborhood of a crack. Thus filming the blade during exposition to ultrasonic waves with an infra-red camera cracks should show up as high temperature (typically red) regions compared to the rest of the object (blue regions). The higher the wave power, the higher the temperature. But of course we should not increase the power until these red regions become clearly obvious, since this can intensify cracks and thus may destroy

turbine blades completely. On the other hand, if the ultrasonic waves are too weak then we might miss cracks.

This situation is a good example for a pattern recognition problem where the shape of the pattern is not known beforehand; a crack could have any shape.<sup>10</sup>

In the setting of this work we can view such a video as a space-time system in  $(2 + 1)$ D. If there is a crack, then it will be a good predictor for the future local behavior of the system. Thus finding informative local predictors is equivalent to finding cracks in the blade. Since LSC  $\mathcal{C}(\mathbf{r}, t)$  is based on minimal sufficient statistics we can “hope” that it suffices to use as little wave power as possible to detect the same amount of cracks as any competing method. Empirical comparisons with real-world data will show.

**Heart muscle contractions:** The heart is pumping blood by continuously contracting and relaxing. In case of a heart disease, it is important to understand what parts of the heart muscle are not working properly so physicians can recommend better medical treatments which specifically focus on these malfunctioning parts. A microscopic inspection of heart muscle cells shows that the dynamics of contraction  $\leftrightarrow$  relaxation is mainly driven by spirals that start from local centers and expand until they hit domain walls (from other expanding spirals), then relax and start over again. For an efficient medical treatment it would be important to know if a malfunctioning of this system comes from anomalies in the spiral centers or in the domain walls or both.

Again, local statistical complexity estimates from these microscopic dynamics should detect the location of anomalies in non-healthy heart muscles.

## 5 Continuous-valued systems: Gaussian random fields

As most relevant real-world applications are continuous-valued, it is necessary to extend theory and algorithms in [Shalizi et al. \(2006\)](#) to the continuous case. Although the principles of LSC do not change, statistical properties and implementation of LSC become substantially more difficult.

Contrary to the discrete case, identical light cones are probability zero events in continuous fields. To get an estimate of the predictive distribution it is not possible to look at the empirical distribution over FLCs conditioned on the specific PLC  $\ell_i^-$ , since this sample will only consist of one FLC for each PLC  $\ell_i^-$ . One way to obtain reasonable estimates of  $\mathbb{P}(\ell^+ | \ell_i^-)$  is as follows:

1. determine local neighbors of each  $\ell_i^-$ :  $N_i(\delta) := \{\ell_j^- | \|\ell_i^- - \ell_j^-\| < \delta\}$ , where  $\|\cdot\|$  is a proper norm on  $\mathbb{R}^{n_p}$ . If we choose  $\delta$  small enough, all PLCs  $\ell_j^- \in N_i(\delta)$  will belong to the same causal state  $\epsilon(\ell_i^-)$  to begin with;
2. estimate  $\mathbb{P}(\ell^+ | \ell_i^-)$  for each  $\ell_i^-$  using the sample  $\{\ell^+ | \ell_j^-\}_{j \in N_i(\delta)}$ .

Determining the causal states from the estimated distributions can be done in various ways: either by clustering in the distribution space (see [Section 3](#)), or by sequential testing of equality of distributions as in [Shalizi et al. \(2006\)](#). The results shown below use the latter method, where the future horizon  $h_f = 1$ . In this case conditional FLC distributions are univariate, and a two-sample Kolmogorov-Smirnov (KS) test can be used to decide whether  $H_0 : \mathbb{P}(\ell^+ | \ell_i^-) = \mathbb{P}(\ell^+ | \ell_j^-)$  or not. The significance level was set to  $\alpha = 10^{-2}$ .

---

<sup>10</sup>Often cracks are linearly shaped strings pointing in a certain direction, which can be detected by template matching from prior knowledge given previous videos ([Gao and Meeker, 2010](#); [Li, Holland, and Meeker, 2010](#)). However, since we cannot rule out the possibility of having non-linear shapes, such a template matching method would miss oddly shaped cracks.

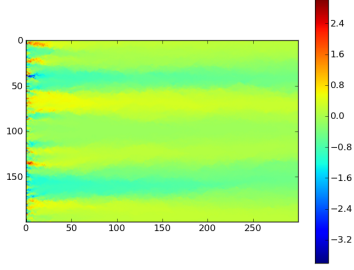
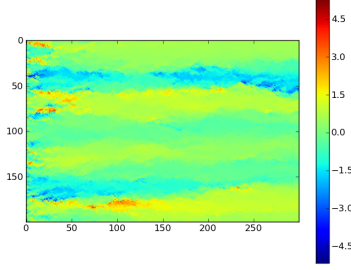
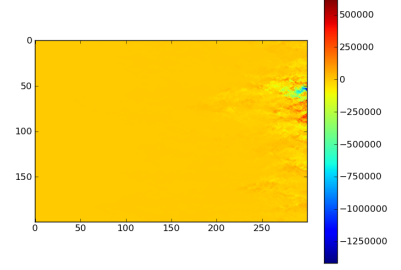
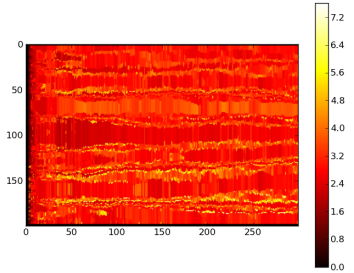
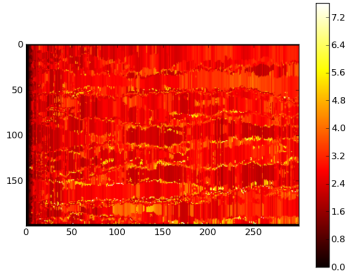
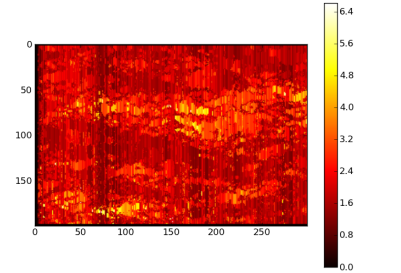
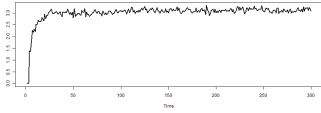
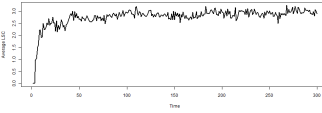
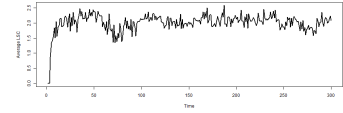
(a) Simulated  $x(\mathbf{r}, t)$  for  $\alpha = 0.5$ (b) Simulated  $x(\mathbf{r}, t)$  for  $\alpha = 0.7$ (c) Simulated  $x(\mathbf{r}, t)$  for  $\alpha = 0.9$ (d) Estimated  $\mathcal{C}(\mathbf{r}, t)$ (e) Estimated  $\mathcal{C}(\mathbf{r}, t)$ (f) Estimated  $\mathcal{C}(\mathbf{r}, t)$ (g) Average complexity  $\bar{\mathcal{C}}(t)$ (h) Average complexity  $\bar{\mathcal{C}}(t)$ (i) Average complexity  $\bar{\mathcal{C}}(t)$ 

Figure 4: Gaussian RFs  $X(\mathbf{r}, t)$  and their LSC. (top row): original field  $X(\mathbf{r}, t)$ ,  $\mathbf{r} = 1, \dots, 200$ ,  $t = 1, \dots, 300$ ; (middle row) LSC by sequential testing using Kolmogorov-Smirnov test on conditional predictive FLC distributions ( $h_f = 1$ , thus univariate) given  $h_p = 2$  PLCs (8-dimensional space); (bottom row) average statistical complexity. (Online pdf version shows colored figures.)

## 5.1 Simulations

Consider a Gaussian RF  $X(\mathbf{r}, t)$ , governed by an AR - ARCH type behavior, where the current pixel  $X(\mathbf{r}, t)$  given its PLC ( $h_p = 2$ ) is normally distributed, with mean equal to the average of the immediate (three) past pixels, and standard deviation equals  $\alpha$  times the empirical standard deviation of the field two time steps into the past:

$$X(\mathbf{r}, t) = \frac{1}{3}(X(\mathbf{r}-\mathbf{1}, t-1) + X(\mathbf{r}, t-1) + X(\mathbf{r}+\mathbf{1}, t)) + \varepsilon(\mathbf{r}, t), \quad \varepsilon(\mathbf{r}, t) \sim \mathcal{N}(0, \sigma^2(\mathbf{r}, t)) \quad (8)$$

$$\sigma(r, t) = \alpha \cdot \hat{\sigma}(X(\mathbf{r}-\mathbf{2}, t-2), X(\mathbf{r}-\mathbf{1}, t-2), \dots, X(\mathbf{r}+\mathbf{2}, t-2)) \quad (9)$$

$$X(\mathbf{r}, 1) = \mathbf{X}_1 \stackrel{iid}{\sim} \mathcal{N}(0, 1), \quad X(\mathbf{r}, 2) = \mathbf{X}_2 \stackrel{iid}{\sim} \mathcal{N}(0, 1). \quad (10)$$

Since the initial conditions are zero-mean Gaussians, it can be seen by an iterative argument that  $\mathbb{E}X(\mathbf{r}, t) = 0$  for all  $\mathbf{r}$  and  $t$ . However,  $\mathbb{E}(X(\mathbf{r}, t) | \ell^-) \neq 0$ .

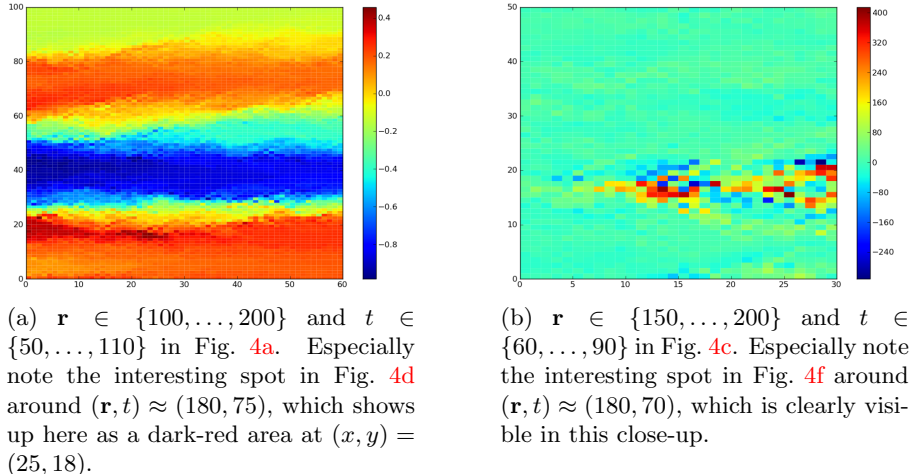


Figure 5: Zoom into “interesting” areas identified by LSC.

The key parameter here is  $\alpha$  which governs the possible dynamics of the field  $X(\mathbf{r}, t)$ :

$\alpha \ll 0.65$ : The future variance iteratively decreases, and so does the empirical standard deviation. For small  $\alpha$ , the variance of  $X(\mathbf{r}, t)$  decays to a constant value for large  $t$ .

$\alpha \approx \in (0.65, 0.75)$ : Here  $X(\mathbf{r}, t)$  shows interesting behavior, with locally emerging structure.

$\alpha \gg 0.75$ : The field shows explosive behavior in the variance; values of  $X(\mathbf{r}, t)$  drift of to  $\infty$  or  $-\infty$ , although locally they are conditionally stationary.

Here I show results of LSC applied to the simulated Gaussian RF, using the adaptations I made to the continuous case described above. Using a PLC horizon of  $h_p = 2$  (8 dimensional configuration space) and a FLC horizon of  $h_f = 1$  (just one pixel  $\rightarrow$  univariate conditional predictive distribution) gives the estimates  $\hat{\mathcal{C}}(\mathbf{r}, t)$  and  $\bar{\hat{\mathcal{C}}}(t)$  shown in the second and third row of Fig. 4. While for the original data, only the  $\alpha = 0.7$  field seems to show interesting patterns, LSC reveals important structure in all three fields. This highlights another practical by-result of LSC: image representations can suffer from bad color scaling due to outliers in the image, which is misleading to our eye as it suggests that there is nothing happening in the right part of  $\alpha = 0.5$  field and left part of  $\alpha = 0.9$  field, for example. However, LSC shows that in both cases the system develops non-trivial dynamics.

Figure 4d suggests that for  $\alpha = 0.5$  something interesting is happening at  $(\mathbf{r}, t) \approx (180, 75)$  (lower left), since  $\mathcal{C}(180, 75) \gg 0$ . Zooming into this area (Fig. 5a) confirms the local LSC increase, showing sudden increases and drops in the field, as well as a curved shape yellow area which appears and disappears at the lower part of the image.

As another example, say we only have one chance to zoom in to one area of Fig. 4c to look for interesting behavior. The complexity map in Fig. 4f reveals many interesting areas, for example  $(\mathbf{r}, t) \approx (180, 70)$ . Again, a zoom into this area validates the LSC findings (see Fig. 5b).

These examples show that the continuous extensions of LSC derived in this work already give promising results, and further theoretical as well as empirical/algorithmic analysis should lead to improvements. Building on these extensions, the next section provides a list of research directions that I propose for further study in the thesis to achieve these improvements.

## 6 Proposed Work

I aim to extend the work of [Shalizi \(2003\)](#); [Shalizi et al. \(2004\)](#) in the following ways:

### 6.1 Continuous valued fields

Provide a more detailed analysis on the methods and algorithms from the previous section, and develop more robust estimators.

A couple of important differences/challenges:

1. estimating (conditional) distributions in high dimension is not as easy as in the discrete case - see Section 6.2.1.
2. two-sample testing of  $H_0 : F = G$  versus  $H_1 : F \neq G$  given samples  $\mathbf{X}_n \sim F$  and  $\mathbf{Y}_m \sim G$  is substantially harder in the continuous case than in the discrete, especially for high-dimensional data (Section 3.1).

### 6.2 Statistical formulation and theory of causal states

Embed local statistical complexity into a more traditional statistical framework, in order to benefit from well-known statistical methods, such as:

#### 6.2.1 Non-parametric estimation and testing in high dimensions

For the identification of causal states it is necessary to compare the conditional predictive distribution of FLCs given PLCs,  $\mathbb{P}(\ell^+ | \ell^-)$ . For continuous valued fields non-parametric tests (Section 3.1) should be used to distinguish different causal states. [Jänicke and Scheuermann \(2010\)](#) circumvent this step by viewing continuous fields as very high-dimensional discrete fields. Results show that this works too, but discretizing the data can lead to different results for different binnings. Also formulating causal states for continuous RVs is a more appealing and challenging statistical problem.

#### 6.2.2 (Spectral) clustering in distribution space

In order to get causal states it is necessary to group predictive FLC distributions  $\{\mathbb{P}(\ell^+ | \ell_i^-)\}_{i=1}^{|\mathbf{S}|}$  into similar distributions. [Shalizi et al. \(2004\)](#) do this by putting all PLCs of  $X(\mathbf{r}, t_0)$  in a list  $\{\ell_j^-\}_{j=1}^{|\mathbf{S}|}$  of random order, iterate through the list, and assign a new light cone to a previously selected PLC as long as a  $\chi^2$  test cannot reject the null hypothesis of identical samples. Although this grouping gives useful results in the discrete case, it is not optimal for classifying distributions, e.g. the initial ordering of PLCs in the list can make a difference in the classification.

Hence, I want to find the causal states using standard clustering algorithms using well-defined (and computable) distance/similarity measure for multivariate, continuous distributions (see Section 3). For many clustering algorithms (e.g. spectral clustering and Laplacian graphs) KL divergence can be ruled out since they require a symmetric similarity measure ([von Luxburg, 2006](#)). For the analysis of the discrete CAs in Fig. 2 I used  $\mathcal{J}(p || q)$ . Since  $\mathcal{J}(p || q)$  gives good results (Fig. 2), there is no apparent reason why it should not work equally well for continuous fields. The only difference lies in estimating the KL divergence, but this can be done consistently also for high-dimensional continuous RVs ([Perez-Cruz, 2008](#); [Wang, Kulkarni, and Verd, 2006, 2009](#)).

I also want to see how resistor divergence  $\mathcal{R}(p || q)$  works in practice; especially because it seems convincing that this symmetrization is less ad-hoc and has nicer properties than  $\mathcal{J}(p || q)$ .

### 6.2.3 Prediction

So far predictive distributions have only been used to cluster the field into causal states and consequently measure the complexity of  $X(\mathbf{r}, t)$ . It is also very interesting to actually predict the state of the system at a future time  $t + h$ :  $X(\mathbf{r}, t + h)$ ,  $h > 0$ . Also spatial prediction might be useful (weather/climate data). There are at least two ways do this:

1. predict the configuration at  $X(\mathbf{r}, t + h)$  by the mode of the conditional predictive distribution of the particular causal state at  $(\mathbf{r}, t)$ .
2. predict FLCs of  $(\mathbf{r}, t)$  directly from the data without ever computing causal states. This can be done by a weighted average of *all* conditional FLCs up to time  $t - 1$ , where the weights are proportional to how similar  $\ell^-(\mathbf{r}, t)$  is to all other PLCs  $\ell^-(\mathbf{q}, \tau)$ ,  $\tau < t$ ,  $\mathbf{q} \in \mathbf{S}$ :

$$\mathbb{R}^{n_f} \ni \widehat{\ell}^+(\mathbf{r}, t) = \sum_{(\mathbf{q}, \tau)} s_{(\mathbf{q}, \tau), (\mathbf{r}, t)} \ell^+(\mathbf{q}, \tau) \in \mathbb{R}^n \text{ for all } \tau < t, \quad (11)$$

where  $s_{(\mathbf{q}, \tau), (\mathbf{r}, t)} = \text{sim}(\ell^-(\mathbf{r}, t), \ell^-(\mathbf{q}, \tau))$  is the similarity between the PLCs (in configuration space). For example, a Gaussian kernel  $\text{sim}(\ell^-(\mathbf{r}, t), \ell^-(\mathbf{q}, \tau)) = \exp(-\varepsilon \|\ell^-(\mathbf{r}, t) - \ell^-(\mathbf{q}, \tau)\|_2^2)$  with squared Euclidean distance.

In the second case it is not necessary to make a hard assignment/threshold of saying if PLCs are in the same causal state or not; we can use a continuous kernel to produce forecasts without ever clustering PLCs into causal states.

Since  $(\mathbf{q}, \tau)$  ranges over the whole space-time,  $\mathbf{q} \in \mathbf{S}$ ,  $\tau = 1, \dots, t - 1$ , the computation of this many similarities  $\{s_{(\mathbf{q}, \tau), (\mathbf{r}, t)}\}$  becomes very expensive. For example, a low resolution video with  $400 \times 300$  pixels, which runs for just 10 seconds with 25 frames per second (fps) gives  $N = 3 \cdot 10^7$  space-time points. Thus in practice, if the field has a high resolution, then computing the causal states is a necessary pre-step before doing predictions.

The predictors  $\ell^+(\mathbf{q}, \tau)$  on the right hand side of (11) are dependent RVs, thus typical prediction theory for linear predictors is not directly applicable anymore. However, if we view each past FLC  $\widehat{\ell}^+(\mathbf{q}, \tau)$ ,  $\tau < t$ , as an expert we can use convergence and optimality results considering prediction with expert advice that even apply to the setting of dependent experts - see [Cesa-Bianchi and Lugosi \(2006\)](#) for an extensive and detailed analysis.

A similar approach to 1. has been taken in [Parlitz and Merkwirth \(2000\)](#), and the predictions have very small error. However, the authors use past rectangles instead of cones, and even mention that light cones might be the better choice; secondly, they only group PLCs in the configuration space, not in the conditional future distribution space.

### 6.2.4 Cross-validation

LSC is an automated pattern recognition technique in the sense that the algorithm finds interesting features automatically, without outside user definition or pre-setting of what characteristics to look for. However, it has several nuisance parameters: a) the PLC and FLC horizon  $h_p$  and  $h_f$ ; b) the velocity  $v$ ; and c) a threshold when distributions are considered similar and when not (either by a significance level  $\alpha$  for two-sample testing, or by a bandwidth parameter  $\varepsilon$  for the (Gaussian) similarity kernel). Sometimes knowledge about the physical system at hand might give an idea about how to choose these values, but

in general it is not clear what the optimal parameters are for a given dataset.

Selecting the right light cone depth faces a bias  $\leftrightarrow$  variance trade-off: small light cones can capture fine structures (low bias), but results will change a lot for another observation of the same system (large variance). Light cone depth is also important for the conditional density estimation, because the larger the depth, the sparser the light cones in the configuration space - thus we need more data to estimate their predictive distribution accurately. Hence there is a practical limitation of not letting the light cone depth get too large, because then light cones will be spread out too much in the configuration space and we cannot get reasonable estimates of  $\mathbb{P}(\ell^+ | \ell^-)$  anymore.

Thus we have to choose the light cone depth such that the bias is not too large, but at the same time try to minimize the variation in the light cone distributions estimates. A standard way to choose optimal nuisance/bandwidth parameters is cross validation (CV). Since the data here is not iid, plain CV will not give consistent estimates.

Racine (2000) gives conditions on how to choose the training and test sets for dependent data to guarantee consistent CV for choosing the correct set of predictors in linear regression. Let  $Z = (Y, X)$  be a matrix of  $n$  observations for a response variable  $Y$  and  $p$  predictors  $X$ . Standard leave-K-out-CV does not work as expected because for dependent data we cannot just remove data points randomly, but have to remove them in blocks around the observation  $z_i$ ; e.g. on each side we remove  $h$  observations, thus in total a block of  $2h + 1$  data points. However, this still does not guarantee consistent model selection and therefore Racine (2000) introduces  $hv$ -block cross validation where one first removes  $2v + 1$  observations around  $z_i$  and then removes another “layer” of size  $h$  on each side of the  $2v + 1$  block. This removes in total  $2h + 2v + 1$  observations from the dataset. The parameter  $v$  controls the relationship between the training and test set;  $h$  controls how independent test and training set are: higher  $h$  makes them less dependent. If  $h$  and  $v$  grow with a certain rate of the sample size, this procedure is leads to consistent model selection for linear models. In Appendix B I describe an adaptation of  $hv$ -block CV to the light cone setting.

Results for  $hv$ -block CV applied to continuous Gaussian RF are shown in Fig. 6: here  $h_f = 1$  is held fixed, but the PLC horizon  $h_p$  and the bandwidth  $\varepsilon$  of the Gaussian kernel are varied. In both cases,  $hv$ -block cross validation picks the correct PLC size.

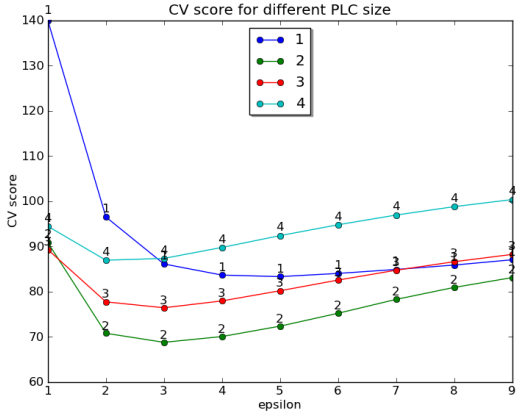
### 6.2.5 Hypothesis testing

For a dataset  $x(\mathbf{r}, t)$  we might be interested in hypothesis tests of (at least) three kinds:

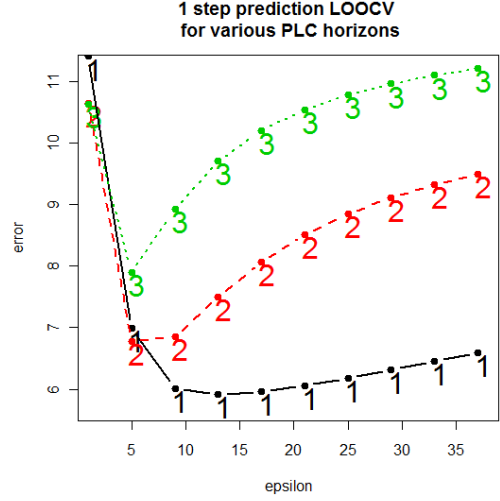
1.  $H_0 : \mathcal{C}(\mathbf{r}, t) = 0$  versus  $H_1 : \mathcal{C}(\mathbf{r}, t) > 0$ : this is a test to see if there is something interesting going on or not. In the vibrothermography context this would be a test of  $H_0$ : “blade contains no crack” versus  $H_1$ : “blade contains at least one crack”.

For the Bernoulli iid field in Fig. 2c such a test would not reject the null hypothesis, since  $\widehat{\mathcal{C}}(\mathbf{r}, t) \equiv 0$  for all  $(\mathbf{r}, t)$ .

2.  $H_0 : \Delta\overline{\mathcal{C}}(t) = 0$  versus  $H_1 : \Delta\overline{\mathcal{C}}(t) > 0$ : biologists are often interested to know if an eco-system (of bacteria, for example) has organized over time or not.
3.  $H_0 : \mathcal{C}(\mathbf{r}, t) = \mathcal{C}_0(\mathbf{r}, t)$  versus  $H_1 : \mathcal{C}(\mathbf{r}, t) \neq \mathcal{C}_0(\mathbf{r}, t)$ : this might be useful in cases where the theoretical complexity of a system can be computed analytically, and we want to test if the observed dynamics are actually a sample of this particular system, or if different dynamics play an important role. This could be of interest to physicists who can compute the (physical) entropy of a system. For



(a) LOOCV 3-step prediction for the  $\alpha = 0.7$  field  $X(\mathbf{r}, t)$  in Fig. 4b with true  $h_p = 2$ .



(b) LOOCV 1-step prediction for simulated field with true  $h_p = 1$ .

Figure 6: Cross-validation for dependent data using prediction (11): a grid search over four different PLC horizons  $h_p = 1, 2, 3, 4$  and several  $\varepsilon$ s in the Gaussian kernel  $\exp(-\varepsilon\|\ell_i^- - \ell_j^-\|_2^2)$ .

example [Shalizi et al. \(2006\)](#) compute the theoretical free energy of a discrete  $2D$  CA and compare it to  $\widehat{\mathcal{C}}(\mathbf{r}, t)$ : visually there are no differences.

In all the above cases it is necessary to have results on error bounds/asymptotic properties of the estimator  $\widehat{\mathcal{C}}(\mathbf{r}, t)$ .

### 6.3 Properties of the LSC estimator

Derive error bounds/asymptotic properties (bias/variance) of several LSC estimators. Under the null iid case the estimator for  $\overline{\mathcal{C}}(t)$  is related to the entropy of the spectrum of a random matrix, where the entries are, for example, the p-value of a test for equal predictive distributions. If the null is true, then these p-values are uniformly distributed in  $[0, 1]$ , and random matrix theory regarding the spectrum of symmetric, uniform matrices or, more general, of similarity/distance/adjacency matrices might be helpful, e.g. [Bogomolny, Bohigas, and Schmit \(2003\)](#); [Karoui \(2010\)](#); [Malarz \(2008\)](#).

### 6.4 Applications

Apply the methods to real world data (see Section 4), in particular  $2D$  fields (videos).

## References

- Ali, S. M. and S. D. Silvey (1966). A General Class of Coefficients of Divergence of One Distribution from Another. *Journal Of The Royal Statistical Society Series B* 28, 131–140.
- Ascher, D., P. F. Dubois, K. Hinsen, J. Hugunin, and T. Oliphant (1999). *Numerical Python* (UCRL-MA-128569 ed.). Livermore, CA: Lawrence Livermore National Laboratory.
- Bogomolny, E., O. Bohigas, and C. Schmit (2003, March). Spectral properties of distance matrices. *Journal of Physics A Mathematical General* 36, 3595–3616.
- Cesa-Bianchi, N. and G. Lugosi (2006, March). *Prediction, Learning, and Games*. Cambridge University Press.
- Cover, T. M. and J. Thomas (1991). *Elements of Information Theory*. Wiley.
- Cox, R. T. (2001, February). *Algebra of Probable Inference*. Johns Hopkins University Press.
- Efron, B. (2004). Large-Scale Simultaneous Hypothesis Testing: The Choice of a Null Hypothesis. *Journal of the American Statistical Association* 99(465), 96–104.
- Feldman, D. P. and J. P. Crutchfield (1997). Measures of statistical complexity: Why? Working papers, Santa Fe Institute.
- Gao, C. and W. Q. Meeker (2010). A statistical method for crack detection from vibrothermography inspection data.
- Hyvärinen, A. and E. Oja (2000). Independent component analysis: algorithms and applications. *Neural Networks* 13, 411–430.
- Ibarra-castanedo, C., M. Susa, M. Klein, M. Grenier, J.-M. Piau, W. B. Larby, A. Bendada, and X. Maldague (2008). Infrared thermography: principle and applications to aircraft materials.
- Jänicke, H. (2009). *Information Theoretic Methods for the Visual Analysis of Climate and 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.
- Jeffreys, H. (1946). An Invariant Form for the Prior Probability in Estimation Problems. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences* 186(1007), 453–461.
- Jin, J. and T. T. Cai (2007). Estimating the null and the proportion of non-null effects in large-scale multiple comparisons. *J. Amer. Statist. Assoc* 102, 496–506.
- Johnson, D. and S. Sinanovic (2001). Symmetrizing the Kullback-Leibler Distance. Technical report, IEEE Transactions on Information Theory.
- Jolliffe, I. T. (2002, October). *Principal Component Analysis* (Second ed.). Springer.
- Jones, E., T. Oliphant, P. Peterson, et al. (2001–). SciPy: Open source scientific tools for Python.

- Karoui, N. E. (2010). The spectrum of kernel random matrices. *Annals of Statistics* 38(1), 1–50.
- Lee, A. B. and L. Wasserman (2010). Spectral connectivity analysis. *Journal of the American Statistical Association* 105(491), 1241–1255.
- Lehmann, E. L. and G. Casella (1998, August). *Theory of Point Estimation (Springer Texts in Statistics)* (2nd ed.). Springer.
- Li, M., S. D. Holland, and W. Q. Meeker (2010). Automatic Crack Detection Algorithm For Vibrothermography Sequence-Of-Images Data. *AIP Conference Proceedings* 1211(1), 1919–1926.
- Li, Q., E. Maasoumi, and J. S. Racine (2009, February). A nonparametric test for equality of distributions with mixed categorical and continuous data. *Journal of Econometrics* 148(2), 186–200.
- Malarz, K. (2008, March). Spectral properties of adjacency and distance matrices for various networks. *ArXiv e-prints*.
- Parlitz, U. and C. Merkwirth (2000, Feb). Prediction of spatiotemporal time series based on reconstructed local states. *Phys. Rev. Lett.* 84(9), 1890–1893.
- Perez-Cruz, F. (2008). Kullback-Leibler Divergence Estimation of Continuous Distributions.
- 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.
- Racine, J. (2000). A Consistent Cross-Validatory Method For Dependent Data: hv-Block Cross-Validation. *Journal of Econometrics* 99, 39–61.
- Renshaw, J., S. D. Holland, and R. B. Thompson (2008, August). Measurement of crack opening stresses and crack closure stress profiles from heat generation in vibrating cracks. *Applied Physics Letters* 93(8), 081914–1 – 081914–3.
- Rizzo, M. L. and G. J. Székely (2010). DISCO analysis: A nonparametric extension of analysis of variance. *ArXiv e-prints* 4(2), 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(4), 515–530.
- Schiff, J. L. (2008). *Cellular Automata: A Discrete View of the World (Wiley Series in Discrete Mathematics & Optimization)*. Wiley-Interscience.
- Schwartzman, A., R. Dougherty, J. Lee, D. Ghahremani, and J. Taylor (2009, January). Empirical null and false discovery rate analysis in neuroimaging. *NeuroImage* 44(1), 71–82.
- Seghouane, A.-K. and S.-I. Amari (2007). The AIC Criterion and Symmetrizing the Kullback-Leibler Divergence. *Neural Networks, IEEE Transactions on* 18(1), 97–106.
- Shalizi, C. R. (2003). Optimal nonlinear prediction of random fields on networks. In *Discrete Mathematics and Theoretical Computer Science, AB(DMCS):1130*, pp. 11–30.
- 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. *Phys. Rev. E* 73(3), 036104.

- Shalizi, C. R., K. L. Shalizi, and R. Haslinger (2004, Sep). Quantifying self-organization with optimal predictors. *Phys. Rev. Lett.* *93*(11), 118701.
- Van Rossum, G. (2003, September). *The Python Language Reference Manual*. Network Theory Ltd.
- von Luxburg, U. (2006, August). A tutorial on spectral clustering. Technical Report 149, Max Planck Institute for Biological Cybernetics.
- Wang, Q., S. R. Kulkarni, and S. Verd (2006). A Nearest-Neighbor Approach to Estimating Divergence between Continuous Random Vectors. In *IEEE International Symposium on Information Theory*.
- Wang, Q., S. R. Kulkarni, and S. Verd (2009). Divergence estimation for multidimensional densities via k-nearest-neighbor distances. *IEEE Trans. Inform. Theory* *55*(5), 2392 – 2405.
- Wiskott, L. and T. J. Sejnowski (2002, April). Slow Feature Analysis: Unsupervised Learning of Invariances. *Neural computation* *14*(4), 715–770.
- Wolfram, S. (1983, Jul). Statistical mechanics of cellular automata. *Rev. Mod. Phys.* *55*(3), 601–644.

## A Information Theory

Let  $X$  be a discrete RV taking values in a finite alphabet  $\mathcal{A} = \{a_1, \dots, a_n\}$  with probability mass function (pmf)  $\mathbb{P}(X = a_k) = p_k$ . An interesting question is the following: how uncertain are we about the outcome of  $X$ ? Or equivalently, how informative would one particular outcome  $a_i$  of  $X$  be?

Information theory provides an answer to this question; in particular  $-\log_2 p_k$  measures the uncertainty of the event  $\{X = a_k\}$  - see [Cox \(2001\)](#) for an excellent description.

**Definition A.1** (Entropy). *The entropy of a discrete RV  $X$*

$$\mathcal{H}(X) = - \sum_{k=1}^n p_k \log_2 p_k = -\mathbb{E}_p \log_2 p \quad (12)$$

*measures the average uncertainty of  $X$ .*

The units of measurements of  $\mathcal{H}(X)$  are binary digits or *bits*, and can be interpreted as how many yes/no question one has to ask - on average - until knowing what value  $X$  has taken. Since  $\mathcal{H}(X)$  does not depend on the actual values  $X$  can take, but only on the probabilities  $\mathbb{P}(X = a_k)$ , it is common to write  $\mathcal{H}(p)$  for  $\mathcal{H}(X)$ .

For example, the entropy of a Bernoulli RV  $X \sim \text{Bern}(p)$  with  $\mathbb{P}(X = 1) = p$  (e.g. a coin spin with probability  $p$  of coming up heads) equals

$$\mathcal{H}(p) = - [p \log_2 p + (1 - p) \log_2 (1 - p)]. \quad (13)$$

Figure 7 shows (13) as a function of  $p$ . Again, it is intuitively clear that we are most unsure about the outcome of the coin spin if heads and tails are equally likely, i.e.  $p = 0.5$ ; there is no uncertainty if either heads or tails are probability zero events.

**Notation A.2** (Base 2 logarithm versus natural logarithm). *If instead of  $\log_2$  we had used the natural logarithm  $\ln$  in (12), the results would only change by a constant factor of  $\log_2 e$  and the unit of measurement would be natural units or nats. In computer science it is common to use  $\log_2$  whereas in statistics it is common to use  $\ln$ . Here I will always use  $\log_2$  for information theory related formulas and  $\ln$  for statistics formulas (e.g. for the log-likelihood function).*

Entropy in (12) has many useful properties; here I will only list the most important ones related to LSC. A more extensive list can be found in [Cover and Thomas \(1991, p. 41-42\)](#)

**Properties A.3** (Entropy). *For discrete probability distributions  $p = \{p_k\}_{k=1}^n$ :*

1.  $\mathcal{H}(p) \geq 0$  for all discrete  $p$ .
2.  $\mathcal{H}(p) = 0$  if and only if  $p_k = 1$  for one  $k$  and  $p_k = 0$  for all other  $k$ . Clearly there is no uncertainty if a “random” variable can only take one value.
3.  $\mathcal{H}(p)$  achieves a maximum for the uniform distribution  $p_k = \frac{1}{n}$  for all  $k = 1, \dots, n$ . This confirms our intuition that we are most uncertain about the outcome of an experiment which can take only finitely many values if all happen with equal probability.
4. it obeys the so called data processing inequality

$$\mathcal{H}(f(X)) \leq \mathcal{H}(X), \quad (14)$$

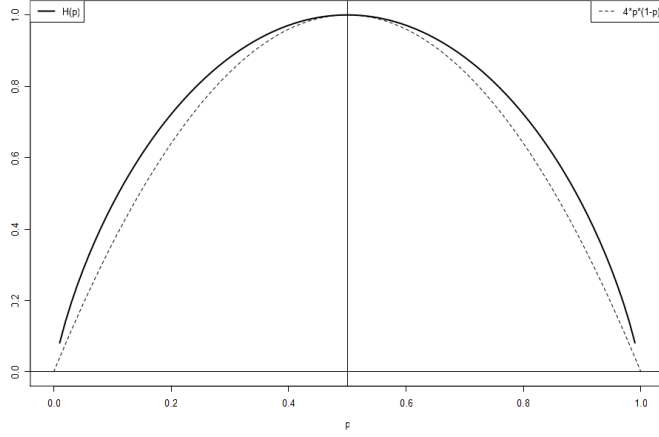


Figure 7: Entropy (solid) of a Bernoulli RV  $X$  with success probability  $\mathbb{P}(X = 1) = p \in [0, 1]$ ; for comparison  $f(p) = 4p(1 - p)$  (dashed).

with equality if and only if  $f$  is invertible. This means that transforming data cannot increase information.

**Definition A.4** (Joint Entropy). *The joint entropy  $\mathcal{H}(X, Y)$  of two RVs,  $X$  and  $Y$ , is the entropy of their joint distribution.*

**Definition A.5** (Conditional Entropy). *The conditional entropy of  $X$  given  $Y$ ,  $\mathcal{H}(X | Y)$ , equals*

$$\mathcal{H}(X | Y) = \sum_y \mathbb{P}(Y = y) \sum_x \mathbb{P}(X = x | Y = y) \log_2 \mathbb{P}(X = x | Y = y) \quad (15)$$

$$= -\mathbb{E}_{X, Y} \log_2 \mathbb{P}(X = x | Y = y) \quad (16)$$

$$= \mathcal{H}(X, Y) - \mathcal{H}(Y). \quad (17)$$

**Definition A.6** (Mutual Information). *Let  $X$  and  $Y$  be two RVs with probability distributions  $p(x)$  and  $q(y)$ . Then the information that variable  $Y$  conveys about  $X$  is defined as the reduction in uncertainty about  $X$  given an observation of  $Y$ , i.e.*

$$\mathcal{I}(X; Y) = \mathcal{H}(X) - \mathcal{H}(X | Y) \quad (18)$$

$$= \mathbb{E}_{X, Y} \log_2 \frac{P(X, Y)}{P(X)P(Y)} \quad (19)$$

$$= -\mathbb{E}_X \log P(X) - (-\mathbb{E}_{X, Y} \log P(X | Y)) \quad (20)$$

where  $P(X, Y)$  is the joint probability,  $P(X)$  &  $P(Y)$  are the marginals, and  $\mathbb{E}_Z$  denotes expectation with respect to the (possibly multivariate or conditional) RV  $Z$ .

Mutual information  $\mathcal{I}(X; Y) \geq 0$  is symmetric and equals zero if and only if  $X$  is independent of  $Y$ ,  $X \perp\!\!\!\perp Y$ .

**Definition A.7** (Differential Entropy). *Let  $X$  have probability density function  $f(x)$ . Then the differential entropy of  $X$  equals*

$$\mathcal{H}(X) = - \int f(x) \log_2 f(x) dx. \quad (21)$$

For a list of properties see [Cover and Thomas \(1991, p. 256\)](#)

## A.1 Kullback-Leibler divergence and variants

KL divergence is an information theoretic measure of the difference between two probability distributions  $p$  and  $q$ . For discrete distributions it is defined as

$$\mathcal{D}_{KL}(p \parallel q) := \sum_i p_i \log_2 \frac{p_i}{q_i} = \mathbb{E}_p \log_2 \frac{p}{q}, \quad (22)$$

where  $p_i = \mathbb{P}(X_i = a_i)$  and  $q_i = \mathbb{P}(Y_i = a_i)$ . As for entropy the units of measurement are bits.

For continuous (multivariate) RVs  $X \sim p(\mathbf{x})$  and  $Y \sim q(\mathbf{y})$

$$\mathcal{D}_{KL}(p \parallel q) := \int p_x(\mathbf{x}) \log_2 \frac{p_x(\mathbf{x})}{q_y(\mathbf{x})} d\mathbf{x} = \mathbb{E}_q \log \frac{p(\mathbf{X})}{q(\mathbf{X})} \quad (23)$$

While  $\mathcal{D}_{KL}(p \parallel q)$  is a measure for the difference of two distributions, it is not a metric because it a) does not satisfy the triangle inequality, and b) is asymmetric: in general  $\mathcal{D}_{KL}(p \parallel q) \neq \mathcal{D}_{KL}(q \parallel p)$ . Since many clustering algorithms require a symmetric distance/similarity measure, the original KL divergence cannot be used directly. There have been various attempts to symmetrize KL divergence ([Seghouane and Amari, 2007](#)); below I discuss two common choices.

### A.1.1 J-divergence

[Jeffreys \(1946\)](#) suggested the arithmetic mean

$$\mathcal{J}(p \parallel q) := \frac{\mathcal{D}_{KL}(p \parallel q) + \mathcal{D}_{KL}(q \parallel p)}{2} \quad (24)$$

as a symmetrized version of  $KL$  divergence, which is also known as J-divergence. But this is a rather heuristic than well-founded way to do it.

### A.1.2 Resistor divergence

[Johnson and Sinanovic \(2001\)](#) symmetrize KL divergence using the harmonic instead of the arithmetic mean. Resistor KL divergence  $\mathcal{R}(p \parallel q)$  between  $p$  and  $q$  is defined as

$$\frac{1}{\mathcal{R}(p \parallel q)} := \frac{1}{\mathcal{D}_{KL}(p \parallel q)} + \frac{1}{\mathcal{D}_{KL}(q \parallel p)}. \quad (25)$$

Resistor divergence has several advantages over J-divergence; in particular, it has “nice” geometric interpretations and relations to Chernoff bounds and Bhattacharyya distance (see [Johnson and Sinanovic, 2001](#), for details).

## B Cross Validation

It is not clear yet how important the assumption of a linear model is to obtain the consistency result for  $hv$ -block cross validation in [Racine \(2000\)](#). However, from a practical perspective it is straightforward to adapt the  $hv$ -block cross validation to random fields and the light cone setting:

1. view  $X(\mathbf{r}, t)$  as a high-dimensional time series  $Y_t$ , e.g. for a video stack the pixel values into a vector for each time point

2. partition the data in the  $h\nu$ -block way into  $K$  subsamples (e.g. slice out a couple of neighboring columns of an image for a 1D CA; or remove a short sequence of the video) and
3. choose a loss function  $\text{loss}(\cdot, \cdot)$ . For example, 0/1 loss for categorical fields (e.g. (few) colors in an image), or  $\mathbb{L}_p$  norm for continuous valued fields (e.g. a temperature field, or an image with millions of colors).
4. make a list of various PLC horizons  $h_1, \dots, h_p$ . For each  $h_i \in \{h_1, \dots, h_p\}$ :

- (a) Estimate predictive distributions based on the rest (of the image/video)

$$\mathbb{P}(\ell^+(\mathbf{r}, t) \mid \ell^-(\mathbf{r}, t))_{h_i} \quad (26)$$

where the sub-script  $h_i$  indicates that light cone depth  $h_i$  was used.

- (b) Construct predictors for  $X(\mathbf{r}, t)$  and predict the removed part  $X(\mathbf{r}, t)^{(k)}$  of the image (see [Parlitz and Merkwirth, 2000](#)). Here  $^{(k)}$  means that the  $k^{\text{th}}$  slice has been removed.
- (c) Compute the average prediction error of  $\hat{x}(\mathbf{r}, t)^{(k)}$  compared to the observed  $x(\mathbf{r}, t)^{(k)}$  and the estimated risk

$$\hat{R}_{CV}(h_i) = \frac{1}{K} \sum_{k=1}^K \text{loss} \left( x(\mathbf{r}, t), \hat{x}(\mathbf{r}, t)^{(k)} \right) \quad (27)$$

5. choose that  $h_i$  which gives the smallest empirical risk and estimate causal states again with this particular  $h_i^*$ , but now for the whole system with no observations removed.

Assuming that results in [Racine \(2000\)](#) also hold for more general models and reasonable predictors the above procedure should - except for some refinements - lead to consistent light cone size selection.

## C Algorithms and Implementation

Fast implementation of the methods is very important in order to actually use these automated pattern recognition techniques in practice. Optimally a program should take either a sequence of consecutive images or directly a video as input, and after not too long a time return estimates of  $\mathcal{C}(\mathbf{r}, t)$  and informative figures/data.

Parallel computing resources can improve computation time substantially, since there are many trivially parallelizable parts of the analysis:

- the LSC analysis is done for each fixed  $t$  separately and the necessary computations are independent of each other.
- for clustering in distribution space it is necessary to compute pairwise distances/similarities/tests for all PLCs at a given time  $t_i$ . Computing these (symmetric) similarity matrices  $\in \mathbb{R}^{|\mathbf{S}| \times |\mathbf{S}|}$  can also be done in parallel.

**Remark C.1.** *For fast development/coding and exploratory data analysis I use Python ([Van Rossum, 2003](#)). Since Python's main libraries for scientific computing (`numpy` ([Ascher, Dubois, Hinsén, Hugunin, and Oliphant, 1999](#)) and `scipy` ([Jones, Oliphant, Peterson, et al., 01](#))), and many additional packages are wrappers of standard C/C++ libraries, the speed loss should not be too large.*

*If necessary (parts of) the code can later be converted to native C/C++ code and then linked to Python.*

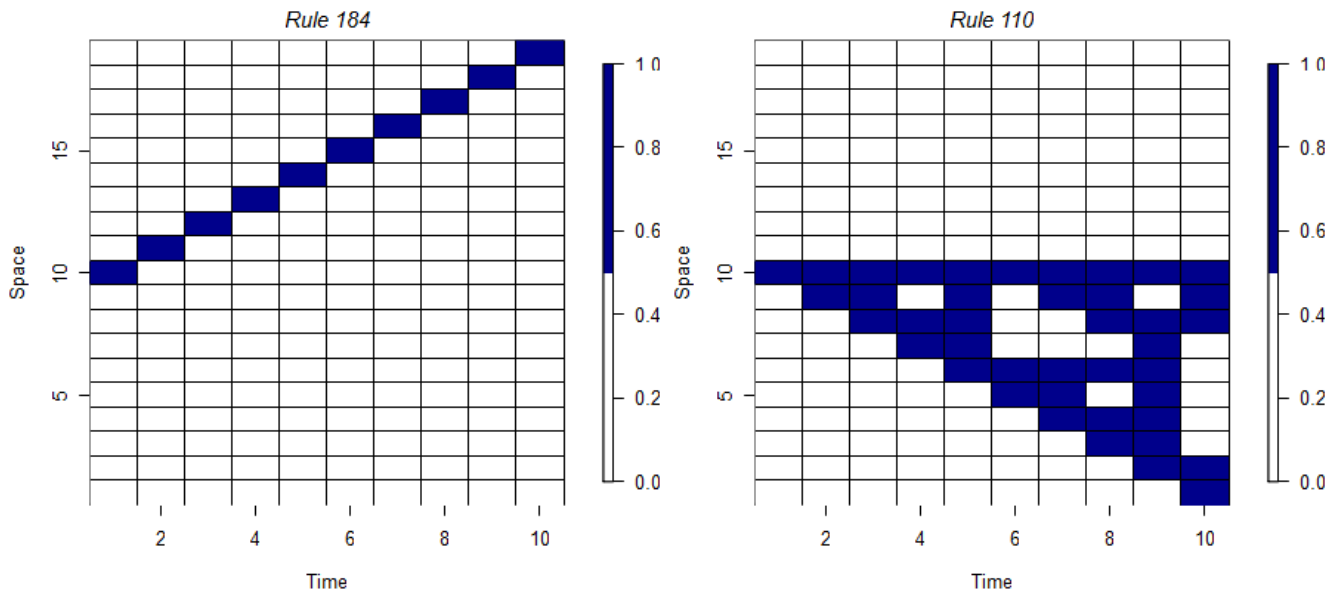


Figure 8: Propagation of binary cell structure in Cellular Automata, rule 184 and rule 110. Here the initial state  $X(\mathbf{r}, 1)$  consists of all zeros except the center pixel,  $X(\mathbf{10}, 1)$ , which equals 1. For dynamics on a large scale and for non-trivial initial configuration see Fig. 1.

## D Cellular Automata in Detail

As described in Section 1.1, Cellular Automata (CA) are dynamic space-time systems where each pixel updates itself according to a certain rule.

These simple rules can generate complex behavior on a larger scale, however, the large scale figures make it difficult to see how CAs actually work. Figure 8 shows a very high-resolution zoom to the patterns of the rules 110 and 184, which I use in the Introduction.

For simplicity both configurations start at the same initial condition  $X(\mathbf{r}, 1)$ , which has all but one pixel equal to zero (white fields). A “rule” determines the value of a pixel at time  $t$  given its three immediate past local neighbors. As 3 cells can take  $2^3 = 8$  different binary combinations, and each of the 8 configurations can lead to different (binary) future pixels, there are a total of  $2^8 = 256$  basic - so called elementary - rules.

For example, for both rules three adjacent white fields ( $x(\mathbf{r} - \mathbf{1}, t - 1), x(\mathbf{r}, t - 1), x(\mathbf{r} + \mathbf{1}, t - 1) = (0, 0, 0)$ ), also lead to a future white pixel  $r(\mathbf{t} + \mathbf{1}, =) 0$  (consider the lower half of rule 184 and the upper half of rule 110).

However, the configuration ( $x(\mathbf{r} - \mathbf{1}, t - 1), x(\mathbf{r}, t - 1), x(\mathbf{r} + \mathbf{1}, t - 1) = (0, 0, 1)$ ) leads to different results: rule 110 returns a blue pixel,  $x(\mathbf{r}, t) = 1$ , while rule 184 gives a white pixel,  $x(\mathbf{r}, t) = 0$ .

If the past three pixels are all 1 (blue), then  $t = 9$  in rule 110 shows that the pixel at  $t = 10$  will be white. Since this configuration does not occur in rule 184 we cannot infer from this simulation what rule 184 would do. The last pages of [Schiff \(2008\)](#) or [mathworld.wolfram.com/ElementaryCellularAutomaton.html](http://mathworld.wolfram.com/ElementaryCellularAutomaton.html) show a complete overview of all 256 elementary rules. In particular, rule 184 gives a blue pixel for three blue past neighbors.

## E Local Sensitivity

While LSC identifies those areas which pass information across space-time, it is not clear yet that these are really the elements driving the system forward. Local sensitivity analysis (Shalizi et al., 2006) infers structure by perturbing the system at each point  $(\mathbf{r}, t)$  and measuring how much the system changes in the future by this perturbation. If the change is large, then  $(\mathbf{r}, t)$  is important; otherwise it's not. Doing this for each point in the field, gives a local sensitivity map of the field.

While this works well for simulations, for real-world data it is not immediately obvious how to achieve such a perturbation. However, the methods developed here for local statistical complexity can be used to approximate the local sensitivity measure, even for systems which can not be simulated from.

Assume we want to check if the point  $(\mathbf{r}_0, t_0)$  in the observed field  $X(\mathbf{r}, t)$ ,  $\mathbf{r} \in \mathbf{S}$ ,  $t \in \{1, \dots, T\}$  is very sensitive. Since the data is observed it is not possible to perturb  $X(\mathbf{r}_0, t_0)$  and see how the system would evolve; however, using the light cone setting it is possible to check how much a perturbation of the PLC  $\ell_0^- := \ell^-(\mathbf{r}_0, t_0)$  with additive noise  $\varepsilon$ ,  $\tilde{\ell}_0^- := \ell_0^- + \varepsilon$ , changes the predictive distribution. If the conditional distribution  $\mathbb{P}(\ell^+ | \tilde{\ell}_0^-)$  is far away from  $\mathbb{P}(\ell^+ | \ell_0^-)$  (in some metric), then  $(\mathbf{r}_0, t_0)$  corresponding to PLC  $\ell_0^-$  is very sensitive; if they are very similar, then this point is not sensitive.

Section 5 and 6.2.2 show how to estimate the function  $\epsilon : \ell^- \mapsto [\ell^-]$ . Thus for an estimate of the future conditional distribution of  $\ell^+ | \tilde{\ell}_0^-$  we can look at the distribution of  $\epsilon(\tilde{\ell}_0^-)$ .