ABSTRACT

Title of dissertation: STATISTICAL METHODS FOR ANALYZING TIME SERIES DATA DRAWN FROM COMPLEX SOCIAL SYSTEMS

David Darmon, Doctor of Philosophy, 2015

Dissertation directed by: Professor Michelle Girvan
Department of Physics

Professor William Rand
Smith School of Business

The rise of human interaction in digital environments has lead to an abundance of behavioral traces. These traces allow for model-based investigation of human-human and human-machine interaction 'in the wild.' Stochastic models allow us to both predict and understand human behavior. In this thesis, we present statistical procedures for learning such models from the behavioral traces left in digital environments.

First, we develop a non-parametric method for smoothing time series data corrupted by serially correlated noise. The method determines the simplest smoothing of the data that simultaneously gives the simplest residuals, where simplicity of the residuals is measured by their statistical complexity. We find that complexity regularized regression outperforms generalized cross validation in the presence of serially correlated noise.

Next, we cast the task of modeling individual-level user behavior on social me-

dia into a predictive framework. We demonstrate the performance of two contrasting approaches, computational mechanics and echo state networks, on a heterogeneous data set drawn from user behavior on Twitter. We demonstrate that the behavior of users can be well-modeled as processes with self-feedback. We find that the two modeling approaches perform very similarly for most users, but that users where the two methods differ in performance highlight the challenges faced in applying predictive models to dynamic social data.

We then expand the predictive problem of the previous work to modeling the aggregate behavior of large collections of users. We use three models, corresponding to seasonal, aggregate autoregressive, and aggregation-of-individual approaches, and find that the performance of the methods at predicting times of high activity depends strongly on the tradeoff between true and false positives, with no method dominating. Our results highlight the challenges and opportunities involved in modeling complex social systems, and demonstrate how influencers interested in forecasting potential user engagement can use complexity modeling to make better decisions.

Finally, we turn from a predictive to a descriptive framework, and investigate how well user behavior can be attributed to time of day, self-memory, and social inputs. The models allow us to describe how a user processes their past behavior and their social inputs. We find that despite the diversity of observed user behavior, most models inferred fall into a small subclass of all possible finitary processes. Thus, our work demonstrates that user behavior, while quite complex, belies simple underlying computational structures.

STATISTICAL METHODS FOR ANALYZING TIME SERIES DATA DRAWN FROM COMPLEX SOCIAL SYSTEMS

by

David Darmon

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2015

Advisory Committee:
Professor Michelle Girvan, Chair/Advisor
Professor William Rand, Co-Advisor
Professor Hector Bravo
Professor Radu Balan
Professor James Reggia

Dedication

To my parents, for always believing in me, and to my brother and sister, for showing me how to become a scientist.

Acknowledgments

First, I would like to thank my advisors, Michelle Girvan and Bill Rand. From my first meeting with Michelle, I knew that she would offer just the right insights to launch me into a lifelong journey exploring complex systems. Her creativity and ability to cut to the heart of an issue have been invaluable in both my research and my training as a scientist. Bill introduced me to the value of bringing a diverse methodological toolset to any problem, in business or otherwise. His curiosity continues to serve as the model I aspire to. Without Michelle and Bill, I would not be the researcher that I am today.

Many of my best memories of graduate school took place at the AMSC House on 37th Avenue. The residents of this house, without a doubt, made my success in graduate school possible. The 'founding fathers,' Lee Mendelowitz, Joe Paulson, Andrew Brandon, and Stefan 'Stefe' Doboszczak, made the AMSC House a home. The future residents, James Murphy, Ryan Hunter, Matt Guay, and Chae Clark, kept it one. I cannot imagine graduate school without them. From the after parties to the late night video game sessions to the weekend morning debates, I know I will always miss our time together.

I am also forever indebted to the AMSC department and the friends I have made there. I could not have asked for a better department. I especially want to thank Virginia Forstall and Lucia Simonelli for always believing in me. Asymptotic high five!

My friends from 'life before grad school' have kept me grounded and prevented

me from cloistering in the ivory tower. Dave McClung constantly reminds me that you do not need a PhD to maintain a passion for science. I am grateful our childhood friendship continues on into adulthood. Chris Orser, who I first met by a happy accident, continues to amaze me with his depth and breadth of knowledge. He constantly reminds me that there is more to the world than science, and that I do not, in fact, have all the answers.

It almost goes without saying, which also means I do not say it enough: my family has been the bedrock of my life. With the Macintosh he brought home from work, my father introduced me to computing at an early age. Combined with dry ice fireworks and other DIY chemistry experiments, he provided the spark that grew into my passion for science. My mother has been a rock throughout my life, and graduate school has been no exception. I cannot begin to thank her for the support she has given me throughout my life. Our conversations kept me sane, and her unconditional love kept me going. My brother and sister warned me, only partially in jest, against graduate school. From my brother's '8 in the morning until 10 or 12 at night' work schedule to my sister's forever-dying cell lines, they had their reasons. Despite my foolhardiness, they continued to support me, and offered invaluable life advice.

And to all the innumerable others who have helped get me to where I am today: thank you.

Table of Contents

List of Figures

List of Abbreviations

ϵ	Mapping from Past to Causal State
$\{X_t\}$	Output Process
$\{Y_t\}$	Input Process
$\{S_t\}$	Causal State Process
$\{A_t\}$	Aggregate Output Process
$\mathcal{X}$	Output Process Alphabet
$\mathcal{Y}$	Input Process Alphabet
$\mathcal{S}$	Set of Causal States
h_μ	Entropy Rate
C_μ	Statistical Complexity
C_0	Topological Complexity
L	History Length
α	The Statistical Power Associated with CSSR / transCSSR

AR	Autoregressive
CRR	Complexity Regularized Regression
CSM	Causal State Model
CSSR	Causal State Splitting Reconstruction
ESN	Echo State Network
transCSSR	Transducer Causal State Splitting Reconstruction

Chapter 1: Introduction

> *Human beings, viewed as behaving systems, are quite simple; the apparent complexity of our behavior over time is largely a reflection of the complexity of the environment in which we find ourselves [...]*
>
> – Herbert Simon, *The Sciences of the Artificial*, page 11

For the first time in human history, we have massive data sets about how humans interact with each other and with their environment. This data arises from diverse settings, from cell phone call records to geotagged images to interactions on social media. The prevalence of such data has caused a renaissance in the social sciences, allowing for the quantitative study of how large populations of humans behave outside of the laboratory. However, the abundance of such data does not in itself offer understanding. Instead, we must turn to models that describe and explain this data. This has been the core mission behind the nascent field of computational social science [3]. In this thesis, we seek to advance this mission by developing statistical methods for understanding time series data drawn from complex social systems.

The main stochastic models and statistical tools we use in our investigations originate from the discipline of computational mechanics [4]. Computational me-

chanics began with the study of nonlinear dynamical systems [5], and was initially used as a tool to study the route to chaos of the logistic map. It has since been developed into a methodology for time series [4, 6], time-varying random fields [7], and input-output systems [6, 8, 9]. Computational mechanics has been used to investigate many application domains, including ecology [10], crystallography [11], neuroscience [12], anomaly detection [13] and social media [14].

The key insight behind the various facets of computational mechanics is the value of viewing a system in terms of its *predictive distribution*. In this sense, computational mechanics has much in common with older work on predictive processes [15] and more recent work on modeling autonomous and controlled dynamical systems [16]. Unlike most modeling approaches, computational mechanics provides a constructive procedure for determining the unique, minimally complex, maximally predictive representation of a time series, input-output system, or time-varying random field. Thus, computational mechanics provides the simplest model that can predict as well as any rival model. Moreover, because the model construction relies solely on distributions over observed symbol sequences, computational mechanics allows for the determination of models directly from data with few *a priori* assumptions. In the terminology of computational mechanics, approaches for doing so are known as reconstruction algorithms, since they reconstruct the appropriate model from the supplied data. Such reconstruction algorithms embody the law of parsimony, also known as Ockham's razor, since they infer the simplest model that predicts well.

In this thesis, we apply inferential methods arising from computational me-

chanics to model several data sets originating from complex social systems. In Chapter 2, we develop a new method for tuning a non-parametric smoother in the case of serially correlated residuals. Most standard methods for tuning non-parametric smoothers assume that the residuals can be treated as independent and identically distributed noise. In contrast, we leverage computational mechanics to allow for serial correlation in the residuals. We then apply this new method to time series data arising from the Dow Jones Industrial Average, and investigate how market behavior has changed in terms of large- and small-scale dynamics of day-to-day performance.

In Chapter 3, we consider the task of predicting the behavior of individuals on a social media service. We cast this problem in terms of forecasting a categorical time series, and present two solutions. The first, based on computational mechanics, begins by assuming that the observed data is generated by as simple a process as possible, and then adds complexity to the model as the data requires it. The second, based on reservoir computing, starts from a model that allows for very complicated dynamics, and then relaxes down to a simpler model. We demonstrate the performance of these models on a heterogeneous data set drawn from user behavior on social media, and investigate the characteristics of user behavior that lead one model to outperform the other.

In Chapter 4, we move from predicting individual-level behavior to predicting the behavior of collections of individuals. We seek to predict times in which the aggregate behavior is elevated compared to a baseline. We consider three complementary approaches: the first based on seasonal variability, the second based on

aggregate memory, and the third based on individual excitability. We find that the best model for this task varies depending on the acceptable trade-off between true and false positives, but that the non-seasonal approaches perform best overall.

In Chapter 5, we move from a predictive to a descriptive framework, and explore the computational landscape of user behavior on social media. To do so, we infer seasonal, self-driven, and socially-driven models of user behavior across a heterogeneous collection of 15K users on Twitter. We find that a small class of models can describe the majority of observed user behavior, but that there is great diversity across the models within this class.

This thesis is based on the following publications:

- Chapter 2: D. Darmon and M. Girvan. Complexity-regularized regression for serially-correlated residuals with applications to stock market data. *Entropy*, 17(1):1–27, 2014.

- Chapter 3: D. Darmon, J. Sylvester, M. Girvan, and W. Rand. Predictability of user behavior in social media: Bottom-up v. top-down modeling. In *ASE/IEEE Int'l Conf. on Social Computing*, pages 102–107, 2013.

- Chapter 4: J. Harada, D. Darmon, M. Girvan, and W. Rand. Forecasting high tide: Predicting times of elevated activity in online social media. In *IEEE/ACM Int'l Conf. on Advances in Social Network Analysis and Mining*, 2015.

- Chapter 5: D. Darmon, W. Rand, M. Girvan. The Computational Landscape of User Behavior on Social Media. In preparation.

Chapter 2: Complexity Regularized Regression for Time Series with Serially Correlated Errors

2.1 Introduction

When studying the short-term behavior of a time series, a common assumption is that the time series can be treated as a realization from a trend stationary stochastic process. That is, it is assumed that the time series can be modeled as the sum of a deterministic trend and a stationary stochastic process, where the deterministic trend is assumed to vary slowly compared to the stochastic process [17]. As an example, consider the closing price of the Dow Jones Industrial Average (DJIA) over time, shown in Figure 2.1. Over large enough timescales, the market exhibits clear trends. However, when considering short timescale (e.g., inter-day) behavior of the market, these long-term trends could mask the dynamics of day-to-day fluctuations. For example, because the value of the market tends to increase over time, nearby time points will tend to be positively correlated. However, this long-term correlation tells us nothing about the short-term dynamics of the market. A natural solution is to estimate the trend and remove it. The problem of estimating the long-term trend present in a time series can be cast as a time-domain regression problem,

where we regress the observed values of the time series on the time index. Within this framework, nonparametric regression methods may be used to infer the underlying trend without making strong *a priori* assumptions on its form. The literature on nonparametric regression is rich and includes techniques, such as kernel-based methods, smoothing splines, wavelets and series expansions, in terms of orthogonal functions [17–19]. A complementary approach to non-parametric regression for more flexible modeling can be found in the tools from robust statistics, which offer statistical procedures that are flexible to deviations from an assumed model [20]. While the general issues related to robust statistics have received considerable attention in the literature, their application to time series analysis has largely been limited to robust tests for serial correlation [21, 22] and linear trends [23]. To the best of our knowledge, little to no work has been done on robust statistics (in the formal sense) for trend estimation under correlated errors.

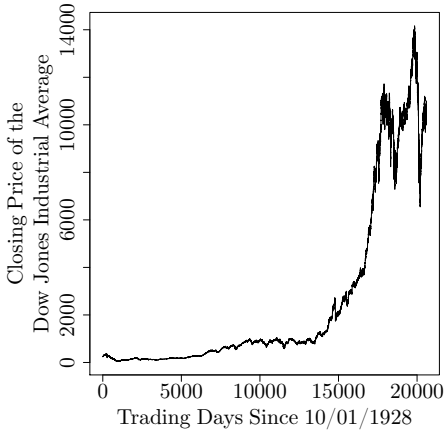


Figure 2.1: The closing price of the Dow Jones Industrial Average as an example of a non-stationary time series.

While nonparametric methods have the advantage that they can adapt to

regularities in the data, they also come with tuning parameters that must be set by the investigator. These tuning parameters, also known as smoothing parameters, include the bandwidth for kernel-based methods, the effective degrees of freedom for smoothing splines and the number of basis functions for orthogonal expansions. The value for the smoothing parameter is often chosen via data-driven methods, such as cross-validation, where the data is split into a training set used to infer the model and a tuning set used to select the smoothing parameter [18]. However, many such procedures are designed for regression where the residuals are uncorrelated, as might be the case when performing regression on data in which each data point corresponds to a separate measurement from some underlying population. The assumption of uncorrelated residuals clearly does not hold for time-domain regression where we expect short-term serial correlations. For example, early work in econometrics found non-trivial serial correlations in stock prices [24]. Serial correlations in residuals greatly impact the performance of automated, data-dependent methods for choosing smoothing parameters. For instance, in [25], Hart shows that for kernel regression estimation, when the residuals are drawn from an order-one autoregressive (AR(1)) process [17] with coefficient ϕ, for $\phi \gtrsim 0.17$ (with the exact value depending on the choice of kernel), in the limit of infinite data, cross-validation will choose an estimate that nearly interpolates the data. Thus, even for a very simple model of residuals with weak serial correlations, the standard method for choosing the smoothing parameter of a nonparametric regression method will result in a trend estimate that adapts to the correlations in the residuals, rather than reflecting the true trend. In this case, neither the trend nor the residuals are correctly estimated.

This is especially problematic when the properties of the residuals are the object of study, as a near-interpolating trend estimate will cause the residuals to look like numerical noise. Many approaches have been proposed to generalize cross-validation for serially-correlated errors [26–29]. These methods typically involve block-wise versions of cross-validation, where appropriately chosen blocks of a time series are removed during each fold of the cross-validation procedure. See [30] for a review of other literature on regression with correlated errors.

A new class of nonparametric regression methods, first proposed in [31], considers the regression problem from a different perspective, where the focus shifts from the regression curve to the residuals. Instead of considering the estimator's fidelity to the underlying curve, the method seeks to make the residuals look as random as possible while maintaining as simple a regression curve as possible. This method thus hinges on an often overlooked point from regression: under the assumption of most methods, residuals resulting from a smoothing method should look like white noise. However, because of this construction, the method from [31] does not immediately apply to time series with serially-correlated residuals.

In this chapter, we develop a nonparametric regression technique that is model-agnostic with respect to both the long-term trend and the serial correlations in the residuals. This method relies on tools from the field of computational mechanics [4], a formalism for dissecting the structure and randomness present in a stationary stochastic process. Computational mechanics allows us to greatly expand the class of possible residuals considered in [31]. Any nonparametric smoother may be used to estimate the trend, and the residuals need not be white noise, though we do limit

memory length. We then apply this technique to the Dow Jones Industrial Average, a time series where we expect serially-correlated residuals, and investigate how both the long-term and short-term behavior of the market has changed over time.

2.2 Methodology

2.2.1 Regression for Time Series

A typical model for time series is that the observed value Y_t can be treated as the sum of a "true" trend r_t plus some deviation from the trend η_t [17]. That is, we have the model:

$$Y_t = r_t + \eta_t, \quad t = 1, \ldots, T. \tag{2.1}$$

The residual process $\{\eta_t\}$ is typically specified in terms of its first two moments. In particular, it is assumed to have zero mean:

$$E[\eta_t] = 0, \text{ for all } t \tag{2.2}$$

(if not, this non-zero value would be incorporated into r_t) and some autocorrelation structure dependent on the lag between two time points,

$$R(t, s) = R(|t - s|) = E[\eta_t \eta_s], \tag{2.3}$$

which specifies the form of the serial correlation. It should be noted that the form (2.1) for a time series addresses a very particular kind of non-stationarity. Other types of non-stationarity commonly occur in time series, including, but not limited to, heteroskedastic and heavy-tailed deviation processes. These types of non-stationarities are typically investigated using autoregressive conditional heteroskedastic (ARCH) models and their generalizations [17]. As we assume the trend stationary model, we do not address these types of non-stationarities with our method.

As stated and without careful interpretation, (2.1) can be problematic, both theoretically and practically [17]. One interpretation of this formulation is to consider $\{Y_t\}$ as the discretization of a sample path from a continuous time stochastic process:

$$Y(t) = r(t) + \eta(t). \tag{2.4}$$

Such an interpretation frequently occurs with financial time series due to the prevalence of stochastic differential equation models, such as the famous Black–Scholes model [32] for options. This formulation also frequently occurs in longitudinal and functional data analysis, where the function $r(t)$ is estimated using several independent realizations from (2.4) [33, 34]. However, this model is inappropriate for time-domain smoothing, since it assumes that both $r(t)$ and $\eta(t)$ vary continuously in time. Thus, for small values of Δt, we expect $Y(t \pm \Delta t)$ to be nearly the same as $Y(t)$, so nothing is gained from smoothing about the time index t. Moreover, under

this formulation and without any additional assumptions, because of the smoothness in $\eta(t)$, it is impossible to extract the trend with only a single realization. Because of this, a popular alternative formulation of (2.1) for time-domain smoothing is to consider the model:

$$Y_t = g(t/T) + \eta_t, t = 1, 2, \ldots, T, \tag{2.5}$$

which places the estimation problem within the framework of nonparametric estimation with equispaced design points [35–37]. This formulation explicitly assumes that the time trend $r(t) = g(t/T)$ varies more slowly than the stochastic component $\{\eta_t\}$, thus motivating the use of smoothing about the time index and allowing for the recovery of the time trend from a single realization.

In nonparametric regression, we seek an estimator $\hat{r}_t$ that should capture the true trend r_t without picking up too much of the false "trend" introduced by the residual term η_t. In the case of white noise, for example, this can be done by averaging over nearby time points. If the noise terms truly are uncorrelated, this averaging process reduces the pointwise variance in $\hat{r}_t$, but also increases its pointwise bias, since, in general, $r_t \neq r_{t'}$ for $t \neq t'$. The amount of smoothing is decided by a smoothing parameter λ. Some standard smoothing methods for time series include kernel smoothing, smoothing splines and local polynomial smoothing, all of which have an associated smoothing parameter [17].

The choice of the smoothing parameter falls within the larger statistical framework of model selection [38]: from a certain class of models, how do we choose the

model that best reflects the underlying process that generated the data? The model selection procedure depends on the operationalization of "best": a model may be chosen to maximize a likelihood, minimize a certain error function, *etc.* In the case of estimating a regression function via smoothing, a standard approach is to choose the smoothing parameter λ by cross-validation on the estimated mean-squared prediction error between the observation Y_t and regression function $\hat{r}_t$ [18]. Let the T data points be indexed by $I = \{1, 2, \ldots, T\}$. To perform cross-validation, we partition the indices into K disjoint subsets $\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_K$. For each subset of indices, we use the data indexed by $I \setminus \mathcal{P}_k$ to estimate $\hat{r}_t^{(-k)}(\lambda)$, *i.e.*, $\hat{r}_t^{(-k)}(\lambda)$ is estimated using all of the data except the data indexed by the subset $\mathcal{P}_k$. The mean-squared error is then computed on the held out data for each subset:

$$\widehat{\mathrm{MSE}}(\hat{r}^{(-k)}; \lambda) = \frac{1}{|\mathcal{P}_k|} \sum_{t \in \mathcal{P}_k} (Y_t - \hat{r}_t^{(-k)}(\lambda))^2, \quad k = 1, \ldots, K. \tag{2.6}$$

The estimate for the mean-squared error is then determined by averaging the mean-squared error over the held-out subsets, giving:

$$\widehat{\mathrm{MSE}}(\lambda) = \frac{1}{K} \sum_{k=1}^{K} \widehat{\mathrm{MSE}}(\hat{r}^{(-k)}; \lambda). \tag{2.7}$$

Finally, the smoothing parameter is taken to minimize this estimate of the mean-squared error, giving:

$$\hat{\lambda} = \arg\min_{\lambda} \widehat{\mathrm{MSE}}(\lambda). \tag{2.8}$$

This approach to choosing the smoothing parameter can be problematic when the residuals are serially correlated [29], as we have discussed in the introduction.

2.2.2 Model-Free Regression

As we have stated, regression models of the form (2.1) typically require the specification of a probability model for the stochastic component $\{\eta_t\}$. Recent work by P. L. Davies and co-authors has proposed methods for nonparametric regression without such models [31, 39]. We recast their problem, which is stated for general regression, in terms of time-domain regression. The basic idea, as summarized in [40], is to choose the simplest regression function that makes the residuals "look random". The fact that the residuals should look random is a natural consequence of the statistical model for the regression function. The problem of deciding whether the residuals look sufficiently random is well developed and typically involves simple diagnostic plots and tests on the residuals [41,42]. As a simple example, consider the case where the observed time series is a sinusoid over a single period corrupted by a small amount of white noise. Using a linear trend will induce both short- and long-range correlation in the residuals: residuals near the peaks/troughs of the sinusoid will be positively correlated with residuals near the peaks/troughs and negatively correlated with residuals near the troughs/peaks. At the opposite extreme, a near-interpolating trend would result in uncorrelated residuals, but the estimated trend will also have many degrees of freedom. Davies *et al.*'s approach seeks to balance between these two extremes.

The setup for Davies *et al.*'s approach is as follows. We observe T observations of a time series $Y_1, \ldots, Y_T$, and we seek the regression function r, such that we will model $Y_t = r_t$. For a given choice of r, we may compute the residuals:

$$\eta_t = Y_t - r_t, t = 1, \ldots, T. \tag{2.9}$$

Define $\eta(r) = (\eta_1, \ldots, \eta_T)$. We then specify a test for randomness in these residuals, $R(\eta(r))$ where:

$$R(\eta(r)) = \begin{cases} 1: \text{ reject randomness in } \eta(r) \\ 0: \text{ do not reject randomness in } \eta(r) \end{cases} \tag{2.10}$$

For example, R might be the Wald–Wolfowitz runs test [43], a nonparametric test for the independence of binary random variables. We will return to this idea shortly when we propose our extension to Davies *et al.*'s work. We also define a "complexity" measure on r, $\psi(r)$. For example, ψ might measure the number of extrema of r or the integrated squared second-derivative ("wiggliness") of r,

$$\psi(r) = \int (r_t'')^2 \, dt. \tag{2.11}$$

Once R and ψ are specified, we seek the r that solves:

$$\min_r \psi(r) \tag{2.12}$$

$$\text{subject to } R(\eta(r)) = 0. \tag{2.13}$$

That is, we seek the minimally complex regression function $\hat{r}$, such that the residuals "look random". This approach has been operationalized by Davies *et al.* in their runs method for nonparametric regression [31].

2.2.3 Computational Mechanics

A key part of Davies *et al.*'s method was the assumption that the residuals η_i should look "random", where they operationalize random to mean that the residuals should appear like a realization from a white noise process. We could allow for the residuals to appear like realizations from more general stochastic processes, but we then need simple criteria to characterize the randomness of the residuals. Computational mechanics, a formalism for investigating stationary stochastic processes, provides such criteria. We now present a brief overview of computational mechanics. A high-level review may be found in [44]. A more mathematical treatment may be found in [4].

We restrict ourselves to a discrete time, discrete state stochastic process $\{X_t\}_{t\in\mathbb{Z}}$ taking values from the finite alphabet $\mathcal{X}$. For example, when $\{X_t\}_{t\in\mathbb{Z}}$ corresponds to a stochastic process defined over all bi-infinite binary strings, $\mathcal{X} = \{0, 1\}$. We will use the standard convention of denoting a realization from this process at a fixed time t by x_t. For a time point t, we define the past of the process as:

$$X_{-\infty}^{t-1} = (\ldots, X_{t-2}, X_{t-1}) \tag{2.14}$$

and the future (including the present) as:

$$X_t^\infty = (X_t, X_{t+1}, \ldots), \tag{2.15}$$

and denote the set of all semi-infinite pasts by $\boldsymbol{\mathcal{X}}^-$ and all semi-infinite futures by $\boldsymbol{\mathcal{X}}^+$. We will denote particular realizations of semi-infinite pasts and futures by $x_{-\infty}^{t-1}$ and x_t^∞, respectively. Computational mechanics presents a particular model for use in the prediction of this process. For prediction, we ultimately desire to make a statement about the future of the process, conditioned on the particular past we have observed. That is, we seek:

$$P(X_t^\infty | X_{-\infty}^{t-1} = x_{-\infty}^{t-1}). \tag{2.16}$$

While we might be able to predict using the entire past of the process, the insight of computational mechanics is that we can instead use a statistic that compresses the past as much as possible without losing any predictive ability. It can be shown that the unique minimal sufficient predictive statistic of the past $X_{-\infty}^{t-1}$ for the future X_t^∞ of a conditionally stationary stochastic process is the equivalence class over predictive distributions. For two pasts $x_{-\infty}^{t-1}$ and $y_{-\infty}^{t-1}$, we define an equivalence relation, such that $x_{-\infty}^{t-1} \sim y_{-\infty}^{t-1}$ if:

$$P(X_t^\infty | X_{-\infty}^{t-1} = x_{-\infty}^{t-1}) = P(X_t^\infty | X_{-\infty}^{t-1} = y_{-\infty}^{t-1}) \tag{2.17}$$

as probability mass functions. In other words, two pasts are equivalent if they

result in statistically-equivalent futures. Using this equivalence relation, we can define equivalence classes over pasts p, such that

$$[p] = \{x_{-\infty}^{t-1} \in \boldsymbol{\mathcal{X}}^- : P(X_t^\infty | X_{-\infty}^{t-1} = x_{-\infty}^{t-1}) = P(X_t^\infty | X_{-\infty}^{t-1} = p)\}. \tag{2.18}$$

In other words, for each possible predictive distribution, we choose a candidate past p, and $[p]$ represents all pasts that induce this predictive distribution. We can thus think of p as a particular past or as the label for this class of pasts. Typically, we will take the second perspective. We define our statistic $\epsilon : \boldsymbol{\mathcal{X}}^- \to \mathcal{S}$ as mapping a past into the equivalence class for that past,

$$\epsilon(X_{-\infty}^{t-1}) = [X_{-\infty}^{t-1}]. \tag{2.19}$$

The statistic ϵ has been proven [4] to be the unique, minimal sufficient statistic of the past of a stationary stochastic process for its future. We can think of ϵ as partitioning the set of all pasts $\boldsymbol{\mathcal{X}}^-$ based on the conditional futures they induce. The combination of the equivalence classes, as well as the allowed transitions between them is called the ϵ-machine or causal state model for the process $\{X_t\}_{t\in\mathbb{Z}}$. The mapping by ϵ of the stochastic process to its predictive equivalence classes results in a new stochastic process $\{S_t\}_{t\in\mathbf{Z}}$, called the causal state process. One of the important properties of this process is its relationship to the statistical complexity, denoted C_μ, of the stochastic process. The statistical complexity of a stochastic process is the average number of bits of its past necessary to optimally predict its

future. For a conditionally stationary stochastic process, the statistical complexity is equivalent to the Shannon entropy of the causal state process,

$$C_\mu = H[S] \tag{2.20}$$

$$= -E[\log_2 P(S)] \tag{2.21}$$

$$= -\sum_{s \in \mathcal{S}} P(S = s) \log_2 P(S = s), \tag{2.22}$$

where $\mathcal{S}$ is the set of equivalence classes and $P(S = s)$ is the asymptotic probability associated with causal state s. The statistical complexity is also equivalent to the mutual information between the past of the process and the causal state associated with that past and, thus, captures the amount of information about the past stored in the causal state. A complementary quantity associated with a stochastic process is its entropy rate,

$$h_\mu = \lim_{t \to \infty} H[X_t | X_{-\infty}^{t-1}], \tag{2.23}$$

which represents the average uncertainty in the next symbol given the past. When the ϵ-machine representation of a stochastic process is available, the entropy rate is computable [45] in terms of the uncertainty in the next symbol conditional on the

current causal state,

$$h_\mu = H[X_t|S_{t-1}] \tag{2.24}$$

$$= -E[\log_2 P(X_t|S_{t-1})] \tag{2.25}$$

$$= -\sum_{s\in\mathcal{S}}\sum_{x\in\mathcal{X}} P(X_t = x, S_{t-1} = s)\log_2 P(X_t = x|S_{t-1} = s) \tag{2.26}$$

$$= -\sum_{s\in\mathcal{S}} P(S_{t-1} = s)\sum_{x\in\mathcal{X}} P(X_t = x|S_{t-1} = s)\log_2 P(X_t = x|S_{t-1} = s), \tag{2.27}$$

where, again, $P(S_{t-1} = s)$ is the asymptotic probability associated with causal state s.

The computational mechanics formalism requires knowledge of the full predictive distribution (2.16) in order to determine the equivalence relation that defines the ϵ-machine. Since this distribution is not known in practice, we must infer the ϵ-machine associated with $\{X_t\}_{t\in\mathbb{Z}}$ using a statistical procedure. For this work, we use the causal state splitting reconstruction (CSSR) [46] algorithm. CSSR has been used in many application domains, including ecology [10], crystallography [11], neuroscience [12], anomaly detection [13] and social media analysis [47]. This algorithm provides an estimator for the ϵ-machine associated with a realization of the observed process $\{X_t\}_{t=1}^{T}$ by splitting candidate causal states. To do this, a maximum history length $L_{\max}$ is chosen, and all histories are initially placed in a single state. The value of L is then incremented from zero to $L_{\max}$, and histories x_{t-L}^{t-1} in a state are split if their one-step-ahead predictive distribution (called a morph in the computational

mechanics literature):

$$P(X_t|X_{t-L}^{t-1} = x_{t-L}^{t-1}) \tag{2.28}$$

differs significantly (at size α) from the one-step-ahead predictive distribution associated with their causal state,

$$P(X_t|S_{t-1} = \hat{\epsilon}(x_{t-L}^{t-1})).$$

The states resulting from this procedure are precausal, in the sense that they are optimal for one-step-ahead prediction. The precausal states are then refined to the causal states by taking advantage of the unifilarity of the ϵ-machine [45]. That is, for a given causal state s_{t-1} and an emission symbol x_t, the causal state s_t at the next time step updates as $s_t = T(s_{t-1}, x_t)$, where $T(\cdot, \cdot)$ is a one-to-one mapping from the previous causal state and the emission symbol to the next causal state. The precausal states are split to ensure this one-to-one mapping holds. The entire CSSR procedure results in an ϵ-machine that is a consistent estimator for the true ϵ-machine assuming the true stochastic process is conditionally stationary, has finitely many causal states and has finite-length suffixes of length $L_{\max}$ or smaller in each causal state [48].

2.2.4 Complexity Regularized Regression

Using the tools presented in the previous section, we now extend Davies *et al.*'s approach. Again, we compute the residuals:

$$\eta_t = Y_t - r_t, t = 1, \ldots, T. \tag{2.29}$$

We then transform the residuals η_t into binary random variables using the Heaviside function Θ to give:

$$B_t = \Theta(\eta_t) \tag{2.30}$$

$$= \begin{cases} 0 & : \eta_t \leq 0 \\ 1 & : \eta_t > 0 \end{cases} \tag{2.31}$$

This binary sequence $\{B_t\}_{t=1}^{T}$ is then used to infer a causal state model via the CSSR algorithm. Call this estimator for the causal state model $\hat{\epsilon}$. The estimator $\hat{\epsilon}$ consists of the estimates for the equivalence classes of pasts, the predictive distributions those equivalence classes induce and the allowed transitions between the equivalence classes.

We use the inferred causal state model $\hat{\epsilon}$ to extend Davies *et al.*'s approach in two ways. First, we replace the constraint term $R(\eta(r))$ by $C_\mu(B(r))$, the statistical complexity of the causal state model inferred from the binarized residuals. For an independent and identically distributed stochastic process, $C_\mu = 0$, and we see that if we enforce the constraint $C_\mu(B(r)) = 0$, we recover the same criterion from Davies

et al.'s runs test-based regularization term $R(\eta(r))$, though it should be noted that the runs test may be more powerful than using C_μ to test for independence. Second, instead of directly inferring $\hat{r}$, we will assume a nonparametric model for $\hat{r}$, indexed by a smoothness level λ, and infer the $\hat{r}_\lambda$, such that:

$$\hat{r}_\lambda = \arg\min_{r_\lambda} C_\mu(r_\lambda). \tag{2.32}$$

For example, with smoothing splines, λ might be the effective degrees of freedom. For kernel smoothing methods, λ might be the bandwidth of the kernel used. If we take $\psi(r_\lambda)$ to be (2.11), then $\psi(r_\lambda)$ will be monotonic in λ, and we can instead state our optimization problem as:

$$\hat{\lambda} = \arg\min_{\lambda} C_\mu(\lambda). \tag{2.33}$$

Thus, we see that this method seeks the simplest regression function, as measured by λ, which makes the residuals have minimal statistical complexity. We call this method complexity-regularized regression (CRR).

2.2.4.1 Details for Operationalization

The statistical complexity of an ϵ-machine depends on both the number of causal states associated with the machine and the probabilities associated with those causal states. Thus, the number of causal states gives another proxy for the structure present in a stochastic process. In [49], the topological complexity of an

ϵ-machine was defined as the logarithm of the number of states $N(\epsilon)$ of the model ϵ,

$$C_0 = \log_2 N(\epsilon). \tag{2.34}$$

The topological complexity is an upper bound for the statistical complexity of a causal state model. Thus, we can take C_0 as a proxy for the statistical complexity of the causal state model. We do this for two reasons. First, statistical fluctuations inherent in inferring C_μ from finite data will have less of an impact on C_0. Second, by virtue of how the sequence $\{B_t\}_{t=1}^{T}$ is generated, changes in the number of states will be more useful than changes in the probabilities of the transitions between those states. Simply, topological changes in the causal state model are more useful for the task at hand. Thus, in practice, we choose:

$$\hat{\lambda} = \arg\min_{\lambda} C_0(\lambda). \tag{2.35}$$

The CSSR algorithm has two parameters: α, a significance level used in the state-splitting step of the algorithm, and $L_{\max}$, the maximal history to consider when inferring (5.10). The significance level α controls the probability that we do not assign a history to an existing causal state when it belongs to that causal state and is fixed at $\alpha = 0.001$ for all experiments in this chapter. The maximal history $L_{\max}$ balances the complexity of the causal state models that can be inferred by CSSR and the accuracy with which the one-step-ahead predictive distributions are

inferred. Thus, the value of $L_{\max}$ controls the well-known bias-variance tradeoff present in all model selection problems [18]. If $L_{\max}$ is too small, the true causal state model will not be inferable using CSSR, because the histories will not resolve correctly into their true causal states. If $L_{\max}$ is too large relative to the length of the time series, the one-step-ahead predictive distributions will be poorly estimated, which will lead to spurious splitting of histories. A useful heuristic for choosing $L_{\max}$, as recommended in [46], is to take it to be the largest value, such that the joint distribution is consistently estimated. For the class of stochastic processes that include those with finite-state ϵ-machine representations, this bound is given by:

$$L_{\max} < \frac{\log_2 T}{h_\mu + c}, \tag{2.36}$$

where h_μ is the entropy rate of the stochastic process and c is some positive constant [50]. For all examples in this chapter, we fix $L_{\max} = 5$.

2.3 Simulation Experiments

In this section, we demonstrate CRR with a synthetic trend stationary time series that decomposes as in (2.1) into a trend plus residual activity about the trend. We take the trend to be the sum of a finite number of sinusoids with a single dominant frequency. Thus, we assume that the underlying trend has a single dominant scale. We allow serial correlations in the residuals by sampling them from a linear autoregressive process of order one. As mentioned in the Introduction, even for very weak serial correlation in such a process, standard methods, such as

cross-validation, fail at choosing an appropriate smoothing parameter for the trend estimate. By varying the serial correlation in the residuals, we can explore how the performance of CRR compares to methods like cross-validation with increasing serial correlation.

2.3.1 The Generative Model

To test the performance of complexity regularized regression, we sampled 1000 regression curves, indexed by $s = 1, 2, \ldots, 1000$, using the generative model:

$$r^*_{s,t} = \sum_{i=1}^{10} \cos(2\pi\omega_{s,i}t + \delta_{s,i}), \quad t = 0, 1, \ldots, 4999 \tag{2.37}$$

where $\omega_{s,i} \overset{\text{i.i.d.}}{\sim} N(\omega_0, 0.001^2)$ and $\delta_{s,i} \overset{\text{i.i.d.}}{\sim} \text{Uniform}(0, 2\pi)$. For the high frequency examples, we take $\omega_0 = \frac{1}{100}$, and for the low frequency examples, we take $\omega_0 = \frac{1}{10{,}000}$. Thus, each regression curve is the sum of 10 sinusoids with random frequencies and phases, but with a single dominant scale dictated by ω_0. Each regression curve is then normalized, so that its range lies in $[-1, 1]$, giving the final set of curves $r_{s,t}$, $s = 1, \ldots, 1000$. The range-normalization was done to maintain the signal-to-noise ratio between the true regression curve and the residuals. The set $\{r_{s,t}\}_{s=1}^{1000}$ provides a test bed of trends that have a single principle scale (either at a low or high frequency) with variation in that structure dictated by the random frequencies and phase shifts. See Figure 2.2 for sample realizations from (2.37).

Using these true regression curves, we generate the observed values $Y_{s,t}$ using

the model:

$$Y_{s,t} = r_{s,t} + \eta_{s,t} \tag{2.38}$$

where the noise sequence is either white noise or an AR(1) process. In the white noise case, the residuals are taken to be $\eta_t \overset{\text{i.i.d.}}{\sim} N(0, \sigma^2)$. In the correlated noise case, we take the residuals to be samples from an AR(1) process with variance σ^2 and lag-one coefficient ϕ. That is, the residuals are a realization of:

$$\eta_t = \phi\, \eta_{t-1} + \epsilon_t, \quad t = 0, \ldots, 4999 \tag{2.39}$$

with $\epsilon_t \sim N(0, (1-\phi^2)\sigma^2), t = 0, \ldots, 4999$. We take ϵ_t to have variance $(1-\phi^2)\sigma^2$, so that the pointwise variance of η_t is σ^2, making the pointwise noise comparable between the white noise and autoregressive processes. The serial correlation between any two residuals separated by a time lag h is given by:

$$\text{Corr}(\eta_t, \eta_{t+h}) = \phi^h. \tag{2.40}$$

Thus, for positive ϕ, nearby points will be correlated, with that correlation decaying exponentially in the time lag h. In the following numerical experiments, we take $\sigma = 0.1$ and vary $\phi \in \{0.25, 0.5, 0.75\}$.

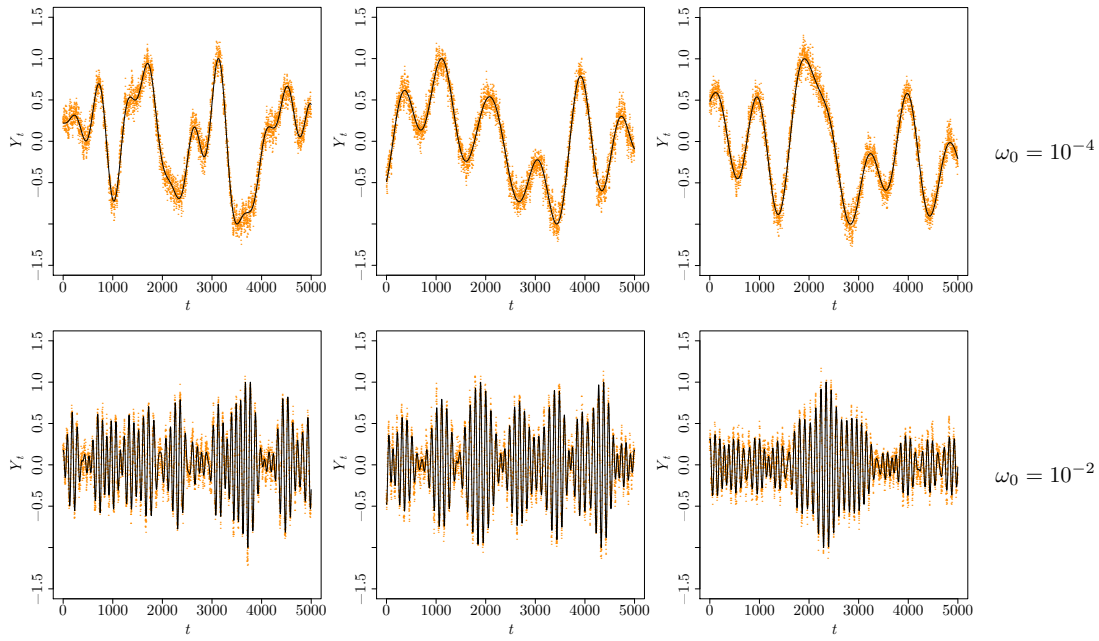


Figure 2.2: Six example realizations from (2.38) with $\{\eta_t\}$ taken to be an order-one linear autoregressive process with $\sigma^2 = 0.1$ and $\phi = 0.75$. The realizations of Y_t are in orange, and the regression curves r_t are in black.

For each realization $Y_{s,t}$, a smoothing spline was used to infer a nonparametric regression function $\hat{r}(t)$ [18]. A smoothing spline is the function that satisfies:

$$\hat{r}(t) = \arg\min_{r \in C^2} \sum_{i=1}^{n} (r(t_i) - y_i)^2 + \lambda \int \{r''(t)\}^2 \, dt, \tag{2.41}$$

where C^2 is the space of twice differentiable functions. The solution to this optimization problem is a natural cubic spline with knots at each of the design points t_i, with coefficients regularized by an amount determined by the smoothing parameter $\lambda \geq 0$. As λ goes to zero, the smoother reduces to the natural cubic spline interpolant of the points $\{(t_i, y_i)\}_{i=1}^{T}$. As λ grows towards larger and larger values, the smoother does not allow any second-derivatives, and we recover the least-squares

fit to the points. Thus, as described in (2.11), λ is the parameter that allows us to control the complexity of the regression function $\hat{r}(t)$. In practice, we will use the effective degrees of freedom $\widehat{\text{dof}}$ of $\hat{r}(t)$ to control the complexity of the regression function. The effective degrees of freedom range from one, which corresponds to the least squares fit, to n, which corresponds to the natural cubic spline interpolant of the data.

For CRR, the residuals were computed for each effective degree of freedom $\widehat{\text{dof}} \in \{1, 6, 11, \ldots, 5001\}$. The CRR degree of freedom $\widehat{\text{dof}}^*$ was chosen as the smallest value that minimized $C_0(\lambda)$. For cross-validation-based regression, we use generalized cross-validation (GCV), a standard cross-validation-based method for choosing the smoothing parameter for a linear smoother [18].

We measure the goodness-of-fit of the inferred $\hat{r}$ by the mean-squared error between the true curve r and the inferred curve $\hat{r}$ at the design points $t \in \{0, \ldots, 4999\}$,

$$\text{MSE}(r, \hat{r}) = \frac{1}{5000} \sum_{t=0}^{4999} (r(t) - \hat{r}(t))^2. \tag{2.42}$$

2.3.2 Simulation Results

We begin by walking through an example of using CRR for a particular realization from (2.37) with $\phi = 0.75$, a case with large positive correlation in the residuals. After building the causal state models with the degrees of freedom for the smoothing spline ranging from one (a linear fit via least squares) to 5000 (a cubic spline interpolant), we can visualize how the topological complexity C_0 varies as the

degrees of freedom increase. Two example plots are shown in Figure 2.3. The left and right panels correspond to low and high frequency trends $r_{s,t}$. The numbers of degrees of freedom chosen by generalized cross-validation and complexity regularized regression are indicated by the red and blue lines, respectively. By (2.35), the choice of 46 and 196 degrees of freedom for the low and high frequency trends correspond to the lowest degrees of freedom for which the number of causal states drops to its minimum, in this case two states.

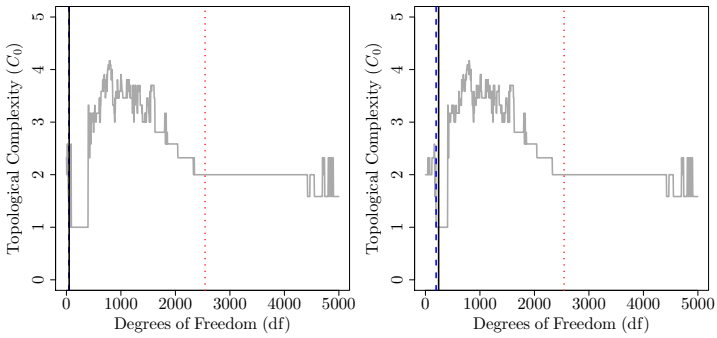


Figure 2.3: The topological complexity C_0 as a function of the number of degrees of freedom for the smoothing spline for the low (**left**) and high (**right**) frequency trends with $\phi = 0.75$. The blue dashed and red dotted vertical lines indicate the degrees of freedom chosen by complexity-regularized regression (CRR) and generalized cross-validation (GCV), respectively. The black solid vertical lines indicate the optimal choice of degrees of freedom for the given realization with respect to the mean-squared error between the true and estimated trends given by (2.43).

Because we know the true value for $r_{s,t}$, in the simulation study, we can also compute the value dof*, such that:

$$\text{dof}^* = \arg\min_{\text{dof}} \text{MSE}(r, \hat{r}_{\text{dof}}). \tag{2.43}$$

This value represents the best choice of the smoothing parameter to minimize the

mean-squared error given the data at hand, if we knew the true trend. The optimal value dof^* is indicated in Figure 2.3 by the black vertical line. We see that for both the low- and high-frequency trends, the degrees of freedom chosen by complexity-regularized regression are much closer than the generalized cross-validation values.

We then define the smoothing parameter bias for a given realization s using either tuning method as:

$$\text{Bias}\left(\widehat{\text{dof}}_s\right) = \widehat{\text{dof}}_s - \text{dof}_s^*, \tag{2.44}$$

or the deviation of the data-driven value from the optimal value if we knew the true trend. Computing this bias across all thousand realizations from (2.37) gives a measure of how close the method came to recovering the true trend using a smoothing spline and the data at hand. See Figure 2.4 for the distribution of the smoothing parameter biases across all of the simulation conditions. A zero bias indicates that the data-driven method performed as well as possible, a positive bias indicates undersmoothing and a negative bias indicates oversmoothing. We see that except for the case where the residuals η_t are white noise, CRR results in a much lower bias, with a tendency to oversmooth the data. By comparison, GCV drastically undersmooths the data. This agrees with the theoretical result reported in [25], though their result was for kernel regressors, not smoothing splines. Both smoothing splines and kernel-based methods are linear smoothers, so we expect the theoretical result to extend to smoothing splines with small modifications.

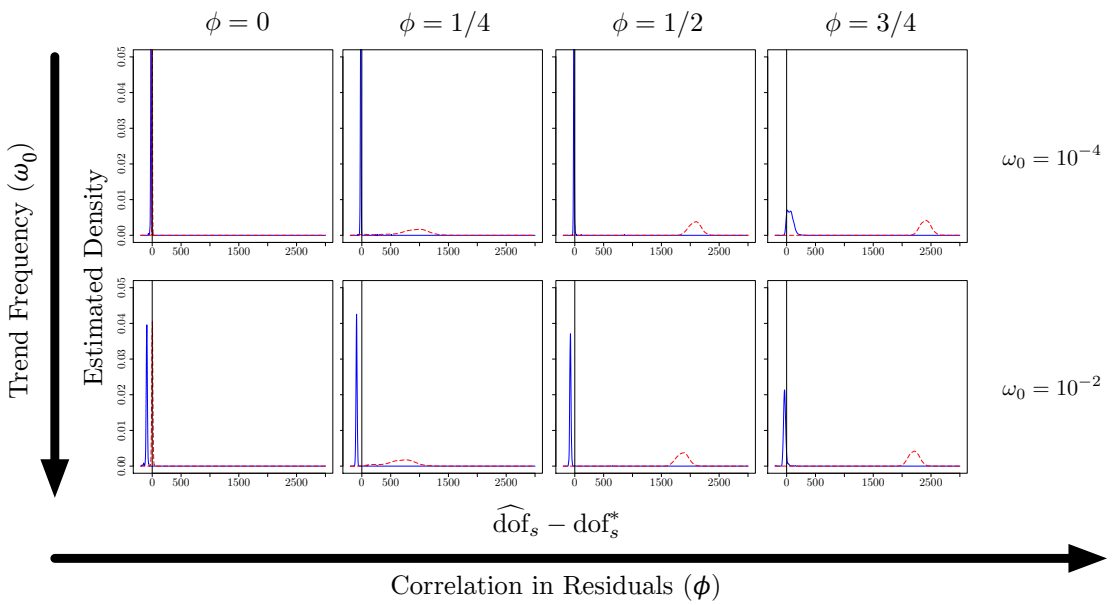


Figure 2.4: The distribution of biases $\widehat{\text{dof}}_s - \text{dof}_s^*$ between the smoothing parameter chosen GCV (dashed red) or CRR (solid blue) and the optimal value for the realization $Y_{s,t}$. A bias of zero (denoted by the black vertical line) indicates that the method performed as well as the best regression curve in the class of all smoothing splines.

Next, we examine the trends inferred for example low- and high-frequency trends as we vary the correlation in the residuals, shown in Figure 2.5. The top panels correspond to a low frequency trend, and the bottom panels correspond to a high frequency trend. As we move from left to right in the figure, the correlation in the residuals increases from zero to 0.5. As we saw from considering the smoothing parameter bias, the trend inferred using generalized cross-validation (red) under-smooths as we increase the correlation in the residuals, while the trend inferred using complexity regularized regression (blue) tends to track the true trend (grey) well, even for large values of correlation. We have also included the trend inferred using the Davies and Kovac run method (green) using the default tuning parameter

values in the `ftnonpar` package for R.

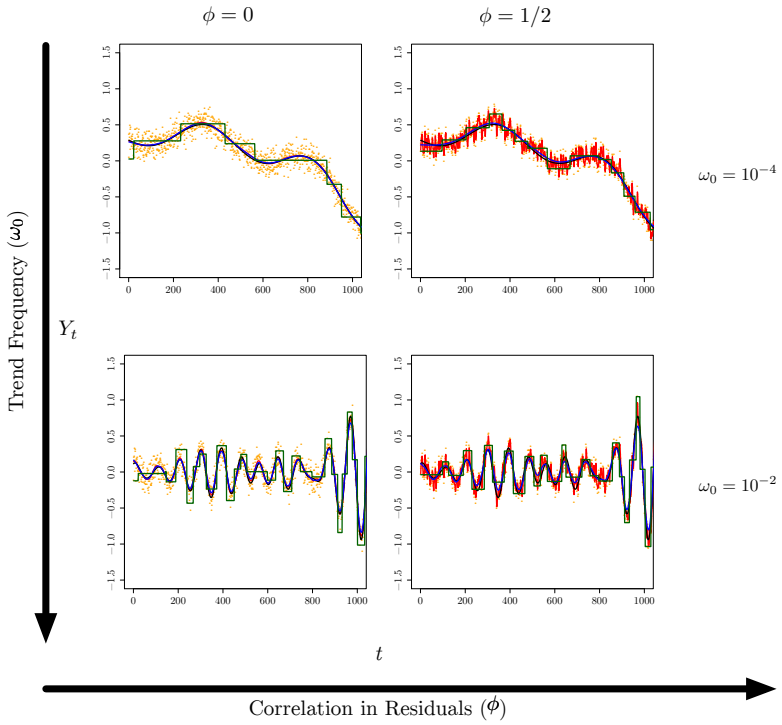


Figure 2.5: The true regression curve (black) and the estimates via GCV (red), CRR (blue) and Davies and Kovac's run method (green); for example, low (**top**) and high (**bottom**) frequency realizations from (2.37). Note that for all values of ϕ and ω_0, the CRR curves (blue) are in good agreement with the true regression curve, while GCV (red) shows good agreement only for uncorrelated residuals ($\phi = 0$), and Davies and Kovac's run method (green) differs substantially from the true regression curve in all cases.

Finally, we quantify the performance of each of the data-driven methods using the mean-squared error (2.42) between the inferred trend and the true trend. We computed the mean-squared error for each of the 1000 realizations across the frequency and residual conditions. These results are summarized in Figure 2.6, which shows the distribution of the mean-squared errors for each condition. We see that GCV performs extremely well when the residuals are uncorrelated. This is unsurprising, since GCV approximates leave-one-out cross-validation, and for un-

correlated residuals, leave-one-out cross-validation is a nearly unbiased estimator for the mean-squared error [19].Thus, for this case, using GCV to choose the degrees of freedom will perform about as well as we can with smoothing splines. As we increase the correlation, however, we see a robustness in the performance of CRR that GCV does not share. In particular, as the residuals become more correlated, CRR maintains a low mean-squared error, while the mean-squared error for GCV increases with increasing correlation.

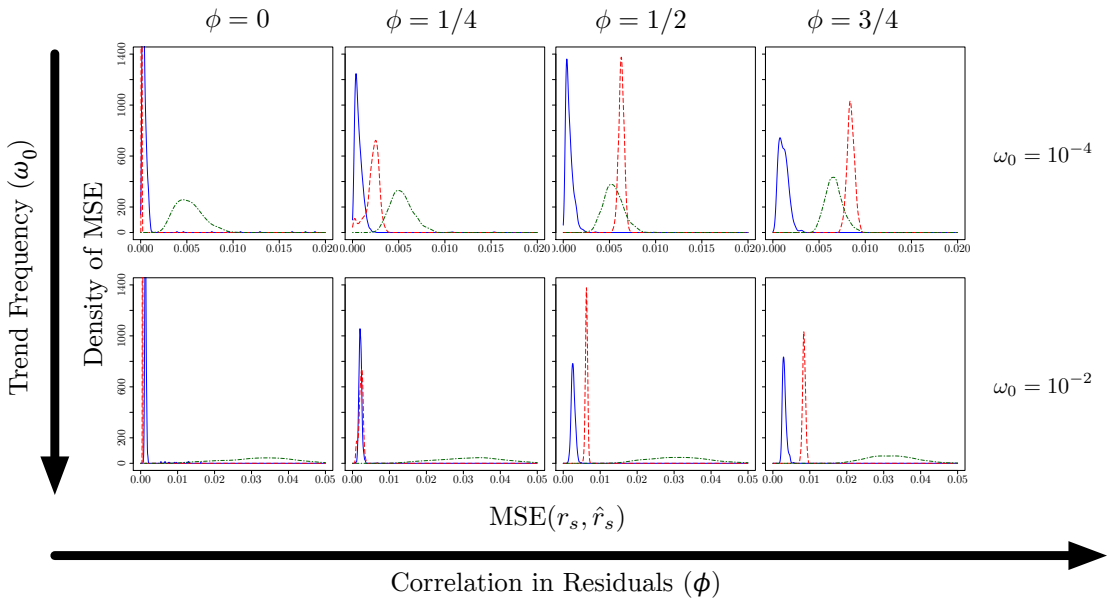


Figure 2.6: The distribution of the mean squared errors (2.42) using CRR (blue solid), GCV (red dashed) and Davies and Kovac's run method (green dot-dash) for the white noise and AR(1) residuals with $\phi \in \{0, 0.25, 0.5, 0.75\}$ for the low frequency (**top**) and high frequency (**bottom**) trends.

2.4 Financial Time Series

2.4.1 Modern Practices in Econometrics for Trend Stationary Time Series

In the study of time series occurring in macroeconomics, a common approach to analyzing systems of interest involves removing a (presumably) deterministic trend from the observations and then treating the residuals as realizations from a stationary stochastic process [51]. This can either be done in the time-domain, in the state-domain or in a mixture of the two. For time-domain smoothing, one of the most commonly-used tools for detrending data is the Hodrick–Prescott filter [52], which is essentially a special case of the smoothing spline [53]. In their original formulation, Hodrick and Prescott presented a heuristic choice of the smoothing parameter for quarterly data (such as the U.S. gross domestic product). Several authors have addressed how the choice of the smoothing parameter impacts the correlation structure of the residuals [54–56]. Data-driven approaches for choosing the smoothing parameter should be pursued, but as others have discussed [53], and we have demonstrated with our simulation study, care must be taken in the assumptions implicit to the chosen method. Other popular approaches include autoregressive models with a conditional mean that changes linearly in time [57]. It should be noted that the definition of a "trend" in econometric time series has remained open, even according to one of the leading researchers in the field [58]. Therefore, care must be taken in interpreting the results from an application of a

method like CRR to such time series. In this spirit, we frame our study here as a worked example with real data, rather than a definitive statement about any "true" trends present in these time series.

We next apply complexity-regularized regression to a particular econometric time series: the closing prices of the Dow Jones Industrial Average from January 2, 1930, to December 31, 2009. This corresponds to 80 years of the market's activity and covers 20,093 trading days. We divide the data into four double-decade periods (1930 to 1949, 1950 to 1969, 1970 to 1989, 1990 to 2009) and investigate how both the large timescale and intraday dynamics of the market have changed over these periods. We follow the same procedure for choosing the smoothing parameter as in the simulation experiments.

2.4.2 Macroscale Dynamics of the Market

Diagnostic plots for the topological complexity C_0 as a function of the degrees of freedom are shown in Figure 2.7. As before, we see that the generalized cross-validation procedure allows for many more degrees of freedom compared to the complexity regularization procedure. These diagnostic plots exhibit a property that did not occur in the simulation experiments: the minimizer (2.35) sometimes occurs at an isolated point that does not correspond to a "stable" location in the landscape of inferred states. For example, for the diagnostic plot for the 1930 to 1949 period, we see that the minimizer (2.35) occurs at an isolated point at one degree of freedom. Similarly, the minimizer for the 1970 to 1989 period occurs at an isolated point at 81

degrees of freedom. These isolated minimizers represent fragile ϵ-machines that do not persist with small perturbations in the trend. Because of this, we have modified the operationalization to choose (2.35), such that it corresponds to the smallest value λ that belongs to an "island" of some width in degrees of freedom. For this study, we have set the island length to two. We note that applying this modification to the operationalization does not alter the results from our simulation study.

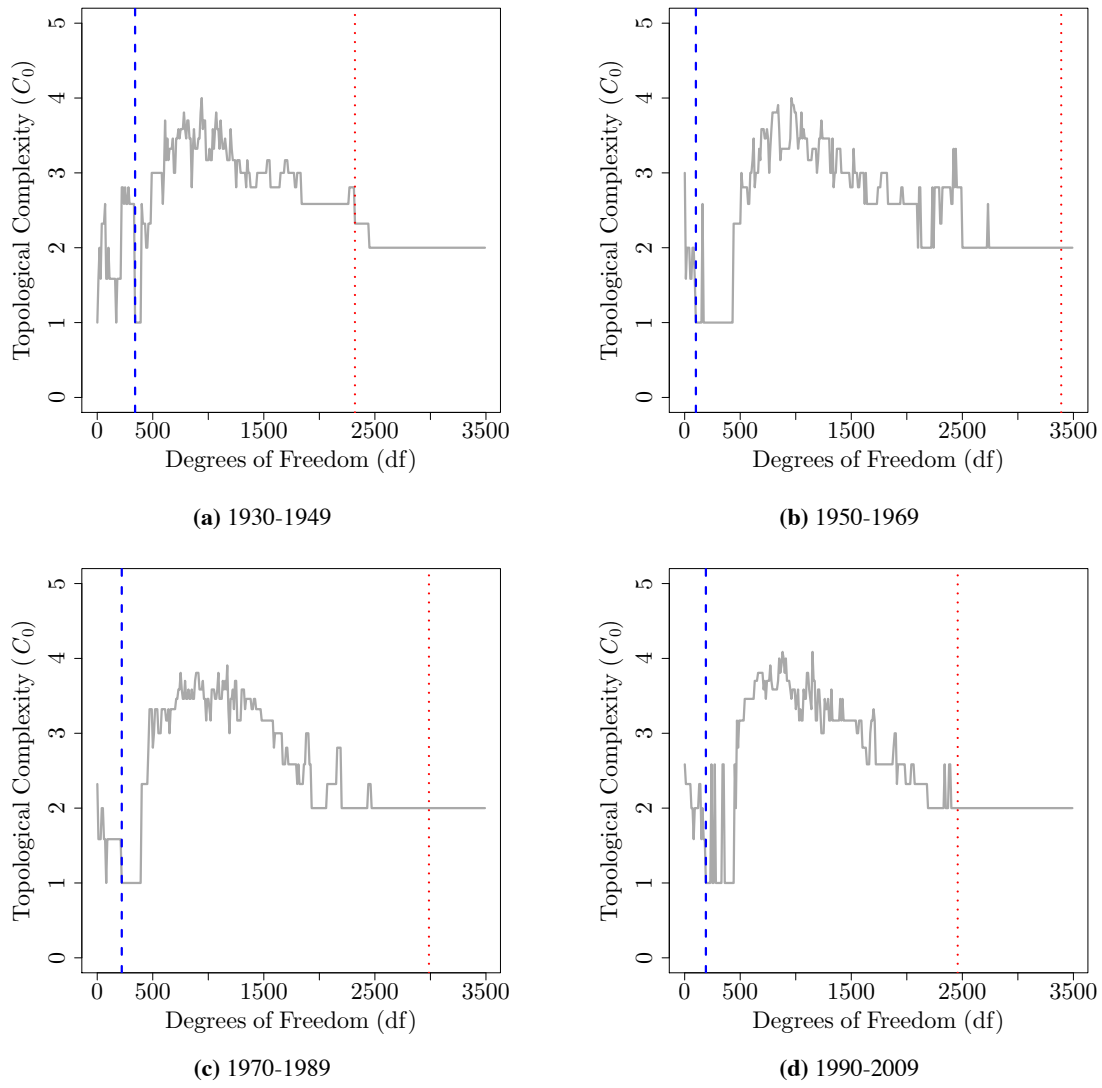
The trends for each double-decade period are shown in Figure 2.8. To characterize the overall state of the market in each double-decade period, we next compute the average curvature of the trend,

$$\psi_T(r) = \frac{1}{T} \int_0^T (r''(t))^2 \, dt. \tag{2.45}$$

The average curvature $\psi_T(r)$ captures how quickly the market changes direction in its large-scale dynamics. This value, in addition to the number of trading days and the estimated degrees of freedom of the trend, are reported in Table 2.1 for each double-decade period.

Table 2.1 demonstrates an important difference between the degrees of freedom and the average curvature: they capture two different senses of smoothness. In particular, the average curvature is scale-dependent. We see an instance of this with the trend from 1990 to 2009, which has a much larger average curvature than the other trends. By inspection of Figure 2.8, we see that this is because, during this double decade, the magnitude of the price of the market greatly increased. Thus, the "acceleration" of the trend during this time period has become larger, resulting

in a larger average curvature. However, if we consider the degrees of freedom over time, which capture a scale-independent sense of the complexity of r, we see that long-term trend of the market exhibited greater complexity between 1930 and 1949 than during any of the other double-decade periods. We also note that the low number of degrees of freedom for 1950 to 1969 is most likely artificial: by inspection of Figure 2.7, we see that the optimal value occurs in a small island, and a value in the larger island around 200 might be more appropriate. This motivates considering the island length as a possible tuning parameter that may be set by the investigator.



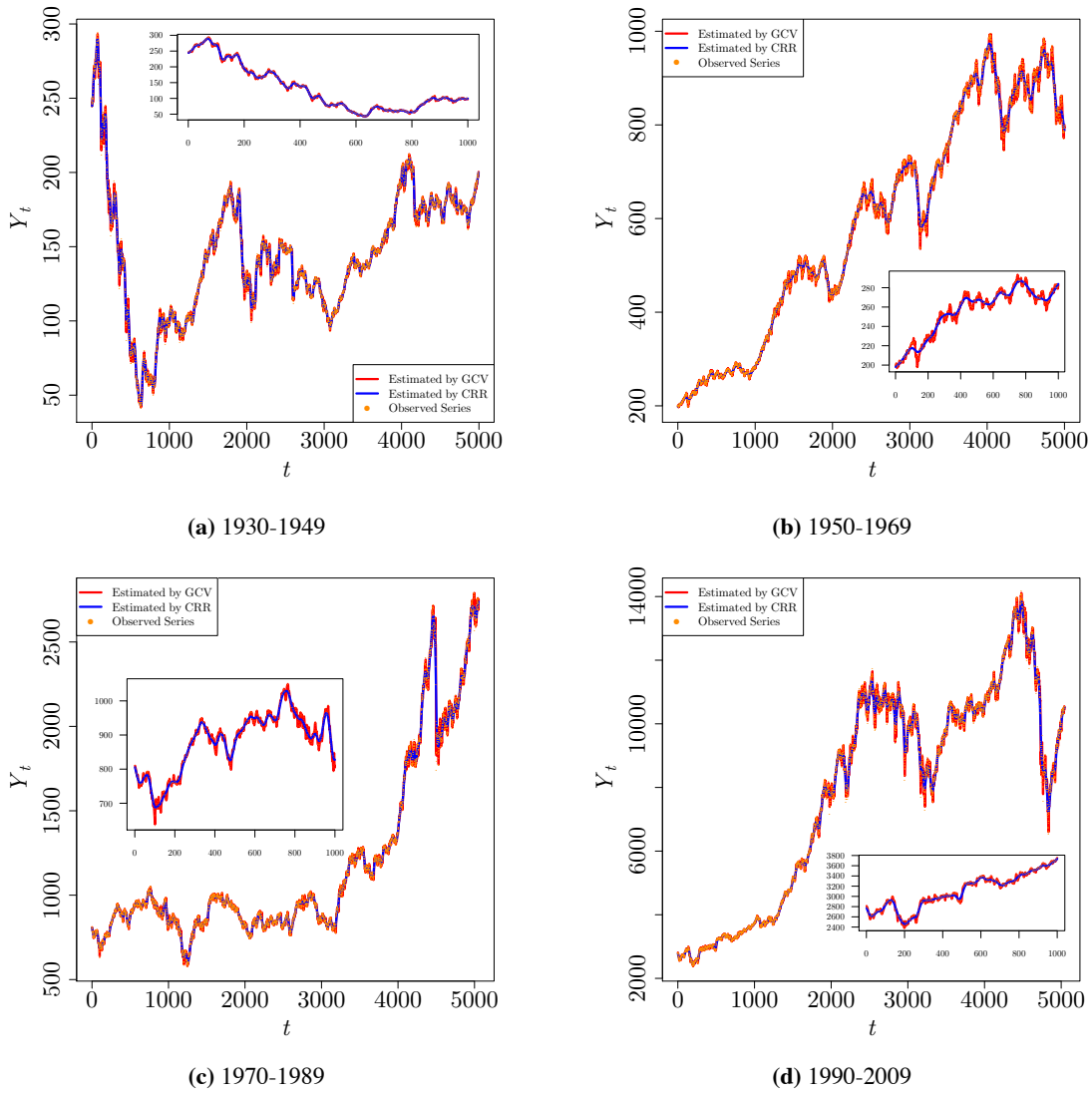
(a) 1930-1949

(b) 1950-1969

(c) 1970-1989

(d) 1990-2009

Figure 2.7: The topological complexity C_0 as a function of the degrees of freedom of the smoothing spline for each double-decade period. The vertical red and blue lines indicate the degrees of freedom chosen by GCV and CRR.



(a) 1930-1949

(b) 1950-1969

(c) 1970-1989

(d) 1990-2009

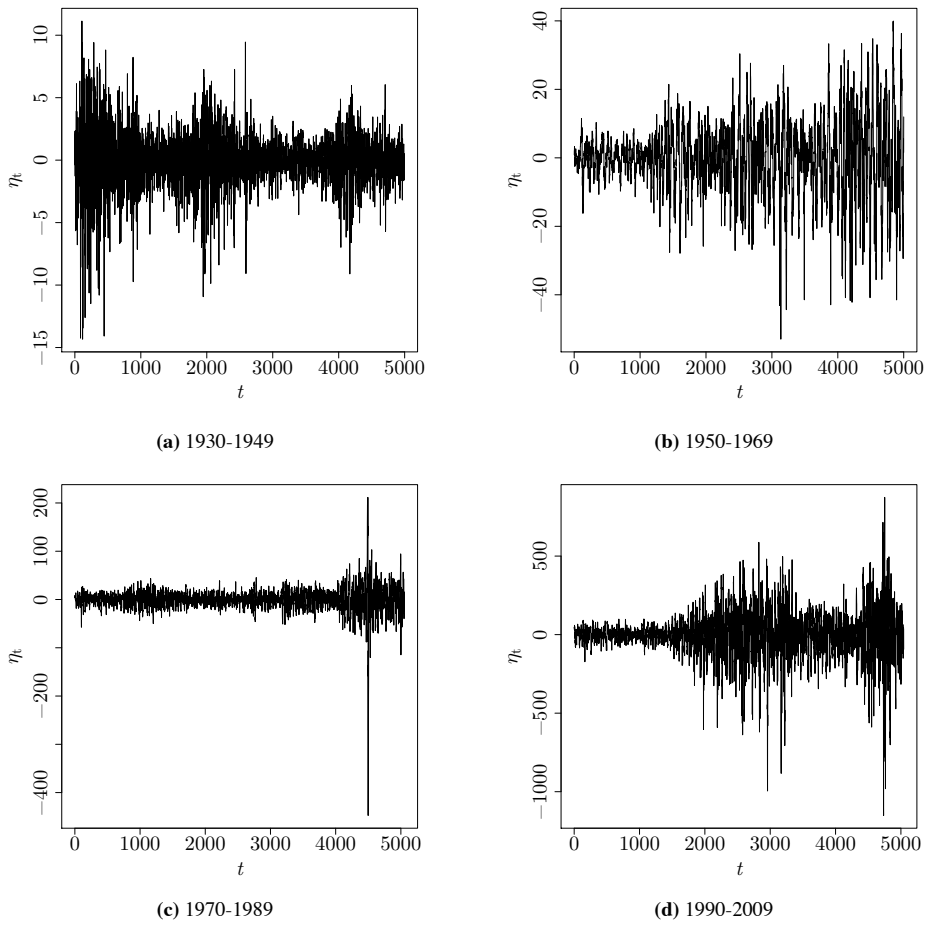
Figure 2.8: The inferred trends using CRR (blue) and GCV (red) for the DJIA time series for the double-decade periods from 1930 to 2009. The insets demonstrate the trend for the first 1000 trading days in each double-decade period, to highlight the short-term fluctuations about the long-term trend.

Table 2.1: The number of trading days (T), CRR degrees of freedom ($\widehat{\text{dof}}_{\text{CRR}}$) and average curvature of the trend ($\psi_T(\hat{r})$) for the four double-decade periods from 1930 to 2009.

Time Period	T	$\widehat{\text{dof}}_{\text{CRR}}$	$\psi_T(\hat{r})$
1930–1949	4996	341	0.002727
1950–1969	5000	101	0.000361
1970–1989	5054	221	0.033354
1990–2009	5043	191	0.897089

2.4.3 Microscale Dynamics of the Market and the Associated Causal State Models

Previous work has considered the microscale dynamics of various markets using tools from computational mechanics. The authors in [1] used inter-day data from the Standard & Poor's 500 index to construct causal state models. The authors in [2] constructed causal state models using high-frequency, single minute resolution data from the Standard & Poor's 500 index, the Korean Stock Exchange (KOSPI) and the Nikkei index. Both papers used first-order differencing of either the price or log-price to detrend the time series before binarizing. Use of first-order differencing is closely related to an assumption that the trend in the time series can be approximated by a linear function, at least locally. This is equivalent to using a non-parametric regression method with a very small amount of smoothing, as we have seen occurs when using data-driven methods with correlated residuals. First-order differencing is also related to the Box–Jenkins approach to modeling time series, where higher-order differences of a time series are treated as realizations from a stationary autoregressive moving average (ARMA) model [17].

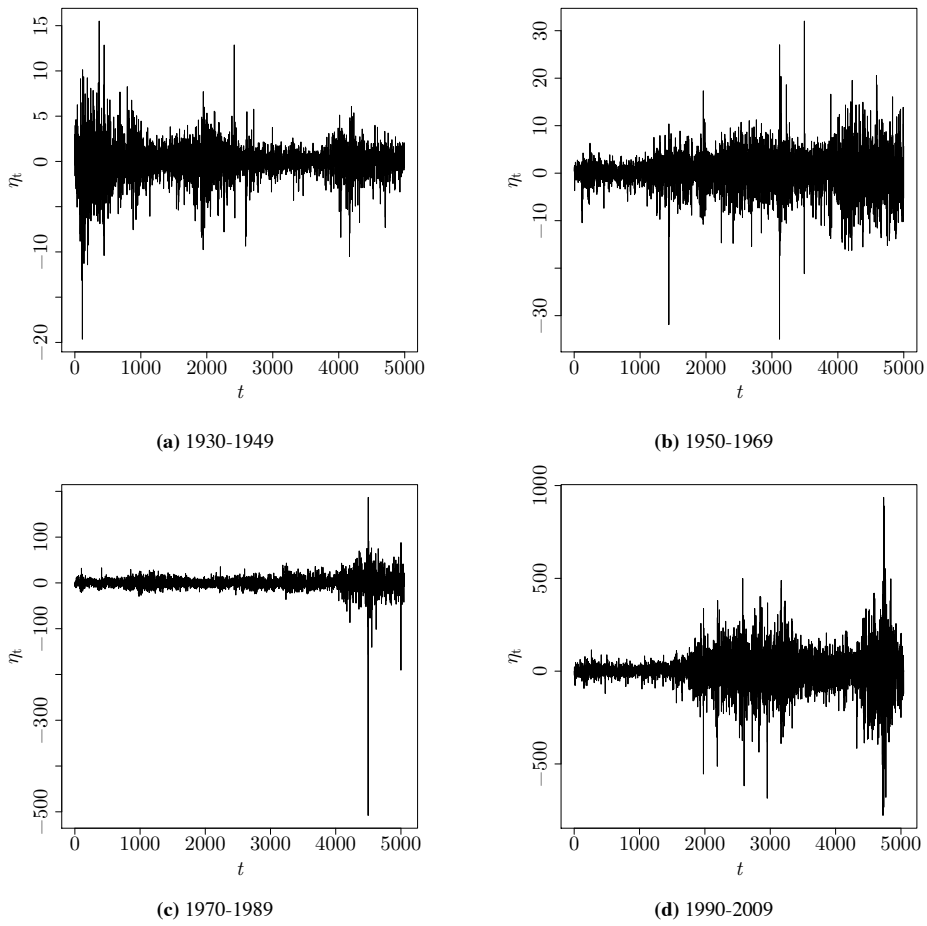


(a) 1930-1949 **(b)** 1950-1969

(c) 1970-1989 **(d)** 1990-2009

Figure 2.9: The residuals $\hat{\eta}_t$ inferred using CRR for each double-decade period. Note that the residuals exhibit strong non-stationarity after detrending via CRR.

We also consider the computational structure of the microscale dynamics, but make no assumption on the trend being locally linear. Instead, we consider the residuals $\{\hat{\eta}_t\}_{t=1}^{T}$ inferred from the CRR-based smoothing. These residuals for each double-decade period are shown in Figure 2.9. We see that the residual series, despite the detrending, are non-stationary: for example, the point-wise variance clearly changes over time. The same is true if we perform the detrending using first-order differences, as shown in Figure 2.10.



(a) 1930-1949

(b) 1950-1969

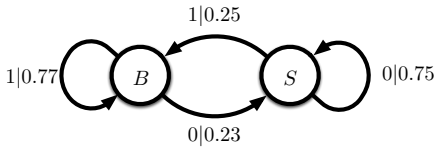
(c) 1970-1989

(d) 1990-2009

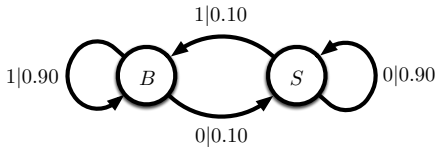
Figure 2.10: The residuals $\hat{\eta}_t$ computed using first-order differencing for each double-decade period, similar to the methods used in [1, 2]. Note that the residuals exhibit strong non-stationarity, even after differencing.

Next, we construct causal state models using the binarized residuals $\{B_t\}_{t=1}^{T}$. These causal state models are equivalent to those constructed in the smoothing parameter selection step of complexity-regularized regression, and we use CSSR with the same parameter values $\alpha = 0.001$ and $L_{\max} = 5$. The causal state models for the double-decade periods are shown in Figure 2.11. Each node corresponds to a causal state (an equivalence class over pasts), and each directed edge corresponds to an allowed transition out of that state, annotated with $b \mid p$, where b is the symbol

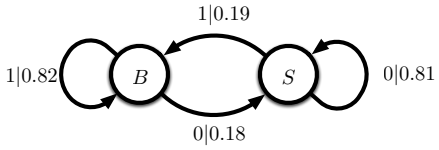
emitted (either zero when below the trend or one when above the trend), and p is the probability of emitting that symbol, given the current causal state. We see that all four decades are characterized by the same two-state causal state model, with differing transition probabilities. The first state (B) represents when the market tends to remain above the prevailing trend, and a second state (S) represents when the market tends to remain below the prevailing trend. Interestingly, the causal state models for 1970 to 1989 and 1990 to 2009 are very similar, with only minor differences in the probabilities associated with the transitions out of state S.



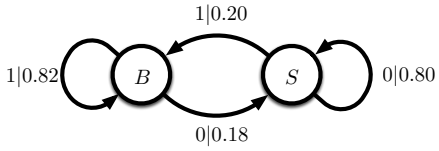
(a) 1930 to 1949: $C_\mu = 1.0$ bit, $h_\mu = 0.79$ bits per symbol



(b) 1950 to 1969: $C_\mu = 1.0$ bit, $h_\mu = 0.48$ bits per symbol



(c) 1970 to 1989: $C_\mu = 1.0$ bit, $h_\mu = 0.68$ bits per symbol



(d) 1990 to 2009: $C_\mu = 1.0$ bit, $h_\mu = 0.71$ bits per symbol

Figure 2.11: The causal state models associated the binarized residuals B_t after removing the inferred trend $\hat{r}_t$ for each double-decade period. Note that the overall structure of the causal state models remain fixed while the transition probabilities change from time period to time period.

The statistical complexity C_μ and the entropy rate h_μ of the binarized residuals are reported in Table 2.2. As described previously, the statistical complexity characterizes the amount of memory in a stochastic process, in the sense that it quantifies the number of bits of the past necessary to optimally predict the future.

The entropy rate characterizes the intrinsic randomness in the process [44], in the sense that the entropy rate quantifies the uncertainty in the future of the process after accounting for its entire past. Together, the entropy rate and statistical complexity give a picture of the predictability of the process and the computational overhead necessary to optimally perform the prediction. We see that for all four double-decade periods, the statistical complexity is one. That is, in order to predict whether or not the market will be above or below the prevailing trend the next day, we need only know whether the market is above or below the prevailing trend on the current day. This memory does not change from double-decade period to double-decade period. However, the entropy rate does differ, indicating that despite the similar memory, the intrinsic randomness of day-to-day fluctuations has changed over time.

Table 2.2: The statistical complexities C_μ and entropy rates h_μ for the causal state models inferred from the binarized residuals for each double-decade period.

Time Period	C_μ (bits)	h_μ (bits per symbol)
1930–1949	1.0	0.79
1950–1969	1.0	0.48
1970–1989	1.0	0.68
1990–2009	1.0	0.71

2.5 Discussion and Future Work

When performing data-driven non-parametric regression, choosing the appropriate tuning parameter to learn from the data is paramount. Oversmoothing the data will miss out on important details, while undersmoothing the data

will pick up spurious structure introduced by noise. We have proposed a method for choosing this tuning parameter in the situation where the residuals are correlated. We have seen that complexity-regularized regression outperforms generalized cross-validation, a popular data-driven approach, in the correlated case. Moreover, complexity-regularized regression does so with no assumptions on the properties of the residual process other than stationarity and short memory. Thus, our approach presents a non-parametric alternative to more standard methods [59], which assume a parametric form for the residual process. In addition, we have seen that complexity-regularized regression outperforms the original runs-based method of Davies *et al.*, in the case of correlated residuals, while still maintaining the spirit of model-free regression.

To apply our method, we have removed a great deal of information from the residuals by only considering their signs in constructing a causal state model. A similar loss of information occurs when using, for instance, the Wald–Wolfowitz runs test. Keeping the magnitudes, as well as the signs, of the residual series should give a more accurate representation of its "randomness". Recent work has extended the techniques of computational mechanics to continuous-valued, discrete-time stochastic processes [60] without the need to introduce a (somewhat arbitrary) discretization. This new formalism also associates a statistical complexity with any continuous-valued time series, with a similar interpretation in terms of the amount of past necessary to predict the future of a time series. The statistical complexity of the continuous-valued residuals would incorporate more information about the residuals and might improve the complexity regularization formalism.

We have seen from the DJIA example that, even after removing the most prominent trend in the data, non-trivial non-stationarities in the residuals remain. This is most likely due to the multi-scale nature of the time series, which presumably contains shorter timescale weekly and monthly seasonalities in addition to the longer timescale overall trend. To account for these non-stationarities, we could iteratively use complexity-regularized regression to generate a family of trends at different timescales. For example, we might consider the trend inferred using the methodology as representing the lowest frequency components of the trend. We could then treat the inferred residuals as a fresh input to complexity-regularized regression and estimate a higher-frequency trend in the residuals. We could continue in this manner until the residuals appear "random enough" by some criterion.

2.6 Conclusion

A new method for nonparametric regression has been proposed to handle the case of serially-correlated residuals, as commonly occurs in time series analysis. The method is "model-free," in the sense presented in [40], *i.e.*, we assume no model for the residuals and, instead, infer a regression curve to force the residuals to satisfy some criterion of randomness. The algorithm works by employing standard nonparametric regression estimators and choosing their smoothing parameter, so as to make the residuals look random, in that the statistical complexity of the binarized residuals is minimized. The approach was found to outperform GCV when the residuals are correlated.

We have applied complexity-regularized regression to analyzing the day-to-day price associated with the Dow Jones Industrial Average from 1930 to 2009. Our approach allows us to recover both long-term trends in the market and short-term behavior.

Chapter 3: Predictability of User Behavior in Social Media: Bottom-Up v. Top-Down Modeling

3.1 Introduction

At the most abstract level, an individual using a social media service may be viewed as a computational agent [61]. The user receives inputs from their surroundings, combines those inputs in ways dependent on their own internal states, and produces an observed behavior or output. In the context of a microblogging platform such as Twitter, the inputs may be streams from other Twitter users, real world events, etc., and the observed behavior may be a tweet, mention, or retweet. From this computational perspective, the observed behavior of the user should give some indication of the *types* of computations the user is doing, and as a result, an insight into viable behavioral models of that user on social media. Large amounts of observational data are key to this type of study. Social media has made such behavioral data available from massive numbers of people at a very fine temporal resolution.

As a first approximation to the computation performed by a user, we might consider only the user's own past behavior as possible inputs to determine their

future behavior. From this perspective, the behavior of the user can be viewed as a point process with memory, where the only observations are the time points when social interactions occurred [62]. Such point process models, while very simple, have found great success in describing complicated dynamics in neural systems [63], and have recently been applied to social systems [64, 65].

We propose extending this previous work by explicitly studying the *predictive* capability of the point process models. That is, given observed behavior for the user, we seek a model that not only captures the dynamics of the user, but also is useful for predicting the future behavior of the user, given their past behavior. The rationale behind this approach is that if we are able to construct models that both reproduce the observed behavior and successfully predict future behavior, the models capture something about the computational aspects, in the sense outlined above, of the user.

Since in practice we never have access to all of a user's inputs, nor to their internal states, we cannot hope to construct a 'true' model of a user's behavior. Instead, we construct approximate models. In particular, we consider two classes of approximate models: causal state models and echo state networks.

The causal state modeling approach, motivated by results from computational mechanics, assumes that every individual can initially be modeled as a biased coin, and then adds structure as necessary to capture patterns in the data. It does this by expanding the number of states necessary to represent the underlying behavior of the agent. Causal state models have been used successfully in a number of different domains, including elucidating the computational structure of neural spike

trains [12], uncovering topic correlations in social media [66], and improving named entity recognition in natural language processing [67]. As opposed to the simple-to-complex approach used by causal state modeling, echo state networks start by assuming that agent behavior is the result of a complex set of internal states with intricate relationships to the output variables of interest, and then simplifies the weights on the relationships between the internal states and the output variables over time. Echo state networks have proven useful in a number of different domains including wireless networking [68], motor control [69], and grammar learning [70].

Our motivation for considering these two models was twofold. First, they share a structural similarity in that they both utilize hidden states that influence behavior and incorporate past data when making future decisions. Second, they approach modeling from two different perspectives. As mentioned, both representations have a notion of internal state, and the observation of past behavior moves the agent through the possible states. It is the model of these dynamics through the states that makes it possible to use these methods to predict an individual's behavior. Moreover, whereas computational mechanics seeks to construct the simplest model with the maximal predictive capability, echo state networks relax down from very complicated dynamics until predictive ability is reached. Due to this difference, we hypothesize that there are some users that will be easier to predict using a causal state modeling approach, and a different set of users that will be easier to predict using an echo state network approach.

In the rest of this chapter, we explore this hypothesis. We begin by describing the two approaches we used and their relevant literature. After this, we describe

the data used to test the predictive ability of these methods, and the investigations that we carried out to evaluate this ability. Finally, we conclude with limitations of the present work and future avenues of research.

3.2 Methodology

3.2.1 Notation

For each user, we consider only the relative times of their tweets with respect to a reference time. Denote these times by $\{\tau_j\}_{j=1}^{n}$. Let the reference start time be t_0 and the coarsening amount be Δt. From the tweet times, we can generate a binary time series $\{X_i\}_{i=1}^{T}$, where

$$X_i = \begin{cases} 1 & : \exists \tau_j \in [t_0 + (i-1)\Delta t, t_0 + i\Delta t) \\ 0 & : \text{ otherwise} \end{cases} . \tag{3.1}$$

In words, X_i is 1 if the user tweeted at least once in the time interval $[t_0 + (i - 1)\Delta t, t_0 + i\Delta t)$, and 0 otherwise. Because the recorded time of tweets is restricted to a 1-second resolution, a natural choice for Δt is 1 second. However, due to limitations in the amount of data available we will coarsen the time series to less than this resolution. Thus, in this chapter, we consider the behavior of the user as a point process, only considering the timing of the tweets, and discarding any informational content *in* the tweet (sentiment, retweet, mention, etc.).

Once we have the user's behavior encoded in the sequence $\{X_i\}_{i=1}^{T}$, we wish to perform one-step ahead prediction based on the past behavior of the user. That

is, for a time bin $[t_0 + (i-1)\Delta t, t_0 + i\Delta t)$ indexed by i, we wish to predict X_i given a finite history $X_{i-L}^{i-1} = (X_{i-L}, \ldots, X_{i-2}, X_{i-1})$ of length L. This amounts to a problem in autoregression, where we seek a function r from finite pasts to one-step ahead futures such that we predict X_i using

$$\hat{X}_i = \underset{x_i \in \{0,1\}}{\arg\max}\, r(x_i; x_{i-L}^{i-1}). \tag{3.2}$$

If we assume that $\{X_i\}_{i=1}^T$ was generated by a stochastic process, the optimal choice of r would be the conditional distribution

$$r(x_i; x_{i-L}^{i-1}) = P(X_i = x_i | X_{i-L}^{i-1} = x_{i-L}^{i-1}), \tag{3.3}$$

and the optimal prediction would be the x_i that maximizes this conditional probability. If we further assume that $\{X_i\}_{i=1}^T$ is a conditionally stationary stochastic process [71], the prediction function simplifies to

$$r(x_i; x_{i-L}^{i-1}) = P(X_L = x_i | X_0^{L-1} = x_{i-L}^{i-1}), \tag{3.4}$$

independent of the time index i.

Because in practice we do not have the conditional distribution available, we consider two approaches to inferring the prediction function r: one from computational mechanics [4] and the other from reservoir computing [72], specifically the echo state network [73]. These two methods for inferring r differ dramatically in their implementations. Computational mechanics seeks to infer the simplest model

that will capture the data generating process, while echo state networks generate a complex set of oscillations and attempt to find some combination of these that will map to the desired output.

3.2.2 Computational Mechanics

Computational mechanics proceeds from a state-space representation of the observed dynamics, with hidden states $\{S_i\}_{i=1}^{T}$ determining the dynamics of the observed behavior $\{X_i\}_{i=1}^{T}$. The hidden state S_i for a process, called the causal or predictive state, is the label corresponding to set of all pasts that have the same predictive distribution as the observed past x_i. We call the mapping from pasts to labels ϵ. Two pasts x and x' have the same label $s_i = \epsilon(x) = \epsilon(x')$ if and only if

$$P(X_i^{\infty}|X_{-\infty}^{i-1} = x) = P(X_i^{\infty}|X_{-\infty}^{i-1} = x') \tag{3.5}$$

as probability mass functions. That is, the two pasts x and x' give statistically equivalent predictions over all possible futures x_i^{∞}. Now, instead of considering $P(X_i^{\infty}|X_{i-L}^{i-1} = x_{-\infty}^{i-1})$, we consider the label for the past $s_i = \epsilon(x_{-\infty}^{i-1})$, and use $P(X_i^{\infty}|S_i = s_i)$. We then proceed with the prediction problem outlined above. The state S_i (or equivalently the mapping ϵ) is the unique minimally sufficient predictive statistic of the past for the future of the process. Because the hidden states $\{S_i\}_{i=1}^{T}$ can be thought of as generating the observed behaviors $\{X_i\}_{i=1}^{T}$, they are called the *causal states* of the process. The resulting model is called an ϵ-machine (after the statistic ϵ) or a causal state model (after the causal state S).

Of course, in practice the conditional distribution $P(X_i^{\infty}|X_{-\infty}^{i-1} = x)$ is not known, and must be inferred from the data. Beyond the advantage of computational mechanics's state-space representation as a minimally sufficient predictive statistic, it also admits a way to infer the mapping ϵ directly from data. We will infer the model using the Causal State Splitting Reconstruction (CSSR) algorithm [46]. As the name CSSR implies, the estimate $\hat{\epsilon}$ is inferred by splitting states until a stopping criterion is met. The algorithm begins with a null model, where the data generating process is assumed to have a single causal state, corresponding to an IID process. It continues to split states (representing a finer partition of the set of all pasts) until the partition is next-step sufficient and recursively calculable. The resulting $\hat{\epsilon}$ and the estimated predictive distributions $\hat{P}(X_i|S_i = \hat{\epsilon}(x_{i-L}^{i-1}))$ can then be used to estimate the prediction function, giving

$$\hat{r}_{\mathrm{cm}}(x_i; x_{i-L}^{i-1}) = \hat{P}(X_i = x_i|S_i = \hat{\epsilon}(x_{i-L}^{i-1})). \tag{3.6}$$

We will refer to the estimated $\hat{\epsilon}$ and associated predictive distributions as the *causal state model* for a user.

3.2.3 Echo State Networks

Neural networks can be divided into feed-forward and recurrent varieties. The former are easier to train but lack the capacity to build rich internal representations of temporal dynamics. In contrast, the latter are naturally suited to representing dynamic systems, but their learning algorithms are more computationally intensive

and less stable. Echo state networks attempt to resolve this conflict by using randomly selected, fixed weights to drive the recurrent activity and only training the (far simpler) output weights.

In addition to simplifying the training process, echo state networks shift the problem into a higher dimensional space [74]. This technique of dimensional expansion is commonly employed in machine learning, for instance by Support Vector Machines, Multilayer Perceptrons, and many kernel methods. A decision boundary which is nonlinear in the original problem space is often linear in higher dimensions, allowing a more efficient learning procedure to be used [75, 76].

The echo state networks we used here consists of 10 input nodes, 1 output node and a "reservoir," consisting of 128 hidden nodes, which is randomly and recurrently connected. The connection weights $\mathbf{W}$ within the reservoir as well as the weights to it from the input and output nodes ($\mathbf{W}_{\text{in}}$ and $\mathbf{W}_{\text{fb}}$, respectively) are sampled uniformly at random from the interval $[0, 1]$. $\mathbf{W}$ is also scaled such that the spectral radius $\rho(\mathbf{W}) < 1$ [77]. This scaling ensures the network will exhibit the "echo state property:" the effect of previous reservoir states and inputs will asymptotically approach zero as time passes rather than persisting indefinitely or being amplified [78]. Only the weights $\mathbf{W}_{\text{out}}$ from the reservoir to the output nodes are trained. The goal is to draw on the diverse set of behaviors within the reservoir and find some linear combination of those oscillations which match the desired output.

States of reservoir nodes $\mathbf{y}_t$ are updated according to

$$\mathbf{y}_t = \sigma(\mathbf{W}_{\text{in}}\mathbf{x}_t + \mathbf{W}\mathbf{y}_{t-1} + \mathbf{W}_{\text{fb}}z_{t-1}) \tag{3.7}$$

where $\mathbf{x}_t$ is the current network input, $\mathbf{z}_{t-1}$ is the previous network output, and σ is the logistic sigmoid function. The output of the network is determined by

$$z_t = \sigma(\mathbf{W}_{\text{out}}\,[\mathbf{x}_t|\mathbf{y}_t]) \tag{3.8}$$

where | represents a vertical concatenation.

The training procedure involves presenting the network with each input in the sequence and updating the internal reservoir. The inputs and reservoir states are collected row-wise in a matrix $\mathbf{S}$. We redefine the network's targets during training to be $z'_t = \sigma^{-1}(z_t)$ and collect them row-wise in $\mathbf{D}$. This allows us to use a standard pseudo-inverse solution to compute the output weights $\mathbf{W}_{\text{out}} = (\mathbf{S}^{-1}\mathbf{D})^{\text{T}}$ which minimizes the MSE of the network on the training output.

3.3 Data Collection and Preprocessing

The data consists of the Twitter statuses of 12,043 users over a 49 day period. The users are embedded in a 15,000 node network collected by performing a breadth-first expansion from a seed user. Once the seed user was chosen, the network was expanded to include his/her followers, only including users considered to be active (users who tweeted at least once per day over the past one hundred tweets). Network

collection continued in this fashion by considering the active followers of the active followers of the seed, etc.

The statuses of each user were transformed into a binary time series using their time stamp, as described in the Methodology section. In this chapter, only tweets made between 7 AM and 10 PM (EST) were considered. For any second during this time window, a user either tweets, or does not. Thus, each day can be considered as a binary time series of length 57,600, with a 1 at a timepoint if the user tweets, and a 0 otherwise.

Because of statistical and computational limitations, the time series were further coarsened by binning together disjoint intervals of time. We considered time windows with length equal to ten minutes ($\Delta t = 600$). Thus, we created a new time series by recording a 1 if any tweeting occurs during a ten minute window, and a 0 otherwise. Once we have the (either coarsened or not) time series, we can visualize the behavior of a user over the 49 day period by using a rastergram. A rastergram visualizes a point process over time and over trials. The horizontal axis corresponds to the time point in a particular day, and the vertical axis corresponds to the day number. At each time point, a vertical bar is either present (if the user tweeted on that day at that time) or absent. Visual inspection of rastergrams serves as a first step towards understanding the behavior of any given user. Figure 3.1 demonstrates the original and coarsened time series for two users.

The users were further filtered to include only the top 3,000 most active users over the 49 day period. A base activity measure was determined by the proportion of seconds in the 7 AM to 10 PM window the user tweeted, which we call the tweet

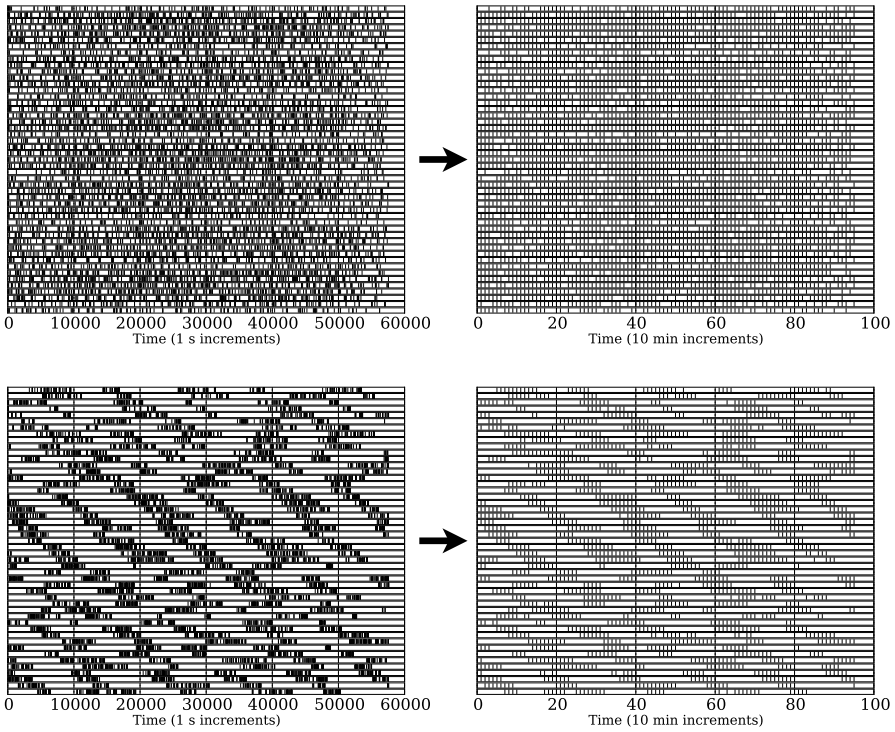


Figure 3.1: Coarsening of two users. Each row in the rastergram corresponds to a single day of activity for a fixed user. The original time series are at single second resolution, resulting in 57,600 time points in each day. After binning together activity using disjoint (partitioned) ten minute windows, there are 96 time points in each day ($T = 96$).

rate. Of the top 3,000 users, these tweet rates ranged from 0.38 to 8.5×10^{-5}. 90% of the top 3,000 users had a tweet rate below 0.05. The distribution of the tweet rates amongst the top 3,000 users is shown in Figure 3.2.

3.4 Results and Discussion

3.4.1 Testing Procedure

The 49 days of user activity were partitioned, chronologically, into a 45 day training set and a 4 day testing set. This partition was chosen to account for possible

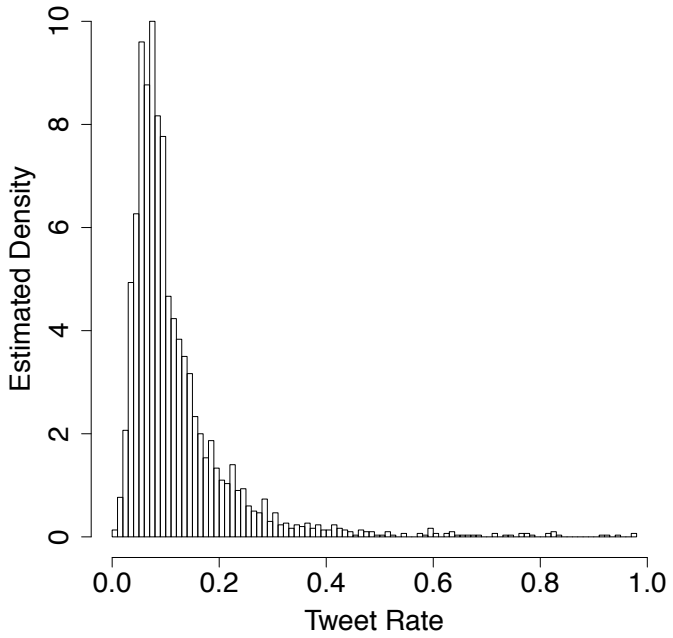


Figure 3.2: The observed distribution of the fraction of time spent tweeting (tweet rate) over the 49 day period for all of the users. 90% of the 3,000 users had a tweet rate below 0.05.

changes in user behavior over time, which would not be captured by using a shuffling of the days. Thus, for each user, the training set consists of 4,320 timepoints, and the testing set consists of 384 timepoints.

The only parameter for the causal state model is the history length L to use. This was treated as a tuning parameter, and the optimal value to use was determined by using 9-fold cross-validation on the training set. The maximal history length $L_{\max}$ that can be used and still ensure consistent estimation of the joint distribution is dependent on the number of time points n, and is bounded by

$$L_{\max} < \frac{\log_2 n}{h + \epsilon}, \tag{3.9}$$

where h is the entropy rate of the stochastic process and ϵ is some positive con-

stant [50]. Thus, because $0 \leq h \leq 1$ for a stationary stochastic process with two symbols, as a practical bound, we take

$$L_{\max} < \log_2 n.$$

For this data set, the bound requires that $L_{\max} < 12$. Thus, we use the 9-fold cross-validation to reconstruct causal state models using histories of length 1 through 11, and then choose the history length that maximizes the average accuracy rate taken over all of the folds.

Experiments showed that the echo state network was robust to varying parameter choices as long as the echo state property is achieved [79, 80]. As a result all networks were created with $\rho(\mathbf{W}) = 0.99$ and $L_{\text{ESN}} = 10$.

3.4.2 Comparison to Baseline

In all cases, we compute the accuracy rate of a predictor using zero-one loss. That is, for a given user, we predict the time series $X_1, \ldots, X_{n_{\text{test}}}$ as $\hat{X}_1, \ldots, \hat{X}_{n_{\text{test}}}$ and then compute

$$\text{Accuracy Rate} = \frac{1}{n_{\text{test}}} \sum_{i=1}^{n_{\text{test}}} \mathbb{1}[\hat{X}_i = X_i]. \tag{3.10}$$

We compared the accuracy rates on the causal state model and echo state network to a baseline accuracy rate for each user. The baseline predictor was taken to be the majority vote of tweet vs. not-tweet behavior over the training days,

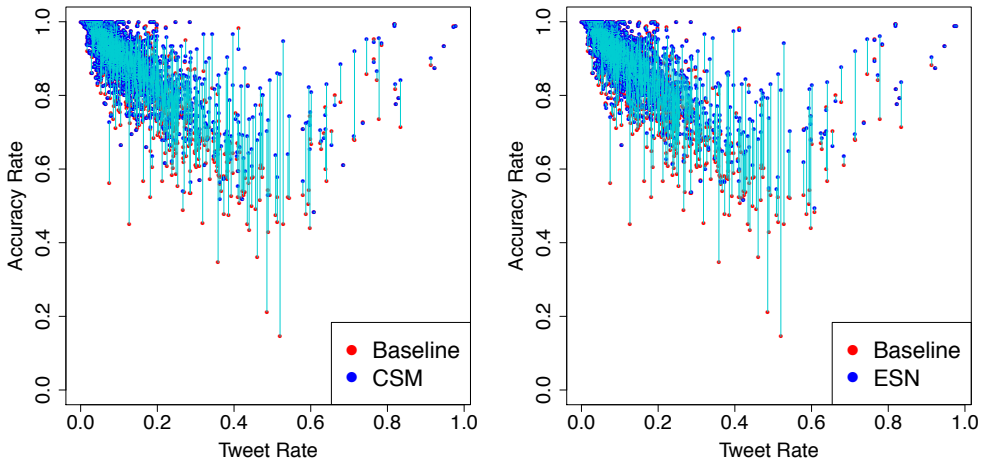
regardless of the user's past behavior. That is, for the baseline predictor we take

$$\hat{X}_i = \begin{cases} 0 & : \hat{p} \leq \frac{1}{2} \\ \\ 1 & : \hat{p} > \frac{1}{2} \end{cases}, \tag{3.11}$$

where $\hat{p} = \frac{1}{n_{\text{train}}} \sum_{j=1}^{n_{\text{train}}} X_j$. This is the optimal predictor for a Bernoulli process where the $\{X_i\}$ are independent and identically distributed Bernoulli random variables with parameter p. In the context of our data, for users that usually tweeted in the training set, the baseline predictor will always predict that the user tweets, and for users that usually did not tweet in the training set, the baseline predictor will always predict the user does not tweet. For any process with memory, as we would expect from most Twitter users, a predictor should be able to outperform this base rate.

The comparison between the baseline predictor and the casual state model and echo state network predictors are shown in Figure 3.3. In both plots, each red point corresponds to the baseline rate on the testing set for a given user, and the blue point corresponds to the accuracy rate on the testing set using one of the two models. Here, the tweet rate is computed in terms of the coarsened time series. That is, the tweet rate is the proportion of ten minute windows over the 49 day period which contain one or more tweet. Clearly, the model predictions show improvement over the baseline prediction, especially for those users with a tweet rate above 0.2.

To make this more clear, the improvement as a function of the tweet rate of each user is shown in Figure 3.4 for both methods. Breaking the users into



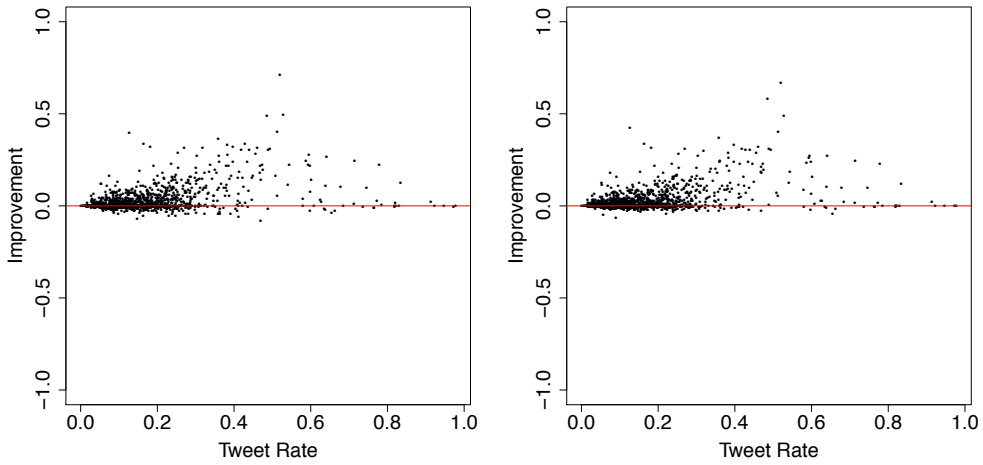
(a) Causal State Model (b) Echo State Network

Figure 3.3: The improvement over the baseline accuracy rate for the casual state model and echo state network. In both plots, each red point corresponds to the baseline accuracy rate for a user, and the connected blue point is the accuracy rate using either the causal state model or the echo state network.

two groups, with the high tweet rate group having a tweet rate greater than 0.2 and the low tweet rate group having a tweet rate greater than or equal to 0.2, we can estimate the conditional density of improvements among these groups. These estimated densities are shown in Figure 3.5. We see that most of the improvement lies in the high tweet rate group, while the low tweet rate group is concentrated around 0 improvement.

3.4.3 Typical Causal State Models for the Users

The causal states $\{S_i\}_{i=1}^T$ of a stochastic process $\{X_i\}_{i=1}^T$ form a Markov chain, and the current causal state S_i plus the next emission symbol X_{i+1} completely determine the next causal state S_{i+1} [4]. These two properties of a causal state model allow us to write down an emission-decorated state-space diagram for a given



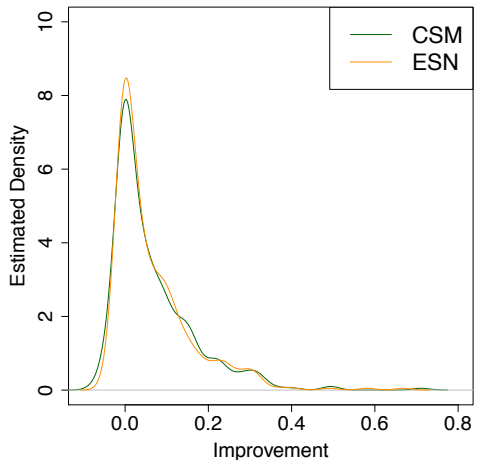
(a) Causal State Model (b) Echo State Network

Figure 3.4: The improvement over the baseline accuracy rate for the causal state model and the echo state network. For both models, the greatest improvement occurred for a coarsened tweet rate near $\frac{1}{2}$.

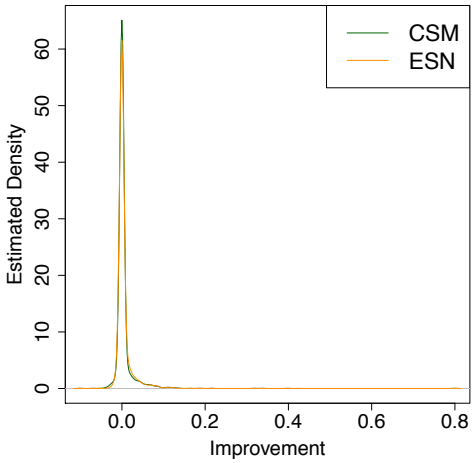
user. That is, the diagram resembles the state-space diagram for a Markov (or Hidden Markov) model, with the additional property that we must decorate each transition between states by the symbol emitted during that transition.

Several such diagrams are shown in Figure 3.6. Each circle corresponds to a causal state, and each arrow corresponds to an allowable transition. The arrows are decorated with $e_{ij}|p_{ij}$, where e_{ij} is the emission symbol observed transitioning from causal state i to causal state j, and p_{ij} is the probability of transitioning from causal state i to causal state j. For example, Figure 3.6(a) corresponds to a Bernoulli random process with success probability p. At each time step, the causal state returns to itself, emitting either a 1, with probability p, or a 0, with probability $1 - p$.

The four causal state models shown are typical examples of the models observed in 79.3% of the 3,000 users. The model corresponding to Figure 3.6(a) is



(a) High Tweet Rate



(b) Low Tweet Rate

Figure 3.5: The distribution of improvements for both the causal state model and echo state network, with the users partitioned into 'High Tweet Rate' (tweet rate greater than 0.2) and 'Low Tweet Rate' (tweet rate lower than 0.2) groups.

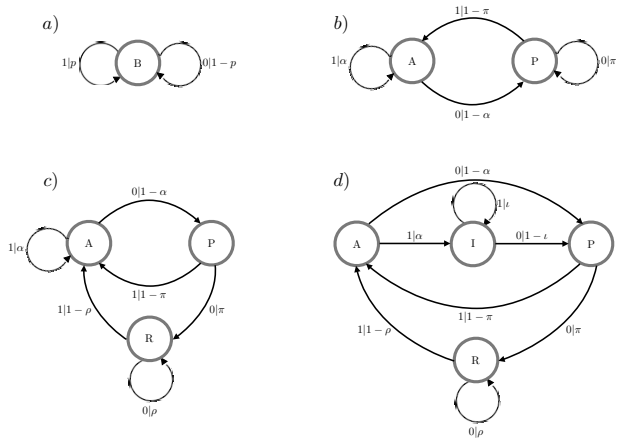


Figure 3.6: Typical 1, 2, 3, and 4-state causal state models. Of the 3,000 users, 383 (12.8%), 1,765 (58.8%), 132 (4.4%), and 100 (3.3%) had these number of states, respectively.

simple: the user shows no discernible memory and so the behavior is a biased coin flip. Only 383 (12.8%) of the users correspond to this model. The second model, Figure 3.6(b), displays more interesting behavior. We see that such users have two states, labeled A (active) and P (passive). While the user is in state A, it may stay in state A, continuing to emit 1s, or transition to state P emitting a 0. While in state P, the user may stay in state P, continuing to emit 0s, or transition to state A emitting a 1. Thus, these two states correspond to a user that is typically active or passive over periods of time, exhibiting 'bursting' behavior as in the second user in Figure 3.1.

Users corresponding to the causal state models shown in Figure 3.6(c) and Figure 3.6(d) exhibit even more involved behavior. Both have a rest state R, where the user does not tweet. However, the active states show more structure. For example, in Figure 3.6(c) we see that the user has an active state A, but sometimes

transitions to state P emitting a 0, where the user can then return back to the active state A or transition to the rest state R. Figure 3.6(d) shows similar behavior, but with an additional intermediate state I. While these models match our intuitions about how a typical Twitter user might behave, it is important to note that the models result entirely from applying CSSR to the data, and did not require any *a priori* assumptions beyond conditional stationarity.

3.4.4 Direct Comparison between the Performance of the Causal State Models and the Echo State Networks

Given the striking similarity in performance between the causal state model and the echo state network, we next compared them head-to-head on each user. The improvement for the causal state model vs. the improvement for the echo state network on each user is shown in Figure 3.7. As expected given the previous results, the improvements for each method are very strongly correlated.

Next, we investigated the top 20 users for which the causal state model or the echo state network outperformed the other model. For those users where the causal state model outperformed, the clearest indicator was the structured (near deterministic) behavior of the users. The top four such users are shown in Figure 3.8. The causal state model inferred from the data can be used to characterize the structure of the observed dynamics in a formal manner [4]. Because the hidden states $\mathcal{S} = \{s_1, \ldots, s_{|\mathcal{S}|}\}$ determine the observed dynamics, the entropy over those states can be used to characterize the diversity of behaviors a process is capable of.

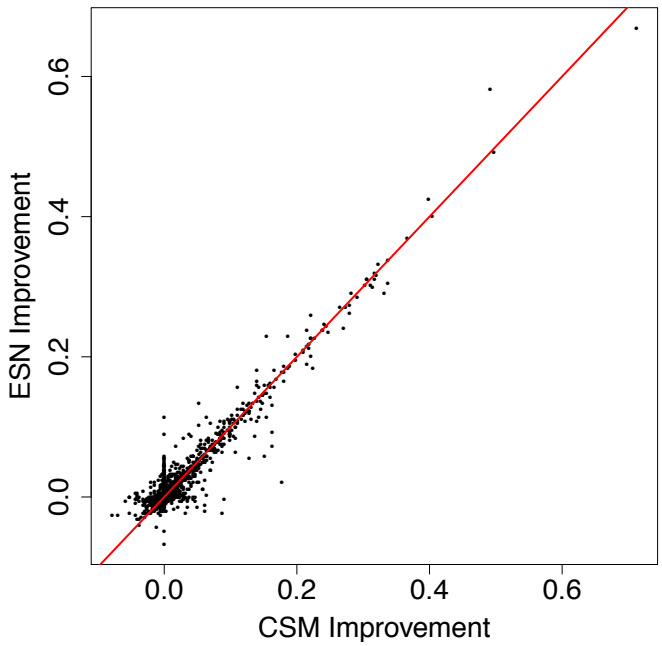


Figure 3.7: The improvement over baseline for the causal state model vs. the improvement over baseline for the echo state network. The red line indicates identity, where the two methods improve equally over the baseline predictor.

The entropy over the causal state process is called the *statistical complexity* of the process, and given by

$$C_\mu = H[S] \tag{3.12}$$

$$= -\sum_{s \in \mathcal{S}} P(S = s) \log_2 P(S = s). \tag{3.13}$$

Informally, it is the average number of bits of the past of a process necessary to optimally predict its future. For example, for an IID process, $C_\mu = 0$ bits, since none of the past is necessary to predict the future, while for a period-p process, $C_\mu \approx \log_2 p$ bits, since it takes $\log_2 p$ bits of the past to synchronize to the process.

Of the top twenty users best predicted by the causal state model, the average

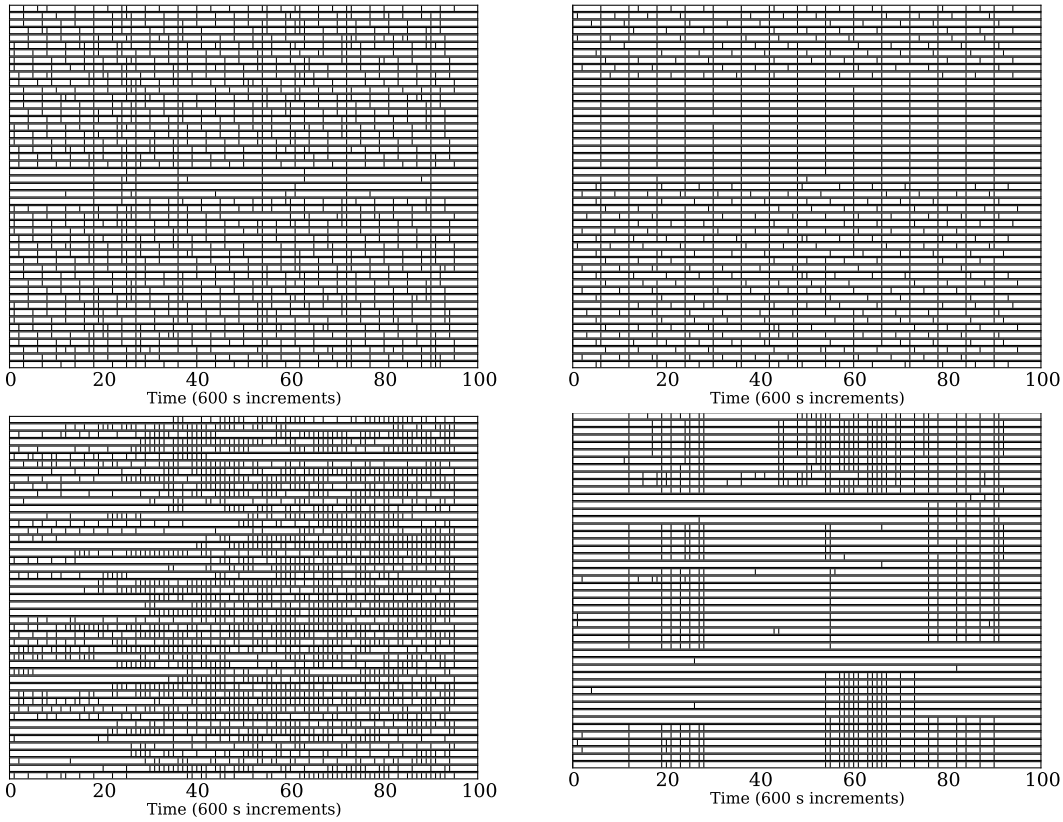


Figure 3.8: Raster plots for the four users where the causal state model most outperformed the echo state network. Note that in all but the bottom left case, the users show highly 'patterned' behavior. This is typical of the top twenty users for which the causal state model outperformed the echo state network.

statistical complexity was 3.99 bits, while the top twenty users best predicted by the echo state network had an average statistical complexity of 2.72 bits. Figure 3.9 shows the difference between the two methods as a function of the inferred statistical complexity. We see that the causal state models tend to outperform the echo state network for high statistical complexity users, while the echo state network tends to outperform for the low (near 0 bits) statistical complexity users.

Of the top twenty users best predicted by the echo state network, we observed

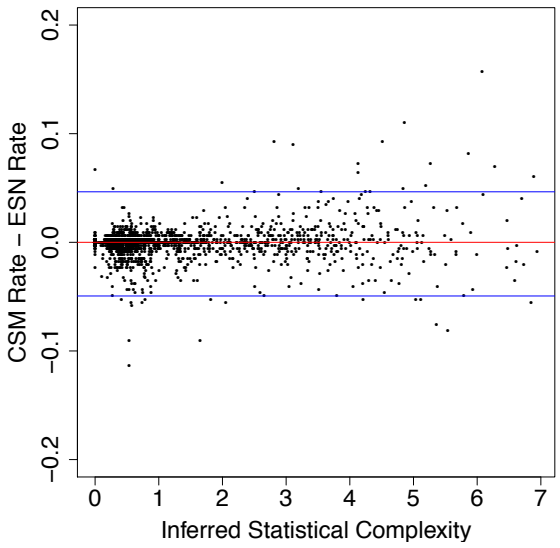


Figure 3.9: The difference in improvement between the causal state model and the echo state network for each user as a function of the inferred statistical complexity C of each user. The blue lines indicate the cutoff points above and below which the top twenty best users for the causal state model and echo state network, respectively, lie, and correspond to 0.0465 and -0.0494.

that the test set tended to differ from the training set. To test this hypothesis, we estimated the entropy rates of the test and training sets. The entropy rate h_μ of a stochastic process $\{X_i\}_{i=1}^{\infty}$ is defined as the limit of the block entropies of length L as the block length goes to infinity,

$$h_\mu = \lim_{L\to\infty} \frac{1}{L} H[X_1, \ldots, X_L]. \tag{3.14}$$

Thus, the entropy rate can be approximated by estimating block entropies

$$H_L = \frac{1}{L} H[X_1, \ldots, X_L] \tag{3.15}$$

of larger and larger block sizes and observing where the block entropies asymptote, as

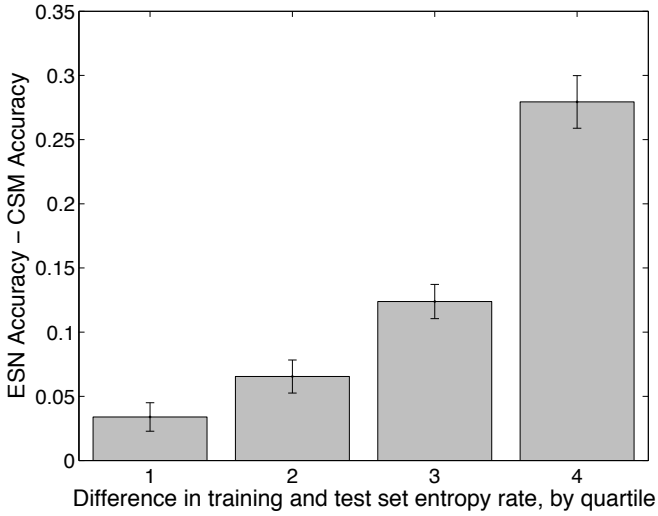


Figure 3.10: The difference in accuracy rates between the causal state model and the echo state network for each user, binned into quartiles by the absolute value of the difference in entropy rates for the training and testing sets. The causal state model performs best when this difference is low, and the echo state network performs best when it is high.

they must for a stationary stochastic process [81]. Unlike block-1 entropy (Shannon entropy), the entropy rate accounts for long range correlations in the process that may explain apparent randomness.

As we observed in the top twenty users, we see that overall the causal state model tends to perform best relative to the echo state network when the training and test set are similar, while the echo state network tends to outperform in the cases where the training and test set differ. This can be seen in Figure 3.10, in which the users have been grouped into quartiles by the absolute value of the difference between training and test set entropy rates.

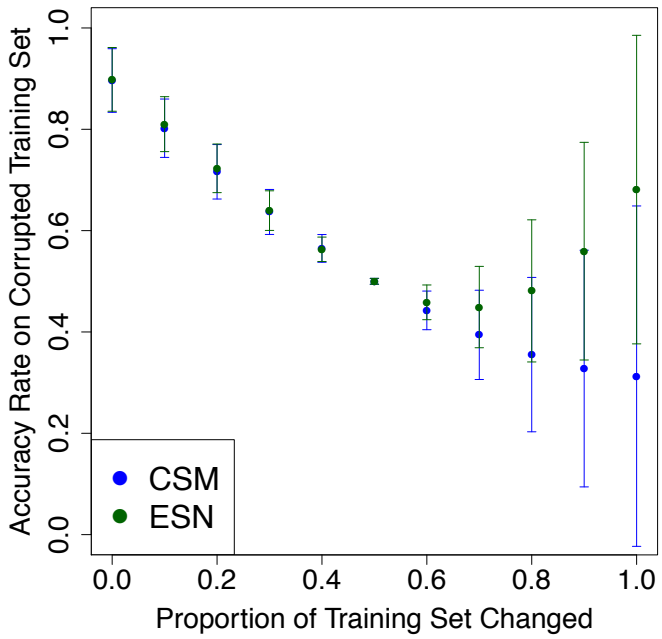


Figure 3.11: The accuracy rate of the causal state model and echo state network tested on its training data, with the training data corrupted by flipping a proportion q of the bits. Bars indicate plus or minus one standard deviation in the accuracy rates across all users.

3.4.5 Bit Flip Experiment

To further explore this difference between the two models, we performed the following 'bit flip' experiment. For each user, we trained both the causal state model and the echo state network on the full 49 days of data. We then tested the users on the same data, but with some proportion q of data set flipped such that 0s become 1s and vice versa, with q ranging from 0 to 1 in increments of 0.1. This allows us to synthetically create examples where the training and test sets differ as much or as little as desired.

The result of this experiment is shown in Figure 3.11. The causal state model performs as expected, with the accuracy rate degrading as the corruption in the

training set approaches 50%. Beyond this point, the large variance in the accuracy rates result from the different types of models inferred from the data. In particular, the 58.8% of users with a two-state 'bursting' causal state model as in Figure 3.6(b) continue to perform well, as the recoding of a burst of zeros or ones does not effect the predictive capability of the model.

The echo state networks show the same degradation in accuracy rate as the corruption in the training set approaches 50%, but beyond this amount they begin to show improvement. The large variance in the accuracy rates is again explained by a bimodality in the accuracy rates.

We believe the improvement in accuracy the echo state networks display when more than 50% of bits are changed is a result of many of the networks having learned a simple "trend-following" model: if you are in a tweeting state, continue tweeting; if you are in a non-tweeting state, continue not tweeting. This is very similar to the commonly observed two-state causal state model (Figure 3.6(b)) with one important difference — the echo state network does not fix the probabilities of being in either the active or passive states based on the training data. When a high proportion of bits have been flipped a sequence of, for instance, short periods of activity embedded in long stretches of quiescence will become the inverse: short periods of silence and long stretches of activity. A causal state model which has learned a two-state solution based on the original data will struggle since it expects different probabilities than those observed in the corrupted sequences, while an echo state network that has learned only to follow the recent trend will be able to adapt to the new, altered sequences so long as there are long trends remaining in the data.

The echo state network thus displays less fidelity to the observed data, but in doing so may be better able to adapt to particular perturbations if the patterns change, for example a user who maintains a 'bursting' pattern over time, but changes the length of these bursts.

3.5 Conclusion and Future Work

Overall, the causal state models and the echo state networks both showed improvement, and in some cases drastic improvement, over a baseline predictor. Moreover, for a large proportion of the users, the two methods gave very similar predictive results, as exemplified by Figure 3.7. Out of all the users, 58.8% had inferred causal state models similar to Figure 3.6(b), where a user has a tweeting state A and a non-tweeting state P. This bursting-type behavior is naturally captured by the echo state network, and thus the similarity in performance on these users is to be expected.

We have observed that predictability of user behavior is not homogeneous across the 3,000 users considered, and in many cases the *reason* for the difficulty in prediction differs across users. In some cases, considering a long enough history of a user's behavior is enough to predict their future behavior, but others still appear random after accounting for previous behavior.

In this chapter, we have shown that by building representations of the latent states of user behavior we can start to predict their actions on social media. We have done this using two different approaches, which have different ways of captur-

ing the complexity of user behavior. Causal state modeling starts from a simple model and adds structure, while echo state networks start with complex descriptions and simplify relationships. We hypothesized that these two methods would perform differently when applied to a diverse collection of users derived from a real world social media context. Our results indicate that the two methods perform differently under different conditions. Specifically, computational mechanics provides a better model of a user's behavior when it is highly structured and does not change dramatically over time, while the echo state network approach seems to be more adaptive, while at the same time giving up some of the deep structure present in the behavior. Moreover, we have shown that both methods are robust to noise and decay gracefully in performance.

Ultimately, the two methods performed very similarly on a large proportion of the users. It should be noted that this was not expected. The two methods differ drastically in their modeling paradigm, and the data was quite dynamic, providing plenty of opportunity for differentiation. Our best explanation is that in the end, and as noted above, most users exhibit only a few latent states of behavioral processing, and as such any model which is able to capture these states will do well at capturing the behavior of users. We could test this hypothesis in future work by restricting the number of states that both the echo state network and the computational mechanics approach can use, and observing if the results change substantially.

However, before we address that question, there are several other limitations of the present work that need to be addressed. One of the biggest weaknesses of the present approach is its failure to incorporate exogenous inputs to a user. That

is, we have treated each user as an autonomous unit, and only focused on using their own past behavior to predict their future behavior. In a social context, such as Twitter, it makes more sense to incorporate network effects, and then examine how the behavior of friends and friends of friends directly impact a user's behavior. For example, the behavior of many of the users, especially those users with a low tweet rate, may become predictable after incorporating the behavior of users in their following network. The computational mechanics formalism for doing so has been developed in terms of random fields on networks [7] and transducers [6], but it has yet to be applied to social systems.

We have also simplified the problem down to its barest essentials, only considering whether a tweet has occurred and not its content. Information about the content of a tweet should not *decrease* the predictive abilities of our methods, and could be incorporated in future work, for example, by extending the alphabet of symbols which we allow X_i to take.

This study has also focused on user behavior over a month and a half period. With additional data, a longitudinal study of users' behaviors over time could be undertaken. We have implicitly assumed the conditional stationarity of behavior in our models, but these assumptions could be tested by constructing models over long, disjoint intervals of time and comparing their structure.

We have seen that taking a predictive, model-based approach to exploring user behavior has allowed us to discover typical user profiles that have predictive power on a popular social media platform. Moreover, we have shown this using two different modeling paradigms. In the near future, we plan to extend this work to take into

account the social aspects of this problem, and see how network effects influence user behavior. However, the increase in predictive power *without* explicitly incorporating social factors gives us reason to believe that it is possible to make predictions in the context of user interactions in social media. Such predictions could be useful in any number of domains. For instance, in a marketing type approach, these models could be used to understand who will respond to a message that is sent out to a group of users, and potentially even assist in the determination of whether or not a particular piece of content will go viral. Predicting user behavior on social media has the potential to be transformative in terms of both our understanding of human interactions with social media, and the ability of organizations to engage with their audience.

Chapter 4: Forecasting High Tide: Predicting Times of Elevated Activity in Online Social Media

4.1 Introduction

For a wide variety of organizations, companies, and individuals there is a growing interest in using social media to get their message out. For instance, brand managers are often tasked with launching promotions that raise the awareness of their brand among users of social media. However, the signal that a brand is trying to convey can easily get lost in the 'noise' produced by other brands, individuals, bots, etc. While good content is important to engage an audience, it is also important to know when users will pay attention to the content in order to increase the chance that the message is spread. Therefore, a brand manager must consider not only what they want to say, but *when* they want to say it.

In order to effectively spread a message on a social media platform, an important first step is to understand the patterns of user engagement. After receiving and becoming aware of information, users on a social media platform then evaluate the content of the information and decide whether it should be retransmitted or not. Previous research has examined different criteria for this decision, including

the sender's level of activity and the freshness of the information [82], as well as the user's benefit from spreading the information [83]. In this research, instead of exploring criteria related to evaluation of content and the decision to retransmit, we focus on timing when users on social media are engaged in retransmission behavior. A key assumption of our approach is that information is most likely to be retransmitted during the highest activity periods. In particular, in this chapter, we study the task of predicting user engagement on Twitter, and we measure engagement in terms of the number of users actively issuing retweets.

It is well known that user activity on social media services follows both diurnal and weekly patterns [84, 85]. For example, Figure 4.1 demonstrates the number of users active on Twitter out of a collection of 2145 over a four week period, starting on Mondays at 9am EST. At the daily level, the number of users actively retweeting increases over the course of a day and then decreases at night. However, the times of peak activity also fluctuate from week to week. Such fluctuations in social systems has been attributed to the fact that observed aggregate social behavior, driven by individual human actions, can be described as mixtures of Poisson and non-Poisson processes, where these processes can be seen as modeling individual decision making [86]. Thus, we expect the aggregate behavior of a collection of users who have different decision making processes to exhibit significant temporal fluctuation from seasonality from week to week. In order to effectively reach a large number of activated users, it is therefore important to determine when they are the most engaged by tracking such fluctuations while controlling for the diurnal and weekly seasonality. Moreover, recent work studying the attention of users on Twitter has

found that retweets of a given tweet typically occur on the time scale of minutes [87, 88]. Given this observation, it is also important that we track seasonal fluctuations at a fine temporal resolution.

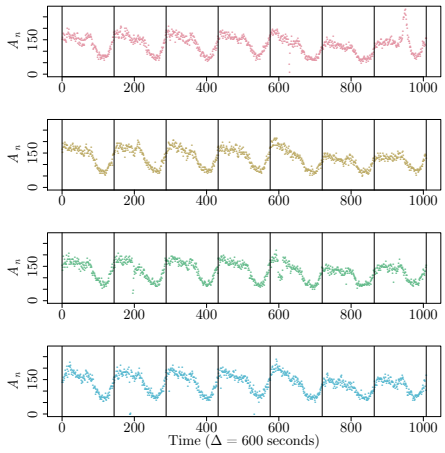


Figure 4.1: The number of users actively retweeting during disjoint ten minute windows. Each row corresponds to a week, and each column corresponds to a day of the week, starting from Monday.

In order to model the number of active retweeters on Twitter at any given time, we propose three approaches: a seasonality model that assumes the overall retweet activity on Twitter is fully explained by the time-of-day and day-of-week, an autoregressive model that explicitly models deviations from the day-to-day seasonality, and an aggregation-of-individuals approach that models the activity patterns of each individual user and then aggregates these models to describe the overall activity pattern.

The seasonality model is based on the assumption that user engagement over time can be explained by seasonal patterns at the daily and weekly level. Thus, in order to predict the time when users are engaged at a certain level using this model, we consider only engagement patterns from the past.

The autoregressive model seeks to describe the population-level fluctuations about the seasonality using a simple linear autoregressive model. We assume that the deviations from seasonality have memory where we can think of this memory in terms of activation / deactivation of the users on Twitter. For example, a certain topic might become popular over the course of several hours, leading to activity greater than expected by the baseline seasonality. Such bursts of activity have been observed on both Twitter and blogging platforms [89]. By noting when and how such bursts occur, we can better predict the number of users active on Twitter compared to using seasonality alone.

The aggregation-of-individuals model explicitly views the overall activity as the accumulation of the activity patterns of all of the users under consideration. In particular, we model each user-to-be-aggregated as a point process with memory [64]. In this case, each individual user can become activated / deactivated, depending on their own previous behavior and the behavior of their inputs. By viewing the user as a computational unit, we can build a predictive model of how they interact with Twitter. This approach has been successfully applied to individual level prediction [90] on Twitter, where many high volume users were found to be well-described by such a model. We can then aggregate these individual level models to produce a global prediction of activity levels that accounts for individual-level activation.

In the rest of this chapter, we explore the problem of identifying periods of high activation on a social media platform. We begin by describing our three models and relevant literature. Then we describe the data sets used to test the predictive ability

of these models for the proposed problem. Next, we review the predictive ability of the various models, and compare the benefits and tradeoffs of each approach. Finally, we conclude with the limitations of the present work and future directions to extend and improve it.

4.2 Related Work

A large body of work has investigated the dynamics of technology-mediated human interaction. Relevant to our work, [91] found that human behavior on email services is dominated by bursty-type behavior, with periods of high activity separated by long stretches of inactivity. The authors of [92] found stereotypical temporal patterns in the interaction between blogs and mainstream media news. Studies of Twitter have found similar stereotypical aggregate behavioral patterns for the popularity of particular hashtags over time [92,93]. More recent work has sought to develop first principle mathematical models explicitly geared towards human behavior on social media [94, 95].

A great deal of work has been done on the problem of predicting the future popularity of individual tweets and hashtags based on their features. As a very recent example, in [96], the authors performed an experiment to investigate how the wording of a tweet impacts whether it is retweeted, controlling for both the author and the topic of the tweet. In [97], the authors predict the volume of tweets about a hashtag day-to-day using features extracted from a corpus of tweets containing that hashtag on previous days. Similar studies can be found in [98–101]. The problem

of predicting individual tweet, hashtag, and topic popularity has been well-studied, and these references are only meant to give a sampling of the much larger literature on the subject.

The problem of predicting the total volume of tweets over time has attracted much less attention from the research community. Notable exceptions include [102–104]. In [102], the authors build a predictive model for the overall volume of tweets related to a particular hashtag. Similar to one of our approaches, the authors do this by aggregating individual predictive models for a universe of users, where the users were chosen if they previously tweeted on a topic and followed a user who also tweeted on that topic. They then identified predictive models for each user at the resolution of days, where predictions were made based on previous activity of a user and their local network structure. The goal of predicting day-resolution volume from users on a particular topic differs greatly from predicting high volume times from a collection of users determined based on their network properties, which is the goal of this chapter. In [103], the authors seek to determine the one hour period in which the followers of a given collection of users are most likely to be active. However, their investigation is purely sociological in nature, in that they make no predictions, and the data used in their analysis only covered a single week of activity. Thus, their approach is not directly applicable to forecasting retweet volume from streaming data. Finally, in [104], the authors use a two state Hidden Markov modeling framework, where the hidden states correspond to when the user is either in an active mode or an inactive state. Using these models, they predict the expected interarrival time for a user given their observed previous behavior by filtering their

hidden state, and make predictions based on this time. Thus, this approach is similar in spirit to our aggregation-of-individuals approach. However, they assume a particular hidden state model architecture that is homogeneous across users, while our approach, as we will see, allows for model heterogeneity across users. Moreover, while their approach could be used to predict total retweet volume by aggregating their individual model predictions, they focus on individual level prediction.

4.3 Methodology

Here we define our exact problem and the proposed solutions. Consider a set $\mathcal{U} = \{u_1, u_2, \ldots, u_U\}$ of U users. Each user in $\mathcal{U}$ has an individual retweet history. Let Δ be a time interval; here we take $\Delta = 10$ minutes. Then for each user u in the set of users $\mathcal{U}$, we specify their retweet activity during any window of length Δ by

$$X_n(u) = \begin{cases} 1 & : \text{ user } u \text{ retweeted between times} \\ & \qquad (n-1)\Delta \text{ and } n\Delta \\ 0 & : \text{ otherwise} \end{cases} . \tag{4.1}$$

That is, $\{X_n(u)\}_{n=1}^N$ specifies the retweet activity of the user during each of the N time intervals $[0, \Delta), [\Delta, 2\Delta), \ldots, [(N-1)\Delta, N\Delta)$.

The total number of users active during any time interval $[(n-1)\Delta, n\Delta)$ is then given by

$$A_n = \sum_{u \in \mathcal{U}} X_n(u). \tag{4.2}$$

This is the value we seek to predict.

4.3.1 Seasonality

For the seasonality model, we assume that retweet activity shows day-to-day variability, but regularity from week-to-week. We assume that the seasonality repeats every T timesteps,

$$s_n = s_{n+jT}, \qquad j = 1, 2, \ldots \tag{4.3}$$

and that the observed number of users retweeting A_n is given by

$$A_n = s_n + \epsilon_n \tag{4.4}$$

where ϵ_n can be thought of as the deviation from the seasonality at any given time n. Under the assumption of seasonality, we infer the seasonal component by averaging across W weeks [105],

$$\hat{s}_n = \frac{1}{W} \sum_{j \in \{0,1,\ldots,W-1\}} A_{n+jT}, \quad n = 1, \ldots, T. \tag{4.5}$$

Figure 4.2 shows the aggregate retweet activity across the four weeks from Figure 4.1 with the estimated seasonality superimposed.

If we assume that $\{\epsilon_n\}_{n=1}^{N}$ is a realization from a white noise process, the optimal predictor under mean-squared loss for A_n is s_n, the seasonality. Thus, we

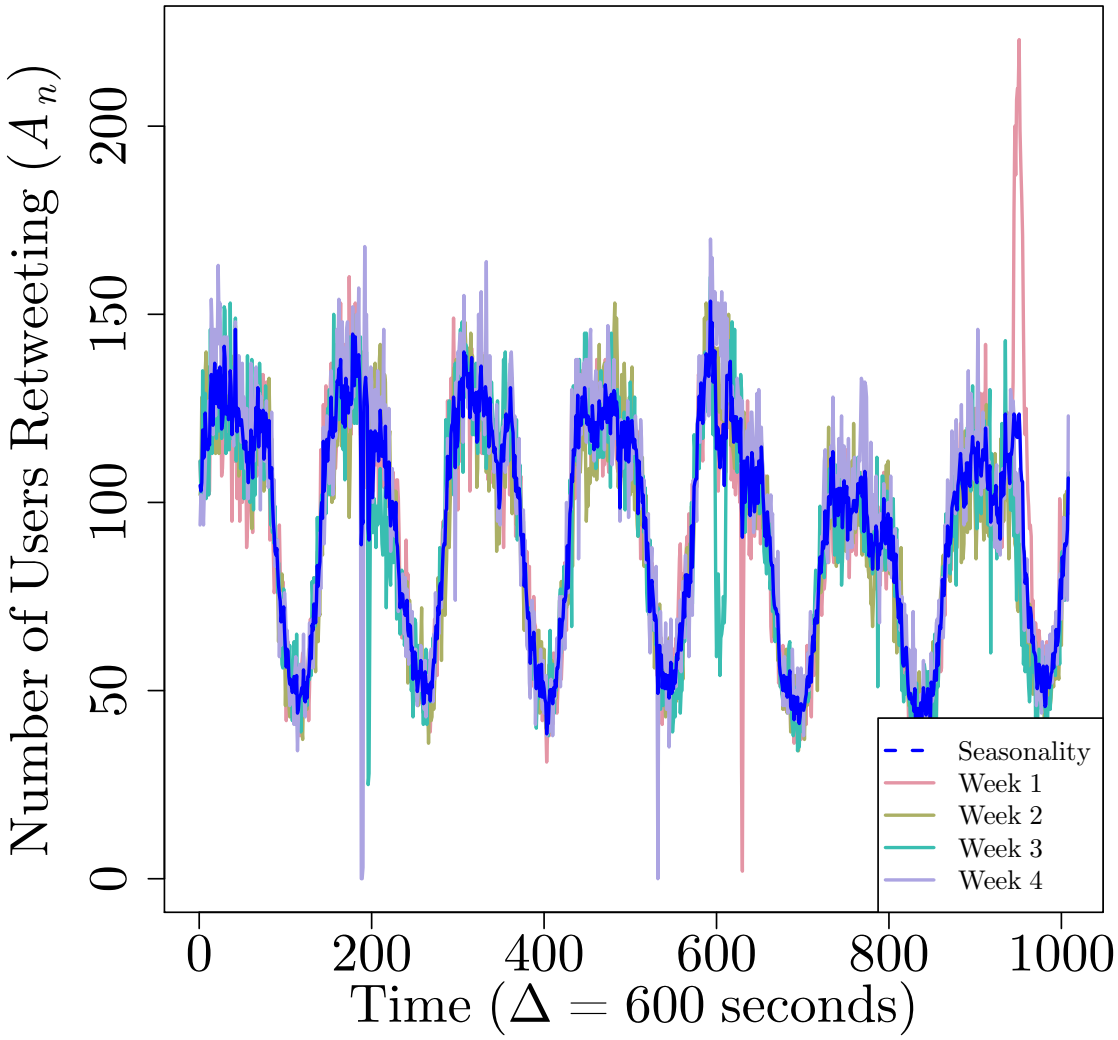


Figure 4.2: The number of users retweeting A_n over four consecutive weeks. The estimated seasonality $\hat{s}_n$ is shown in blue.

use our estimator for the seasonality as the predictor for the seasonality model,

$$A_n^{\mathrm{S}} = \hat{s}_n. \tag{4.6}$$

4.3.2 Aggregate Autoregressive Model

In the seasonality model, we have assumed that the residuals $\{\epsilon_n\}_{n=1}^N$ are white noise. More explicitly, we have assumed that they show no autocorrelation: $E[\epsilon_t \epsilon_s] = \sigma_\epsilon^2 \delta_{st}$, where σ_ϵ^2 is the variance of the white noise process and δ_{st} is the Kronecker delta. A more reasonable model for the residual would incorporate memory, since aggregate social systems are known to exhibit such memory [106]. Thus, a simple refinement of the previous model allows for memory in the deviations from seasonality. More explicitly, we consider the model

$$A_n = s_n + Y_n \tag{4.7}$$

where we now take $\{Y_n\}_{n=1}^N$ to be a realization from an autoregressive process of order p, an AR(p) model [105]. That is, we consider the dynamics of Y_n to be governed by

$$Y_n = \sum_{j=1}^{p} b_j Y_{n-j} + \epsilon_n \tag{4.8}$$

where $\{\epsilon_n\}$ is again a white noise process with mean 0 and variance σ_ϵ^2.

The predictor for the aggregate autoregressive model is

$$A_n^{\mathrm{AR}} = \hat{s}_n + \sum_{j=1}^{\hat{p}} \hat{b}_j \hat{Y}_{n-j}, \tag{4.9}$$

where $\hat{Y}_n = A_n - \hat{s}_n$ is the deviation of the observed aggregate retweeting activity

from the estimated seasonality at time n. We choose the autoregressive order $\hat{p}$ by minimizing the the Akaike information criterion on the training set [107].

4.3.3 Aggregation of Causal State Models

Before describing the aggregation procedure, we briefly review computational mechanics, which is our basic modeling approach for the individual-level models. [4] provides a more in-depth introduction to computational mechanics, and [90] describes an application of computational mechanics to modeling individual user activity on Twitter. Computational mechanics provides a framework for describing stationary [108] (and more generally, *conditionally* stationary [71]), discrete-time, discrete-alphabet stochastic processes by linking the observed process to a hidden state process. In this way, the formalism of computational mechanics is closely related to Hidden Markov Models and other state-based models of discrete-alphabet stochastic processes [109]. In particular, any conditionally stationary stochastic process $\{X_n\}$ naturally induces a hidden state process $\{S_n\}$, where the transition structure of the hidden state process is determined by the predictive distribution of $\{X_n\}$. The hidden state process $\{S_n\}$ is always Markov, and the combination of its Markov chain representation and the state conditional emission probabilities $P(X_n = x|S_{n-1} = s)$ is called the *causal state model* or ϵ-*machine* for the stochastic process $\{X_n\}$. In the case where the predictive distribution for $\{X_n\}$ is unknown, machine reconstruction algorithms can be used to automatically infer the ϵ-machine that best describes the observed data $\{X_n\}_{n=1}^{N}$. We use the Causal State Splitting

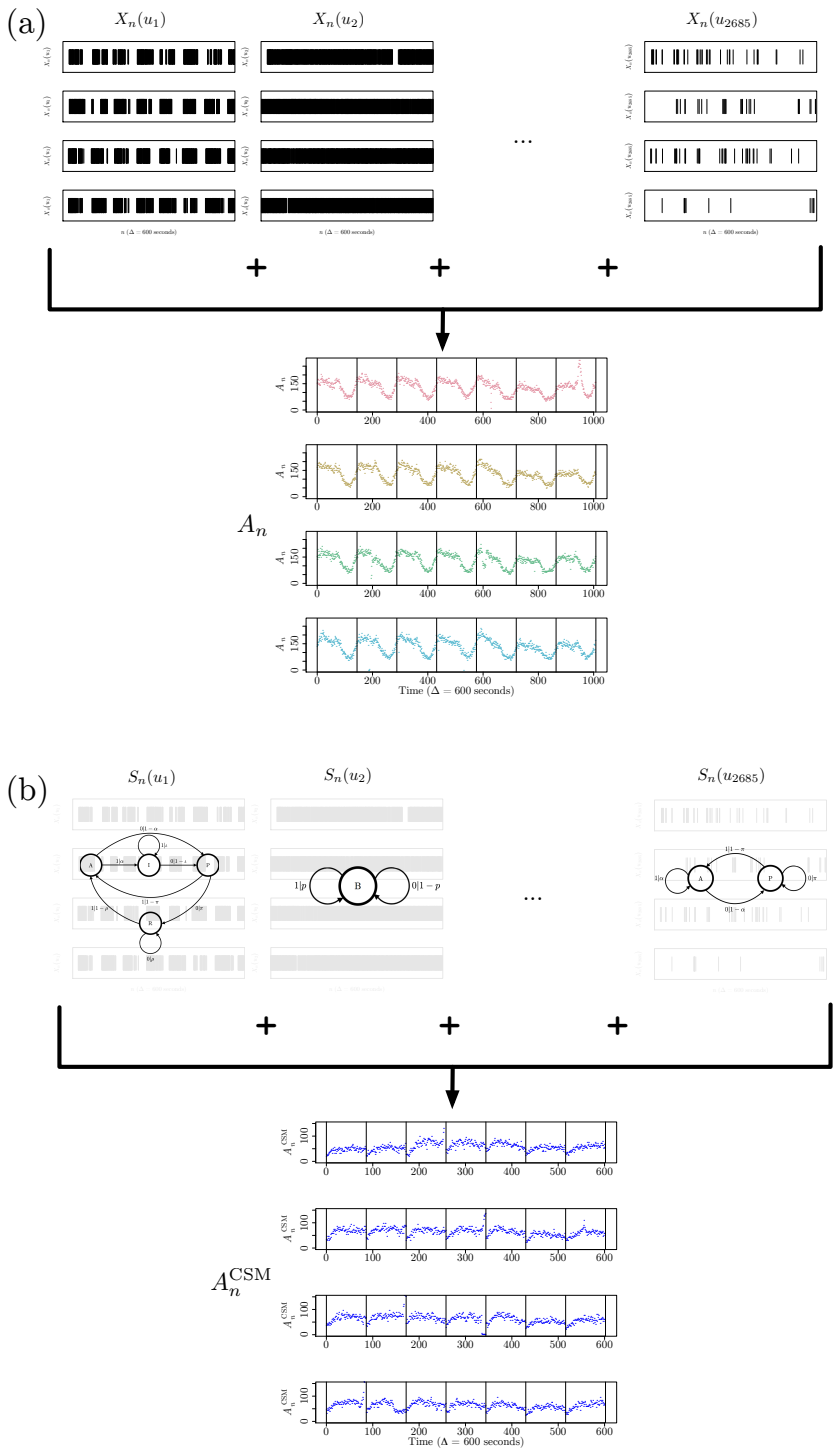


Figure 4.3: A demonstration of how (a) the retweet volume A_n results from the summation of the individual retweet behavior $\{X_n(u)\}_{u \in \mathcal{U}}$ of the users in $\mathcal{U}$ and (b) the aggregation-of-individuals prediction A_n^{CSM} is formed via filtering through each user u's ϵ-machine.

Reconstruction (CSSR) algorithm [46] to infer an ϵ-machine for each user's observed retweeting activity.

As with the autoregressive model, the CSSR algorithm requires a maximum history length $L_{\max}$ to look into the past in order reconstruct the ϵ-machine associated with a user u's behavior. While theory exists for choosing the largest $L_{\max}$ such that we can consistently infer the one-step-ahead predictive distributions used in CSSR [50], we take the practical approach of choosing $L_{\max}$ based on cross-validation. In particular, we perform 5-fold cross validation using the log-likelihood of the held out data as our objective function. The form of the log-likelihood associated with a realization from a stochastic process under an ϵ-machine model may be found in [12].

For each user u, we reconstruct their associated ϵ-machine. We then perform prediction as follows: at time $n-1$, we determine the current causal state $S_{n-1}(u)$ for each user u based on their activity pattern $X_1^{n-1}(u) = (X_1(u), X_2(u), \ldots, X_{n-1}(u))$. The causal state $S_{n-1}(u)$ specifies the one-step-ahead predictive distribution for each user, $P(X_n(u) = 1 | S_{n-1}(u) = s(u))$. We then aggregate these probabilities to form our prediction for the number of active users at the next time step,

$$A_n^{\text{CSM}} = \sum_{u \in \mathcal{U}} P(X_n(u) = 1 | S_{n-1}(u) = s(u)). \tag{4.10}$$

This can be seen to be the expected number of users active at time n given the causal states of the users at time $n-1$, under the assumption that the behavior of a user u at time n is independent of the causal states of all others users at time $n-1$

given the causal state of u at time $n-1$.

4.4 Data Collection and Selection of $\mathcal{U}$

We begin with a collection of 15000 Twitter users whose statuses (Tweet text) were collected over two disjoint five week intervals: from 25 April 2011 to 29 May 2011 and from 1 October 2012 to 5 November 2012. The users are embedded in a 15000 node network collected by performing a breadth-first expansion of the active followers of a random seed user. In particular, the network was constructed by considering the followers of the seed user, and including those followers considered active (i.e. users who tweeted at least once per day over the past one hundred days). The collection of users continued from the followers of these followers, etc., until 15000 users were included. From this network of users, the subset of users $\mathcal{U}$ was chosen to account for 80% of the retweet volume for the first four weeks in the five week period under consideration. That is, we take u_1 to be the user issuing the greatest number of retweets, then u_2 to be the user issuing the second greatest number of retweets, etc., until we reach the user u_U such that the total number of retweets issued by the users in $\mathcal{U}$ account for 80% of the retweet volume. This results in $U = 2145$ users for the 2011 collection and $U = 1610$ users for the 2012 collection. Because we are interested in predicting times of greatest retweet activity, for each day we only consider the retweet activity from 6 AM EST to 10 PM EST. The data used in our analysis can be made available upon request by the corresponding author.

4.5 Results

In the following results, we use the first four weeks of the five week periods from 2011 and 2012 for inference of the three model types, and leave the last weeks from each year for testing. As described in the methodology section, we choose the parameters of each model as follows. The seasonality model has no tuning parameter, and we use the full four weeks to infer the seasonality component. We choose the model order p of the autoregressive model to maximize the Akaike information criterion on the four week training period. For each causal state model in the aggregation-of-individuals model, we infer the user-specific history length L by 5-fold log-likelihood cross-validation over the 28 days in the training sets.

4.5.1 Adjustment to the Aggregation-of-Individuals Model

As described in the methodology section, the predictor for the aggregation-of-individuals model (4.10) is equivalent to the expected number of users in $\mathcal{U}$ who are active at time step n given their causal states at $n-1$ under a certain independence

assumption. In particular, we have taken the predictor to be

$$A_n^{\mathrm{CSM}} = E\left[\sum_{u\in\mathcal{U}} X_n(u)\middle| S_{n-1}(u_1),\ldots,S_{n-1}(u_U)\right] \tag{4.11}$$

$$= \sum_{u\in\mathcal{U}} E[X_n(u)|S_{n-1}(u_1),\ldots,S_{n-1}(u_U)] \tag{4.12}$$

$$= \sum_{u\in\mathcal{U}} E[X_n(u)|S_{n-1}(u)] \tag{4.13}$$

$$= \sum_{u\in\mathcal{U}} P(X_n(u)=1|S_{n-1}(u)), \tag{4.14}$$

where going from (4.12) to (4.13) we make the assumption that for all $u \in \mathcal{U}$, the observed behavior of user u at time n is independent of the causal states of all other users u' at time $n-1$, given the causal state of user u at time $n-1$. While such an independence relationship holds when conditioning on the local causal states of a time-varying random field [7], it need not be true when conditioning on the marginal causal states.

Motivated by the form of the deviation of (4.10) from the predicted value (see Figure 4.4), we define the adjusted aggregation-of-individuals predictor as

$$A_n^{\mathrm{CSM}^*} = \beta_0 + \beta_1 A_n^{\mathrm{CSM}}, \tag{4.15}$$

where the parameters β_0 and β_1 were estimated by regressing the true values A_n from the training set on the unadjusted aggregation-of-individuals predictions A_n^{CSM} from the training set. We will use this predictor for the remainder of this work, and address alternative corrections in the conclusion.

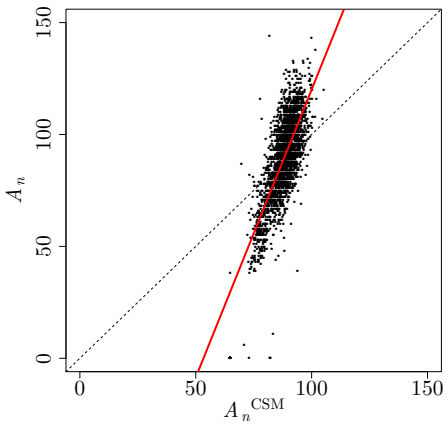


Figure 4.4: The transformation of the aggregation-of-individuals model used to adjust for associations in user behavior. The red line corresponds to the linear least squares fit from regressing the true values A_n from the training set on the unadjusted aggregation-of-individuals predictions A_n^{CSM} from the 2011 data.

4.5.2 Predicting Activation Level at Varying Thresholds

We next present an experiment to test the predictive capability for each of the three proposed models. As mentioned in the introduction, ideally a potential influencer would like to choose the optimal time(s)-of-day to send out a message such that the largest number of users will be active around those times. As a proxy for this goal, we consider the task of identifying whether or not the activity level over an interval of length Δ will fall into the $100p^{\text{th}}$ percentile for that day. As an example, how well can we predict whether the number of activated users falls within the 80^{th} percentile for a given day?

Let N_Δ be the number of timepoints to predict on in a day ($N_\Delta = 86$ for this analysis). For a given day $d \in \{1, 2, \ldots, 7\}$ in the testing set, the true distribution

of the activity levels is given by

$$F_d^{\text{True}}(a) = \frac{1}{N_\Delta} \sum_{n=n_{\text{train}}+N_\Delta(d-1)+1}^{n_{\text{train}}+N_\Delta d} \mathbb{1}\left[A_n \leq a\right]. \tag{4.16}$$

We then define the historical distribution of the activity levels for a day d in terms of the estimated seasonality for that day from the training set

$$F_d^{\text{Hist}}(a) = \frac{1}{N_\Delta} \sum_{n=N_\Delta(d-1)+1}^{N_\Delta d} \mathbb{1}\left[A_n^S \leq a\right]. \tag{4.17}$$

We will use $F_d^{\text{Hist}}(\hat{A}_n)$ to predict whether or not a predicted activity level $\hat{A}_n$ exceeds the quantile p^* of activity for a given day, where $\hat{A}_n$ is one of A_n^{S}, A_n^{AR}, or $A_n^{\text{CSM}^*}$. That is, for a threshold p, we predict the indicator for whether the activity at time n will exceed some quantile p^* as

$$\hat{I}_n(p) = \begin{cases} 1 & : F_{d(n)}^{\text{Hist}}(\hat{A}_n) > p \\ 0 & : \text{ otherwise} \end{cases}. \tag{4.18}$$

Whether or not the activity at time n exceeded the quantile p^* is then given in terms of the true distribution as

$$I_n = \begin{cases} 1 & : F_{d(n)}^{\text{True}}(A_n) > p^* \\ 0 & : \text{ otherwise} \end{cases}. \tag{4.19}$$

Table 4.1: The AUC for each of the methods on the test weeks from 2011 and 2012.

Year	Seasonality	Autoregressive	Agg. of Individuals
2011	0.778	0.773	**0.825**
2012	0.720	**0.773**	0.771

As we vary the threshold value p, the true positive rate is given by

$$\mathrm{TPR}(p) = \frac{\sum_{n=n_{\mathrm{train}+1}}^{n_{\mathrm{test}}} \mathbb{1}\left[\hat{I}_n(p) = 1, I_n = 1\right]}{\sum_{n=n_{\mathrm{train}+1}}^{n_{\mathrm{test}}} \mathbb{1}\left[I_n = 1\right]} \tag{4.20}$$

and the false positive rate is given by

$$\mathrm{FPR}(p) = \frac{\sum_{n=n_{\mathrm{train}+1}}^{n_{\mathrm{test}}} \mathbb{1}\left[\hat{I}_n(p) = 1, I_n = 0\right]}{\sum_{n=n_{\mathrm{train}+1}}^{n_{\mathrm{test}}} \mathbb{1}\left[I_n = 0\right]}. \tag{4.21}$$

We show the ROC curves associated with the fixed quantile $p^* = 0.8$, along with their AUCs, for the test weeks from 2011 and 2012 in Figure 4.5 and Table 4.1. The true and false positive rates are computed using the last 86 of the 96 time points in each day, since both the autoregressive and aggregation-of-individuals models require up to ten timepoints to begin prediction depending on the model order p or largest history length L, respectively.

Overall, based on the AUC values, the aggregation of individuals model performs best in 2011 and the autoregressive model performs best in 2012. However, inspection of the ROC curves indicates that based on the desired balance between true and false positives, each of the models may outperform the others, with no model strictly dominating. For example, if a high false positive rate is acceptable,

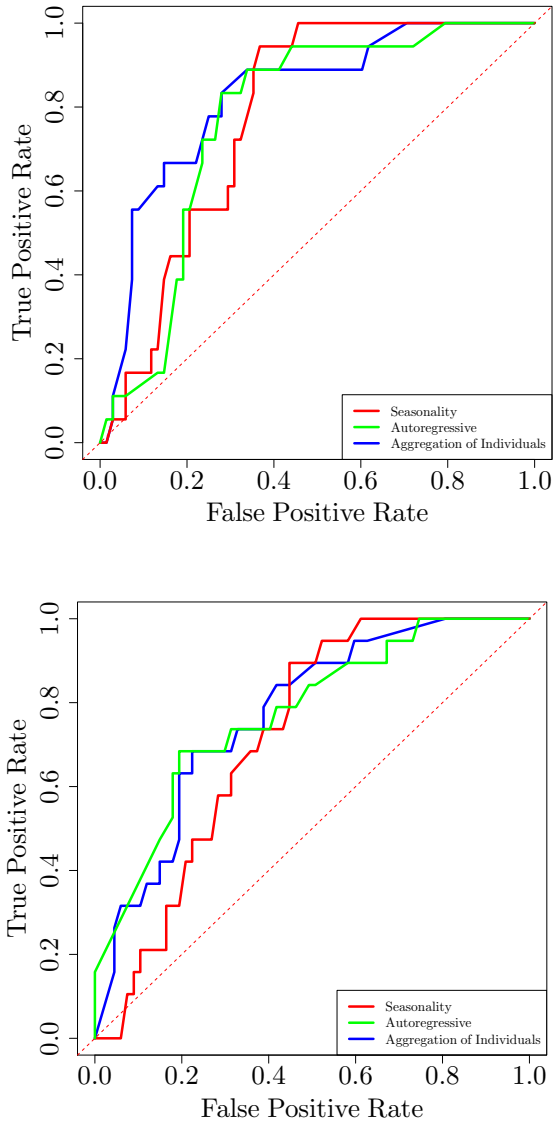


Figure 4.5: The ROC curves associated with each of the three models for the testing week in 2011 (top) and 2012 (bottom) with p^* fixed at 0.8. The AUC values for the seasonality, autoregressive, and aggregation-of-individuals models for 2011/2012 are 0.778/0.720, 0.773/0.773, and 0.825/0.771.

the seasonality model achieves the lowest false positive rate to give a 100% true positive rate on the testing sets in 2011 and 2012. However, the seasonality model generally underperforms when the desired false positive rate is low, in which case either the aggregation of individuals model (in 2011) or the autoregressive model (in 2012) performs better. While we only report on predicting under the condition that $p^* = 0.8$, we find similar results for $p^* \geq 0.7$.

4.5.3 Utility of Individual Level Models Beyond Aggregate Prediction

Though we do not focus on individual-level prediction in this chapter, we wish to highlight some of the possible advantages offered by the aggregation-of-individuals approach not immediately evident from the ROC analysis above. In particular, as demonstrated in Figure 4.3, the aggregation-of-individuals generates individual level, behavioral models for each user u. These models have the advantage of being interpretable. Consider the four models in Figure 4.6. The models can be represented as directed graphs, where each vertex corresponds to a causal state, and each arrow corresponds to an allowed emission from that state. The arrows are decorated with the emission symbol $x \in \{0, 1\}$ (*i.e.* user u either retweets or does not during a time interval) and the causal state conditioned emission probability $P(X_n(u) = x|S_{n-1}(u) = s)$ of transitioning from state s while emitting symbol x. That is, each arrow is decorated as $x|P(X_n(u) = x|S_{n-1}(u) = s)$.

These models allowed for user-specific targeting. Consider the model repre-

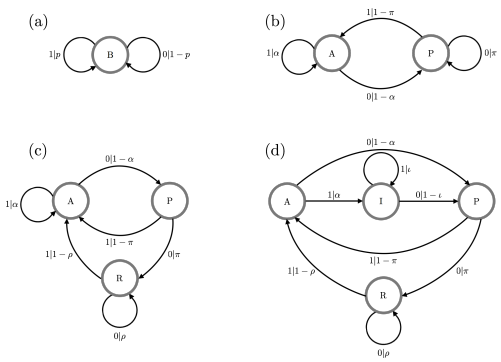


Figure 4.6: Four example ϵ-machines inferred from the users. (a) A user who retweets at random with bias p. (b) A user who retweets in a bursty manner, with an active state A and a passive state P. (c) A user who retweets in a bursty manner, with a refractory state R. (d) A user who retweets in a bursty manner with both a refractory state R and an intermittent state I.

sented by (b). Users of this type tend to retweet in a bursty manner, with an active state A and a passive state P. This corresponds to a simple order-1 Markov model. For such users, it is sufficient to target them when they have recently retweeted. Users exhibiting behavior like models (c) or (d) require more subtle targeting. Model (c) has the same active and passive states as in (b), but with an additional refractory state R that occurs after the user is quiescent while in the passive state P. Depending on the balance between $1 - \rho$ and α, which correspond to the probabilities of retweeting from the active and refractory states, it may be more beneficial to target the user when they are currently active or when they are 'resting' in the refractory state. Model (d) is similar to model (c), with an additional intermediate state I that occurs after the user has issued a retweet from the active state A. Again, depending on the balance between α, $1 - \rho$, and ι, the user can be targeted for when they are most likely to retweet. Many of the users have simple ϵ-machines similar to (a), (b), (c), and (d), which allow for this sort of user-specific targeting.

4.6 Conclusion

We have found that while user retweet activity clearly exhibits seasonality from day-to-day and week-to-week, seasonality alone does not explain the times of high user activity on social media. Incorporating additional information about either the deviations from seasonality or the behavioral patterns of individual users allows for more accurate prediction of times of high volume, especially when a low false positive rate is desired. Since overexposure to a message may lead to reduced user engagement (content fatigue) due to the repetitive nature of the message, it can be said that having a low false positive is important in this motivating example.

In future work, we will explore more sophisticated models that should provide even greater predictive power. For example, the individual models used in the aggregation-of-individuals method did not incorporate social inputs to the users beyond their own previous behavior. The computational mechanics framework allows for the incorporation of inputs via either dynamic random field-based [7] or transducer-based [6, 9] models of a user's behavior. Such an extension could eliminate the need for the adjustment to the aggregation-of-individuals predictor needed to translate the model's output to a prediction.

This work highlights that in building predictive models for complex social systems, a multi-level view of the system under consideration often leads to improved predictive ability. Thus, in the predictive problem considered in this chapter, influencers who track potential user engagement can use complexity modeling to make better informed decisions.

Chapter 5: The Computational Landscape of User Behavior on Social Media

5.1 Introduction

The current decade has been marked by an increasing availability of high-resolution, heterogeneous data sets capturing human behavior in both real-world and digital environments [110–113]. This has made possible, for the first time, large scale investigations into human behavior across diverse groups of individuals. Of such phenomena, human communication patterns are perhaps one of the most well-studied. Such studies have included written correspondences [114], email correspondences [115, 116], and call/SMS records [117, 118]. The complexities of these behavioral patterns include heavy tails, seasonality, and burstiness. This is certainly still an active field of research, and many authors have called into question whether the observed patterns are truly universal characteristics of human behavior or epiphenomena of the methods used in data collection and analysis [91, 119, 120].

The standard model for human communication patterns treats the observed behavior as the realization from some sort of point process. Typically, the point process is taken to be a renewal process, where the observed behavior is completely

specified by a distribution over the times between activity. To account for the complex properties of human behavior enumerated above, the interevent distribution is specified to have a heavy right tail, which naturally gives rise to burstiness. The authors of [115, 116] develop a refinement of this model which incorporates seasonality by allowing an individual to pass between passive and active states, where the behavior within the active state is governed by a Poisson process. Further refinements of this model allow the activity during the active periods to follow non-Poissonian dynamics [120].

Our approach differs in at least four ways from the standard approach just described. First, motivated by the field of computational mechanics [4], we define our models explicitly in terms of a *predictive* representation of the observed behavior. Second, we do not assume the behavior of individuals only depends on the time between actions. Third, we seek to understand the behavior locally in time, where locality is defined around periods of activity. In this sense, our approach is most similar to [115, 116], except that we do not assume that the time local behavior follows a Poisson process. Finally, we explicitly incorporate the interactive aspect of online social media services, something missing from much of the work on modeling human interevent distributions, with [104] as a notable exception.

In [121], the authors set out to elucidate the structural properties of stochastic processes using tools from computational mechanics. To do so, they restricted their investigation to the subset of stochastic processes that are *finitary*, that is, those stochastic processes that admit a representation with a finite number of causal states [122]. In this work, instead of elucidating all possible finitary models, we

approach the problem from the opposite direction: we seek to trace the computational landscape of human behavior in digital environments by discovering the finitary models present in user behavior, and then investigate their computational structure.

We consider four models for user behavior on social media. Figure 5.1 provides a schematic representation of these models. The most general model (a) assumes that a user's future behavior is influenced by both their past behavior and the past behavior of their social network. Models (b) and (c) are two restrictions of this model, the former where we assume that user's future behavior is only influenced by their past behavior, and the latter where we assume that the user's future behavior is only influenced by the past behavior of their social network. Finally, model (d) corresponds to the case where the user's behavior is entirely explained by the time of day.

In the rest of the chapter, we proceed as follows. First, in Section 5.2 we motivate and develop the four models just presented, and propose methods for inferring them. In Section 5.3.1 we explore the descriptive performance of these models on a real world data set derived from 15K users on the microblogging platform Twitter. In Section 5.3.2, we investigate the structure of the models present amongst the users in our data set, and discuss the implications of these models. In Section 5.3.4, we present case studies of user behavior, and the insights gained from the different models. Finally, we conclude with the implications of our present work for the study of human communication patterns.

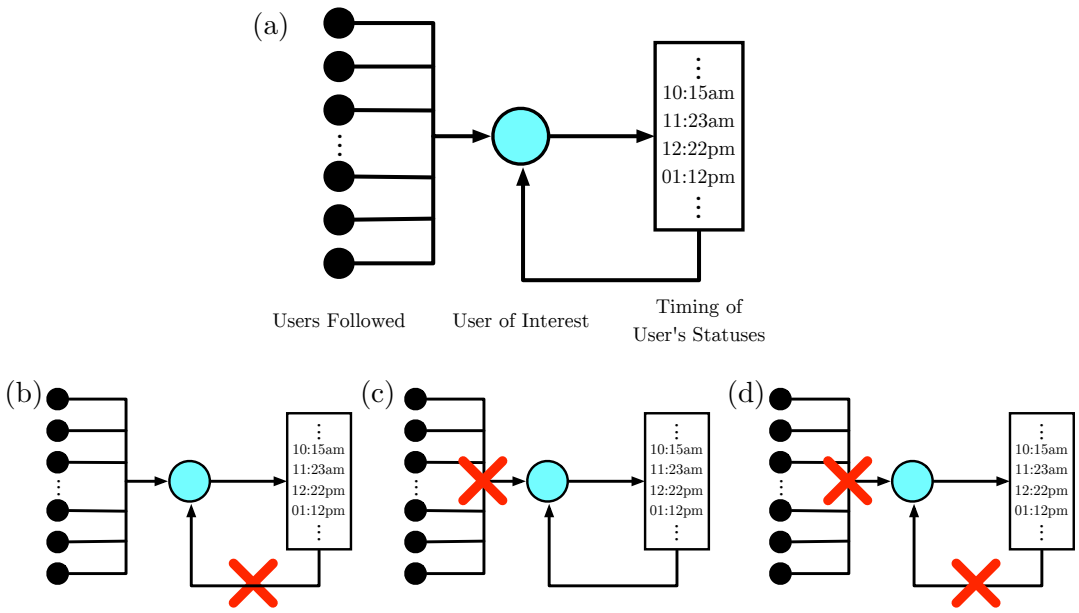


Figure 5.1: A schematic representation of the classes of models that we consider in this chapter. (a) The most general case, where the user's observed behavior is influenced by their social inputs and their own past behavior. (b) The self-driven case, where the user's behavior only depends on their past behavior. (c) The socially-driven case, where the user's behavior only depends on their social inputs. (d) The seasonally-driven case, where the user's behavior can be attributed to the time-of-day.

5.2 Methodology

5.2.1 User Behavior as a Discrete-Time Point Process

Consider the behavior of a user on a social media service. At any given time instant, a user either posts to the social media service or not. Thus, the user's behavior may be modeled as a point process, where events correspond to posts. A naive model of the user's behavior might assume that they are equally likely to use the service during any time instant. Under this model, the time between uses is exponentially distributed, and their activity pattern would correspond to a

realization from a Poisson process. However, human communication patterns are known to be exhibit non-trivial complexities not accounted for by this model [91, 123], and thus more flexible models are required. In the following sections, we present three models that capture the observed complexity of human behavior in very different ways: a seasonally-driven model where the user's behavior is accounted for by time-of-day; a self-driven model where a user's behavior results from self-feedback; and a socially-driven model where a user's behavior results from both social- and self-feedback.

In practice, the information about human behavior on digital services is reported in seconds. Because we are interested in human-scale interactions between a user, their inputs, and the social media service, this time resolution is too fine. We begin by discretizing time into intervals of length δ. We then ask if, during an interval $[(t-1)\delta, t\delta)$, the user was active. We denote this value for a user v by $X_t(v)$ and have

$$X_t(v) = \begin{cases} 1 & : \text{user } v \text{ active during } [(t-1)\delta, t\delta) \\ 0 & : \text{otherwise} \end{cases} \tag{5.1}$$

The choice of δ specifies the time scale of interest. For example, if we take $\delta = 1$ day, then the process $\{X_t(v)\}$ captures the weekly patterns of behavior that the user exhibits. If instead we take $\delta = 1$ hour, then $\{X_t(v)\}$ captures the daily patterns of the user. In this chapter, we will take $\delta = 10$ minutes, because we are interested in the short-timescale behavior of user behavior and user-user interaction. However, it is important to note that there is no single 'correct' resolution when considering the

behavior of a point process, and a multi-timescale analysis may be appropriate [124]. Moreover, different time resolutions may be more or less appropriate for different users. Figure 5.2 demonstrates the activity patterns $\{X_t(v)\}$ of six users at the 10 minute resolution, represented as a rastergram. Each row of the rastergram corresponds to a single day of activity, and each column of the rastergram corresponds to a ten minute window within a single day. A point occurs in the rastergram when $X_t(v) = 1$ for that day and time.

In a social media setting, a user has access to information provided by other users on the service. For example, a user might passively examine the messages generated by other users they follow, observe a particular form of communication directed at them, or actively investigate a keyword or topic. Generically, we will denote the inputs to a user as $Y_t(v)$. We will generally assume that the inputs to the user can be mapped to a finite alphabet $\mathcal{Y}$. As an example, if we consider $Y_t(v)$ to correspond to whether or not the user v receives a mention during the time interval $[(t-1)\delta, t\delta)$, then we take $\mathcal{Y} = \{0, 1\}$, where $y = 0$ corresponds to no mention during that time interval, and $y = 1$ corresponds to one or more mentions.

Our goal in this chapter is the develop several contrasting models of a user's observed behavior $\{X_t(v)\}$. We take a predictive view of modeling, where we seek to infer the probability that the user engages with the social media service, given their past history of engagement and the past history of their inputs. Let $X_{-\infty}^{t-1}(v) = (\ldots, X_{t-2}(v), X_{t-1}(v))$ be the past behavior of user v, and let $X_t^{\infty}(v) = (X_t(v), X_{t+1}(v), \ldots)$ be the future behavior of the user. Similarly, let $Y_{-\infty}^{t-1}(v) = (\ldots, Y_{t-2}(v), Y_{t-1}(v))$ and $Y_t^{\infty}(v) = (Y_t(v), Y_{t+1}(v), \ldots)$ be the past and future val-

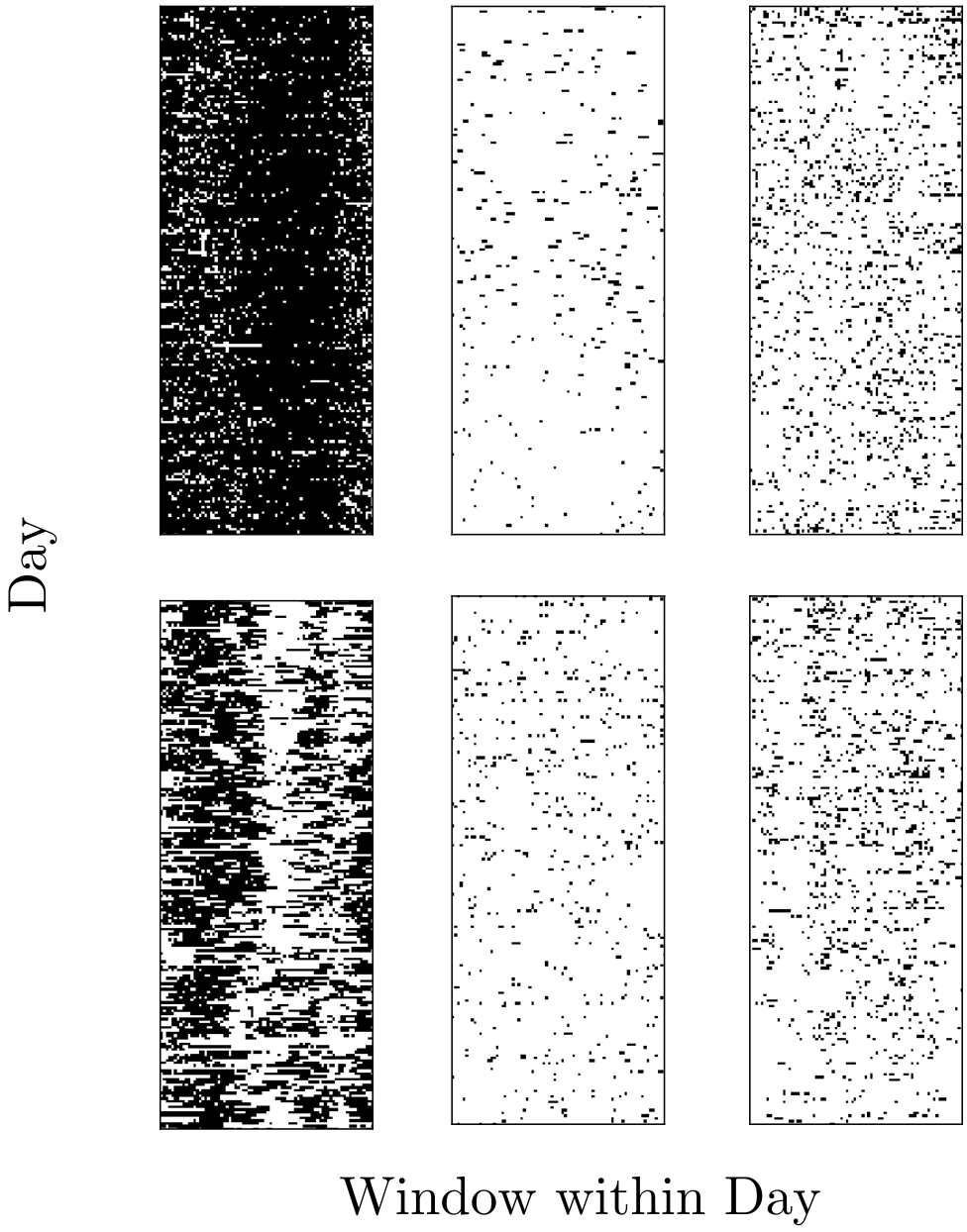


Figure 5.2: The activity patterns for six users on Twitter represented as rastergrams. Each row of a rastergram corresponds to a day, and each column corresponds to a δ length window within a day.

ues of the user's inputs, considering t as the present. Then we are interested in determining

$$P(X_t^{\infty}(v) \mid X_{-\infty}^{t-1}(v) = x_{-\infty}^{t-1}, Y_{-\infty}^{t-1} = y_{-\infty}^{t-1}(v)), \tag{5.2}$$

the distribution over user v's behavior starting from time t, given their own past behavior and the past behavior of their inputs. For ease of presentation, in the following sections, we drop the dependence on v in the notation, but emphasize that for each user v, we assume a unique model for that user's behavior.

5.2.2 Seasonally-Driven Model: Inhomogenous Bernoulli

Thus far, we have specified our model of human behavior in terms of a discrete-time point process: the observed behavior of the user is either active or quiescent during any given interval of time. One of the simplest models that can capture some of the complexity of human behavior is a renewal process [125, 126]. From this perspective, the activity of the user is taken to occur at random times, with the time between occurrences governed by a distribution over the interarrival times. For example, if we take the interarrival distribution to be geometric with parameter p, then the renewal process is a Bernoulli process, the discrete-time analog of a Poisson process. Typically, the interarrival distribution is taken to have a long tail, to capture the fact that human behavior tends to be bursty, with long periods of quiescence punctuated by periods of high activity. See Figure 5.2 for examples of users who exhibit such behavior. Popular distributions for the interarrival times

include log-normal, power law, and stretched exponential distributions [91].

Due to the inherent seasonality in human behavior, time-homogeneous renewal process-based models are almost certainly misspecified. For example, a typical user on Twitter will be more likely to be active during the daylight hours in their geographic area than during the nighttime hours. This fact may explain the long tails typically observed in studies of the activity patterns of humans [127]. Moreover, we see such daily and weekly seasonality patterns in the aggregate behavior of users on Twitter. Figure 5.3 demonstrates the observed seasonality in the aggregated number of tweets issued by the users studied in this chapter. Because of this, we consider a time-inhomogeneous point process model for a user's observed activity, where the probability a user is active during any time interval is independent of their previous activity and the activity of their inputs, and varies smoothly with time,

$$P(X_t = 1 \mid X_{-\infty}^{t-1} = x_{-\infty}^{t-1}, Y_{-\infty}^{t-1} = y_{-\infty}^{t-1},) = p(t) \tag{5.3}$$

Moreover, we assume that $p(t)$ is periodic, $p(t + \tau) = p(t)$, with τ chosen such that for a coarsening interval δ, $\tau\delta = 1$ week.

We estimate the individual seasonality $p(t)$ for each user via a Generalized Additive Model (GAM) [128]

$$\text{logit}(p(t)) = \beta_0 + f(t) \tag{5.4}$$

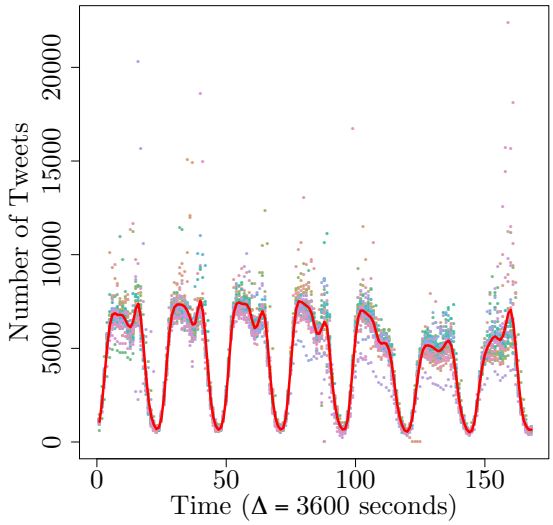


Figure 5.3: The total number of tweets issued per hour by the 15000 users considered in this chapter over one week periods. Each color corresponds to one of 32 weeks. The solid red line corresponds to the weekly seasonality, estimated by averaging across the 32 weeks.

where

$$\text{logit}(x) = \log\left(\frac{x}{1-x}\right) \tag{5.5}$$

using the `mgcv` package in R. Figure 5.4 demonstrates the observed behavior of several users, along with their estimated activity probabilities $p(t)$.

5.2.3 Self-driven Model: The ϵ-machine

The previous model assumes that the user's activity during any time interval is independent of their activity during other time intervals, and accounts for the seasonality and bursting observed in a user by allowing the probability of their activity to vary across time. Alternatively, a user might exhibit burstiness due to

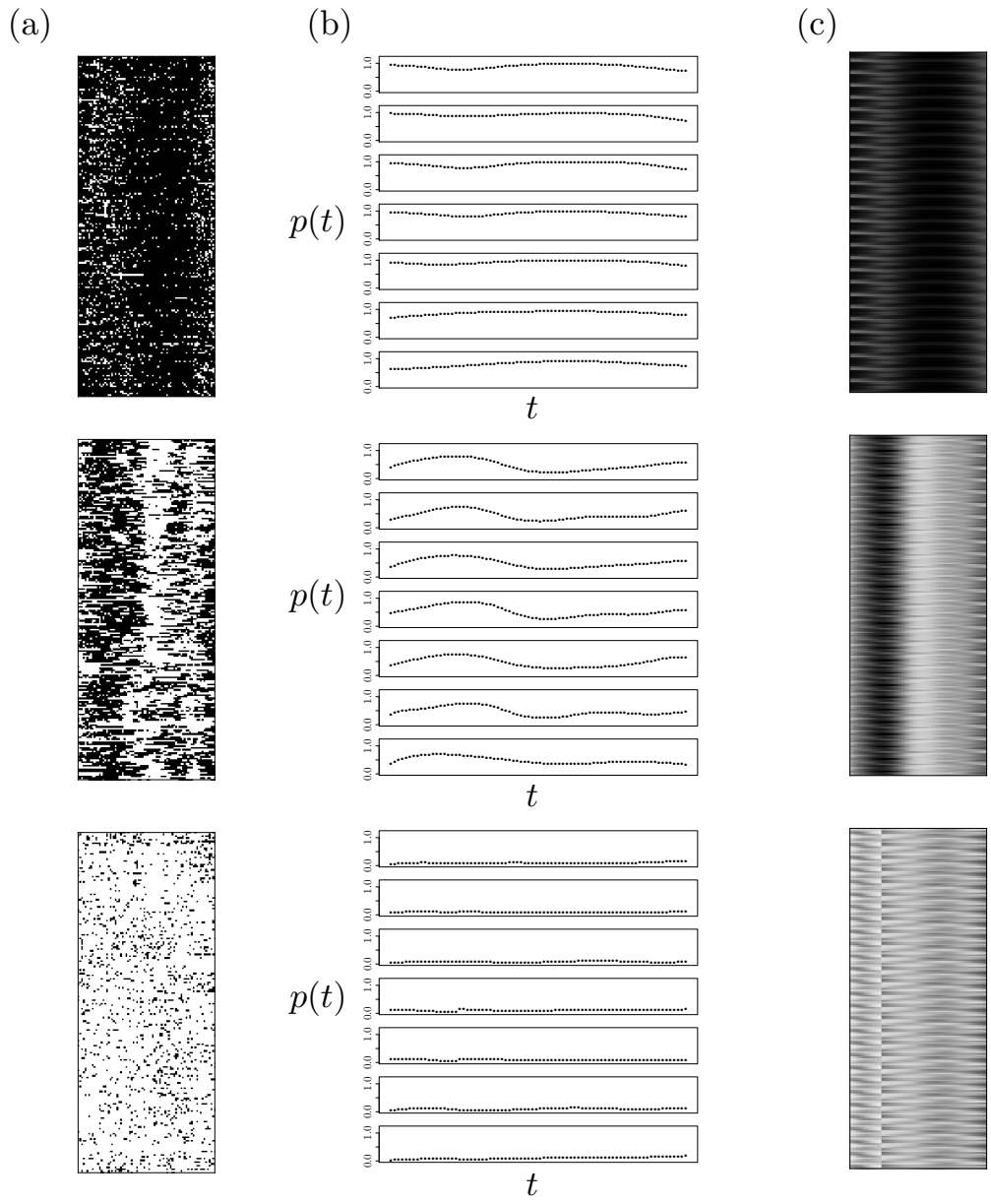


Figure 5.4: (a) Rastergram representation of the activity of three users on Twitter over a 32 week period. (b) The expected activity of the same three users. Each panel corresponds to the expected activity by day-of-week, from Monday to Sunday. (c) The expected activity from (b), laid out in the same format as the rastergram. Note that the color scale for each panel is taken from 0 to $\max_t p_v(t)$ for each user v to make the seasonality in the activity patterns more obvious.

self-excitation. As an example, the user might be isolated from the devices they use to interact with the social media service, which would lead to a period of quiescence.

Then, upon regaining access to their devices, they might use the service, which could lead to a self-excitation to continue using the service.

This sort of behavior motivates an autoregressive model for the user's behavior, where their behavior in the future is determined by their past behavior. That is, the probability that they behave a certain way in the future starting at time t is determined by how they behaved up until time t,

$$\begin{aligned} P(X_t^{\infty} \mid X_{-\infty}^{t-1} = x_{-\infty}^{t-1}, Y_{-\infty}^{t-1} = y_{-\infty}^{t-1},) \\ = P(X_1^{\infty} \mid X_{-\infty}^{0} = x_{-\infty}^{0}). \end{aligned} \tag{5.6}$$

This model assumes that the users behavior is conditionally stationary [129]. For human behavior, this assumption may not hold in general, and thus care must be taken in applying this model with actual data. We address this in Section 5.2.5, where we specify our procedure for day-casting user behavior. Previous work has found this model to perform well with many users on social media platforms such as Twitter [47, 64, 104].

Computational mechanics provides a framework for elegantly handling stochastic processes governed by a dynamic such as (5.6) [4]. We now give a brief introduction to computational mechanics, which will be used for both the self-driven and the socially-driven model. In the time series case, computational mechanics provides the unique, minimally complex, maximally predictive model of a discrete state, discrete time stochastic process $\{X_t\}_{t\in\mathbb{Z}}$ over the alphabet $\mathcal{X}$. The insight of computational mechanics is that when considering the predictive distribution (5.6), it is typically

more useful to consider a *statistic* of the past $x_{-\infty}^{t-1}$ rather than the entire past itself. It can be shown that the unique minimal sufficient predictive statistic of the past $X_{-\infty}^{t-1}$ for the future X_t^{∞} of a conditionally stationary stochastic process is the equivalence class over predictive distributions. For two pasts $u_{-\infty}^{t-1}$ and $v_{-\infty}^{t-1}$, we define an equivalence relation such that

$$u_{-\infty}^{t-1} \sim v_{-\infty}^{t-1} \implies \tag{5.7}$$

$$P(X_t^{\infty} \mid X_{-\infty}^{t-1} = u_{-\infty}^{t-1}) = P(X_t^{\infty} \mid X_{-\infty}^{t-1} = v_{-\infty}^{t-1}) \tag{5.8}$$

as probability mass functions. In words, two pasts are equivalent if they result in statistically equivalent futures. Using this equivalence relation, we can define equivalence classes over pasts p such that

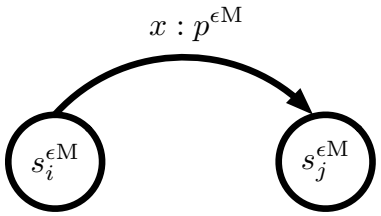
$$\begin{aligned}[p] = \{x_{-\infty}^{t-1} \in \mathcal{X}^- : P(X_t^{\infty} \mid X_{-\infty}^{t-1} = x_{-\infty}^{t-1}) \\ = P(X_t^{\infty} \mid X_{-\infty}^{t-1} = p)\}\end{aligned} \tag{5.9}$$

where $\mathcal{X}^- = \mathcal{X} \times \mathcal{X} \times \ldots$ is the set of all semi-infinite pasts. In words, for each possible predictive distribution, we choose a candidate past p, and $[p]$ represents all pasts which induce this predictive distribution. We can thus think of p as a particular past, or as the label for this class of pasts. Typically, we will take the second perspective. We define our statistic $\epsilon : \mathcal{X}^- \to \mathcal{S}$ as mapping a past into the

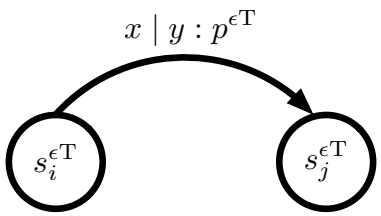
equivalence class for that past,

$$\epsilon(X_{-\infty}^{t-1}) = [X_{-\infty}^{t-1}]. \tag{5.10}$$

We can think of ϵ as partitioning the set of all pasts $\mathcal{X}^-$ based on the conditional futures they induce. The combination of the equivalence classes as well as the allowed transitions between them is called the ϵ-machine for the process $\{X_t\}_{t\in\mathbb{Z}}$. For a stochastic process with a finite number of predictive equivalence classes, the ϵ-machine may be represented as deterministic finite automata, where the states of the automata correspond to the predictive equivalence classes, and the transitions between states are determined by the outputs $x \in \mathcal{X}$. A demonstration of a portion of such a representation is given in Figure 5.5(a). Generically, a stochastic process may have infinitely many predictive equivalence classes, in which case the ϵ-machine representation corresponds to a deterministic automata with infinitely many states. For each time t, we can associate the current past $X_{-\infty}^{t}$ with its mapping under the equivalence relation $S_t^{\epsilon\mathrm{M}} = \epsilon(X_{-\infty}^{t})$. Thus, the equivalence relation induces a new stochastic process $\{S_t^{\epsilon\mathrm{M}}\}_{t\in\mathbb{Z}}$, called the *causal state process*. The causal state process has many favorable properties. For example, $\{S_t^{\epsilon\mathrm{M}}\}_{t\in\mathbb{Z}}$ is Markov, regardless of whether $\{X_t\}_{t\in\mathbb{Z}}$ is. Moreover, the causal state process at time t shields the future of the process from its past, *i.e.* $X_{t+1}^{\infty} \perp X_{-\infty}^{t} \mid S_t^{\epsilon\mathrm{M}}$. This motivates the name *causal* state process.



(a) The causal state transition structure associated with an ϵ-machine.



(b) The causal state transition structure associated with an ϵ-transducer.

Figure 5.5: Transitions between ϵ-machine/transducer causal states. Each transition is labeled by the marginal/joint emission symbol, as well as the transition probability $p^{\epsilon \mathrm{M/T}}$.

5.2.4 Socially-driven Models: The ϵ-transducer

The previous two models assume that either the user is driven seasonally by time-of-day type influences, or that the user is self-driven. On a social media service, we expect a user to interact with other users, and therefore we would expect the user's behavior to be associated with the behavior of those users. For example, a user might become more likely to tweet if they have recently been mentioned by another user. Such social associations are captured by the social inputs $\{Y_t(v)\}_{t \in \mathbb{Z}}$ of a user v. In particular, we will focus on the mention history of the user,

$$Y_t(v) = \begin{cases} 1 & : \text{mention of } v \text{ in } [(t-1)\delta, t\delta) \\ 0 & : \text{otherwise} \end{cases} . \tag{5.11}$$

That is, $Y_t(v)$ corresponds to whether or not the user v received any mentions during the time window of length δ indexed by t.

We take the modeling perspective where the user acts as a *transducer*, mapping

their own past behavior and the past behavior of their social inputs into their future behavior. More explicitly, as with the self-driven example, we seek the minimally complex, maximally predictive model for the user's behavior. Again, computational mechanics provides such a model via the ϵ-transducer [6, 8, 9]. The main insight is the same as for the ϵ-machine: we define an equivalence relationship over joint input-output pasts such that two pasts are equivalent if they induce the same predictive distribution over the future output. For two joint input-output pasts $(r_{-\infty}^{t-1}, u_{-\infty}^{t-1})$ and $(s_{-\infty}^{t-1}, v_{-\infty}^{t-1})$, where $u_{-\infty}^{t-1}, v_{-\infty}^{t-1} \in \mathcal{X}^-$ and $r_{-\infty}^{t-1}, s_{-\infty}^{t-1} \in \mathcal{Y}^-$, we define the equivalence relation such that

$$\begin{aligned} &(r_{-\infty}^{t-1}, u_{-\infty}^{t-1}) \sim (s_{-\infty}^{t-1}, v_{-\infty}^{t-1}) \implies \\ &P(X_t^{\infty} \mid Y_t^{\infty}, X_{-\infty}^{t-1} = u_{-\infty}^{t-1}, Y_{-\infty}^{t-1} = r_{-\infty}^{t-1}) \\ &= P(X_t^{\infty} \mid Y_t^{\infty}, X_{-\infty}^{t-1} = v_{-\infty}^{t-1}, Y_{-\infty}^{t-1} = s_{-\infty}^{t-1}). \end{aligned} \tag{5.12}$$

In [9], the authors show that this equivalence relation is the same as the equivalence relation originally given in [6, 8]. That equivalence relation is specified in terms of two conditions on the joint pasts: equivalence in terms of one-step-ahead predictive distributions for the output, also known as *weak prescience* [4], and determinism / unifilarity on appending input-output pairs (a, b) to the joint pasts. More formally,

the alternative equivalence relation is

$$
\begin{aligned}
&(r_{-\infty}^{t-1}, u_{-\infty}^{t-1}) \sim (s_{-\infty}^{t-1}, v_{-\infty}^{t-1}) \implies \\
&(i)\ P(X_t \mid Y_t, X_{-\infty}^{t-1} = u_{-\infty}^{t-1}, Y_{-\infty}^{t-1} = r_{-\infty}^{t-1}) \\
&= P(X_t \mid Y_t, X_{-\infty}^{t-1} = v_{-\infty}^{t-1}, Y_{-\infty}^{t-1} = s_{-\infty}^{t-1}) \\
&(ii)\ P(X_{t+1} \mid Y_{t+1}, X_{-\infty}^{t} = u_{-\infty}^{t-1} b, Y_{-\infty}^{t} = r_{-\infty}^{t-1} a) \\
&= P(X_{t+1} \mid Y_{t+1}, X_{-\infty}^{t} = v_{-\infty}^{t-1} b, Y_{-\infty}^{t} = s_{-\infty}^{t-1} a)
\end{aligned}
\tag{5.13}
$$

where in condition (ii) we consider only input-output symbols (a, b) admissible based on the pasts $(r_{-\infty}^{t-1}, u_{-\infty}^{t-1})$ and $(s_{-\infty}^{t-1}, v_{-\infty}^{t-1})$. From this equivalence relationship, it is clear that if the next output Y_t is independent of the next input X_t given the joint input-output past $(Y_{-\infty}^{t-1}, X_{-\infty}^{t-1})$, then the equivalence relationship becomes

$$
\begin{aligned}
&(u_{-\infty}^{t-1}, r_{-\infty}^{t-1}) \sim (v_{-\infty}^{t-1}, s_{-\infty}^{t-1}) \implies \\
&P(X_t^{\infty} \mid X_{-\infty}^{t-1} = u_{-\infty}^{t-1}, Y_{-\infty}^{t-1} = r_{-\infty}^{t-1}) \\
&= P(X_t^{\infty} \mid X_{-\infty}^{t-1} = v_{-\infty}^{t-1}, Y_{-\infty}^{t-1} = s_{-\infty}^{t-1}).
\end{aligned}
\tag{5.14}
$$

That is, the equivalence relationship no longer depends on the future input. Unless otherwise stated, we assume the conditional independence between the future input-output pair, and thus use this equivalence relationship. We do so for several reasons. First, for the problem at hand, it seems reasonable to assume that, for sufficiently small time windows, the behavior of a user is independent of their immediate social input, conditional on their own past behavior. That is, an individual requires a finite amount of time to respond to any social input, and thus their immediate

present behavior is unaffected by their immediate social input. Second, and more pragmatically, in order to perform prediction of $\{X_t\}_{t\in\mathbb{Z}}$ without also requiring a model for $\{Y_t\}_{t\in\mathbb{Z}}$, the assumption of conditional independence is required.

In either case, at any given time t, the equivalence relation defines a mapping ϵ from the current joint past $(Y^t_{-\infty}, X^t_{-\infty})$ to its equivalence class $S^{\epsilon\mathrm{T}}_t = \epsilon((Y^t_{-\infty}, X^t_{-\infty}))$, inducing the *channel causal state* process $\{S^{\epsilon\mathrm{T}}_t\}_{t\in\mathbb{Z}}$. Transitions between channel causal states occur on joint input-output pairs, and thus an ϵ-transducer, like an ϵ-machine, can be represented as a deterministic automata. The associated representation is given in Figure 5.5(b).

If the user's behavior is purely driven by their social inputs, then we can consider a special case of (5.14) where their future behavior is independent of their past behavior, conditional on their past social inputs. That is, the associated predictive distribution reduces to

$$\begin{aligned} P(X^\infty_t \mid X^{t-1}_{-\infty} = x^{t-1}_{-\infty}, Y^{t-1}_{-\infty} = y^{t-1}_{-\infty},) \\ = P(X^\infty_1 \mid Y^0_{-\infty} = y^0_{-\infty}). \end{aligned} \tag{5.15}$$

We call this special case the self-memoryless transducer to emphasize that the user maintains a memory of their past inputs while forgetting their past outputs. In this case, the equivalence relation (5.14) reduces to

$$\begin{aligned} & r^{t-1}_{-\infty} \sim s^{t-1}_{-\infty} \implies \\ & P(X^\infty_t \mid Y^{t-1}_{-\infty} = r^{t-1}_{-\infty}) = P(X^\infty_t \mid Y^{t-1}_{-\infty} = s^{t-1}_{-\infty}). \end{aligned} \tag{5.16}$$

5.2.5 Data Collection and Pre-processing

The activity of the 15K users was collected over a 49 week period, from 6 June 2014 to 15 May 2015. After data cleaning to account for outages in the data collection, 44 weeks of data were generated. We did not include the quiescent users in our analysis. As described above, the self-driven and socially-driven models assume that a user's behavior can be modeled as a conditionally stationary stochastic process, where the distribution over futures is independent of the time index conditional on the observed past of the user or the observed joint input / output past, respectively. In order to make this assumption approximately true, we 'daycast' the time series associated with each user as follows. For a user, we determine their native time zone (as self-reported on Twitter), and window their activity to be between 9 AM and 10 PM during their local time. We take this time window to capture the waking hours of a typical individual.

For this study, we split the 44 weeks of data into 28 weeks of training data and 16 weeks of testing data. The training data is used to select and infer the models, as we describe in the next section. The testing data is used for the comparison of these models in terms of their descriptive performance. This train/test split is performed to ensure that we obtain unbiased estimates of how the models perform for each user.

5.2.6 Model Inference and Selection

For the seasonally-driven model, the only model parameter associated with each user is the smoothing parameter for the splines used to estimate the non-parametric term $f(t)$ in (5.4). This parameter is chosen using generalized cross validation [130] on the 28 weeks of training data.

For both the ϵ-machine and ϵ-transducer models, we use the Causal State Splitting Reconstruction (`CSSR`) algorithm [46] to infer the models from data. We describe the algorithm for ϵ-machine reconstruction here. The modification of `CSSR` for ϵ-transducer reconstruction is provided in Appendix A. `CSSR` works via a two-phase process that takes advantage of the fact if a set of states is weakly prescient and deterministic, then it is prescient [48]. The first phase of the algorithm determines a set of weakly prescient states. It begins by assuming that all histories induce the same one-step-ahead predictive distribution. This is equivalent to assuming the process is independent and identically distributed over the alphabet $\mathcal{X}$, or to grouping together all histories into a candidate causal state represented by the suffix $*\lambda$, where λ is the null symbol. At each successive step, the histories in each candidate causal state are grown by one symbol into the past, and a statistical test of size α is performed to check whether the history's one-step-ahead predictive distribution matches its parent state. If not, the history is compared against all of the remaining candidate causal states. Finally, if the history does not agree with any of the candidate causal states, it is split into a new candidate casual state. Such potential splitting is performed for $L = 1, 2, \ldots, L_{\max}$ where $L_{\max}$ is the maximum

history length used.

At the end of the first phase, a set of candidate causal states has been determined. The histories in each candidate causal state have statistically equivalent predictive distributions within the causal states, and statistically distinct predictive distributions between the causal states. The true causal states have this property, in addition to being unifilar. The second phase of `CSSR` first removes transients and then splits the candidate causal states to ensure that they are unifilar. That is, for a candidate causal state, the transitions from that state are determined by growing each history in that state forward by a single symbol. By unifilarity, all of the histories should transition to the same state upon appending a particular emission symbol. If histories transition to more than one state, those histories are split into new candidate causal states. This procedure is repeated until no new splits occur. Because this procedure only ever refines the candidate causal states, the states returned retain the property of weak prescience, while gaining the property of unifilarity, and thus gain the property of causal states.

The `CSSR` algorithm, in both the ϵ-machine and ϵ-transducer case, requires the specification of α, the size of the hypothesis test used first phase of `CSSR`, and $L_{\max}$, the maximum history length used in determining the candidate causal states. The size α controls to probability of splitting a history from a candidate causal state when it should not be split, and thus indirectly controls the number of causal states associated with the model. We fix α at 0.001 for all examples in this chapter. The maximum history length $L_{\max}$ directly balances between the flexibility of the model and the precision with which the probabilities may be estimated. As an example,

suppose a maximum history length $L_{\max}$ is sufficient to resolve the causal states. In the extreme case that each history of length $L_{\max}$ specifies a unique predictive distribution (an order $L_{\max}$ Markov model), then the model would result in $|\mathcal{X}|^{L_{\max}}$ causal states. However, as we increase $L_{\max}$, we also necessarily decrease the number of examples of each history used to estimate the predictive distribution. This can result in spurious splitting of histories.

We use a cross-validation [130] approach to choose the appropriate $L_{\max}$ for each user. In particular, for each user, we define the empirical total variation (ETV) distance between their observed behavior and the model predictions over an index set $\mathcal{T}_k$ as

$$\begin{aligned} &\mathrm{ETV}(L_{\max}, k) \\ &= \frac{1}{|\mathcal{T}_k|} \sum_{t \in \mathcal{T}_k} \left(\frac{1}{2} \sum_{x \in \mathcal{X}} \left| \delta_{X_v(t),x} - p_v^{(L_{\max},-k)}(x,t) \right| \right) \end{aligned} \tag{5.17}$$

where $\delta_{x',x}$ is the Kronecker delta and $p_v^{(L_{\max},-k)}(x,t)$ is the probability of observing outcome x at time t using the model inferred with all of the data except that from the index set $\mathcal{T}_k$. In the binary case, (5.17) reduces to

$$\begin{aligned} &\mathrm{ETV}(L_{\max}, k) \\ &= \frac{1}{|\mathcal{T}_k|} \sum_{t \in \mathcal{T}_k} \left| \delta_{X_v(t),1} - p_v^{(L_{\max},-k)}(1,t) \right| \end{aligned} \tag{5.18}$$

Thus, we see that (5.17) quantifies the model performance by comparing the actual outcome for the user to the estimated probability of that outcome using the model.

We choose the index sets $\{\mathcal{T}_k\}_{k=1}^K$ with $K = 5$ at the level of days, such that each index set $\mathcal{T}_k$ corresponds to 39 days of data, with the remaining 157 days used to infer the model, with the index sets disjoint. We can then compute the average of the empirical total variation over the held out sets,

$$\mathrm{ETV}(L_{\max}) = \frac{1}{K}\sum_{k=1}^{K}\mathrm{ETV}(L_{\max}, k), \tag{5.19}$$

and choose $L_{\max}$ to minimize this value. We perform this optimization using $L_{\max}$ from 1 to 6, which for $\delta = 10$ minutes corresponds to a time span between ten minutes and an hour.

5.3 Results

5.3.1 Descriptive Performance Across the Model Classes

We begin by examining the ability of the four models to describe a given user's behavior. To do so, we compute the ETV, as defined by (5.17), between the held out test data and the cross-validated models of each type. This provides us with a measure of how the models generalize to unseen behavior, and thus an indication of how well the models describe a user's behavior. Because the ETV for a given user depends on their overall activity level, we standardize the ETV for a model $\mathcal{M}$ by the ETV for the seasonality model, giving us a score function

$$\mathrm{Score}(\mathcal{M}; S) = \frac{\mathrm{ETV}(S)}{\mathrm{ETV}(\mathcal{M})}. \tag{5.20}$$

Table 5.1: Pairwise comparison between the ϵ-machine (ϵM), self-memoryless ϵ-transducer (ϵT-ML), and self-memoryful ϵ-transducer (ϵT-MF) across the users. Of the users with $\text{Score}(\mathcal{M}; S) > 1$ for both $\mathcal{M}_1$ and $\mathcal{M}_2$, the proportion of users with $\text{Score}(\mathcal{M}_1; S) > \text{Score}(\mathcal{M}_2; S)$.

$\mathcal{M}_1$ / $\mathcal{M}_2$	ϵM	ϵT-ML	ϵT-MF
ϵM	—	0.466	0.164
ϵT-ML	0.534	—	0.278
ϵT-MF	0.836	0.680	—

Recalling that a smaller ETV value indicates a smaller distance between the observed behavior and the model predictions, we see that $\text{Score}(\mathcal{M}; S)$ will be greater than 1 when model $\mathcal{M}$ outperforms the seasonal model, and smaller than 1 otherwise.

The scores across all users for all models are shown in Figure 5.6. The diagonal shows the distribution of scores across the users for each model type. The self- and socially-driven models generally perform better than the seasonal model, with all of the score distributions having heavy tails to the right. We see that the self-memoryful ϵ-transducer performs best, with a score greater than 1 for 82.1% of the users. The self-memoryless ϵ-transducer is next best, with a score greater than 1 for 79.4% of the users. The ϵ-machine has a score greater than 1 for 72% of the users.

We further summarize the pairwise comparisons between the non-seasonal models in Table 5.1. As expected, the self-memoryful ϵ-transducer outperforms both the ϵ-machine and the self-memoryless ϵ-transducer on most users. However, the users are much more equally split between those where the ϵ-machine outperforms the self-memoryless ϵ-transducer (46.6%) and *vice versa* (53.4%).

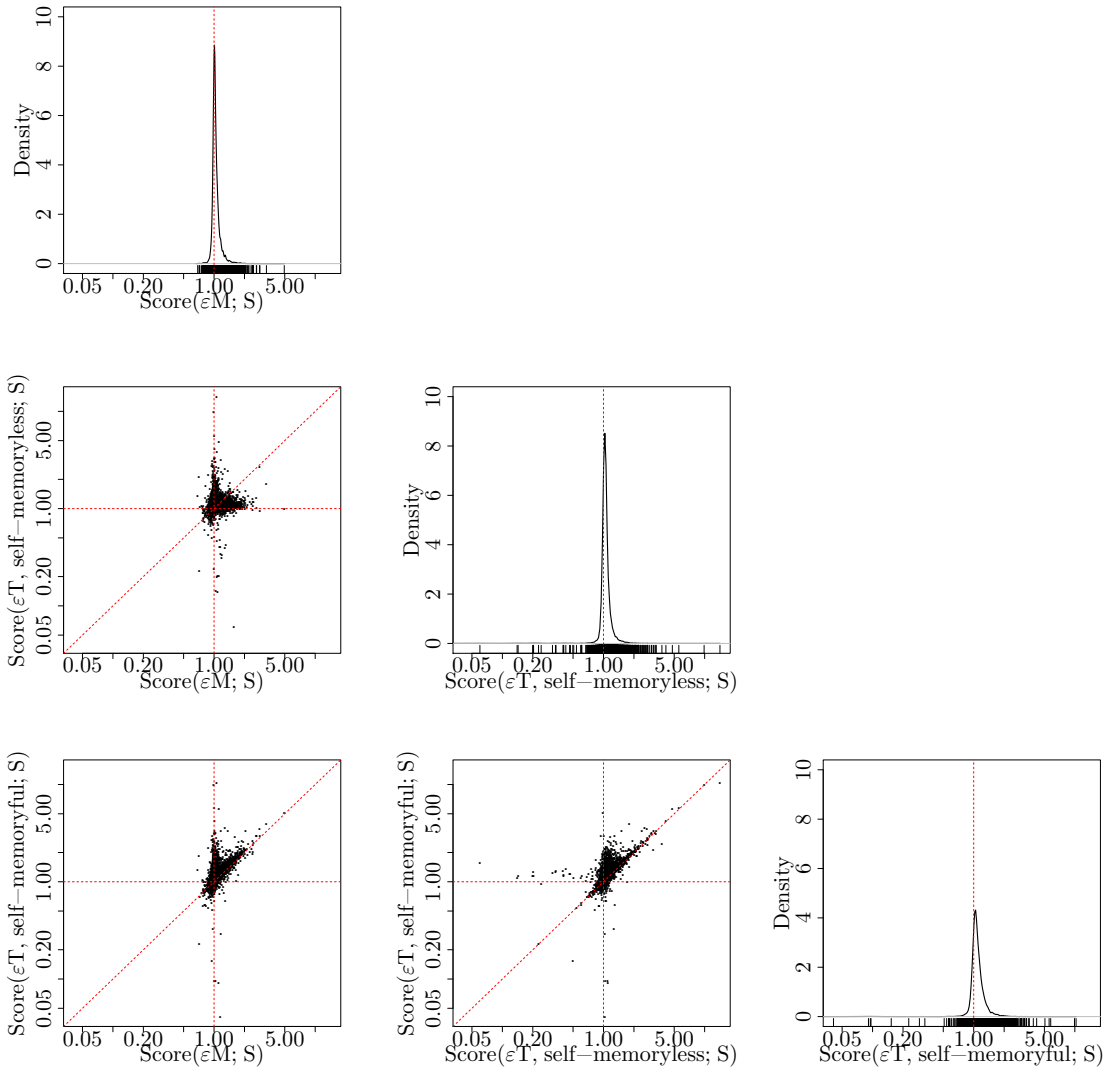


Figure 5.6: The relative descriptive performance of the models for each user on the test set data using the ETV-based score defined by (5.20). The diagonal entries show the density of scores for the ϵ-machine, self-memoryless ϵ-transducer, and self-memoryful ϵ-transducer across the users. The off-diagonal entries compare the scores between the different models.

5.3.2 ϵ-machine Causal Architectures

We next explore the typical ϵ-machine architectures across the users. Figure 5.9(a) shows the distribution of the number of states for each user's ϵ-machine.

The number of causal states for an ϵ-machine gives a rough indication of the complexity of the user's behavior since each causal state indicates a further refinement of the past for predictive sufficiency. In fact, the logarithm of the number of states is called the *topological complexity* of the ϵ-machine [49]. We see that most users are best described by models with a small number of states, with 95% of users having 13 or fewer causal states.

We next consider the general types of stochastic processes captured by many of the ϵ-machines. We find that a large proportion of the users have ϵ-machines which correspond to a generalization of a discrete-time renewal process. Recall that a discrete-time renewal process is a point process such that the lengths $\{N_i\}$ of periods of quiescence (runs of 0s between successive 1s) are independent and distributed according to an inter-arrival distribution $f_0(n) = P(N = n)$ [131]. Equivalently, discrete-time renewal processes can be defined in terms of the survival function $w_0(n) = P(N \geq n)$. Because discrete-time renewal processes are a special case of the more general processes described by (5.6), their ϵ-machine architecture takes on a very particular form [131]. The ϵ-machine for a discrete-time renewal process has a unique start state transitioned to after a period of activity, and transitions after a period of quiescence traverse a chain of states that accumulates the number of time points since since the last active period. We reproduce the generic architecture found amongst the renewal process ϵ-machines in Figure 5.7 (left). This is a special finite state case of the more general architecture for a discrete-time renewal process. In the nomenclature introduced in [131], this is an $\tilde{n}$ eventually Δ_0-Poisson process with characteristic parameters $(\tilde{n}, \Delta_0 = 1)$, where $\tilde{n}$ refers to the number of quiescent

time steps necessary for the ϵ-machine to behave as a Poisson (Bernoulli) process, and the Δ_0 refers to the smallest resolution at which the inter-event times may be coarse-grained and remain geometrically distributed. Such a process has an inter-event distribution

$$f_0(n) = \begin{cases} p_0(n) & : n = 0, \ldots, \tilde{n} \\ f_0(\tilde{n})\lambda_0^{n-\tilde{n}} & : n > \tilde{n} \end{cases} . \tag{5.21}$$

where $\{p_0(n)\}_{n=0}^{\tilde{n}}$ specify the initial $\tilde{n}+1$ values of the inter-event distribution and $\lambda_0 = \frac{1-\sum_{n=0}^{\tilde{n}} p_0(n)}{1-\sum_{n=0}^{\tilde{n}-1} p_0(n)}$. We note that using `CSSR` with finite $L_{\max}$ necessarily results in the reconstruction of finite state ϵ-machines, and thus for an $\tilde{n}$ eventually Δ_0-Poisson processes with $\tilde{n} > L_{\max}$, the inferred ϵ-machine will be an approximation to the longer memory process. In fact, this motivates a particular family of parametric models with parameters $\tilde{n}$ and $\{p_0(n)\}_{n=0}^{\tilde{n}}$ which specify the initial inter-event behavior. We emphasize that this particular family of parametric models was not assumed, but rather discovered via the use of `CSSR`.

A renewal process is specified by a distribution $f_0(n)$ over run lengths $\{N_i\}$ of quiescence. For such a process, the distribution $f_1(m)$ over run lengths $\{M_i\}$ of activity follows a geometric distribution. One could also define a process where these roles are reversed: the distribution $f_1(m)$ over run lengths of activity takes an arbitrary form, and the distribution $f_0(n)$ over run lengths of quiescence follows a geometric distribution. We call such a process a *reverse renewal process*, since the roles of quiescence and activity are reversed. The ϵ-machine for a reverse renewal

process is given in Figure 5.7 (right). In analogy to the $\tilde{n}$ eventually Δ_0-Poisson process, we call this process a reverse $\tilde{m}$ eventually Δ_1-Poisson process, which has the inter-quiescence distribution given by

$$f_1(m) = \begin{cases} p_1(m) & : m = 0, \ldots, \tilde{m} \\ f_1(\tilde{m})\lambda_1^{m-\tilde{m}} & : m > \tilde{m} \end{cases} \tag{5.22}$$

where $\{p_1(m)\}_{m=0}^{\tilde{m}}$ specify the initial $\tilde{m}+1$ values of the inter-quiescence distribution and $\lambda_1 = \frac{1-\sum_{m=0}^{\tilde{m}} p_1(m)}{1-\sum_{m=0}^{\tilde{m}-1} p_1(m)}$.

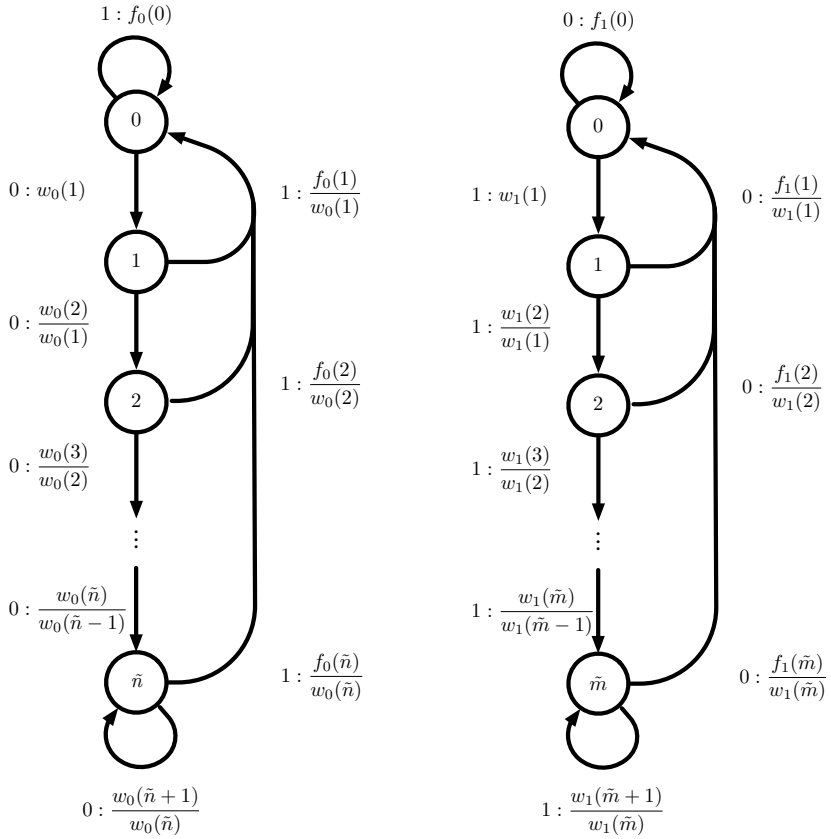


Figure 5.7: The ϵ-machine representations of an eventually Δ_0-Poisson process with characteristic $(\tilde{n}, \Delta_0 = 1)$ (left) and a reverse eventually Δ_1-Poisson process with characteristic $(\tilde{m}, \Delta_1 = 1)$ (right).

More generally, we can define a class of processes such that the distributions

over run lengths of activity *and* quiescence are allowed to deviate from the geometric distribution. We call such a process a mixed renewal process. A mixed renewal process switches between periods of quiescence and activity with probabilities governed by the quiescence length $f_0(n)$ and activity length $f_1(m)$ distributions. The ϵ-machine for a mixed renewal process with geometric tails for both the inter-arrival and inter-quiescence distributions is given in Figure 5.8. We call such a process a mixed $(\tilde{m}_1, \tilde{n}_0)$ eventually (Δ_1, Δ_0)-Poisson process. Again, this class of processes offers another parametric model for user behavior, with parameters $\tilde{n}_0, \tilde{m}_1, \{p_0(n)\}_{n=0}^{\tilde{n}}\}$, and $\{p_1(m)\}_{m=0}^{\tilde{m}}\}$.

Because of the stereotyped architecture of the ϵ-machines for renewal, reverse renewal, and mixed renewal processes, we can easily identify those users whose ϵ-machines have these architectures. An ϵ-machine represents a mixed renewal process if and only if there is precisely one state transitioned to on an x from a state transitioned to on an $x' \neq x$, $x \in \{0, 1\}$. For example, the state transitioned to on a 1 from states transitioned to on a 0 represents the start of a run of 0s. The ϵ-machines for renewal / reverse renewal processes have this property, in addition to only having a single state transitioned to on a 1 / 0. Thus, renewal and reverse renewal processes are a subset of mixed renewal processes. Using these rules, we can identify which users' models correspond to renewal, reverse renewal, or mixed renewal processes. We find that 1881 (13.1%) of the ϵ-machines correspond to (homogeneous) Bernoulli processes, 5408 (37.7%) correspond to two-state renewal / reverse renewal processes, 2713 (18.9%) correspond to pure renewal processes with three or more states, 85 (0.59%) correspond to pure reverse renewal processes with three or more states, and

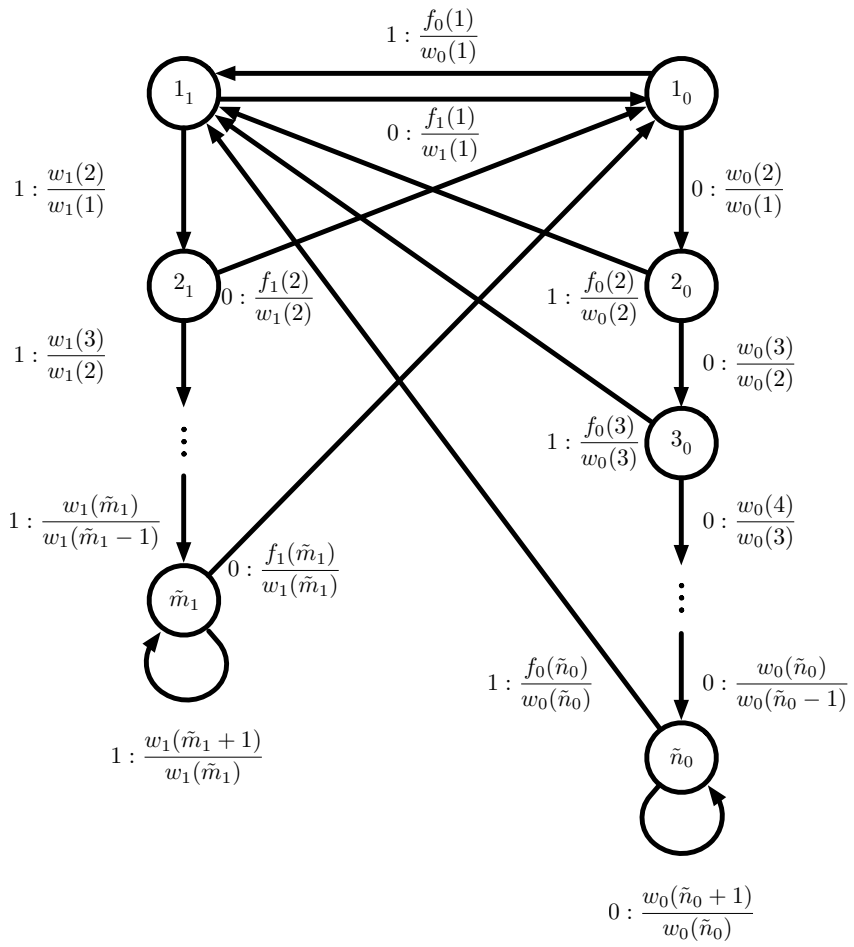
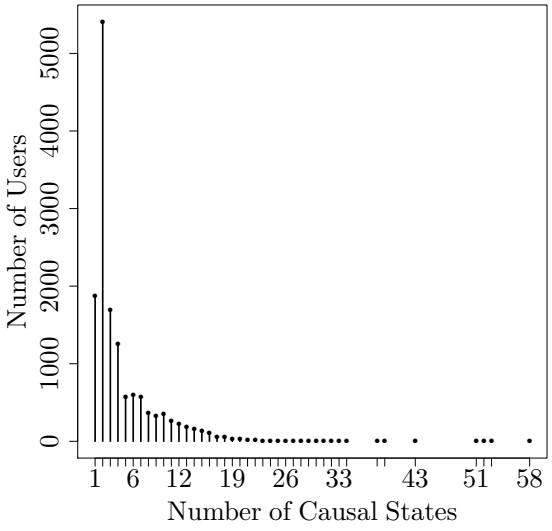


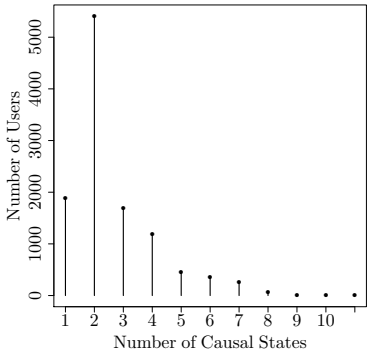
Figure 5.8: The ϵ-machine representation of a mixed eventually (Δ_1, Δ_0)-Poisson process with characteristic $(\tilde{m}_1, \tilde{n}_0, \Delta_1 = 1, \Delta_0 = 1)$.

1250 (8.7%) correspond to mixed renewal processes with four or more states. Thus, Figure 5.9(a) can be seen to be the mixture of mixed renewal and non-mixed renewal users. We decompose the distribution into Figures 5.9(b) and 5.9(c), respectively. This demonstrates that most of the user's with a large number of states are not of the mixed renewal type.

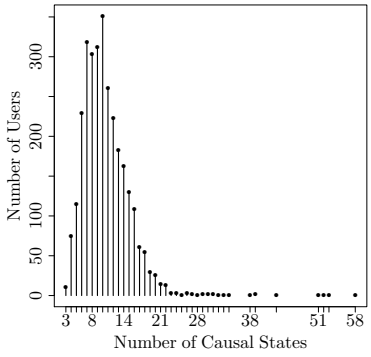
As we have seen, by definition the non-mixed renewal users must be such that their ϵ-machine has one or more states transitioned to on an x from a state transitioned to on an $x' \neq x$. In practical terms, this means that for these users,



(a) For all users.



(b) For mixed renewal users.



(c) For non-mixed renewal users.

Figure 5.9: The distribution of the number of causal across the 14342 active users in the data set. (a) The distribution for all users. (b) The distribution for users with mixed renewal ϵ-machines. (c) The distribution for users with non-mixed renewal ϵ-machines.

Table 5.2: The number of ϵ-machines and ϵ-transducers by their mixed renewal order.

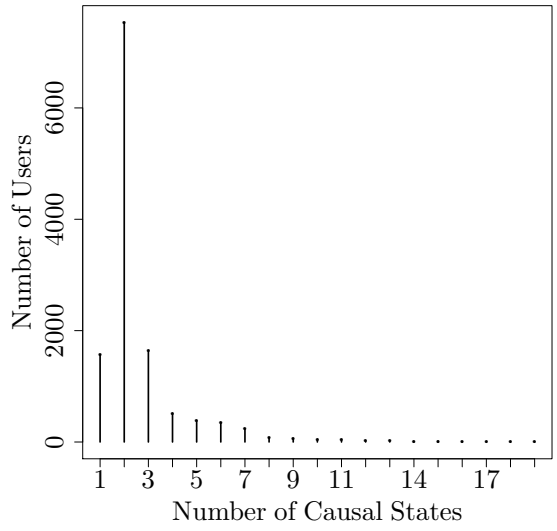
Mixed Renewal Order	# of ϵMs	# of ϵTs
0	1881 (13.1%)	648 (5.1%)
1	9546 (66.7%)	8249 (65.3%)
2	611 (4.3%)	400 (3.2%)
3	493 (3.4%)	309 (2.4%)
4	530 (3.7%)	243 (1.9%)
5	518 (3.6%)	220 (0.02%)
6	134 (1.0%)	147 (0.01%)

knowledge of the time since a user switched from a period of activity / quiescence to a period of quiescence / activity is not sufficient to resolve a causal state. However, in many cases it is sufficient to know the behavior of the user immediately prior to a switch from quiescence to activity or vice versa. For example, a user may behave differently when they have switched from active to passive after just being active compared to after just being passive. These cases correspond to generalizations of the mixed renewal process to higher orders. Table 5.2 summarizes the number of models that correspond to a mixed renewal model of a certain order. For example, a zeroth order mixed renewal model corresponds to a Bernoulli process, a first order mixed renewal process corresponds to the model architecture in Figure 5.8, etc. We see that many of the models resolve to mixed renewal models of higher orders. In total, 95% of the users have an ϵ-machine in agreement with a mixed renewal model of order 6 or smaller.

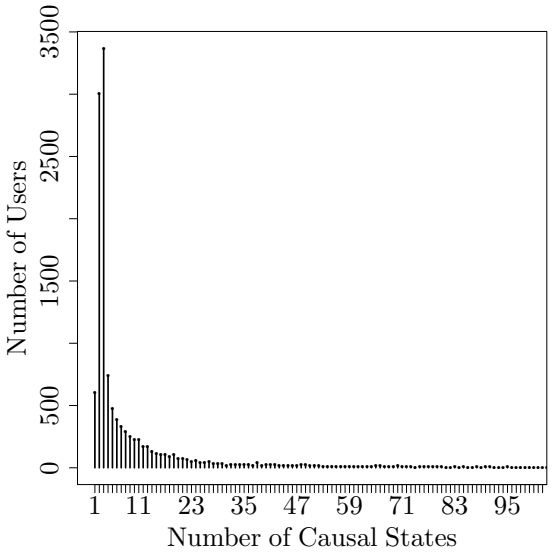
5.3.3 ϵ-transducer Causal Architectures

Thus far, we have considered the models associated with user behavior when we ignore their inputs. Next we turn to the models that incorporate those inputs, namely the ϵ-transducer. Recall that the input $\{Y_t\}$ we consider is whether a user was mentioned during time interval t. We begin by considering the distribution of the number of states across the ϵ-transducer models. Figure 5.10 shows these distributions for the self-memoryless (top) and self-memoryful (bottom) cases. Again, as with the ϵ-machines, we see that most users are well-described by ϵ-transducers with a small number of states, with 95% having 6 states or fewer in the self-memoryless case and 90% having 25 or fewer states in the self-memoryful case. For the self-memoryless case, 12059 of the 12641 mentioned users (95%) have an ϵ-transducer with the architecture given in Figure 5.11. That is, the user has a 'just-mentioned' state (labelled 0), and subsequent periods without the user receiving a mention lead to transitions away from this state, until a terminal state $\tilde{n}$ is reached. This architecture is analogous to the ϵ-machine architecture of a renewal process shown in Figure 5.7. In particular, the causal states map to the time since the user was mentioned, with all times of length $\tilde{n}$ or longer mapped to the same state. Thus, when viewed as purely socially-driven, the relevant quantity to track for almost all of the users is the time since they were last mentioned.

A similar overarching 'counting' model architecture is also present amongst the memoryful ϵ-transducers. Recalling that another way to view the states of a mixed renewal process is as counting the length of runs of x since the last $x' \neq x$,



(a) Self-memoryless.



(b) Self-memoryful.

Figure 5.10: The distribution of channel causal states in the self-memoryless (top) and self-memoryful (bottom) cases. We exclude 1885 of the 14342 active users who did not receive any mentions.

we can generalize this to the ϵ-transducer by considering states that count the lengths of runs of input-output symbols (y, x) since the last input-output symbol $(y', x') \neq (y, x)$. As in the memoryless ϵ-transducer case, we call this a mixed renewal-like process, since the causal states act in a similar fashion to those for

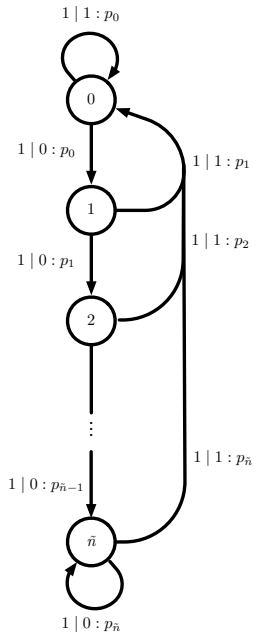


Figure 5.11: The self-memoryless ϵ-transducer architecture associated with 12059 of the 12641 mentioned users (95%). Note that we suppress the transitions $0 \mid y : 1-p_n$, since the transitions do not depend on x in the self-memoryless case.

a mixed renewal process. We present a schematic representation of the partitioning of the channel causal state space in Figure 5.12. For these ϵ-transducers, the causal states can be partitioned based on the runs of (y, x) they count since the last $(y', x') \neq (y, x)$. Thus, we begin by dividing the set of causal states into four quadrants, based on the runs of (y, x) which they count. All states in a quadrant labeled by (y, x) are transitioned to on (y, x). Then, the causal states within a quadrant are further partitioned into thirds, where each third corresponds to the symbol (y', x') seen before the current run of (y, x). Thus, each third has a unique start state that is transitioned to on a (y, x) from a state transitioned to on a (y', x'). 10216 of 12641 (81%) of the mentioned users have an ϵ-transducer in this mixed renewal-like class. Note that the partitioning given in Figure 5.12 is the most general possible for this type of ϵ-transducer. The quadrants / thirds within a quadrant may further

collapse, as dictated by the structure of the ϵ-transducer. For example, Figure 5.13 is a mixed renewal-like transducer inferred for 27% of the mentioned users. This ϵ-transducer has three states, which correspond to runs of $(0, 0)$, $(0, 1)$, and $(1, *)$ and are labeled as such. In this case, the quadrants corresponding to $(1, 0)$ and $(1, 1)$ collapse, since the corresponding state counts runs of $y = 1$ regardless of the user behavior x. Moreover, all thirds within a given quadrant also collapse, since the states treat runs of (y, x) as the same from any $(y', x') \neq (y, x)$. In terms of the actual behavior of the user, we see that the state labeled $(0, 0)$ corresponds to when the user has been both quiescent and unmentioned in the recent past. In this case, the user has probability β of being active given this state. The state labeled $(0, 1)$ corresponds to when the user has been active, but not mentioned, in the recent past. In this case, the user has probability $\alpha > \beta$ of being active given this state. Finally, the state labeled $(1, *)$ corresponds to the case where the user has been mentioned in the recent past, regardless of whether or not the user has been active. The user has probability $\gamma > \beta$ of being active given this state. Thus, for over a quarter of the users, we see that knowledge of the recent past of both their own and their inputs behaviors provides sufficient information for predicting their future behavior. In particular, each of the quadrants requires only a single state, whereas in the most general model of this type with $L_{\max} = 6$ allows for $6 \times 3 = 18$ states per quadrant.

As with the renewal, reverse renewal, and mixed renewal processes inferred from the ϵ-machines, this mixed renewal-like ϵ-transducer motivates a particular parametric model, albeit a much more complicated one. In this case, we need to specify the chain lengths $\tilde{n}_{(y',x'),(y,x)}$ within each third (y', x') of a quadrant (y, x).

However, this results in at most $L \cdot |\mathcal{X}| \cdot |\mathcal{Y}| \cdot (|\mathcal{X}| \cdot |\mathcal{Y}| - 1)$ states overall, compared to $(|\mathcal{X}| \cdot |\mathcal{Y}|)^L$ states in the most general model, and therefore a linear growth in model complexity as a function of history length L compared to a geometric growth.

Again, as with the mixed renewal process, the mixed renewal-like ϵ-transducer generalizes to higher orders by considering the input-output behavior immediately prior to a switch from (y', x') to $(y, x) \neq (y', x')$. For example, a second order mixed renewal-like ϵ-transducer would distinguish between a user becoming quiescent and unmentioned after being mentioned twice in the past compared to going unmentioned before the previous mention. Many of the users exhibit ϵ-transducers of higher order as shown in Table 5.2. Of the 12641 mentioned users, 78% are mixed renewal-like of order 6 or smaller.

5.3.4 Case Studies

Thus far, we have considered the model architectures and performances in the aggregate, providing an aerial view of the computational landscape of the users. Next we turn to two case studies that demonstrate how the different models capture different views of a user's behavior. Figure 5.14 highlights two users in the score-score plane defined by the ϵ-machine score and the self-memoryless ϵ-transducer score. We choose this pair of models because, unlike with self-memoryful ϵ-transducer, they are not sub/super-models of each other.

For each of the users highlighted in Figure 5.14, we present various observed and inferred properties of their behavior in Figures 5.15 and 5.16. The first panel

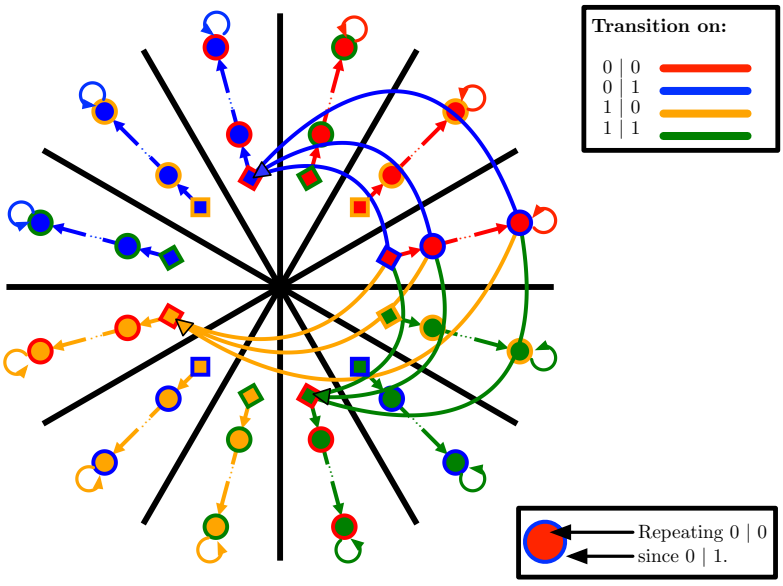


Figure 5.12: A schematic demonstrating the partitioning of the transducer state space associated with a renewal-like ϵ-transducer. Each quadrant is determined by the input-output symbol pair being 'counted,' and each third within a quadrant is determined by the input-output pair the count begins from. We only show outgoing transitions for the first third of the first quadrant, which correspond to transitions of $0 \mid 1$, $1 \mid 0$ or $1 \mid 1$ after observing $0 \mid 0$.

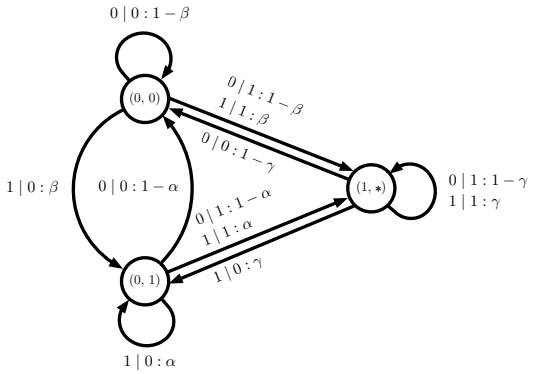


Figure 5.13: The most common self-memoryful ϵ-transducer architecture associated with 3376 of the 12641 mentioned users (27%).

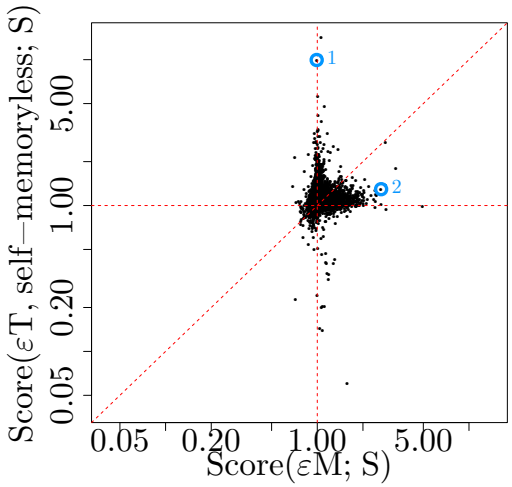


Figure 5.14: The position of the two users taken as case studies in the score-score plane defined by the ϵ-machine and self-memoryless ϵ-transducer scores.

(a) corresponds to the mentions the user receives, represented as a rastergram. The second panel (b) corresponds to their activity. The third panel (c) shows their inferred seasonality $p(t)$. The fourth (d), fifth (e), and sixth (f) panels show the inferred ϵ-machine, self-memoryless ϵ-transducer, and self-memoryful ϵ-transducer, respectively, for the user.

The first user, depicted in Figure 5.15, corresponds to an individual strongly

driven by their social inputs. We can see directly from inspection of the rastergrams (a) and (b) that the user is typically active immediately following a mention, but not always. In fact, the user appears to have a strong seasonality to their behavior, as shown in (c), but this seasonality mostly reflects the seasonality of their social influence. Turning to the ϵ-machine for their observed behavior, we see that it corresponds to that of a renewal process with $\tilde{n} = 3$, thus indicating a short memory for the transition from activity to quiescence. However, this again largely reflects the dynamics of the mention time series, which we can see by turning to the self-memoryless ϵ-transducer (e). The self-memoryless ϵ-transducer is renewal-like with $\tilde{n} = 1$: when mentioned, the user almost always becomes active, and when unmentioned, the user almost always stays quiescent. By incorporating the user's own behavior with the self-memoryful ϵ-transducer, we see that the user exhibits both self and social memory in the sense captured by the mixed renewal-like model. The user's self-memoryful ϵ-transducer is identical to Figure 5.13. Thus, to predict this user, it is sufficient to track whether they have just been active or whether they have just been mentioned.

The second user corresponds to a case where the ϵ-machine outperforms the self-memoryless ϵ-transducer. Unlike in the first case study, it is unclear from direct inspection of the rastergrams (a) and (b) how well we can attribute the users behavior to their mentions. While it appears that periods of activity may be initiated by a mention, use of the rastergrams alone is inconclusive. The ϵ-machine (d) for the user's behavior, a mixed renewal type, indicates that they possess memory for both their quiescence and activity: as they switch from quiescence to activity, they

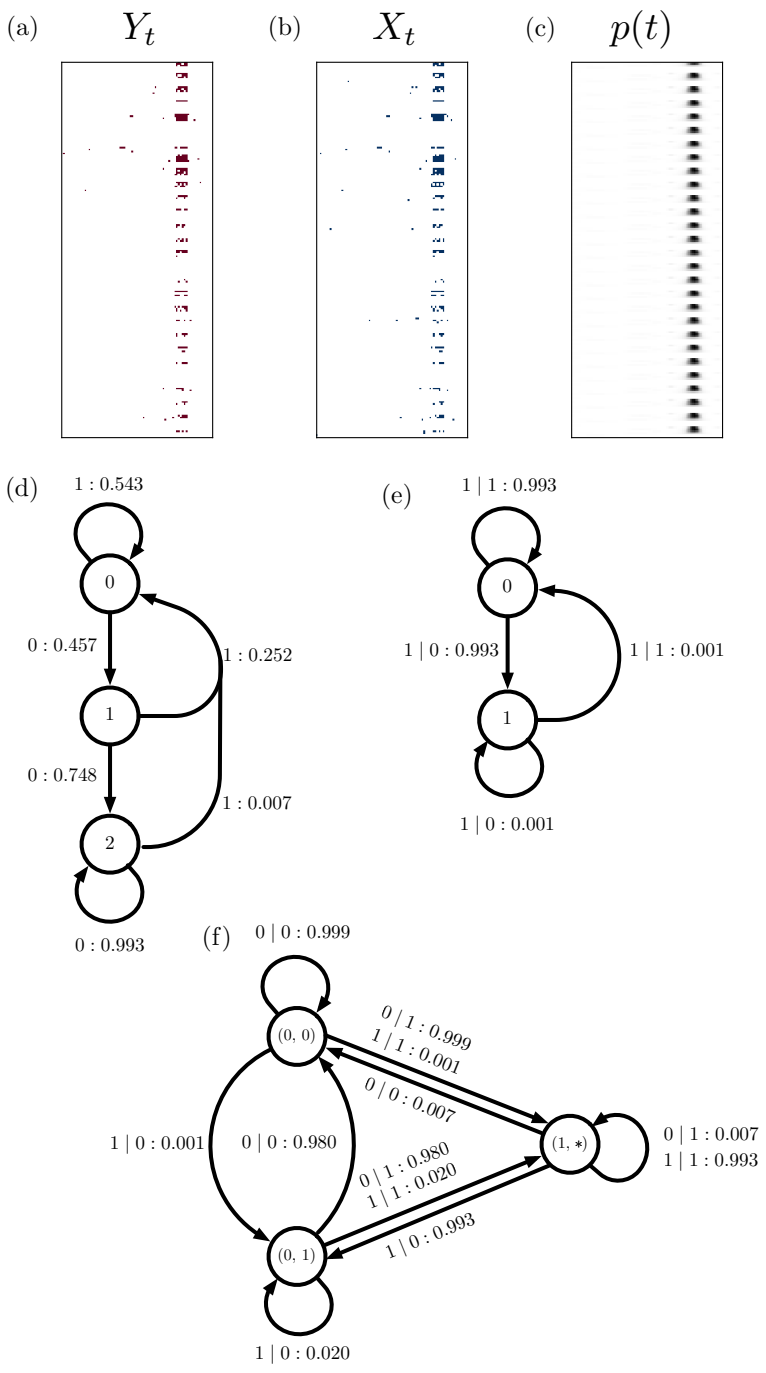


Figure 5.15: Case study for user marked 1 in Figure 5.14. (a) The mention input Y_t for the user. (b) The activity X_t of the user. (c) The estimated seasonality $p(t)$ for the user. (d) The ϵ-machine for the user's activity. (e) The self-memoryless ϵ-transducer for the user's activity. (e) The self-memoryful ϵ-transducer for the user's activity.

are more likely to remain active, and *vice versa*. We see from the self-memoryless ϵ-transducer (e) that the user does exhibit memory of their mentions, with the chance of them becoming active decaying with the time since the last mention in a renewal-like manner. Finally, turning to the self-memoryful ϵ-transducer, we see that it is of the mixed renewal-like type. However, unlike the first case study, this user distinguishes between all four of the {quiescent, active} $\times$ {unmentioned, mentioned} conditions. Moreover, when the user switches to the unmentioned and quiescent condition (top right quadrant), they maintain a memory as to whether they had previously been active (bottom left / right quadrants) or not (top left quadrant).

5.4 Conclusions

In this chapter, we have developed and applied a modeling framework for human behavior in digital environments. The approach begins by viewing a user's behavior as a discrete-time point process at a prespecified temporal resolution, and then considers four possible stochastic models that might give rise to the user's behavior, namely the seasonal, self-driven, socially-driven, and self- and socially-driven processes approximated by an inhomogeneous Bernoulli process, an ϵ-machine, and self-memoryless/memoryful ϵ-transducers.

We have found that simple computational architectures, as specified by their ϵ-machines and ϵ-transducers, describe much of the observed behavior of the users in our data set. A renewal process model, or its generalizations to reverse renewal and mixed renewal processes, was found to be appropriate for approximately 80%

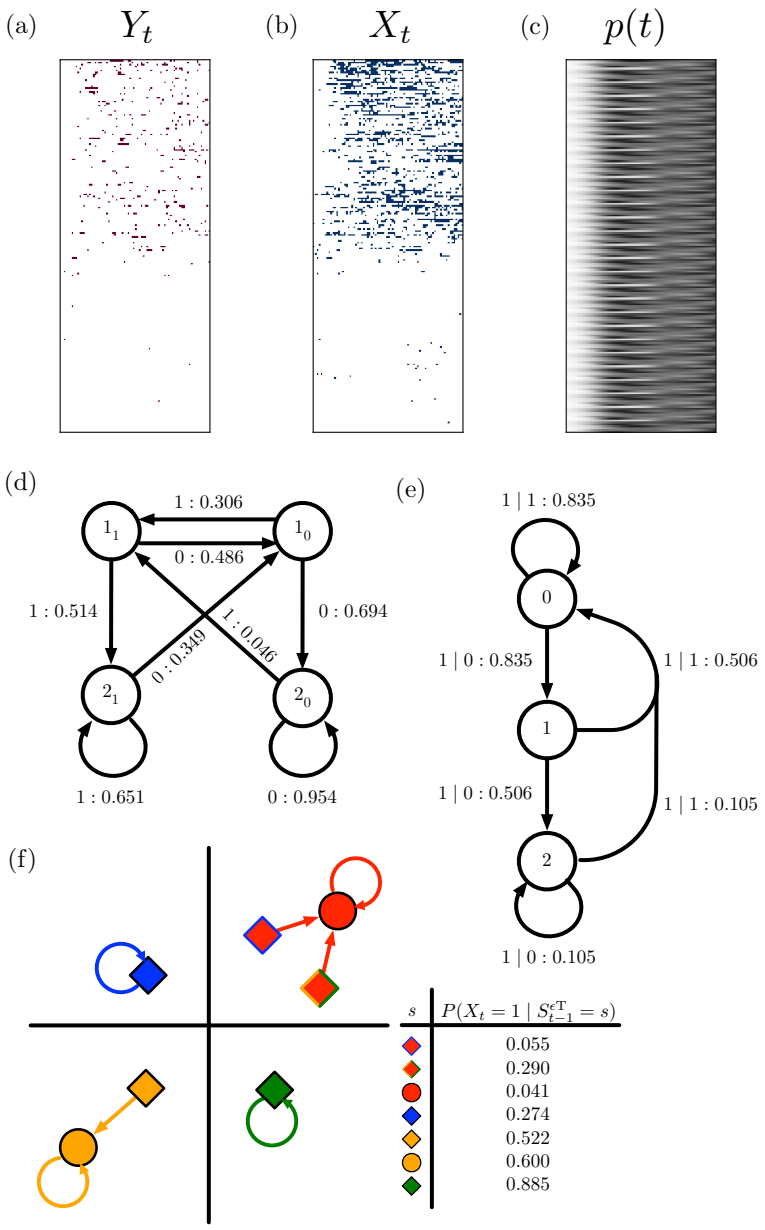


Figure 5.16: Case study for user marked 2 in Figure 5.14. (a) The mention input Y_t for the user. (b) The activity X_t of the user. (c) The estimated seasonality $p(t)$ for the user. (d) The ϵ-machine for the user's activity. (e) The self-memoryless ϵ-transducer for the user's activity. (e) The self-memoryful ϵ-transducer for the user's activity. We suppress between-quadrant transitions since they are implied by the color scheme given in Figure 5.12 and provide the associated probabilities in the table.

of the users in our study. This is in agreement with much of the literature on human communication patterns. However, we emphasize that we did not *assume*

such models *a priori*, but rather *discovered* their prevalence by using non-parametric modeling in an exploratory fashion. In fact, the appearance of reverse renewal and mixed renewal processes demonstrate that renewal process models alone are not sufficient to describe, for example, the burstiness observed in human communication patterns. Moreover, we discovered a new class of renewal-like models that generalize renewal processes to input-output systems. We found that this class of models describes over 70% of the users in terms of the interaction between their activity and their social inputs. The prevalence of these stereotyped ϵ-machines/transducers motivates the use of either frequentist (such as the cross-validation approach used in this paper) or Bayesian (as recently developed in [132]) approaches that take advantage of these structures *a priori* during the estimation process. In addition to the renewal-like models, more general models were necessary for over 20% of the users in the self-driven case and nearly 30% of the users in the self- and socially-driven case.

The apparent complexity of user behavior seems to arise from a simple computational landscape. Our present work lays out an initial sketch of this landscape's features. We hope this work motivates further exploration of this landscape and refinement of its map.

Appendix A: `transCSSR` for ϵ-Transducer Reconstruction

There are many algorithms for inferring ϵ-machines from data, from the topological methods first presented in [5] to more recent methods based on Bayesian methods [132]. Additional algorithms have been developed based on spectral methods [133] and integer programming [134]. However, the most popular method is the Causal State Splitting Reconstruction (CSSR) algorithm [46]. It is also the only current ϵ-machine reconstruction algorithm that provides a provably consistent (in the statistical sense) estimator for a stochastic process's ϵ-machine under mild conditions on the stochastic process.

The theory for ϵ-transducers has only recently been developed, and therefore there are currently very few algorithms for ϵ-transducer reconstruction from finite data. Sketches of CSSR-like algorithms for ϵ-transducer reconstruction are provided in [4, 12], however they are not developed beyond suggestions for the reader. In this appendix, we develop the ideas originally suggested in these prior works, and present a generalization of CSSR for ϵ-transducer reconstruction from data resulting from input-output systems. In homage to CSSR, we call our algorithm `transCSSR`, a portmanteau of transducer and CSSR. The `transCSSR` algorithm has been implemented in Python, and is maintained at `http://github.com/ddarmon/`

`transCSSR-master/`.

As stated in Chapter 5, there are at present two formulations of input/output computational mechanics that differ only in how they condition on the future input to the system in forming the predictive distribution for the future output. The original formulation [6, 8] does not condition on the future input, while the more recent formulation [9] does. However, the original formulation is a special case of the recent one, and thus we focus on this more general version in our explication of the `transCSSR` algorithm.

A.1 An Outline of the Algorithm

Consider an input/output alphabet pair $(\mathcal{Y}, \mathcal{X})$, a joint realization $(\bar{y}, \bar{x}) = (y_1^N, x_1^N)$ of length N, a maximum lookup length $L_{\max}$, and a significance level α. Our goal is to estimate the set of causal states $\mathcal{S}$. The `transCSSR` algorithm goes about this estimation process by taking advantage of a key consequence of the definition of the causal states in terms of equivalence relation. Namely, as shown in [9], the equivalence relation

$$\begin{aligned} &(r_{-\infty}^{t-1}, u_{-\infty}^{t-1}) \sim (s_{-\infty}^{t-1}, v_{-\infty}^{t-1}) \implies \\ &P(X_t^{\infty} \mid Y_t^{\infty}, X_{-\infty}^{t-1} = u_{-\infty}^{t-1}, Y_{-\infty}^{t-1} = r_{-\infty}^{t-1}) \\ &= P(X_t^{\infty} \mid Y_t^{\infty}, X_{-\infty}^{t-1} = v_{-\infty}^{t-1}, Y_{-\infty}^{t-1} = s_{-\infty}^{t-1}). \end{aligned} \tag{A.1}$$

in terms of the input-conditioned predictive distribution over semi-infinite futures X_t^∞ induces the same partition of the joint pasts as the equivalence relation

$$
\begin{aligned}
&(r_{-\infty}^{t-1}, u_{-\infty}^{t-1}) \sim (s_{-\infty}^{t-1}, v_{-\infty}^{t-1}) \implies \\
&(i)\ P(X_t \mid Y_t, X_{-\infty}^{t-1} = u_{-\infty}^{t-1}, Y_{-\infty}^{t-1} = r_{-\infty}^{t-1}) \\
&= P(X_t \mid Y_t, X_{-\infty}^{t-1} = v_{-\infty}^{t-1}, Y_{-\infty}^{t-1} = s_{-\infty}^{t-1}) \\
&(ii)\ P(X_{t+1} \mid Y_{t+1}, X_{-\infty}^{t} = u_{-\infty}^{t-1}b, Y_{-\infty}^{t} = r_{-\infty}^{t-1}a) \\
&= P(X_{t+1} \mid Y_{t+1}, X_{-\infty}^{t} = v_{-\infty}^{t-1}b, Y_{-\infty}^{t} = s_{-\infty}^{t-1}a)
\end{aligned}
\tag{A.2}
$$

in terms of the input-conditioned predictive distribution over *one-step ahead* futures X_t (condition (i)) when we enforce unifilarity (condition (ii)). Thus, if we determine a minimal partition of histories such that it (i) induces the same *one-step ahead* predictive distribution for all histories in a state and is (ii) unifilar, we have determined candidate causal states for the input-output process, and thus an estimate for the input-output process's ϵ-transducer.

`transCSSR` does so in three stages:

1. Initialization
2. Homogenization
3. Determinization.

A.1.1 Initialization

The initialization step begins by setting $L = 0$ and assuming that all joint histories result in the same predictive distribution. That is, the set of candidate causal states is taken to be $\hat{\mathcal{S}} = \{s_0\}$ with $s_0 = \{(*, *)\}$, where $(*, *)$ is the null joint history. The one-step-ahead predictive distribution is then taken to be

$$P(X_0 \mid Y_0, S = s_0) = P(X_0 \mid Y_0). \tag{A.3}$$

That is, we begin by assuming that the transducer is completely memoryless of both the input and output pasts.

A.1.2 Homogenization

In the homogenization procedure, the joint histories in each candidate causal state are grown by one input/output symbol, and a statistical test is used to determine whether the resulting one-step-ahead predictive distribution is identical to the one-step-ahead predictive distribution of the parent causal state. This is equivalent to testing that the next output symbol is independent of the past, given the current causal state, the shielding property of causal states [9].

In practice, this is done as follows:

1. For each $s \in \hat{\mathcal{S}}$, estimate the one-step-ahead predictive distribution associated with that causal state.

   (a) When $L = 0$, use (A.3).

(b) When $L > 0$, for each joint input / output history of length L, $(y_{-L}^{-1}, x_{-L}^{-1})$, estimate the one-step-ahead predictive distribution $P\left(X_0 \mid Y_0, (Y_{-L}^{-1}, X_{-L}^{-1}) = (y_{-L}^{-1}, x_{-L}^{-1})\right)$ as

$$\begin{aligned} &\hat{P}\left(X_0 = x \mid Y_0 = y, (Y_{-L}^{-1}, X_{-L}^{-1}) = (y_{-L}^{-1}, x_{-L}^{-1})\right) \\ &= \frac{\nu\left(X_0 = x, Y_0 = y, (Y_{-L}^{-1}, X_{-L}^{-1}) = (y_{-L}^{-1}, x_{-L}^{-1})\right)}{\nu\left(Y_0 = y, (Y_{-L}^{-1}, X_{-L}^{-1}) = (y_{-L}^{-1}, x_{-L}^{-1})\right)} \end{aligned} \tag{A.4}$$

where $\nu(\cdot)$ gives the counts of the occurrences of instances of $(\cdot)$ in the data stream (y_1^N, x_1^N). This is the maximum likelihood estimate for the one-step-ahead predictive distribution in the non-parametric case.

(c) The one-step-ahead predictive distribution for s is taken to be the weighted average of the one-step-ahead predictive distributions for each history in s,

$$\begin{aligned} &\hat{P}\left(X_0 = x \mid Y_0 = y, S = s\right) \\ &= \frac{1}{Z} \sum_{(y_{-L}^{-1}, x_{-L}^{-1}) \in s} \nu\left((Y_{-L}^{-1}, X_{-L}^{-1}) = (y_{-L}^{-1}, x_{-L}^{-1})\right) \times \\ &\hat{P}\left(X_0 = x \mid Y_0 = y, (Y_{-L}^{-1}, X_{-L}^{-1}) = (y_{-L}^{-1}, x_{-L}^{-1})\right) \end{aligned} \tag{A.5}$$

where $Z = \sum_{(y_{-L}^{-1}, x_{-L}^{-1}) \in s} \nu\left((Y_{-L}^{-1}, X_{-L}^{-1}) = (y_{-L}^{-1}, x_{-L}^{-1})\right)$.

2. For each $s \in \hat{\mathcal{S}}$, test that s is Markovian.

(a) For each $(y_{-L}^{-1}, x_{-L}^{-1}) \in s$ and for each $(a, b) \in \mathcal{Y} \times \mathcal{X}$, grow the history $(y_{-L}^{-1}, x_{-L}^{-1})$ into the past by (a, b). Estimate the probability $P\left(X_0 \mid Y_0, (Y_{-(L+1)}^{-1}, X_{-(L+1)}^{-1}) = (a y_{-L}^{-1}, b x_{-L}^{-1})\right)$ using (A.4).

(b) Test that the new symbol $(ay_{-L}^{-1}, bx_{-L}^{-1})$ has the same one-step-ahead predictive distribution as its parent using the hypothesis test with the null hypothesis

$$P\left(X_0 \mid Y_0, (Y_{-(L+1)}^{-1}, X_{-(L+1)}^{-1}) = (ay_{-L}^{-1}, bx_{-L}^{-1})\right) = P\left(X_0 \mid Y_0, \hat{S} = s\right) \tag{A.6}$$

for all $(a, b) \in \mathcal{Y} \times \mathcal{X}$. If we do not reject the null, add $(ay_{-L}^{-1}, bx_{-L}^{-1}))$ to s.

(c) If we do reject the null, test whether $(ay_{-L}^{-1}, bx_{-L}^{-1})$ belongs to one of the other causal states $s^* \neq s$ using the restricted alternative hypothesis,

$$P\left(X_0 \mid Y_0, (Y_{-(L+1)}^{-1}, X_{-(L+1)}^{-1}) = (ay_{-L}^{-1}, bx_{-L}^{-1})\right) = P\left(X_0 \mid Y_0, \hat{S} = s^*\right). \tag{A.7}$$

If we do not reject the null for more than one state s^*, we add $(ay_{-L}^{-1}, bx_{-L}^{-1})$ to the state with the smallest test statistic.

(d) If we reject the restricted alternative hypothesis for all $s^* \neq s$, we create a new state and add $(ay_{-L}^{-1}, bx_{-L}^{-1})$ to that state.

3. Increment L by 1.

4. Repeat steps 1–3 until we reach the maximum history length $L_{\max}$.

Any hypothesis test for comparing two distributions may be used in Steps 2b and 2c. In our implementation, we use the G-statistic [14]. Consider the case

of two distributions with common support $\mathcal{A}$. Denote the distributions over $\mathcal{A}$ by $\mathbf{p}_1 = (p_{11}, \ldots, p_{1|\mathcal{A}|})$ and $\mathbf{p}_2 = (p_{21}, \ldots, p_{2|\mathcal{A}|})$. Consider a sample $\mathbf{X}_1 = (X_{1,1}, \ldots, X_{1,n_1})$ of length n_1 from $\mathbf{p}_1$ and $\mathbf{X}_2 = (X_{2,1}, \ldots, X_{2,n_2})$ of length n_2 from $\mathbf{p}_2$. We wish to use the samples $\mathbf{X}_1$ and $\mathbf{X}_1$ to perform the hypothesis test

$$\begin{aligned} H_0 &: \mathbf{p}_1 = \mathbf{p}_2 \\ H_1 &: \mathbf{p}_1 \neq \mathbf{p}_2 \end{aligned} \tag{A.8}$$

The G-statistic associated with these samples is

$$G = 2 \sum_{r=1}^{2} \sum_{c=1}^{|\mathcal{A}|} n_r \hat{p}_{rc} \log \frac{\hat{p}_{rc}}{\bar{p}_c}, \tag{A.9}$$

where $\hat{p}_{rc} = \frac{\nu(X_r,\cdot=c)}{n_r}$ are the maximum likelihood estimates for p_{rc} under the alternative hypothesis and $\bar{p}_{rc} = \frac{\nu(X_1,\cdot=c)+\nu(X_2,\cdot=c)}{n_1+n_2}$ are the maximum likelihood estimates for p_{rc} under the null. The G-statistic is asymptotically $\chi^2(|\mathcal{A}| - 1)$.

A.1.3 Determinization

At the end of the homogenization stage of `transCSSR`, each state corresponds to a collection of histories that are statistically equivalent in terms of one-step-ahead prediction, and are distinct from each other state in terms of their own one-step-ahead predictive distribution. The causal states have this property, in addition to being unifilar. That is, for each possible joint input/output symbol, transitions between states should be deterministic. The determinization step of `transCSSR` results in such a collection of states by splitting histories from a state when they

transition to different states on the same input/output pair (a, b).

In practice, this is done as follows:

1. Determine transient states using the state transition structure, and remove them, thus leaving only recurrent states.

2. For each state $s \in \hat{\mathcal{S}}$:

   (a) For each input/output pair $(a, b) \in \mathcal{Y} \times \mathcal{X}$:

      i. For all joint histories $(y_{-(L_{\max}-1)}^{-1}, x_{-(L_{\max}-1)}^{-1}) \in s$, determine the state they transition to on accepting (a, b), $\epsilon((y_{-(L_{\max}-1)}^{-1}a, x_{-(L_{\max}-1)}^{-1}b))$. Call these the successor states for the histories $(y_{-(L_{\max}-1)}^{-1}, x_{-(L_{\max}-1)}^{-1})$ on (a, b).

      ii. If there are n_s successor states with $n_s > 1$ on accepting (a, b), then the transition for state s is not deterministic on (a, b). To ensure determinism, create $n_s - 1$ new states, and apportion the histories in s such that each new state contains histories that have the same transition on accepting (a, b). Go to i.

   (b) Repeat (a) until every $(y_{-(L_{\max}-1)}^{-1}, x_{-(L_{\max}-1)}^{-1}) \in s$ has the same successor state on each (a, b).

3. Repeat steps 1 and 2 until no more splitting occurs.

This process can only generate states containing fewer histories, and therefore always terminates. In the extreme case, it terminates with each history assigned to its own state. Since the new states are always split from a state where all histories

give statistically equivalent predictions over one-step-ahead futures, the new states also maintain this property. Therefore, the final collection of states is both weakly prescient and deterministic, and therefore causal. Pseudo-code for the steps of `transCSSR` is given in Algorithm A.1.3.

We next turn to demonstrating `transCSSR` on data generated from an example input/output channel: the odd random channel.

Algorithm 1 Pseudo-code for the algorithm `transCSSR`. Arguments: $\mathcal{Y}, \mathcal{X}$: the discrete alphabets for the input and output processes; (y_1^N, x_1^N) the joint input/output sequence of length N with each joint symbol (y, x) drawn from $\mathcal{Y} \times \mathcal{X}$; $L_{\max}$, the maximum history length used when estimating candidate causal states; α, the probability of falsely rejecting the null hypothesis (A.6) or the restricted alternative hypothesis (A.7).

I. Initialization: $L \leftarrow 0$, $\Sigma \leftarrow \{\{\emptyset\}\}$
II. Homogenization:
while $L < L_{\max}$ **do**
for each $s \in \Sigma$ **do**
estimate $\hat{P}(X_t|\hat{S} = s)$
for each $(y_{t-(L-1)}^{t-1}, x_{t-(L-1)}^{t-1}) \in \mathcal{Y}_{t-(L-1)}^{t-1} \times \mathcal{X}_{t-(L-1)}^{t-1}$ **do**
for each $(a, b) \in \mathcal{Y} \times \mathcal{X}$ **do**
estimate
$p \leftarrow \hat{P}(X_t|(Y_{t-L}^{t-1}, X_{t-L}^{t-1}) = (ay_{t-(L-1)}^{t-1}, bx_{t-(L-1)}^{t-1}))$
$\textsc{Test}(\Sigma, p, (ay_{t-(L-1)}^{t-1}, bx_{t-(L-1)}^{t-1}), s, \alpha)$
end for
end for
end for
$L \leftarrow L + 1$
end while
III. Determinization:
Remove transient states from Σ
`recursive` $\leftarrow$ False
while Not `recursive` **do**
`recursive` $\leftarrow$ True
for each $s \in \Sigma$ **do**
for each $(a, b) \in \mathcal{Y} \times \mathcal{X}$ **do**
$(\mathbf{y}_0, \mathbf{x}_0) \leftarrow$ first $(y_{t-(L_{\max}-1)}^{t-1}, x_{t-(L_{\max}-1)}^{t-1}) \in s$
$T(s, (a, b)) \leftarrow \hat{\epsilon}((\mathbf{y}_0 a, \mathbf{x}_0 b))$
for each $(\mathbf{y}, \mathbf{x}) \in s, (\mathbf{y}, \mathbf{x}) \neq (\mathbf{y}_0, \mathbf{x}_0)$ **do**
if $\hat{\epsilon}((\mathbf{y}a, \mathbf{x}b)) \neq T(s, (a, b))$ **then**
create new state $s' \in \Sigma$
$T(s', (a, b)) \leftarrow \hat{\epsilon}((\mathbf{y}a, \mathbf{x}b))$
for each $(\mathbf{y}', \mathbf{x}') \in s$ such that
$\hat{\epsilon}((\mathbf{y}'a, \mathbf{x}'b)) = \hat{\epsilon}((\mathbf{y}a, \mathbf{x}b))$ **do**
$\textsc{Move}((\mathbf{y}', \mathbf{x}'), s, s')$
end for
`recursive` $\leftarrow$ False
end if
end for
end for
end for
end while

TEST$(\Sigma, p, (a\mathbf{y}, b\mathbf{x}), s, \alpha)$
  if null hypothesis (A.6) passes a test of size α **then**
    $s \leftarrow \{(a\mathbf{y}, b\mathbf{x})\} \cup s$
  else if restricted alternative hypothesis (A.7) passes a test of size α for $s^* \in \Sigma, s^* \neq s$ **then**
    MOVE$((a\mathbf{y}, b\mathbf{x}), s, s^*)$
  else
    create new state $s' \in \Sigma$
    MOVE$((a\mathbf{y}, b\mathbf{x}), s, s')$
  end if

MOVE$((\mathbf{y}, \mathbf{x}), s_1, s_2)$
  $s_1 \leftarrow s_1 \setminus \{(\mathbf{y}, \mathbf{x})\}$
  re-estimate $\hat{P}(X_t|\hat{S} = s_1)$
  $s_2 \leftarrow s_2 \cup \{(\mathbf{y}, \mathbf{x})\}$
  re-estimate $\hat{P}(X_t|\hat{S} = s_2)$

A.2 A Worked Example – The Odd Random Channel

We now demonstrate `transCSSR` using one of the channels presented in [9]. Consider the odd random channel, whose ϵ-transducer is given in Figure A.1. The generative story for the odd random channel is as follows. The channel stores the parity of the input sequence, whether an even or odd number of 1s have occurred since the last 0. If the parity of the input sequence is even, that is, 0, 2, 4, etc., 1s have occurred since the last 0, then the channel acts as the identity: the input symbol is outputted. If the parity of the input sequence is odd, then the output symbol is random, taking the values 0 and 1 with equal probability. Thus, as demonstrated in Figure A.1, the ϵ-transducer has two states corresponding to the parity of the input sequence: A, when the parity of the input stream is even, and B, when the parity of the input sequence is odd. Alternatively, the Odd Random Channel may be defined in terms of its transition matrices $\mathbf{T}^{(x|y)}$

$$T_{ij}^{(x|y)} = P(X_t = x, S_t = s_j \mid Y_t = y, S_{t-1} = s_i), \tag{A.10}$$

which are given by

$$\mathbf{T}^{(0|0)} = \begin{pmatrix} 1 & 0 \\ 1/2 & 0 \end{pmatrix}, \mathbf{T}^{(0|1)} = \begin{pmatrix} 0 & 0 \\ 1/2 & 0 \end{pmatrix}, \mathbf{T}^{(1|0)} = \begin{pmatrix} 0 & 0 \\ 1/2 & 0 \end{pmatrix}, \mathbf{T}^{(1|1)} = \begin{pmatrix} 0 & 1 \\ 1/2 & 0 \end{pmatrix}. \tag{A.11}$$

Despite the simple two state ϵ-transducer representation of the Odd Random

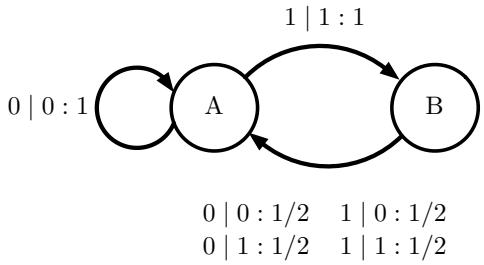


Figure A.1: The ϵ-transducer representation of the Odd Random Channel. The Odd Random Channel has two states, determined by the parity of the input sequence. When the parity is even (state A), the channel acts as the identity, taking the current input as the current output. When the parity is odd (state B), the output is chosen uniformly from $\{0, 1\}$.

Channel, it has no finite state presentation purely in terms of the joint histories. This is analogous to the sophic processes [135] of discrete-valued, discrete-time stochastic processes which can not be represented as a Markov chain of any finite order. A heuristic argument for why the Odd Random Channel has no finite state presentation goes as follows. The behavior of the Odd Random Channel is completely determined by the current parity of the input. However, the current parity of the input is only known once the most recent 0 is encountered, and one may have to look infinitely far into the past to encounter such a 0, if one even exists.

In the following sections, we demonstrate the steps of `transCSSR` with a data generated by taking a fair Bernoulli process as the input to the Odd Random Channel. All probabilities are estimated using a joint data stream of length $N = 100000$.

A.2.1 Sufficiency for the Odd Random Channel

We begin by estimating $P(X_0 \mid Y_0)$ from the data stream, which gives us the stochastic matrix

y \ x	0	1
0	0.832	0.168
1	0.167	0.833

where each entry is $\hat{P}(X_0 = x \mid Y_0 = y)$. These predictive probabilities correspond to the state of complete ignorance about whether the transducer is in the even or odd state. They correspond to a weighted average of the the channel probabilities, with weights given by the probability of the parity being even / odd. Note that when the input is a fair Bernoulli process, the probability of being in the even state is 2/3 and the probability of being in the odd state is 1/3. Thus, we begin with a single causal state $s_0 = \{(*, *)\}$, which has this associated predictive distribution.

Next, we take $L = 1$. There is only one causal state with a single joint history, so we grow its joint history $(*, *)$ into the past by each $(a, b) \in \mathcal{Y} \times \mathcal{X}$, and estimate the stochastic matrix for each new history according to (A.4). The associated stochastic matrices are given in Figure A.2. We see that each new history has a predictive distribution that is clearly different from that of the parent history $(*, *)$. Moreover, the histories $(*0, *0), (*0, *1)$, and $(*1, *0)$ all have equivalent predictive distributions, so they are all assigned to the same causal state. In fact, with these length 1 joint histories all belong in the even state, since either we have just observed a zero as with $(*0, *0)$ and $(*0, *1)$, or we have observed a history

that must have occurred in the odd state and leads to a transition to the even state as with $(*1, *0)$. However, the history $(*1, *1)$ does not resolve into any of the recurrent causal states: the 1 observed from the output could have resulted from the input having either even or odd parity. Again, the predictive distribution corresponds to a weighted sum of their predictive distributions. Thus, for $L = 1$, we have three candidate causal states, $s_0 = \{(*, *)\}, s_1 = \{(*0, *0), (*0, *1), (*1, *0)\}$, and $s_2 = \{(*1, *1)\}$. The candidate causal states and their predictive distributions are given in Figure A.3.

Next, we take $L = 2$. We have already accounted for the children histories for causal state s_0. The stochastic matrices for each of the histories in s_1 and s_2 are given in Figure A.4. We see that all of the children histories generated from the histories in s_1 have the same predictive distribution as their parents. This occurs because all of the parent histories in s_1 already resolve to the even state, and therefore their backward time children must also belong to this state. Also note that for the first time we have encountered input-output pairs that are not allowed by this transducer. For example, the child $(*00, *01)$ of $(*0, *1)$ cannot occur: the Odd Random Channel cannot emit a 1 on receiving a 0 when it is in the even state. We denote these non-admissible input-output pairs by a dash $(-)$ in the stochastic matrix for these entries. The children of $(*1, *1)$ resolve into two causal states: the histories $(*01, *01), (*01, *11)$, and $(*11, *01)$ are moved to a new causal state s_3, and the history $(*11, *11)$ is moved to the existing causal state s_0. Thus, for $L = 2$, we have four candidate causal states given in Figure A.5.

Finally, we take $L = 3$. We do not give the 40 new predictive distributions

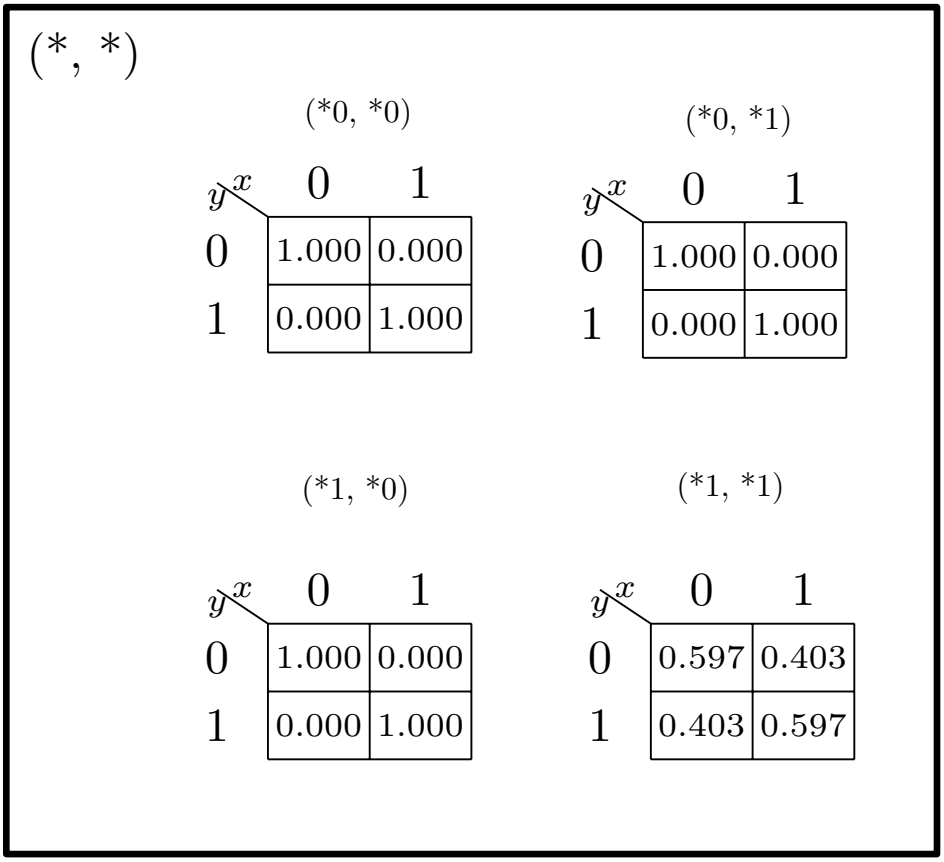


Figure A.2: The stochastic matrices $\hat{P}(X_0 = x \mid Y_0 = y, (Y_{-1}, X_{-1}) = (\mathbf{y}, \mathbf{x}))$ for the Odd Random Channel.

resulting from the 10 parent histories of length $L = 2$. Instead, we skip to the final partitioning after splitting children histories, shown in Figure A.6. We see that no new causal states were formed during this stage of the Homogenization; all of the children histories had predictive distributions equivalent to those of the existing causal states.

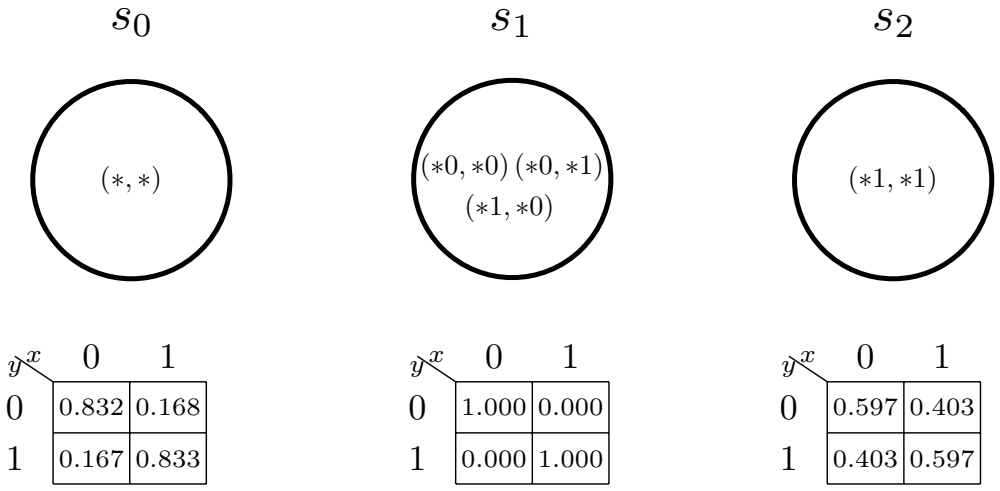


Figure A.3: The candidate causal states and their predictive distributions $P(X_t = x|Y_t = y, S_{t-1} = s)$ after the $L = 1$ Homogenization step.

A.2.2 Determinization for the Odd Random Channel

We begin the Determinization step by considering the allowed transitions between the causal states listed in Figure A.6, and removing any transient causal states. The transitions between the states is given in Figure A.7. We see that states s_1 and s_3 are transient. We outline state s_0 since it corresponds to the unique start state for transducer when we begin transduction without knowledge of whether the Odd Random Channel is in the even or odd state.

After removal of s_0 and s_2, we see that transitions between s_1 and s_3 are unifilar. Thus, the Determinization step terminates without splitting any histories from these states. The estimate of the ϵ-transducer resulting from `transCSSR` is shown in Figure A.8. By comparing to Figure A.1, we see that the inferred ϵ-

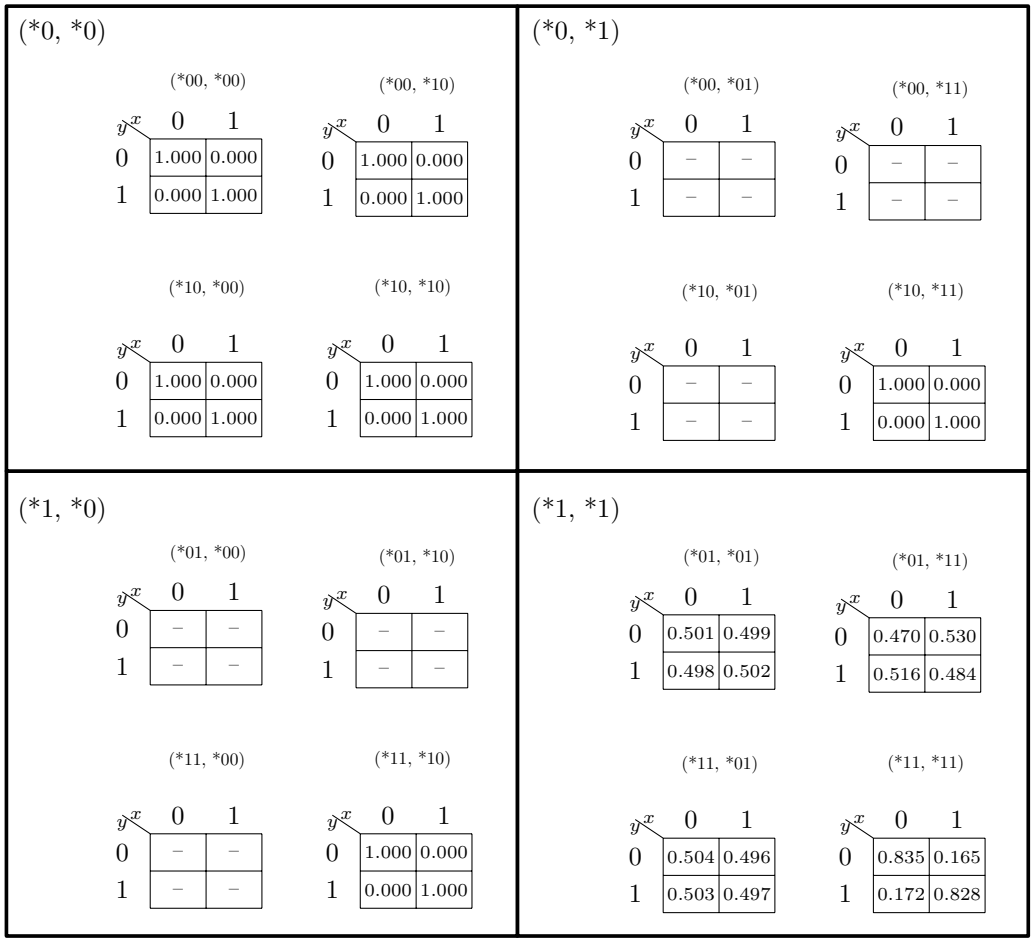


Figure A.4: The stochastic matrices $\hat{P}(X_0 = x \mid Y_0 = y, (Y_{-2}^{-1}, X_{-2}^{-1}) = (\mathbf{y}, \mathbf{x}))$ for the Odd Random Channel.

transducer agrees with the underlying ϵ-transducer up to statistical variation in the predictive probabilities.

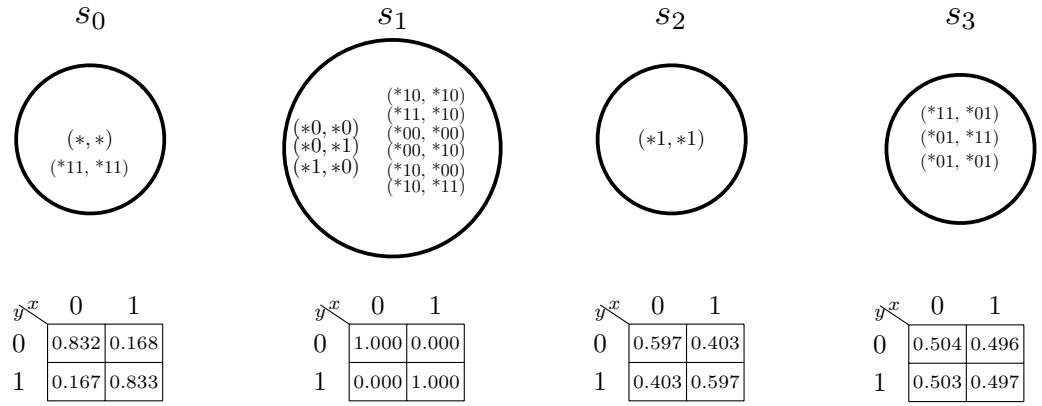


Figure A.5: The candidate causal states and their predictive distributions $P(X_t = x|Y_t = y, S_{t-1} = s)$ after the $L = 2$ Homogenization step.

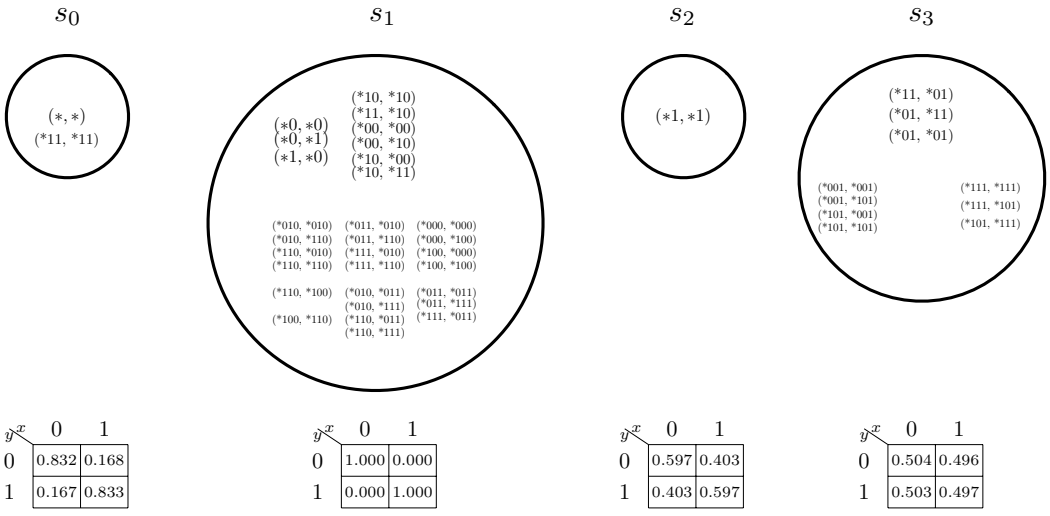


Figure A.6: The candidate causal states and their predictive distributions $P(X_t = x|Y_t = y, S_{t-1} = s)$ after the $L = 3$ Homogenization step.

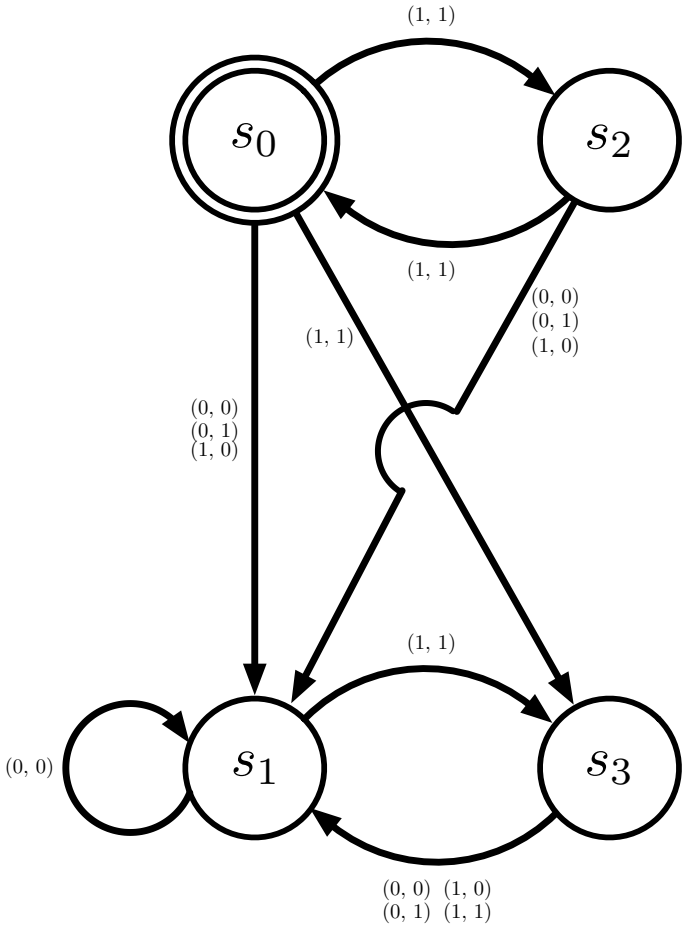


Figure A.7: The allowed transitions between the candidate causal states after the $L = 3$ Homogenization step. The edges are labeled by the input-output pair (y, x).

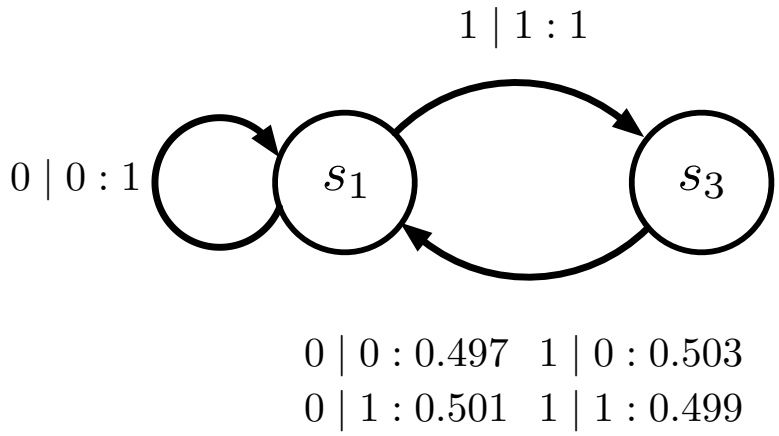


Figure A.8: The inferred ϵ-transducer for the Odd Random Channel, with $N = 100000$, $L_{\max} = 3$, and $\alpha = 0.001$. Compare to Figure A.1.

Bibliography

[1] J.B. Park, J. Won Lee, J.S. Yang, H.H. Jo, and H.T. Moon. Complexity analysis of the stock market. *Physica A: Statistical Mechanics and its Applications*, 379(1):179–187, 2007.

[2] Jae-Suk Yang, Wooseop Kwak, Taisei Kaizoji, and In-mook Kim. Increasing market efficiency in the stock markets. *The European Physical Journal B*, 61(2):241–246, 2008.

[3] Claudio Cioffi-Revilla. *Introduction to Computational Social Science: Principles and Applications*. Springer Science & Business Media, 2013.

[4] Cosma Rohilla Shalizi and James P Crutchfield. Computational mechanics: Pattern and prediction, structure and simplicity. *Journal of Statistical Physics*, 104(3-4):817–879, 2001.

[5] James P Crutchfield and Karl Young. Inferring statistical complexity. *Physical Review Letters*, 63(2):105, 1989.

[6] Cosma Rohilla Shalizi. *Causal architecture, complexity and self-organization in the time series and cellular automata*. PhD thesis, University of Wisconsin–Madison, 2001.

[7] Cosma Rohilla Shalizi. Optimal nonlinear prediction of random fields on networks. *Discrete Mathematics and Theoretical Computer Science*, pages 11–30, 2003.

[8] James P Crutchfield. Optimal structural transformations–the ϵ-transducer. Technical report, UC Berkeley Physics Research Report.

[9] Nix Barnett and James P Crutchfield. Computational mechanics of input-output processes: Structured transformations and the ϵ-transducer. *arXiv preprint arXiv:1412.2690*, 2014.

[10] Fabio Boschetti. Mapping the complexity of ecological models. *Ecological complexity*, 5(1):37–47, 2008.

[11] Dowman P Varn and James P Crutchfield. From finite to infinite range order via annealing: The causal architecture of deformation faulting in annealed close-packed crystals. *Physics Letters A*, 324(4):299–307, 2004.

[12] Robert Haslinger, Kristina Lisa Klinkner, and Cosma Rohilla Shalizi. The computational structure of spike trains. *Neural Computation*, 22(1):121–157, 2010.

[13] Asok Ray. Symbolic dynamic analysis of complex systems for anomaly detection. *Signal Processing*, 84(7):1115–1130, 2004.

[14] Peter Harremoës and Gábor Tusnády. Information divergence is more χ^2-distributed than the χ^2-statistics. In *Information Theory Proceedings (ISIT), 2012 IEEE International Symposium on*, pages 533–537. IEEE, 2012.

[15] Frank B Knight. A predictive view of continuous time processes. *The Annals of Probability*, pages 573–596, 1975.

[16] Satinder Singh, Michael R James, and Matthew R Rudary. Predictive state representations: A new theory for modeling dynamical systems. In *Proceedings of the 20th conference on Uncertainty in artificial intelligence*, pages 512–519. AUAI Press, 2004.

[17] Jianqing Fan and Qiwei Yao. *Nonlinear time series: nonparametric and parametric methods*. Springer Verlag, 2003.

[18] Trevor Hastie, Robert Tibshirani, Jerome Friedman, T Hastie, J Friedman, and R Tibshirani. *The elements of statistical learning*. Springer, 2009.

[19] Larry Wasserman. *All of nonparametric statistics*. Springer, 2006.

[20] Peter J Huber and Elvezio M Ronchetti. Robust statistics. 2009.

[21] Takeaki Kariya. Locally robust tests for serial correlation in least squares regression. *The Annals of Statistics*, pages 1065–1070, 1980.

[22] Takeaki Kariya et al. A robustness property of the tests for serial correlation. *The Annals of Statistics*, 5(6):1212–1220, 1977.

[23] Helle Bunzel and Timothy J Vogelsang. Powerful trend function tests that are robust to strong serial correlation, with an application to the prebisch–singer hypothesis. *Journal of Business & Economic Statistics*, 23(4):381–394, 2005.

[24] Sidney S Alexander. Price movements in speculative markets: Trends or random walks. *Industrial Management Review*, 2:7–26, 1961.

[25] Jeffrey D Hart. Kernel regression estimation with time series errors. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 173–187, 1991.

[26] Prabir Burman, Edmond Chow, and Deborah Nolan. A cross-validatory method for dependent data. *Biometrika*, 81(2):351–358, 1994.

[27] Jeffrey D Hart and Seongbaek Yi. One-sided cross-validation. *Journal of the American Statistical Association*, 93(442):620–631, 1998.

[28] Jeff Racine. Consistent cross-validatory model-selection for dependent data: hv-block cross-validation. *Journal of econometrics*, 99(1):39–61, 2000.

[29] Patrick S Carmack, William R Schucany, Jeffrey S Spence, Richard F Gunst, Qihua Lin, and Robert W Haley. Far casting cross-validation. *Journal of Computational and Graphical Statistics*, 18(4):879–893, 2009.

[30] Jean Opsomer, Yuedong Wang, and Yuhong Yang. Nonparametric regression with correlated errors. *Statistical Science*, 16(2):134–153, 2001.

[31] PL Davies and Arne Kovac. Local extremes, runs, strings and multiresolution. *Annals of Statistics*, pages 1–48, 2001.

[32] Fischer Black and Myron Scholes. The pricing of options and corporate liabilities. *The journal of political economy*, pages 637–654, 1973.

[33] Jeffrey D Hart and Thomas E Wehrly. Kernel regression estimation using repeated measurements data. *Journal of the American Statistical Association*, 81(396):1080–1088, 1986.

[34] Bernard W Silverman et al. Smoothed functional principal components analysis by choice of norm. *The Annals of Statistics*, 24(1):1–24, 1996.

[35] Peter Hall and Jeffrey D Hart. Nonparametric regression with long-range dependence. *Stochastic Processes and Their Applications*, 36(2):339–351, 1990.

[36] Peter M Robinson. Large-sample inference for nonparametric regression with dependent errors. *The Annals of Statistics*, 25(5):2054–2083, 1997.

[37] Iain M Johnstone and Bernard W Silverman. Wavelet threshold estimators for data with correlated noise. *Journal of the royal statistical society: series B (statistical methodology)*, 59(2):319–351, 1997.

[38] Gerda Claeskens and Nils Lid Hjort. *Model selection and model averaging*, volume 330. Cambridge University Press Cambridge, 2008.

[39] P Laurie Davies, Arne Kovac, and Monika Meise. Nonparametric regression, confidence regions and regularization. *The Annals of Statistics*, 37(5B):2597–2625, 2009.

[40] Larry Wasserman. Low assumptions, high dimensions. *Rationality, Markets and Morals*, 2(49), 2011.

[41] R Dennis Cook and Sanford Weisberg. Residuals and influence in regression. 1982.

[42] John Fox. *Regression diagnostics: An introduction*, volume 79. Sage, 1991.

[43] James Vandiver Bradley. *Distribution-free statistical tests*. Prentice-Hall, 1968.

[44] J. Crutchfield. Between order and chaos. *Nature Physics*, 8(1):17–24, 2011.

[45] Christopher J Ellison, John R Mahoney, and James P Crutchfield. Prediction, retrodiction, and the amount of information stored in the present. *Journal of Statistical Physics*, 136(6):1005–1034, 2009.

[46] Cosma Rohilla Shalizi and Kristina Lisa Klinkner. Blind construction of optimal nonlinear recursive predictors for discrete sequences. In Max Chickering and Joseph Y. Halpern, editors, *Uncertainty in Artificial Intelligence: Proceedings of the Twentieth Conference (UAI 2004)*, pages 504–511, Arlington, Virginia, 2004. AUAI Press.

[47] David Darmon, Jared Sylvester, Michelle Girvan, and William Rand. Predictability of user behavior in social media: Bottom-up v. top-down modeling. In *Social Computing (SocialCom), 2013 International Conference on*, pages 102–107. IEEE, 2013.

[48] Cosma Rohilla Shalizi, Kristina Lisa Shalizi, and James P Crutchfield. An algorithm for pattern discovery in time series. Technical Report 02-10-060, Santa Fe Institute, 2002.

[49] James P Crutchfield. The calculi of emergence: computation, dynamics and induction. *Physica D: Nonlinear Phenomena*, 75(1):11–54, 1994.

[50] Katalin Marton and Paul C Shields. Entropy and the consistent estimation of joint distributions. *The Annals of Probability*, pages 960–977, 1994.

[51] DN Dejong and C Dave. *Structural Macroeconomics*. Princeton: Princeton University Press, 2007.

[52] Robert J Hodrick and Edward C Prescott. Postwar us business cycles: an empirical investigation. *Journal of Money, credit, and Banking*, pages 1–16, 1997.

[53] Robert L Paige and A Alexandre Trindade. The hodrick-prescott filter: A special case of penalized spline smoothing. *Electronic Journal of Statistics*, 4:856–874, 2010.

[54] Timothy Cogley and James M Nason. Effects of the hodrick-prescott filter on trend and difference stationary time series implications for business cycle research. *Journal of Economic Dynamics and control*, 19(1):253–278, 1995.

[55] Morten O Ravn and Harald Uhlig. On adjusting the hodrick-prescott filter for the frequency of observations. *Review of Economics and Statistics*, 84(2):371–376, 2002.

[56] Torben Mark Pedersen. The hodrick–prescott filter, the slutzky effect, and the distortionary effect of filters. *Journal of Economic Dynamics and Control*, 25(8):1081–1101, 2001.

[57] James Douglas Hamilton. *Time series analysis*, volume 2. Princeton university press Princeton, 1994.

[58] Halbert White and Clive WJ Granger. Consideration of trends in time series. *Journal of Time Series Econometrics*, 3(1), 2011.

[59] Donald Cochrane and Guy H Orcutt. Application of least squares regression to relationships containing auto-correlated error terms. *Journal of the American Statistical Association*, 44(245):32–61, 1949.

[60] Georg M Goerg and Cosma Rohilla Shalizi. Licors: Light cone reconstruction of states for non-parametric forecasting of spatio-temporal systems. *arXiv preprint arXiv:1206.2398*, 2012.

[61] Simon DeDeo. Evidence for non-finite-state computation in a human social system. *arXiv preprint arXiv:1212.0018*, 2012.

[62] Patrick O Perry and Patrick J Wolfe. Point process modeling for directed interaction networks. *arXiv preprint arXiv:1011.1703*, 2010.

[63] Fred Rieke. *Spikes: Exploring the neural code*. The MIT Press, 1999.

[64] Greg Ver Steeg and Aram Galstyan. Information transfer in social media. In *Proceedings of the 21st International World Wide Web Conference*, pages 509–518. ACM, 2012.

[65] Yoon-Sik Cho, Aram Galstyan, Jeff Brantingham, and George Tita. Latent point process models for spatial-temporal networks. *arXiv preprint arXiv:1302.2671*, 2013.

[66] Jean-Philippe Cointet, Emmanuel Faure, and Camille Roth. Intertemporal topic correlations in online media. In *Proceedings of 1st International Conference on Weblogs & Social Media (ICWSM)*, 2007.

[67] Muntsa Padró and Llus Padró. A named entity recognition system based on a finite automata acquisition algorithm. *Procesamiento del Lenguaje Natural*, 35:319–326, 2005.

[68] Herbert Jaeger and Harald Haas. Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. *Science*, 304(5667):78–80, 2004.

[69] Matthias Salmen and Paul G Ploger. Echo state networks used for motor control. In *Proc. IEEE Conf. on Robotics and Automation (ICRA)*, pages 1953–1958. IEEE, 2005.

[70] Matthew H Tong, Adam D Bickett, Eric M Christiansen, and Garrison W Cottrell. Learning grammatical structure with echo state networks. *Neural Networks*, 20(3):424–432, 2007.

[71] S Caires and JA Ferreira. On the nonparametric prediction of conditionally stationary sequences. *Probability, Networks and Algorithms*, (4):1–32, 2003.

[72] Benjamin Schrauwen, David Verstraeten, and Jan Van Campenhout. An overview of reservoir computing: Theory, applications and implementations. In *Proc. 15th European Symposium on Artificial Neural Networks*, 2007.

[73] Herbert Jaeger. The 'echo state' approach to analysing and training recurrent neural networks. Technical Report 148, Fraunhofer Institute for Autonomous Intelligent Systems, 2001.

[74] Herbert Jaeger. Overview of reservoir recipes: A survey of new RNN training methods that follow the reservoir paradigm. Technical Report 11, School of Engineering and Science, Jacobs University, July 2007.

[75] Colin Campbell. Kernel methods: A survey of current techniques. *Neurocomputing*, 48(1):63–84, 2002.

[76] Trevor Hastie, Robert Tibshirani, Jerome Friedman, and James Franklin. The elements of statistical learning: Data mining, inference and prediction. *The Mathematical Intelligencer*, 27(2):83–85, 2005.

[77] Michael Buehner and Peter Young. A tighter bound for the echo state property. *IEEE Trans. Neural Networks*, 17(3):820–824, 2006.

[78] Mantas Lukoševičius and Herbert Jaeger. Reservoir computing approaches to recurrent neural network training. *Computer Science Review*, 3(3):127–149, 2009.

[79] Mustafa C Ozturk, Dongming Xu, and José C Príncipe. Analysis and design of echo state networks. *Neural Computation*, 19(1):111–138, 2007.

[80] Ali Rodan and Peter Tino. Minimum complexity echo state network. *IEEE Trans. Neural Networks*, 22(1):131–144, 2011.

[81] Thomas M Cover and Joy A Thomas. *Elements of information theory*. Wiley-interscience, 2012.

[82] Andrew T Stephen, Yaniv Dover, Lev Muchnik, and Jacob Goldenberg. Fresh is best: The effect of source activity on the decision to retransmit content in social media. *Available at SSRN 1609611*, 2014.

[83] Olivier Toubia and Andrew T Stephen. Intrinsic vs. image-related utility in social media: Why do people contribute content to twitter? *Marketing Science*, 32(3):368–392, 2013.

[84] Scott A Golder and Michael W Macy. Diurnal and seasonal mood vary with work, sleep, and daylength across diverse cultures. *Science*, 333(6051):1878–1881, 2011.

[85] Nir Grinberg, Mor Naaman, Blake Shaw, and Gilad Lotan. Extracting diurnal patterns of real world activity from social media. In *ICWSM*, 2013.

[86] Albert-Laszlo Barabasi. The origin of bursts and heavy tails in human dynamics. *Nature*, 435(7039):207–211, 2005.

[87] Nathan Oken Hodas and Kristina Lerman. How visibility and divided attention constrain social contagion. In *Privacy, Security, Risk and Trust (PASSAT), 2012 International Conference on and 2012 International Confernece on Social Computing (SocialCom)*, pages 249–257. IEEE, 2012.

[88] Nathan O Hodas and Kristina Lerman. Attention and visibility in an information-rich world. In *Multimedia and Expo Workshops (ICMEW), 2013 IEEE International Conference on*, pages 1–6. IEEE, 2013.

[89] Jaewon Yang and Jure Leskovec. Patterns of temporal variation in online media. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 177–186. ACM, 2011.

[90] David Darmon, Jared Sylvester, Michelle Girvan, and William Rand. Predictability of user behavior in social media: Bottom-up v. top-down modeling. In *ASE/IEEE Int'l Conf. on Social Computing*, pages 102–107, 2013.

[91] K-I Goh and A-L Barabási. Burstiness and memory in complex systems. *EPL (Europhysics Letters)*, 81(4):48002, 2008.

[92] Jure Leskovec, Lars Backstrom, and Jon Kleinberg. Meme-tracking and the dynamics of the news cycle. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 497–506. ACM, 2009.

[93] Janette Lehmann, Bruno Gonçalves, José J Ramasco, and Ciro Cattuto. Dynamical classes of collective attention in twitter. In *Proceedings of the 21st international conference on World Wide Web*, pages 251–260. ACM, 2012.

[94] Christian Bauckhage, Kristian Kersting, and Fabian Hadiji. Mathematical models of fads explain the temporal dynamics of internet memes. In *ICWSM*, 2013.

[95] Christian Bauckhage, Kristian Kersting, and Bashir Rastegarpanah. Collective attention to social media evolves according to diffusion models. In *Proceedings of the Companion Publication of the 23rd International Conference on World Wide Web*, pages 223–224, 2014.

[96] Chenhao Tan, Lillian Lee, and Bo Pang. The effect of wording on message propagation: Topic-and author-controlled natural experiments on twitter. *arXiv preprint arXiv:1405.1438*, 2014.

[97] Zongyang Ma, Aixin Sun, and Gao Cong. Will this #hashtag be popular tomorrow? In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pages 1173–1174. ACM, 2012.

[98] Liangjie Hong, Ovidiu Dan, and Brian D Davison. Predicting popular messages in twitter. In *Proceedings of the 20th international conference companion on World wide web*, pages 57–58. ACM, 2011.

[99] Sasa Petrovic, Miles Osborne, and Victor Lavrenko. Rt to win! predicting message propagation in twitter. In *ICWSM*, 2011.

[100] Jiang Yang and Scott Counts. Predicting the speed, scale, and range of information diffusion in twitter. *ICWSM*, 10:355–358, 2010.

[101] Bongwon Suh, Lichan Hong, Peter Pirolli, and Ed H Chi. Want to be retweeted? large scale analytics on factors impacting retweet in twitter network. In *Social computing, 2010 IEEE Second International Conference on*, pages 177–184. IEEE, 2010.

[102] Yiye Ruan, Hemant Purohit, David Fuhry, Srinivasan Parthasarathy, and Amit Sheth. Prediction of topic volume on twitter. *WebSci (short papers)*, 2012.

[103] Esam Alwagait and Basit Shahzad. Maximization of tweet's viewership with respect to time. In *Computer Applications & Research (WSCAR), 2014 World Symposium on*, pages 1–5. IEEE, 2014.

[104] Vasanthan Raghavan, Greg Ver Steeg, Aram Galstyan, and Alexander G Tartakovsky. Modeling temporal activity patterns in dynamic social networks. *IEEE Transactions on Computational Social Systems*, 2013.

[105] Jianqing Fan and Qiwei Yao. *Nonlinear time series*. Springer, 2002.

[106] Joachim Mathiesen, Luiza Angheluta, Peter TH Ahlgren, and Mogens H Jensen. Excitable human dynamics driven by extrinsic events in massive communities. *Proceedings of the National Academy of Sciences*, 110(43):17259–17262, 2013.

[107] Rob J Hyndman and Yeasmin Khandakar. Automatic time series for forecasting: the forecast package for r. *Journal of Statistical Software*, 27(3), 2008.

[108] Geoffrey Grimmett and David Stirzaker. *Probability and random processes*, volume 2. Oxford Univ Press, 1992.

[109] Michael L Littman, Richard S Sutton, and Satinder P Singh. Predictive representations of state. In *NIPS*, volume 14, pages 1555–1561, 2001.

[110] Stephen Eubank, VS Kumar, Madhav V Marathe, Aravind Srinivasan, and Nan Wang. Structural and algorithmic aspects of massive social networks. In *Proceedings of the fifteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 718–727. Society for Industrial and Applied Mathematics, 2004.

[111] Ryan Compton, David Jurgens, and David Allen. Geotagging one hundred million twitter accounts with total variation minimization. In *Big Data (Big Data), 2014 IEEE International Conference on*, pages 393–401. IEEE, 2014.

[112] Adam DI Kramer, Jamie E Guillory, and Jeffrey T Hancock. Experimental evidence of massive-scale emotional contagion through social networks. *Proceedings of the National Academy of Sciences*, 111(24):8788–8790, 2014.

[113] Jameson L Toole, Carlos Herrera-Yaqüe, Christian M Schneider, and Marta C González. Coupling human mobility and social ties. *Journal of The Royal Society Interface*, 12(105):20141128, 2015.

[114] Joao Gama Oliveira and Albert-László Barabási. Human dynamics: Darwin and einstein correspondence patterns. *Nature*, 437(7063):1251–1251, 2005.

[115] R Dean Malmgren, Daniel B Stouffer, Adilson E Motter, and Luís AN Amaral. A poissonian explanation for heavy tails in e-mail communication. *Proceedings of the National Academy of Sciences*, 105(47):18153–18158, 2008.

[116] R Dean Malmgren, Jake M Hofman, Luis AN Amaral, and Duncan J Watts. Characterizing individual communication patterns. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 607–616. ACM, 2009.

[117] Zhi-Qiang Jiang, Wen-Jie Xie, Ming-Xia Li, Boris Podobnik, Wei-Xing Zhou, and H Eugene Stanley. Calling patterns in human communication dynamics. *Proceedings of the National Academy of Sciences*, 110(5):1600–1605, 2013.

[118] Ye Wu, Changsong Zhou, Jinghua Xiao, Jürgen Kurths, and Hans Joachim Schellnhuber. Evidence for a bimodal distribution in human communication. *Proceedings of the national academy of sciences*, 107(44):18803–18808, 2010.

[119] Mikko Kivelä and Mason A Porter. Estimating inter-event time distributions from finite observation periods in communication networks. *arXiv preprint arXiv:1412.8388*, 2014.

[120] Gordon J Ross and Tim Jones. Understanding the heavy tailed dynamics in human behavior. *arXiv preprint arXiv:1505.01547*, 2015.

[121] Benjamin D Johnson, James P Crutchfield, Christopher J Ellison, and Carl S McTague. Enumerating finitary processes. *Theo. Comp. Sci.*, 2012.

[122] Karoline Wiesner and James P Crutchfield. Computation in finitary stochastic and quantum processes. *Physica D: Nonlinear Phenomena*, 237(9):1173–1195, 2008.

[123] Diego Rybski, Sergey V Buldyrev, Shlomo Havlin, Fredrik Liljeros, and Hernán A Makse. Communication activity in a social network: relation between long-term correlations and inter-event clustering. *Scientific reports*, 2, 2012.

[124] SE Marzen, MR DeWeese, and JP Crutchfield. Time resolution dependence of information measures for spiking neurons: Atoms, scaling, and universality. *arXiv preprint arXiv:1504.04756*, 2015.

[125] Javier Esteban, Antonio Ortega, Sean McPherson, and Maheswaran Sathiamoorthy. Analysis of twitter traffic based on renewal densities. *arXiv preprint arXiv:1204.3921*, 2012.

[126] Christian Doerr, Norbert Blenn, and Piet Van Mieghem. Lognormal infection times of online information spread. *PloS one*, 8(5):e64349, 2013.

[127] Hang-Hyun Jo, Márton Karsai, János Kertész, and Kimmo Kaski. Circadian pattern and burstiness in mobile phone communication. *New Journal of Physics*, 14(1):013055, 2012.

[128] Trevor J Hastie and Robert J Tibshirani. *Generalized additive models*, volume 43. CRC Press, 1990.

[129] S Caires and JA Ferreira. On the non-parametric prediction of conditionally stationary sequences. *Statistical inference for stochastic processes*, 8(2):151–184, 2005.

[130] Trevor Hastie, Robert Tibshirani, Jerome Friedman, T Hastie, J Friedman, and R Tibshirani. *The elements of statistical learning*, volume 2. Springer, 2009.

[131] S Marzen and JP Crutchfield. Informational and causal architecture of discrete-time renewal processes. *Entropy*, 17(7):4891–4917, 2015.

[132] Christopher C Strelioff and James P Crutchfield. Bayesian structural inference for hidden processes. *Physical Review E*, 89(4):042119, 2014.

[133] Dowman P Varn, Geoffrey S Canright, and James P Crutchfield. ϵ-machine spectral reconstruction theory: a direct method for inferring planar disorder and structure from x-ray diffraction studies. *Acta Crystallographica Section A: Foundations of Crystallography*, 69(2):197–206, 2013.

[134] Elisabeth Paulson and Christopher Griffin. Computational complexity of the minimum state probabilistic finite state learning problem on finite data sets. *arXiv preprint arXiv:1501.01300*, 2014.

[135] Benjamin Weiss. Subshifts of finite type and sofic systems. *Monatshefte für Mathematik*, 77(5):462–474, 1973.