Understanding the Predictive Power of Computational Mechanics and Echo State Networks in Social Media

David Darmon
Dept. of Mathematics
University of Maryland
College Park, MD 20740
ddarmon@math.umd.edu

Jared Sylvester
Dept. of Computer Science
University of Maryland
College Park, MD 20740
jsylvest@umd.edu

Michelle Girvan
Dept. of Physics
University of Maryland
College Park, MD 20740
girvan@umd.edu

William Rand
Ctr. for Complexity in Business
University of Maryland
College Park, MD 20740
wrand@rhsmith.umd.edu

ABSTRACT

There is a large amount of interest in understanding users of social media in order to predict their behavior in this space. Despite this interest, user predictability in social media is not well-understood. To examine this question, we consider a network of fifteen thousand users on Twitter over a seven week period. We apply two contrasting modeling paradigms: computational mechanics and echo state networks. Both methods attempt to model the behavior of users on the basis of their past behavior. We demonstrate that the behavior of users on Twitter can be well-modeled as processes with self-feedback. We find that the two modeling approaches perform very similarly for most users, but that they differ in performance on a small subset of the users. By exploring the properties of these performance-differentiated users, we highlight the challenges faced in applying predictive models to dynamic social data.

I INTRODUCTION

At the most abstract level, an individual using a social media service may be viewed as a computational agent [1]. 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 [2]. Such point process models, while very simple, have found great success in describing complicated dynamics in neural systems [3], and have recently been applied to social systems [4, 5].

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 various fields, including elucidating the computational structure of neural spike trains [6], uncovering topic correlations in social media [7], and improving named entity recognition in natural language processing [8]. 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

©ASE 2012

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 [9], motor control [10], and grammar learning [11].

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 paper, 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.

II METHODOLOGY

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

$$X_i = \left\{ \begin{array}{ll} 1 & : \exists \tau_j \in [t_0 + (i-1)\Delta t, t_0 + i\Delta t) \\ 0 & : \text{ otherwise} \end{array} \right. . \quad (1)$$

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 paper, 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

$$\hat{X}_i = \underset{x_i \in \{0,1\}}{\arg\max}\, r(x_i; x_{i-L}^{i-1}). \quad (2)$$

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

$$r(x_i; x_{i-L}^{i-1}) = P(X_i = x_i | X_{i-L}^{i-1} = x_{i-L}^{i-1}), \quad (3)$$

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 [12], the prediction function simplifies to

$$r(x_i; x_{i-L}^{i-1}) = P(X_L = x_i | X_0^{L-1} = x_{i-L}^{i-1}), \quad (4)$$

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 [13] and the other from reservoir computing [14], specifically the echo state network [15]. 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.

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

$$P(X_i|X_{i-L}^{i-1} = x) = P(X_i|X_{i-L}^{i-1} = x') \qquad (5)$$

as probability mass functions. Now, instead of considering $P(X_i|X_{i-L}^{i-1} = x_{i-L}^{i-1})$, we consider the label for the past $s_i = \epsilon(x_{i-L}^{i-1})$, and use $P(X_i|S_i = s_i)$. 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 $P(X_i|X_{i-L}^{i-1} = x)$ 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 [16]. 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

$$\hat{r}_{\text{cm}}(x_i; x_{i-L}^{i-1}) = \hat{P}(X_i = x_i|S_i = \hat{\epsilon}(x_{i-L}^{i-1})). \qquad (6)$$

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

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 [17]. 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 [18, 19].

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 $\rho(\mathbf{W}) < 1$ [20]. 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 [21]. 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

$$\mathbf{y}_t = \sigma(\mathbf{W}_{\text{in}}\mathbf{x}_t + \mathbf{W}\mathbf{y}_{t-1} + \mathbf{W}_{\text{fb}}z_{t-1}) \qquad (7)$$

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

$$z_t = \sigma(\mathbf{W}_{\text{out}}\,[\mathbf{x}_t|\mathbf{y}_t]) \qquad (8)$$

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 $\mathbf{W}_{\text{out}} = (\mathbf{S}^{-1}\mathbf{D})^{\mathsf{T}}$ which minimizes the MSE of the network on the training output.

III 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 paper, only tweets made between 7 AM and 10 PM (EST) were considered. Since most of the users in our dataset reside on the East Coast of the United States, this windowing was chosen because of the conditional stationarity assumption on $\{X_i\}_{i=1}^{T}$: users would have different conditional distributions during waking and sleeping hours. 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. In theory, this coarsening weakens our predictive ability: in the limit of infinite data, the data processing inequality tells it is always better for prediction to have raw data rather than a function of the data [22]. However, because of the practical constraints of finite data and finite computing power, the coarsening of the data allows for the inference of tractable models which are useful in practice. 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 1 demonstrates the original and coarsened time series for two users.

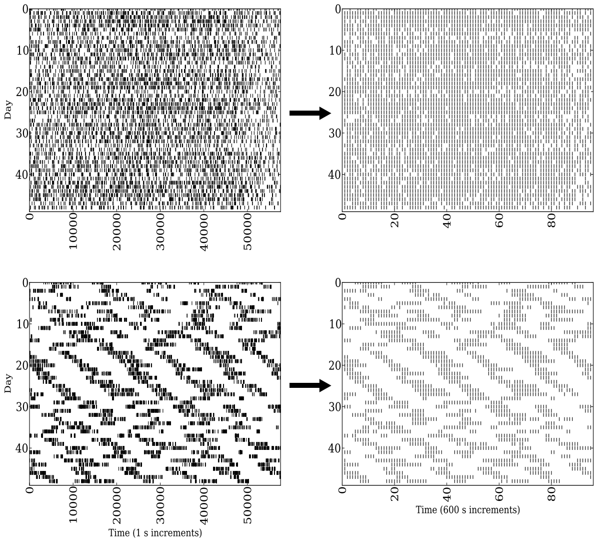

FIGURE 1: Coarsening of two users. Each row in the rastergram corresponds to a single day of activity for a fixed user. The original time series are at single second resolution, resulting in 57,600 time points in each day. After binning together activity using disjoint (partitioned) ten minute windows, there are 96 time points in each day ($T = 96$).

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 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 2.

IV RESULTS AND DISCUSSION

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 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

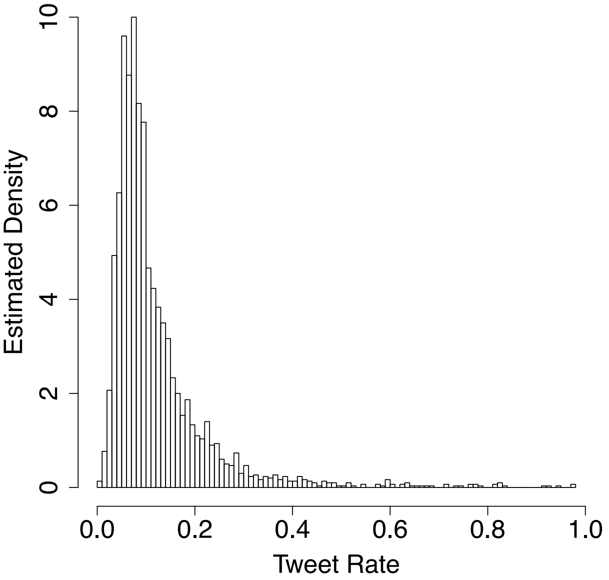

FIGURE 2: The observed distribution of the fraction of time spent tweeting (tweet rate) over the 49 day period for all of the users. 90% of the 3,000 users had a tweet rate below 0.05.

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

$$L_{\max} < \frac{\log_2 n}{h + \epsilon}, \tag{9}$$

where h is the entropy rate of the stochastic process and ϵ is some positive constant [23]. 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 0 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 [24, 25]. As a result all networks were created with $\rho(\mathbf{W}) = 0.99$ and $L_{\text{ESN}} = 10$.

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

$$\text{Accuracy Rate} = \frac{1}{n_{\text{test}}} \sum_{i=1}^{n_{\text{test}}} 1[\hat{X}_i = X_i]. \tag{10}$$

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

$$\hat{X}_i = \begin{cases} 0 & : \hat{p} \leq \frac{1}{2} \\ 1 & : \hat{p} > \frac{1}{2} \end{cases}, \tag{11}$$

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 causal state model and echo state network predictors are shown in Figure 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 4 for both methods. Breaking the users into 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 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.

©ASE 2012

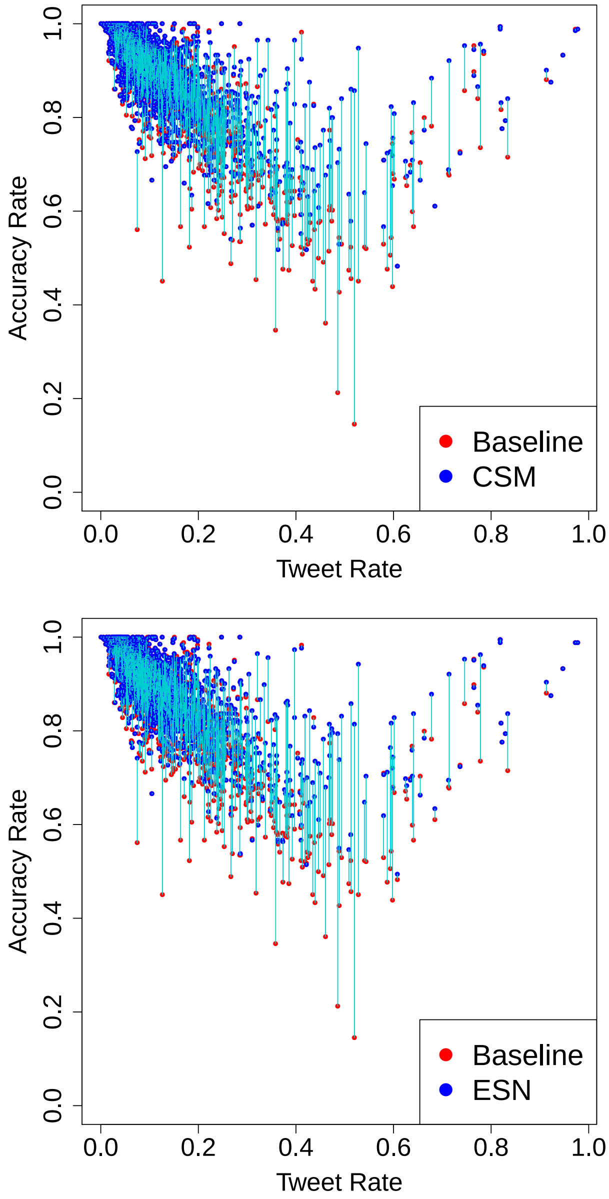

FIGURE 3: The improvement over the baseline accuracy rate for the causal state model (top) and echo state network (bottom). In both plots, each red point corresponds to the baseline accuracy rate for a user, and the connected blue point is the accuracy rate using either the causal state model or the echo state network.

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} [13].

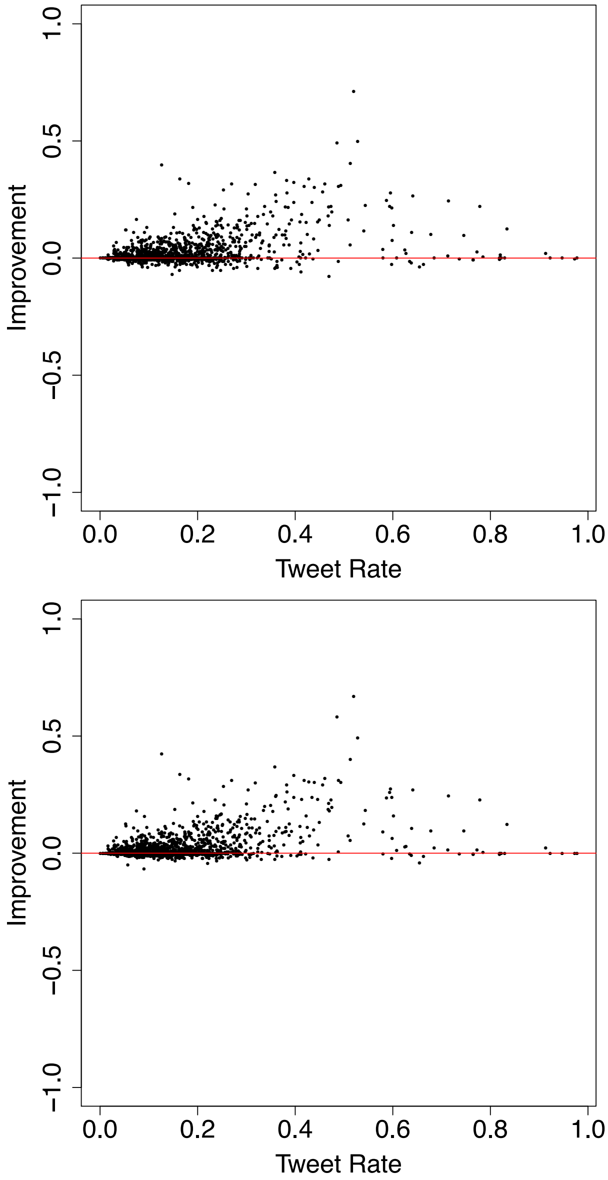

FIGURE 4: The improvement over the baseline accuracy rate for the causal state model (top) and the echo state network (bottom). For both models, the greatest improvement occurred for a coarsened tweet rate near $\frac{1}{2}$.

These two properties of a causal state model allow us to write down an emission-decorated state-space diagram for a given 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 6. Each circle corresponds to a causal state, and each arrow corresponds to an allowable transition. The arrows

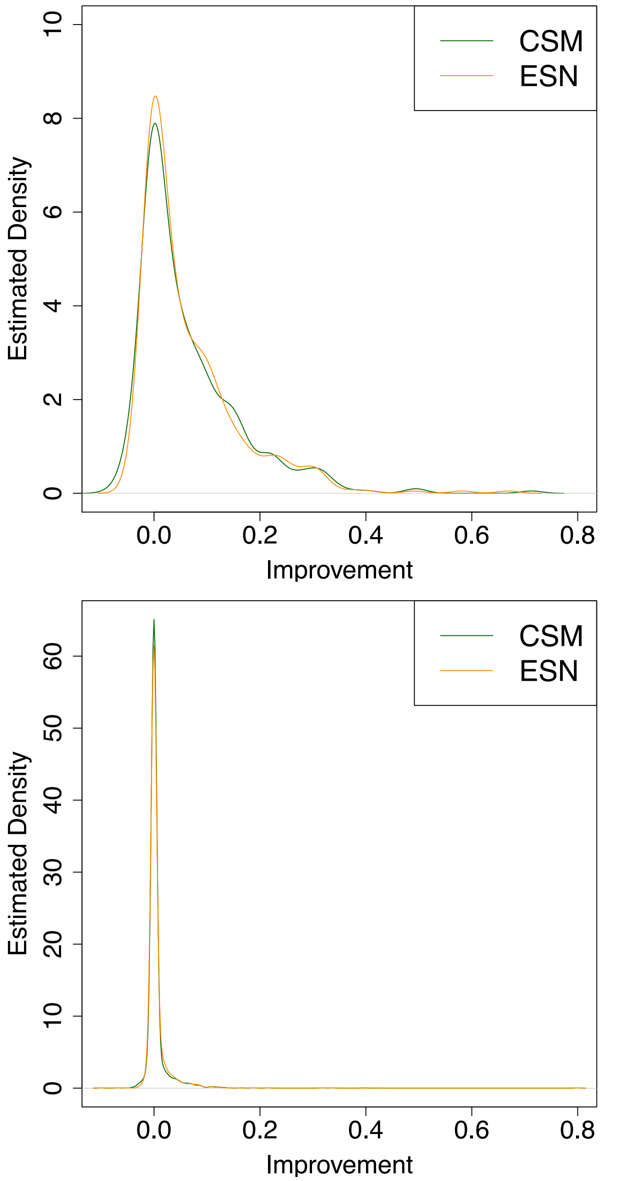

FIGURE 5: The distribution of improvements for both the causal state model (top) and echo state network (bottom), with the users partitioned into 'High Tweet Rate' (tweet rate greater than 0.2) and 'Low Tweet Rate' (tweet rate lower than 0.2) groups.

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 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 $1 - p$.

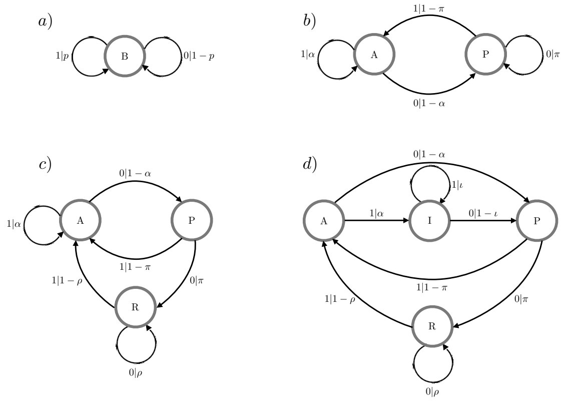

FIGURE 6: Typical 1, 2, 3, and 4-state causal state models. Of the 3,000 users, 383 (12.8%), 1,765 (58.8%), 132 (4.4%), and 100 (3.3%) had these number of states, respectively.

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 6(a) is 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 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 1.

Users corresponding to the causal state models shown in Figure 6(c) and Figure 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 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 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.

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 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 8. The causal state model inferred from the data can be used to characterize the structure of the observed dynamics in a formal manner [13]. 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. The entropy over the causal state process is called the *statistical complexity* of the process, and given by

$$\begin{aligned} C &= H[S] && (12) \\ &= -\sum_{s \in \mathcal{S}} P(S = s) \log_2 P(S = s). && (13) \end{aligned}$$

Informally, it is the number of bits of the past of a process necessary to optimally predict its future. For example, for an IID process, $C = 0$, since none of the past is necessary to predict the future, while for a period-p process, $C = \log_2 p$, 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 statistical complexity was 3.99, while the top twenty users best predicted by the echo state network had an average statistical complexity of 2.72. Figure 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) statistical complexity users.

Of the top twenty users best predicted by the echo state network, we observed 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 of a stochastic process $\{X_i\}_{i=1}^{\infty}$ is defined as the limit of the block entropies

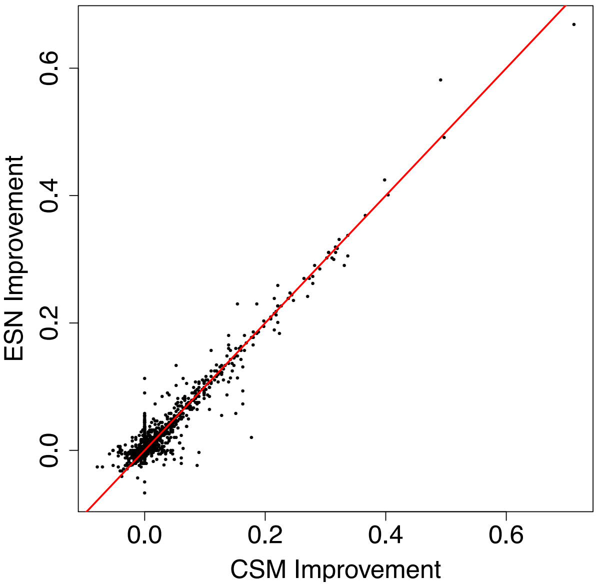

FIGURE 7: The improvement over baseline for the causal state model vs. the improvement over baseline for the echo state network. The red line indicates identity, where the two methods improve equally over the baseline predictor.

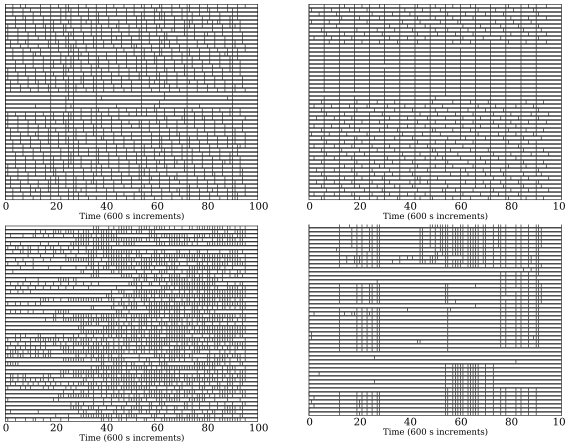

FIGURE 8: Raster plots for the four users where the causal state model most outperformed the echo state network. Note that in all but the bottom left case, the users show highly 'patterned' behavior. This is typical of the top twenty users for which the causal state model outperformed the echo state network.

of length L as the block length goes to infinity,

$$h = \lim_{L\to\infty} \frac{1}{L} H[X_1, \ldots, X_L]. \tag{14}$$

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

$$H_L = \frac{1}{L} H[X_1, \ldots, X_L] \tag{15}$$

of larger and larger block sizes and observing where the block entropies asymptote, as they must for a stationary stochastic process [22]. 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 10, in which the users have been grouped into quartiles by the absolute value of the difference between training and test set entropy rates.

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 by systematically adding noise into the time series.

The result of this experiment is shown in Figure 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 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.

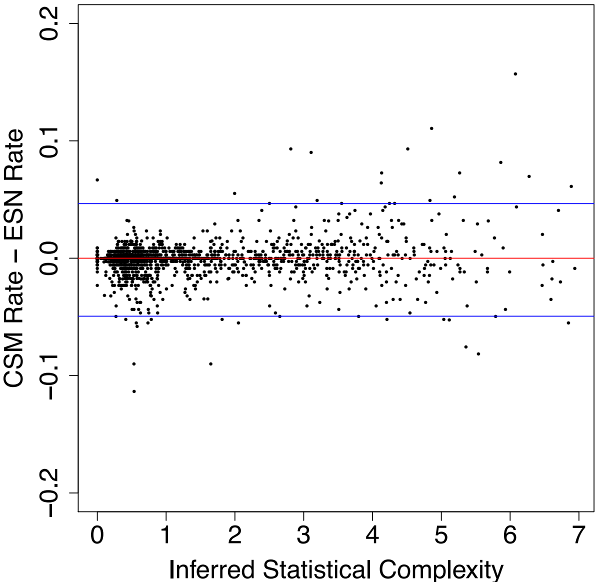

FIGURE 9: The difference in improvement between the causal state model and the echo state network for each user as a function of the inferred statistical complexity C of each user. The blue lines indicate the cutoff points above and below which the top twenty best users for the causal state model and echo state network, respectively, lie, and correspond to 0.0465 and -0.0494.

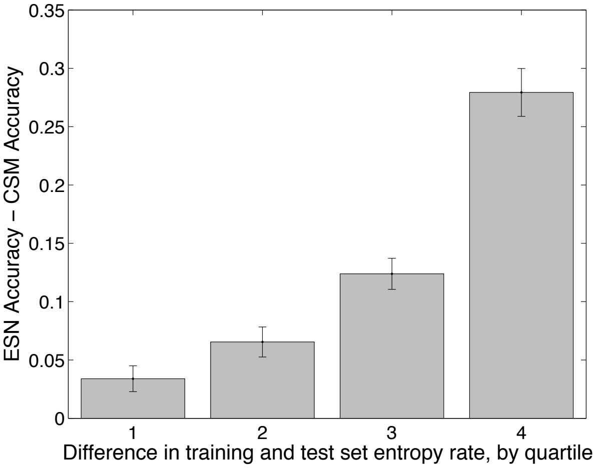

FIGURE 10: The difference in accuracy rates between the causal state model and the echo state network for each user, binned into quartiles by the absolute value of the difference in entropy rates for the training and testing sets. The causal state model performs best when this difference is low, and the echo state network performs best when it is high.

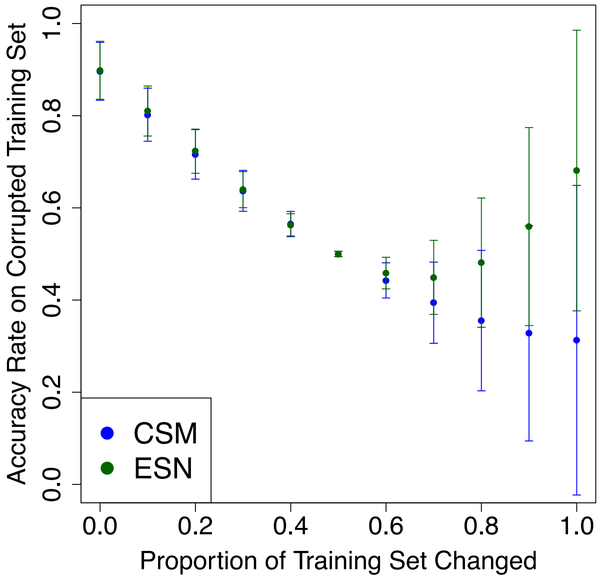

FIGURE 11: The accuracy rate of the causal state model and echo state network tested on its training data, with the training data corrupted by flipping a proportion q of the bits. Bars indicate plus or minus one standard deviation in the accuracy rates across all users.

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 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.

6 OVERVIEW OF RESULTS

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 7. Out of all the users, 58.8% had inferred causal state models similar to Figure 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.

V CONCLUSION AND FUTURE WORK

In this paper, 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 capturing 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 [26] and transducers [27], 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, which take into account social context, 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.

ACKNOWLEDGMENT

The authors gratefully acknowledge both the NSF (IIS-1018361) and DARPA (YFA-N66001-12-1-4245; STTR-D13PC00064) for their support of this research.

References

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

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

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

[4] Greg Ver Steeg and Aram Galstyan, "Information transfer in social media", in *Proc. 21st Int'l World Wide Web Conf.* ACM, 2012, pp. 509–518.

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

[6] Robert Haslinger, Kristina Klinkner, and Cosma Shalizi, "The computational structure of spike trains", *Neural Comp.*, vol. 22, no. 1, pp. 121–157, 2010.

[7] 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.

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

[9] Herbert Jaeger and Harald Haas, "Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication", *Science*, vol. 304, no. 5667, pp. 78–80, 2004.

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

[11] Matthew H Tong, Adam D Bickett, Eric M Christiansen, and Garrison W Cottrell, "Learning grammatical structure with echo state networks", *Neural Networks*, vol. 20, no. 3, pp. 424–432, 2007.

[12] S Caires and JA Ferreira, "On the nonparametric prediction of conditionally stationary sequences", *Probability, Networks and Algorithms*, vol. 4, pp. 1–32, 2003.

[13] Cosma Rohilla Shalizi and James P Crutchfield, "Computational mechanics: Pattern and prediction, structure and simplicity", *Journal of Statistical Physics*, vol. 104, no. 3-4, pp. 817–879, 2001.

[14] 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.

[15] Herbert Jaeger, "The 'echo state' approach to analysing and training recurrent neural networks", Tech. Rep. 148, Fraunhofer Institute for Autonomous Intelligent Systems, 2001.

[16] Cosma Rohilla Shalizi and Kristina Lisa Klinkner, "Blind construction of optimal nonlinear recursive predictors for discrete sequences", in *Proc. 20th Conf. on Uncertainty in Artificial Intelligence*, Max Chickering and Joseph Halpern, Eds., Arlington, Virginia, 2004, pp. 504–511, AUAI Press.

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

[18] Colin Campbell, "Kernel methods: A survey of current techniques", *Neurocomputing*, vol. 48, no. 1, pp. 63–84, 2002.

[19] Trevor Hastie, Robert Tibshirani, Jerome Friedman, and James Franklin, "The elements of statistical learning: Data mining, inference and prediction", *The Mathematical Intelligencer*, vol. 27, no. 2, pp. 83–85, 2005.

[20] Michael Buehner and Peter Young, "A tighter bound for the echo state property", *IEEE Trans. Neural Networks*, vol. 17, no. 3, pp. 820–824, 2006.

[21] Mantas Lukoševičius and Herbert Jaeger, "Reservoir computing approaches to recurrent neural network training", *Computer Science Review*, vol. 3, no. 3, pp. 127–149, 2009.

[22] Thomas M Cover and Joy A Thomas, *Elements of Information Theory*, Wiley–Interscience, 2012.

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

[24] Mustafa C Ozturk, Dongming Xu, and José C Príncipe, "Analysis and design of echo state networks", *Neural Computation*, vol. 19, no. 1, pp. 111–138, 2007.

[25] Ali Rodan and Peter Tino, "Minimum complexity echo state network", *IEEE Trans. Neural Networks*, vol. 22, no. 1, pp. 131–144, 2011.

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

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

©ASE 2012