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

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

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

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

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

***Abstract*—Recent work has attempted to capture the behavior of users on social media by modeling them as computational units processing information. We propose to extend this perspective by explicitly examining the predictive power of such a view. We consider a network of fifteen thousand users on Twitter over a seven week period. To evaluate the predictability of the users, we apply two contrasting modeling paradigms: computational mechanics and echo state networks. Computational mechanics seeks to construct the simplest model with the maximal predictive capability, while echo state networks relax from very complicated dynamics until predictive capability is reached. We demonstrate that the behavior of users on Twitter can be well-modeled as processes with self-feedback and compare the performance of models built with both the statistical and neural paradigms.**

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 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. 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.

978-0-7695-5137-1/13 $26.00 © 2013 IEEE
DOI 10.1109/SocialCom.2013.22

II. Methodology

A. 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

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

That is, 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

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

If we make certain assumptions on $\{X_i\}_{i=1}^T$, in particular that it is a conditionally stationary stochastic process [12], the optimal choice for $r(x_i; x_{i-L}^{i-1})$ is the conditional distribution of X_i given $X_{i-L}^{i-1} = x_{i-L}^{i-1}$. 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].

B. 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

$$P(X_i | X_{i-L}^{i-1} = x) = P(X_i | X_{i-L}^{i-1} = x') \quad (3)$$

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 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

$$\hat{r}_{\text{cm}}(x_i; x_{i-L}^{i-1}) = \hat{P}(X_i = x_i | S_i = \hat{\epsilon}(x_{i-L}^{i-1})). \quad (4)$$

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

C. 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. ESNs 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, ESNs 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

$$\mathbf{y}_t = \sigma(\mathbf{W}_{\text{in}}\mathbf{x}_t + \mathbf{W}\mathbf{y}_{t-1} + \mathbf{W}_{\text{fb}} z_{t-1}) \quad (5)$$

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

$$z_t = \sigma(\mathbf{W}_{\text{out}}\,[\mathbf{x}_t|\mathbf{y}_t]) \tag{6}$$

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 of the active followers of a seed user. The statuses of each user were transformed into a binary time series using their time stamp, as described above. 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 them into disjoint ten minute intervals ($\Delta t = 600$). 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 constraints of finite data and computing power, the coarsening of the data allows for the inference of tractable, practically useful models. Given a time series, we can visualize the behavior of a user over the 49 day period by using a rastergram (Fig. 1). 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.

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.

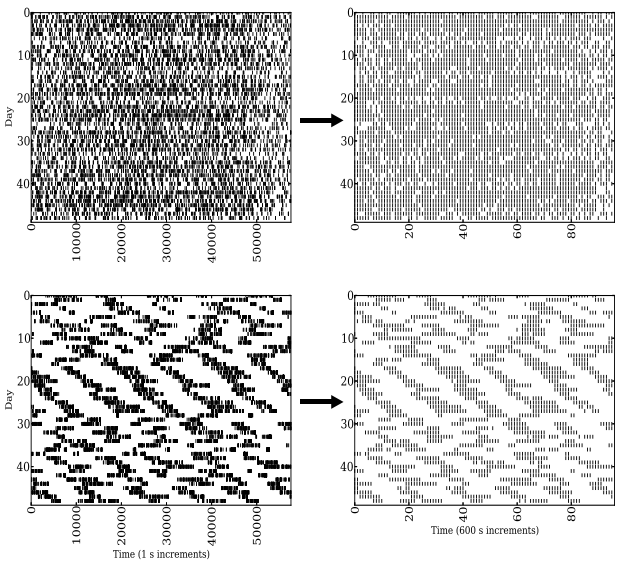


Fig. 1: Coarsening of two users. Each row in the rastergram corresponds to a single day of activity for a fixed user. The original time series are at single second resolution, resulting in 57,600 time points in each day. After binning together activity using disjoint (partitioned) ten minute windows,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

IV. Results and Discussion

A. 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 set. The maximal history length $L_{\max}$ that can be used and still ensure consistent estimation of the conditional distributions depends on the amount of data available [23]. 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 $L_{\text{ESN}} = 10$.

B. Comparison to Baseline

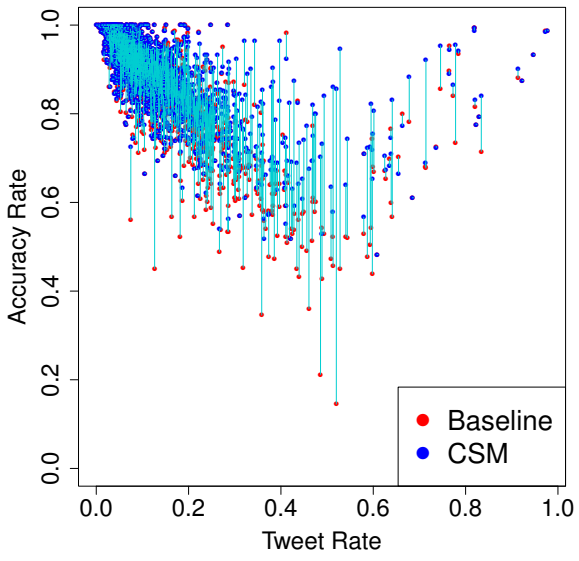
In all cases, we compute the accuracy rate of a predictor using zero-one loss. That is, for a given user, we predict the time series $X_1, \ldots, X_{n_{\text{test}}}$ as $\hat{X}_1, \ldots, \hat{X}_{n_{\text{test}}}$ and then compute the fraction of $\{\hat{X}_i\}_{i=1}^{n_{\text{test}}}$ which match $\{X_i\}_{i=1}^{n_{\text{test}}}$. We compared the accuracy rates on the CSM and ESN 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

$$\hat{X}_i = \begin{cases} 0 & : \hat{p} \leq \frac{1}{2} \\ 1 & : \hat{p} > \frac{1}{2} \end{cases} , \tag{7}$$

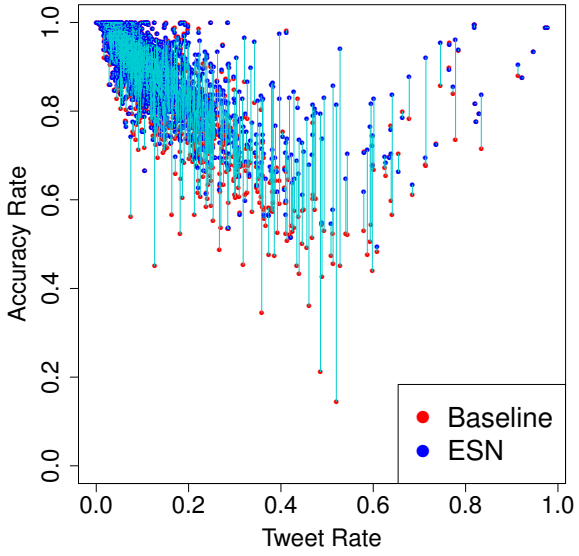
where $\hat{p} = \frac{1}{n_{\text{train}}} \sum_{j=1}^{n_{\text{train}}} X_j$. Here, the tweet rate $\hat{p}$ is computed in terms of the coarsened time series. That is, the tweet rate is the proportion of ten minute windows over the 49 day period which contain one or more tweet. For any process with 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 CSM and ESN predictors are shown in Figure 2. 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. 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.

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 4. As expected given the previous results, the improvements for each method are very strongly correlated. However, there are some differences between ESN and CSM when it comes to *which* individuals they are able to predict the best. We will investigate this difference in future work.



(a) Causal State Model



(b) Echo State Network

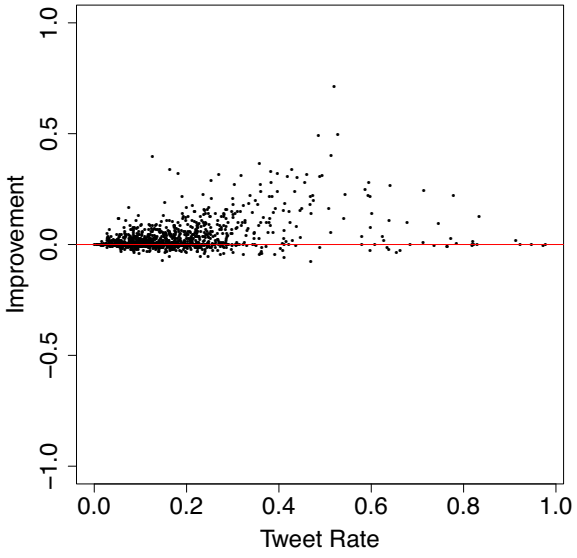
Fig. 2: The improvement over the baseline accuracy rate for the causal state model and echo state network. In both plots, each red point corresponds to the baseline accuracy rate for a user, and the connected blue point is the accuracy rate using either the causal state model or the echo state network.

V. Conclusion and Future Work

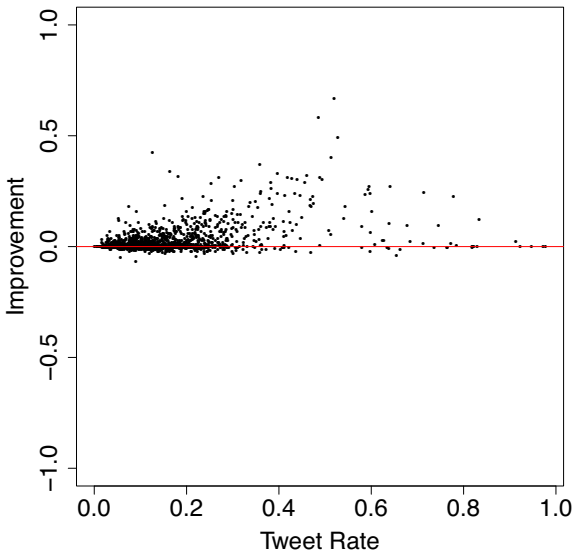
In this paper, we have shown that by building representations of the latent states of user behavior we can start to predict their actions on social media. We have done this using two different approaches, which have different ways of capturing the complexity of user behavior. Causal state modeling starts from a simple model and adds structure, while echo state networks start with complex descriptions and simplify relationships.

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 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



(a) Causal State Model



(b) Echo State Network

Fig. 3: 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 $\frac{1}{2}$.

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.

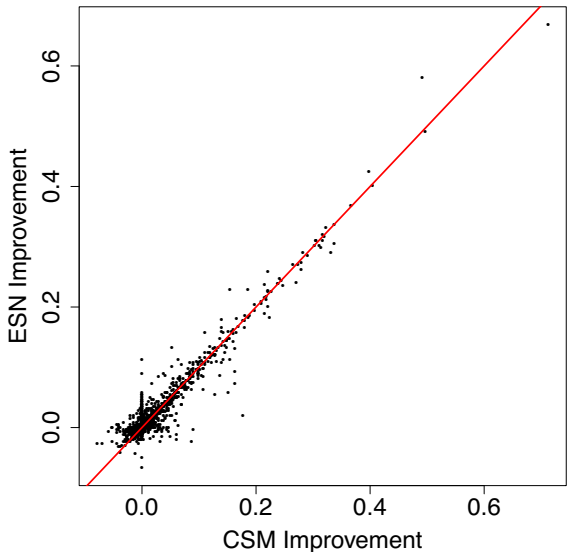


Fig. 4: The improvement over baseline for the causal state model vs. the improvement over baseline for the echo state network. The red line indicates identity, where the two methods improve equally over the baseline predictor.

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. As we have noted ESN and CSM do differ in their predictive capabilities on some users. In future work, we will investigate on which users the ESN and CSM predictions differ, and examine possible causes for those differences. 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, from a marketing perspective these models could be used to understand who will respond to a message that is sent out to a group of users, and potentially assist in determining whether or not a 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] S. DeDeo, "Evidence for non-finite-state computation in a human social system," *arXiv preprint arXiv:1212.0018*, 2012.

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

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

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

[5] Y.-S. Cho, A. Galstyan, J. Brantingham, and G. Tita, "Latent point process models for spatial-temporal networks," *arXiv preprint arXiv:1302.2671*, 2013.

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

[7] J.-P. Cointet, E. Faure, and C. Roth, "Intertemporal topic correlations in online media," in *Proceedings of 1st International Conference on Weblogs & Social Media (ICWSM)*, 2007.

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

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

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

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

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

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

[14] B. Schrauwen, D. Verstraeten, and J. Van Campenhout, "An overview of reservoir computing: Theory, applications and implementations," in *Proc. 15th European Symposium on Artificial Neural Networks*, 2007.

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

[16] C. R. Shalizi and K. L. Klinkner, "Blind construction of optimal nonlinear recursive predictors for discrete sequences," in *Proc. 20th Conf. on Uncertainty in Artificial Intelligence*, M. Chickering and J. Halpern, Eds. Arlington, Virginia: AUAI Press, 2004, pp. 504–511.

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

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

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

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

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

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

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

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

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

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

[27] ——, "Causal architecture, complexity and self-organization in the time series and cellular automata," Ph.D. dissertation, University of Wisconsin–Madison, 2001.