Evaluating Theories of Collaborative Cognition Using the Hawkes Process and a Large Naturalistic Data Set

Mohsen Afrasiabi (afrasiabi@wisc.edu)
University of Wisconsin-Madison
Department of Psychology, 1202 W. Johnson Street
Madison, WI 53706 USA

Mark G. Orr (mo6xj@Virginia.edu)
University of Virginia
Biocomplexity Institute and Initiative
charlotsville, VA 22911

Joseph L. Austerweil (austerweil@wisc.edu)
University of Wisconsin-Madison
Department of Psychology, 1202 W. Johnson Street
Madison, WI 53706 USA

Abstract

People spontaneously collaborate to solve a common goal. What factors affect whether teams are successful? Due to lack of large-scale naturalistic data and methods for investigating scientific questions on such data, previous work has either focused on very concrete cases, such as surveys of business teams, or abstract cases, such as GridWorld games, where agents coordinate their movement so that each agent can get to their own goal without obstructing other agents. We propose a computational framework based on the multivariate Hawkes process and a novel algorithm for parameter estimation on large data sets. We demonstrate the potential of this method by applying it to a large database of programming teams, public GitHub repositories. We analyze factors known to influence team performance, such as leader organization style and team cognitive diversity, as well as other factors, such as the burstiness of effort, that are difficult to test using existing methods.
Keywords: Collaborative cognition; Hawkes process; Organizational psychology; Bayesian nonparametrics

Introduction

People naturally form groups to collaborate towards a common goal. We coordinate to navigate the world (Ho et al., 2016), to protest inequalities (Korkmaz et al., 2018), to increase efficiency and well-being (Simon, 1991), to solve problems (Miller, 1951) and crises (Militello et al., 2007), to conduct science (Wuchty et al., 2007) and for many other goals. Previous work on *collaborative cognition* tends to either focus on case studies, such as using surveys of company employees (Kozlowski & Ilgen, 2006), or abstract situations, such as game-theoretic analyses of whether to cooperate or defect in the Prisoner's Dilemma (Rand & Nowak, 2013). Although these methods have been drastically increased our understanding of collaboration and competition (e.g., what mechanisms promote cooperation in competitive scenarios; Kleiman-Weiner et al. 2016; Rand & Nowak 2013), there is a need to bridge this gap. In this paper, we propose a large-scale natural data set and computational framework for analyzing human collaboration.

Technical approaches for theoretical development, conceptualization, and modeling of collaborative cognition come in many forms, each with specific strengths and weakness. For example, agent-based simulation can represent individuals interacting in dynamic network structures, but suffer from issues, such as computational difficulties in scaling the number of agents to realistic numbers, the number of free parameters (whether in model choice or explicit parameters), what the right level of abstraction should be, and how to evaluate them with respect to empirical data. This methodology has been extremely powerful, for example, it is unclear we would have discovered without these models that cooperation in the Prisoner's Dilemma can emerge from natural selection when the agents play according to how they are networked (Ohtsuki et al., 2006). But due to the simplifications, it is unclear whether this approach can be applied to any phenomena of interest (Louie & Carley, 2008).

In this paper, we focus on one aspect of collaborative cognition: how teams act as if they are a single mind when solving a common task (Searle, 1995; Bacharach et al., 2006). There are two major challenges facing collective cognition research on this perspective: (1) a lack of naturalistic data of real-world problems in the process of being solved and (2) a lack of formal methods for evaluating such data, which are richly-structured discrete data over continuous time (Kozlowski et al., 2016). For example, recent work has explored how pairs of agents can learn to coordinate and generalize their coordination in "Grid Worlds" – an environment consisting of a grid, two circle avatars in the grid, and two goals that the avatars try to get to without impeding each other (Austerweil et al., 2016; Ho et al., 2016). To address the first problem, we propose analyzing projects (called repositories) on GitHub, an online social coding platform, as a source of large-scale, naturalistic data of humans self-organizing towards solving a common goal. To address the second problem, we propose using the multivariate Hawkes process (Hawkes, 1971), a Bayesian nonparametric process, that, unlike Poisson processes, can capture the bursty nature of work on GitHub. To do so, we derive a novel approximation technique that can estimate parameters for a set of richly structured discrete data.

Introduction to GitHub

GitHub is an online social coding platform. Users can create projects, called repositories, which are publicly accessible. It is built on the decentralized software version control platform `git`. Each `git` user of a repository has a full-fledged version of the project and full control of their local version. They then can share their changes to others working on the project who can decide whether to merge them into their own repository.

Given how decentralized projects managed by `git` are and the importance of clear leadership for project success in some tasks from empirical research in Industrial and Organizational Psychology (Kozlowski & Ilgen, 2006; D. Wang et al., 2014), one may be surprised that GitHub is one of the most popular platforms for collaborative programming projects. This is because GitHub affords coordination with other team members in a few ways. (1) Only some members are "owners" of the repository, who are allowed to accept proposed changes to the project (any owner can make another member an owner – the original creator is the first owner of the repository), (2) a set of `Events` that keep track of actions taken by each member to global repositories, and (3) conversations through different media, such as e-mail lists or Reddit. Although the third method of coordinating is important, we leave it for future research. We will focus on repository ownership and events to analyze collaborative cognition on GitHub.

There are six main types of `Events` that we focus on: `CreateEvent`, `ForkEvent`, `DeleteEvent`, `PullRequestEvent`, `PushEvent`, `IssueEvent`, and `WatchEvent`. Every event is stored with the time when it occured. Some event types have subtypes that enable team members to discuss the event. A `CreateEvent` occurs when someone creates a new repository or (more commonly) creates a new "branch", which is a copy of the project attached to the main one. Branches are often used to prototype new features. Sometimes the prototype works and a team member proposes incorporating it back into the main project, which is a `PullRequestEvent` (an owner then either accepts or rejects the merger, sometimes after comments from different members). Sometimes the prototype does not work, in which case it gets deleted, which is catalogued by a `DeleteEvent`. A `PushEvent` occurs when someone updates a file in the main public repository. Team members that discover problems or want to raise other issues can do so with an `IssueEvent`. Finally, anyone interested in a project can get regular updates to any changes by "watching" the repository. Whenever a new person watches the repository, a `WatchEvent` occurs. Although these events do not catalogue all work by a team, they provide a lot of information about how team members collaborate and develop a project. We will analyze them to test theories of collaboration, but first we present our computational framework.

A Computational Framework for Teamwork

We formulate our model as a Bayesian nonparametric Point Process. It is a multivariate Hawkes process, where the dimensions correspond to the different types of Events and marks correspond to the properties of the Event. For example, an IssueEvent will be one dimension in the multivariate Hawkes process, and values of the IssueEvent (such as the user, the repository, etc) are all part of the mark.

In this section, we first define Stochastic Marked Non-Homogeneous Poisson Point Processes. Next, we define the univariate Hawkes process with a simple mark. Then, we formulate a multivariate Hawkes process. Throughout, we will introduce notation that will become increasingly catered to the special case of modeling GitHub.

Stochastic Marked Non-Homogeneous Poisson Point Processes

A *Marked Point Process* is a sequence of marked random points, where each point $H_i = (t_i, e_i)_{i=1,\dots}$ is composed of a continuous-valued time value ($t_i \in \mathcal{R}_+$, positive real numbers) and a *mark* ($e_i \in \mathcal{E}$, an arbitrary event space $\mathcal{E}$). For the specific case of modeling GitHub, marks are multivariate points taking values in the space, $\{1,2,\dots,E\} \times \{1,2,\dots,U\} \times \{1,2,\dots,R\}$, where E is the number of Event Types, U is the number of agents, and R is the number of repositories.[1] The framework allows for observed mark types to influence the rates of EventTypes, which will be important for capturing dependencies betweeen EventTypes. For example, a PushEvent is more likely after a CreateEvent then a WatchEvent.

A *Non-homogeneous Marked Poisson Point Process* is a special case of a Marked Point Process, where the number of points in a period of time $[a,b]$ is Poisson distributed with parameter $\int_a^b \lambda(t)dt$. $\lambda(t)$ is an intensity function or the instantaneous rate for points to arrive at time t. To capture relations between EventTypes, agents, and repositories, $\lambda_\theta(t)$ will be dependent on $\theta = (e,u,r)$, which corresponds to the rate of users u producing events of type e in repository r. The interactions between the stream of events for users in different repositories can be distributions other than *Possion*. They are defined as appropriate for the domain, which is how we will include psychologically-based representations in future work. For this article, we assume each repository, event types, and users are marked processes with empirical distributions extracted from real repository data.

Multivariate Hawkes Process with Agent Types, Repositories, and Communities

In the models discussed above, all events arrive independently, either at a constant rate (for Poisson process) or governed by an intensity function (for the non-homogeneous Poisson processes). In both cases, they are independent of events that previously occurred. However, in social environments, the arrival of an event increases the likelihood of observing events in the future. To model this phenomena we use a *Hawkes Point Process* with a *self-exciting* kernel in which an event arrival explicitly depend on past events (Hawkes, 1971). A *Point Process* is a *Hawkes Process* if the conditional intensity function $\lambda_r(t|H_i = (t_i, e_i)_{i=1,\dots})$ is:

$$\lambda_r^*(t) = \lambda_r(t|H_1,\dots,H_n) = \lambda_{r,0}(t) + \sum_{i:t>t_i} \phi(t - t_i;\beta) \qquad (1)$$

[1] Technically, the number of users and repositories are random variables themselves. Then the second and third dimension of the mark would each be counting processes. $U(t)$ could encode the number of users at time t and the probability of a point having a value on the second-dimension beyond $U(t)$ is null. The same can be done for repositories.

where $\lambda_{r,0}(t)$ is the repository intensity based on prior or exogenous information. The events generated from $\lambda_{r,0}(t)$ are called *immigrant* events. Note that when $\phi = 0$, we recover a *Poisson Process*. $\phi(t;\beta)$ is a kernel function and typically decays with increasing t and β are its parameters. The most common decay function is the scaled exponential taking the following form: $\phi(t;\alpha,\omega) = \alpha\omega\exp\{-\omega t\}$, where $\beta = (\alpha,\omega)$, $\alpha \geq 0$ and $\omega > 0$ and $\alpha < \omega$. Another widely used kernel for modeling social behavior is the power-law function: $\phi(t;\alpha,\eta,\gamma) = \alpha(t+\gamma)^{-(\eta+1)}$, where $\alpha \geq 0$, $\gamma > 0$, $\eta > 0$ and $\alpha < \eta\gamma^{\eta}$.

After observing an event, the intensity is large for some time and then decays to zero. Thus, more recent events influence the current event's intensity more than older events. This results in a *self-excitatory* process, where bursts of points in a small time period lead to a large increase in intensity in that region. By defining $\phi(t)$ differently, it is also possible to capture *self-inhibiting* processes (Yang et al., 2015), which will be important in capturing an user waiting for other users (e.g., respond to an `IssueEvent`). Both properties violate the *memoryless* property, and thus, Hawkes processes capture a broader set of Point Processes then standard nonhomogeneous Possion Processes.

As our model is multi-user, multi-event and multi-repository we will use the *multivariate* formulation of the Hawkes process. The basic assumption behind the multivariate Hawkes process is that the arrival of an event in one dimension can affect the arrival rates of events in other dimensions according to some generative process. The specification of the generative process can be as richly structured as appropriate for the domain. This enables analysis of structured discrete data over continuous events. We model this dependence in the following manner: each repository is a Hawkes process, the Hawkes processes for repositories are interdependent, and the event types and users as marks. In this paper, we use pairwise correlations to capture repository interdependence and the joint probability of pairs of Event Types is estimated from our data set.

Using an exponential kernel function, the conditional intensity $\lambda_r^*(t)$ is:

$$\lambda_r^*(t) = \lambda_{r,0}(t) + \sum_{i:t>t_i} \alpha_{r_i,r}\omega_{r_i,r}\exp(\omega_{r_i,r}(t-t_i)), \quad (2)$$

where $\alpha_{r_i,r}$ is an interactivity matrix defining how the r_i dimension influences the r dimension given the values of features across the different dimensions at time t. We approximate this matrix via maximum likelihood estimation. The likelihood of repository r with parameter set $\beta = (\alpha,\omega)$ and λ_0 is (Ozaki, 1979):

$$l_r = \exp\left\{-\int_0^T \lambda_r(t|\,\{t_j\}_{j=1}^N)dt\right\}\prod_{i=1}^N \lambda_r(t_i|\,\{t_j\}_{j=1}^{i-1}) \quad (3)$$

and the log-likelihood, with some simplification, is:

$$\begin{aligned}\log l_r(\{t_i\}\,|\eta_r) = &-\lambda_{r,0}T + \sum_{i=1}^N \alpha_r\left(\exp(-\omega_r(T-t_i)) - 1\right)\\ &+ \sum_{i=1}^N \log\left(\lambda_{r,0} + \alpha_r\omega_r\Omega_r(i)\right)\end{aligned}$$

where $\Omega_r(i) = \sum_{t_j<t_i}\exp(-\omega_r(t_j-t_i)), \forall i \geq 2$ and $\Omega_r(1) = 0$.

Unfortunately we cannot optimize the log-likelihood directly, because the curvature vanishes. So, we estimate the parameters by extending a version of *Maximum a Posteriori Expectation Maximization* (Zipkin et al., 2016). Let $\tau = (t_i)$ be the sequence of actions performed on a repository and $M = M_{ij}$ be a branching matrix of an immigrant event, where $M_{ij} = 1$ if event i is an offspring of event j. M is the causal cascade structure of sequence of actions performed in a repository. Let $p(\Upsilon;F)$ be a prior on $\Upsilon = (\eta,\lambda_0)$ with hyperparameter F. We perform *MAP* estimation using the EM algorithm to maximize the event stream posterior, $p(\Upsilon|\tau,M) \propto p(\tau,M|\Upsilon)p(\Upsilon|F)$. Let $\log P(\tau,M|\Upsilon,F) = \log p(\tau,M|\Upsilon) + \log p(\Upsilon|F)$ be the event stream probability. We decompose the first term in the following manner: $\log p(\tau,M|\Upsilon) = \mathfrak{L}_1(\lambda_0,\tau) + \mathfrak{L}_2(\eta,\tau) + \mathfrak{L}_3(\eta,\tau)$ where

$$\begin{aligned}\mathfrak{L}_1(\lambda_0,\tau) &= -\lambda_0 T + b(\log\lambda_0 + \log T) - \log m!\\ \mathfrak{L}_2(\eta,\tau) &= -n\Phi(\eta) + \sum_i d_i\Phi(\eta) - \log m_i!\\ \mathfrak{L}_3(\eta,\tau) &= \sum_{ij} M_{ij}\left[\log\phi(t_i-t_j;\theta) - \log\Phi(\theta)\right]\end{aligned}$$

where $m = \sum_i M_{ii}$, $m_i = \sum_j M_{ij}$, and $\Phi(\eta) = \int_0^\infty \phi(t;\eta)dt$.

$$\begin{aligned}\log p(\tau,M;\Upsilon) = &-\lambda_0 T + m\log\lambda_0 + b\log T - \log(m!) +\\ &\sum_i[-\Phi(\eta) + m_i\log\Phi(\eta)\log(m_i!)]\\ &+\sum_{ij} M_{ij}\log\phi(t_i-t_j;\eta) - \log\Phi(\eta)\end{aligned}$$

In the E-step of the MAP EM algorithm, we compute the current distribution over M. As M is a matrix of branching variables, each is Bernoulli and so M can be expressed as the expected branching matrix $P = [p_{ij}]$ based on the data τ and our current parameter estimate Υ^k. The expected branching matrix at each iteration is $P^{k+1} = \mathbb{E}[M|\tau,\Upsilon^k]$. In the M-step, we update our parameter estimate to maximize the expectation of the event stream posterior log-likelihood:

$$\begin{aligned}\Upsilon^{k+1} &= \arg\max_{\Upsilon}\mathbb{E}[\mathfrak{L}(\tau,M;\Upsilon,F)|M = P^{k+1}]\\ &= \arg\max_{\Upsilon}(\mathbb{E}[\log p(\tau,M;\Upsilon)|M = P^{k+1}] + (\mathbb{E}[\log p(\Upsilon,F)])\end{aligned}$$

We use a Gamma prior on α and ω, with parameters (s,t) and (u,v), respectively. Extending the method in Zipkin et al. (2016), the EM update steps can be derived using the immigrant/offspring interpretation. The ith event is either an immigrant or an offspring of one of the previous events. The probability that the ith event is an immigrant event is proportional to λ_0^k, while the probability that it is an offspring of event j for $j < i$ is proportional to the kernel function $\phi(t_i - t_j; \alpha^k, \omega^k)$. The E-step update then is

$$P_{ij}^{k+1} = \begin{cases} \frac{1}{\Lambda^k(i)} & \text{for } i = j \\ \frac{1}{\Lambda^k(i)}\phi(t_i - t_j; \alpha^k, \omega^k) & \text{for } j < i \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

where the normalization factor is $\Lambda^k(i) = \lambda_0^k + \sum_{j<i} \phi(t_i t_j; \alpha_k, \omega_k)$. Finally the M-step is

$$\mu^{k+1} = \frac{1}{T}\sum_i P_{ii}^{k+1} \quad \alpha^{k+1} = \frac{1}{n+t}[\sum_{j<i} P_{ij}^{k+1} + s - 1] \quad (5)$$

$$\omega^{k+1} = \frac{\sum_{j<i} P_{ij}^{k+1} + s - 1}{\sum_{j<i} P_{ij}^{k+1}(t_i - t_j) + v} \quad (6)$$

Analyzing Teamwork on GitHub

We now present how GitHub can be used as a naturalistic, large-scale data set and the Hawkes process to analyze the dynamics of collaborative cognition. We used a data set of events from public repositories on GitHub at the start of midnight on March 1st 2017 to 11:59pm on August 31st 2017. We retrieved 456,195 events across 8,083 repositories.

One issue is that not all repositories are collaborative projects. For example, many repositories are used for web pages, software tutorials (e.g., learning how to fork repositories), and other personal usage. Further, many projects become inactive and abandoned without being deleted. We follow best practices for studying GitHub repositories from previous work in computer science (Kalliamvakou et al., 2016) by filtering repositories according to the following criteria: (1) there are at least 10 Events (not counting `WatchEvent`) in the data set, and (2) at least three unique "active" users. We define an active user of a repository to be someone who had at least one `CreateEvent` or `PushEvent` with it. Using these criteria, our filtered data set was comprised of 390,277 events across 1,235 repositories. This leaves us with 86% and 15% of the total events and repositories, respectively.

Are Hawkes Processes Really Necessary?

Before testing collaborative cognition hypotheses, we provide some justification for using a more complex process, a Hawkes process, rather than a standard Poisson process. From a qualitative perspective, Figure 2 shows the stream of events over time from a representative project and the best fits from a Poisson process and a Hawkes process using an exponential and power-law kernel. Due to its memorylessness property, the Poisson process is simply unable to recreate the bursty dynamics of the event stream. For our data, the Hawkes process with an exponential kernel provides the best qualitative and quantitative fit. Thus, for the remainder of the paper, we only consider the Hawkes process with an exponential kernel. A quantitative comparison of the model fits is computationally challenging due to the large number of repositories. Thus, we approximated by calculating the root

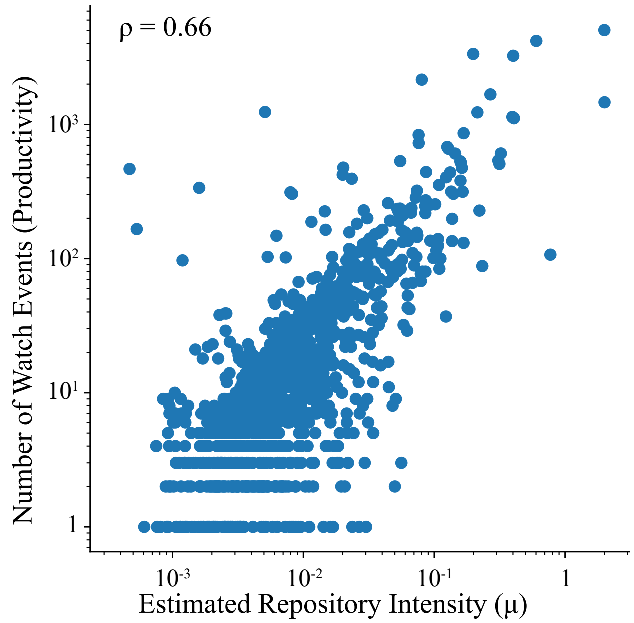

Figure 1: The repository intensity (μ) of the Hawkes Process as estimated from the GitHub data. It corresponds closely to the productivity of the repository.

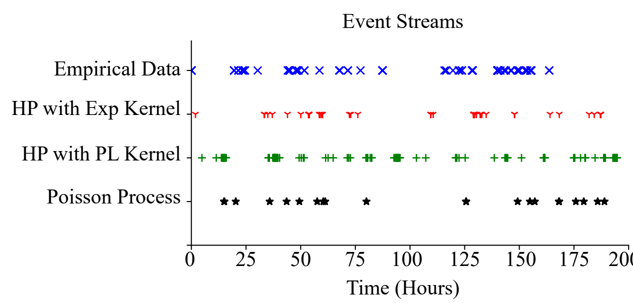

Figure 2: A representative GitHub event stream and samples from a best-fit Poisson Process and Hawkes Processes with an exponential and a power-law kernel.

mean squared error (RMSE) of 200 randomly sampled repositories and then 200 randomly sampled events within each of those repositories. The approximate RMSE for the Hawkes and Poisson processes were 7.27 and 11.81. Further, Figure 1 the number of watch events is closely related to the estimated repository intensity ($\rho = 0.66, p < 0.001$), validating our novel estimation procedure.

Testing collaborative cognition

We now turn to testing three different phenomena in collaborative cognition and assess how they affect performance: leadership organization style, diversity, and event dynamics. There is no clear definition of what makes a repository successful on GitHub (especially one that can be automatically applied to all repositories). We use the number of

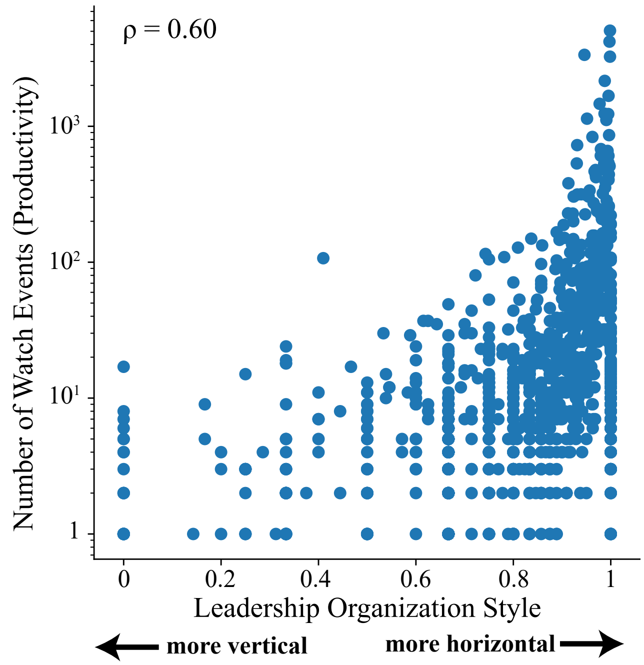

Figure 3: Shared leadership is more successful.

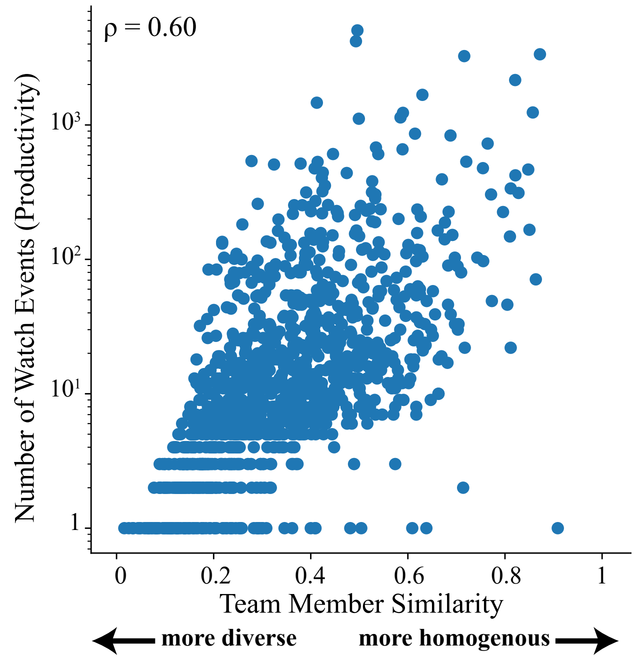

Figure 4: Teams with less cognitive diversity are more productive. The cognitive role of a team member was quantified as the distribution of event types that they produced.

`WatchEvents` for a repository in the six month period as a measure of project success. When a person chooses to *watch* a repository, it means they receive regular updates on any changes to the repository. These are people who are interested in the progress of a project, but do not necessarily contribute to it. In fact, they probably do not, as previous work found that only about 5% of people who watch a repository end up contributing to it (Sheoran et al., 2014).

Leadership organization style. Previous survey studies and meta-analyses of them have found that shared leadership (what we call "horizontal") is positively associated with group performance (D. Wang et al., 2014). We test whether this relationship holds in our large-scale, naturalistic collaboration data set. Team members in a repository are split into two groups: *owners* and *users*. Users can create their own version of a project and build on it on their own. However, they can only propose changes to the global repository (or the team's project). We define leadership style as the percentage of active users who are not owners that work on the project. Lower scores imply a vertical leadership style, where only a few team members are leaders. Larger scores imply a horizontal leadership style, where most team members are leaders. As shown in Figure 3, most teams are horizontally organized and there is a strong positive relation between horizontal organization and performance ($\rho = 0.60, p < 0.001$).

Cognitive Diversity. How does the diversity of roles within a team affect performance? Recent work found that diversity of roles (cognitive diversity) is positively related to team creativity when there are leaders that serve as role models for other team members, but negatively associated otherwise (X.-H. Wang et al., 2016). Given that we found higher performance in programming projects when the leadership style was more distributed, we expect that cognitive diversity may hurt productivity on GitHub, rather than enhance it.

To assess the role of cognitive diversity in team performance on GitHub, we quantified the similarity between two users as the inner product of the distributions of events produced by each user across all repositories. The diversity score of a repository was defined to be the average pairwise similarity of active repository users. Due to computational constraints, for repositories with many users, we approximated the quantity by averaging 10,000 randomly selected pairs of users. Figure 4 shows that teams with less diverse roles performed better ($\rho \approx 0.60, p < 0.001$).

Bursts. Are particular leadership organizations related with differences in how bursty the team's progress is on the project? Is burstiness related to performance? Thanks to the Hawkes process formalism, we can address this question by examining the relation between leadership style and the fit α parameter associated with the repository. Interestingly, Figure 5 shows that more centralized leadership organization is associated with burstier progress ($\rho = 0.39, p < 0.001$). However, burstiness has only a very weak effect on performance ($\rho = -0.13, p < 0.001$). Note that this analysis was only possible to conduct due to the computational formalism for analyzing teamwork presented in this paper.

Discussion, Limitations, and Conclusions

In this article, we proposed, validated, and used a novel computational framework for analyzing large-scale real-world

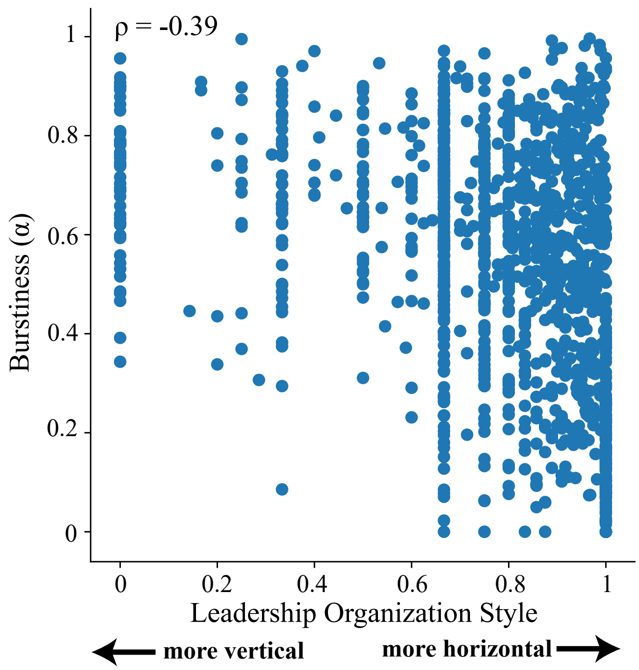

Figure 5: More vertically organized leadership styles are burstier.

collaboration data: The Multivariate Hawkes Process. We demonstrated how it can be used to test constructs in collaborative cognition. For example, we found that horizontal leadership structures were more successful. This may be specific to programming projects that naturally break into different pieces that can be worked on individually and integrated later. Future work will need to follow up on this and the other findings

As a proof of concept, we made a number of assumptions and simplifications. We assumed the only relation between events and teams are pairwise correlations. Further, we ignored an event's content, focusing on statistical patterns. In future work we plan to extend our work to address these limitations and incorporate social and cognitive principles (e.g., scripts for how events usually occur on GitHub; Schank & Abelson 1977), and examine whether the framework generalizes to analyzing other social domains (e.g., Reddit). Recent work suggests cognitive structures, such as shared memory, are essential for understanding team performance (DeChurch & Mesmer-Magnus, 2010). Additionally, we assumed that our results generalize to all task types solved by teams. However, psychologists have organized task types into ontologies (Wildman et al., 2012), and we plan to examine whether our results generalize across tasks. Shared programming projects may lend themselves more naturally to distributed, horizontal leadership structure, whereas a clear leader or established organizational identity may be needed to solve other tasks, such as putting out a fire (Mesmer-Magnus et al., 2018).

Our computational framework is built using probabilistic modeling. This enables us to conduct principled analyses that would otherwise be difficult or impossible in other frameworks. Recent work has analyzed determining automated interventions on social media using a similar probabilistic modeling framework (Farajtabar et al., 2017). For example, using point processes and Markov decision processes, Farajtabar et al. (2017) created a method for mitigating the spread of Fake News through online social networks. We are excited to adapt these techniques into our framework, which would enable us to see how intervening on GitHub repositories (e.g., stopping support for TensorFlow) or counterfactual questions (e.g., how would machine learning applications be affected if TensorFlow were never made public).

Acknowledgments

The research is based upon work supported by the Defense Advanced Research Projects Agency (DARPA), via the Air Force Research Laboratory (AFRL; contract #: FA8650-18-C-7826). The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of DARPA, the AFRL or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon.

References

Austerweil, J. L., Brawner, S., Greenwald, A., Hilliard, E., Ho, M. K., Littman, M. L., … Trimbach, C. (2016). The impact of outcome preferences in a collection of non-zero-sum grid games. In *AAAI Spring Symposium 2016 on Challenges and Opportunities in Multiagent Learning for the Real World.*

Bacharach, M., Gold, N., & Sugden, R. (2006). *Beyond individual choice: Teams and frames in game theory.*

DeChurch, L. A., & Mesmer-Magnus, J. R. (2010). The cognitive underpinnings of effective teamwork: A meta-Analysis. *Journal of Applied Psychology*, *95*(1), 32–53.

Farajtabar, M., Yang, J., Ye, X., Xu, H., Trivedi, R., Khalil, E., … Zha, H. (2017). Fake news mitigation via point process based intervention. In *ICML.*

Hawkes, A. (1971). Spectra of some self-exciting and mutually exciting point processes. In *Biometrkia* (pp. 89–90).

Ho, M. K., MacGlashan, J., Hilliard, E., Trimbach, C., Brawner, S., Gopalan, N., … Austerweil, J. L. (2016). Feature-based joint planning and norm learning in collaborative games. In *Proceedings of the 38th annual meeting of the cognitive science society.*

Kalliamvakou, E., Gousios, G., Blincoe, K., Singer, L., German, D. M., & Damian, D. (2016). An in-depth study of the promises and perils of mining GitHub. *Empirical Software Engineering*, *21*, 2035–2071.

Kleiman-Weiner, M., Ho, M. K., Austerweil, J. L., Littman, M. L., & Tenenbaum, J. B. (2016). *Coordinate to coop-*

erate or compete: Abstract goals and joint intentions in social interaction.

Korkmaz, G., Monica, C., Kraig, A., Lakkaraju, K., Kuhlman, C. J., & Vega-Redondo, F. (2018). Coordination and common knowledge on communication networks. In *Proceedings of the 17th international conference on autonomous agents and multiagent systems* (pp. 1062–1070).

Kozlowski, S. W. J., Chao, G. T., Grand, J. A., Braum, M. T., & Kuljanin, G. (2016). Capture the multilevel dynamics of emergence: Computational modeling, simulation, and virtual experimentation. *Organizational Psychological Review*, *6*(1).

Kozlowski, S. W. J., & Ilgen, D. R. (2006). Enhancing the effectiveness of work groups and teams. *Psychological Science in the Public Interest*, *7*(3), 77-124.

Louie, M. A., & Carley, K. M. (2008). Balancing the criticisms: Validating multi-agent models of social systems. *Simulation Modelling Practice and Theory*, *16*, 242–256.

Mesmer-Magnus, J. R., Asencio, R., Seely, P. W., & DeChurch, L. A. (2018). How organizational identity affects team functioning: The identity instrumentality hypothesis. *Journal of Management*, *44*(4), 1530–1550.

Militello, L. G., Patterson, E. S., Bowman, L., & Wears, R. (2007). Information flow during crisis management: challenges to coordination in the emergency operations center. *Cognition, Technology & Work*, *9*(1), 25–31.

Miller, G. A. (1951). *Language and communication*.

Ohtsuki, H., Hauert, C., Lieberman, E., & Nowak, M. A. (2006). A simple rule for the evolution of cooperation on graphs and social networks. *Nature*, *441*, 502–505.

Ozaki, T. (1979). Maximum likelihood estimation of hawkes' seld-exciting point processes. In *Annual institure of statistical mathematics 31* (pp. 145–155).

Rand, D. G., & Nowak, M. A. (2013). Human cooperation. *Trends in Cognitive Sciences*, *17*(8), 413-425.

Schank, R., & Abelson, R. (1977). *Scripts, plans, goals, and understanding*. LEA.

Searle, J. R. (1995). *The construction of social reality*.

Sheoran, J., Blincoe, K., Kalliamvakou, E., Damian, D., & Ell, J. (2014). Understanding "Watchers" on GitHub. In *Proceedings of the 11th working conference on mining software repositories* (p. 336-339). NY, NY: ACM.

Simon, H. A. (1991). Organization and markets. *Journal of economic perspectives*, *5*(2), 25-44.

Wang, D., Waldman, D. A., & Zhang, Z. (2014). A meta-analysis of shared leadership and team effectiveness. *Journal of Applied Psychology*, *99*(2), 181–198.

Wang, X.-H., Kim, T.-Y., & Lee, D.-R. (2016). Cognitive diversity and team creativity: Effects of team intrinsic motivation and transformational leadership. *Journal of Business Research*, *69*, 3231–3239.

Wildman, J. L., Thayer, A. L., Rosen, M. A., Salas, E., Mathieu, J. E., & Rayne, S. R. (2012). Task types and team-level attributes: Synthesis of team classification literature. *Human Resource Development Review*, *11*(1), 97–129.

Wuchty, S., Jones, B. F., & Uzzi, B. (2007). The increasing dominance of teams in production of knowledge. *Science*, *316*, 1036–1039.

Yang, Q., Wooldridge, M. J., & Zha, H. (2015). Trailer generation via a point process-based visual attractiveness model. In *ICAI* (pp. 2198–2204).

Zipkin, J. R., Schoenberg, F. P., Coronges, K., & Bertozzi, A. L. (2016). Point-process models of social network interactions: parameter estimation and missing data recovery. In *European journal of applied mathematics* (p. 27(03):502529).