

# Collaborative Dynamics in Distributed Software Development Projects

Alessandro Lomi <sup>1</sup>    Duy Vu <sup>2,1</sup>

<sup>1</sup>Università della Svizzera Italiana, Lugano (Switzerland)

<sup>2</sup>University of Melbourne (Australia)

XXI Organization Science Winter Conference  
February 5 - 8, 2015

# Agenda

Motivation and background

Empirical setting

Relational event models

Model specification

Results

Discussion

# Motivation

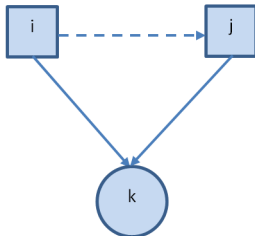
# Homophily and Influence

The co-evolution of one-mode and two-mode networks:

- ▶ One-mode following network:
  - ▶ A directed network similar to Twitter
  - ▶ Developers follow each other to get updates about each other's activities.
- ▶ Two-mode networks:
  - ▶ Commit network: developers commit their code to code repositories, i.e. projects.
  - ▶ Comment network: developers discuss issues within each code repository, i.e. bugs or new features.

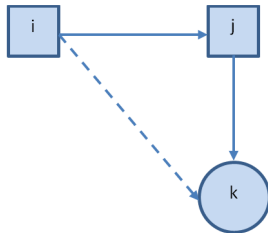
# Homophily

Developers who commit code to the same repositories are more likely to follow each other.



# Influence

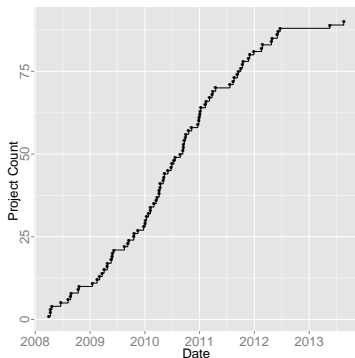
Developers tend to commit code to repositories that their followed coders or *followees* also contribute to.



# Code repositories

The data set contains 90 projects in GitHub which have highest stars across different programming languages:

Time	Repo ID	Name	Language
2008-04-01 05:20:41	5326	beanstalkd	C
2008-04-11 03:58:20	104307	paperclip	Ruby
...	...	...	...



# Programming languages

Distribution of 90 top starred projects over 13 programming languages:

Programming Language	Count
C	10
C#	8
C++	8
CSS	3
Go	1
Java	8
JavaScript	9
PHP	9
Python	10
R	4
Ruby	10
Scala	9
TypeScript	1



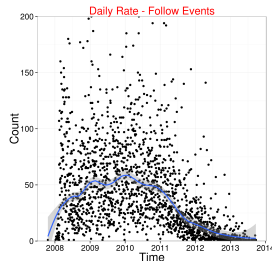
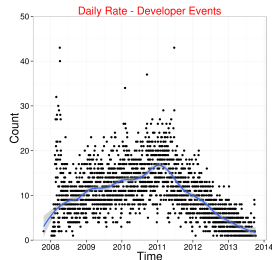
# Developers

25866 developers who have made at least one contribution to these 90 projects:

Time	User ID
2008-01-20 08:52:30	11026
2008-01-22 19:14:11	28799
...	...

82994 follow events between these developers:

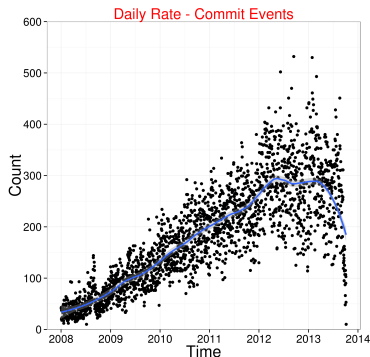
Time	Follower	Followee
2008-02-20 06:56:25	11026	54258
2008-02-20 10:42:42	23291	76482
...	...	...



# Code commits

411278 commits from developers to repositories:

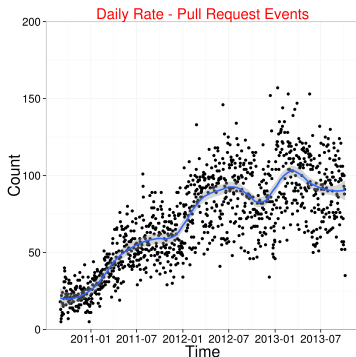
Time	User ID	Repo ID
2008-03-31 23:59:42	160043	76945
2008-04-01 00:00:26	12438	9
...	...	...



# Pull requests

Some projects require developers submit pull requests first before code from their cloned repositories can be committed to these main repositories.

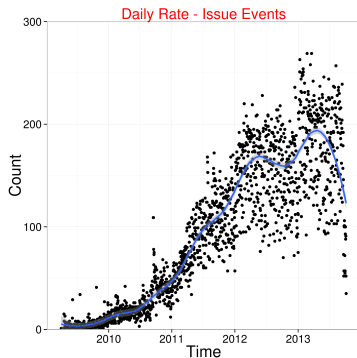
There are 78952 pull request events.



# Issues

150361 issues have been reported on these 90 projects:

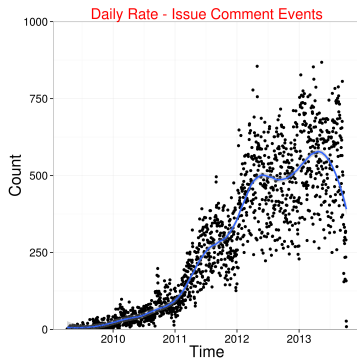
Time	Issue ID	Repo ID
2009-04-01 13:24:23	147349	78852
2009-04-05 15:25:03	109731	79166
...	...	...



# Issue comments

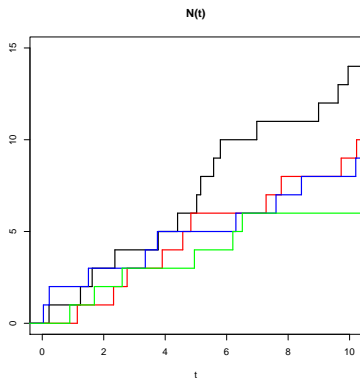
428512 issue comments have been exchanged:

Time	User ID	Issue ID
2009-04-05 15:26:20	169822	109731
2009-04-06 20:31:37	16170	109731
...	...	...



# Relational event models: counting processes

- ▶ The counting process  $N_{ij}(t)$  = cumulative number of events from node  $i$  to node  $j$  by time  $t$ .
- ▶ Combining the  $N_{ij}(t)$  gives a **multivariate** counting process  $\mathbf{N}(t) = \{N_{ij}(t), 1 \leq i \leq n, 1 \leq j \leq m\}$ .
- ▶ We are modelling the matrix  $\mathbf{N}(t)$  whose state space is the set of non-negative and non-decreasing matrices.



$\mathbf{N}(t)$

	1	2	3	...	$m$
1	2	0	0	...	0
2	0	1	7	...	1
3	7	5	0	...	1
...	1	0	5	...	6
$n$	0	3	4	...	1

# Relational event models: conditional intensity functions

The Cox PH function (Cox, 1972) for the directed edge  $(i, j)$ :

$$\lambda_{ij}(t|\mathbf{H}_{t-}) = R_{ij}(t)\alpha_0(t) \exp [\boldsymbol{\beta}^\top \mathbf{s}(t, i, j)],$$

where

- ▶  $\mathbf{H}_{t-}$  is the network history up to but not including time  $t$ ,
- ▶  $R_{ij}(t)$  is the “at-risk” indicator (i.e. nodes  $i$  and  $j$  were created by time  $t$ ),
- ▶  $\alpha_0(t)$  is the nuisance baseline intensity,
- ▶  $\mathbf{s}(t, i, j) = (s_{ij,1}(t), \dots, s_{ij,p}(t))$  is a  $p$ -vector of covariates or statistics for directed edge  $(i, j)$  constructed based on  $\mathbf{H}_{t-}$ ,
- ▶  $\boldsymbol{\beta}$  is the vector of coefficients to estimate.

**The set of network statistics  $\mathbf{s}(t, i, j)$  is where the cross dependencies among counting processes is defined.**

## Relational event models: other events

Different multivariate count processes can be placed on different sets of nodes to model different event types:

- ▶ Binary counting processes between developers for follow events
- ▶ Counting processes between developers and projects for commit events

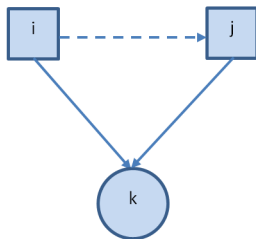
Separate conditional intensity functions and network statistics need to be defined for these types of counting processes.



## Follow events: homophily effect

Developers who commit code to the same repositories are more likely to follow each other.

$$s(t, i, j) = \sum_k \mathbb{I}[N_{ik}(t) > 0] \times \mathbb{I}[N_{jk}(t) > 0]$$



## Follow events: reciprocity effect

Developers tend to reciprocate their following relationship.

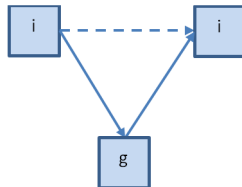
$$s(t, i, j) = \mathbb{I}[F_{ji}(t) > 0]$$



## Follow events: transitivity effect

Developers tend to follow followees of their followees.

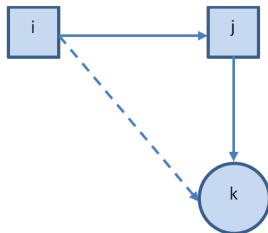
$$s(t, i, j) = \sum_{g \neq i, j} \mathbb{I}[F_{ig}(t) > 0] \times \mathbb{I}[F_{gj}(t) > 0]$$



## Commit events: influence effect

Developers tend to commit code to repositories that their followed coders or *followers* also contribute to.

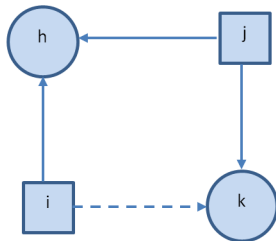
$$s(t, i, k) = \sum_j \mathbb{I}[F_{ij}(t) > 0] \times \mathbb{I}[N_{jk}(t) > 0]$$



## Commit events: four cycle effect

Developers engaging the same issues in the past are more likely to collaborate on solving new issues?

$$s(t, i, k) = \sum_{h \neq k: N_{ih}(t) > 0} \left( \sum_{j \neq i: N_{jk}(t) > 0} \mathbb{I}[N_{jh}(t) > 0] \right).$$



# Estimation

There are 669,024,090 possible following edges between developers to model.

There are 2,327,940 possible commit edges between developers and projects to model.

The estimation is based on 79,365 follow events and 376,799 commit events observed from the beginning of April 2008 until the beginning of October 2013.

For each event, a sample of 10 at-risk edges is sampled for the current risk set to reduce the computation time.

# Main results

	Coefficient	Standard Error	P-value
Homophily	0.10340	0.01047	<.0001
Reciprocity	0.46274	0.00479	<.0001
Transitivity	1.16974	0.00932	<.0001
Repeated collaboration	0.04971	0.00567	<.0001
Social influence	0.83551	0.01088	<.0001

## Follow event model: other effects

	Coefficient	Standard Error	P-value
Sender followees	0.54630	0.00531	<.0001
Sender commits	-0.47270	0.01017	<.0001
Sender pull requests	-0.82037	0.07437	<.0001
Sender issue comments	-0.40747	0.01602	<.0001
Receiver followers	0.72080	0.00644	<.0001
Receiver commits	-0.11979	0.00815	<.0001
Receiver pull requests	-0.89690	0.08177	<.0001
Receiver issue comments	-0.18058	0.01350	<.0001
Cyclic closure	0.08352	0.00557	<.0001
Sharing issue comments	-0.60915	2.33441	0.7941
Sharing pull requests	0.00379	0.03043	0.9008
Follow assortativity	-0.25615	0.00641	<.0001
Commit assortativity	-0.30439	0.01832	<.0001
Pull request assortativity	-10.81903	0.92280	<.0001
Issue comment assortativity	-0.81427	0.05102	<.0001



## Commit event model: other effects

	Coefficient	Standard Error	P-value
Developer followees	-0.67158	0.01449	<.0001
Developer followers	0.02204	0.00940	0.0191
Developer issue comments	0.12101	0.01472	<.0001
Developer pull requests	0.50125	0.01283	<.0001
Developer recency	-0.17668	0.01168	<.0001
Developer engagement	0.39975	0.01278	<.0001
Developer collaborators	-0.40163	0.00935	<.0001
Project recency	-0.01796	0.00909	0.0482
Project popularity	0.86551	0.01268	<.0001
Project shared developers	-0.44144	0.01349	<.0001
Commit repetition	44.12857	0.34189	<.0001
Attention spread	-0.02502	0.00346	<.0001

# Summary Results

- ▶ **Homophily (+)**
  - Shared project developers tend to follow each other
- ▶ **Reciprocity (+)**
  - Developers tend to reciprocate their relationships
- ▶ **Transitivity (+)**
  - Developers tend to follow followees of their followees
- ▶ **Influence (+)**
  - Developers tend to work on the same projects with their followees
- ▶ **4-cycles ( $D1 \rightarrow I1 \leftarrow D2 \Rightarrow$  bipartite closure)**
  - Tendency towards repeated collaboration





# Acknowledgement

This research is supported by:

- ▶ Swiss National Science Foundation, FNS No. 105514\_133273.
- ▶ Australia Research Council Discovery Project DP120102902.

Thank You!

# References

-  Cox, D. R. (1972), Regression models and life-tables, *Journal of the Royal Statistical Society, Series B*, 34, 187–220.
-  Andersen, P., and Gill, R. (1982), Cox's Regression Model for Counting Processes: A Large Sample Study, *The Annals of Statistics*, 10, 1100–1120.
-  Borgan, O., Goldstein, L., and Langholz, B. (1995), Methods for the Analysis of Sampled Cohort Data in the Cox Proportional Hazards Model, *The Annals of Statistics*, 23, 1749–1778.
-  Butts, C. (2008), A relational event framework for social action, *Sociological Methodology*, 38, 155–200.