

Open Borders? Immigration in Open Source projects

Christian Bird, Alex Gourley, Prem Devanbu

Anand Swaminathan, Greta Hsu

Graduate School of Business

背景

- FLOSSプロジェクトはほとんどがボランティア活動によって支えられており管理者が必要
- その管理者になるためには現在の管理者とコミュニケーションをとり管理者の信用を得ることが重要である
- 今回はその管理者になるまでになにが必要なのかを調べていく

関連研究

- FLOSSの開発者を見つけるにあたって
- 三つの大きな要素
 - 社会的地位(コミュニティでの地位)
 - 技術レベル
 - 上の二つは時間とともに上昇していく
- 技術的な関与(パッチ投稿など)
 - 維持するのは難しく時間と共に減っていく

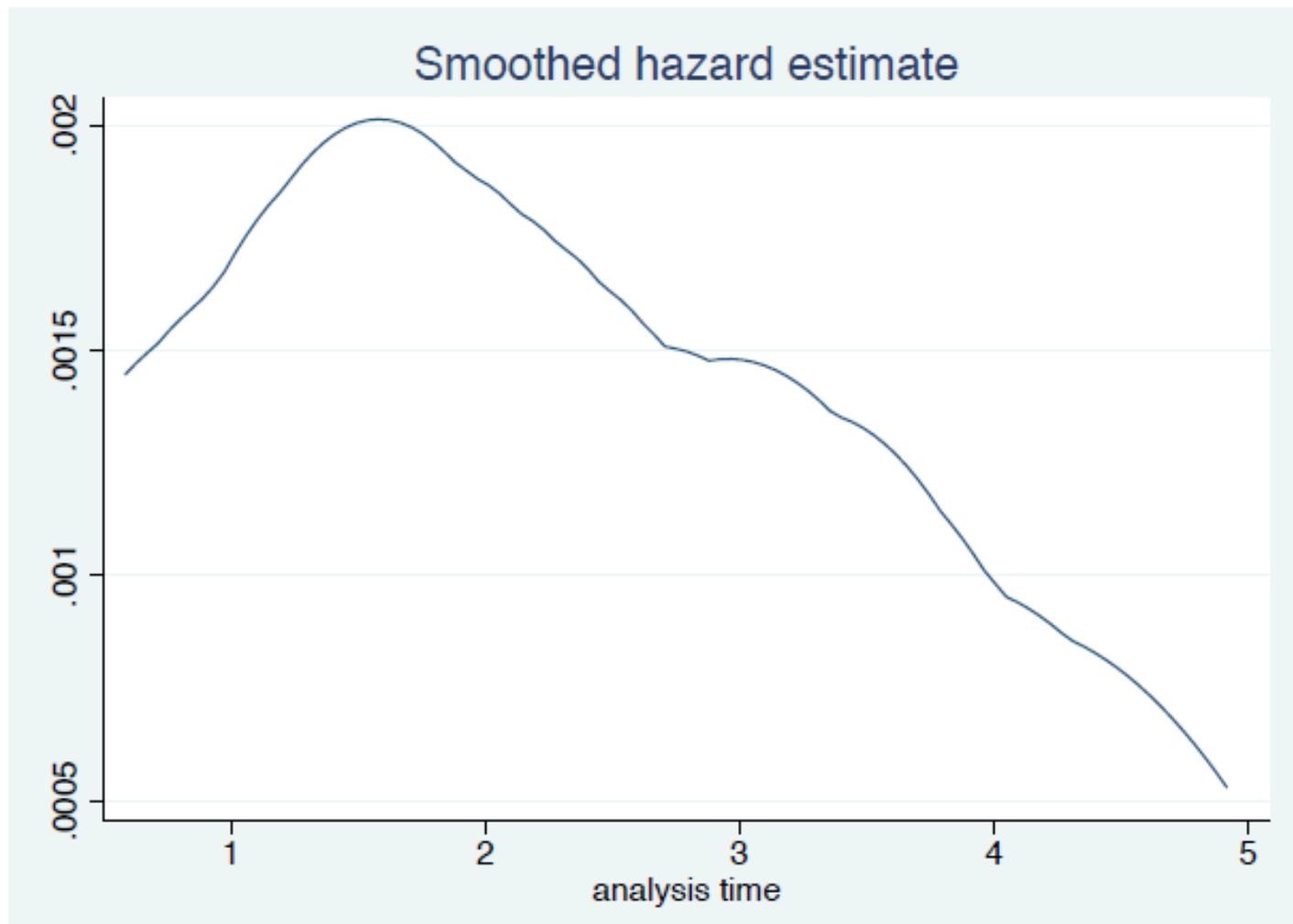
仮説

- 開発者になる可能性はある時期までは上昇しつづけ、その後に減少していくのではないか
- 技術的アプローチ（パッチ投稿など）の数は開発者になる可能性が上がるのではないか
- その人の社会的地位は開発者の可能性に影響を与えているのではないか

アプローチ1

- まず本当に開発者の確率がある時期まで増加していきその後に減っていくのかを調べるためにハザード比をつかって検証する
- ハザード比
 - あるものが一定時間後に予測しているものになる確率を表した図
- 今回は年単位で開発者になっている確率を調べる
- 対象プロジェクト Phostgres

結果：ハザード比



アプローチ2

- ハザード比で見た結果確かにある時期まで開発者になる確率は上がっていきそしてその後低下していていることが理解できる
- 次にこの上昇した原因を突き止めるためにハザード比分析を行っていく
- 対象プロジェクトApache Phostgres Python
- 理由 成功した大きなFLOSSプロジェクト

ハザード比分析

- ハザード比分析は時間依存の確率をモデル化
(病気になってからのいつまで生き残ったなど)
- その現象がいつどんな「要因」によって発生し、どの程度の期間起こっていたのか調べることができるもの

ハザード比分析について

- まず時間 t の間隔である単位 Δt の間になにか重要な「要因」が起こりその間に終了していても考慮しないこと
- ハザード比の関数式 $\lambda(t)$

$$\lambda(t) = \lim_{\delta t \rightarrow 0} \frac{P(t \leq T < t + \delta t \mid T > t)}{\delta t}$$

ハザード比分析について

- 「要因」がまだ起こっていない場合も考慮する
- 「要因」が時間 X で発生すると仮定し、 X が t より後と考えた時の式

$$P(X > t) = \exp\left(-\int_0^t \lambda(s) ds\right)$$

- 「要因」が定数の場合

$$P(X > t) = \exp(-\lambda_c t)$$

ハザード比分析について

- 「要因」が複数ある場合

$$\lambda(t) = f(t) * g(\mathbf{x}) \quad g(\mathbf{x}) = \exp(\mathbf{b} \cdot \mathbf{x})$$

- 複数のものが要因で、それによって単調でない依存期間が起こるとすると区分向上指数ハザード比モデルが必要になる

$$f(t) = \exp(\overset{\text{定数}}{\alpha_{tp_k}}) \text{ for } t \in tp_k, \text{ where}$$
$$tp_k = \underbrace{(c_{k-1}, c_k]}_{\text{区間}}$$

調査要因

- Nは記録数(総数) cは候補者数
iは最終的に開発者になった数
- パッチ投稿したかどうか(patch)
- Indegreeの全体の割合(indegree)
- パッチ投稿が反映された確率(success_pct)
- 調べる月までにメッセージを送った総数(sent_cum)
- 一つのコミュニティ内に何人開発者がいる(dev_cum)
- プロジェクト期間中の観察されていない時間を操作するための変数で単純にカレンダー一年数(time_trend)

Variable	Mean	Std. Dev	Min	Max
(Postgres) $n = 169,118$ $c = 3,283$ $i = 19$				
patch	.04	.20	0	1
norm_indegree	.0024	.0065	0	.18
success_pct	1.27	9.78	0	100
devs_cum	13.92	4.14	1	20
sent_cum	13.12	63.55	1	2370
(Apache) $n = 50,170$ $c = 1,445$ $i = 30$				
patch	.10	.30	0	1
indegree	.0026	.0049	0	.087
success_pct	1.13	8.98	0	100
sent_cum	7.34	18.18	1	273
devs_cum	46.24	9.76	3	57
(Python) $n = 45,216$ $c = 1,320$ $i = 62$				
patch	.04	.20	0	1
norm_indegree	.0098	.032	0	.70
success_pct	.62	6.98	0	100
devs_cum	58.69	14.21	0	79
sent_cum	13.87	54.44	1	1648

Table 1. Univariate Statistics of the predictors

Variable	Apache Model			Postgres Model			Python Model		
	Coef.	z	P > Z	Coef.	z	P > Z	Coef.	z	P > Z
tp1 (0-6 months)	-6.69	-2.97	0.003	-20.63	-10.62	0.000	1.10	0.67	0.5
tp2 (6-12)	-6.09	-2.66	0.008	-5.20	-2.32	0.021	1.18	0.69	0.492
tp3 (12-24)	-6.48	-2.7	0.007	-6.51	-2.9	0.004	0.098	0.06	0.955
tp4 (24-36)	-7.98	-2.93	0.003	-5.77	-3.00	0.003	-0.23	-0.13	0.9
tp5 (36-48)	-6.79	-2.53	0.011	-5.98	-2.88	0.004	0.10	0.06	0.96
tp6 (48-60)	-8.16	-2.60	0.009	-8.14	-2.88	0.004	0.91	0.47	0.64
indegree*	61.21	2.78	0.005	72.2	5.46	0.000	5.49	2.92	0.004
patch*	2.71	5.09	0.000	0.98	1.42	0.154	2.09	5.95	0.000
acceptance_rate*	0.15	2.03	0.043	0.0039	0.45	0.656	0.0062	0.10	0.918
sent_cum*	0.014	4.20	0.000	.00158	1.16	0.246	0.0041	1.86	0.062
devs_cum	0.14	1.84	0.066	1.09	2.17	0.03	0.14	2.81	0.005
time_trend	-0.91	-1.90	0.057	-2.06	-1.93	0.054	-1.46	-3.19	0.001

結果

- Pach
 - Apacheのハザード比15(1500%確率が上昇する)
 - Physonのハザード比8(800%確率が上昇する)
- Acceptance_rate(success_rateの言い替え)
 - Apacheのハザード比3.8(380%確率が上昇する)
- Sent_cum
 - Apacheのハザード比2.5(250%確率が上昇する)
- Indegree
 - Apacheのハザード比1.34(34%確率が上昇する)
 - Postgreのハザード比1.60(60%確率が上昇する)
 - Pythonのハザード比1.19(19%確率が上昇する)

結論

- 単調ではない依存について
 - 三つともハザード比が増加し、一定期間後に減少する傾向があったがPythonのみ統計的に有意ではなかった
- パッチ投稿の影響
 - 三つとも影響はあったがPostgresは統計的に有意ではなかった理由としてはPostgresは非開発者のうちの50%がパッチ投稿をしており他の二つよりもパッチ投稿が一般的だからだろうと
- 社会的地位について
 - 影響は穏やかではあるが三つともに影響があった

まとめ

- 単調ではない変化はApacheとPostgresに顕著に現れた
- パッチ投稿による影響はApacheとPythonに強い影響を与えた
- 社会的地位(コミュニティの地位)は穏やかではあるが三つともに影響を与えた

所感

- 変数のsucces_rateをacceptance_rateと途中で言い換えなかった
- 勘違いかもしれないが計算ミスをしていそうな場所があり、理解できなかった
- ミスではないかと感じた箇所について
 - ApachのSent_cumの値はおそらくeの指数計算でハザード比分析で求めた係数と分散をかけて計算しているのだがこの部分はApachの係数とPhostgresの分散を計算している