

Developer Initiation and Social Interactions in OSS: A Case Study of the Apache Software Foundation

Mohammad Gharehyazie · Daryl Posnett ·
Bogdan Vasilescu · Vladimir Filkov



概要

- コミッター予測を社会的貢献（メール）と技術的貢献（パッチ投稿）に基づいて行った
- 最初の1か月のデータでも高い精度で予測できた

背景

- OSSのコミュニティ構造はオニオンモデル等で表現され信頼される開発者がコア開発者になる
 - コア開発者 = コミッター
- 開発者が信頼を得るためには、社会的・技術的な貢献をプロジェクトに対して行う必要がある
- どれくらいの貢献をすればコミッターになれるかはわからない

目的とアプローチ

- 目的
 - OSSプロジェクトでの役割の変化を再検証すること
- アプローチ
 - 技術的・社会的貢献を計測する
 - 技術的貢献：パッチ投稿, バグ修正
 - 社会的貢献：メーリングリストでのコミュニケーション
 - 新規開発者の活動初期の技術的・社会的貢献からコミッターとなるかどうかの予測を行う

リサーチクエスチョン

- RQ1：技術的・社会的貢献メトリクスを用いてコミッターになる開発者を予測できるのか？
- RQ2：活動初期の期間を変えた場合どれくらいの精度で予測することができるのか？
- RQ3：コミッタ昇格直前の開発者間のコミュニケーションはどうなっているのか？
- RQ4：プロジェクトの後半の方がコミッターになりにくいのか？

データセット

- 対象プロジェクト

Project	#Users	#Committers	#Mails	#Patches	Start	End
Ant	1416	44	17300	1482	2000-01	2012-03
Axis2_c	600	24	11152	754	2004-01	2012-03
Log4j	539	18	3811	166	2000-12	2012-03
Lucene	2155	41	43922	5576	2001-09	2012-02
Pluto	266	24	3017	259	2003-10	2011-09
Solr	840	19	14411	4090	2006-01	2010-04

- データソース
 - メーリングリスト
 - バージョン管理システム
 - 不具合管理システム

データ整形：名寄せ

- Birdの“Mining Email Social Networks”でのやり方をまねて名寄せを行った
- 手順
 - Dr, Mr, Jrなどを削除
 - レーベンシュタイン距離を計算
 - 完全一致した名前をマージ
 - レーベンシュタイン距離が0.93以上のものをマージ

ソーシャルネットワークの構築

- Birdの“Mining Email Social Networks”でのやり方をまねてネットワークを構築
- ある開発者の投稿に対して返信をしていれば、リンクをはるようになる
- developer list (dev@ant.apache.org)をみると会員のコミッターシップをみることができる

パッチ投稿の取得

- パッチの投稿方法にはメーリングリストでの投稿, および不具合管理システムでの投稿がある
 - アタッチメントファイルのファイル名
 - *.diff , *.patch , patch.txt , patch.zip ... etc
 - メール, BTS上でのコメント
 - +++ や --- がメッセージに含まれる

	Maling Lists		Bugzilla	Jira	
	Inline	Attachments	Attachments	Inline	Attachments
Ant	413	976	93	0	0
Axis2_c	63	200	0	27	464
Log4j	87	57	19	0	3
Lucene	141	107	0	88	5240
Pluto	15	26	0	8	210
Solr	30	4	0	48	4008

コードリポジトリのマイニング

- Gitからコミッターの情報を取得する
 - 初めてコミットした日がコミッター昇格日
- コミッターをフィルタリング
 - フィルタリング方法
 - コミッター昇格前に3か月以上活動している
 - メーリングリストとのひもづけができています

Project	#Committers	#Filtered Committers	#Remaining Committers
Ant	44	13	31
Axis2_c	24	3	21
Log4j	18	8	10
Lucene	41	9	32
Pluto	24	10	14
Solr	19	5	14

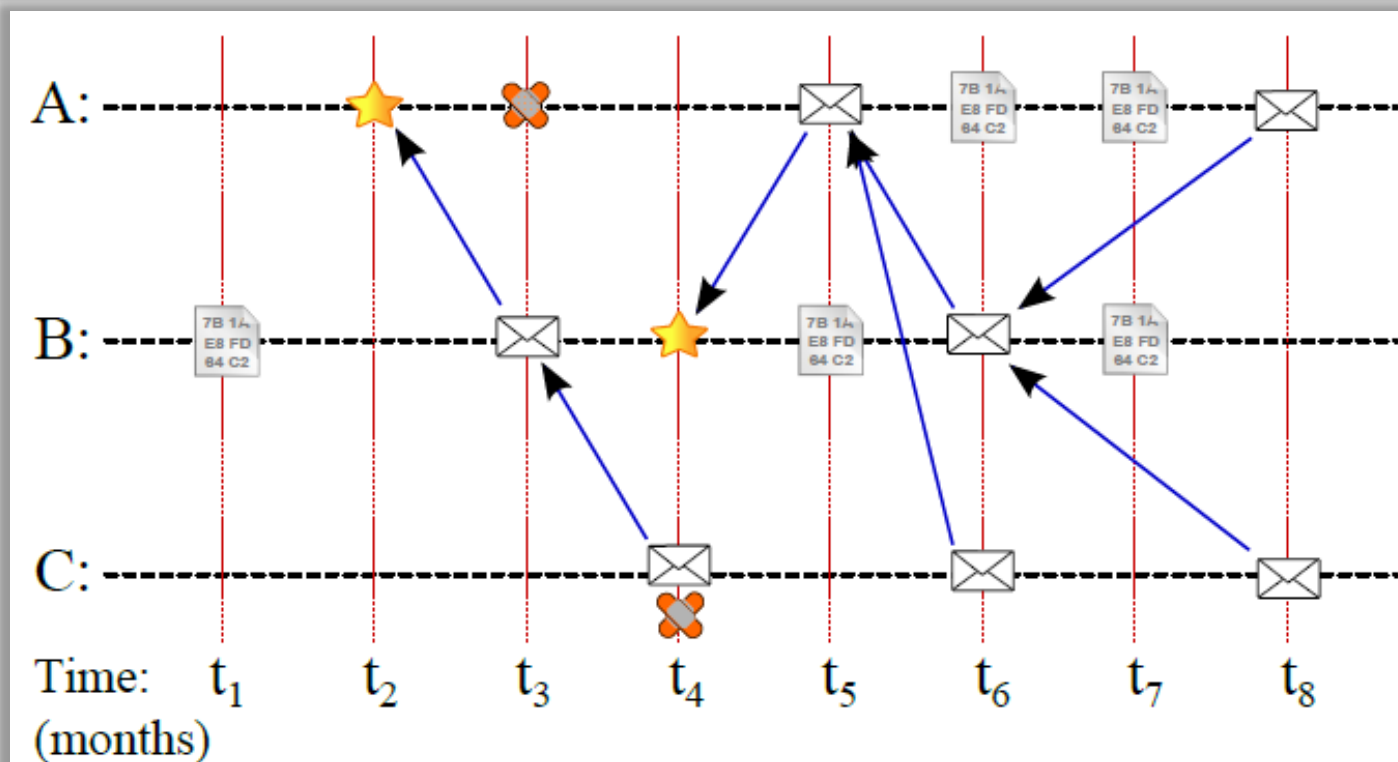
活動メトリクスの計測(1)

メトリクス名	説明
<i>Initiated</i>	コミッター昇格以前にメーリングリストでの活動のあるコミッターを識別するためのフラグ(boolean)
<i># of Patches</i>	プロジェクトに参加し始めた最初のkか月の間に投稿したパッチ数
<i># of Messages</i>	メーリングリストでの議論のメールを投稿した数と返信が返ってきた数の合計
<i># of Threads</i>	投稿したスレッド数
<i>Neighbors</i>	ネットワークを構築した際の隣のノード数（やりとりをした開発者の数）
<i>Neighbor Committers</i>	ネットワークを構築した際の隣のノードのコミッター数（やりとりをしたコミッターの数）
<i>Project age</i>	初めてメーリングリストに参加した日から初めてネットワークに登場した日（リプライを初めてもらった日）の日数

活動メトリクスの計測(2)

Table 4. Social and technical activity measures for sample dataset in Figure 3 assuming $k = 3$.

ID	Initiated	Num. Patches	Num. Messages	Num. Threads	Neighbors	Neighbor Committers	Project Age
A	TRUE	1	5	1	2	1	90
B	FALSE	0	6	1	2	1	90
C	FALSE	1	3	0	2	2	120

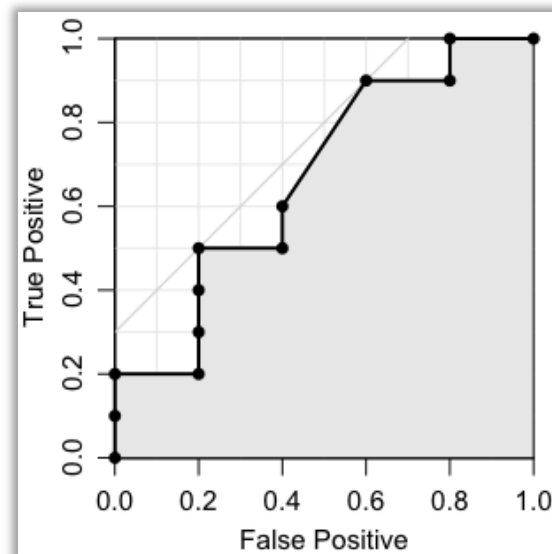


予測モデル構築

- ロジスティック回帰分析によって予測を行う
 - 目的変数 : *Initiated*
 - 説明変数 : その他
- それぞれの予測モデルにおいてZ検定を行う
- すべてのメトリクスの値をLogをとり, 0.5プラスすることで異分散を和らげる
- VIFの値を調べ, 多重共線性のチェックを行う

評価

- プロジェクト間での予測
 - クロスプロジェクトでの予測をしたい
 - しかし, プロジェクト間での「文化」がある
 - 30回のパッチ投稿はAプロジェクトでは多いがBでは少ない
 - そこで予測ではなく相関分析をしp値で判定する
- 評価指標としてAUROCを用いる
 - The Area Under an ROC Curve
 - ROC曲線
 - 判別境界面を変えていったときの True Positive と False Positive の 変遷をプロットした曲線

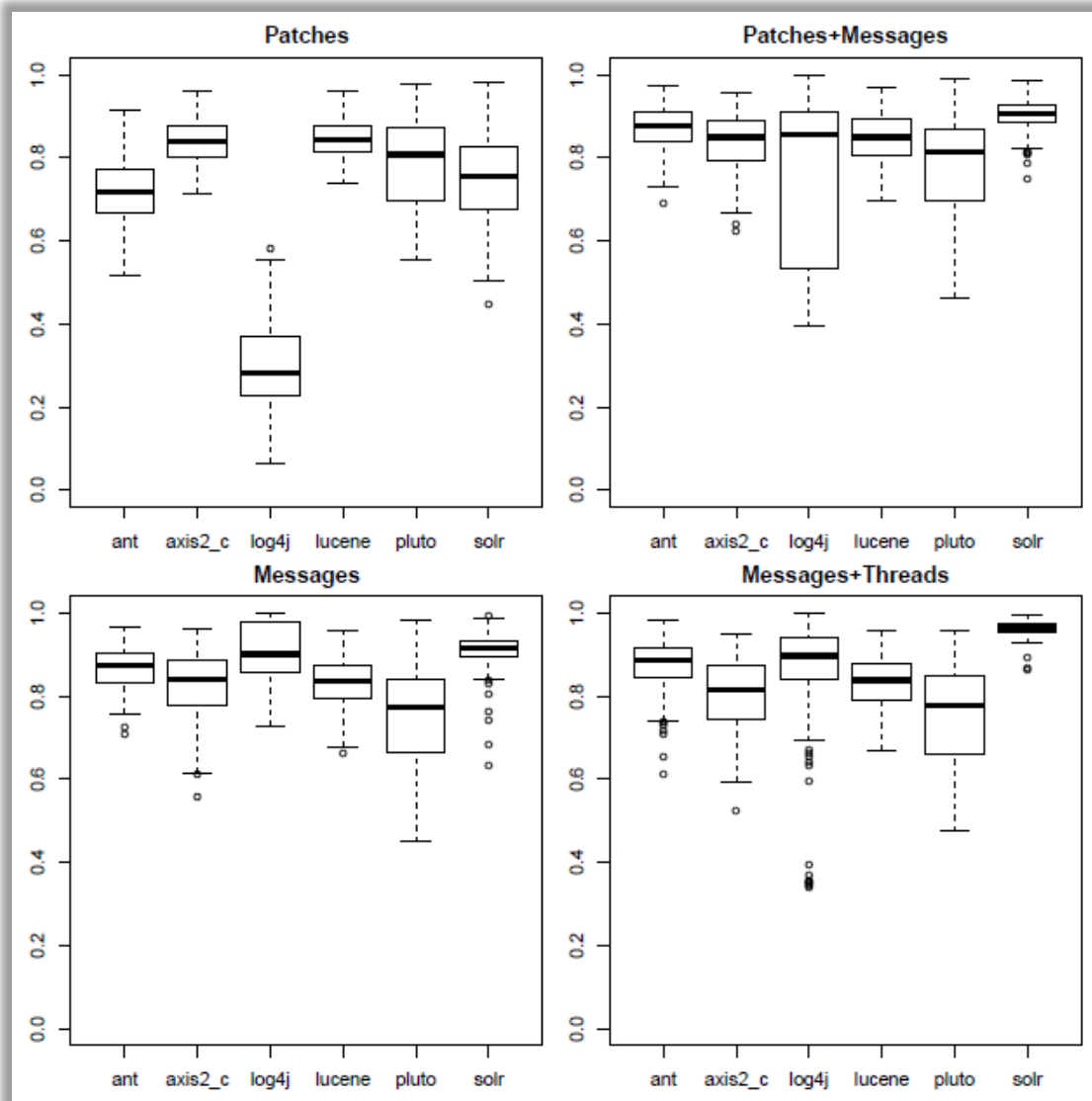


実験手順

- 過学習を防ぐためにホールドアウト検定を行う
 - ホールドアウト検定：集めたデータのうち何割かを評価に使う（この研究では1/3を使う）
 - 2/3のデータで学習し，1/3のデータで評価する
- コミッターと一般開発者の割合をそろえるために層化抽出法を用いる
 - 層化抽出法
 - 母集団をグループに分け各グループからランダムにサンプル
 - コミッターと一般開発者という層に分ける
 - サンプルングを250回繰り返す

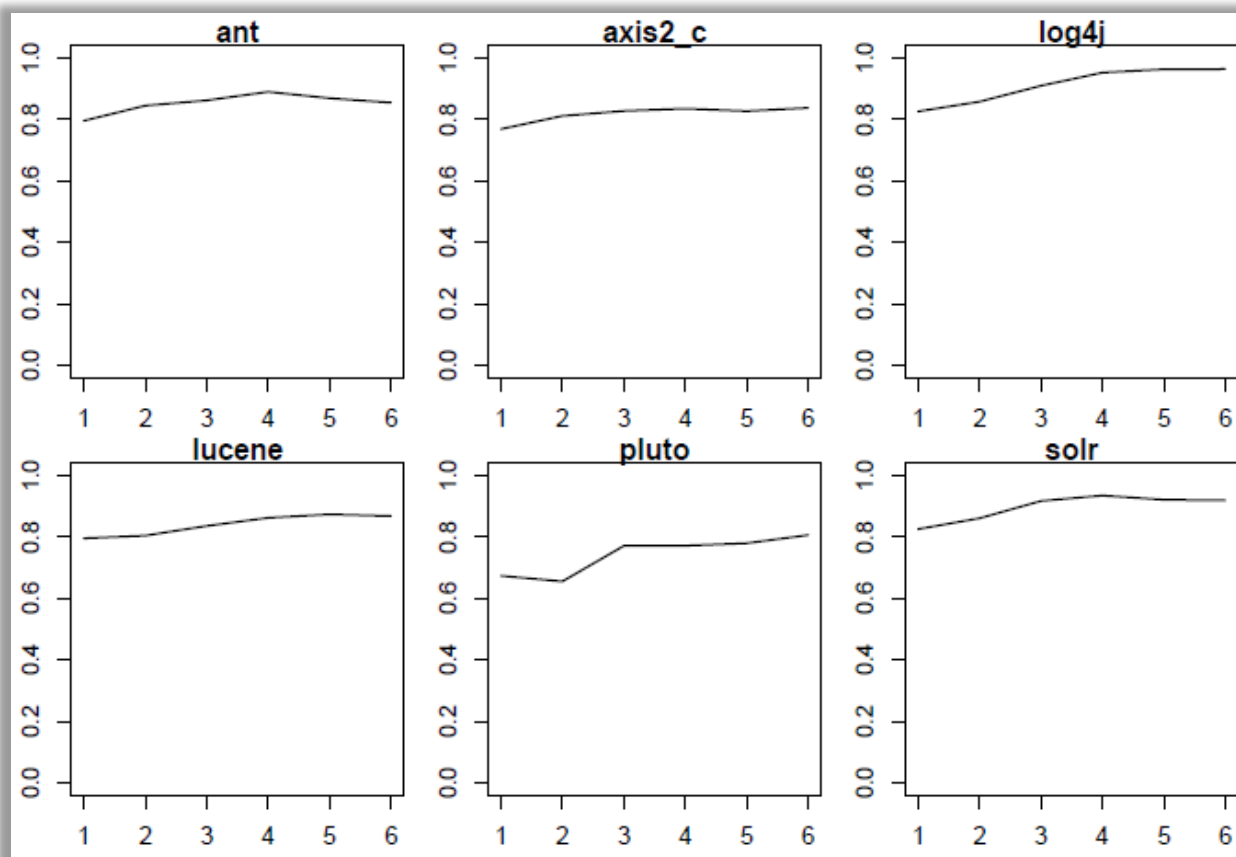
結果：RQ1

- RQ1：技術的・社会的貢献メトリクスを用いてコミッターになる開発者を予測できるのか？
 - 社会的貢献メトリクスを用いたモデルの方が予測精度が高くなった



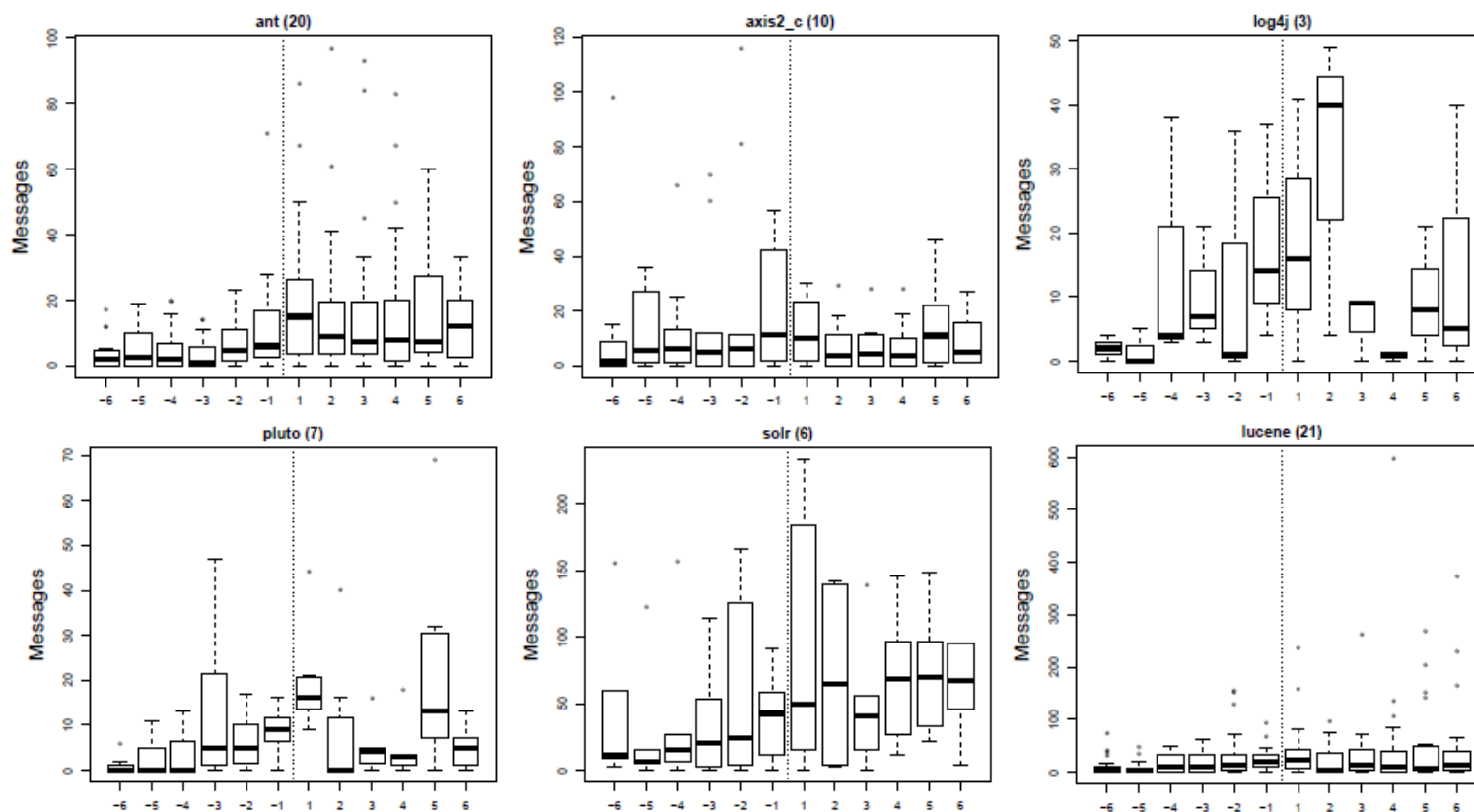
結果：RQ2

- RQ2：活動初期の期間を変えた場合どれくらいの精度で予測することができるのか？



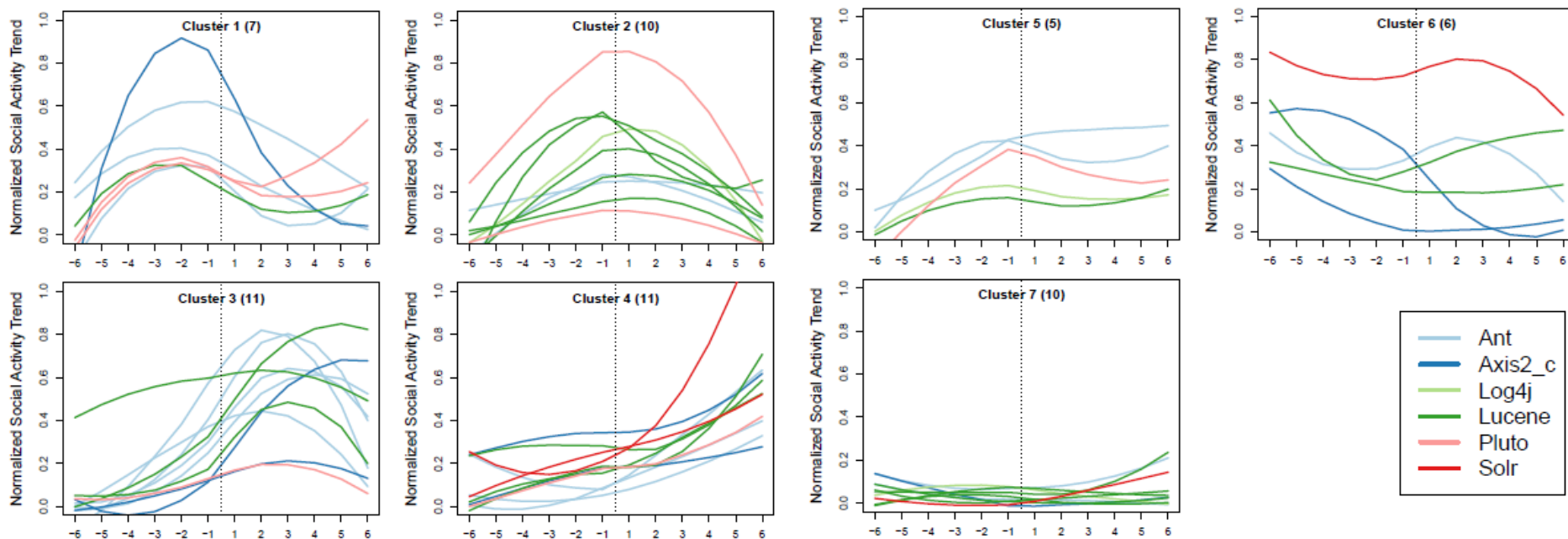
結果：RQ3

- RQ3：コミッター昇格直前の開発者間のコミュニケーションはどうなっているのか？



結果：RQ3

- RQ3：コミッター昇格直前の開発者間のコミュニケーションはどうなっているのか？
 - 基本的に昇格直前にピークを迎えている開発者が多く、昇格後に活動量が低調になる開発者もいる



結果：RQ4

- RQ4：プロジェクトの後半の方がコミッターになりにくいのか？
 - すべてのプロジェクトにおいて, 'Project age'と 'Initiation'の相関係数がマイナスであった
 - IsSecondメトリクスを定義し検証する
 - IsSecond：分析期間の中央値以降か否か

	Ant	Axis2_c	Log4j	Lucene	Pluto	Solr
(Intercept)	-5.5	-4.21	-7.69	-4.74	-2.99	-6.59
Number of patches	0.62	0.59	-0.76	0.62	0.85	-0.2
Number of messages	1.08	0.56	1.89	0.72	0.49	1.17
IsSecond	-0.36	-2.08	-0.9	-2.46	-2.1	-1.34
(Intercept)	-5.76	-4.93	-7.04	-5.42	-3.8	-6.33
Number of messages	1.24	0.82	1.78	0.99	0.88	1.07
IsSecond	-0.57	-1.84	-0.99	-2.67	-2.01	-1.29

- 後半の方がコミッターになりにくいことがわかった

まとめと所感

- まとめ

- 技術的な貢献よりも社会的な貢献の方がコミッターを選出する上で重要視されていることがわかった
- 新規開発者の活動初期（1～3か月）のデータだけでも十分にコミッターになるか否かの予測ができた
- プロジェクトの後半の方がコミッターになりにくいことがわかった

- 所感

- サンプルング方法とかそれでいいん？って感じ
- PrecisionやRecallの値はどうなった？って感じ