

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

Empirical Software Engineering

October 2015, Volume 20, [Issue 5](#), pp 1318–1353

# 背景

- OSSコミュニティを持続させるためには新たな開発者の参入が必要不可欠(特にコア開発者)
- プロジェクトに十分な貢献を行った開発者がコア開発者となる
- どのような活動(社会的？技術的？)によりコア開発者になったか調査&予測

- RQ1：コア開発者の加入についてパッチ活動や社会的コミュニケーションによりどの程度モデル化できるか？
- RQ2：コア開発者の加入についてプロジェクトの初期活動からどの程度正確に予測できるか？
- RQ3：昇格前後のコミュニケーションに違いはあるか？
- RQ4：プロジェクトの後半にコア開発者となるのは容易？困難？

# データセット

- Apache Software Foundationから6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 |

- メーリングリスト, 不具合管理システム, バージョン管理システムからデータを抽出

# データの整形(名寄せ)

- 開発者は別のエイリアスを持つことが多い
  - <John Smith, smith@gmail.com>,  
<Smith, John@smith.com>,  
<John S., J.smith@ucdavis.edu> など
- レーベンシュタイン距離を計算してマージした<sup>[1]</sup>
  - ある文字列Aに対して『1文字の追加・削除・置換』を何回繰り返せば他の文字列Bになるか
  - この時の最小距離のことを指す
    - すうがく→すがく→すがた であればレーベンシュタイン距離「2」

[1]C. Bird, A. Gourley, P. Devanbu, M. Gertz, and A. Swaminathan, "Mining email social networks," in MSR. ACM, 2006, pp. 137-143.

# データの整形(ソーシャルネットワーク)

- 開発者間の関係をエッジで表現
- ある開発者に対して返信を行っていればエッジを作成<sup>[1]</sup>

[1]C. Bird, A. Gourley, P. Devanbu, M. Gertz, and A. Swaminathan, "Mining email social networks," in MSR. ACM, 2006, pp. 137-143.

# データの整形(パッチ投稿)

- メーリングリスト, 不具合管理システムに投稿されたファイル进行分析
- .diff や .patchをパッチとする
- ※patch.txtのようなファイルも存在する
  - コメントも分析
  - "+++"や"---"を含んだファイルをパッチと判定

# データの整形(コア開発者)

- 初コミット日＝コア開発者への昇格日とする
- 少なくとも昇格前に3ヶ月以上の活動が行われている開発者のみにフィルタリングする

| 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                    |

# データセットの整形(集計)

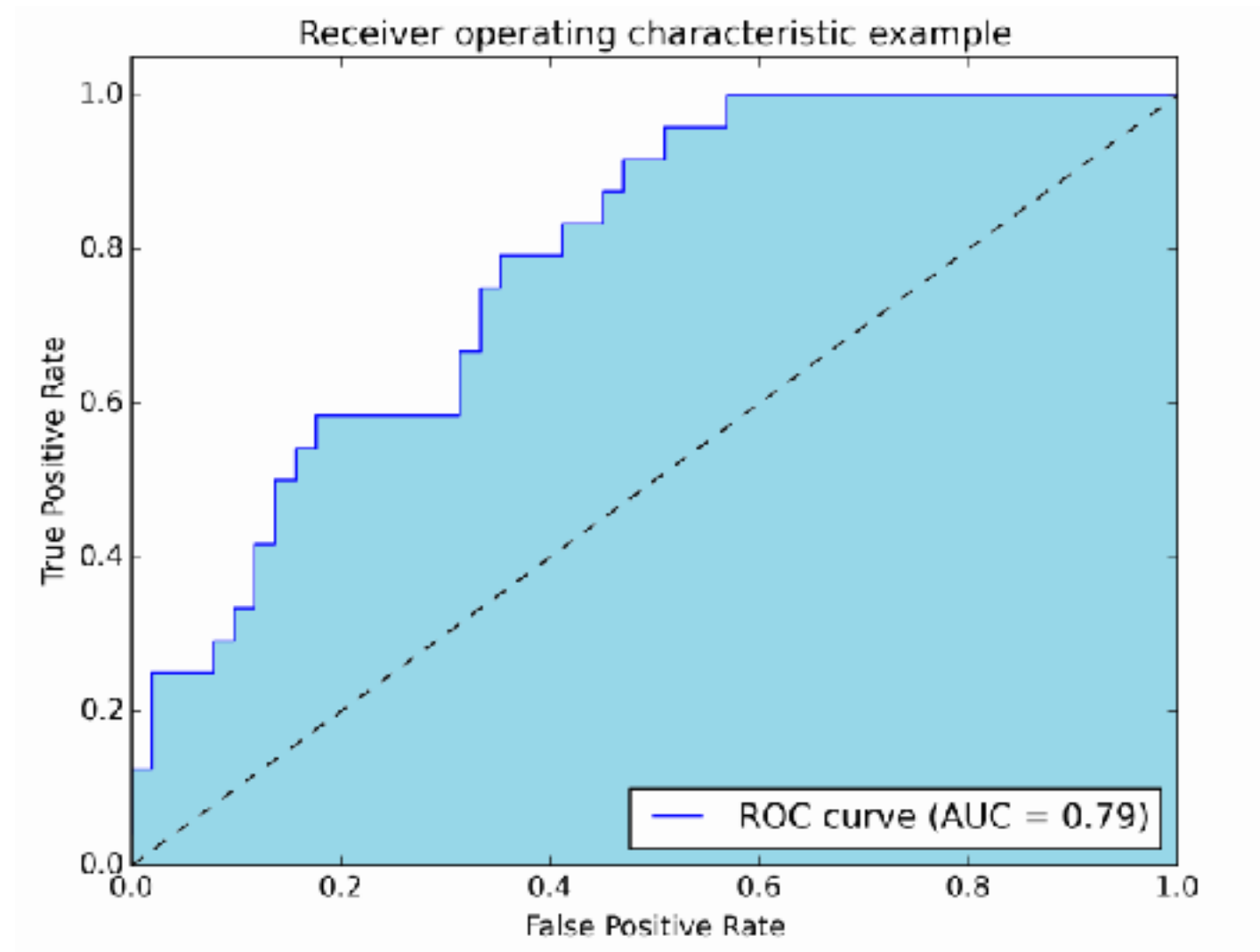
- 各メトリクスを集計する
  - Initiated コア開発者かどうか
- Number of Patches(Messages,Threads)
- Neighbor (Committers)
- Project age

# モデル構築

- ロジスティック回帰分析を行う
- 各メトリクスは+0.5 & Log を取っている
  - +0.5 0のものもLogを取れるようにするため
  - Log 各メトリクスを同一尺度で評価するため
- 多重共線性をチェックしてメトリクスを選択

# 評価

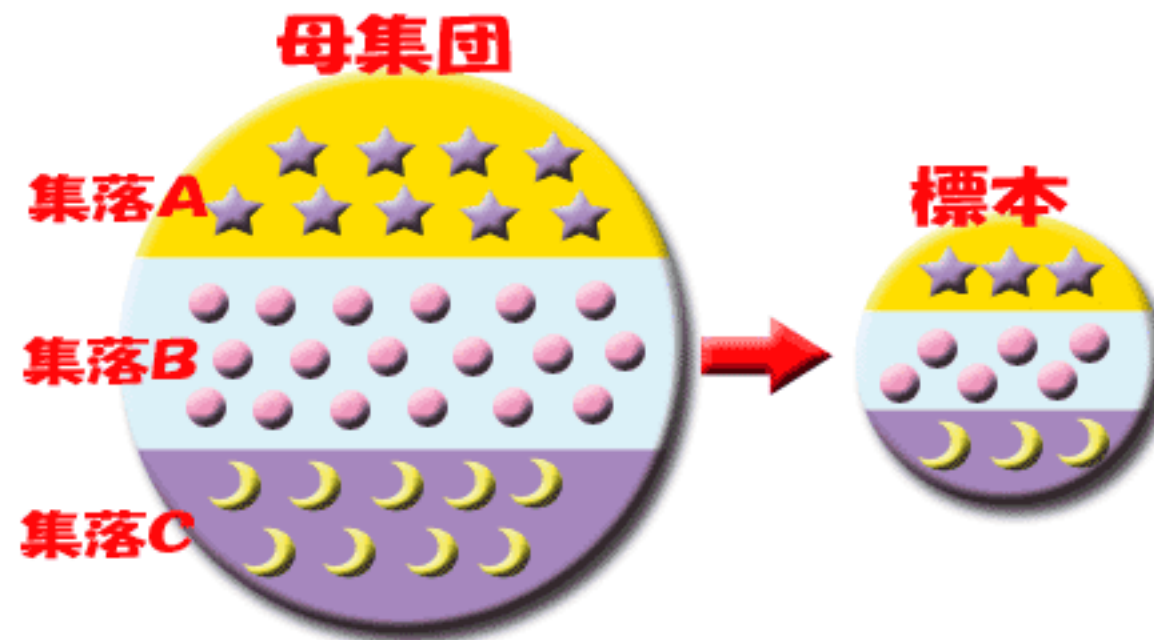
- AUROC面積を使用してモデルの識別能力を評価
- 面積が大きいほど識別能力が高い



# 訓練方法

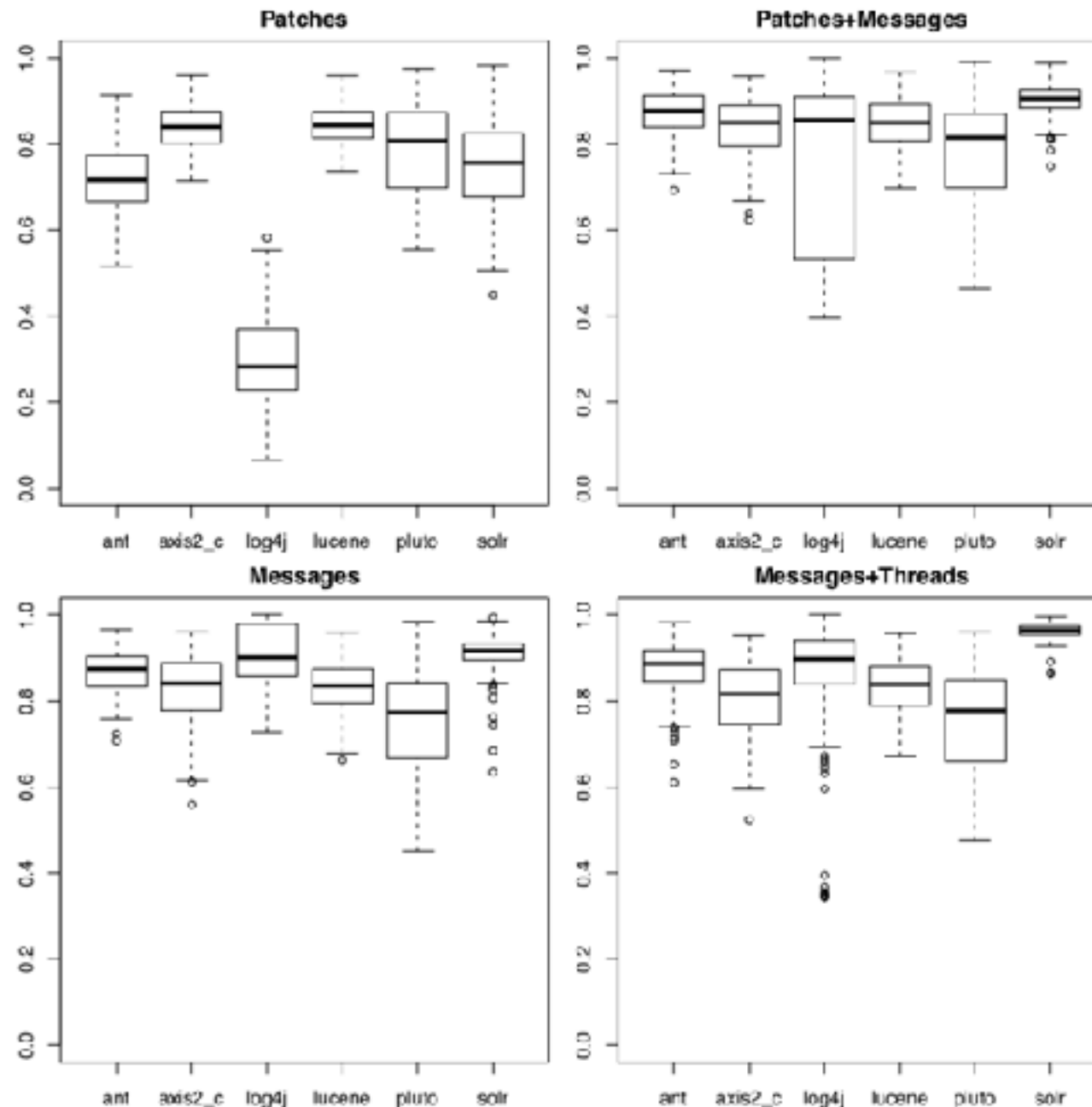
- ホールドアウト検定により過学習を防ぐ
  - 2/3 訓練データとして学習
  - 1/3 テストデータとして評価
- 層状サンプリングにより開発者をバランスよく抽出

## 層化抽出法（比例配分法）



# RQ1について

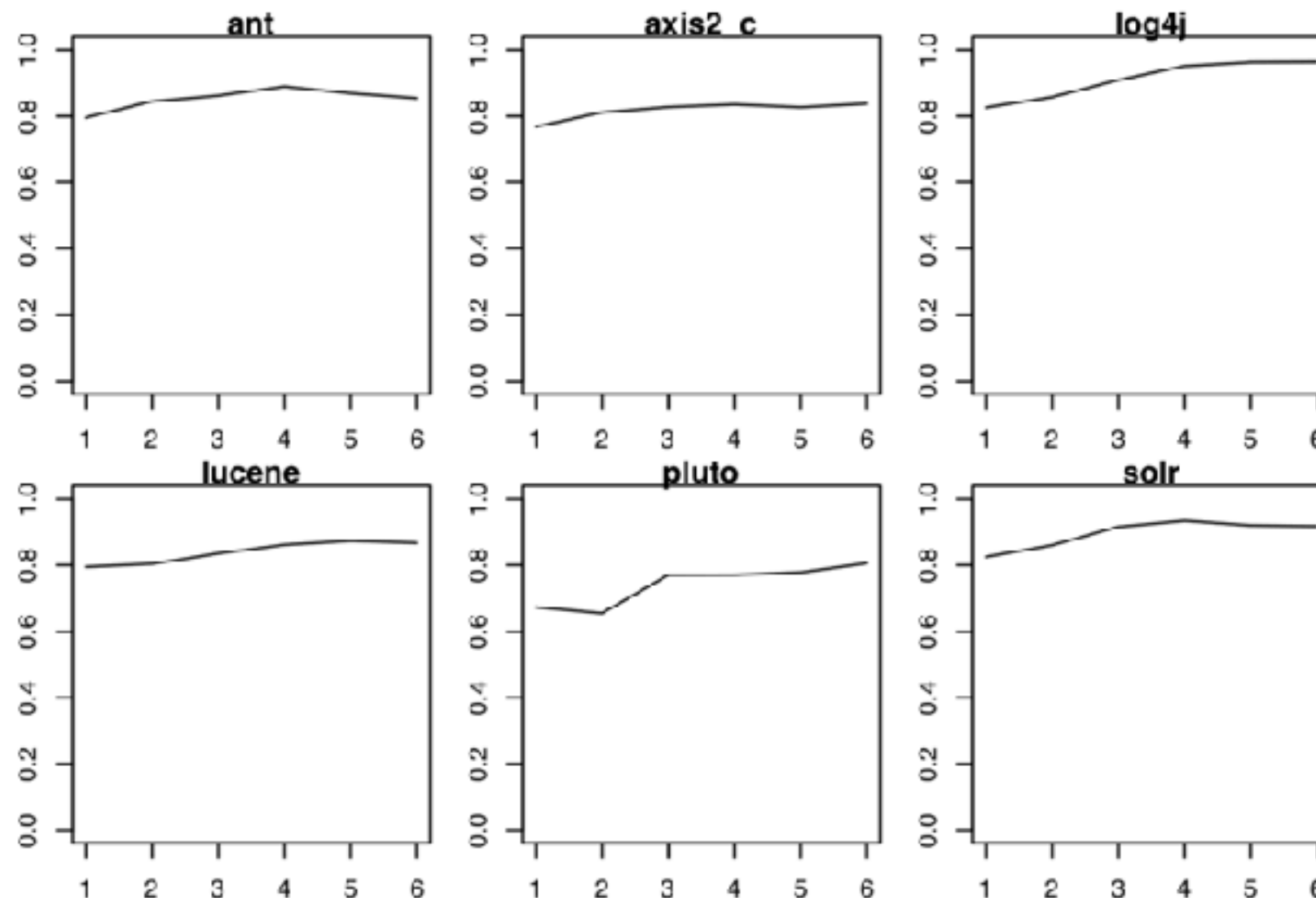
- コア開発者の加入についてパッチ活動や社会的コミュニケーションによりどの程度モデル化できるか？



パッチよりソーシャルの方が精度が高い

# RQ2について

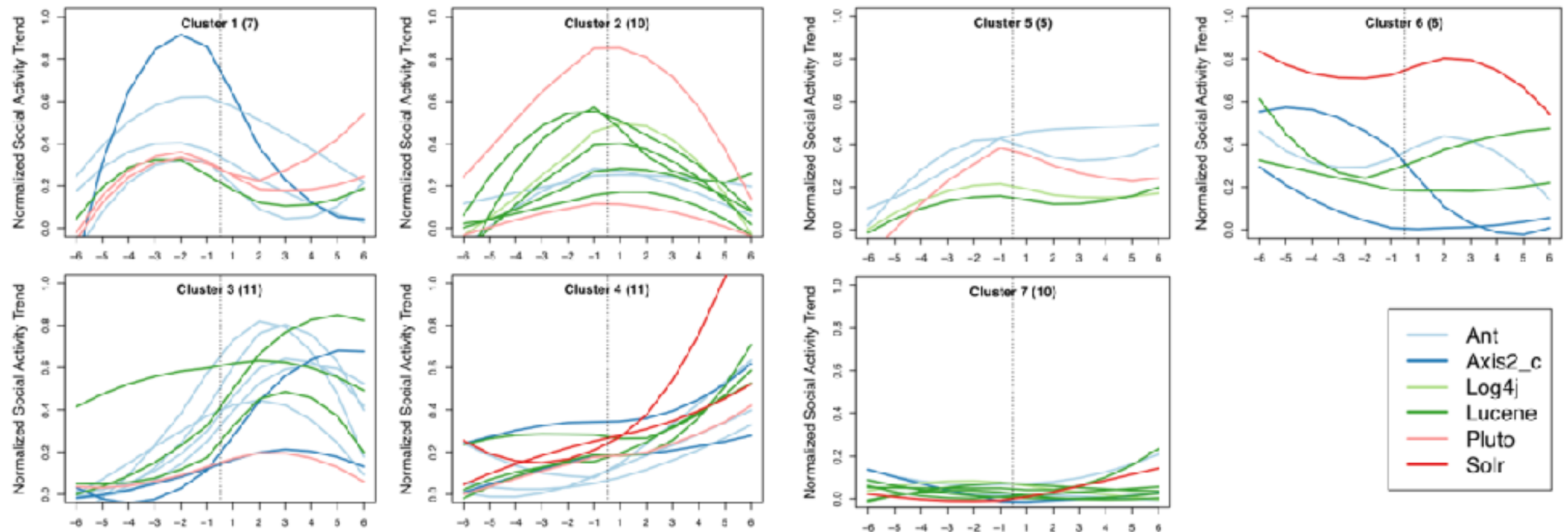
- コア開発者の加入についてプロジェクトの初期活動からどの程度正確に予測できるか？



3ヶ月あれば十分高精度な予測が可能

# RQ3について

- 昇格前後のコミュニケーションに違いはあるか？



開発者ごとに多種多様

# RQ4について

- プロジェクトの後半にコア開発者となるのは容易？困難？

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

- 後半にコア開発者となるためには前半以上の貢献が必要

# まとめ

- 社会的なメトリクスの方が技術的なメトリクスよりも識別性能の高いモデルを作ることができる
- ある程度の予測は1ヶ月で、確かな予測は3ヶ月のデータでコア開発者の予測が行える
- コア開発者昇格後の活動は開発者ごとに異なり多岐に渡る

# 所感

- precision, recallがどうなったか気になる
  - クラスが極端に偏っているのでPR曲線を使うべき？
- パッチのaccept率とかも使えそう
- 関係が増えていたが、どのように増えたのか気になる
  - 実際の会話の内容など