

商業用ゲームにおける MOD 導入による売上の変化に関する研究

PM コース 矢吹研究室 1142003 赤松 佳紀

1. 研究背景

ゲームには MOD と呼ばれる拡張プログラムが存在する。これは、現在販売されているゲームに新たな別の要素を取り入れるものだ。MOD の説明は下記のようになる。

MOD とは Modification の略であり、主にパソコン用ゲームの改造データである (Addon と呼ばれる場合もある)。ユーザーが独自に開発しているために基本的には無料で配布される。この点だけ見れば商業用ゲームにおいて芳しくない結果になると思われがちであるが、MOD によって追加された要素がユーザーに認められ、評価されることによって通常よりも知名度の獲得ができることになるため企業側にとって良い結果になる。

しかし、商業用ゲームにおいて MOD の導入を許可した例は今までに少なく、シムシティシリーズ、グランド・セフト・オートシリーズ等あるが、その中でも Minecraft は MOD 導入の影響によって知名度が急上昇した商業用ゲームである。今回は MOD の導入を許可した商業用ゲームの 1 つの成功事例として Minecraft を挙げることにした。

Minecraft とは Notch (マルクス・ペルソン) とその会社 (Mojang AB) の社員が開発したサンドボックスゲームである。日本語圏での一般的な略称は、『マイクラ』。このゲームの特徴としては、立方体のブロックで世界が構築されており、ブロックを設置したり破壊したりすることができる。さらに、地上には草原、森、砂漠、雪原などのさまざまな気候や地形が存在し、地下には洞窟や廃坑などが広がっており、昼夜の概念も存在するためプレイヤーはその世界で自由に生活することができる。もう一つの特徴として Crafting (工作) があり、ブロックやアイテムを組み合わせることで新しいアイテムを作ることができる高い場所にブロックを設置するには足場を作る必要があったり、上位の発掘ツールを使用してより高度な素材を手に入れたり、耐久値のある道具を使うことで効率よくブロックを破壊できたり、敵の侵入を防ぐ為にアイテムを設置したり武器を作ったり、機械的な物を作ったり実験するなど、ゲーム的な楽しさが追加されているのが Minecraft の大きな特徴と言える。自由度が高い為 MOD の相性はとても良いとされる。

そのため、Minecraft の売上と MOD プロジェクト数の相関を調査することで、MOD の導入がゲームの売上にどのような影響を与えているのか明

らかになるのではないかと考えた。

2. 研究の目的

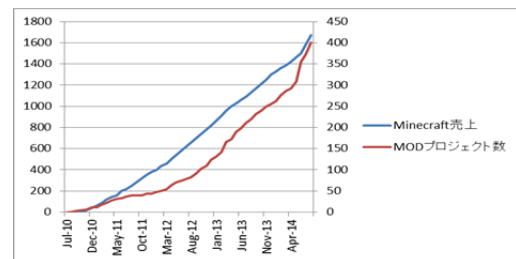
本研究では、Minecraft の売上と GitHub 上の Minecraft MOD プロジェクト数の相関を調査し、MOD の導入が商業用ゲームに与える影響を明らかにする。

3. 研究方法

GitHub 上で行われている Minecraft MOD プロジェクトを月別でグラフにする。さらに、Minecraft の売上を月別でグラフにし、2 種類のデータの相関を調査する。2 種類のグラフが右上がりの場合両方に相関があるという結果になる。また、2 種類のグラフに相関が見られない場合、別の要因があると考えられる為、別の要因を調査する。

4. 調査結果

GitHub API をもとに Minecraft MOD の時系列データを取得し売上と比較できるようグラフにした。



この結果からは MOD が売上に関係しているとは言えない。

5. 考察

今回の調査結果から、なぜ求めていた結果が出なかったのか考察した。その理由として挙げられるのが、調査する前の段階でどのような結果が出れば本研究の目的を達成できるのかを明確にしていなかったということである。しかし、今後このような研究を行う際に自動でデータを収集できるアプリ開発は成功したと言える。

参考文献

- [1] 大塚弘記. WEB+DB PRESS plus シリーズ GitHub 実践入門 Pull Request による開発の変革. 株式会社技術評論社, 2014.
- [2] The GitHub Blog. The Octoverse in 2012. 2012-12-19. <https://github.com/blog/1359-the-octoverse-in-2012>
- [3] Gigazine. 「Microsoft が「Minecraft」開発元を 2680 億円で買収、その経緯とは?」, 2014-09-16. <http://gigazine.net/news/20140916-microsoft-acquired-minecraft/>

OSS 開発プロジェクトにおけるタスク処理過程の定量分析

プロジェクトマネジメントコース 矢吹研究室 1142009 安藤 勇樹

1. 研究背景

ソフトウェア開発のためのホスティングサービスである GitHub では様々なソフトウェアが開発されている。2013 年 12 月には GitHub 上に 1000 万件のリポジトリが作成され、ユーザ数は 400 万人を超えた。数多くのプロジェクトが公開されている GitHub を調査すれば、ソフトウェア開発プロジェクトの分類が可能であると考えられる。

過去に GitHub 上のプロジェクトのチケットを調査し、プロジェクトを分類するという研究があり、プロジェクトの分類が可能であるということが明らかにされていた[2]。しかし、この研究では分類の解釈を人間が主観的に行っており、客観性に欠けているという問題があった。そのため、本研究ではデータマイニング手法を用いて分類を客観的に行う。

GitHub には、リポジトリの人気指標の 1 つに Star が存在する。Star とは、気になるリポジトリをブックマークできる機能である。他者から Star を押された数を Star 数とし、Star 数が多いリポジトリは人気が高いことを示している。本研究では、Star 数を基準にプロジェクトを選択し、調査をする。

本研究では、プロジェクトを分類するためにチケットを調査する。チケットとは、ソフトウェア開発中に発生した作業や変更履歴の内容を登録する作業指示書である。チケットには未完了チケットと完了済チケットの 2 種類が存在する。未完了チケットは作業が完了されていないチケットを示し、完了済チケットは作業が完了されているチケットを示す。チケットによって作業の進捗状況を可視化できるため、進捗管理が容易になる。

このチケットを中心に開発する手法をチケット駆動開発という。これは作業を開始する前に必ずチケットを発行することを原則とした開発手法である。この開発手法を運用しているプロジェクトは、未完了チケット数と完了済チケット数の時系列変化から進捗状況を判断できる。

2. 研究目的

GitHub 上のプロジェクトを対象とし、プロジェクト成功の成否を調査する。調査後、チケット数の時系列変化に着目し、データマイニング手法を用いて成功の成否に関連するパターンを発見する。

3. 研究方法

本研究では、Star 数がプロジェクトの成功要因であると仮定する。この仮定の真偽を検証するため、100 件のプロジェクトから Star 数とリリースされたソフトウェアのバージョンを取得し、相関関係を調査する。成功の成否は、リリースされたソフトウェアのバージョンから判断する。Ver. 1.0 以上をリリースしているプロジェクトを成功とし、Ver. 1.0 未満でリリースが止まっているプロジェクトを失敗とする。

プロジェクトの分類には Issues を利用する。Issues (GitHub 上でのチケット) は GitHub 内の Star 数ランキング上位 50 件のプロジェクトとランダムに選択した 50 件のプロジェクトから、API を用いて取得する。取得する Issues は OpenIssues (GitHub 上での未完了チケット) と ClosedIssues (GitHub 上での完了済チケット) の 2 種類である。

分類に使用する解析手法は、階層クラスター分析と非階層クラスター分析、自己組織化マップである。その変数は、時系列データにフィットする線形式の係数と Issues の数、Issues を完了するまでの所要時間の平均と標準偏差である。分類結果から、成功の成否に関連するパターンを発見する。

4. 結果・考察

100 件のプロジェクトの Issues データを時系列解析し、成功の成否に関連するパターンを発見した。成功しているプロジェクトには Star 数が多く、Issues を完了するまでの所要時間が短いという共通点が見られた。また、失敗しているプロジェクトには Star 数が少なく、Issues を完了するまでの所要時間が長いという共通点が見られた。

Issues を完了するまでの所要時間がプロジェクト成功の成否に関連している理由は、Issues の完了の遅れがプロジェクトの進捗に関わるからであると考えられる。

参考文献

- [1] 小川明彦, 酒井誠. チケット駆動開発. 翔泳社, 2012-8-23.
- [2] 久保孝樹. チケットを活用するオープンソースソフトウェア開発の実態調査. 千葉工業大学, 2013, 卒業論文.

バージョン管理システムを活用するソフトウェア開発の開発フロー

ソフトウェア開発コース 矢吹研究室 1142032 小野寺航己

1. 研究の背景

当研究では、バージョン管理システム **GitHub** を用いたソフトウェア開発プロジェクトにおける、開発フローを対象とする。

バージョン管理システムは、何度も変更を加えたファイルであっても、作成日時や、編集日時、変更点を随時保管しておくことで、複数の人間が過去のファイルや、変更点の確認、ファイルの状態を復元することなどの、管理を可能とするシステムである。

代表的なバージョン管理システムとして **Git** がある。**Git** では、このシステムを使っている各利用者が自由にできる領域（ローカルリポジトリ）に、**Git** に保管されたデータや情報の全履歴を含んだ、完全な複製が作られる。このような仕組みがあるため、ネットワークにアクセスできなくても、履歴の閲覧や変更の記録といった、作業ができる。

この **Git** を用いたウェブシステムとして **GitHub** がある。**GitHub** は **Git** の機能を提供するウェブサービスであり、世界中の人々が自分の作品を保存、公開することもでき、ソフトウェア開発プロジェクトのための共有サービスでもある。

ソフトウェア開発では、一つのソフトウェアに対して複数のメンバが同時に編集を行うことで、複数のリリースバージョンがある中で、機能の追加やバージョン管理といった、作業を並行して行わなければならない状況がある。

Git にはそのような状況を支援する機能としてブランチがある。ブランチは、履歴の流れを分岐して記録できる機能で、同じソフトウェアに対して複数の変更を、履歴を保持しつつ同時におこなうことができる。この機能があるため、ソフトウェア開発のツールとして **GitHub** が用いられることが多い。

2. 研究目的

GitHub を用いたソフトウェア開発プロジェクト毎に、適切な開発フローを選択する基準を明確にする。

3. 研究方法

GitHub を用いる開発フローを網羅的に調査して、調査した開発フローのリスクを明らかにする。

そのリスクをデータマイニング手法により分類し、その後、プロジェクトの性質に応じて適切な開発フローを選択できるようなガイドを作成する。

使用する手法は階層的クラスター分析である。調査した **GitHub** を用いた開発フローと、そのフローを用いることで想定されるリスクをもとにした分析から開発フローの選択基準を発見する。

4. 結果

クラスター分析をしたところ、表 1 のような 5 つのクラスターに分類することができた。それぞれのクラスターに属するフローには、「フローが自動化されている」、「リポジトリを複数使用しているもの」、「使用したブランチを破棄するもの」、「アジャイル開発のような流れが可能なもの」、「使用するブランチが少ないもの」という共通点があった。

表 1. 開発フロー分析結果

1	フィヨルドフロー, イストフロー
2	Aming フロー, サイボウズフロー, 矢吹研フロー①, 矢吹研フロー②
3	ラクスルフロー, LINE フロー
4	Git フロー[1], キャスレーフロー
5	GitHub フロー[1], はてなブログフロー[2], 日本 CAW フロー

5. 考察

分析結果から、「フローを自動化することで、定期的にテストを行いたいかどうか」、「リポジトリを複数使いたいかどうか」、「ブランチを盛り込むべき機能やバージョンで管理したいのか」、「アジャイル開発でプロジェクトを行うのか」、「プロジェクトメンバにかかる負荷を少なくしたいのか」という点が、適切な開発フローを選択する基準になるのではないかと考える。

参考文献

- [1] 大塚弘記. **GitHub 実践入門 Pull Request** による開発の変革. p. 199-253. 技術評論社, 2014.
- [2] 新野淳一. はてなブログチームの開発フローと **GitHub** (前編). Publickey. http://www.publickey1.jp/blog/14/githubgithub_kaigi_2014.html (参照 2014-09-19)

ビッグデータ解析ソフトウェアを用いた GitHub データマイニング

プロジェクトマネジメントコース 矢吹研究室 1142046 小池 由也

1. 研究背景

ソフトウェア開発プロジェクトのための共有ウェブサービスである、GitHub のプロジェクトについて調べれば、オープンソフトウェア開発プロジェクトの実態がつかめるはずである。

実際に GitHub を調べて分かったことの例として、怒りの表現を含むコミットメッセージの割合、地域によるオープンソースプロジェクトへの貢献者などの分布図などがあげられる。これらの結果は、GitHub Data challenge というイベントで上位に入賞している分析結果である。

GitHub のデータ解析は難しい。なぜなら、データが膨大なため、その収集と処理が難しいからである。データの収集が難しいという問題は、一つのプロジェクトにより簡単になった。大量のデータを集めるために、GitHub のプロジェクトのタイムラインを記録し、アーカイブ化させ、簡単にアクセスできるようにするためのプロジェクトで GitHub Archive である。

データの処理が難しいという問題は、データ量が多すぎるために膨大な量のデータを処理するソフトウェアが少ない点である。GitHub Archive と連動させデータ処理ができるソフトウェアに Google BigQuery がある。BigQuery は、簡単にビッグデータを処理するためのソフトウェアであり、SQL に似たクエリを従来のやり方よりも短時間で簡単に実行できる。このソフトウェアの登場により手軽に大量のデータを処理することができるようになった。

これまでの調査で、オープンソフトウェア開発でどのようなプログラミング言語がよく使われているかを調べることに成功したが、プロジェクトが Fork される確率の、プログラミング言語による違いが分かれば、オープンソフトウェア開発プロジェクトについての理解が深まると思われる。Fork とは、GitHub 上で公開されている成果物に独自の変更を加える際に行う複製のことである。

2. 研究目的

GitHub 上で公開されているオープンソフトウェア開発プロジェクトを Google BigQuery を利用し調査する。オープンソースソフトウェアの開発プロジェクトにおいて、使用するプログラミング言語が異なると、Fork される確率、つまりプロジ

ェクトに貢献する人が現れる確率が異なるということが課題研究の段階で分かった。

しかし、この結果は、Fork された回数が多いものについてのみ調査して得られたものであった。そこで本研究では、Fork された回数が非常に少ないものも対象にして、プログラミング言語による貢献者の出現確率を調査する。

3. 研究方法

大量のデータを処理することが予想されるので Google BigQuery を利用する。Google BigQuery を使って、GitHub 上のプロジェクトが採用しているプログラミング言語と Fork されている数を集める。BigQuery で集めたデータをプログラミング言語ごとに統計処理する。

4. 結果

Google BigQuery を利用して 2012 年 2 月 12 日から 2014 年 11 月 18 日までの GitHub のタイムラインデータを集めた。集めた約 20 万件プロジェクトのデータを MySQL で整理した結果の一部が表 1 である。

	プロジェクト数	平均Fork数	標準偏差
全体	209405	62.7892	353.0447
JavaScript	37795	127.6464	571.9613
Ruby	35802	152.9987	591.7913
Java	24371	15.8745	54.6264
Python	23760	21.0954	63.4868
PHP	21377	36.4115	137.6646
C	12340	15.5236	64.0051
C++	10216	35.4933	140.845
Shell	7060	31.1657	181.6001
Objective-C	6857	48.5569	111.374

表 1 言語ごとの平均 Fork 数と標準偏差

5. 考察

上記の結果を JavaScript と Ruby で見比べてみると JavaScript の方が Ruby よりプロジェクトの数は多いのだが、Fork される確率は、Ruby の方が JavaScript より多い。他の言語でもプロジェクト数が少ないのに平均 Fork 数が多いものもあるのでプロジェクトを立ち上げる側で多い言語が必ずしも参加者側からも人気とは言えないことが分かる。

参考文献
[1] Brian Doll. The GitHub Data Challenge
<https://github.com/blog/1118-the-github-data-challenge>
e (参照 2015-1-26).

物語を活用するプロジェクトマネジメント教育

プロジェクトマネジメントコース 矢吹研究室 1142064 鈴木淳子

1. 研究の背景

プロジェクトマネジメントに関する知識及び活動には、多くの経験から蓄積された実務的な経験が必要となる。そのため、多くの企業でプロジェクトマネジメントの知識を習得するための研修が行われている。

本研究では、PM の知識を習得するために行われている「【PM 桃太郎！】～プロジェクトマネジメントを学ぼう～」^[1]（以後、PM 桃太郎）というセミナーに着目する。

このセミナーは、特定非営利活動法人 Layer Box が主催しており、小学校高学年から中学生を対象とし、2 日間かけて行われるプログラムである。そのプログラムでは対象者が理解しやすいように、昔話の「桃太郎」を用いている。

プログラムの成果物は、グループごとに作成された絵本であり、その内容は、「桃太郎」をプロジェクトマネジメントの視点から解釈し、考察したものである。

2. 研究の目的

本研究の目的は、物語を活用してプロジェクトマネジメントの知識を学ぶことのできるプログラムを考案することである。

本研究では、上記のプログラムを考案し、プロジェクトマネジメントを学ぶ大学生を対象に考案したプログラムを実施し、結果を検証する。

3. 研究の方法

本研究のプログラムは、千葉工業大学社会システム科学部プロジェクトマネジメント学科に所属する 1 年生が参加する、東京ディズニーシーでのオリエンテーションで実施する。

対象者は、千葉工業大学社会システム科学部プロジェクトマネジメント学科 1 年生の矢吹グループ（以後、矢吹グループ）に所属する 10 名であり、私が考案したプログラムを実施し、その結果を分析し、考察する。

プログラムの実施内容を以下に記載する。

(1)期間

4 月 21 日～7 月 5 日（東京ディズニーシーのオリエンテーションの期間）

(2)内容

はじめに、私が「PM 桃太郎^[1]」について調査を

行い、プログラムを考える。

次に、対象者が東京ディズニーシーに関する物語を調査し、調査した物語に沿って PM の一連の活動を擬似する。私はこの活動に沿って、考案したプログラムを実施する。

(3)検証方法

プログラムの実施後、アンケート調査を行う。次に、学生の成績（一般的な科目の成績と PM に関する授業の成績）を収集し、分析をする。

本研究は、分散分析を用いて、本プログラムの効果を検証する。さらに、本研究で考案したプログラムは、学生を対象とするため、学生が本プログラムで学びたいと思うようなプログラムを考案しなければならない。そのため、本プログラムが学生の興味を引くことができたかどうかを検証する必要がある。その成果を検証するために、アンケート調査を行う。

4. 研究結果

分析結果から、私の考案したプログラムを実施した矢吹グループと実施していない学生とでは、本プログラムを実施した違いによる学力の差が出なかったと示された。しかし、アンケート調査によって、プログラムを実施した矢吹グループのプログラムにおける満足度を得ることができた。これらの研究結果をふまえ、プログラムの修正を行い、完成したプログラムについて考察をし、本研究を終了した。

5. 考察

私の考案したプログラムは、効果をきちんと証明できなかった。しかし、本研究課題研究から続けていた本研究は、課題研究の最終発表や卒業研究の中間発表にて、多くの学生が興味を示して積極的に話を聞いてくれた。このことは、本手法が学生にプロジェクトマネジメントへの興味を持たせるための有効な方法であることを示唆している。

参考文献

[1] 特定非営利活動法人 Layer Box. 【PM 桃太郎！】～プロジェクトマネジメントを学ぼう～. (2013-10-21).

<http://www.aoniyoshi.us/layerbox/110326-pmmomotaro.htm>

Wikipedia 人的資源マイニング

プロジェクトマネジメントコース 矢吹研究室 1142066 曾我勇貴

1. 研究背景

Wikipedia は、多くの人がボランティアで執筆するオンライン百科事典プロジェクトである。

Wikipedia は 2001 年 1 月 15 日に創設された。2003 年 6 月 20 日に「ウィキメディア財団」がジミー・ウェールズにより創設された。現在 2014 年 9 月 1 日では、287 言語の Wikipedia が開設されており、この全てのオープンコンテキストの知識資源は無料で一般に提供されている。

多くの人々が参加するプロジェクトの代表例である Wikipedia を調査することによって、このような形式のプロジェクトのマネジメントについての有意義な知見が得られることが期待される。

ウィキペディアの成功理由についてはさまざまな考察がされている。例えば、「Google」の検索結果の上位(最初のページ)に「Wikipedia」の項目が表示されていることが多い。これは、Wikipedia にリンクするウェブページがおおくなった結果であるが、このことによって、Wikipedia を利用する人はさらに増えたはずである。

しかし、Wikipedia の衰退が始まっているという報告もある[1]。Wikipedia 英語版では、執筆者の協力を失いつつあり、2007 年から 2012 年までに執筆者の 30%を失っている。(現在 3 万 1000 人。世界の Wikipedia の総計は 25 万人である[1])。Wikipedia の日本語版の統計[2]を見ても同じような傾向がみられ、2009 年 11 月 1 日の 11,310 人に対して、2013 年 12 月 12 日は、10880 人と減少している。

本研究では、参加者の振る舞いのデータをマイニングすることによって、Wikipedia の成功理由を見つけることを目指す。さらに、衰退してきている Wikipedia の執筆者の離脱を防ぐ手段について考案する。

2. 研究目的

Wikipedia を一つのプロジェクトとみなし、このオンライン百科事典で人的資源がどのように活用されているかを調査する。この調査により、オープンな共同作業プロジェクトにおける、人的資源マネジメントのあり方についての知見を得たい。

3. 研究方法

Wikipedia の作業履歴データを取得し、どのよう

に人的資源が活用されているかを明らかにする。また、オープンな共同作業プロジェクトにおける人的資源マネジメントのあり方を提案する。

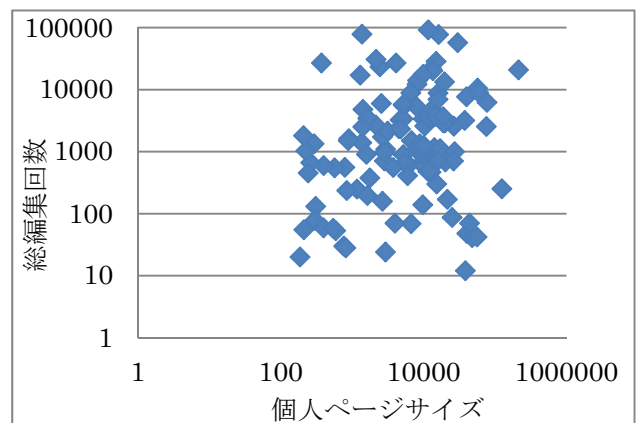
4. 結果

Wikipedia の編集回数の多い 200 人[3]について、以下の項目を調査した。

- Wikipedia での編集回数
- 編集者自身について書かれた Wikipedia 上のページのバイト数
- 1 回の編集における更新バイト数の平均と標準偏差

5. 結論・考察

Wikipedia での総編集回数と編集者自身について書かれた Wikipedia 上のページのバイト数の関係は図のようになる。



貢献度合いを可視化するランキングや、貢献者についての情報を記録する仕組みの存在が、Wikipedia のようなオープンなプロジェクトの成功に寄与していることが示唆される。

参考文献

- [1] Alessandro Longo.(2013)「Wikipedia が岐路に立っている？」<http://wired.jp/2013/11/27/the-decline-of-wikipedia/>(参照 2013-11-27).
- [2] Tom Simonite (2013)「ウィキペディアの衰退」<http://www.technologyreview.com/featuredstory/520446/the-decline-of-wikipedia/>(参照 2013-10-22).
- [3] 秋雪 (2006)「Wikipedia:編集回数の多いウィキペディアンの一覧」<http://ja.wikipedia.org/wiki/Wikipedia:編集回数の多いウィキペディアンの一覧>(参照 2014-8-11)

リアルタイムに動画を処理するスマートフォンアプリケーションの開発

プロジェクトマネジメントコース 矢吹研究室 1142078 土井 貴司

1. 研究背景

1.1 現代の情報社会

私たち人間は、多くの情報を目から得ている。近年では、スマートフォン・タブレットの利用者が急増している。なぜなら、スマートフォン・タブレット向けのアプリケーションやサービスの提供が急速に広がりを見せているからである。それにより、以前よりも手軽に写真や画像、動画を見ることができた。また、写真や画像はカメラアプリケーションなどを用いて、編集や画像処理ができるようになった。

1.2 画像処理

画像処理とは、デジタル化した写真や絵画、カメラなどで撮った映像などの画像情報を見やすくするために、別の加工・変換したり、その画像の形状や色などの特徴を抽出したり、画像が何を表しているのか認識したりする処理をいう[1]。

1.3 画像処理における問題点

スマートフォンやタブレットにより、アプリケーションなどを用いて処理することが容易になってきている。しかし、リアルタイムに動画を処理するアプリケーションを調べたところ、見当たらなかった。動いているものを消したり、見やすいものに変えたり、特定の情報だけを取り出して得たりしようとする画像処理技術を一般ユーザーが出来るわけではない。例えば、一般ユーザーが専用ソフトを使おうとしても、言葉の意味が理解できず、開発途中でうまくいかず断念することがある。したがって、画像処理をするのは容易ではない。カメラアプリケーションや画像編集アプリケーションのように、動画を処理するアプリケーションはないかと考えた。

1.4 iOS アプリケーションの開発

iOS 専用の開発ツールとして、Xcode を用いる。Xcode は、強力なソースエディタや視覚的なデバッガなど、優れたアプリケーションを作成するための完全なツールセットである。作成したものを確認するためには、iOS Developer Program への登録が必要である。それにより、実機上でどのように動作するのか確認することができる。

2. 研究目的

本研究では、リアルタイムに動画を処理するスマートフォンアプリケーションを開発する。開発するアプリケーションは、スマートフォンやタブ

レットのベンチマークとなることも期待される。リアルタイムに動画を処理するスマートフォンアプリケーションは、動いているものを消すということが目的である。そして、このようなアプリケーションを開発することで、いつでも簡単に利用でき、編集する時間と手間が省くことができる。また、開発に必要なツールやプログラミングソースについて、知識を得ることができる。

3. 研究方法

リアルタイムに動画を処理するスマートフォンアプリケーションの開発をするうえで、iOS の実機上で動かせるようにする。iOS アプリケーションの開発に必要な Xcode で作成する。作成したものは、iPhone Developer Program で登録したのちに、実機上で確認する。このとき、使用するプログラミング言語は Objective-C である。

4. 結果

開発を行う際に、iOS Developer Program などのツールを揃え、開発に必要なカメラとビデオ動画アプリケーションを完成させ、実装に成功した。動いている物を消すには、カメラの各ドットの RGB 値の累積を記録し、フレーム数で割ることににより平均値が求められ、消すことができた。

5. 考察

実際に RGB の平均値を取ることで、動くものを消すことができた。長く時間を取れば、最頻値の方が正確に処理できると思うが、コマ数を記憶する為のメモリが多く必要になるため、今回の目的にはあまり適していないと言える。リアルタイムに処理する際には、平均値で十分であることが本研究で実証された。

参考文献

- [1] 村上伸一. 学生のための画像処理プログラミング演習. 東京電機大学出版, 2012-4-10.
- [2] 大重美幸. 詳細! Objective-C iPhone アプリ開発 入門ノート Xcode5+iOS 7 対応. ソーテック社, 2013-11-02.
- [3] 木下誠. 実践! iPhone アプリ開発. ビデオカメラアプリの作り方 (1) - セッションの作成. <http://news.mynavi.jp/column/iphone/040/>, (2015-1-20)

データマイニング教育におけるアクティブ・ラーニングの実践

プロジェクトマネジメントコース 矢吹研究室 1142104 松本 併太

1. 研究背景

中央教育審議会が検討する学習指導要領の全面改訂で、目玉の一つがアクティブ・ラーニングの導入だといわれている。

文部科学省が定義するアクティブ・ラーニングの定義とは次のようなものである。「アクティブ・ラーニングとは、教員による一方的な講義形式の教育とは異なり、学修者の能動的な学修への参加を取り入れた教授・学習法の総称。学修者が能動的に学修することによって、認知的、倫理的、社会的能力、教養、知識、経験を含めた汎用的能力の育成を図ることである。発見学習、問題解決学習、体験学習、調査学習等が含まれるが、教室内でのグループ・ディスカッション、ディベート、グループ・ワーク等も有効なアクティブ・ラーニングの方法である。」⁽¹⁾

1990年代以降は、基礎的な知識に加え、多様性や創造性や他者と交渉する力などを備えた、新しい社会を創出できる人材が求められるようになった。そこで大学では、講義で知識を一方的に教えるだけでは不十分で、学生が自ら頭を使って考えたり議論したりするような教育を行うことが必要となったのである。

アクティブ・ラーニングは大学における従来の講義方法から脱却するための良い教育理念であることは確かである。また、アクティブ・ラーニングの方法は、生涯にわたって学び続ける力、主体的に考える力のために必要な能力とされていることも確かである。したがって、高度教養教育のデザインにアクティブ・ラーニングの手法を取り入れてゆくべきことは、必須である。

2. 研究目的

アクティブ・ラーニングをデータマイニング教育に取り入れ、学習者の能動的な学習への参加を取り入れた能力の育成を図る。与えられたデータをマイニングするのではなく、データをどうやって集めるか、データ収集法の設計から考え、学習する。与えられたデータでは学習者自身の問題ではなく、馴染みがないが、この研究では学習者自身を被験者とするすることで、その問題が解決されるという利点がある。

3. 研究方法

この計画は5週間に分けて行う。対象は千葉工業大学データマイニング入門を受講している学生、122人とする。千葉工業大学では、学生にタブレット端末を貸与している。その貸与しているタブレット端末の利用実態調査を題材とし、データを収集する。4、5人で1グループに分け32グループにし、何を知りたいか考え、各グループ、質問を3つずつ考えてもらう。それをGoogleフォームにまとめ、アンケートを作成する。各グループが考えた質問をまとめたアンケートをすべての学生に回答してもらい、解析手法を学んだ後、自分のグループの質問の結果と全ての質問の結果、2つをデータマイニングしてもらう。その結果からわかったことを発表してもらう。

4. 結果・考察

データマイニング入門の授業でアクティブ・ラーニングを計画の通り運用した。「動画サイトは何を利用していますか？」や「GPAはどの範囲に当てはまりますか？」などの質問を学習者自身がタブレット端末の利用と関連があるかを考えて作っていたことで、学習者の能動的な学習への参加を取り入れた能力の育成をすることが出来た。しかし、学習者のデータマイニング結果から有意な結果があまり得られなかったことから、質問作りの段階からどのような結果を得ることが出来るか想定して計画していく必要があったと考える。

この実験で、千葉工業大学データマイニング入門の授業にアクティブ・ラーニングを運用した。今回の結果を生かすことによって、データマイニング教育に学習者の能動的な学習への参加を取り入れた能力の育成が出来る。

参考文献

- [1] 中央教育審議会. 用語集. 「新たな未来を築くための大学教育の質的転換に向けて～生涯学び続け、主体的に考える力を育成する大学へ～」, p. 37, 2012.
- [2] 教育改革 ing. アクティブ・ラーニング. Kawaijuku Guideline, Vol. 11, p. 44-51, 2010.
- [3] 池田光穂. 教育方法としてのアクティブラーニング. <http://www.cscd.osaka-u.ac.jp/user/rosaldo/131112AL.html> (参照 2015-01-19).

GitHub 活動ログからの人的資源マネジメント

プロジェクトマネジメントコース 矢吹研究室 1142106 丸山 準人

1. 研究背景

人材マネジメントに統計分析を活用する試みがスポーツ界で広まっている。

例えば、野球界ではマネー・ボールという実際にあった物語がある。その中で登場するアスレックスは、他の球団に比べ資金面で劣っていた。そこで選手の成績を統計的に分析し、よりチームのために貢献できる選手を少ない資金で獲得した。その結果、全球団の中で最高の勝率を記録した。

サッカー界ではチームを統計的な手法で強化するのは、野球チームを統計的な手法で強化するよりも難しかった。その中でセイバーメトリクスを利用して成功を収めているチームがある。それは、プレミアリーグ（イングランドのリーグ）のニューカッスルである[1]。ニューカッスルは、ライバルチームの 1/5 の資金しかなく次々とチームの核となる選手を引き抜かれていた。しかし、怪我をしたために他のクラブの興味を引かなくなっていた元有名選手を獲得した結果、チームはリーグ戦 7 位という好成績を収め、EL（ヨーロッパリーグ）への出場権を獲得した。トップ 10 が目標だったチームには EL 出場という結果は、成功と言っても良いだろう。

本研究では、スポーツ界で行われているような統計解析手法を用いた人材マネジメントの、ソフトウェア開発の現場への導入を検討する。

現在、OSS 開発は GitHub 上で行われていることが多い。

OSS 開発とは、オープンソースソフトウェア開発のことである。その特徴として企業、個人など参加形態を問わずに誰でもプロジェクトに参加することが可能である。

過去に GitHub 上で行われているプロジェクトの各メンバの活動ログを収集し、役割分担の実態を明らかにする研究が行われていた[2]。この研究で、Push する行為とリポジトリにスターを付ける行為は別のメンバが行っていることが多いことが明らかになった。

そこで今回は、プロジェクトのコミット総数と個人がコミットの追加、修正で書いた行数が何行かを調べて、プロジェクトへの貢献度を調査する。

2. 研究目的

GitHub 上で多く行われている OSS 開発のプロ

ジェクトを用いて、活動ログを統計解析手法で分析する。その結果から個人のプロジェクトへの貢献度を調査する。

3. 研究方法

- ① GitHub 上で行われている OSS 開発のプロジェクトのコミット総数を調査する。
- ② ①で調査したプロジェクトメンバの個人がコミットの追加・修正をした行数を調査する。
- ③ ②で得られた活動ログを統計解析手法で分析する。
- ④ ③で得られた結果から個人のプロジェクトへの貢献度がどのくらいいかなのかを考察する。

4. 結果

今回 5 つのプロジェクトを調査した。square/spoon というプロジェクトを例に挙げる。編集回数が多い Jake Wharton は 1 回のコミットで平均約 91 行の追加・修正をしていた。編集回数が少ない Jack Harrison は 1 回のコミットで 1 行の追加・修正しかしていなかった。このように、メンバによるプロジェクトへの貢献度の違いを定量化することができた。

5. 考察

例に挙げたプロジェクトに参加していたのは 27 名だった。そのメンバがコミットで追加・修正した行数は合計で 35,449 だった。そのうちの 31,669 の行数は 3 名によって書かれていた。このことからこのプロジェクトの全体の行数の約 8 割はメンバの約 2 割が書いている結果となった。

一般に言われているパレートの法則（80:20 の法則）が OSS 開発の現場でも成り立つ可能性が示唆される。

参考文献

- [1] 山中忍. プレミアリーグ版“マネー・ボール”？清貧クラブのニューカッスルが躍進。2012. <http://number.bunshun.jp/articles/-/216542>
- [2] 関口元基. オープンソースソフトウェア開発における役割分担の実態調査。千葉工業大学, 2013, 卒業論文。

ユーザ関係の分析による Twitter コミュニティ抽出

PM コース 矢吹研究室 1142123 渡邊雄大

1. 研究背景

コミュニケーションツールとして、Social Networking Service (SNS) を使用している人はとても多くいる。その中でも Twitter は、SNS を代表する 1 つである。なぜなら Twitter はアクティブユーザ数が 2 億 3 千万人もいることだけでなく、ツイートと呼ばれるマイクロブログが一日平均で 5 億件も送信されているからだ(2014 年 9 月現在)。そのため Twitter は、調査する価値のある SNS の 1 つであると考えられる。

Twitter はツイートと呼ばれる短い文字列を投稿するためのサービスである。自分以外のユーザのツイートを読むためには、そのユーザのページにアクセスするか、そのユーザをフォローする必要がある。フォローしているユーザのツイートは、ひとまとめにされ、タイムラインを形成する。誰が誰をフォローしているかという情報（フォロー関係）は、Twitter におけるユーザのつながりの一つの表現である。仮に Twitter 上でユーザがコミュニティを形成していたとすれば、フォロー関係にもそれが反映されていると思われる。そのコミュニティを抽出することができれば、フォローすべきユーザの発見が容易になるなど、Twitter のユーザビリティが大きく向上することが期待される。

2. 研究目的

検索したいユーザの Twitter 上に持っているフォロー関係から、そのユーザの持つ実際のコミュニティを Twitter の機能であるリツイートを解析することによって見つけ出す。さらに、この研究を行うことで、プロジェクトを円滑に行うための人的資源マネジメントとして活用することを目指す。

3. 類似研究

ユーザのフォロワーとのつながりについての研究として[1]が挙げられる。この研究では、フォロー関係からクラスタリング分析を行い、その結果からクラスタを出力できるシステムを作成している。この研究はシステムが結果として出力したクラスタの内容と、手作業で作成した理想のクラスタを比べることで研究の評価を行っている。

4. 研究方法

この研究を説明する際に使用する用語を以下の表 1 にまとめた。

表 1 使用する用語

リツイート A	観測されたリツイート
ツイート A	リツイート A のオリジナルのツイート
ユーザ A	ツイート A のユーザ
ユーザ B	リツイート A のユーザ
指標 1	ユーザ B がユーザ A をフォローしていない確率
指標 2	ユーザ B がツイート A をリツイートした人たちがフォローしていない確率

まずこの研究では指標 1 と指標 2 を導き出す。そのために TwitterAPI の User streams の機能を使い、自分のフォローしているユーザの行ったリツイートのみを抽出し保存する。この保存されたリツイートの中からリツイート A を使用し研究を行ったとする。この場合はユーザ B のフォローしている相手とツイート A をリツイートしたユーザ B 以外の相手を取得する。取得したリツイートの全体と指標 1、指標 2 の関係は以下の図 1 のようになる。

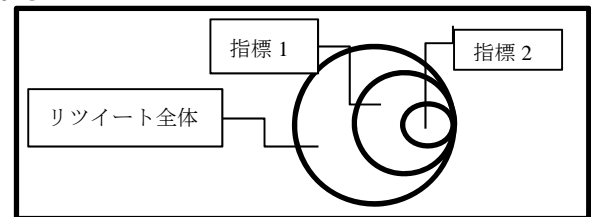


図 1 リツイート全体、指標 1、指標 2 の関係

5. 研究結果

この研究では成果物を研究の結果から得られた 2 つの指標とする。指標 1 は 60%であった。指標 2 は 40%であった。

6. 考察

指標 1 と指標 2 の結果から確認できるように、多くのユーザ達がフォローして無いユーザのツイートも確認し、リツイートを行っていることが分かった。これはフォロー関係によって繋がっていると思われていた従来の Twitter ユーザ同士のネットワークの形ではない別の形のネットワークがある可能性が示された。

参考文献

[1] 山下拓也・佐藤晴彦・小山聡・栗原正仁(2013)「フォロー関係に基づく Twitter ユーザの分類」『第 75 回全国大会講演論文集』pp.107-109 情報処理学会