

Android アプリストアにおける不自然なレーティング・レビューの解析

孫 博† 渡邊 卓弥† 秋山 満昭‡ 森 達哉†

†早稲田大学 基幹理工学研究科
169-8555 東京都新宿区大久保 3-4-1

{sunshine, watanabe, mori}@nsl.cs.waseda.ac.jp

‡NTT セキュアプラットフォーム研究所
180-8585 東京都武蔵野市緑町 3-9-11

akiyama.mitsuaki@lab.ntt.co.jp

あらまし 現在のモバイルアプリストアにおいては、ユーザが様々なアプリを選択する際に、レビューやレーティングを参考にすることは一般的である。しかし、悪性アプリを多くのユーザにダウンロード感染させるために、アプリのレビューやレーティングを偽装する攻撃者が存在する可能性がある。本研究では、アプリストアで公開されているアプリのレーティング情報とレビュー情報を解析し、アプリストアに潜んでいる不自然な評価情報の状況を明らかにすることを狙いとする。本発表では、収集したデータセット、レーティング情報とレビュー情報の解析手法および解析の結果を報告する。

The analysis of unusual ratings and reviews on Android app store

Bo Sun† Takuya Watanabe† Mitsuaki Akiyama‡ Tatsuya Mori†

†School of Fundamental Science and Engineering, Waseda University
3-4-1 Okubo, Shinjuku, Tokyo 169-8555, JAPAN
{sunshine, watanabe, mori}@nsl.cs.waseda.ac.jp

‡NTT Secure Platform Laboratories
3-9-11 Midoricho, Musashino-shi, Tokyo 180-8585
akiyama.mitsuaki@lab.ntt.co.jp

Abstract In general, mobile users choose their favorite apps by referring to the ratings and reviews on app stores. However, adversary may leverage fake ratings and reviews by impersonating normal ones in order to make mobile users download malicious apps. We aim to reveal such unusual information hidden in app stores by analyzing the ratings and reviews on app stores. In this paper, we describe dataset collected from two Android app stores, the methods for analyzing ratings and reviews on two Android app stores and analysis results about unusual ratings and reviews.

1 はじめに

背景: スマートフォンの普及に伴い、大量のアプリが開発されている [1]。ユーザは公式およびサードパーティのアプリケーション配布プラットフォーム（以下アプリストア）を通じて容易にアプリを入手することが出来る。アプリストアでは、通常のユーザが書き込んだ評判情報に加え、ダウンロード数や開発者等のメタ情報を収集・公開している。モバイルユーザが様々なアプリを選択する際に、レビューやレーティングを参考にすることは一般的である。

BBC (British Broadcasting Corporation) の調査によると、幾つかの会社は自分商品の売り上げを伸ば

すために、レビュアーを雇ってポジティブなレビューと高いレーティングを投稿させている。現在、少なくとも 20% の商品の評価情報は不自然であることが報告されている [2]。

ユーザがレビューやレーティングを頼りにアプリを検索する状況を考えると、アプリ自体の脆弱性だけではなく、配布用サイトに付随するレビューやレーティング等の評判情報も攻撃者の標的となる可能性がある。具体的に攻撃者が悪性アプリを多くのユーザにダウンロード感染させるために、アプリのレビューとレーティングを偽造することで悪性アプリを良性アプリに偽装する脅威が存在する可能性がある。このような評判情報の汚染は悪性アプリをダ

ダウンロードしてしまう個々のユーザに対して危害を与えるだけでなく、アプリストア全体の悪評と信頼低下につながるリスクがある。このような不自然な評判情報による脅威の検証は重要な課題であるため、本研究は、Android アプリストアを解析対象として不自然なレーティングとレビューの状況を明らかにすることを目的とする。

提案: 上記の研究目的を達成するために、本研究は Android アプリストアから類似の挙動を示すレビューアグループ（以下類似レビューアグループ）を検出するアプローチをとる。類似レビューアグループの抽出に関しては、大規模なデータセットから有用な情報まで絞り込むスケールダウンの方針を実現する。具体的には、まず、前処理としては、データセットにあるノイズの除去を行うことで、データ規模の縮小とデータ正確性の確保を兼ね備える。次に、類似レビューア候補を検索することによってデータ量をさらに削減する。最後に、不自然なレーティングとレビュー解析用の類似レビューアグループを見つけ出す。従来手法 [3] より Frequent Itemset Mining を用いて類似レビューアグループを検索する技術が提案されてきた。これに対して、本研究は類似レビューアグループの組み合わせパターンの多様性を考慮した上で、類似レビューアグループの抽出を実現できるアルゴリズムを提案した。そのような類似レビューアグループを把握することにより、類似レビューアグループによる不自然なレーティングやレビューを早期に特定することが期待できる。

貢献: 本研究の主要な貢献は下記のとおりである。

- 公式およびサードパーティのアプリストアを対象に英語および中国語のレビューとレーティングを収集し、解析した。
- 大規模なレビューとレーティング情報から効率よく類似レビューアグループを検出するアルゴリズムを開発した。
- 二つの Android アプリストアから計 6 件の類似レビューアグループが捏造した 314 件の不自然なレーティングとレビューを検出でき、Android アプリストアに不自然なレーティングとレビューが存在することを明らかにした。

本論文の構成は以下の通りである。はじめに 2 章では関連研究と本研究の比較を示す。次に 3 章では本研究の提案手法を示す。4 章で提案手法の評価に用いるデータの詳細を述べた後、5 章にて Android ストアにおける不自然なレーティングとレビューの状況を示す。次に、6 章は本研究の欠点を述べる。最後に、7 章は本研究のまとめと今後の展望を示す。

2 関連研究

本章は不自然なレーティングとレビューの解析に関する従来研究を示す。

Jindal ら [4] はアマゾンからの商品レビューをデータセットとしてスパムレビューの解析を行った。解析手法については bi-gram と Jaccard distance を利用し、全ての類似レビューアペアを見つけ出した。その後、商品 ID とレビューア ID を分類基準として四つの種類に分けてスパムレビューを特定した。

Mukherjee ら [5] は Jindal らと同じくアマゾンからの商品レビューを解析対象としたが、スパムレビューではなく、類似レビューアグループの検索を行った。検索手法では、まず、Frequent Itemset Mining を使って候補の類似レビューアグループを全部洗い出した。その後、レーティング、評価時間およびレビュー内容により類似レビューアグループを見つけ出した。

上記の二つの従来研究とも Android アプリストアにおけるレーティングとレビューを解析対象としなかった。本研究では、Android ストアにおいて不自然なレーティングとレビューが存在するかどうかを明らかにする。また、本研究は、従来手法の Frequent Itemset Mining [3] よりデータセットの前処理と最適な類似レビューアグループを検索できるアルゴリズムを提案して不自然なレーティングとレビューの解析に適用する。

3 レーティングやレビューの解析手法

本章でははじめに類似レビューアグループを定義する。次にレビュー情報からノイズを除去する前処理を説明する。最後に類似レビューアグループを検出するアルゴリズムを示す。

3.1 類似レビューアグループの定義

図 1 に類似レビューアグループの例を示す。ノイズを取り除いた後のアプリの集合は $\mathbf{A} = \{a_1, \dots, a_c\}$ である。 $\mathbf{U}_i = \{u_{i1}, \dots, u_{im(i)}\}$, $\mathbf{S}_i = \{s_{i1}, \dots, s_{im(i)}\}$ を各アプリ a_i 内のレビューアとレーティングの集合とする。 $m(i)$ は各々のアプリ a_i が持つレビューア、レビュー、レーティングの数である。類似レビューアグループは下記の全ての条件を満たす。

表 1: 類似レビュアグループの解析手順

順番	対応処理	対応処理の内容
1	前処理 (3.2 節)	データセットにあるノイズの除去
2	類似レビュア候補の抽出 (3.3 節)	類似レビュアグループの定義の条件 1 と条件 2
3	類似レビュアグループの検出 (3.3 節)	類似レビュアグループの定義の条件 3

条件 1:各アプリ a_i に対して平均レーティング $V(a_i) = \frac{1}{m(i)} \sum_{1 \leq j \leq m(i)} s_{ij}$, 例 (図 1): $V(a_8) = 3$. もし $V(a_i) < 5$ であれば, そのようなアプリ a_i を残す.

条件 2:条件 1 から抽出したアプリ a_i の中に, レーティング $s_{ij} = 5 (1 \leq j \leq m(i))$ のレビュアを取り出す. 例 (図 1): アプリ a_5 に関しては, レーティング $s_{51} = s_{53} = s_{54} = s_{56} = s_{59} = 5$ のため, レビュア u_1, u_3, u_4, u_6, u_9 を抽出する. レーティングの値の幅では, Google Play は $s_{ij} = t (1 \leq t \leq 5)$ であり, サードパーティーは $s_{ij} = 0.5t (1 \leq t \leq 10)$ である.

条件 3:条件 2 で残されたアプリ a_i の中から類似レビュアグループの検索を行う. G_k はレビュアグループを構成するレビュアの集合である. 例 (図 1): $G_1 = \{u_1, u_3, u_4, u_6\}$. k はレビュアグループの数を指す. 各ユーザ u_j が評価しているアプリ a_i の集合を $A(u_j)$ とする. 例 (図 1): $A(u_1) = \{a_1, a_3, a_5, a_6, a_8\}$, $A(u_3) = \{a_1, a_3, a_5, a_6, a_9\}$, $A(u_6) = \{a_1, a_3, a_5, a_6\}$. レビュアグループ内メンバーが同時に評価しているアプリの集合を $\theta_k = \bigcap_{u_j \in G_k} A(u_j)$ とする. 例 (図 1): $\theta_k = \{a_1, a_6, a_8, a_9\}$, もし, 集合要素の個数 $|G_k| \geq 3$ 且つ $|\theta_k| \geq 3$ であれば, 類似レビュアグループとして判定する.

表 1 に上記の全ての条件を満たすための対応処理の流れを示す. 類似レビュアグループを効率良く検索できるために, 対応処理は大きく三つのステップに分けられる.

まず, 前処理 (3.2 節) では, デフォルトのネーム ID や同じアプリの異なるバージョンにあるレビュー等のノイズを取り除き, 類似レビュア候補の抽出用のデータを用意する. 次に, 類似レビュアグループ定義の条件 1 と 2 を満たすための処理を行い, 類似レビュアグループ検索用のデータを出力する. 最後に, 類似レビュアグループの検出アルゴリズムでは, 3 以上のアプリを評価している 3 人以上のレビュアグループを検索する.

レビュアグループのサイズに関しては, グループに評価されるアプリ数とグループ内のレビュア人数とも 2 未満の場合, 2 人の普通のユーザがたまたま同じアプリを評価しており, 不自然なレーティングとレビューではない可能性が高いため, 今回は評

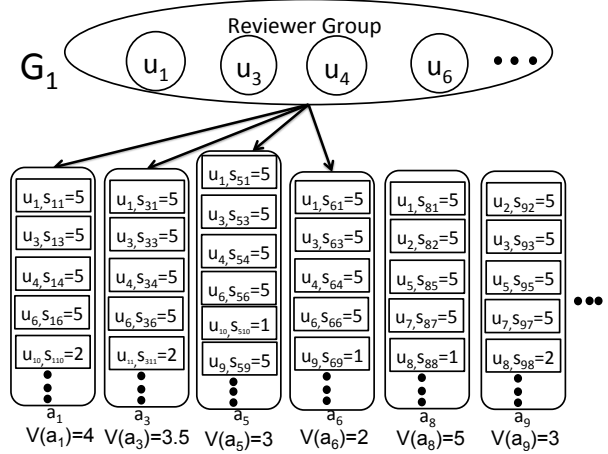


図 1: 類似レビュアグループの例

価アプリ数とレビュア人数とも 3 以上であると定義した. 残り 2 節で紹介する解析手法を用いて類似レビュアグループは Android アプリストアに存在するかどうか明らかにする.

3.2 データセットの前処理

データセットの前処理にはノイズを取り除くために下記のように三つのステップがある.

ステップ 1:二つの Android アプリストアとも異なるバージョンの同じアプリにおける全てのレビューが一致しており, 古いバージョンの同じアプリにあるレビューを全て削除して 1 件の最新バージョンのアプリにある全てのレビューのみ残すようにする.

ステップ 2:二つの Android アプリストアでは, ユーザがレビューを書く時にネーム ID を付けなければ, 各自システムのデフォルトのネーム ID ('A Google User' と '安楽网友') に切り替えられる. デフォルトのネーム ID は異なるユーザである可能性が高いため, データセットのノイズとなってしまう. そこで, ユニークの ID を確保するためにデフォルトのネーム ID のレビューを削除する.

ステップ 3:二つの Android アプリストアにおいては, 2 以下のレビュアを有するアプリが存在する. さらに, デフォルトのネーム ID のレビューを削除した後, 2 以下のレビュアを持つアプリが増加する可能性がある. 類似レビュアグループの定義に

よって3人以上のレビュアグループをターゲットにしてあるため、2以下のレビュアを持つアプリを検索対象から外した方が解析時間を短縮できる。そのため、類似レビュアグループ検索アルゴリズムの効率を良くするために、2以下のレビュアが有するアプリを削除する。

3.3 類似レビュアグループの検索手法

類似レビュア候補の抽出 (条件1と2):前述したように、類似レビュア候補の抽出では、類似レビュアグループの定義にある条件1と2をクリアする。即ち、前処理を通じてノイズを除去されたデータから定義の条件1と2に沿った類似レビュア候補の抽出処理を行い、次節の類似レビュアグループ検索用のデータを用意する。類似レビュア候補の抽出処理については、まず、残されたアプリに対して平均レーティングは5未満であるアプリを選定する。その後、そのアプリにある全ての5レーティング付きのレビュアとレビューを抽出する。類似レビュアグループを検索する前に、類似レビュア候補の抽出処理を実行することにより、定義の条件1と2を満たすだけでなく、データ規模を更に縮小し、コンピューターリソースを節約することができた。

類似レビュアグループの検出 (条件3):従来手法 [3]では、グループ内のアイテム数を閾値として指定して閾値以上の類似アイテムグループの組み合わせを全て出力する。例えば、3人以上の類似レビュアグループを検索すると、もしグループのレビュア人数は10人だとしたら、その10人から3人を選択する組み合わせから10人を選択する組み合わせまで出力される。そのため、類似アイテムグループの組み合わせの算出と全ての組み合わせに対する後処理は時間がかかってしまう。そこで、本研究では、全ての組み合わせではなく最大の類似レビュアグループを見つけ出せるために、類似レビュアグループの検索アルゴリズムを提案した。類似レビュアグループの検索アルゴリズムは二つのステップから構成されている。

- **ステップ1:**ステップ1では最大レビュア数のグループを検索することを目的とする。簡単のため、三つのアプリを用いて基本的な考え方と手順を説明する、前述したようにノイズを取り除いた後のアプリの集合は $\mathbf{A} = \{a_1, \dots, a_c\}$ である。各アプリ a_i における全てのレビュアの集合は $\mathbf{U}(a_i)$ とする。 $\mathbf{U}(a_1)$, $\mathbf{U}(a_2)$, $\mathbf{U}(a_3)$ は説明例として挙げられる。互いに $\mathbf{Q} = \mathbf{U}(a_1) \cap \mathbf{U}(a_2) \cap \mathbf{U}(a_3)$

のような論理積を行った後、もし集合要素の個数 $|\mathbf{Q}| \geq 3$ の場合、 \mathbf{Q} を類似レビュアグループとして判定する。全体処理の流れに関しては、集合 $\mathbf{U} = \{\mathbf{U}(a_1), \dots, \mathbf{U}(a_c)\}$ をアルゴリズムの入力データとして利用する。 \mathbf{U} から二つずつの要素 $\mathbf{U}(a_x)$ を選択して共通した部分を見つける論理積 $\mathbf{Q}_k = \mathbf{U}(a_x) \cap \mathbf{U}(a_y) (1 \leq x \leq c, x+1 \leq y \leq c)$ を行う。その後、類似レビュアグループの定義により、3人以上のレビュアグループを検索する必要があるため、もし $|\mathbf{Q}_k| \geq 3$ という条件を満たす \mathbf{Q}_k の個数は2以上存在する場合、それらに対してもう一度論理積 $\mathbf{G}_z = \mathbf{Q}_x \cap \mathbf{Q}_y (1 \leq x \leq n, x+1 \leq y \leq n, n \leq k)$ を行い、もし $|\mathbf{G}_z| \geq 3$ であれば、 \mathbf{G}_z は類似レビュアグループとなる。全体の集合が論理積により異なる集合に変換される回数を集合の変換回数 p とする。例えば、 $\mathbf{U}(a_i)$ から \mathbf{Q}_k に変換されると集合の変換回数 $p = 1$ となる。集合の変換回数 p に関しては、 p が大きければ大きいほど、検出された類似レビュアグループ内のレビュア人数が減少していく。今回は最大のレビュア数を求めているため、集合の変換回数 $p = 2$ に設定した。

- **ステップ2:**ステップ1では最大レビュア数のグループが存在するアプリ数は3件しか検索できないため、ステップ2はステップ1で抽出された類似レビュアグループが評価している全てのアプリ数を見つけ出すことを目指す。ステップ1から検出された類似レビュアグループの中に同じ構成となるレビュア集合 $\mathbf{G}_x = \mathbf{G}_y (1 \leq x \leq z, x+1 \leq y \leq z)$ があれば、それらが評価しているアプリを合併する。

上記の二つのステップを通じて類似レビュアグループを検出できた。本研究が提案したアルゴリズムは容易に実現できる。

4 データ収集環境とデータセット

本研究では解析に必要なレーティングやレビュー等のデータを二つのAndroidアプリストアから収集し、データセットを構築してきた。本章では、そのデータセットの収集方法と統計値を示す。

4.1 データ収集環境

データセットの多様性を考慮した上で、二つのAndroidアプリストア (Google Play [6] とサードパーティー [7]) が公開されている評価情報を取得した。

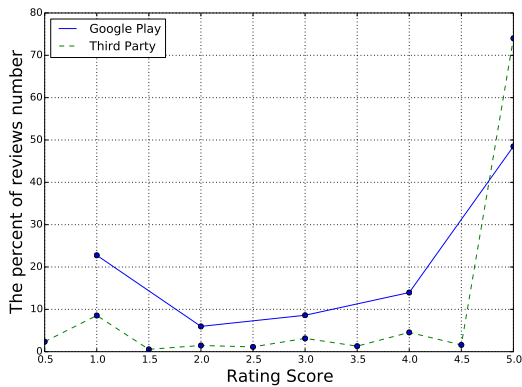


図 2: レーティングの分布図

Google Play とサードパーティーに関しては評価情報が記述されている HTML のタグが異なっており、それぞれの Android アプリストアに適用できるようなデータ収集環境を下記のように構築した。

Google Play: Google Play のデータ収集には商用 API である Google Play Store API [8] を採用した。Google Play Store API はカテゴリによるアプリ ID の検索、アプリのレビューとレーティングの取得及びアプリのバージョンやアプリの平均レビュー等のアプリの本体情報の取得といった多種多様な機能が有する。アプリ ID とレビューのページ番号を HTTP 通信の get メソッドのパラメータとして指定し、Google Play Store API の提供者が用意しておいたサーバーにリクエストを送信した後、アプリのレビュー、レーティング及び評価時間が記載される Json 形式のレスポンスが返ってくる。

サードパーティー: サードパーティーのデータ収集には Splinter [9] を用いたクローラーを実施した。Splinter は URL へのアクセスや HTML タグに囲まれた内容の取得等のブラウザ行動を完全に自動化することができる。Splinter はヘッドレスブラウザ PhantomJS [10] だけではなく、Selenium [11] を通じてブラウザのエミュレーションより上位のレイヤーで各種のブラウザ (Firefox と Chrome) を動かすこともできる。本研究では、ブラウザのエミュレーションではなく Firefox ブラウザを実際に動作して評価情報に書かれているページ番号を変更しながら、HTML のタグにある評価情報とアプリのバージョンやアプリの平均レビュー等のアプリの本体情報を取得した。

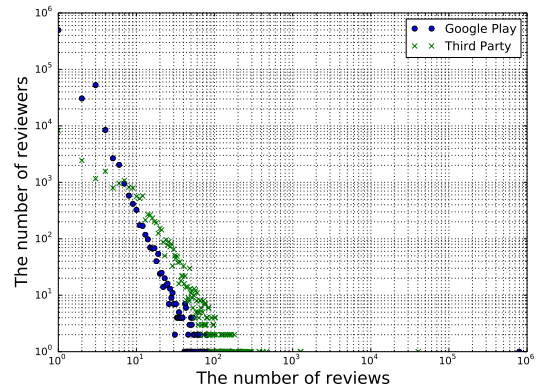


図 3: レビュー数とレビュアー人数の関係

表 2: データセットの内訳

	アプリ数	ユニークレビュアー数	レビュー数
Google Play	26,018	596,858	1,609,173
サードパーティー	19,788	22,747	209,262
合計	45,806	619,605	1,818,435

4.2 データセット

表 2 に示すように、前述したデータ収集環境を利用して Google Play とサードパーティー、合計約 4.5 万のアプリに対応するメタデータを収集した。Google Play とサードパーティーとも各アプリに対して最新の 100 件のレビューを取得している。各アプリのレビュー数は 100 件より小さいケースや一件も存在しないケースがあり、収集したレビューの合計は 170 万件となった。また、複数のレビューを投稿したレビュアーが多数存在し、ID から識別されるユニークなレビュアーの数は約 60 万であった。

Google Play とサードパーティーにおける各レーティング数の分布図は図 2 に示されている。Google Play が 5 階級の分類となっていることに対して、サードパーティーでは 10 階級が採用されている。二つの Android アプリストアとも 5 レーティング付きのレビュー数はそれぞれ全体レビュー数の約 70%、50% を占めており、極めて多いことが分かった。

図 3 はレビュー数とレビュアー人数の関係を示す。二つの Android アプリストアともにべき乗分布にしたがうことがみてとれる。1 件のレビューしか書いていないレビュアーが圧倒的に多いことに対して、一人のレビュアーが大量のレビューを作成していることがわかる。また、2 件以上のレビューを書いたレビュアーはそれぞれ全体の約 11% と 49% である。

表 3: 前処理のデータフロー

手順	アプリストア	アプリ数	ユニークレビュー数	レビュー数
前処理 (3.2 節)	Google Play	18,011	596,741	815,888
	サードパーティー	6,257	21,460	163,948
	合計	24,268	618,201	979,836
類似レビュー候補の抽出 (3.3 節)	Google Play	15,034	325,128	419,537
	サードパーティー	3,029	15,054	64,751
	合計	18,063	340,182	484,288

表 4: Google Play の類似レビューグループの例

アプリ ID	ネーム ID	投稿日時	レビュー	平均レーティング
com.kiwiple.ovjet	Annabelle Castro	April 26, 2014	"Good"	3.9
	Eugene Castro	January 5, 2014	"Good"	
	Eugene Castro	November 10, 2013	"Good"	
	Eugene castro	October 5, 2013	"Good"	
com.nate.android.portalmimi	Eugene Castro	January 10, 2014	"Good"	3.9
	Annabelle Castro	January 9, 2014	"Good"	
	Eugene castro	November 13, 2013	"Verrygood"	
com.btb.minihompy	Annabelle Castro	April 26, 2014	"Good"	3.9
	Eugene Castro	January 10, 2014	"Good"	
	Eugene castro	November 13, 2013	"Good"	

5 解析結果

本章では類似レビューグループの解析結果について述べる。

5.1 前処理と類似レビュー候補の抽出の結果

表 3 に前処理が行われた後のデータセットの内訳を示す。前処理が行われた後、アプリ数、レビュー者の人数、レビュー数に関しては、Google Play は、それぞれ約 30%、0.01%、49%減少したことに対してサードパーティーは約 68%、6%、22%削減された。Google Play のレビュー数が約 49%減った原因としては、デフォルトのネーム ID ('A Google User') によるレビューがたくさん存在するからである。また、サードパーティーの場合は 2 以下のレビューを持つアプリがよく見られているため、サードパーティーのアプリ数は約 68%減少した。類似レビュー候補の抽出を通じてアプリ数、レビュー者の人数、レビュー数は更に半分ぐらい減らすことができた。前処理は高性能フィルターとする機能を果たしたことを確認できた。前述したような二つのステップを通じてデータセットにあるノイズの除去を行った後のものを類似レビューグループ検索の入力として利用する。

5.2 Google Play の解析結果

Google Play では、1 件の類似レビューグループのみ検出したため、その例を表 4 に示す。Google

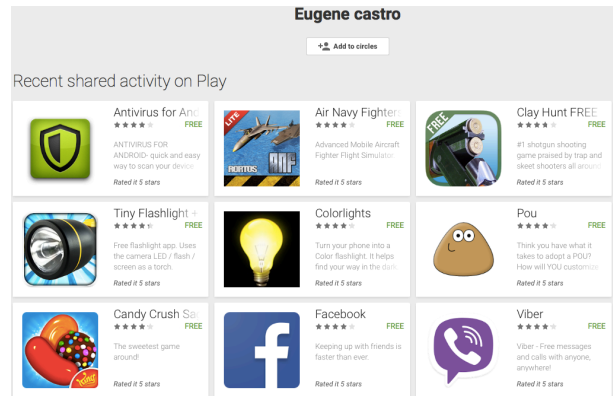


図 4: Google Play の個人アプリ使用の履歴

Play のネーム ID は大文字と小文字を違う文字として扱い、Eugene Castro と Eugene castro は異なるユーザであることが認識されているため、偽アカウントの作成は容易にできる。3 人のレビュー者が 3 件異なるアプリで類似したレビューまたは同様なレビューを作成していることが確認された。また、グループ内の各アプリの平均レーティングは 5 未満であり、全てのレビュー付きのレーティングは 5 となっている。

図 4 に示すように、Google Play では、個人アプリ使用の履歴が公開サービスとして提供されている。その使用履歴にはそのユーザがレビューを書いたアプリの情報や付けたレーティング等が載せられている。Google Play の個人アプリ使用の履歴を用いてグループにいる 3 人のレビューを確認した結果、3 人のレビューは全て使用したアプリに対して 5 のレーティングを付けたことが分かった。さらに、3

表 5: 類似レビューアグループ検索の結果

手順	アプリストア	グループ数	最大グループサイズ	ユニークレビュー数	レビュー数
類似レビューアグループの検索 (3.4 節)	Google Play	1	3	3	10
	サードパーティー	5	14	37	304

表 6: サードパーティーの類似レビューアグループの例

アプリ ID	ネーム ID	作成日時	レビュー	平均レーティング
app1	羅克文	2011-08-16 00:16:02	review1	3.5
	羅克文	2011-08-16 00:15:13	review2	
	羅克文	2011-08-16 00:14:21	review3	
	吉別時代	2011-08-16 00:08:53	review4	
	吉別時代	2011-08-16 00:07:42	review5	
app2	某 a	2011-08-16 00:07:20	review6	4.0
	羅克文	2011-08-10 23:29:31	review7	
	羅克文	2011-08-16 00:13:45	review8	
	羅克文	2011-08-16 00:13:31	review1	
	吉別時代	2011-08-16 00:11:06	review4	
app3	羅克文	2011-08-16 00:10:50	review5	4.0
	羅克文	2011-08-16 00:12:56	review2	
	羅克文	2011-08-16 00:12:12	review3	
	吉別時代	2011-08-16 00:10:10	review6	
	某 a	2011-08-10 23:35:40	review7	
app3	羅克文	2011-08-16 00:19:22	review8	4.0
	羅克文	2011-08-16 00:19:10	review1	
	羅克文	2011-08-16 00:18:59	review2	
	羅克文	2011-08-16 00:18:17	review3	
	吉別時代	2011-08-16 00:04:28	review4	
app3	吉別時代	2011-08-16 00:04:14	review5	4.0
	吉別時代	2011-08-16 00:03:49	review6	
	某 a	2011-08-10 23:32:10	review7	

人のレビューとも表 4 に示されたアプリ以外のアプリ (wechat や Viber 等) に 5 レーティング付きのレビューを投稿することを確認できた。

上記に述べた特徴により、このような類似レビューアグループが生成したレビューとレーティングは不自然だと言える。

5.3 サードパーティーの解析結果

表 5 に示すように、サードパーティーについては 5 件の類似レビューアグループが発見された。各グループのサイズは異なっており、最大のグループサイズは 14 である。全てのグループにおいてはユニークなレビューア人数は 37 人であり、レビュー数は 307 件となっている。全てのレビューの作成日時は 2011 年となっており、アプリのリリース直後の古いものである。

各グループにおけるレビュー数とレビューア人数が多いため、一部を割愛して一つのグループにいる 3 人のレビューのみ表 6 に示す。表 6 に記載されている app1-3 と review1-8 の詳細情報が図 5 に示される。サードパーティーの類似レビューアグループでは、Google play のように 3 件の異なるアプリに同一のレビューを書くだけでなく、単一のレビューアが短期間に複数のレビューを投稿するケースが観測された。例えば、app1 のネーム ID '告別時代' は約 1 分半で 3 回レビュー (review4-review6) を作成

	中国語の原文	日本語の翻訳
app1	他社区	モバイルコミュニティ
app2	游吧	旅行しよう
app3	他社区宠物	ペット
review1	交友, 可以搜索附近的好友, 发起活动, 扩大交友圈, 还可以认识更多国外的朋友。	位置情報を利用して近くにいる人を検索・追加することにより、たくさんの友達を作ることができる。さらに、外国人の友達に出会うことも可能
review2	语音评论, 对好玩的事, 有趣的照片, 建筑, 可以随时发表语音评论, 并向朋友分享	音声レビュー, 気になることに對して何時でも音声によるレビューを作成できる。また、友達にも紹介できる
review3	是目前用户发展最快的移动社区。	現在、発展速度が一番早いモバイルコミュニティ
review4	他社区是一款完全免费的手机交友社区, 具有社区多媒体“免费短信”、位置交友及游戏众多独创功能	無料のSNSソフトであり、無料SMS、友達作り、ゲーム等の機能が持つ。
review5	无需短信费用, 无需打开客户端, 即可和社区好友保持多媒体短信的实时沟通, 不经过任何好友信息	SMS費用がかからないだけでなく、アプリも開く必要はない。何時でも友達からのメッセージを受け取ることができる。
review6	可以在全球任何地方使用“他社区”! 无限享受	世界のどこでも利用できるソフト。
review7	支持	応援する
review8	他社区可以帮助您找到很多有用的信息, 如: 查找您附近的景点、酒店、饭店或标志性建筑。一目了然。	有用な情報をたくさん見つけてくれる。例えば、あなたの周辺にある観光スポット、ホテル、レストラン、一目瞭然。

図 5: レビュー内容と翻訳文

している。さらに、そのレビュー (review4-review6) の内容ではそれぞれ異なっているが、各アプリに対する評価や感想等ではなく、アプリ機能の説明文やアプリの宣伝広告となっていることが確認された。また、app1, app2 と app3 は機能的に異なるアプリであるが、同じ会社が開発したものである。ひとつの推察として、アプリ開発会社が意図的に高いレーティングや良いレビューを投稿した、あるいはそのような評判を操作するサービスを利用した可能性が考えられる。このような類似レビューアグループによって作成されたレビューとレーティングは不自然だと言える。

5.4 解析結果のまとめ

二つの Android アプリストアとも類似レビューアグループ検索手法により見つけられたレビューアたちはそれぞれ 3 人と 37 人存在しており、計 314 件の不自然なレーティングとレビューを確認できた。今回 Google Play については、1 件の類似レビューアグループしか検出されなかった原因はデータセット規模が不十分であると考えられる。データセット規模の制限について 6 章で議論する。

6 議論

本章では、本研究の制限事項と今後の課題について述べる。

6.1 データセットの収集

データ収集の範囲については、今回は各アプリに対して最新のトップ 100 レビューのみ収集している。未収集部分には類似レビューグループが存在する可能性があると考えられる。また、各アプリにおけるレビュー数だけではなく、巡回するアプリ数も増やすことにより、より多くの類似レビューグループを検索できる。そのため、データセット規模の拡大に今後の課題として取り組む。データセット規模の拡大により、たくさんの新しい不自然なレーティングとレビューを発見することが期待できる。データ収集の速度に関しては、毎回同じサーバーに通信してレビューの収集を行うため、並列処理を実行すると、向こうのサーバーに DOS 攻撃だと認識され、通信が切断されてしまうことがある。短期間に大きいデータセットに拡張するため、データ収集の速度の向上も今後の課題となる。

6.2 悪性アプリとの関連性

不自然なレーティングとレビューのあるアプリは悪性ではなかったことが確認された。今回不自然なレーティングとレビューが検出されるアプリ数は少ないため、不自然なレーティングとレビューと悪性アプリとの関連性があるかどうか判断できない。今後、拡大されたデータセットの解析結果に基づいて悪性アプリとの関連性を調査する予定である。

6.3 レーティングとレビューの解析手法

今回は類似レビューグループによる不自然なレーティングとレビューの解析を行った。今後、他の様々なレーティングとレビューの解析手法を試す必要がある。新たなレーティングとレビューの解析手法を利用することで、いままで得られなかった有効な視点を新たに獲得できる可能性がある。

7 まとめ

本研究では、Android アプリストアにおける不自然なレーティングとレビューの状況を明らかにする

目的を達成するために、二つの Android アプリストアからレーティングとレビューを収集した後、提案した類似レビューグループの検索手法に基づいて不自然なレーティングとレビューの解析を行った。その結果、二つの Android アプリストアとも不自然なレーティングとレビューが存在することを確認できた。解析データの大規模化、類似レビューグループの検索手法の改良、他の様々なレーティングとレビューの解析手法の開発及び不自然なレビューと悪性アプリ関係の解明は今後の課題である。

参考文献

- [1] AppBrain, “Android operating system statistics.” <http://www.appbrain.com/stats/>.
- [2] British Broadcasting Corporation, “Fake online customer reviews 'for sale', BBC finds.” <http://www.bbc.com/news/business-33186836>.
- [3] C. Borgelt, “Frequent item set mining,” *Wiley Interdisc. Rev.: Data Mining and Knowledge Discovery*, vol. 2, no. 6, pp. 437–456, 2012.
- [4] N. Jindal and B. Liu, “Opinion spam and analysis,” in *Proceedings of the International Conference on Web Search and Web Data Mining, WSDM 2008, Palo Alto, California, USA, February 11-12, 2008*, pp. 219–230, 2008.
- [5] A. Mukherjee, B. Liu, and N. S. Glance, “Spotting fake reviewer groups in consumer reviews,” in *Proceedings of the 21st World Wide Web Conference 2012, WWW 2012, Lyon, France, April 16-20, 2012*, pp. 191–200, 2012.
- [6] Google Corporation, “Google Play.” <https://play.google.com/store>.
- [7] Nduoa Corporation, “Nduoa.” <http://www.nduoa.com>.
- [8] mashape, “Google Play Store API.” <https://www.mashape.com/maxcanna/google-play-store#multiple-application-details>.
- [9] nikolas, “splinter.” <https://splinter.readthedocs.org/en/latest/>.
- [10] Phantomjs, “Phantomjs.” <http://phantomjs.org>.
- [11] Selenium, “Selenium.” <http://www.seleniumhq.org>.