



Title	Android用アプリケーションの部品グラフを対象としたべき乗則の調査
Author(s)	神田, 哲也; 真鍋, 雄貴; 松下, 誠 他
Citation	情報処理学会第73回全国大会講演論文集. 2011, 2011(1), p. 491-492
Version Type	VoR
URL	<a href="https://hdl.handle.net/11094/50118">https://hdl.handle.net/11094/50118</a>
rights	ここに掲載した著作物の利用に関する注意 本著作物の著作権は情報処理学会に帰属します。本著作物は著作権者である情報処理学会の許可のもとに掲載するものです。ご利用に当たっては「著作権法」ならびに「情報処理学会倫理綱領」に従うことをお願いいたします。
Note	

*The University of Osaka Institutional Knowledge Archive : OUKA*

<https://ir.library.osaka-u.ac.jp/>

The University of Osaka

## Android用アプリケーションの部品グラフを対象としたべき乗則の調査

神田 哲也<sup>†</sup> 真鍋 雄貴<sup>‡</sup> 松下 誠<sup>‡</sup> 井上 克郎<sup>‡</sup><sup>†</sup> 大阪大学基礎工学部情報科学科<sup>‡</sup> 大阪大学大学院情報科学研究科

## 1 はじめに

関数やクラスなどのソフトウェア構成要素のことをソフトウェア部品と呼ぶ。ソフトウェア部品(部品)は、継承、メソッド呼び出しなどで互いに利用することで、ソフトウェアとしての機能を実現している。部品を頂点とし、部品間の利用関係を有向辺で表現したグラフをソフトウェア部品グラフ(部品グラフ)と呼ぶ。部品グラフは、ソフトウェア解析の際に用いられている[2]。

既存研究では、JDKや一般的なJavaアプリケーションの集合を対象とした部品グラフにおいて、出次数の値の大きな部分と入次数がべき乗則に従うことが知られている[4]。しかし、既存研究では調査対象としたソフトウェアのドメインに制限を設けていない。そのため、ソフトウェアのドメインによって部品グラフの性質が変化するのは明らかになっていない。

我々は、ソフトウェアのドメインとして、オペレーティングシステムやミドルウェア、いくつかのアプリケーションを含む携帯端末向けのプラットフォームであるAndroid[1]に着目した。Android用のアプリケーションの多くはJavaで記述されるが、携帯端末特有の機能を活用するために独自のAPIが多数存在しており、一般的なJavaアプリケーションとは違った特徴を持つ。

本研究では、ソフトウェアのドメインが部品グラフの性質に影響を与えるか確認するため、Android上で動作するアプリケーション集合の部品グラフがべき乗則を示すか調査した。調査の結果、Android用アプリケーションの部品グラフもJava一般のアプリケーションの部品グラフと同様のべき乗則を示すことが確認された。

## 2 べき乗則の調査

## 2.1 対象

米Google社が提供するオープンソース・ソフトウェア・ホスティングサービスであるGoogle Codeに登録されているプロジェクトのうち、“Android”のラベルがついた検索結果上位1000件から、Android用アプリケーション約1300個を取得した。また、Android開発者向けサイトからAndroidのソフトウェア開発キット(以下Android SDK)のソースコードを取得した。

## Investigation into Power Law in Software Component Graph of Android Applications

<sup>†</sup>Tetsuya KANDA, <sup>‡</sup>Yuki MANABE, <sup>‡</sup>Makoto MATSUSHITA, <sup>‡</sup>Katsuro INOUE

<sup>†</sup>Department of Information and Computer Sciences, School of Engineering Science, Osaka University <sup>‡</sup>Graduate School of Information Science and Technology, Osaka University

実験では、Android SDKのみ(以下SDK)とAndroid SDKにAndroid上で動作するアプリケーション集合を加えたもの(以下SDK+Apps)の2つのデータセットを対象に解析を行い、Javaソフトウェアを対象とした既存研究[4]でのJDK単体(以下JDK)、JDKに一般的なJavaアプリケーション集合を加えたもの(以下ALL)の実験結果と比較した。既存研究と調査規模を比較した結果を表1に示す。

## 2.2 手法

始めに、部品間の利用関係を解析し、部品グラフを構築する。部品の利用関係の解析には、SPARS/R[3]を用いた。SPARS/RはSPARS-J[2]の後継として開発されたJavaソースコードの部品検索システムであり、検索用データベースに部品を登録する際、部品間の利用関係を解析する。このデータベースから、各部品の利用関係を得る。

次に、得られた利用関係から部品グラフを構築し、各頂点の出次数、入次数の度数分布を求める。そして、分布を視覚的に確認するために累積度数分布を両対数軸でプロットする。

最後に、部品グラフの次数がべき乗則にどの程度従っているかを評価するために、プロットした結果を直線に回帰したときの傾き $a$ 、自由度調整済み寄与率 $R^{*2}$ を求める。自由度調整済み寄与率とは、説明変数が増えるに従い寄与率(重相関係数の2乗)が1に近づくことを補正した値であり、標本値を $Y$ 、 $Y$ の平均を $\bar{Y}$ 、推定値を $\hat{Y}$ 、標本数を $N$ 、説明変数の数を $p$ として以下の式で表される。度数分布が直線に近いほど、自由度調整済み寄与率は1に近くなる。

$$R^{*2} = \frac{\sum_i (Y_i - \hat{Y}_i)^2 / (N - p - 1)}{\sum_i (Y_i - \bar{Y})^2 / (N - 1)}$$

自由度調整済み寄与率が1に近いほど、分布はべき乗則に従っていると言える。分布がべき乗則に従っているならば、両対数軸でプロットした値は直線上に並ぶからである。

表 1: 調査規模の比較

対象	頂点数	辺数	コード行数
SDK	4,437	95,891	0.7M
SDK+Apps	31,109	758,747	3.7M
JDK	11,556	107,198	1.1M
ALL	180,637	1,808,982	14M

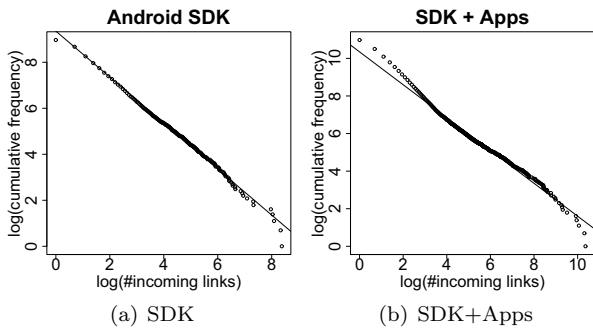


図 1: 入次数

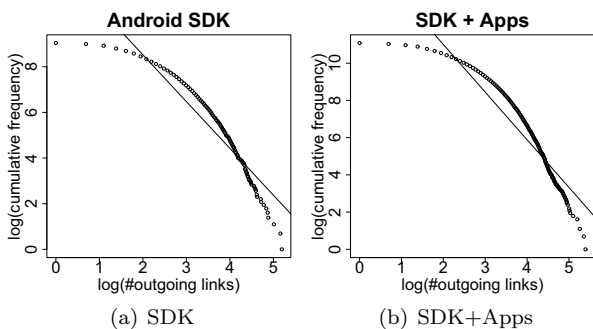


図 2: 出次数

表 2: 入次数：直線回帰時の傾き  $a$  と自由度調整済み寄与率  $R^{*2}$ 

	$a$	$R^{*2}$
SDK	$-0.99 \pm 0.00556$	0.994
SDK+Apps	$-0.87 \pm 0.00510$	0.986
JDK	$-1.11 \pm 0.00866$	0.987
ALL	$-1.02 \pm 0.00145$	0.999

表 3: 出次数：直線回帰時の傾き  $a$  と自由度調整済み寄与率  $R^{*2}$ 

出次数	$a$	$R^{*2}$
SDK	$-2.05 \pm 0.0767$	0.876
SDK+Apps	$-2.56 \pm 0.0860$	0.870
JDK	$-2.10 \pm 0.0818$	0.876
ALL	$-2.66 \pm 0.0693$	0.903

### 2.3 結果

各データセットに対する入次数のプロットとプロットに対する回帰直線を図 1, 出次数のプロットとプロットに対する回帰直線を図 2 に示す。また、直線回帰時の傾き  $a$ , 自由度調整済み寄与率  $R^{*2}$  を Java アプリケーションと比較した結果を表 2, 表 3 に示す。

以下、調査対象とした部品集合ごとに結果を述べる。

### SDK

表 2, 表 3 より入次数, 出次数ともに *JDK* と比べ傾きがやや小さい。しかし, 図 1(a), 図 2(a) より累積度数分布は入次数と出次数の値の大きな部分で直線に近い形状を示している。寄与率も入次数では高く, 出次数では入次数の場合と比べやや低い値となっていることから, *JDK* とほぼ同じ性質をもつ集合であることがわかる。

### SDK+Apps

図 1(b), 図 2(b) から, 入次数はべき乗則に従い, 出次数は次数の大きな部分のみべき乗則に従っている。そのため, このグラフは *ALL* の部品グラフに近い性質を示している。

図 1(b) を見ると入次数の小さい部分が直線から外れている。これは対象が小規模なアプリケーションを多く含むため, 利用される回数が少ない部品が多いからであると考えられる。また, 対数軸のため軸に近い部分はプロットされる点の数が少なく, 誤差が生じやすい。グラフの形状や寄与率が 1 に近いことから見ても, 全体としてはべき乗則によくあてはまっていることが分かる。出次数は, Java 一般の場合と比較してやや曲線に近いものの, 値の大きな部分が直線に近い形をしている。

## 3 まとめ

本研究では, 部品グラフを構築する対象を Android 用アプリケーションに限定しても, Java 一般のアプリケーションと同様に次数分布がべき乗則に従うかを調査した。その結果, Android 用アプリケーションの部品グラフも, Java 一般のアプリケーションの部品グラフと同様に, 入次数は全体が, 出次数も次数の大きな部分がべき乗則に従うことを確認した。

今後の課題として, Android 用アプリケーション以外の他のドメインに対しても同様の調査を行うことや, 利用関係の種類ごとにべき乗則が成り立つかを調査することなどが考えられる。

## 参考文献

- [1] Android. <http://www.android.com/>.
- [2] K. Inoue, R. Yokomori, T. Yamamoto, M. Matsushita, and S. Kusumoto. Ranking significance of software components based on use relations. *IEEE Transactions on Software Engineering*, Vol. 31, No. 3, pp. 213–225, March 2005.
- [3] SPAPS/R. <http://demo.spars.info/r/>.
- [4] 市井誠, 松下誠, 井上克郎. Java ソフトウェアの部品グラフにおけるべき乗則の調査. 電子情報通信学会論文誌 D, Vol. J90-D, No. 7, pp. 1733–1743, (2007-7).