ソースコードの重複度を用いた オープンソースソフトウェアライセンス違反の検出

田 中 健 介[†1] 肥 後 芳 樹[†1] 楠 本 真 二[†1]

本稿では，ソースコードの重複度を用いてライセンス違反を検出可能であるかを調査した結果について報告する．

Detecting open source software licensing violations based on duplicated code information

Kensuke Tanaka,[†1] Yoshiki Higo[†1] and Shinji Kusumoto[†1]

This paper reports the result of an investigation whether duplicated code information is useful for detecting open source licensing violations.

1. はじめに

近年オープンソースソフトウェアは広く普及し，そのソースコードを利用したソフトウェアの開発事例が増加している．オープンソースソフトウェアを利用する際，ライセンスを遵守する必要がある．ライセンス違反の例として著作者の不明記や特定ドキュメント不足などが挙げられる．しかし，ライセンスに違反しているかどうかを発見することは非常に困難である．

ソースコードを利用する場合，コード片（ソースコードの一部），ファイル，ディレクトリ単位でコピー&ペーストが行われており，利用元ソフトウェアと利用先ソフトウェアの間（以降，このような関係を利用関係とする）には類似したコード片（コードクローン）が存在する．よって，コードクローンの割合（以降、この割合を重複度とする）を調査することにより，利用元ソフトウェアを特定できると考えられる．

本稿では，重複度を用いてライセンス違反の検出を目的として行った実験について述べる．

2. 重複度の計算方法

2.1 コードクローン検出法

これまでに様々なコードクローン検出手法が提案されている．今回は様々なソフトウェア間のコードクローンを検出するので，適用対象が大きくなる．そのためスケーラビリティの高い CCFinder[1)] を用いる．

2.2 重 複 度

1 節で述べた重複度を下記のように定義する．

$$\text{重複度} = \frac{CT(A) + CT(B)}{A,\ B \text{ の全字句数}} \quad (1)$$

ただし，A，B はソフトウェア，$CT(A)$，$CT(B)$ は A において B とコードクローンになっている字句数，B において A とコードクローンになっている字句数とする．

3. 実 験

重複度を調査することによってライセンス違反を検出可能かどうか調べる実験を行った．重複度を降順にソートしたとき，上位でライセンス違反ソフトウェアが検出されるかどうかを確認する．

3.1 対 象

C 言語によって開発されたオープンソースソフトウェア 233 個を用いて実験を行った．**表 1** にライセンスとソフトウェア数を示す．総行数は 18,033,563 行，総ファイル数は 45,604 個である．

3.2 手 順

手順 1 表 1 に示したようにオープンソースソフトウェアを準備する．

手順 2 準備したソフトウェアにライセンス違反が既知であるソフトウェアのペアを 2 組混入する．

手順 3 手順 1,2 で準備したソフトウェア群から 2 つ

†1 大阪大学大学院情報科学研究科
Graduate School of Information Science and Technology, Osaka University

のソフトウェアを選択し，CCFinder を用いてソフトウェア間のコードクローンを検出する．その情報を用いて式 (1) で定義した重複度の計算を行う．この計算をソフトウェアの組合せ全てについて行う．

手順 4　異なるライセンスを持つソフトウェア間の重複度のみを降順にソートし，手順 2 で混入したライセンス違反 2 組の重複度の順位を確認する．異なるライセンス間のみを調査するのは，ライセンス違反の多くは異なるライセンスを持つソフトウェアで起こっているからである．

3.3 実 験 環 境

- CPU ： Pentium4 3.00GHz
- メモリ ： 2.00GB
- OS ： Windows XP Professional

計算時間は約 6 日を要した．今回は 233 個のソフトウェアを用いて実験を行ったので，27,028 回 CCFinder を実行し，その出力から重複度の計算を行った．

3.4 結　　果

異なるライセンスを持つソフトウェア間の重複度は 22,721 組得られた．これを降順にソートした結果，2 番目と 15 番目で混入したライセンス違反ソフトウェアが得られた．つまり，重複度の高いソフトウェア間でライセンス違反の可能性があることを確認できた．

また**図 1** にヒートマップを用いて重複度を可視化したものを示す．横軸，縦軸の各マスがソフトウェアを表し，太線がライセンスの境界を示している．このヒートマップでは，色が白いマスが重複度が低く，黒いマスが重複度が高い．このように可視化することにより一目で重複度の高いライセンスやソフトウェアがわかる．

4. 考　　察

4.1 重複度について

式 (1) で定義した重複度では利用関係にあるソフトウェア間でも，一方のソフトウェアの規模に比べもう一方の規模が非常に小さい場合，重複度の値が低くなる場合があった．しかし，規模の小さいソフトウェア

表 1　使用ライセンス

ライセンス	ソフトウェア数
Apache Software License	15
Apache License 2.0	31
Berkeley Software Distribution License	38
GNU General Public License	34
GNU Lesser General Public License	32
IBM Public License	2
MIT License	58
Mozilla Public License	23

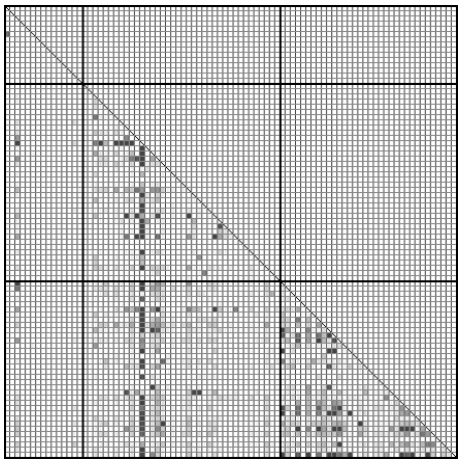

図 1　重複度ヒートマップ

においてコードクローンになっている字句の割合を考えることで対策が可能である．

4.2 重複度の閾値について

本稿ではどの程度の重複度のソフトウェアまでを調査する必要があるか確認できなかった．今回の実験では利用関係の有無に関係なく重複度を計算，降順にソートした．閾値の決定方法を考察するためには，ライセンス条項に従ったドキュメントを参照して利用関係にあるソフトウェアのみを取得し，重複度を調査する必要があると考えられる．

5. ま と め

異なる 2 つのオープンソースソフトウェア間の重複度を用いて，ライセンス違反を検出できるかどうかを調査した．今後の課題として，本調査結果からライセンス違反検出手法をまとめていくこと，並列計算を用いた計算時間短縮が挙げられる．

謝辞　本研究は，文部科学省「次世代 IT 基盤構築のための研究開発」（研究開発領域名：ソフトウェア構築状況の可視化記述の開発普及）の委託に基づいて行われている．また，文部科学省科学研究費補助金基盤研究 (C)(課題番号:20500033)，および若手研究 (スタートアップ)(課題番号:19800022) の助成を得て行われている．

参 考 文 献

1) Kamiya, T., Kusumoto, S. and Inoue, K.: CCFinder: A multi-linguistic token-based code clone detection system for large scale source code, IEEE Trans. Softw. Eng., Vol.28, No.7, pp.654-670 (2002)