人工知能学会第2種研究会資料
SIG-KST-2013-01-01(2013-07-31)

既存プログラム関連資産共有のための機械学習の活用

Applications of Machine Learning Techniques for the Program Asset Sharing

笈田佳彰[1]　岡田伊策[1]　齋藤稔[1]　田邊誠一郎[2]　彭聯凱[2]　小峰正美[2]　稗方和夫[3]

Yoshiaki OIDA[1], Isaac OKADA[1], Minoru SAITO[1], Seiichiro TANABE[2], Liankai PENG[2], Masami KOMINE[2], and Kazuo HIEKATA[3]

[1]富士通株式会社
[1]FUJITSU LIMITED.
[2]株式会社富士通システムズ・イースト
[2]Fujitsu Systems East Limited
[3]東京大学
[3]THE UNIVERSITY OF TOKYO

アブストラクト：IT 企業では過去に開発した大量のプログラム関連資産を蓄積している。担当者や同一組織内において資産の再利用はなされているものの、組織を跨いで資産を把握・共有・再利用できている例は少ない。本稿では、プログラム関連資産共有のための取り組みの全貌を紹介するとともに、プログラム関連資産の一部である設計書を基に機械学習を行い、自動的にプログラム関連資産を可視化する手法を提案する。これにより、既存プログラム関連資産の正確な現状把握や、客観的な根拠に基づくプログラム開発可否の適切な判断支援を狙う。また、IT 企業の一部の既存プログラム関連資産に対して、本可視化手法を試験活用した結果を評価する。

1. 緒言

IT 企業では過去に開発した大量のプログラム関連資産を蓄積している。これらの資産は担当者や同一組織内において再利用はなされているものの、組織を跨いで資産を把握・共有・再利用できている例は少ない。

三上ら[１]は、これらの問題に対して様々なプラクティスを提案しているが、その他の原因として考えられる、再利用可能なプログラム関連資産の認知度が低いことと、資産が適切にメンテナンスや機能拡張されないことに関する施策は弱い。一方で、企業における文書については、利活用促進のためにクラスタリングやクラシフィケーション等の機械学習技術が応用されている[２]。

そこで本研究では、プログラム関連資産の利活用促進のために機械学習を活用することを考える。既存のプログラム関連資産を自動で整理し、各組織が保持する資産の一覧を提示することで、資産の組織間の認知度向上と既存資産の正確な現状把握を狙う。同時に、客観的な根拠に基づく再利用しやすいプログラム開発可否、メンテナンス可否等の適切な判断支援を考える。

本稿では、プログラム関連資産共有のための取り組みの全貌を紹介するとともに、プログラム関連資産の一部である設計書を基に機械学習を行い、自動的にプログラム関連資産を可視化する手法を提案する。また、IT 企業の一部の既存資産に対して、本可視化手法を試験活用した結果を評価する。

以下、2 章にて研究対象について説明する。続く 3 章、4 章でプログラム関連資産共有に適した可視化の形態と可視化手法について提案し、5 章でケーススタディを行い、提案手法を評価する。6 章で考察を述べ、終章にて本稿をまとめ、今後の課題に触れる。

2. プログラム関連資産

本稿におけるプログラム関連資産とは、開発されたソースファイル、バイナリファイル等のプログラム類、そのプログラムの設計書や仕様書、テスト仕様書等のドキュメント類に加えて、その他関連する種々のデータ等を含む資産を意味する（図 1）。

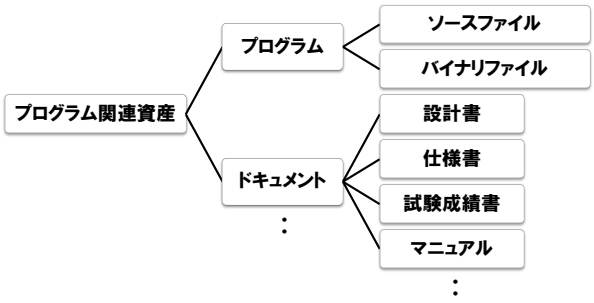

図 1 プログラム関連資産

3. ソリューション機能二次元分布図

3.1. ソリューション

本稿で扱う「ソリューション」とは、業務上の問題点の解決や要求の実現を行うための情報システムの一種であり、特定の業種、事業規模において再利用可能な情報システムを指す。「ソリューション」は 100 以上の「機能」から構成されるケースが多い。また、「ソリューション」を再利用し、特定の顧客に適用する場合は、一部をカスタマイズすることも多い。具体的には、「物流ソリューション」や「人事ソリューション」等が挙げられる。

3.2. 機能

本稿で扱う「機能」とは、プログラムとドキュメントを保持するプログラム関連資産（2 章参照）の最小単位を意味し、システムやサブシステムよりも小さい粒度のプログラム関連資産の総称である。具体的には、「債務残高照会機能」や「ログイン ID 受信機能」等が挙げられる。「機能」が集まり、連携することで、3.1 の「ソリューション」が形成される。

3.3. ソリューション機能二次元分布図概要

プログラム関連資産を共有する手段として、縦軸に「業務カテゴリ」（本稿における「業務カテゴリ」とは、業務を区分する分類を意味する）、横軸をソリューション ID とした、二次元の表に対して、「機能」を分布させた図を考える（図 2）。○は「機能」が 1 つ以上存在することを表す。

図 2 の例でいえば、ソリューション 1 には「発注」「受注」の「業務カテゴリ」に属する「機能」は 1 つ以上存在するが、「見積り」等に関する「機能」は存在しないことを意味する。

組織横断的に既存プログラム関連資産が網羅的にマッピングすることができれば、資産状況の確実な把握が可能となる、また、「ソリューション」間の「機能」の重複や差分から、次期プログラム開発可否の判断が容易かつ客観的となる。

業務カテゴリ ＼ ソリューションID	ソリューション1	ソリューション2	ソリューション3	‥	ソリューションM
予算				‥	
生産計画		○		‥	
発注	○		○	‥	○
見積り		○		‥	
原価管理		○		‥	○
売変管理				‥	○
在庫管理		○		‥	○
債権管理				‥	
債務管理		○		‥	○
：	：	：	：	‥	：
売上				‥	○
受注	○		○	‥	○

図 2 ソリューション機能二次元分布図の例

3.4. ソリューション機能二次元分布図生成のための要件

ソリューション機能二次元分布図生成のための要件として、以下の 2 点を考える。

①画一的な基準によって分類され、分布図の軸のばらつきが小さいこと

②逐次新規「機能」が追加されるため、分布図の生成コストが小さいこと

そこで本稿では、計算機により自動で「機能」を分類し、自律的にソリューション機能二次元分布図がアップデートされることを狙う。

4. 機械学習を用いた可視化手法

4.1. 概要

機械学習を活用し、自律的に可視化される手法を提案する。具体的には以下の 2 つのタスクを適切なタイミングで実施することを考える。

①既存資産を用いた「業務カテゴリ」の生成

既存資産を用いてゼロから「業務カテゴリ」の集合を生成する際に実施する。（2 年に 1 回程度、大幅な「機能」の見直しの際）

②新規資産登録時における自動カテゴリ割当て

①で生成されて「業務カテゴリ」の集合に対して、新規「機能」をマッピングする際に実施する。（新規「機能」追加時毎）

4.2. 本手法で扱うプログラム関連資産

本手法においては図 3 上図のように、3.2 で述べた「機能」を対象とし、「機能」の中でも、プログラム類を保持せず、設計書を保持する「機能」のみを対象とした。図 3 下図のように 1 つの「機能」に対して、その「機能」を開発するための複数個の設計書が紐付くような構造をとる。

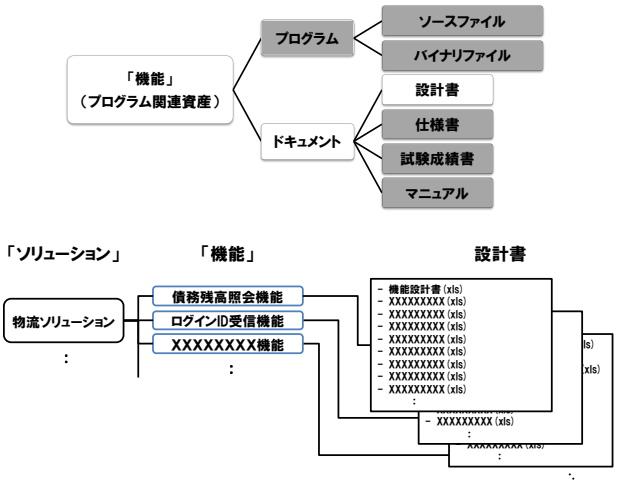

図 3 本手法の対象とするプログラム関連資産（上図：本手法の対象と図 1 との対応関係，下図：分類対象の構造）

4.3. 既存資産を用いた「業務カテゴリ」の生成

図 5 に既存資産を用いた「業務カテゴリ」の生成の流れを示す。

①ベクトルの生成

「業務カテゴリ」生成のためのクラスタリングを行う際には、「機能」に紐付く設計書群からベクトルを作成する必要がある。具体的なベクトルの作成手順を図 4 に示す。設計書群をテキストに変換し、そのテキストについて形態素解析を行う。分割された形態素（素性）の出現頻度をカウントすることにより、素性の数だけ次元を持つベクトルを生成する。なお、形態素解析エンジンには lucene-gosen[３]を使用した。

特徴量として使用する形態素（素性）を選定する際、平ら[４]の研究では、Support Vector Machine による識別の際は、品詞だけを指定し、特にストップワードを用いずに多くの用語を用いた方が分類精度の向上が図れると書かれている。しかし、教師なし学習の一種であるクラスタリングは、教師あり学習と違い、素性が多ければ多いほど良い結果が出力されるというわけではない。

そこで、業務的に意味のありそうなカテゴリが出力されるように、業務に関する素性だけを用いる必要がある。その際の工夫として本稿では、以下の 2 点を考える。

○サ変接続名詞のみの利用

名詞という制限だけでは、素性の数が膨大になってしまうため、名詞の中でも、業務に関係する可能性の高いサ変接続名詞を利用する。

○ストップワード登録

業務と無関係な用語について、業務担当者にヒアリングを実施し、ストップワードとして登録する。

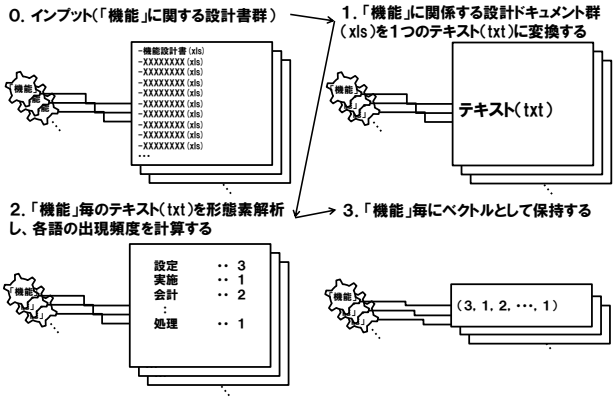

図 4 設計書からベクトルを生成する流れ

②クラスタリング

①で生成されたベクトルを用いてクラスタリングを行う。クラスタリングの方法として、最も一般的な手法である K-means 法を利用した。ただし、特定のクラスタ内の「機能」数が閾値より多い場合は再度、K-means 法を用いてクラスタリングを行う。このことにより、同一クラスタ内における「機能」数の一定化を図る。

③「業務カテゴリ」の設計・作成

クラスタリングされた結果を元に、適切な「業務カテゴリ」を命名する。本来は人が見て適切なカテゴリ名を付与するが、自動化のため「機能」の機能名に含まれる語の中で、クラスタ内で頻出の語 3 つ抽出することで、業務カテゴリ名とし、各「機能」に付与する。

④ソリューション機能二次元分布図の作成

分類対象の「機能」はどの「ソリューション」を構成する「機能」であるかという情報は保持している。それに加えて、各「機能」はクラスタリングを経て、業務カテゴリ名の情報を保持する。よってこの二つの情報を X 軸、Y 軸として二次元分布図を形成することが可能となる。例えば、ソリューション

4 を構成する「機能」であり、「業務カテゴリ」3 という業務カテゴリ名が付与されていれば、X=ソリューション 4, Y=「業務カテゴリ」3 の成分が○となり、ソリューション 4 が「業務カテゴリ」3 の「機能」を含むことが直観的にわかる。

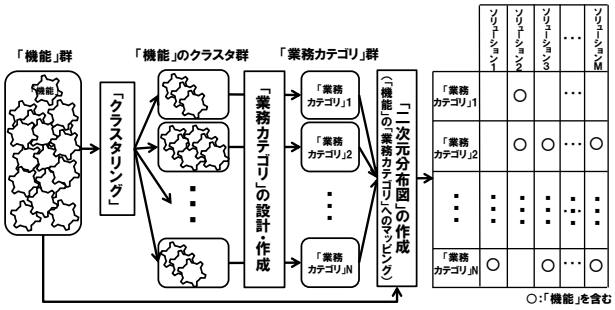

図 5 既存プログラム関連資産を用いた「業務カテゴリ」生成の流れ

4.4. 新規資産登録時における自動カテゴリ割当て

図 6 に新規資産登録時における自動カテゴリ割当ての流れを示す。

①教師データの作成

4.3 において、生成された「業務カテゴリ」に含まれる「機能」は正しい分類されているという前提の下、それらの「機能」群と対応する「業務カテゴリ」を教師データとして用いる。

②SVM の学習

「機能」の設計書群を 4.3 の①と同様の手続きにより素性ベクトルを生成し、その素性ベクトルを用いて、Support Vector Machine (以下、SVM)を学習させる。

SVM はテキスト分類において非常に高い分類能力を持つ[5][6]。具体的には、(式 1) で表される最小化問題を解く。またカーネル関数として (式 2) で表されるガウスカーネルを用いる。本稿においては、分類の対象とする各「機能」が複数のカテゴリを持ち得ることから SVM を 1vs1 法により組み合わせた多クラス分類を行う。その際、C と γ に関するグリッドサーチ (パラメータの変更と交差検定を繰り返し、適切な値を探索する手法) により、パラメータの調整を行う。なお、実装には LIBSVM(A Library for Support Vector Machines)[7]を使用した。

$$\min \quad \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + C\sum_{i=1}^{l}\xi_i \qquad \text{(式 1)}$$

$$\text{subject to } y_i(\boldsymbol{w}^T\phi(\boldsymbol{x_i}) + b) \geq 1 - \xi_i,$$
$$\xi_i \geq 0\,.$$

$$K(\boldsymbol{x_i}, \boldsymbol{x_j}) = \exp\left(-\gamma\|\boldsymbol{x_i} - \boldsymbol{x_j}\|^2\right) \ , \gamma > 0 \qquad \text{(式 2)}$$

③カテゴリ付与

新規「機能」が追加された際は、その「機能」に関する設計書群を 4.3 の①と同様の手続きにより素性ベクトルを生成する。4.4 の②で学習させた SVM に対して、生成された素性ベクトルをインプットデータとして入力し、SVM による判別結果 (「業務カテゴリ」) をその「機能」に自動で付与する。

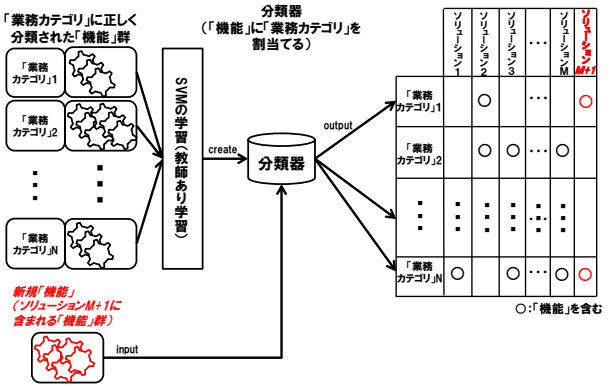

図 6 新規プログラム関連資産登録時における自動カテゴリ割当ての流れ

5. ケーススタディ

5.1. 概要

本ケーススタディにおいては、クラスタリングによる、「業務カテゴリ」の生成に関するケーススタディと、SVM により他クラス分類器を作成し、「業務カテゴリ」の自動付与を行うケーススタディの二つを実施する。

5.2. 業務カテゴリ軸の生成に関する評価

本ケーススタディでは、4 つの「ソリューション」に含まれる 447 の「機能」を対象とした。「ソリューション」別機能数内訳を表 1 に示す。

表 1 「ソリューション」別機能数内訳

ソリューション名	機能数
ソリューション 1	56
ソリューション 2	175
ソリューション 3	129
ソリューション 4	87
計	447

4.3 の手法より 70 のクラスタ群を持つ結果を得た。また、素性ベクトルの次元数は 651 となった。

これらのクラスタリング結果を、全ての「機能」について、業務内容に詳しい有識者 (2 名) に評価

してもらい、×、△、○のいずれかのラベルを付与した。1 クラスタに対して、1「機能」しか含まないクラスタは「その他」というラベルを付与した。評価基準は以下のとおりである。

×：その「機能」が属するクラスタ内で、確実に間違っていると断言できる「機能」。

△：間違っているとは断言できないが、違和感がある「機能」。

○：△、×以外（違和感を感じない）の「機能」。

有識者（2 名）によって「機能」に対して付与されたラベルの内訳を表 2 に示す。また、クラスタ内から「2 名ともが違和感がある『機能』(つまり、××、×△、△△と評価された『機能』)」を除くために必要な修正量を表 3 に示す。

表 2 「機能」に対して付与されたラベル内訳

	Aさん		Bさん	
○	282	63.1%	351	78.5%
△	69	15.4%	51	11.4%
×	75	16.8%	24	5.4%
その他	21	4.7%	21	4.7%
計	447	100%	447	100%

表 3 クラスタ正常化のために必要な修正量

A.修正不要	27	55.1%
B.2割以下の修正	9	18.4%
C.5割以下修正	11	22.4%
D.それ以上の修正	2	4.1%
その他(合計に数えない	(21)	
計	49	100.0%

「機能」単位で集計した際に、○の付与された機能数は両社共に 6 割を超え、クラスタ単位で集計した際に修正が不要なクラスタが半数を超え、また 2 割以下の修正を許せば、正常とみなせるクラスタが、7 割以上を占める。自動的に計算機によって分類した場合でも違和感のないクラスタが生成できることがわかった。

また、有識者に比較的よく分類されていると評価されたクラスタ（10「機能」以上を保持し、修正が 1 割以下のクラスタ）の特徴としては、クラスタ内で最も頻繁に出現する素性が 70 クラスタ中において固有の語ばかりであった。

5.3. 自動カテゴリ割当てに関する評価

ソリューション機能二次元分布図に新規「機能」を追加する際に、正しい「業務カテゴリ」に分類されるかどうかの評価を行う。教師データ（「機能」と適切な「業務カテゴリ」が対になっているデータ）として、前節のクラスタリング結果（各「機能」がクラスタに分類されている）を用いた。ただし、1 クラスタに 1「機能」しか含まないクラスタは、カテゴリを生成できていないと判断し、教師データから排除した。

教師データをランダムに 8 等分し、交差検定を行いった結果、69.0%となった。その際、SVM の主要パラメータである C とγについてチューニングを行い、$C=2^{17.5}$、$\gamma=2^{-19}$ の場合に最大となった（図 7）。よって、70%弱の精度で、新規「機能」に対して適切な「業務カテゴリ」を付与できること意味する。

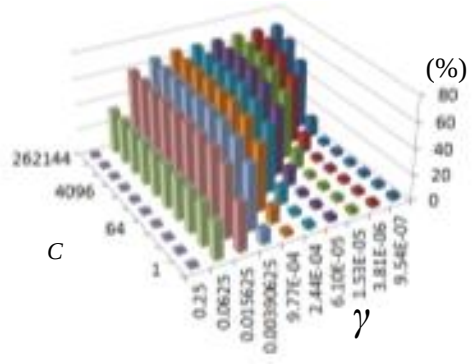

図 7 グリッドサーチ（C, γ）の結果

6. 考察

6.1. 「業務カテゴリ」生成時の業務知識の扱い

本手法における「業務カテゴリ」生成の段階で、業務知識をストップ―ワードという形で組み込んだ。ストップワードに用語を追加するということは、「その用語に関した分類をさせない」というネガティブな方向の知識の使い方である。本来、業務担当者は、「このような観点で分類させたい」といったポジティブな方向に知識を使うことの方を先に考える。

今後は、登録した用語についてはサ変接続名詞ではなくとも素性としての使用を可能にすることや、素性ごとに重要度を設定し、業務上の重要語を認識できるようにすることを考える。また、同時に様々な業務担当者の知識を組み込みやすいような仕組みを考える必要がある。

6.2. 運用面について

本手法では、新規「機能」追加時には既存の「業務カテゴリ」を付与する。また、「業務カテゴリ」生成の際には、既存の資産から全ての「業務カテゴリ」を生成する。そのためこれらの二つの方法を組み合

わせることで、業務内容の変化に対応する必要がある。本稿では、2 年に一回程度クラスタリングを実施し「業務カテゴリ」を追加することを想定しているが、実際に適切なタイミングはわからない。今後、運用を始めていく中で、「業務カテゴリ」の使いやすさの定量的な指標を設定し、それが閾値を下回った際にクラスタリングを実施する等の運用ルールを決めていく必要がある。

7. 結言

企業内に蓄積されている過去に開発されたプログラム関連資産の共有のための可視化手法を提案した。具体的には、クラスタリングを用いて分類の軸（カテゴリの種類）を自動生成し、SVM を用いて生成された軸に対して自動的に「機能」をマッピングする手法である。有識者へのヒアリングを通じて得られた「分類に寄与して欲しくない用語」をストップワードに加えることで、クラスタリング結果を制御できるような工夫を施した。

本稿においては、既存プログラム資産を可視化する上での機械学習の結果に関する評価は行った。しかし、ソリューション機能二次元分布図が組織間の資産共有に寄与するかどうかの評価はできていない。今後、実業務に対する効果についても実証検証を進め、報告する。

8. 参考文献

[1] 三上徹也, 高橋辰徳, 中山清喬: IBM プロフェッショナル論文 組織的アセット再利用サイクルの推進アプローチ. Provision, No.53, pp. 56-62, (2007)

[2] 後藤和之, 平博司, 宮部泰成: 企業の情報と知識の利活用を促進する対話型文書分類システム, 東芝レビュー, Vol. 65, No. 2, pp. 60-63, (2010)

[3] Lucene-gosen, Availabled at <http://code.google.com/p/lucene-gosen/> Accessed on: Feb 18th 2013

[4] 平博順 , 春野雅彦 : Support Vector Machine によるテキスト分類における属性選択, 情報処理学会論文誌, Vol. 41, No. 4, pp. 1113-1123,(2000)

[5] Dumas, S., Platt, J., Heckerman, D. and Sahami, M.: Inductive Learning Algorithms and Representations for Text Categorization, Proc., 7th International Conference for Information and Knowledge Management, (1998)

[6] Joachims, T.: Text Categorization with Support Vector Machines, Proc., 10th European Conference on Machine Learning (ECML), (1998)

[7] LIBSVM -- A Library for Support Vector Machines, Availabled at <http://www.csie.ntu.edu.tw/~cjlin/libsvm/> Accessed on: Feb 18th 2013