JFPUG 2017年度 第3回会合
2017年10月13日　於：主婦会館プラザエフ

ソフトウェア開発に関する スモールデータの分析技術

岡山大学大学院自然科学研究科
門田 暁人
monden@okayama-u.ac.jp

本日の話

- ソフトウェア開発データは，
  - スモールデータである．サンプルサイズ＝数十～数百
  - データ項目は多い（場合がある）．
  - 欠損値が多い．
  - 外れ値・異常値が多い．
  - 性質の異なる個体が混在している．
- スモールデータ向けの分析技術
  1. 矛盾ケースの除去
  2. オーバーサンプリング
  3. 2ステップ予測
  4. 原型分析
- 事例

技術1

矛盾ケースの除去*

*P. Phannachitta, J. Keung, K.E. Bennin, A. Monden, K. Matsumoto: Filter-INC: Handling Effort-Inconsistency in Software Effort Estimation Datasets. APSEC 2016: 185-192

データ例

- Kemererデータセット[1]より抜粋

_ID	Language	Hardware	Duration	KSLOC	AdjFP	EffortMM
1	1	1	17	253.6	1217.1	287.0
2	1	2	7	40.5	507.3	82.5
3	1	3	15	450.0	2306.8	1107.3
4	1	1	18	214.4	788.5	86.9
5	1	2	13	449.9	1337.6	336.3
6	1	4	5	50.0	421.3	84.0
7	2	4	5	43.0	99.9	23.2
8	1	2	11	200.0	993.0	130.3
9	1	1	14	289.0	1592.9	116.0
10	1	1	5	39.0	240.0	72.0
11	1	1	13	254.2	1611.0	258.7

[1] G. A. Liebchen and M. Shepperd: Data Sets and Data Quality in Software Engineering, PROMISE'08, pp.39-44, 2008.

データ例

_ID	Language	Hardware	Duration	KSLOC	AdjFP	EffortMM
1	1	1	17	253.6	1217.1	287.0
2	1	2	7	40.5	507.3	82.5
3	1	3	15	450.0	2306.8	1107.3
4	1	1	18	214.4	788.5	86.9
5	1	2	13	449.9	1337.6	336.3
6	1	4	5	50.0	421.3	84.0
7	2	4	5	43.0	99.9	23.2
8	1	2	11	200.0	993.0	130.3
9	1	1	14	289.0	1592.9	116.0
10	1	1	5	39.0	240.0	72.0
11	1	1	13	254.2	1611.0	258.7

データ例

データをじっと眺めていると. . .

_ID	Language	Hardware	Duration	KSLOC	AdjFP	EffortMM
1	1	1	17	253.6	1217.1	287.0
2	1	2	7	40.5	507.3	82.5
3	1	3	15	450.0	2306.8	1107.3
4	1	1	18	214.4	788.5	86.9
5	1	2	13	449.9	1337.6	336.3
6	1	4	5	50.0	421.3	
7	2	4	5	43.0	99.9	
8	1	2	11	200.0	993.0	
9	1	1	14	289.0	1592.9	116.0
10	1	1	5	39.0	240.0	72.0
11	1	1	13	254.2	1611.0	258.7

開発工数が
倍以上違う

値ほとんど同じ

データ例

_ID	Language	Hardware	Duration	KSLOC	AdjFP	EffortMM
1	1	1	17	253.6	1217.1	287.0
2	1	2	7	40.5	507.3	82.5
3	1	3	15	450.0	2306.8	1107.3
4	1	1	18	214.4	788.5	86.9
5	1	2	13	449.9	1337.6	336.3
6	1	4	5	50.0	421.3	
7	2	4	5	43.0	99.9	
8	1	2	11	200.0	993.0	
9	1	1	14	289.0	1592.9	116.0
10	1	1	5	39.0	240.0	72.0
11	1	1	13	254.2	1611.0	258.7

開発工数が倍以上違う

値ほとんど同じ

こんなデータセットでモデル化せよ，というのがそもそも無理な話

データ例

_ID	Language	Hardware	Duration	KSLOC	AdjFP	EffortMM
1	1	1	17	253.6	1217.1	287.0
2	1	2	7	40.5	507.3	82.5
3	1	3	15	450.0	2306.8	1107.3
4	1	1	18	214.4	788.5	86.9
5	1	2	13	449.9	1337.6	336.3
6	1	4	5	50.0	421.3	84.0
7	2	4	5	43.0	99.9	23.2
8	1	2	11	200.0	993.0	130.3
~~9~~	~~1~~	~~1~~	~~14~~	~~289.0~~	~~1592.9~~	~~116.0~~
10	1	1	5	39.0	240.0	72.0
~~11~~	~~1~~	~~1~~	~~13~~	~~254.2~~	~~1611.0~~	~~258.7~~

両方除去すると，データが減りすぎるため，かえって予測モデルの性能が下がる．

データ例

_ID	Language	Hardware	Duration	KSLOC	AdjFP	EffortMM
1	1	1	17	253.6	1217.1	287.0
2	1	2	7	40.5	507.3	82.5
3	1	3	15	450.0	2306.8	1107.3
4	1	1	18	214.4	788.5	86.9
5	1	2	13	449.9	1337.6	336.3
6	1	4	5	50.0	421.3	84.0
7	2	4	5	43.0	99.9	23.2
8	1	2	11	200.0	993.0	130.3
9	1	1	14	289.0	1592.9	116.0
10	1	1	5	39.0	240.0	72.0
11	1	1	13	254.2	1611.0	258.7

No.11だけを除去してみると？

データ例

_ID	Language	Hardware	Duration	KSLOC	AdjFP	EffortMM
1	1	1	17	253.6	1217.1	287.0
2	1	2	7	40.5	507.3	82.5
3	1	3	15	450.0	2306.8	1107.3
4	1	1	18	214.4	788.5	86.9
5	1	2	13	449.9	1337.6	336.3
6	1	4	5	50.0	421.3	84.0
7	2	4	5	43.0	99.9	23.2
8	1	2	11	200.0	993.0	130.3
9	1	1	14	289.0	1592.9	116.0
10	1	1	5	39.0	240.0	72.0
11	1	1	13	254.2	1611.0	258.7

値が近い

開発工数が倍以上違う

今度は，No.1とNo.9の間で矛盾が生じてしまう．
No.11を削除したのは間違いだった．

データ例

_ID	Language	Hardware	Duration	KSLOC	AdjFP	EffortMM
1	1	1	17	253.6	1217.1	287.0
2	1	2	7	40.5	507.3	82.5
3	1	3	15	450.0	2306.8	1107.3
4	1	1	18	214.4	788.5	86.9
5	1	2	13	449.9	1337.6	315.8
6	1	4	5	50.0	421.3	84.0
7	2	4	5	43.0	99.9	23.2
8	1	2	11	200.0	993.0	130.3
9	1	1	14	289.0	1592.9	116.0
10	1	1	5	39.0	240.0	72.0
11	1	1	13	254.2	1611.0	258.7

値が近い

値が近い

No.9を削除した場合は？
これだと矛盾は生じない．

提案方法：Filter-INC

1. FISI[1]やTEAK[2]などを用いて矛盾ケースiを見つける．
2. ケースiと矛盾するケース集合Cを見つける．
3. Cを全て除去してから再度1.を実施し，ケースiが矛盾ケースとならないならば，ケースiを残しておく．さもなければケースiを削除候補とする．
4. 全ての矛盾ケースが処理されるまで1.に戻る．
5. 削除候補となったケースを全て削除する．

[1] T. K. Le-Do, K.-A. Yoon, Y.-S. Seo, D.-H. Bae: Filtering of Inconsistent Software Project Data for Analogy-Based Effort Estimation. COMPSAC 2010: 503-508

[2] E. Kocaguneli, T. Menzies, A. Bener, J. Keung: Exploiting the Essential Assumptions of Analogy-Based Effort Estimation. IEEE Trans. Software Eng. 38(2): 425-438 (2012)

評価実験

Dataset	Model	Total losses			
		FISI	Filter-INC (FISI)	TEAK	Filter-INC (TEAK)
Kemerer	CART	0	0	7	0
	ABE0-1NN	0	0	1	0
	ABE0-5NN	1	0	11	0
Cocomo81 -e	CART	8	1	2	0
	ABE0-1NN	0	0	2	0
	ABE0-5NN	7	1	2	0
Cocomo81 -so	CART	8	1	2	0
	ABE0-1NN	0	0	2	0
	ABE0-5NN	1	1	2	0
Desharnais -Cobols	CART	2	0	2	0
	ABE0-1NN	0	0	2	0
	ABE0-5NN	6	0	1	0
Desharnais -4GL	CART	0	0	0	0
	ABE0-1NN	0	0	0	0
	ABE0-5NN	0	0	0	0
Nasa93 -c1	CART	8	1	14	0
	ABE0-1NN	0	0	12	0
	ABE0-5NN	7	7	14	0
Nasa93 -c2	CART	1	1	1	1
	ABE0-1NN	8	7	2	0
	ABE0-5NN	8	1	1	0
Nasa93 -c5	CART	0	0	0	0
	ABE0-1NN	0	0	0	2
	ABE0-5NN	0	0	0	0

矛盾ケースを全て除去する従来法(FISI, TEAK)よりもFilter-INCの方がよい.

技術2

オーバーサンプリング*

*K. E. Bennin, J. Keung, P. Phannachitta, A. Monden, S. Mensah: MAHAKIL: Diversity based oversampling approach to alleviate the class imbalance issue in software defect prediction. IEEE Trans. Software Engineering (to appear).

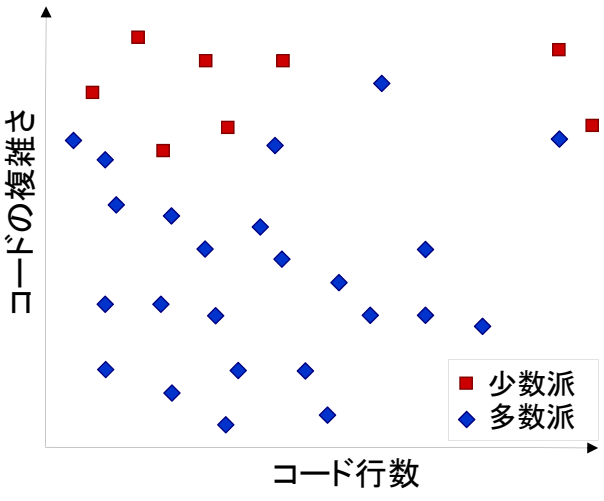
概要

- 機械学習においてデータの少なさはしばしば問題となる.
  - モデル構築時に，データセット中の多数派に支配される.
  - 2群の判別の場合，少数派の群をうまく判別できなくなる.
- オーバーサンプリング
  - データを人工的に追加して，データセット中の偏りをなくす.

ランダムオーバーサンプリング

- 少数派ケースをランダムに複製することで，少数派ケースの数を増加させる.

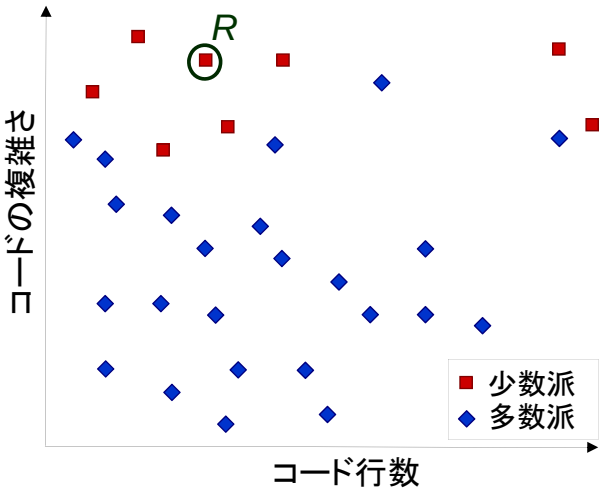
1. ケースの選択
   - 少数派ケースをランダムに1つ選択する.
2. ケースの複製・追加
   - 手順1で選択されたケースを複製し，新たなケースとしてデータセットに追加する.



ランダムオーバーサンプリング

- 少数派ケースをランダムに複製することで，少数派ケースの数を増加させる．

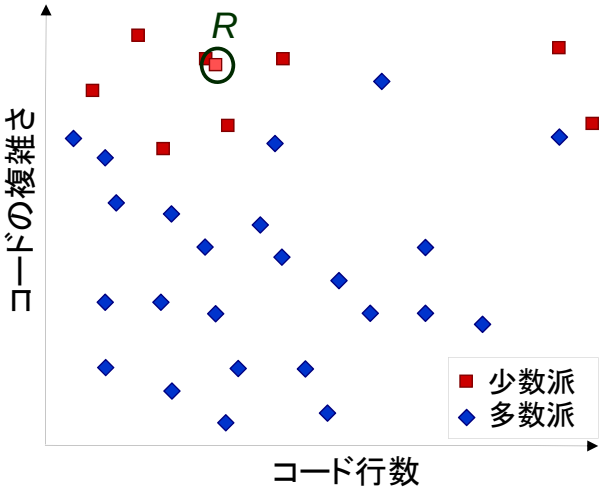
1. ケースの選択
   - 少数派ケースをランダムに1つ選択する．
2. ケースの複製・追加
   - 手順1で選択されたケースを複製し，新たなケースとしてデータセットに追加する．



ランダムオーバーサンプリング

- 少数派ケースをランダムに複製することで，少数派ケースの数を増加させる．

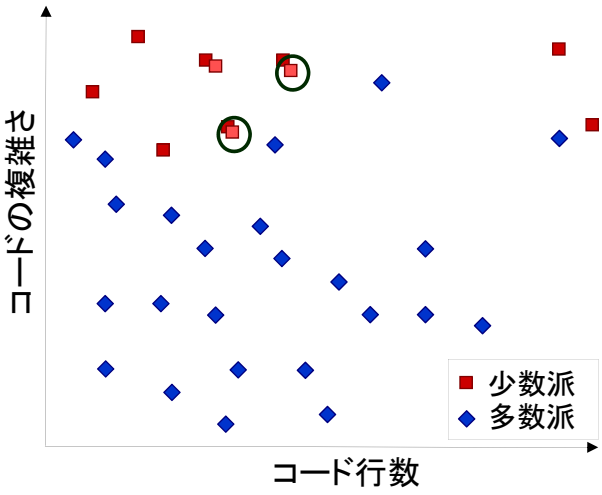
1. ケースの選択
   - 少数派ケースをランダムに1つ選択する．
2. ケースの複製・追加
   - 手順1で選択されたケースを複製し，新たなケースとしてデータセットに追加する．



ランダムオーバーサンプリング

- 少数派ケースをランダムに複製することで，少数派ケースの数を増加させる．

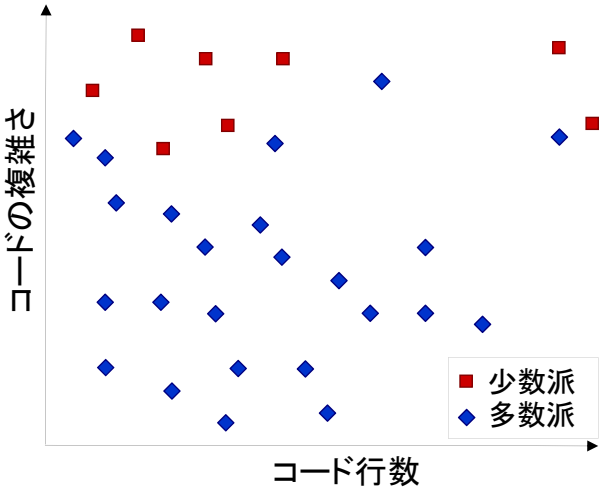
1. ケースの選択
   - 少数派ケースをランダムに1つ選択する．
2. ケースの複製・追加
   - 手順1で選択されたケースを複製し，新たなケースとしてデータセットに追加する．



SMOTE

- *k*-最近傍のケースを基に新たなケースを生成することで，少数派ケースの数を増加させる．

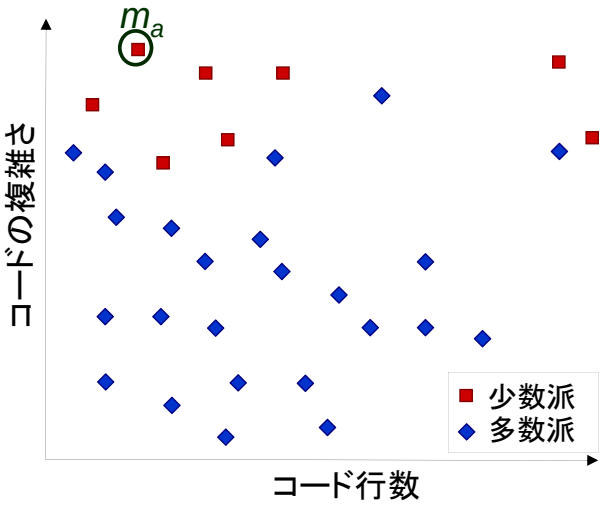
1. ケースの選択
2. *k*-最近傍の特定
3. *k*-最近傍からケースの選択
4. ケースの追加



SMOTE

- k-最近傍のケースを基に新たなケースを生成することで，少数派ケースの数を増加させる．

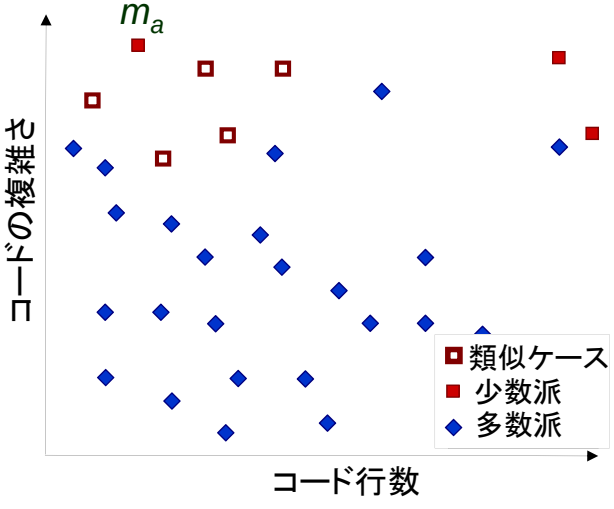
1. ケースの選択
2. k-最近傍の特定
3. k-最近傍からケースの選択
4. ケースの追加



SMOTE

- k-最近傍のケースを基に新たなケースを生成することで，少数派ケースの数を増加させる．

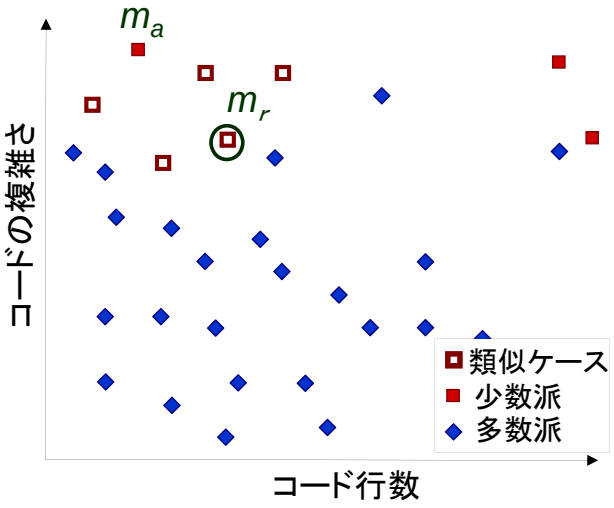
1. ケースの選択
2. k-最近傍の特定
   - 類似度計算により，ケース m_a のk-最近傍を特定する．
   - 類似度指標として，ユークリッド距離を用いた．
3. k-最近傍からケースの選択
4. ケースの追加



SMOTE

- k-最近傍のケースを基に新たなケースを生成することで，少数派ケースの数を増加させる．

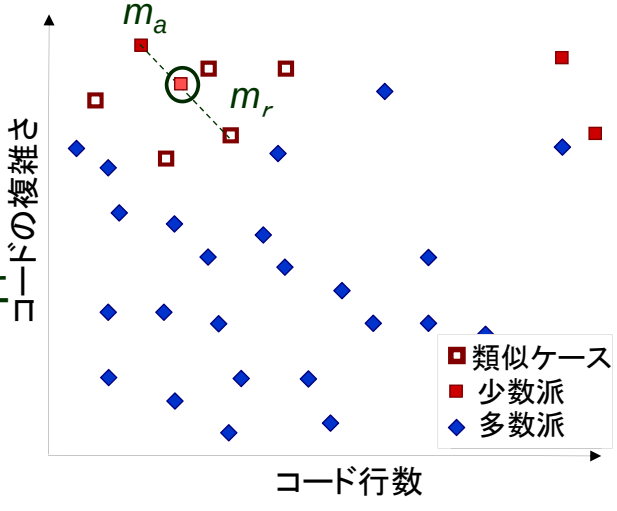
1. ケースの選択
2. k-最近傍の特定
3. k-最近傍からケースの選択
   - k-最近傍からランダムにケースm_rを1つ求める．
4. ケースの追加



SMOTE

- k-最近傍のケースを基に新たなケースを生成することで，少数派ケースの数を増加させる．

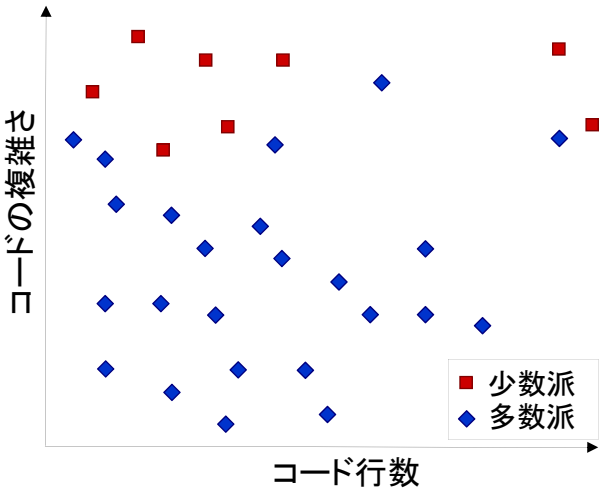
1. ケースの選択
2. k-最近傍の特定
3. k-最近傍からケースの選択
4. ケースの追加
   - ケースm_aとケースm_rの間に新たなケースを追加する．



MAHAKIL

- 典型的ケースと非典型的ケースから新たなケースを生成することで，偏りを解消する．

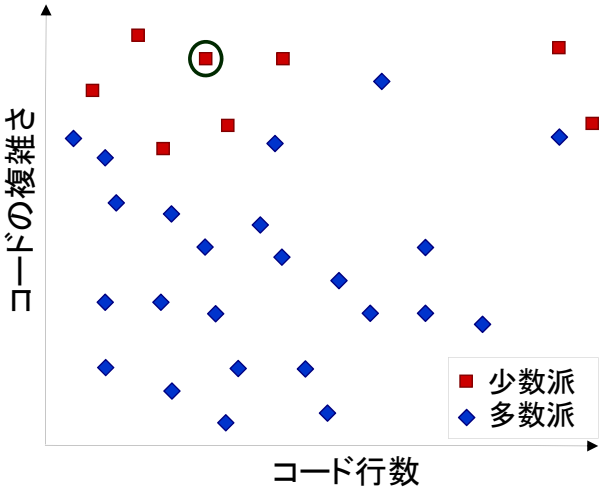
1. 典型的ケースの選択
2. 非典型ケースの選択
3. ケースの追加



MAHAKIL

- 典型的ケースと非典型的ケースから新たなケースを生成することで，偏りを解消する．

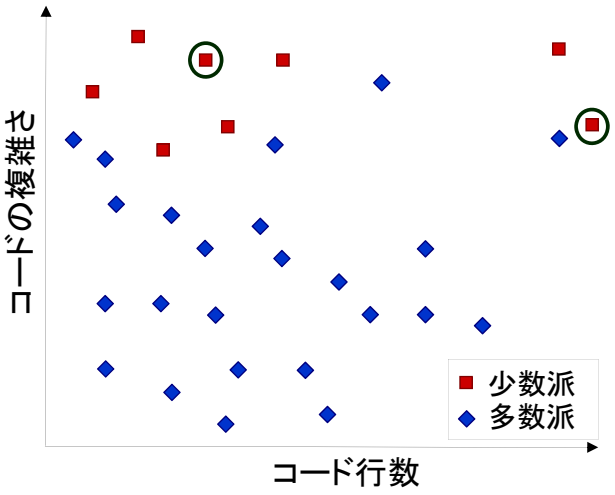
1. 典型的ケースの選択
2. 非典型ケースの選択
3. ケースの追加



MAHAKIL

- 典型的ケースと非典型的ケースから新たなケースを生成することで，偏りを解消する．

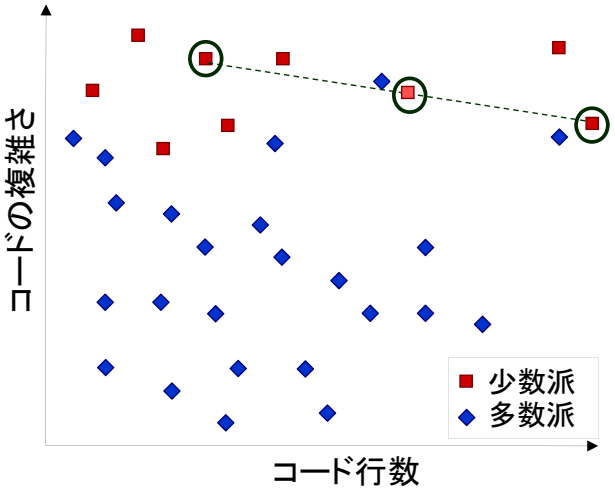
1. 典型的ケースの選択
2. 非典型ケースの選択
3. ケースの追加



MAHAKIL

- 典型的ケースと非典型的ケースから新たなケースを生成することで，偏りを解消する．

1. 典型的ケースの選択
2. 非典型ケースの選択
3. ケースの追加



評価実験

■バグモジュール判別モデル

MAHAKIL採用

サンプリングなし

Model	Sampling	Wins	Losses	Wins-Losses
KNN	MAHAKIL	34	11	23
RF	MAHAKIL	32	11	21
NNET	MAHAKIL	29	9	20
C45	MAHAKIL	29	13	16
SVM	MAHAKIL	23	12	11
RF	SMOTE	21	15	6
NNET	ADASYN	19	14	5
RF	BORDERLINE	27	22	5
C45	NONE	28	27	1
NNET	BORDERLINE	18	17	1
RF	ADASYN	24	23	1
RF	NONE	28	27	1
KNN	NONE	28	28	0
SVM	NONE	27	27	0
NNET	NONE	26	27	-1

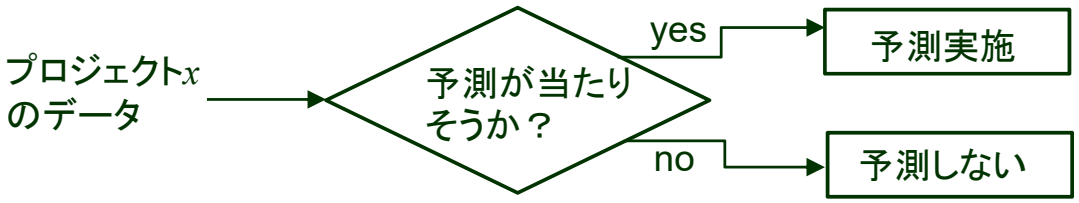
技術3

2ステップ予測*

*木下直樹, 門田暁人: ソフトウェア開発工数の二段階予測のフィージビリティスタディ. コンピュータソフトウェア (to appear).

概要

- ソフトウェア開発プロジェクトは個別性が高いのだから，予測が常に当たるとは限らない．
- 予測が当たりそうな時だけ予測しよう．



  - 例：金融系プロジェクトは，開発プロセスが安定しているので大外れがない．

- 機械学習でなんでもうまくいくとは限らない．
- むしろ機械学習がうまくいかない場面を見極めるべき．

データとモデルの例

PM経験年数	チーム経験年数	開発言語	ファンクションポイント	開発期間	開発工数
2	1	0	214	13	3913
4	3	0	103	4	2422
3	4	0	297	12	2800
1	4	0	237	21	4067
1	2	0	271	17	9051
1	1	0	88	3	2282

↓ 重回帰分析

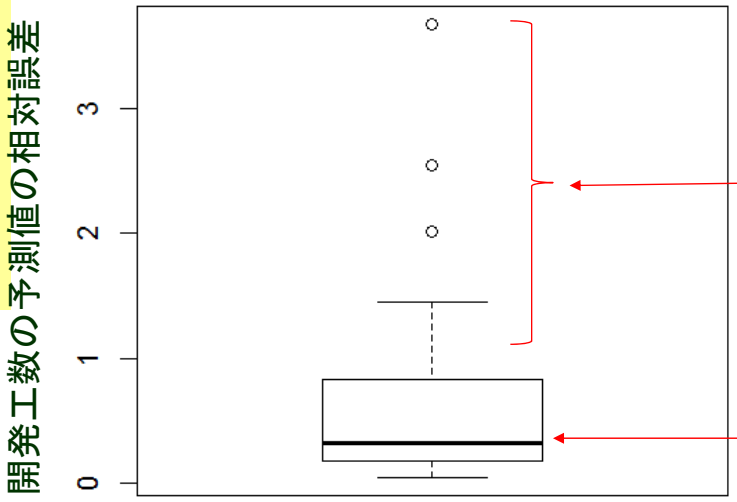
開発工数予測モデル

開発工数 = -214*PM経験年数-577*チーム経験年数-4138*開発言語
+7.28*調整済FP+150*開発期間+2786

予測結果の例

開発工数 = -214*PM経験年数-577*チーム経験年数-4138*開発言語
+7.28*調整済FP+150*開発期間+2786

新規プロジェクトの予測



相対誤差が1.0
すなわち100%を
超えているプロジェクトも
少なくない.

相対誤差の中央値は0.28

予測精度の箱ひげ図

基本アイデア(変数を2つに分ける)

PM経験年数	チーム経験年数	開発言語	ファンクションポイント	開発期間	開発工数
2	1	0	214	13	3913
4	3	0	103	4	2422
3	4	0	297	12	2800
1	4	0	237	21	4067
1	2	0	271	17	9051

② 残りの変数を予測の信頼度の推定に使う.

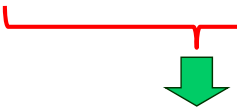
モデルの残差のばらつき

① 開発工数への寄与率の高い変数を予測モデルの構築に使う.

- 全ての変数を予測モデルの構築に使ってしまうと,予測精度の推定はできなくなる.
  - 理想的なモデルにおいては,残差のばらつきの期待値はプロジェクトによらず一定

予測の信頼度の推定

PM経験年数	チーム経験年数	開発言語	調整済FP	開発期間	開発工数
2	1	0	214	13	3913
4	3	0	103	4	2422
3	4	0	297	12	2800
1	4	0	237	21	4067
1	2	0	271	17	9051
1	1	0	88	3	2282



1. 各変数をカテゴリ変数に変換する.
2. 全ての変数について，カテゴリごとにプロジェクト集合を作る.
3. 作成したプロジェクト集合についての残差分散を求める.

プロジェクト集合	残差分散
PM経験年数＝小	2610577
PM経験年数＝大	1520112
チーム経験年数＝小	・・・
チーム経験年数＝大	・・・
開発言語＝0	・・・

} 残差分散の高いプロジェクトカテゴリについては，予測を行わないこととする.

評価実験

■Desharnaisデータセット*

- カナダのあるソフトウェア企業の77件の開発実績データ
- ソフトウェア開発工数予測研究においてよく用いられる.

カテゴリ変数	説明
TeamExp	開発チームの経験年数
ManagerExp	プロジェクトマネージャの経験年数
Transactions	トランザクション数(ファンクション数算出の基礎となる値)
Entities	エンティティ数(ファンクション数算出の基礎となる値)
Lang2	開発言語2を使っているなら1そうでないなら0
Lang3	開発言語3を使っているなら1そうでないなら0
Length	開発期間
PointsAdjust	調整済みファンクションポイント,これが開発規模を表す
Effort	実際にかかった工数

*J. Desharnais, “Analyse Statistique de la Productivitie des Projets Informatique a Partie de la Technique des Point des Function,” Master Thesis, University of Montreal, 1989.

評価実験

■Desharnaisデータセット*

- カナダのあるソフトウェア企業の77件の開発実績データ
- ソフトウェア開発工数予測研究においてよく用いられる.
- 6年分のデータがあるため，過去4年（58件）をモデル構築に用い，将来2年（19件）を予測することを想定する.

TeamExp	開発チームの経験年数
ManagerExp	プロジェクトマネージャの経験年数
Transactions	トランザクション数(ファンクション数算出の基礎となる値)
Entities	エンティティ数(ファンクション数算出の基礎となる値)
Lang2	開発言語2を使っているなら1そうでないなら0
Lang3	開発言語3を使っているなら1そうでないなら0
Length	開発期間
PointsAdjust	調整済みファンクションポイント,これが開発規模を表す
Effort	実際にかかった工数

*J. Desharnais, “Analyse Statistique de la Productivitie des Projets Informatique a Partie de la Technique des Point des Function,” Master Thesis, University of Montreal, 1989.

評価実験

■Desharnaisデータセット*

- カナダのあるソフトウェア企業の77件の開発実績データ
- ソフトウェア開発工数予測研究においてよく用いられる.

重回帰モデルの説明変数として統計的に有意とならなかった変数

モデルの説明変数に採用した変数

目的変数

カテゴリ変数	説明
TeamExp	開発チームの経験年数
ManagerExp	プロジェクトマネージャの経験年数
Transactions	トランザクション数(ファンクション数算出の基礎となる値)
Entities	エンティティ数(ファンクション数算出の基礎となる値)
Lang2	開発言語2を使っているなら1そうでないなら0
Lang3	開発言語3を使っているなら1そうでないなら0
Length	開発期間
PointsAdjust	調整済みファンクションポイント,これが開発規模を表す
Effort	実際にかかった工数

*J. Desharnais, “Analyse Statistique de la Productivitie des Projets Informatique a Partie de la Technique des Point des Function,” Master Thesis, University of Montreal, 1989.

評価実験

■Desharnaisデータセット*
- カナダのあるソフトウェア企業の77件の開発実績データ
- ソフトウェア開発工数予測研究においてよく用いられる.

モデルに採用しなかった変数は，予測の信頼度の推定に用いる.

カテゴリ変数	説明
TeamExp	開発チームの経験年数
ManagerExp	プロジェクトマネージャの経験年数
Transactions	トランザクション数(ファンクション数算出の基礎となる値)
Entities	エンティティ数(ファンクション数算出の基礎となる値)
Lang2	開発言語2を使っているなら1そうでないなら0
Lang3	開発言語3を使っているなら1そうでないなら0
Length	開発期間
PointsAdjust	調整済みファンクションポイント,これが開発規模を表す
Effort	実際にかかった工数

重回帰モデルの説明変数として統計的に有意とならなかった変数

モデルの説明変数に採用した変数

目的変数

*J. Desharnais, “Analyse Statistique de la Productivitie des Projets Informatique a Partie de la Technique des Point des Function,” Master Thesis, University of Montreal, 1989.

評価実験

■Desharnaisデータセット*
- カナダのあるソフトウェア企業の77件の開発実績データ
- ソフトウェア開発工数予測研究においてよく用いられる.

カテゴリ変数	説明
TeamExp	開発チームの経験年数
ManagerExp	プロジェクトマネージャの経験年数
Transactions	トランザクション数(ファンクション数算出の基礎となる値)
Entities	エンティティ数(ファンクション数算出の基礎となる値)
Lang2	開発言語2を使っているなら1そうでないなら0
Lang3	開発言語3を使っているなら1そうでないなら0
Length	開発期間
PointsAdjust	調整済みファンクションポイント,これが開発規模を表す
Effort	実際にかかった工数

工数予測モデル

$$\hat{Y} = -330 + 3673 \times Lang3 + 15.49 \times Length + 14.15 \times PointsAdjust$$

重回帰モデルの説明変数として統計的に有意とならなかった変数

モデルの説明変数に採用した変数

目的変数

*J. Desharnais, “Analyse Statistique de la Productivitie des Projets Informatique a Partie de la Technique des Point des Function,” Master Thesis, University of Montreal, 1989.

© Akito Monden, Okayama University

実験結果

プロジェクト集合		モデル構築時				予測時	
		残差平方平均	残差分散	相対残差分散	残差変動係数	絶対誤差平均	相対誤差平均
全体		521?260	247?286	0.383	0.934	?855	0.745
TeamExp	小	350?607	2?00?6?	0.6?5	?.?73	20?6	0.864
TeamExp	大	692491?4	250244?	0.?64	0.732	?507	0.487
ManagerExp	小	359461?	?6???98	0.?83	0.868	??67	0.544
ManagerExp	大	6355836	3076099	0.534	0.94?	2356	0.89?
Transactions	小	30?9079	?637282	0.566	?.050	867	0.603
Transactions	大	7564?68	2949704	0.20?	0.776	2745	0.874
Entities	小	?573375	66043?	0.688	0.823	???2	0.664
Entities	大	8?70667	326644?	0.?30	0.795	3935	0.973
Lang2	小	53?096?	26?0578	0.487	0.96?	2423	?.038
Lang2	大	4933?85	2229340	0.085	0.854	?075	0.343

実験結果

プロジェクト集合		モデル構築時				予測時	
		残差平方平均	残差分散	相対残差分散	残差変動係数	絶対誤差平均	相対誤差平均
全体		521?260	247?286	0.383	0.934	?855	0.745
TeamExp	小	350?607	2?00?6?	0.6?5	?.?73	20?6	0.864
TeamExp	大	692491?4	250244?	0.?64	0.732	?507	0.487
ManagerExp	小	359461?	?6???98	0.?83	0.868	??67	0.544
ManagerExp	大	6355836	3076099	0.534	0.94?	2356	0.89?
Transactions	小	30?9079	?637282	0.566	?.050	867	0.603
Transactions	大	7564?68	2949704	0.20?	0.776	2745	0.874
Entities	小	?573375	66043?	0.688	0.823	???2	0.664
Entities	大	8?70667	326644?	0.?30	0.795	3935	0.973
Lang2	小	53?096?	26?0578	0.487	0.96?	2423	?.038
Lang2	大	4933?85	2229340	0.085	0.854	?075	0.343

各行はプロジェクトの集合を表す.

実験結果

プロジェクト集合		モデル構築時				予測時	
		残差平方平均	残差分散	相対残差分散	残差変動係数	絶対誤差平均	相対誤差平均
全体		52□3260	247□286	0.383	0.934	□855	0.745
TeamExp	小	350□607	2□0□6□	0.6□5	□.□73	20□6	0.864
TeamExp	大	6924□□4	250244□	0.□64	0.732	□507	0.487
ManagerExp	小	3594□□	□6□□□98	0.□83	0.868	□□67	0.544
ManagerExp	大	6355836	3076099	0.534	0.94□	2356	0.89□
Transactions	小	30□9079	□637□8□	0.566	□.050	867	0.603
Transactions	大	756□□68	2949704	0.20□	0.776	2745	0.874
Entities	小	□573375	6604□□	0.688	0.823	□□□2	0.664
Entities	大	8□70667	3266□□□	0.□30	0.795	3935	0.973
Lang2	小	53□096□	26□0578	0.487	0.96□	2423	□.038
Lang2	大	4933□85	2229340	0.085	0.854	□075	0.343

各行はプロジェクトの集合を表す.

各プロジェクト集合に対する予測精度を表す.

実験結果

プロジェクト集合		モデル構築時				予測時	
		残差平方平均	残差分散	相対残差分散	残差変動係数	絶対誤差平均	相対誤差平均
全体		52□3260	247□286	0.383	0.934	□855	0.745
TeamExp	小	350□607	2□0□6□	0.6□5	□.□73	20□6	0.864
TeamExp	大	6924□□4	250244□	0.□64	0.732	□507	0.487
ManagerExp	小	3594□□	□6□□□98	0.□83	0.868	□□67	0.544
ManagerExp	大	6355836	3076099	0.534	0.94□	2356	0.89□
Transactions	小	30□9079	□637□8□	0.566	□.050	867	0.603
Transactions	大	756□□68	2949704	0.20□	0.776	2745	0.874
Entities	小	□573375	6604□□	0.688	0.823	□□□2	0.664
Entities	大	8□70667	3266□□□	0.□30	0.795	3935	0.973
Lang2	小	53□096□	26□0578	0.487	0.96□	2423	□.038
Lang2	大	4933□85	2229340	0.085	0.854	□075	0.343

各行はプロジェクトの集合を表す.

各プロジェクト集合に対する予測精度を表す.

© Akito Monden, Okayama University

実験結果

プロジェクト集合		モデル構築時				予測時	
		残差平方平均	残差分散	相対残差分散	残差変動係数	絶対誤差平均	相対誤差平均
全体		52□3260	247□286	0.383	0.934	□855	0.745
TeamExp	小	350□607	2□0□6□	0.6□5	□.□73	20□6	0.864
TeamExp	大	6924□□4	250244□	0.□64	0.732	□507	0.487
ManagerExp	小	3594□□	□6□□□98	0.□83	0.868	□□67	0.544
ManagerExp	大	6355836	3076099	0.534	0.94□	2356	0.89□
Transactions	小	30□9079	□63728□	0.566	□.050	867	0.603
Transactions	大	7564□68	2949704	0.20□	0.776	2745	0.874
Entities	小	□573375	6604□□	0.688	0.823	□□□2	0.664
Entities	大	8□70667	3266□□□	0.□30	0.795	3935	0.973
Lang2	小	53□096□	26□0578	0.487	0.96□	2423	□.038
Lang2	大	4933□85	2229340	0.085	0.854	□075	0.343

各行はプロジェクトの集合を表す.

各プロジェクト集合に対する予測精度を表す.

実験結果

予測の信頼度を推定するための尺度

プロジェクト集合		モデル構築時				予測時	
		残差平方平均	残差分散	相対残差分散	残差変動係数	絶対誤差平均	相対誤差平均
全体		52□3260	247□286	0.383	0.934	□855	0.745
TeamExp	小	350□607	2□0□6□	0.6□5	□.□73	20□6	0.864
TeamExp	大	6924□□4	250244□	0.□64	0.732	□507	0.487
ManagerExp	小	3594□□	□6□□□98	0.□83	0.868	□□67	0.544
ManagerExp	大	6355836	3076099	0.534	0.94□	2356	0.89□
Transactions	小	30□9079	□63728□	0.566	□.050	867	0.603
Transactions	大	7564□68	2949704	0.20□	0.776	2745	0.874
Entities	小	□573375	6604□□	0.688	0.823	□□□2	0.664
Entities	大	8□70667	3266□□□	0.□30	0.795	3935	0.973
Lang2	小	53□096□	26□0578	0.487	0.96□	2423	□.038
Lang2	大	4933□85	2229340	0.085	0.854	□075	0.343

各行はプロジェクトの集合を表す.

各プロジェクト集合に対する予測精度を表す.

実験結果

プロジェクト集合		モデル構築時				予測時	
		残差平方平均	残差分散	相対残差分散	残差変動係数	絶対誤差平均	相対誤差平均
全体		5213260	2471286	0.383	0.934	1855	0.745
Entities	小	1573375	660431	0.688	0.823	1112	0.664
ManagerExp	小	3594611	1611198	0.183	0.868	1167	0.544
Transactions	小	3019079	1637281	0.566	1.050	867	0.603
TeamExp	小	3501607	2100161	0.615	1.173	2016	0.864
Lang2	大	4933185	2229340	0.085	0.854	1075	0.343
TeamExp	大	6924914	2502441	0.164	0.732	1507	0.487
Lang2	小	5310961	2610578	0.487	0.961	2423	1.038
Transactions	大	7564168	2949704	0.201	0.776	2745	0.874
ManagerExp	大	6355836	3076099	0.534	0.941	2356	0.891
Entities	大	8170667	3266441	0.130	0.795	3935	0.973

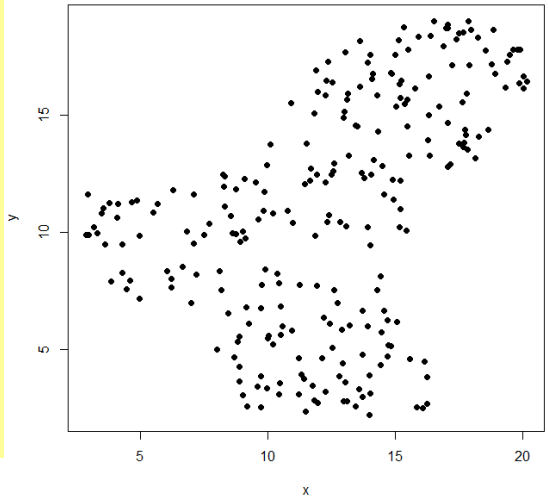
残差分散がもっとも小さくなるEntities=小の場合にのみ予測を行うことで，
絶対誤差平均を1885から1112へ，相対誤差平均を0.745から0.664へと低減できる．

技術4

原型分析*

*瀧本恵介, 門田暁人, 尾上紗野, 畑秀明, 亀井靖高: 原型分析を用いたソフトウェアバグ分析. 電子情報通信学会技術報告, ソフトウェアサイエンス研究会, No. SS2016-33, 2016.

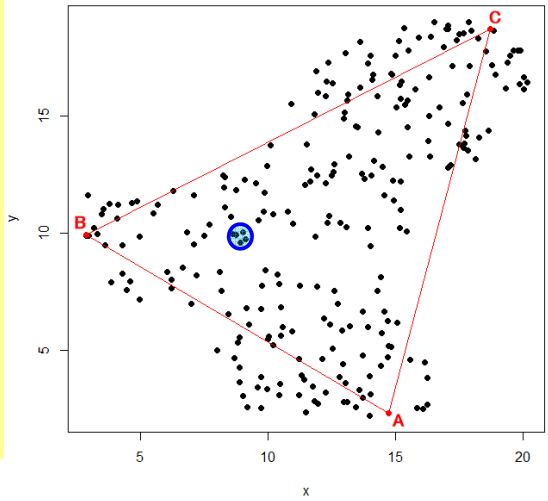
原型分析*とは



- データ集合を代表する極端な値(原型，Archetype)を抽出する手法.

*A Cutler, L Breiman: Archetypal analysis, Technometrics, Vol.36, No.4, pp.338-347, 1994.

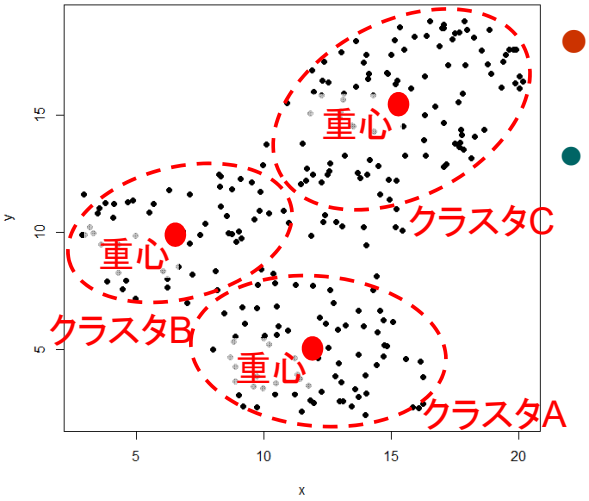
原型分析*とは



- データ集合を代表する極端な値(原型，Archetype)を抽出する手法.
- あるデータ集合から原型分析で3つの原型(点A，B，C)を抽出.
- 各データは，原型A,B,Cを用いて表現できる.

  例）あるデータは，原型Aが30%，原型Bが50%，原型Cが20%で構成されている.

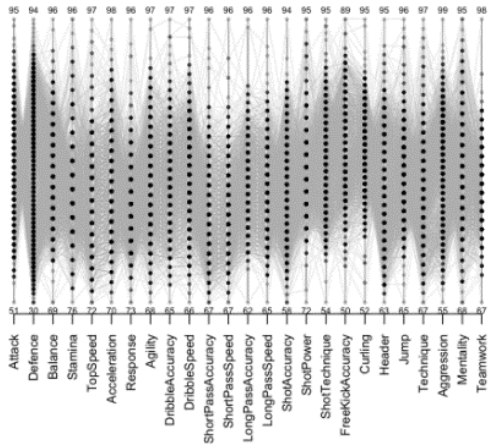
原型分析*とは



- クラスタリングでは，平均的な値が代表点として抽出される．
- 特徴に乏しい点が抽出されてしまう恐れがある．

原型分析を用いた研究例

- サッカーゲームの選手データを原型分析*
  - KONAMIの「ワールドサッカー ウイニングイレブン」を使用．
  - 全25種類のスキルからなる，1658人の選手データを原型分析．

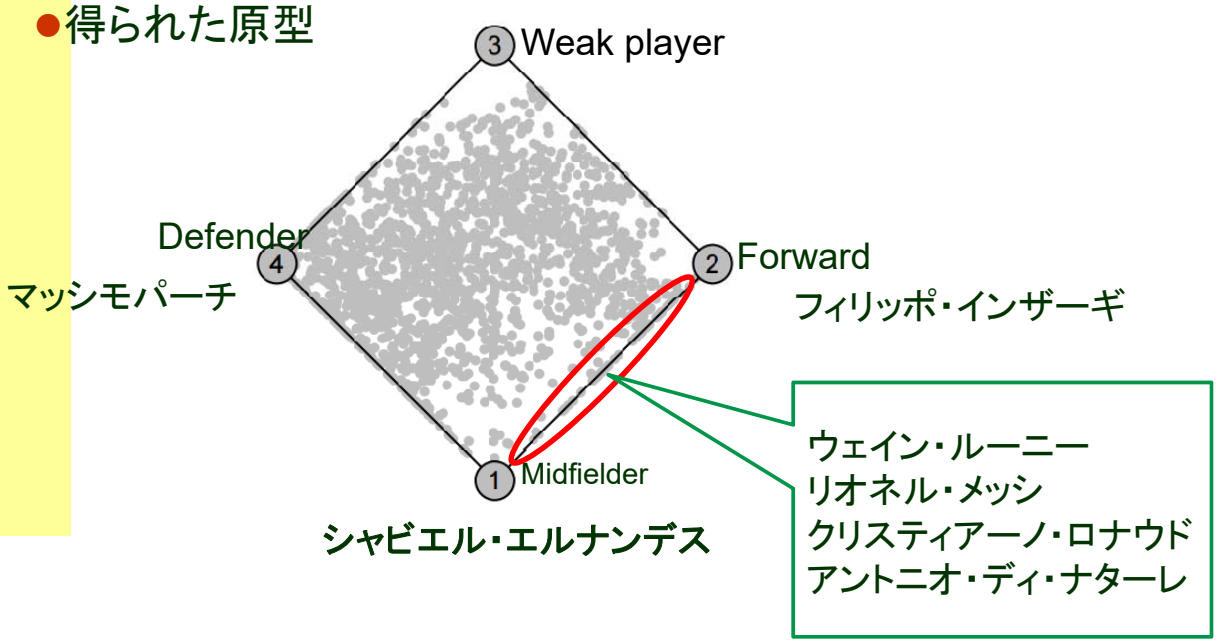


全サッカー選手のスキルの折れ線グラフ

*M. J. A. Eugster, Archetypal Analysis Mining the Extreme, HIIT seminar, http://www.statistik.lmu.de/~eugster/publications/talk-2012-HIIT-archetypes.pdf

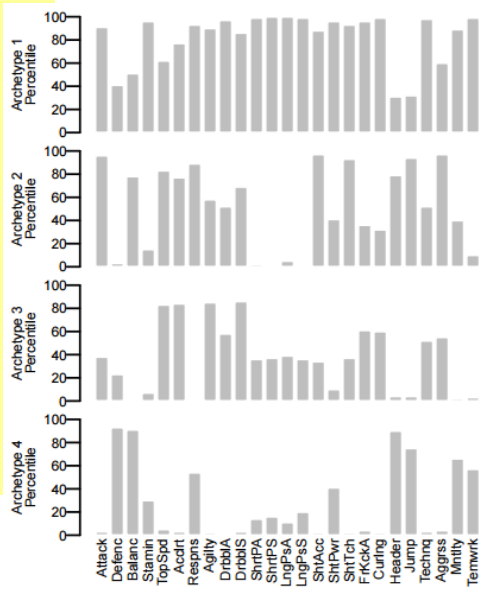
原型分析を用いた研究例

- 得られた原型



*M. J. A. Eugster, Archetypal Analysis Mining the Extreme, HIIT seminar, http://www.statistik.lmu.de/~eugster/publications/talk-2012-HIIT-archetypes.pdf

原型分析を用いた研究例



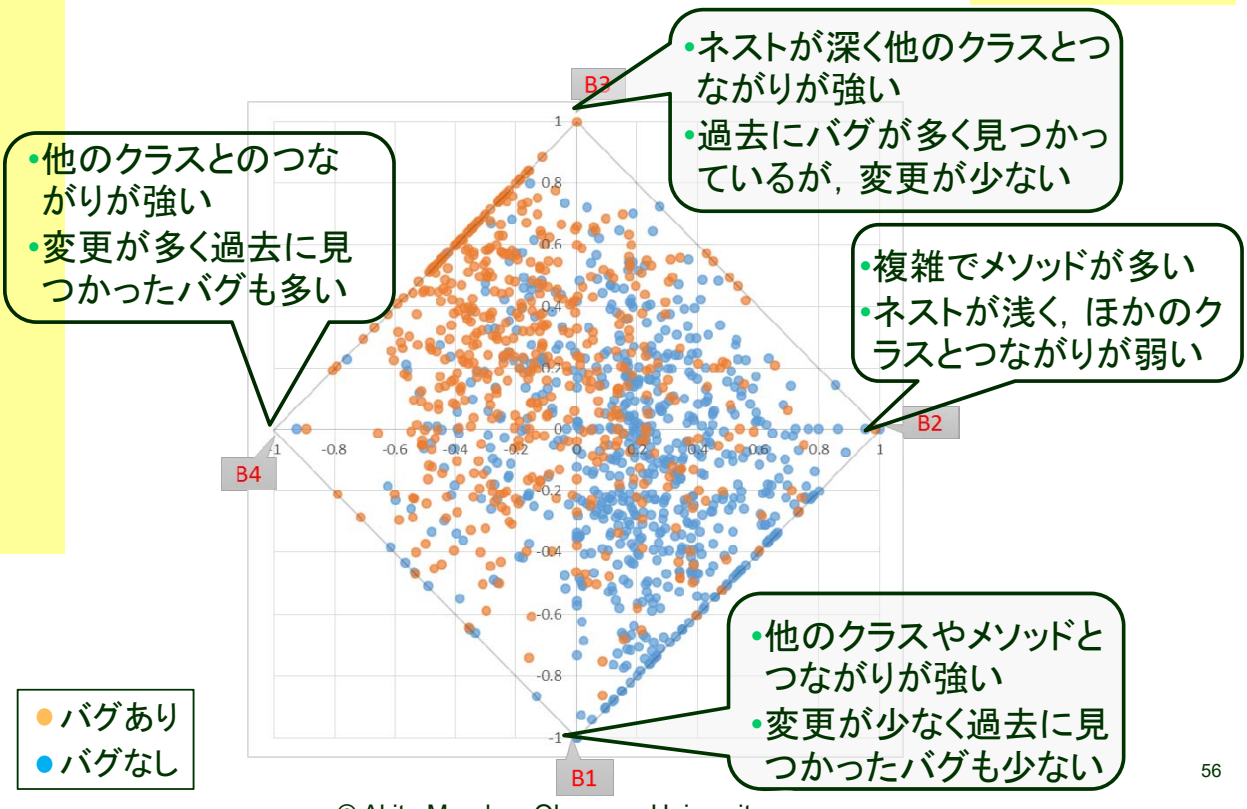
- 原型1：Midfielder型
  ディフェンスやバランス，ヘディングなどを除いて，ほとんどのスキルが高い．
- 原型2：Forward型
  シュート能力が高い一方で，パスに関するスキルが低い．
- 原型3：Weak Player型
  走力が高い一方で，ボールを扱うスキル全般が低い．
- 原型4：Defender型
  ディフェンスやバランス，ヘディング，ジャンプのスキルが高い．

ソフトウェアを対象とした原型分析

ソフトウェア名	バージョン名	モジュール数	バグを含むモジュール数	バグモジュール率
Mylyn	3.0	1241	547	44.1%

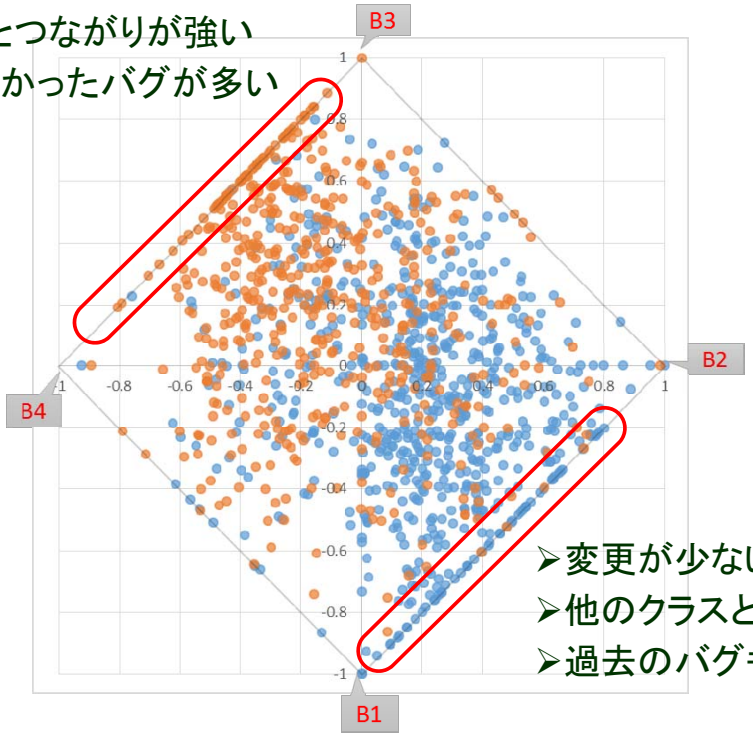
メトリクス	説明	バグとの関係
NBD	最大ネスト数	制御フローの複雑さ
VG/MLOC	サイクロマチック数／実行行数	
NOF/MLOC	フィールド数／実行行数	オブジェクト指向の規模尺度
NOM/MLOC	メソッド数／実行行数	
CBO	クラス間の結合度	モジュール独立性の尺度
RFC/NOM	クラスの応答の規模／メソッド数	
CHURN/MLOC	変更行数／実行行数	開発プロセス尺度
PRE	リリース前6か月以内に見つかったバグ数	

ソフトウェアを対象とした原型分析



ソフトウェアを対象とした原型分析

- 他のクラスとつながりが強い
- 過去に見つかったバグが多い



- 変更が少ない.
- 他のクラスとのつながりが弱い.
- 過去のバグも少ない.

● バグあり
● バグなし

その他の技術

- テキストマイニング
  - N-gram IDF
- アソシエーションルールマイニング A⇒B
  - 欠損値があってもよい.
  - データクリーニング，ルール削減
  - 例外ルール A&R ⇒ not B
  - 結論部の拡張 A ⇒ F (X, Y)≧ α

分析事例

ソフトウェア品質データの分析*

*A. Monden, M. Tsunoda, M. Barker, K. Matsumoto:
Examining Software Engineering Beliefs about System Testing Defects.
IEEE IT Professional 19(2): 58-64 (2017)

ソフトウェア品質データ

プロジェクトID	概要設計			詳細設計			コーディング		
	ドキュメント量	レビュー工数	バグ数	ドキュメント量	レビュー工数	バグ数	SLOC	レビュー工数	バグ数
xxx	xxx	xxx	xxx	xxx	xxx	xxx	xxx	xxx	xxx
xxx	xxx	xxx	xxx	xxx	xxx	xxx	xxx	xxx	xxx
…	…	…	…	…	…	…	…	…	…

単体テスト		結合テスト		システムテスト	
テストケース数	バグ数	テストケース数	バグ数	テストケース数	バグ数
xxx	xxx	xxx	xxx	xxx	xxx
xxx	xxx	xxx	xxx	xxx	xxx
…	…	…	…	…	…

出荷直前に品質問題が発覚する．なんとかしたい．

データのクリーニング

- ４年分のデータ（約300プロジェクト）を対象とする.
- ２つの部署に限定する.
- 欠損値を含むプロジェクトを除外する.
- 小規模プロジェクトを除外する.
- 予実が大きく乖離しているプロジェクトを除外する.

- 部署A：18プロジェクト
- 部署B：16プロジェクト

- 仮説を立て，メトリクスを定義し，分析する.

4つの仮説（SE Beliefs）

一般的に知られている法則（SE Belief）を元に仮説を立てる.

仮説1. レビュー密度＝大 → 品質＝高
仮説2. 上流工程検出バグ密度＝大 → 品質＝低
仮説3. 単体・結合テスト密度＝大 → 品質＝高
仮説4. 再利用コード比率＝大 → 品質＝低

- 品質は，システムテストでのバグ密度により評価する.

4つのメトリクス

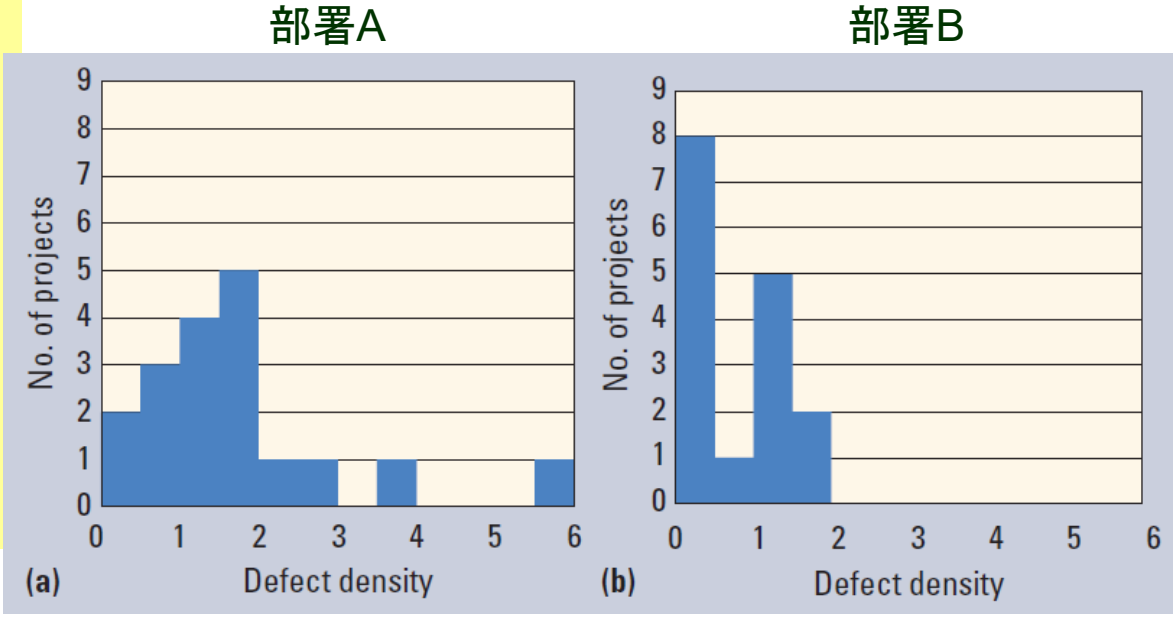
M1. レビュー密度
＝（概要設計レビュー工数＋詳細設計レビュー工数＋コードレビュー工数）÷開発規模（KSLOC）

M2. 上流工程検出バグ密度
＝概要設計バグ密度＋詳細設計バグ密度＋コードレビューバグ密度

M3. 単体・結合テスト密度
＝単体テスト密度＋結合テスト密度

M4. 再利用コード比率
＝流用行数÷（新規＋変更＋流用行数）

システムテストのバグ密度



- 部署Bの方が品質がよい.

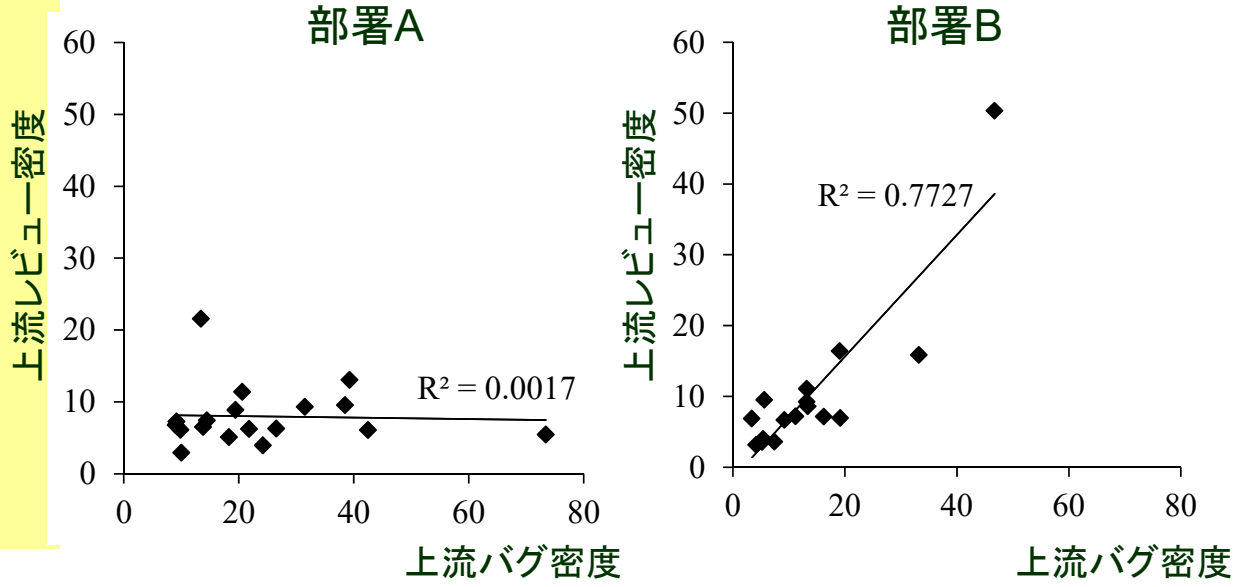
重回帰分析の結果

	部署A		部署B	
説明変数	偏回帰係数	P値	偏回帰係数	P値
M1. 上流レビュー密度	0.0226	0.768	−0.0217	0.355
M2. 上流バグ密度	0.0394	*0.029*	0.0031	0.885
M3. 単体・結合テスト密度平均	−0.0037	0.647	−0.0006	0.893
M4. 再利用比率	2.0649	0.086	1.0512	*0.006*

- 部署Aは，上流バグ密度＝大 → 品質低下
- 部署Bは，再利用比率＝大 → 品質低下

このような違いが出たのはなぜか？

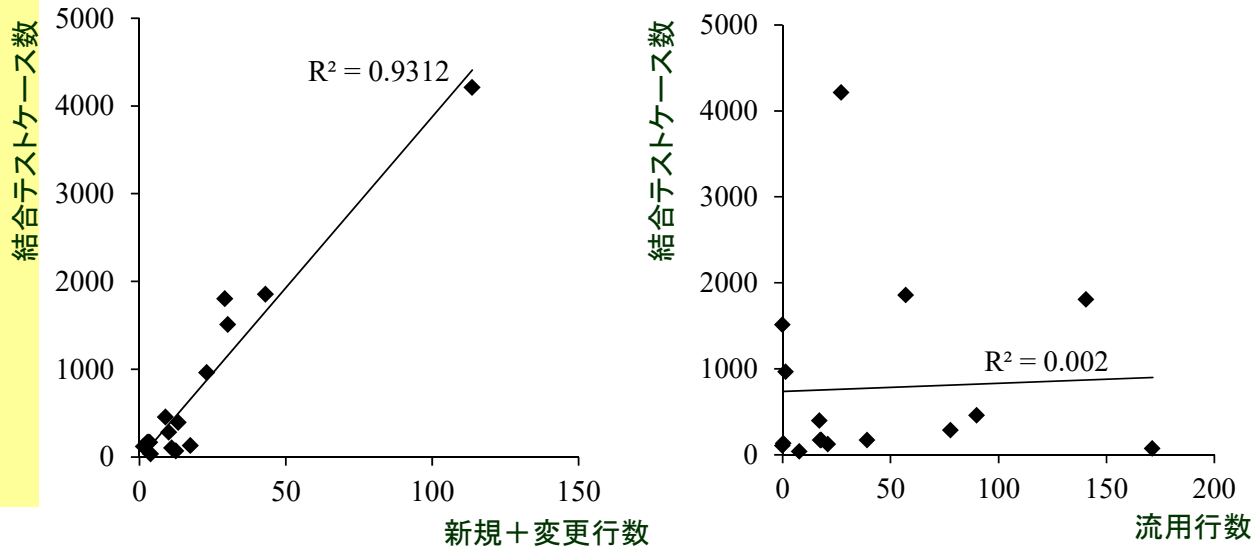
上流バグ密度 ー 上流レビュー密度の分析



部署Bは，上流バグと上流レビュー密度が比例しているが，
部署Aは，上流バグ密度に関わらず上流レビュー密度はほぼ一定

→ 部署Aは追加レビューをすべき？

再利用コード比率に関する分析(部署B)



テストケース数は，新規＋変更行数に比例している．
流用行数とは相関なし．

→ 部署Bは流用部分のテストを増やすべき？

その他の取り組み

- レビュー工数の計画値の妥当性は？
- テストケース数決定時の開発規模の妥当な式は？
  開発規模＝x新規＋y変更＋z流用　とするとき，
  x, y, zの妥当な値は？

まとめ

- 矛盾ケースの除去
- オーバーサンプリング
- 2ステップ予測
- 原型分析

- 分析事例