他プロジェクトの実績データに基づく Fault-Prone モジュール判別の試み

木浦幹雄　　柗本真佑　　亀井靖高　　門田暁人　　松本健一
奈良先端科学技術大学院大学　情報科学研究科
E-mail: {mikio-k,shinsuke-m,yasuta-k,akito-m,matumoto}@is.naist.jp

概要

ソフトウェアテストおよび保守において，fault-prone モジュール（バグを含む確率の高いモジュール）を特定することは，テストの効率化，および信頼性を確保する上で重要である．そのために，従来，多数のモジュールから計測されたプロダクトメトリクスを説明変数とし，モジュールに含まれるバグの有無を目的変数とする判別モデルが多数提案されている．一般に，判別モデルの構築には，旧バージョンのモジュールから計測されたメトリクス値とバグの実績データが用いられる．しかし，この方法は，旧バージョンの存在しない新規開発プロジェクトでは使えない．本稿では，他のプロジェクトのモジュールから計測された実績データを元にモデル構築を行い，fault-prone モジュールの判別を試みた結果について報告する．

1. はじめに

ソフトウェアテストおよび保守において，fault-prone モジュール（バグを含む確率の高いモジュール）を特定することは，テストの効率化やソフトウェアの信頼性を向上する上で重要である．そのために，多数のモジュールから計測されたプロダクトメトリクス（プログラム行数やサイクロマティック数，変更行数など）を説明変数とし，モジュールのバグの有無を目的変数とする，fault-prone モジュール判別モデルが提案されており，線形判別分析 [5]，ロジスティック回帰分析 [1]，ニューラルネット [6] など様々な種類のモデルが存在する．一般にこれらの判別モデルは，旧バージョンのモジュールから計測されたメトリクス値とバグの実績データを記したデータセット（以降，フィットデータ）を用いて構築し，現行バージョンの fault-prone モジュールの予測を行う．

しかし，新規開発プロジェクトでは旧バージョンが存在しないため，判別モデルを構築することができず，fault-prone モジュールを予測することができない．例えば，情報処理推進機構ソフトウェア・エンジニアリング・センターが 19 社から収集した 1419 件のプロジェクトの実績データでは，全体の 59.6%が新規開発プロジェクトであり [2]，フィットデータを用意できないプロジェクトが大半を占める．

一方，現行のプロジェクトとは異なった，他のプロジェクトで計測されたデータセットをフィットデータとして用いることも考えられる．ただし，この場合プロジェクトの特性（アーキテクチャや開発言語，開発規模など）の違いがメトリクス値の傾向の違いとして表れる可能性があり，精度の高いモデルを構築できるとは限らない．また，この場合の予測精度については従来十分に研究されていない．

そこで，本稿では他のプロジェクトのモジュールから計測された実績データを元に判別モデルの構築を行い，fault-prone モジュールの予測を試みた結果について報告する．実験では，NASA IV&V Facility Metrics Data Program[4] が公開しているデータセットの内，8 個のデータセットを用い，自身を除くそれぞれの組み合わせ計 56 通りと，他の全プロジェクトのデータセットをフィットデータとして用いた場合の 8 通りについて予測を行う．

以降，2 章で fault-prone モジュール判別について説明し，3 章で実験の方法とその手順について述べる．4 章で実験結果について述べ，最後に 5 章で本論文の結論と今後の課題を述べる．

2. Fault-Prone モジュール判別

2.1. 諸定義

モジュールとは，ソフトウェアの最小の品質評価単位であり，通常，ソフトウェアは複数のモジュールから構成される．本稿ではソースコード中の 1 ファイルを 1 モジュールと定義する．Fault とは，モジュール中に存在する正しくない処理のことであり，ソフトウェアが所定の機能を遂行できないことの直接の原因となったプログラムの記述誤りである．Fault-prone モジュールとは，fault を含む確率が高いモジュールのことを指す．

2.2. Fault-Prone モジュール判別モデル

Fault-prone モジュール判別モデルは，過去に開発されたモジュールに関するプロダクトメトリクスを説明変数とし，モジュールに含まれるバグの有無を目的変数として構築される．構築した判別モデルに対して新規に開発したモジュールのメトリクス値を入力することで，そのモジュールのバグが含まれる確率やバグの有無を予測する．モデル構築方法としては，線形判別分析 [5] やニューラルネット [6] など様々な手法が存在するが，本稿では fault-prone モジュール判別モデルとしてよく用いられるロジスティック回帰分析 [1] を用いた．

2.3. ロジスティック回帰分析

ロジスティック回帰分析は，説明変数の変動によって目的変数が 2 値のどちらを取る確率が高いかを予測する手法である．ロジスティック回帰分析では，説明変数と目的変数の関係が以下のロジスティック回帰式により定義される．

$$P(y|x) = \frac{1}{1 + e^{-(\beta^T x + \alpha)}} \tag{1}$$

ここで，y は 2 値の目的変数，x は説明変数となる入力ベクトル，β は係数ベクトル，α は係数値であり，P は入力ベクトル x に対して y が 1 となる確率である．

3. 実験

3.1. 概要

本実験では，予測対象プロジェクトとは異なったプロジェクトで計測されたデータセットをフィットデータとして，fault-prone モジュール判別モデルを構築した際の予測精度を評価する．

実験では，フィットデータの取り方として，

1. 予測対象プロジェクト以外の単数のプロジェクトを用いる．
2. 予測対象プロジェクト以外の全てのプロジェクトを用いる．

の 2 通りを行った．

3.2. データセット

実験には，NASA IV&V Facility Metrics Data Program (MDP) [4] が公開しているデータセットを用いた．NASA IV&V Facility MDP とは，NASA の独立検証機構である IV&V Facility が過去のプロジェクトにおける成果物の一部をリポジトリに蓄積し，一般に公開している活動である．公開されているデータセットは複数のプロジェクトで開発されたモジュールに関するメトリクスを記したものである．本稿では 8 個のプロジェクトを実験対象とした．各プロジェクトの概略を表 1 に示す．表中のバグ含有率とはモジュール総数に対してのバグを含んだモジュールの数の割合を指す．また，fault-prone モジュール判別モデルを構築する際の目的変数はモジュールのバグの有無とし，説明変数はモジュールのメトリクスとした．説明変数に用いたメトリクスの名称を表 2 に示す．

3.3. 評価指標

Fault-prone モジュール判別モデルの予測精度の評価指標としては，Alberg Diagram[3] の AUC (Area Under the Curve) を用いた．

Alberg Diagram とはバグを含んでいる可能性の高い順（rank-order）にモジュールを抽出した際に，実際に

表 1. 実験に用いたデータセットの概略

プロジェクト名	開発言語	モジュール数	バグ含有率
CM1	C	505	9.5%
KC3	Java	458	9.4%
KC4	Perl	125	48.8%
MW1	C	403	7.7%
PC1	C	1107	6.9%
PC2	C	5589	0.4%
PC3	C	1563	10.2%
PC4	C	1458	12.2%

表 2. 実験に用いたデータセットのメトリクスの名称

LOC BLANK	空白行の行数
BRANCH COUNT	分岐数
LOC CODE AND COMMENT	コメント入りコード行数
LOC COMMENTS	コメント行数
CYCLOMATIC COMPLEXITY	サイクロマティック複雑度
DESIGN COMPLEXITY	McCabe の design complexity
ESSENTIAL COMPLEXITY	McCabe の essential complexity
LOC EXECUTABLE	実行可能コード行数
HALSTEAD CONTENT	Halstead の content
HALSTEAD DIFFICULTY	Halstead の difficulty
HALSTEAD EFFORT	Halstead の programming effort
HALSTEAD ERROR EST	Halstead の error estimate
HALSTEAD LENGTH	Halstead の length
HALSTEAD LEVEL	Halstead の level
HALSTEAD PROG TIME	Halstead の programming time
HALSTEAD VOLUME	Halstead の volume
NUM OPERANDS	オペランド数
NUM OPERATORS	オペレータ数
NUM UNIQUE OPERANDS	オペランド種
NUM UNIQUE OPERATORS	オペレータ種
LOC TOTAL	総行数

バグを含んでいたモジュールがどれだけ抽出できたかの割合をグラフ化したものである [3]．Alberg Diagram の例を図 1 に示す．グラフの横軸は rank-order に抽出されたモジュールの割合を表し，縦軸は判別モデルによって抽出された，実際にバグを含んでいたモジュールの割合を表す．このグラフは予測精度が高いほど左上に凸になり，逆に判別モデルがランダムに予測を行った場合，右肩上がりの直線を描く．

AUC とは曲線下面積のことであり，ある曲線を積分した値を指す．

これらのことから，Alberg Diagram の AUC とは構築されたモデルの予測精度の高さを値で示したものであり，その値域は [0,1] を取る．予測精度の高いモデルでは AUC の値が大きく，ランダムに予測するモデルの場合，AUC の値は 0.5 程度となる．図 1 の場合，AUC の値は 0.86 であり，非常に高い精度で予測を行ったといえる．

3.4. 実験手順

3.4.1 単数のデータセットで判別モデルを構築

Step 1 判別モデルの構築

ロジスティック回帰分析を用いて，8 個のデータセットから 8 つの判別モデルを構築する．

Step 2 各モジュールのバグを含む確率の予測

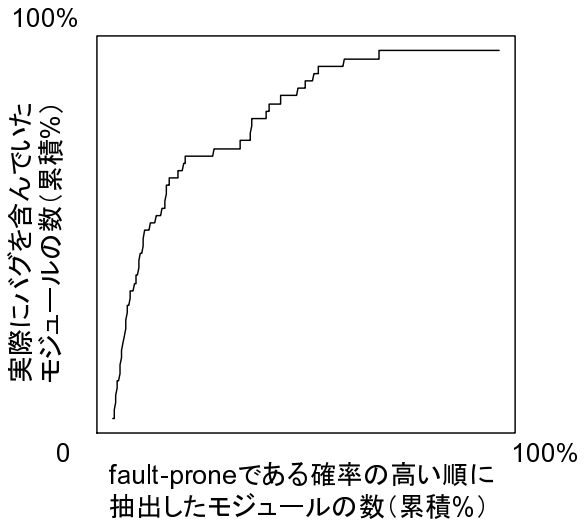


図 1. Alberg Diagram の例

構築した判別モデルを用いて，判別モデル構築に用いたデータセットを除く 7 個のプロジェクトそれぞれに対して，各モジュールのバグを含む確率を予測する．

Step 3 判別モデルの精度の評価

予測した結果を基に Alberg Diagram を計算し，AUC を算出する．

3.4.2 複数のデータセットで判別モデルを構築

Step 1 判別モデルの構築

予測対象データセットを除く 7 個のデータセットを統合し，1 個の判別モデルを構築する．

Step 2 各モジュールのバグを含む確率の予測構築した判別モデルを用いて，予測対象データセットの各モジュールのバグを含む確率を予測する．

Step 3 判別モデルの精度の評価

予測した結果を基に Alberg Diagram を計算し，AUC を算出する．

4. 実験結果

4.1. 単数のデータセットを用いた場合の実験結果

各プロジェクトで収集されたデータセットにおいて判別モデルを構築し，他のプロジェクトで収集されたデータセットそれぞれに判別モデルを適用した場合の AUC を表 3 に示す．

表 3 に示す実験結果によると，AUC の平均値は 0.63 であり，77%のケース（64 件中の 49 件）は AUC が 0.5 以上（ランダムに予測した場合の AUC は 0.5）であった．ランダムに予測する場合と比較して，他プロジェクトの実績データに基づく fault-prone モジュール判別が有効であることがわかった．

4.1.1 フィットデータとして利用した場合の結果

表 3 より，判別するデータセットの傾向に依らず性能の高いモデルの構築に寄与するデータセット，つまりフィットデータとして適しているデータセットが存在することがわかった．その一方で，フィットデータとして適していないデータセットが存在することもわかった．例えば，PC3 や KC3 をフィットデータに用いた場合，AUC の平均値がそれぞれ 0.72 と 0.70 であり，AUC が 0.65 以上であるテストデータはそれぞれ 7 件中 5 件と 7 件あった．一方 PC2 をフィットデータに用いた場合，AUC の最高値は 0.57，平均値は 0.45 であり，ランダムに予測した場合とほぼ同じ値であることから，fault-prone モジュール判別に適していないといえる．

4.1.2 テストデータとして利用した場合の結果

表 3 より，どのようなモデルに対しても判別しやすいデータセット，つまりテストデータとして適しているデータセットが存在することがわかった．一方で，テストデータとして適していないデータセットが存在することもわかった．

例えば，PC2 に対する判別結果は，AUC の平均値が 0.77 であり，7 件中 7 件の AUC が 0.65 以上であった．一方 KC4 に対する判別結果は，AUC の最高値が 0.60,

表 3. 予測対象プロジェクト以外の単一のプロジェクトを用いた場合の結果

フィットデータ＼テストデータ	CM1	KC3	KC4	MW1	PC1	PC2	PC3	PC4
CM1	-	0.58	0.40	0.64	0.68	0.76	0.64	0.72
KC3	0.67	-	0.41	0.69	0.71	0.85	0.75	0.82
KC4	0.63	0.80	-	0.75	0.60	0.73	0.65	0.56
MW1	0.79	0.73	0.69	-	0.61	0.68	0.63	0.63
PC1	0.73	0.32	0.23	0.35	-	0.80	0.69	0.67
PC2	0.44	0.29	0.47	0.31	0.51	-	0.59	0.61
PC3	0.72	0.81	0.60	0.74	0.74	0.77	-	0.63
PC4	0.37	0.64	0.47	0.43	0.55	0.82	0.57	-

表 4. 予測対象プロジェクト以外の全てのプロジェクトを用いた場合の結果

テストデータ	CM1	KC3	KC4	MW1	PC1	PC2	PC3	PC4
-	0.67	0.82	0.46	0.74	0.72	0.87	0.76	0.63

平均値は 0.47 であり，ランダムに予測した場合とほぼ同じ値であることから，fault-prone モジュール判別に適していないといえる．

4.1.3 フィットデータとテストデータの組み合わせに着目した結果

表 3 より，それぞれのデータセットの組み合わせに着目した場合，どちらをモデル構築に用いるかで判別精度に差がある組み合わせと，差がない組み合わせがあった．例えば，PC2 をフィットデータとして用いて KC3 内の fault-prone モジュール判別を試みても AUC は 0.26 であったが，KC3 をフィットデータとして用いて PC2 内の fault-prone モジュール判別を試みると AUC は 0.85 であった．一方で，KC3 と MW1 をフィットデータとテストデータにそれぞれ組み合わせた場合，AUC の値には差がほとんどなかった（0.73，0.69）．

また，フィットデータに適しているからといってテストデータに適しているわけではなく，テストデータに適しているからといってフィットデータに適しているわけではないということがわかった．例えば，KC4 をフィットデータとして用いた場合，自身を除くデータセットをある程度の判別精度（AUC の平均値が 0.64）で予測できたが，KC4 をテストデータとして用いた場合，低い精度（AUC の平均値が 0.47）でしか予測できなかった．一方，PC2 をテストデータとして用いた場合，自身を除くデータセットから作られたモデルによって高い精度（AUC の平均値が 0.77）で判別されるが PC2 をフィットデータとして用いた場合，低い精度（AUC の平均値が 0.46）でしか予測できなかった．

4.2. 複数のデータセットを用いた場合の実験結果

他のプロジェクトで収集したデータセットすべてを 1 つのフィットデータとし，判別モデルを構築した場合の実験結果を表 4 に示す．

表 4 に示す実験結果によると，AUC の平均値は 0.71 であり，8 ケース中の 7 ケースは AUC が 0.5 以上であった．ランダムに予測する場合と比較して，他プロジェクトの実績データに基づく fault-prone モジュール判別が有効であることがわかった．

単数のデータセットを用いた場合と同様，複数のデータセットを用いた場合にも予測しづらい（テストデータに適していない）データセットが存在した．KC4 では，単数のデータセットを用いた場合の AUC の平均値が 0.47 であり，複数のデータセットを用いた場合の AUC が 0.46 であった．

一方，単数のデータセットを用いた場合には特に予測しやすいわけではなかったが，複数のデータセットを用いた場合には予測しやすい（テストデータに適している）データセットが存在した．KC3 では，単数のデータセットを用いた場合には AUC が 0.60 であったが，複数のデータセットを用いた場合の AUC は 0.82 であった．

5. おわりに

本稿では，他のプロジェクトのモジュールから計測された実績データを元にモデル構築を行い，fault-prone モジュールの判別を試みた．実験では，NASA IV&V Facility MDP が公開しているデータセットの内，8 個のデータセットを用い，自身を除くそれぞれの組み合わせ計 56 通りと，他の全プロジェクトのデータセットをフィットデータとした場合の 8 通りについて予測モデルを構築しその性能を実験的に評価した．

実験により得られた主な結果および知見は，以下の通りである．

- ランダムに予測する場合と比較して，他プロジェクトの実績データに基づく fault-prone モジュール判別のほうが有効である．
- どのようなモデルに対しても予測しやすいデータセット，つまりテストデータとして適しているデータセットが存在する．
- 予測するデータセットの傾向に依らず性能の高いモデルの構築に寄与するデータセット，つまりフィットデータとして適しているデータセットが存在する．
- フィットデータに適したデータセットと，テストデータに適したデータセットの間には相関が少ない．
- 複数プロジェクトで収集したデータセットを混ぜてから判別モデルを構築することは有効である．

Fault-prone モジュール判別モデルにとって，どういったデータセットがフィットデータやテストデータに適しているのか，その原因について検討していくことが今後の課題である．

謝辞

本研究の一部は、文部科学省「e-Society 基盤ソフトウェアの総合開発」の委託に基づいて行われた．

参考文献

[1] Andrew R. Gray and Stephen G. MacDonell, “Software Metrics Data Analysis – Exploring the Relative Performance of Some Commonly Used Modeling Techniques”, Empirical Software Engineering, Vol.4, No.4, pp.297–316, 1999

[2] （独）情報処理推進機構ソフトウェア・エンジニアリング・センター，ソフトウェア開発データ白書 2006—IT 企業 1400 プロジェクトの定量データで示す開発の実態，日経 BP 社，2006

[3] Ohlsson N. and Alberg H., “Predicting Fault-Prone Software Modules in Telephone Switches”, IEEE Trans. on Software Engineering, Vol.22, No.12, Dec, 1996

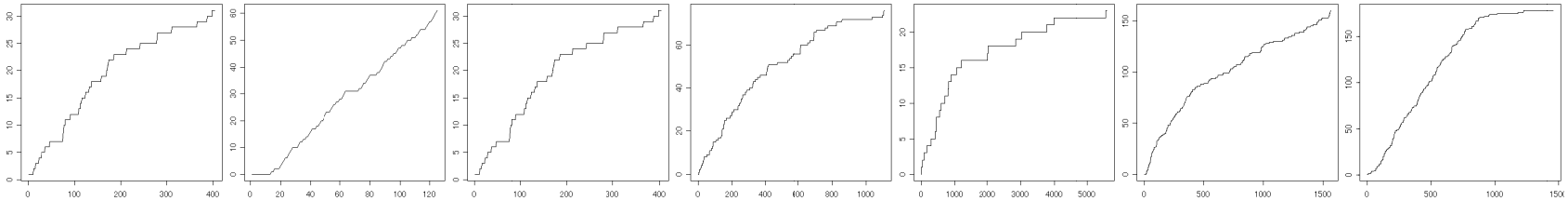
[4] NASA IV&V Facility, Metrics Data Program, http://mdp.ivv.nasa.gov/

[5] Fisher R. A., “The Use of Multiple Measurements in Taxonomic Problems”, Annals Eugenics, Vol.7, Part II, pp.179-188, 1936

[6] Rumelhart, D. E., Hinton, G. E. and Williams, R. J., “Learning Representations by Back-propagating Errors”, Nature, Vol.323, pp.533-536, 1986

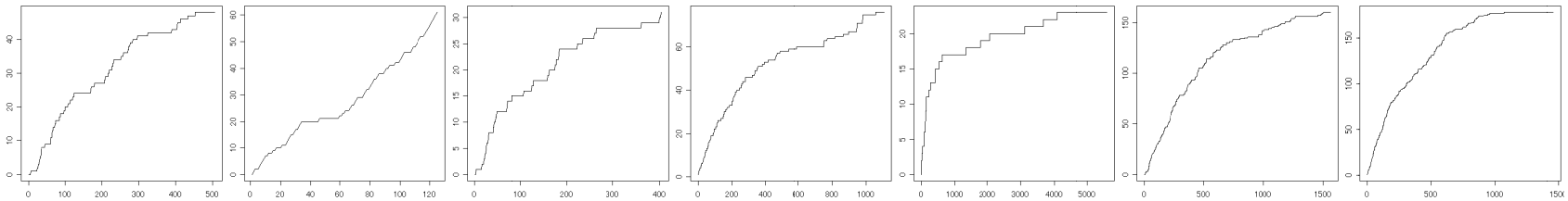
Appendix. A：単数のデータセットを用いた場合それぞれの Alberg Diagram

フィットデータ：CM1



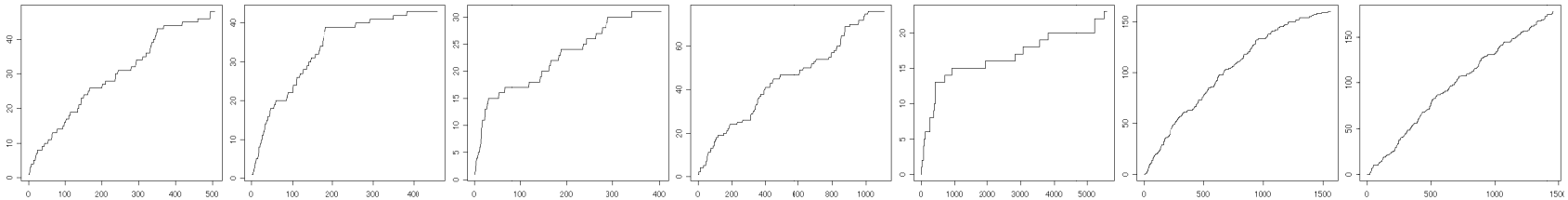
テストデータ：KC3　テストデータ：KC4　テストデータ：MW1　テストデータ：PC1　テストデータ：PC2　テストデータ：PC3　テストデータ：PC4

フィットデータ：KC3



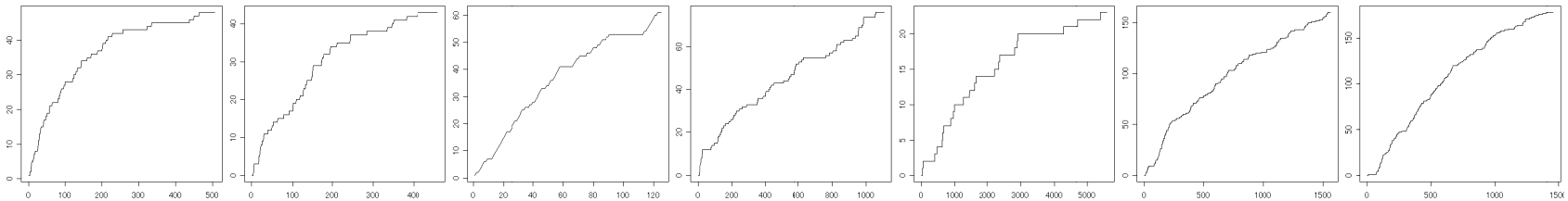
テストデータ：CM1　テストデータ：KC4　テストデータ：MW1　テストデータ：PC1　テストデータ：PC2　テストデータ：PC3　テストデータ：PC4

フィットデータ：KC4



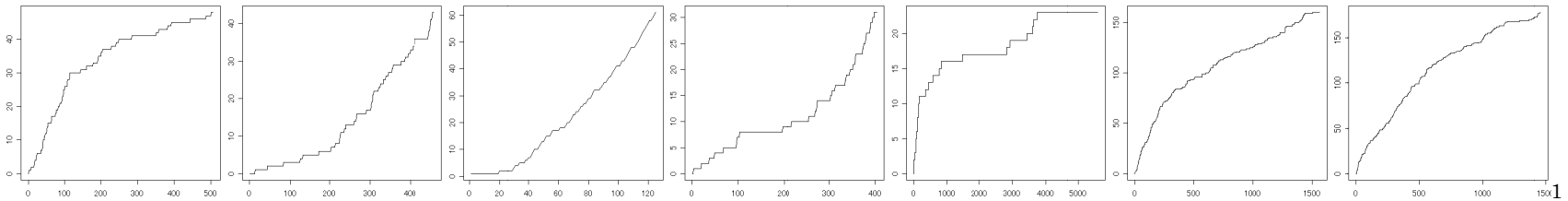
テストデータ：CM1　テストデータ：KC3　テストデータ：MW1　テストデータ：PC1　テストデータ：PC2　テストデータ：PC3　テストデータ：PC4

フィットデータ：MW1



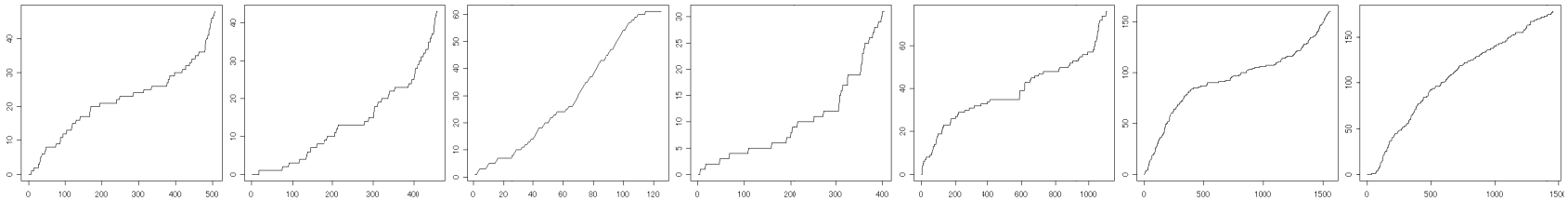
テストデータ：CM1　テストデータ：KC3　テストデータ：KC4　テストデータ：PC1　テストデータ：PC2　テストデータ：PC3　テストデータ：PC4

フィットデータ：PC1



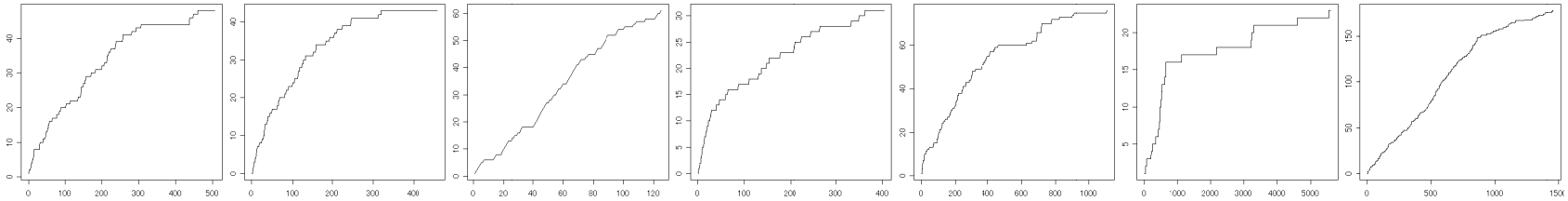
テストデータ：CM1　テストデータ：KC3　テストデータ：KC4　テストデータ：MW1　テストデータ：PC2　テストデータ：PC3　テストデータ：PC4

フィットデータ：PC2



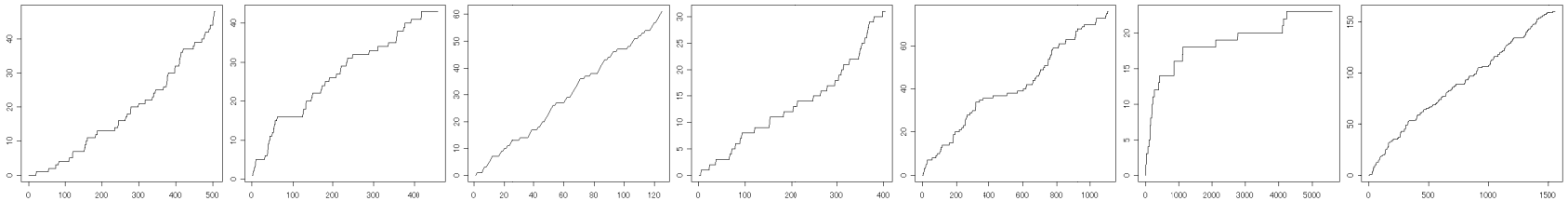
テストデータ：CM1　テストデータ：KC3　テストデータ：KC4　テストデータ：MW1　テストデータ：PC1　テストデータ：PC3　テストデータ：PC4

フィットデータ：PC3



テストデータ：CM1　テストデータ：KC3　テストデータ：KC4　テストデータ：MW1　テストデータ：PC1　テストデータ：PC2　テストデータ：PC4

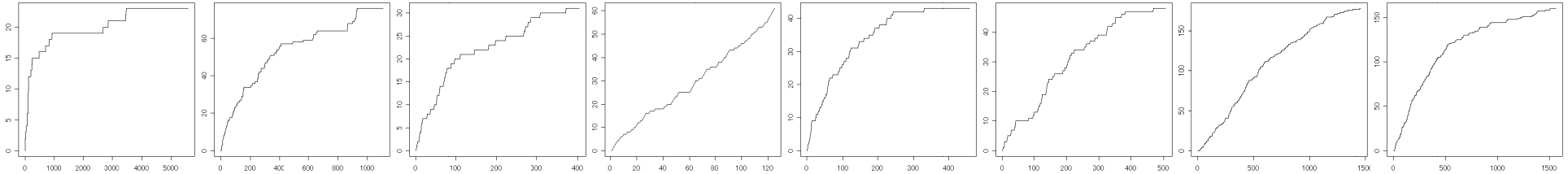
フィットデータ：PC4



テストデータ：CM1　テストデータ：KC3　テストデータ：KC4　テストデータ：MW1　テストデータ：PC1　テストデータ：PC2　テストデータ：PC3

Appendix. B：複数のデータセットを用いた場合それぞれの Alberg Diagram

フィットデータ：テスト対象データ以外全部



テストデータ：CM1　テストデータ：KC3　テストデータ：KC4　テストデータ：MW1　テストデータ：PC1　テストデータ：PC2　テストデータ：PC3　テストデータ：PC4