社団法人 電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
TECHNICAL REPORT OF IEICE.
SS2003-27, KBSE2003-30 (2003-11)

協調フィルタリングに基づくソフトウェア信頼性予測方法

角田 雅照† 大杉 直樹† 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 〒630-0192 けいはんな学研都市

E-mail: †{masate-t, naoki-o, akito-m, matumoto}@is.aist-nara.ac.jp

あらまし ソフトウェア開発プロジェクトにおいて記録されたデータには，しばしば欠損値が含まれる．欠損値が非常に多く含まれているデータに対しては，重回帰分析を用いて高い精度の信頼性予測を行うことは困難である．この問題を解決するため，本論文では協調フィルタリングに基づくソフトウェア信頼性予測方法を提案する．提案方法では，協調フィルタリングを適用して過去に開発されたソフトウェアの中から類似するものを選び出し，類似ソフトウェアの信頼性から，開発中のソフトウェアの信頼性を予測する．実プロジェクトのデータに対して提案手法を適用し，予測精度を評価した結果，提案手法は欠損値を補う従来の重回帰分析よりも高い性能を示した．

キーワード ソフトウェア信頼性，協調フィルタリング，正規化，コサインによる類似度，重回帰分析，欠損値

An Application of Collaborative Filtering for Software Reliability Prediction

Masateru TSUNODA† Naoki OHSUGI† Akito MONDEN† and Ken'ichi MATSUMOTO†

† Graduate School of Information Science, Nara Institute of Science and Technology
Kansai Science City, 630-0192 Japan

E-mail: †{masate-t, naoki-o, akito-m, matumoto}@is.aist-nara.ac.jp

Abstract Data sets recorded in software development projects often contain a large amount of missing values. It is difficult for multivariate regression models to accurately predict software reliability using such data. To solve this problem, we propose a method to apply Collaborative Filtering for software reliability prediction. In our method, we use Collaborative Filtering to select past software projects similar to a target project and to predict its reliability with selected past projects. We applied our method to actual software projects data to estimate accuracy of prediction. The result of the case study showed that Collaborative Filtering showed better prediction performance than conventional regression models.

Keyword software reliability, collaborative filtering, normalize, cosine similarity, multiple regression, missing value

1. まえがき

ソフトウェア開発プロジェクトにおいては，工数の見積りや欠陥発見数の予測などを行う必要があり，そのための見積りや予測を行うモデルが数多く提案されている[1][3]．見積りや予測を行うためには，あらかじめプロジェクトごとに工数や欠陥発生数といったデータを記録しておく必要がある．

しかし，さまざまな理由で欠損値(メトリクスの記録が欠けていること)が発生することがある．例えば，データの収集方針の変更があげられる．記録するメトリクスの種類をある時点から増加させた場合，古いプロジェクトではそのメトリクスが記録されていないことになる．また，プロジェクトに参加しているメンバーがデータの記録を怠ることにより欠損値が発生することもある．このことは，重回帰分析を用いてモデル作成をする際に問題となる．重回帰分析は多変量解析のひとつであり，ソフトウェアの工数見積りや信頼性予測のモデル作成にしばしば用いられている．しかし，重回帰分析はデータに欠損値を含んでいる場合，データから欠損値を含むケースを除外したり，欠損値に平均値を代入するなどの処理を行わないとモデルを作成することができない．よって，データに欠損値を非常に多く含む場合，重回帰分析の適用が困難になる[8]．

そこで，このような問題を解決するために，本論文では協調フィルタリングを用いた信頼性予測を行う方法を提案する．協調フィルタリングにより，過去に記録されたソフトウェアプロジェクトのデータから，予測対象のプロジェクトと類似したプロジェクトを選び出し，予測値の算出を行う．この際，協調フィルタリングはデータに欠損値が多く含まれている場合でも適用可能である．しかし，これまでに提案された協調フィルタリングのアルゴリズムは，ソフトウェアの信頼性予測に対して適用することを前提としていないため，その適用可能性や有用性が明らかでない．本論文では協調フィルタリングをソフトウェアの信頼性予測に対して適用し，予測精度を重回帰分析と比較することにより，適用可能性や有用性を明らかにすることを目的とする．ただし，従来の協調フィルタリングは，例えば 1～5 までの値しかとらないような同一単位を持つデータを対象としているものであり，ソフトウェアプロジェクトのような，メトリクスごとに単位や値域が異なるものにそのまま適用することは困難である．そこで，協調フィルタリングを信頼性予測に適用するために，メトリクスの正規化とメトリクスの正規化解除を行う方法を提案する．また，予測値の算出アルゴリズムについても新たに提案する．

以降，2.では従来研究について説明をする．3.では本論文で提案する協調フィルタリングを用いた予測手順を述べる．4.では欠損値を含むデータに重回帰分析と協調フィルタリングを適用した実験について述べる．5.では，予測精度について考察を行う．6.ではまとめを述べる．

2. 関連研究

欠損値のあるデータに対する重回帰分析の適用については，Strike ら [8]の研究がある．しかし，[8]で述べられているように，欠損値が全体の 40%を超えるデータに対しては，一般に重回帰分析の適用は困難となる．

また，予測対象のプロジェクトと類似したプロジェクトを選び出し，予測値の算出を行う方法については，Shepperd ら[7]の研究がある．[7]らの研究では，類似度の算出に Euclid 距離を用いている．しかし，欠損値については考慮されていない．一方，協調フィルタリングは欠損値の多いデータに対して有効であるといわれている．

協調フィルタリングは，多量に存在する情報からそれぞれのユーザの好みに合うであろうと考えられる情報を選出し，推薦するために利用されている．推薦する相手，及び推薦する情報をシステムが自動的に決定する方式が数多く提案されているが，その最初のシステムは Resnick ら[6]の GroupLens であると言われている[5]．GroupLens は Usenet ニュース の記事の集合から，ユーザの好みにあう記事を推薦するシステムである．ユーザは GroupLens に推薦された記事に対し，5(良い)から 1(悪い)の 5 段階で評価を行う．システムは，その評価を基にユーザ間の類似度を求める．ユーザが高い評価を行った記事は，類似度の高い他のユーザにも推薦される．すなわち，「ある記事に対して同様の評価をするユーザ同士は，他の記事に関しても同様の評価をするであろう」と仮定して，自分と類似した評価をしているユーザが，高い評価をしている記事の推薦を行うのである．本論文では，この GroupLens の考え方をソフトウェアプロジェクトの信頼性予測に適用し，「類似するプロジェクト同士は，欠陥発見数も似ているだろう」と考え，プロジェクト同士の類似度を算出し予測を行う．GroupLens で提案されている協調フィルタリングアルゴリズムは，多くの研究の基礎となっており，GroupLens の方法から派生した多くのアルゴリズムが提案されている．本論文においても，GroupLens のアルゴリズムを基礎として，ソフトウェア予測に適用することを前提としたアルゴリズムを提案する．しかし，推薦システムでは各投票は 1 から 5 などの値をとる同一の単位を持つものであるのに対し，ソフトウェアプロジェクトでは，メトリクスごとに単位が異なる．よって，類似度計算を行う前にメトリクスの正規化を行う必要がある．なお，ソフトウェアの信頼性予測に適用することを前提とした協調フィルタリングアルゴリズムは，従来提案されていない．

3. 予測手順

本論文で提案する協調フィルタリングを用いた予測手順について述べる．従来の協調フィルタリングを用いた推薦システムにおけるユーザをプロジェクト，記事をメトリクス，投票をメトリクス値とみなすことで，協調フィルタリングをソフトウェアの信頼性予測に用いることができる．ここでは，プロジェクト単位で予測を行うものとする．多くのプロジェクトがデータベースに記録されているものとして考える．データベース中の i 番目のプロジェクトを p_i と表し，データベースにはプロジェクト $p_1, p_2, \ldots, p_m$ が記録されている．各プロジェクトはプロジェクトで計測されたメトリクスを保持している．プロジェクト p_i の j 番目のメトリクスを $m_{i,j}$ と表し，p_i でメトリクス $m_{i,1}, m_{i,2}, \ldots, m_{i,n}$ が計測されているものとする．ここでは，プロジェクト p_u の j 番目のメトリクス $m_{u,j}$ を予測することを考える．このプロジェクトをアクティブプロジェクトと呼ぶ．以下の手順に従ってソフトウェアの信頼性予測を行う．

[ステップ 1](メトリクスの正規化) メトリクス値を変換し，値を[0.0, 1.0]の範囲の実数に正規化するために以下の計算を行う．プロジェクト p_i の j 番目のメトリクス $m_{i,j}$ の正規化された値 $normalized(m_{i,j})$は以下の式によって求める．

$$normalized(m_{i,j}) = \frac{m_{i,j} - \min(P_j)}{\max(P_j) - \min(P_j)} \quad (1)$$

ここで，P_j は j 番目のメトリクス値が欠損していないプロジェクトの集合を表す．また $\max(P_j)$並びに $\min(P_j)$はそれぞれ P_j に含まれるプロジェクト p_i のメトリクス $m_{i,j}$ の中で最も大きい値，並びに，最も小さいものを示す．

[ステップ 2](プロジェクト間類似度計算) プロジェクト間の類似度計算は，各プロジェクトをベクトルとみなし，ステップ 1 で正規化した各メトリクスをベクトルの要素とみなすことによって行う．そして 2 つのプロジェクトのベクトルのなす角のコサインを求め，それをプロジェクト間の類似度とする．このとき，欠損値を含むメトリクスはベクトルの要素に含まず，計測されているメトリクスのみをベクトルの要素とする．よって，プロジェクトのメトリクスに欠損値が含まれていても，類似度計算が可能となる．

プロジェクト p_u 並びに p_i において，メトリクス $m_{u,1}, m_{u,2}, \ldots, m_{u,n}$ 並びに $m_{i,1}, m_{i,2}, \ldots, m_{i,n}$ が計測されているとき，p_u と p_i の間の類似度 $sim(p_u, p_i)$は以下の式によって求める．

$$sim(p_u, p_i) = \frac{\sum_{j \in M_u \cap M_i} \left(normalized(m_{u,j}) \times normalized(m_{i,j}) \right)}{\sqrt{\sum_{j \in M_u \cap M_i} \left(normalized(m_{u,j})^2 \right)} \sqrt{\sum_{j \in M_u \cap M_i} \left(normalized(m_{i,j})^2 \right)}} \quad (2)$$

ここで，M_u 並びに M_i は，それぞれプロジェクト p_u 並びに p_i から測定されたメトリクスの集合を表す．

[ステップ 3](メトリクス予測) メトリクスの予測はステップ 2 で求めたプロジェクト間の類似度，並びに，正規化されたメトリクス値 $normalized(m_{u,j})$を用いて，加重平均を行う．アクティブプロジェクト p_u の j 番目のメトリクスの正規化されたメトリクス値 $normalized(\hat{m}_{u,j})$は以下の式によって求める．

$$normalized(\hat{m}_{u,j}) = \frac{\sum_{i \in k\text{-}nearestProjects} \left(sim(p_u, p_i) \times normalized(m_{i,j}) \times amplifier(p_u, p_i) \right)}{\sum_{i \in k\text{-}nearestProjects} sim(p_u, p_i)} \quad (3)$$

ただし，$k\text{-}nearestProjects$はメトリクス m_j が測定されており，かつ，プロジェクト p_u に対して類似度が最も高い k 個のプロジェクトの集合を示す．$Neighborhood$

*size k*は見積の正確さに影響を与えるため，アルゴリズムの評価実験において考慮すべき要因である．また，*amplifier*(p_a,p_i)は次式で求める．

$$amplifier(p_a,p_i)=\frac{\sum_{j\in M_a\cap M_i}\left(\frac{normalized(m_{a,j})}{normalized(m_{i,j})}\right)}{|M_a\cap M_i|} \quad (4)$$

amplifier(p_a,p_i)は，プロジェクト p_i のメトリクス値 $m_{i,j}$ を増幅する乗数である。本論文では，類似度 *sim*(p_a, p_i)をベクトル計算によって求める．プロジェクト p_i の類似度が高い場合，p_a と p_i を表すベクトルのなす角は小さく，メトリクスの傾向は似ている可能性が高い．しかし，ベクトルの長さは異なる可能性があるため，メトリクスの絶対値は似ているとは限らない．式4はプロジェクト p_a のメトリクスが，平均的にプロジェクト p_i の何倍であるかを求める式である．

[ステップ4](メトリクス正規化解除) ステップ3で求めたノーマライズされたメトリクスの予測値 *normalized*($\hat{m}_{a,j}$)から，メトリクスの予測値 $\hat{m}_{a,j}$ を次式で求める．

$$\bar{m}_{a,j}=normalized(\hat{m}_{a,j})\times(\max(P_j)-\min(P_j))+\min(P_j) \quad (5)$$

ただし，P_j は j 番目のメトリクスについて計測を行ったプロジェクトの集合を表す．これはステップ1と逆の計算を行っている．

4. 評価実験

重回帰分析と協調フィルタリングの予測精度の比較を行うために実験を行った．

4.1. 実験用データ

評価実験には，企業のソフトウェア開発プロジェクトで記録，収集されたデータを使用した．これらのデータは特定の数で除算されているため，人月などの単位を持たない．各プロジェクトについては，表1のような18個のメトリクスが含まれている．

評価実験では，試験工数(Y_e)と統合試験で発見された欠陥(fault)の数(Y_f)の予測を行った．これらのメトリクスは，試験工程で可能な限り欠陥を除去し，ソフトウェアの信頼性を高めるために重要なものである．なお，上述したように，特定の数で除算されているために Y_e，Y_f とも整数とはならない．

Y_e の予測の評価実験における説明変数の候補として，試験工程までに得られる X_1 ～X_4，X_6 ～X_{14} の変数を対象とした．Y_f の予測の評価実験における説明変数の候補としては，統合試験までに得られる X_1 ～X_{16} の変数を対象とした．

また，これらのデータには欠損値が多く含まれる．そのため，実験に使用できるデータ件数が Y_e の予測の場合は1081件，Y_f の予測の場合は415件となった(例えば，試験工数が記録されていないプロジェクトは，試験工数の評価実験に使用できない)．それぞれのデータから半数のプロジェクトデータを無作為に抽出し，予測モデル作成のためのトレーニングデータとした．残りのプロジェクトを，作成したモデルの予測精度の評価のためのテストデータとした．この作業を10回繰り返し，テストデータとトレーニングデータを Y_e と Y_f それぞれに10組作成した．

Y_e，Y_f の予測実験のデータにおいて，各変数に欠損値が含まれる割合を表1に示す．また，Y_e の予測実験のデータ全体に含まれる欠損値の割合は44%，Y_f の予測実験のデータ全体に含まれる欠損値の割合は58%であった．また，欠損値を全く含まないプロジェクトは，Y_e，Y_f の予測実験のデータどちらも19件であった．

表1 実験に用いたデータ

変数	説明	欠損値の割合	
		Y_e 予測実験時	Y_f 予測実験時
X_1	汎用システム(汎用システムならば1，それ以外ならば0とする)	76%	80%
X_2	新規開発(新規開発ならば1，それ以外ならば0とする)	7%	4%
X_3	設計工数	0%	2%
X_4	製造工数	0%	2%
X_5	開発規模(ステップ数)	-	0%
X_6	設計工数(正社員分)	87%	85%
X_7	設計工数(派遣社員分)	87%	85%
X_8	設計工数(外注分)	87%	85%
X_9	製造工数(正社員分)	87%	85%
X_{10}	製造工数(派遣社員分)	87%	85%
X_{11}	製造工数(外注分)	87%	85%
X_{12}	基本設計レビューで発見された欠陥(件/頁．発見数を仕様書の頁数で除算)	84%	57%
X_{13}	詳細設計レビューで発見された欠陥(件/頁)	71%	24%
X_{14}	プログラム設計レビューで発見された欠陥(件/頁)	80%	48%
X_{15}	コーディング・単体試験で発見された欠陥の数	-	15%
X_{16}	結合試験で発見された欠陥の数	-	10%
Y_e	試験工数	0%	-
Y_f	統合試験で発見された欠陥の数	-	0%

なお，予測時のデータの正規化は，トレーニングデータにテストデータ中のアクティブデータを1件加えた状態で行われ，アクティブデータが代わるごとにデータの正規化の手順から再実行している．

4.2. 評価基準

実験の評価基準には，以下の5つを使用する．

① 予測誤差の絶対値(AE)の平均．予測誤差の絶対値(AE)は以下の式によって求める．

$$AE=|\hat{Y}-Y| \quad (6)$$

ここで，Y は実測値，$\hat{Y}$ は予測値を表す．

② 予測誤差の絶対値(AE)の分散．

③ 相対誤差の絶対値(RE)の平均．相対誤差の絶対値(RE)は以下の式によって求める[2]．

$$RE=\frac{|\hat{Y}-Y|}{Y} \quad (7)$$

なお，実測値が0のデータは相対誤差の算出時には除外し，別途AEの平均値を求めることとする．

④ 相対誤差の絶対値(RE)の分散．

⑤ 相対誤差の絶対値(RE)が25%以内に収まっているデータの割合[2]．REが25%以下のデータ件数をトレーニングデータの件数で除して求める．

表 2 Y_e予測時の各評価基準(10組のデータの平均値)

	AE		RE		RE25%
	平均	分散	平均	分散	以内
重回帰分析(リストワイズ除去)	*0.70*	*16.45*	*30.22*	*287581.18*	*10%*
(*P*値)	*(0.0020)*	*(0.0020)*	*(0.0020)*	*(0.0059)*	*(0.0020)*
重回帰分析(ペアワイズ除去)	52.75	*97171.84*	*6344.27*	*18937623097.60*	*12%*
(*P*値)	(0.020)	*(0.0020)*	*(0.0059)*	*(0.0020)*	*(0.0020)*
重回帰分析(平均値を代入)	*1.33*	*5.07*	*331.69*	*24208218.49*	*4%*
(*P*値)	*(0.0020)*	*(0.0059)*	*(0.014)*	*(0.0020)*	*(0.0020)*
協調フィルタリング	0.21	2.20	0.82	3.45	36%

表 3 Y_f予測時の各評価基準(10組のデータの平均値)

	AE		RE		AEの平均	RE25%
	平均	分散	平均	分散	(実測値 0)	以内
重回帰分析(リストワイズ除去)	*722.06*	*33584317.31*	*50.13*	*79193.33*	*835.07*	*5%*
(*P*値)	*(0.160)*	*(0.160)*	*(0.0020)*	*(0.0020)*	*(0.0020)*	*(0.0020)*
重回帰分析(ペアワイズ除去)	*1105676.92*	*32010965443737.10*	*169957.34*	*1194754878665.11*	*864100.70*	*0%*
(*P*値)	*(0.0020)*	*(0.0020)*	*(0.0020)*	*(0.0020)*	*(0.0020)*	*(0.0059)*
重回帰分析(平均値を代入)	*30935.13*	850883043.14	*13467.23*	*12753574078.88*	*29829.90*	*2%*
(*P*値)	*(0.0020)*	(1)	*(0.0020)*	*(0.0020)*	*(0.0020)*	*(0.0020)*
協調フィルタリング	554.06	63352285.04	1.79	47.25	6.08	14%

ソフトウェアプロジェクトに関する予測モデルの精度評価にはREが用いられることが多いが(例えば[7][8]など)，重回帰分析はAEが最小になるようにモデルを作成するため，AEの比較も行った．

4.3. 重回帰分析による予測

4.1で述べたY_eと Y_f 各10組の実験データを用いて，Y_eと Y_fの線形予測モデルを作成した．重回帰分析の際には変数増減法を用いた．変数増減法は，説明変数を選択して最良の回帰モデルを探索する手法の一つであり，回帰係数が0かどうかの検定を利用している[7]．このとき，5%の有意水準で係数が0でないといえるときに変数を取り込み，10%の有意水準で係数が0でないとはいえないときに変数を除去した．

説明変数の候補全てを用いて，変数増減法による重回帰分析を行った．このとき，欠損値の処理は一般的に行われる以下の3つの方法で行い，これらの方法ごとにモデルを作成した．

① 欠損値を1つでも含むプロジェクトは除外する(リストワイズ除去)
② 2つの変数間の相関を求める際，2つの変数に欠損地が含まれていなければ計算に用いる．この時，他の変数に欠損値が含まれていてもよい(ペアワイズ除去)
③ ある変数に欠損値が含まれる場合，その変数の平均値を代入する

上記のようにして，1つのデータに対し3つのモデルを作成した．これらのモデルによるY_e予測時の各評価基準値の平均を表 2に，Y_f 予測時の各評価基準値の平均を表 3に示す．また，横軸に予測値，縦軸に実測値をとったグラフ(10組の実験データのうちのある1つに適用したもの)を図 1，図 2に示す．各点が$y = x$の直線に近いほど予測精度が高い．なお，ペアワイズ除去による Y_f予測モデル作成は，5%の有意水準で係数が0でないといえる変数がなかったため，モデルを作成できなかったデータが2組あった．これらのデータについては，モデルが作成されるように16%の有意水準で係数が0でないといえるときに変数を取り込み，17%の有意水準で係数が0でないとはいえないときに変数を除去した．

4.4. 協調フィルタリングによる予測

次に協調フィルタリングによる Y_e, Y_fの予測を行った．最初に適切な *neighborhood size k* を決定するために，4.1で述べた Y_eと Y_f 各10組の実験データに対しkを1から50まで変化させて予測を行った．kごとにREの平均を取ると，Y_eの予測時は図 1(e)，Y_fの予測時は図 2 (e)のようになった．そこでREの平均が最も小さかったkを採用し，予測を行った．Y_eの予測時はk=22，Y_fの予測時はk=1とした．協調フィルタリングによるY_e予測時の各評価基準値の平均を表 2に，Y_f 予測時の各評価基準値の平均を表 3に示す．また，横軸に予測値，縦軸に実測値をとったグラフ(10組の実験データのうちのある1つに適用したもの)を図 1，図 2に示す．

4.5. 予測精度の比較

重回帰分析と協調フィルタリング予測精度を比較するために，各評価基準に差があるかどうかの検定を行った．それぞれのモデルは同じデータに対して作成されている．それを10組のデータに対して行っているので，対応のあるデータに対する検定であるWilcoxonの符号付順位和検定を行った[10]．これは2つの変数の代表値に差があるかどうかを検定するノンパラメトリックな手法である．協調フィルタリングと対照させて重回帰分析の各手法と比較した．そのため，1つの評価基準に対して3回検定を行った．多重比較を繰り返すと第1種の過誤が起きやすくなるため，Bonferroniの方法により有意水準の調整を行った．Bonferroniの方法は有意水準を5%としたい場合，5%を検定回数で割ったものを各検定の有意水準として設定するもので

あり[10], この場合 0.05/3=0.0166 を 3 回の各検定の有意水準に設定した.

Y_eに関する検定結果を表 2に, Y_fに関する検定結果を表 3に示す. 斜体の文字が有意水準 1.25%で代表値に差があるといえたものである. 検定結果をみると, ほとんどの評価基準に有意な差があった. すなわち, 重回帰分析と比較して, 欠損値のあるデータに対するソフトウェア信頼性予測に協調フィルタリングを適用することが可能であるとともに,有用であるといえる.

5. 考察

協調フィルタリングのほうが予測精度が高い理由について考察する. 重回帰分析は 1 つのモデルで全ての予測値を算出しようとする. 欠損値を全く含まない場合は問題がないが, 予測値算出時に, 説明変数に欠損値が含まれている場合, その説明変数の項は 0 になってしまうため, 予測精度が低下してしまうと考えられる. 協調フィルタリングは予測対象のプロジェクトごとにトレーニングデータから予測値を算出し, 1 つだけのモデルで予測を行わない. しかも, 予測対象プロジェクトのデータ(テストデータ)も予測に用いるので, 予測対象プロジェクトに欠損値が含まれていることを予測値に反映できるのに対し, 重回帰分析はモデル作成時には予測対象プロジェクトのデータを一切用いないので, 欠損値により影響されやすいと考えられる.

Y_eと Y_fの予測精度を比較すると, Y_eのほうが予測精度が回帰分析, 協調フィルタリングとも高くなっている. これは Y_e の予測値に影響が大きいと考えられる X_3, X_4がほとんど欠損値がなく記録されているためであると考えられる. これに対し, Y_fでは予測値に影響の大きい変数に欠損値が多い, または表 1の変数だけでは精度の高い予測が困難であるためと考えられる.

検定結果について考察すると,Y_eの予測時において, 協調フィルタリングを適用した場合と, ペアワイズ除去による重回帰分析の AE の平均の差には有意な差があるとはいえなかった. しかし, AE 平均の平均値を比較すると大きく異なっている. これは, ペアワイズ除去による重回帰分析は予測精度が高いこともあれば(図 1(b)), 大幅に外れてしまうこともあったからである. すなわち, ペアワイズ除去は欠損値の多いデータに対し, 他の重回帰分析の欠損値処理法よりも常に高い予測精度のモデルを作成するわけではないと考えられる. また, Y_fの予測時において, 協調フィルタリングを適用した場合と, 平均値代入による重回帰分析の AE の分散の差には有意な差があるとはいえなかった. しかし, 図 2(c)をみると平均値代入による重回帰分析のほうが予測精度が高いわけではなく, むしろほとんど予測できていないことがわかる.

各評価基準の平均値で見れば, 協調フィルタリングの次にリストワイズ除去による重回帰分析が優れている. これは極端に予測精度の低いモデルがなかったことによるものである. リストワイズ除去は欠損値の多いデータに対し, 他の重回帰分析の欠損値処理法と比較して極端に低い予測精度のモデルは作成しないと考えられる.

6. むすび

本論文では, ソフトウェアの信頼性予測に協調フィルタリングを用いるための手法を提案した. 提案した手法を企業のソフトウェア開発プロジェクトのデータに適用した結果より, 協調フィルタリングが欠損値を含むデータに対し, 線形予測モデルと比較して精度の高い予測をすることを示した.

今後の課題は, メトリクスの正規化や類似度計算, メトリクスの予測方法の改善を行うことである. 例えば, 本論文で提案した手法は類似度計算時に全ての変数を使用しているが, 使用する変数を絞り込んで類似度を算出し, 精度を向上させることを考えている.

文　献

[1] B. W. Boehm, "Software Engineering Economics," IEEE Trans. on Software Engineering, vol.10, no.1, pp.4-21, Jan. 1984.

[2] S. D. Conte, H. E. Dunsmore, and V. Y. Shen, Software Engineering Metrics and Models, The Benjamin/Cummings Publishing Company, Inc., Menlo Park, California, 1986.

[3] M. Halstead, Elements of Software Science, Elsevier, New York, 1977.

[4] B. A. Huberman, and M. Kaminsky, "Beehive: A System for Cooperative Filtering and Sharing of Information," Computer Human Interaction, pp.210-217 (1996).

[5] D. M. Pennock, E. Horvitz, S. Lawrence, and C. L. Giles, "Collaborative Filtering by Personality Diagnosis: A Hybrid Memory-and Model-Based Approach," Proc. 16th Conf. on Uncertainty in Artificial Intelligence (UAI-2000), pp.473-480, Stanford, California, United States, June-July 2000.

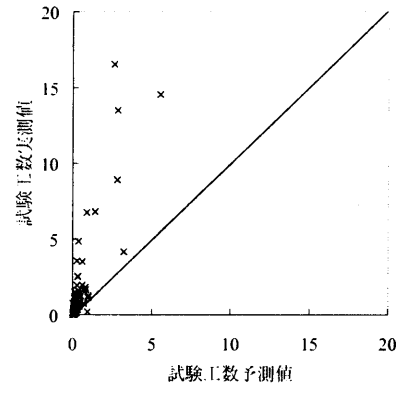
[6] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "GroupLens: An Open Architecture for Collaborative Filtering of Netnews," Proc. ACM Conf. on Computer Supported Cooperative Work (CSCW '94), pp.175-186, Chapel Hill, North Carolina, United States , Oct. 1994.

[7] M. Shepperd, C. Schofield, "Estimating Software Project Effort Using Analogies," IEEE Trans. on Software Engineering, vol.23, no.12, pp.736-743, Nov. 1997.

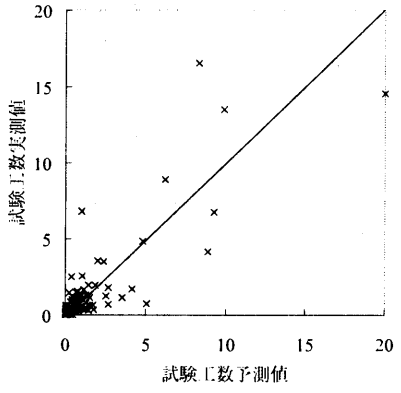
[8] K. Strike, K. E. Emam, N. Madhavji, "Software Cost Estimation with Incomplete Data," IEEE Trans. on Software Engineering, vol.27, no.10, pp.890-908, Oct. 2001.

[9] Windows 版 統計解析ハンドブック 多変量解析, 田中豊, 垂水共之(編), 共立出版, 東京, 1995.

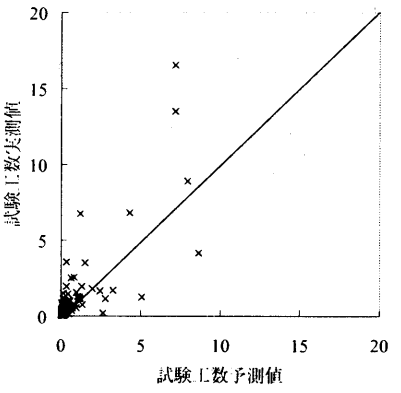
[10] Windows版 統計解析ハンドブック ノンパラメトリック法, 田中豊, 垂水共之(編), 共立出版, 東京, 1999.



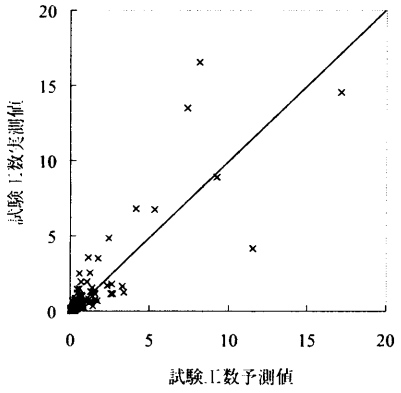
(a) 重回帰分析(リストワイズ除去)



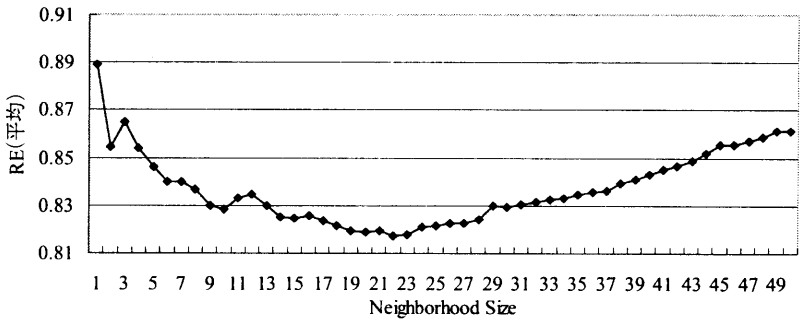
(b) 重回帰分析(ペアワイズ除去)



(c) 重回帰分析(平均値を代入)

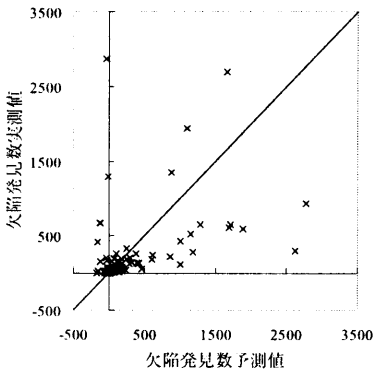


(d) 協調フィルタリング

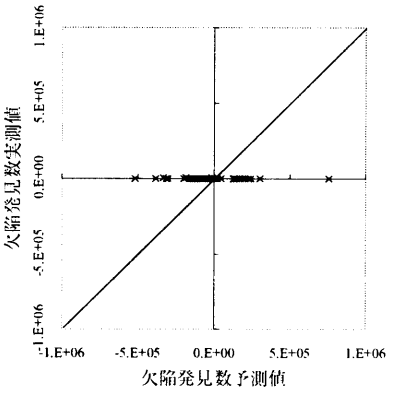


(e) Y_e予測時における neighborhood size と RE の平均の関係

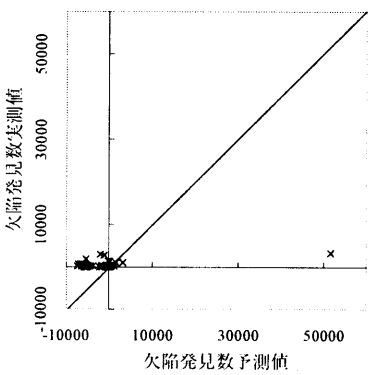
図 1 (a)～(d) Y_e予測時における予測値と実測値の関係



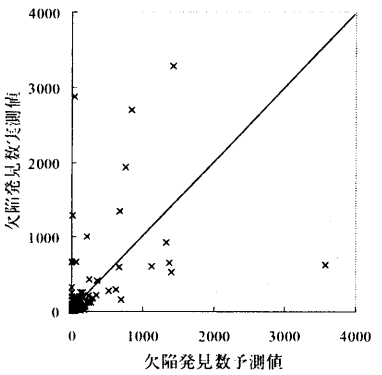
(a) 重回帰分析(リストワイズ除去)



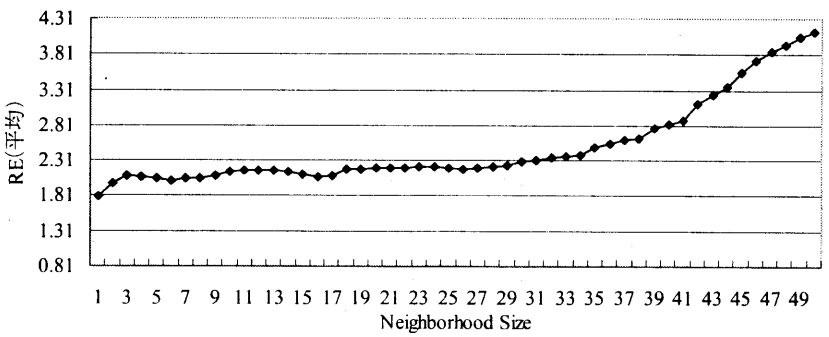
(b) 重回帰分析(ペアワイズ除去)



(c) 重回帰分析(平均値を代入)



(d) 協調フィルタリング



(e) Y_f予測時における neighborhood size と RE の平均の関係

図 2 (a)～(d) Y_f予測時における予測値と実測値の関係