情報処理学会論文誌　データベース　Vol. 1　No. 2　38–53 (Sep. 2008)

カテゴリ型レコードデータからの属性値間の相関性を利用した外れ値検出手法

成田 和世[†1]　北川 博之[†1,†2]

巨大なデータから珍しいイベントや逸脱したオブジェクト，例外等を発見する技術である外れ値検出は，幅広い応用範囲を持つことから，近年ますます注目されている．しかし，既存の外れ値検出手法は連続値を属性値に持つ数値型レコードデータを対象とするものが多い．このような既存外れ値検出手法を用いてカテゴリ型の値を属性値に持つカテゴリ型レコードデータから外れ値を検出することは困難である．そこで，本稿ではこのようなカテゴリ型レコードデータから外れ値となるレコードを検出するためのフレームワークを提案する．我々は，あるレコード内に存在する属性値の集合と本来ならば共起するはずの属性値が，そのレコード内で数多く観測されない場合に，そのレコードを外れ値と見なすこととしている．本稿ではまず，属性間の相関性を表すものとして相関ルールを利用し，外れ値度の式を導入する．また，より高速に外れ値レコードを検出するための効率的アルゴリズムを提示する．実データを用いた実験では，提案手法が関連手法と比較して十分な検出精度を得られること，ナイーブアルゴリズムに比べて提案アルゴリズムの処理速度が数十倍向上することを示す．さらに，提案手法が外れ値として価値のあるレコードを本当に検出できるかどうかを確認するため，実データから外れ値検出を行い，外れ値として得られたレコードの希少性を検証する．

Outlier Detection for Categorical Record Data Considering Associations between Attribute Values

Kazuyo Narita[†1] and Hiroyuki Kitagawa[†1,†2]

Outlier detection, a data mining technique to detect rare events, deviant objects, and exceptions from data, has been drawing increasing attention in recent years. Most existing outlier detection algorithms focus on numerical data sets. It is difficult for such approaches to detect outliers from categorical record databases in which attribute values are categorical. We target categorical record databases and detect records in which many attribute values are not observed even though they should occur in association with other attribute values in the records. To detect such records as outliers, we provide an outlier degree utilizing association rules as associations between attribute values. We also propose an efficient algorithm for detecting such outlier records. Experiments using real data sets show that the proposed method can derive sufficient detection accuracies compared to related works, and that the proposed algorithm runs faster than the brute force algorithm does. Moreover, we introduce interesting records which have been practically detected as outliers from real data sets.

1. はじめに

情報技術の発達にともない，ディジタルデータは巨大化，多様化の一途をたどっている．巨大なデータから規則性や特徴，パターン等を抽出する技術であるデータマイニングは，いまや必要不可欠である．これまで，頻出パターン抽出や分類，回帰分析，クラスタリング等，様々なマイニング技術が提案されてきた[1)–10)]．その中でも，我々は外れ値検出に関する研究を行っている．

外れ値検出は巨大なデータから珍しいイベントや逸脱したオブジェクト，例外等を発見するマイニング技術である．応用例としてネットワークへの不正侵入検知や，オンラインバンクでのなりすまし詐欺の検出，データ中のノイズの発見等があげられ，幅広い分野で活用されている．既存の外れ値検出手法は連続値を属性値に持つ数値型レコードデータを対象としたものが多い[11)–20)]．**図 1** は数値型レコードデータの例と，最も典型的な外れ値検出アプ

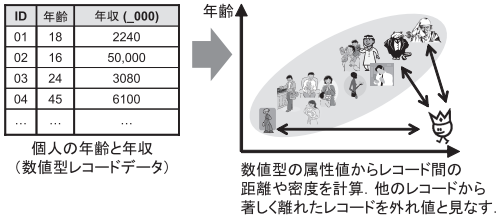


図 1　数値型レコードデータとそれに対する典型的な外れ値検出アプローチの例
Fig. 1　Example of numerical record data and the typical approach for outlier detection.

†1 筑波大学大学院システム情報工学研究科
Graduate School of Systems and Information Engineering, University of Tsukuba
†2 筑波大学計算科学センター
Center for Computational Sciences, University of Tsukuba

© 2008 Information Processing Society of Japan

表 1　カテゴリ型レコードデータの例：オンラインバンクデータ
Table 1　Exsample of categorical record data: data of online bank.

ID	口座番号	IP アドレス	処理	送金区分
001	12345	219.166.42.21	送金	国外
002	12345	219.166.42.21	引落	-
003	23456	111.123.192.1	送金	国内
004	23456	111.123.192.1	送金	国内
005	23456	111.123.192.1	送金	国内
006	23456	111.123.192.1	送金	国内
007	23456	222.123.58.5	送金	国外
008	34567	135.255.255.0	入金	-
⋮	⋮	⋮	⋮	⋮

ローチを図示したものである．図中，左の表は数値型レコードデータの例で，個人の身長と体重に関するデータである．この図が示すように，数値型レコードデータを対象とした多くの既存手法は，属性が持つ連続値からレコード間の距離や密度を計算し，他から著しく離れたレコードを発見し外れ値と見なす．

一方で，世の中にはカテゴリ型の値を属性値に持つ**カテゴリ型レコードデータ**も数多く存在する．たとえば，**表 1** のオンラインバンクデータはカテゴリ型レコードデータである．1 個のレコードが 1 回の取引処理の内容を表している．たとえば ID001 のレコードは，口座番号 “12345” にアクセスした IP アドレス “219.166.42.21” のユーザが，“国外” へ “送金” を行ったことを意味する．このデータから異常な取引を行っているレコードを発見できれば，なりすまし詐欺の検知等につながる．このように，カテゴリ型レコードデータから外れ値を検出する需要は大きい．しかし，図 1 で示されるような数値型レコードデータに特化した多くの既存外れ値検出手法では，多様なカテゴリ型レコードデータに対して汎用的に外れ値検出を行うことは難しい．なぜなら，図 1 のような既存手法をカテゴリ型レコードデータに適用するためには，モデルを構築してカテゴリ型レコードデータを特徴空間へマッピングし，数値データへ変換する必要があるが，データによって最適な特徴空間は異なるので，汎用的なモデルの構築が困難だからである．そこで近年，カテゴリ型レコードデータに焦点を当てた外れ値検出の研究が注目されつつある[21),22)]．

本稿では多様なカテゴリ型レコードデータに対してより汎用的に利用でき，かつ高い検出能力を有する外れ値検出手法を新たに提案する．本稿の寄与する点は大きく 3 つある．以下，各節で概説する．

1.1　外れ値度の導出

本稿では，あるレコード内に存在する属性値の集合と本来ならば共起するはずの属性値が数多く観測されないレコードが外れ値であると考える．たとえば表 1 の ID003 から ID006 のレコードを見ると，口座番号＝“23456”，IP アドレス＝“111.123.192.1”，処理＝“送金”，送金区分＝“国内” の間には強い相関性があることがうかがえる．しかし，ID007 のレコードは口座番号＝“23456”，処理＝“送金” であるにもかかわらず属性 IP アドレスと属性送金区分には異なる値が入っている．このような属性間の強い相関性に反するレコードは，外れ値である可能性が高いと考えられる．この考えに基づき，本稿ではまず属性値間の相関性に着目した外れ値度の式を導出する．属性値間の強い相関性を表すものとして，高い確信度を持つ**相関ルール**を利用する．

1.2　計算処理効率化のためのアルゴリズムの提案

我々の想定する外れ値レコードを検出するための最もナイーブな検出アルゴリズムは，処理にかなりの時間がかかる．そこで本稿では，より効率的な検出アルゴリズムを提示する．提案アルゴリズムは，処理高速化のための 2 つの工夫と，入力データ中に外れ値が存在しない場合に処理を途中で中止する**ファストパス**を取り入れたものである．

1.3　実験による提案手法の有効性の分析

本稿では実データを用いた提案手法の評価実験で，次の 3 つの有効性を示す．(i) 提案手法が実際に外れ値として説得力のあるレコードを検出しうる．(ii) 提案外れ値度は，2 つの関連研究と比較して十分な検出精度を得る力がある．(iii) 提案アルゴリズムは，ブルートフォースアルゴリズムと比べてはるかに高速である．

以下，本稿の構成は次のようである．2 章で本研究に関連する研究について言及する．3 章で本稿で用いる記述や定義について述べる．4 章では属性間の相関性に基づく外れ値度の式を導出する．5 章では，検出処理を途中でやめるファストパスと 2 つの処理高速化の工夫を取り入れた提案アルゴリズムを提示する．6 章で実データを用いた実験とその結果を示す．7 章では実験結果をふまえたうえで，提案手法の性質や利点について議論する．8 章でまとめる．

2. 関 連 研 究

これまで，数値を属性値に持つ数値型レコードデータを対象とした外れ値検出手法が数多く提案されてきた[11)–20)]．これらはカテゴリ値を属性値に持つカテゴリ型レコードデータに焦点を置く本研究とは，着目しているデータが異なっている．これらの手法でカテゴリ型レ

© 2008 Information Processing Society of Japan

コードデータから外れ値を検出することは不可能ではないが，1 章でも述べたようにモデルを構築して数値データに変換する必要があるため，限られた応用範囲でのみ使えるアプローチといえる．これに対して，本研究では多様なカテゴリ型レコードデータに対して，より汎用的に利用可能な外れ値検出の提案を目指している．

文献 23) はカテゴリ型レコードデータに対する外れ値検出の研究であるが，ネットワークへの不正侵入検知を目的としており，アクセスログデータに特化したモデルを構築し，それに基づき外れ値を検出するアプローチとなっている．これもまた，本研究とは目的が異なる．

一方で，カテゴリ型レコードデータに対するより汎用的な外れ値検出手法も，いくつか研究されている[21),22)]．Chan らは本研究と同様，相関ルールを利用した外れ値検出手法を提案している[21)]．しかし，我々が提案する外れ値度が確率的な計算を必要としない一方で，彼らは確率論的な考えに基づき外れ値度を導出している[21)]．Das らもまた，確率的な計算に基づき例外となるレコードを検出する手法を提案している[22)]．本稿では Das らの確率的手法[22)] を，検出精度評価実験における比較対象として用いているので，後で彼らの手法を詳しく説明する．なお，Chan らの手法[21)] は Das らの論文[22)] で検出精度評価実験の比較対象に用いられ，検出能力において Das らの手法を下回る結果となっているため，本稿では比較対象としない．

He らは POS データに代表される**トランザクションデータ**に対して，外れ値となるトランザクションを検出する手法を提案している[5),24)]．彼らは，データ中に頻繁に出現するアイテム集合である**頻出アイテム集合**に着目し，頻出アイテム集合が数多く現れないトランザクションが外れ値であると考えている．トランザクションの正常度は，頻出アイテム集合のサポート値から計算される．本稿と異なり彼らの対象データはトランザクションデータだが，カテゴリ型レコードデータとトランザクションデータは理論上は相互に変換可能である．また，本研究と He らの共通点として，サポートの概念を取り入れている点がある．このことから本稿では，He らの手法[24)] も検出精度評価実験で比較対象に用いることとする．

以下，Das らの手法[22)] と，He らの手法[24)] について，それぞれ概説する．

Das らの確率的手法

Das らは，めったにない属性値集合の組合せを持つレコードが例外であるとして，このようなレコードの起こる確率を推定することで，例外レコードを特定しようとしている．彼らはまず，与えられた学習データの中から，あらゆる属性値の集合の出現数，すなわちその属性値集合を含むレコードの数を数え上げる．そして，この出現数を使って，テストレコード中のあらゆる属性値集合 A，B（$A \cap B = \emptyset$）のペアに対し，周辺確率の比を算出する．最も小さな値を示すペアの比を，そのレコード t の正常度（$r_value(t)$）としている．ここで，t に含まれる属性値集合 A，B（$A \cap B = \emptyset$）のあらゆるペアの集合を $pair(t)$ とすると，$r_value(t)$ は次式で表される．

$$r_value(t) = min_{(A,B)\in pair(t)} \frac{P(A,B)}{P(A)P(B)} \quad (1)$$

もし r_value が，ユーザが与えるしきい値 α 以下ならば，そのレコードは例外と判断される．しかし一般に，データ中に存在する属性値集合は大量なので，可能なすべての属性値集合のペアに対して周辺確率の比を計算していては，時間的コスト，空間的コストともに莫大となる．そこで，彼らはパラメータ $size$ を与え，$size$ を超える濃度の属性値集合に対しては，比の計算を行わないこととしている．

He らの頻出アイテム集合を用いた手法

He らは頻出アイテム集合が数多く現れないトランザクションが外れ値であると考え，トランザクションの正常度を頻出アイテム集合のサポート値を使って計算する．今，F を入力トランザクションから得られるすべての頻出アイテム集合の集合，トランザクション $trans$ をアイテムの集合であるとする．また，アイテム集合 X のサポートを $sup(X)$ と記述する．このとき，He らの導出するトランザクション $trans$ の正常度 $nd(trans)$ は次式で定義される．

$$nd(trans) = \frac{\sum_{X\in F \wedge X \subseteq trans} sup(X)}{|F|} \quad (2)$$

正常度の昇順に並んだトランザクションの中から，トップ l 個が外れ値と見なされる．

3. 準　　備

ここでは，本稿で用いる記述や定義について説明する．カテゴリ型レコードデータ DB はレコードの集合である．$|DB|$ は DB の集合濃度を表し，すべてのレコードの数である．n を，DB に存在する全属性の数とする．レコード $t \in DB$ はタプルである．$t[A]$ は，属性 A の，レコード t における値である．

今，A を属性，a を A の属性値とする．ペア (A, a) を 1 つの**アイテム**と見なすと，カテゴリ型レコードデータは，n 個のアイテムからなる**アイテム集合**の集合と見なすことができる．A_i を DB 上の i 番目の属性（$1 \le i \le n$）とすると，レコード t からマッピングされるアイテム集合 $map(t)$ は，$map(t) = \{(A_i, t[A_i])|1 \le i \le n\}$ で定義される．DB におけるあらゆるペアの数，すなわちアイテムのドメインの大きさを d とする．

© 2008 Information Processing Society of Japan

アイテム集合 X の**サポート** $sup(X)$ は次式で得られる．

$$sup(X) = \frac{|\{t|t \in DB \wedge X \subseteq map(t)\}|}{|DB|} \quad (3)$$

最小サポート $msup$ に対して，$sup(X) \geq msup$ であるならば，X を**頻出アイテム集合**であるとする．また，**極大頻出アイテム集合**とは，すべての頻出アイテム集合の集合の中で，超集合を持たない頻出アイテム集合のことである[4)]．

アイテム集合 X，Y（$X \cap Y = \emptyset$）に対して，記述 $X \rightarrow Y$ を**相関ルール**と呼ぶ．左辺のアイテム集合 X を**前提部**，右辺のアイテム集合 Y を**結論部**という．相関ルールの**確信度** $conf(X \rightarrow Y)$ は，次式で計算される．

$$conf(X \rightarrow Y) = \frac{sup(X \cup Y)}{sup(X)} \quad (4)$$

最小確信度 $mconf$ に対して $conf(X \rightarrow Y) \geq mconf$ を満足し，かつ $sup(X \rightarrow Y) = sup(X \cup Y) \geq msup$ である相関ルール $X \rightarrow Y$ を，本稿では**高確信度ルール**と呼ぶ．

上記のように定義される高確信度ルールは，明らかに，レコードデータ DB における属性値間の相関性に対応している．本稿ではこの高確信度ルールを外れ値度の計算のために用いる．

一般に，ある値 a ばかりが極端に頻出し，他の値がほとんど発生しないような非対称な属性 A が DB 上に存在するとき，真に意味のある相関ルールが，頻出する a の存在に隠されて発見できないことがある．本手法では，このような非対称性のあるデータからも有益な外れ値を検出できるように，レコードをアイテム集合にマッピングする際に次のようなオプションを用意する．

［**マッピングオプション**］ A を属性，a，b をそれぞれ A の異なる属性値とする．極端に頻出する a に関するアイテム (A, a) を無視し，アイテム (A, b) のみを考慮してレコードをアイテム集合にマッピングする．

4. 外れ値度

我々が想定する外れ値レコードは，そのレコードに現れている属性値の集合と，本来なら共起するはずの属性値が数多く観測されないようなレコードである．このようなレコードを発見するために，本稿ではユーザが非常に高い最小確信度を与えることを想定し，それによって得られる高確信度ルールを利用する．確信度 $conf(X \rightarrow Y)$ は X が起こったときに同時に Y が起こる確率に等しいので，明らかに，確信度値が大きいとき，属性値の集合 $\{a|(A, a) \in X\}$ に対して，属性値の集合 $\{b|(B, b) \in Y\}$ は共起する傾向にある．つまり，あるレコード t で，すべてのペア $(A, a) \in X$ に対して $t[A]$ が属性値 a と等しいならば，同時にすべての $(B, b) \in Y$ に対しても $t[B] = b$ となるべきはずであり，t で観測されない b が存在することは稀であると考えられる．このような属性間の相関性に反するような b が，多く存在するレコードが，外れ値であると考える．

定義 1　非充足相関ルール

R を高確信度ルールの集合，Z をアイテム集合とする．ある高確信度ルール $X \rightarrow Y \in R$ に対して，$X \subseteq Z$ であり，$Y \nsubseteq Z$ であるとき，このような高確信度ルール $X \rightarrow Y \in R$ を Z の**非充足相関ルール**と呼ぶ．

今，レコード t に対するアイテム集合 $map(t)$ の非充足相関ルールを，完全に被覆する次のようなアイテム集合を考える．

定義 2　相関性被覆

t をレコード，R を高確信度ルール集合とする．${C_t}^0 = map(t)$ としたとき，t の**相関性被覆**は次のような処理で得られるアイテム集合 ${C_t}^+$ である．

$${C_t}^{i+1} = {C_t}^i \bigcup \left(\bigcup_{X \rightarrow Y \in R, X \subseteq {C_t}^i} Y \right)$$
$${C_t}^+ = {C_t}^\infty$$

定義 2 より，レコード t の相関性被覆は，非充足相関ルールが数多く存在するほど大きくなり，元となった $map(t)$ とかけ離れていく．本研究が想定する高確信度ルールは確信度の値が非常に大きいので，このような高確信度ルールの集合を使って得られる t の相関性被覆が元と比べて大きくなるほど，t は逸脱していると考えられる．

定義 3　外れ値度

レコード t の外れ値度は次式で定義される．

$$od(t) = \frac{|{C_t}^+ - map(t)|}{|{C_t}^+|} \quad (5)$$

定義 4　外れ値レコード

しきい値 mod に対して，$od(t) \geq mod$ であるとき，レコード t を外れ値と見なす．また，このような mod を**最小外れ値度**と呼ぶ．

5. 外れ値検出アルゴリズム

最初に，我々が想定する外れ値レコードを検出する手法として，最もナイーブなアルゴリ

© 2008 Information Processing Society of Japan

表 2　ブルートフォースアルゴリズム
Table 2　Brute Force Algorithm.

入力：DB, $msup$, $mconf$, mod
出力：a complete set of outliers O
1.　頻出アイテム集合の集合 F をマイニング;
2.　F から高確信度ルール集合 R を生成;
3.　O = getOutliers(DB, R);

ズムを**表 2** に示す．

このアルゴリズムでは，最初に頻出アイテム集合をマイニングし（1 行目），頻出アイテム集合の集合 F を得る．次に，F からすべての高確信度ルール $X \to Y$ を生成する（2 行目）．F，R はそれぞれ既存手法を用いて十分生成が可能である．最後に，得られた高確信度ルール集合 R を引数として関数 getOutliers を呼び出し，レコードの外れ値度を計算する．

関数 getOutliers のアルゴリズムは**表 3** に別途リストアップする．各 $t \in DB$ に対して，定義 2 で定めた処理に基づきすべての高確信度ルールをチェックしながら，相関性被覆を作成する．

提案手法では，最小サポート $msup$，最小確信度 $mconf$，最小外れ値度 mod の 3 つのパラメータを必要とする．予備実験から，提案外れ値度は $msup$ や $mconf$ の値に依存してかなり変動することが分かっており，ある $msup$ と $mconf$ の値に対して不適切な mod を与えれば，外れ値レコードがまったく検出されない結果に終わることもありうる．時間とリソースを無駄にしないためにも，そのようなケースをいち早く発見し，処理を中止することは重要である．そこで本稿では，DB 内に外れ値レコードがまったく存在しない場合に，処理を途中で終了するファストパスを提供する．

関数 getOutliers の時間計算量は，与えられた 2 つの引数 DB と R に対して $O(|DB| \times |R|)$ である．一般に $|DB|$ と $|R|$ は巨大であることから，処理にはかなりの時間がかかると考えられる．そこで処理の高速化を目的として，外れ値度の計算が真に必要とされるレコードの絞り込みと，計算に必要でない冗長な相関ルールを削除する工夫を取り入れる．

以下では，まず最初にファストパスについて説明し，次に処理を高速化するための 2 つの工夫について言及する．

5.1　不適切なパラメータに対するファストパス

以下に示す外れ値度の性質により，我々の手法では頻出アイテム集合をマイニングした時点で，入力データ中に与えられたパラメータに対する外れ値が存在しないケースを検出することが可能である．

表 3　関数 getOutliers
Table 3　Function getOutliers.

Function getOutliers（レコード集合 DB，相関ルール集合 R）
1.　$O = \emptyset$;
2.　foreach($t \in DB$) {
3.　　${C_t}^0 = map(t)$;
4.　　$i = 0$, $R' = R$;
5.　　until（${C_t}^i$ が成長をやめる）{
6.　　　$TMP = \emptyset$;
7.　　　foreach($r = X \to Y \in R'$) {
8.　　　　if($X \subseteq {C_t}^i$)
9.　　　　　Y を TMP に加え，r を R' から削除;
10.　　　}
11.　　　${C_t}^{i+1} = {C_t}^i \cup TMP$;
12.　　　$i++$;
13.　　}
14.　　式 5 に基づき外れ値度 $od(t)$ を計算;
15.　　if($od(t) \geq mod$)
16.　　　t を O に加える;
17.　}
18.　return O

性質 1　レコード t に対し，外れ値度 $od(t)$ は $|{C_t}^+|$ が大きくなるにつれて単調に増加する．

証明 1　ある t に対して，式 (5) で与えられる外れ値度 $od(t)$ は，x を変数，c を定数としたときの関数 $(x-c)/x$ に等しくなる．この関数は単調増加関数であるため，性質 1 は正しい．　□

ファストパスのアプローチはマッピングオプションを使わない場合と，使う場合とでいささか異なる．ここではまず，マッピングオプションを使わないケースを考える．このとき，すべてのレコード t に対して $|map(t)| = n$ であることを特記しておく．

定義 5　最大増分
F をすべての頻出アイテム集合の集合とする．このとき，アイテム集合 $I_{\geq 2} = \{i | X \in F \wedge |X| \geq 2 \wedge i \in X\}$ を**最大増分**と呼ぶ．

性質 2　あらゆるレコード t に対して，${C_t}^+ - map(t) \subseteq I_{\geq 2}$ が必ず成り立つ．

証明 2　F から生成される高確信度ルール集合を R とする．任意の要素アイテム $x \in$

© 2008 Information Processing Society of Japan

${C_t}^+ - map(t)$ に対して，$x \in W$ を満たす $Z \to W \in R$ が必ず存在する．高確信度ルールの定義により，$Z \cup W$ は頻出アイテム集合であり，$|Z \cup W| \geq 2$ である．よって，上記性質が成り立つ．□

性質 3　頻出アイテム集合の集合 F に対して，次の条件を満足するならば，あらゆるレコード t の外れ値度 $od(t)$ はつねに $od(t) < mod$ である．

$$|I_{\geq 2}| < \frac{mod}{1 - mod} n \tag{6}$$

証明 3　性質 2 から，$|{C_t}^+| - n \leq |I_{\geq 2}|$ であるので，$|I_{\geq 2}| + n$ が $|{C_t}^+|$ の上限である．また外れ値度 $od(t)$ は $(|{C_t}^+| - n)/|{C_t}^+|$ と計算結果が等価である．この式に $|I_{\geq 2}| + n$ を代入すれば，性質 1 から，任意の t において $od(t) \leq |I_{\geq 2}|/(|I_{\geq 2}| + n)$ である．このとき $|I_{\geq 2}|/(|I_{\geq 2}| + n) < mod$ ならばつねに $od(t) < mod$ となり，不等式を整理すれば $|I_{\geq 2}| < \{mod/(1 - mod)\}n$ である．□

以上の性質から，ファストパスが適応できる．まず頻出アイテム集合をマイニングする段階で，同時に $|I_{\geq 2}|$ を得る．もし条件式 (6) を満足するならば，データ中には外れ値レコードが存在しないものとして，処理を途中で終了する．

マッピングオプションを用いる場合は，n の代わりに最大濃度 $n_{max} = max_{t \in DB}|map(t)|$ を用いる．このときも性質 3 が成り立つことは自明である．

5.2　候補外れ値レコードの絞り込み

外れ値になる見込みのないレコードを外れ値度を計算する前に刈り取ることで，処理速度の向上が期待できる．このような候補外れ値レコードの絞り込みは，外れ値度の上限を求めることで可能となる．

性質 4　ある $msup$ において，$mconf = 0$ であるときの $t \in DB$ の相関性被覆を ${C0_t}^+$，$mconf > 0$ であるときの t の相関性被覆を ${C_t}^+$ とすると，必ず ${C0_t}^+ \supseteq {C_t}^+$ である．

証明 4　ある $msup$ において $mconf_1 > mconf_2$ である 2 つの最小確信度を考える．$mconf_1$ で得られる高確信度ルール集合を R_1，$mconf_2$ で得られる高確信度ルール集合を R_2 とすれば，$R_1 \subseteq R_2$ である．よって，R_1 で作られる相関性被覆 ${C1_t}^+$ と R_2 で作られる相関性被覆 ${C2_t}^+$ は，${C1_t}^+ \subseteq {C2_t}^+$ であり，上記性質が成り立つことは自明である．□

定義 6　極大相関性被覆

M を最小サポート値 $msup$ に対するすべての極大頻出アイテム集合の集合，t をレコード，${MC_t}^0 = map(t)$ とする．t に対して次の処理により得られるアイテム集合 ${MC_t}^+$ を，t の**極大相関性被覆**と呼ぶ．

$${MC_t}^{i+1} = {MC_t}^i \bigcup \left(\bigcup_{m \in M, m \cap {MC_t}^i \neq \emptyset} m \right)$$
$${MC_t}^+ = {MC_t}^\infty$$

性質 5　M を最小サポート $msup$ に対するすべての極大頻出アイテム集合の集合，${MC_t}^+$ を M におけるレコード t の極大相関性被覆とする．このとき最小確信度が $mconf = 0$ における t の相関性被覆を ${C0_t}^+$ とすると，${MC_t}^+ = {C0_t}^+$ である．

証明 5　最小サポート $msup$，最小確信度 $mconf = 0$ のときの高確信度ルール集合を R_0 とする．${MC_t}^0$ と ${C0_t}^0$ はともに $map(t)$ である．今，$i = k$（$k > 0$）において ${MC_t}^k = {C0_t}^k$ を仮定する．定義 6 より，任意の $y_1 \in {MC_t}^{k+1} - {MC_t}^k$ に対して，$y_1 \in m \wedge m \cap {MC_t}^k \neq \emptyset$ となる $m \in M$ が必ず存在する．このとき，$X = m \cap {MC_t}^k$，$Y = m - X$ とすると，$y_1 \in {MC_t}^{k+1} - {MC_t}^k$ から $y_1 \notin {MC_t}^k$ であり，必ず $y_1 \in Y$ である．また $X \subseteq {MC_t}^k$ であるため仮定より $X \subseteq {C0_t}^k$ である．さらに，高確信度ルールの定義より必ず $X \to Y \in R_0$ が存在するので，定義 2 より $Y \subseteq {C0_t}^{k+1}$ であり，$y_1 \in {C0_t}^{k+1}$ がいえる．また，$y_2 \in {MC_t}^k \wedge y_2 \in {MC_t}^{k+1}$ である任意の y_2 に対して，仮定より $y_2 \in {C0_t}^k$ であり，定義 2 より必ず $y_2 \in {C0_t}^{k+1}$ である．以上より ${MC_t}^{k+1} \subseteq {C0_t}^{k+1}$ が成り立つ．また，定義 2 より，任意の $w_1 \in {C0_t}^{k+1} - {C0_t}^k$ に対して $w_1 \in W \wedge Z \subseteq {C0_t}^k$ となる $Z \to W \in R_0$ が存在する．このとき，高確信度ルールの定義から $Z \cup W$ は頻出アイテム集合であり，$Z \cup W \subseteq m'$ である $m' \in M$ が必ず存在する．明らかに $m' \cap {C0_t}^k \neq \emptyset$ であり，仮定より $m' \cap {MC_t}^k \neq \emptyset$ である．よって，定義 6 より $m' \subseteq {MC_t}^{k+1}$ であり，$w_1 \in {MC_t}^{k+1}$ がいえる．また，$w_2 \in {C0_t}^k \wedge w_2 \in {C0_t}^{k+1}$ である任意の w_2 に対して，仮定より $w_2 \in {MC_t}^k$ であり，定義 6 より必ず $w_2 \in {MC_t}^{k+1}$ である．よって，${MC_t}^{k+1} \supseteq {C0_t}^{k+1}$．したがって，${MC_t}^{k+1} = {C0_t}^{k+1}$ が成り立つ．以上から ${MC_t}^+ = {C0_t}^+$ が成り立つ．□

定義 7　外れ値度の上限

R を高確信度ルール集合，M を極大頻出アイテム集合の集合であるとする．このとき次の式で求められる $od_{max}(t)$ を，レコード t の外れ値度の**上限**と呼ぶ．

$$od_{max}(t) = \frac{|{MC_t}^+ - map(t)|}{|{MC_t}^+|} \tag{7}$$

性質 6　最小サポート $msup$ において，レコード t の外れ値度 $od(t)$ と上限 $od_{max}(t)$ の

© 2008 Information Processing Society of Japan

表 4　動物の生態データ
Table 4　Data about individual animals.

ID	タプル					
t_1	false	2	false	true	true	false
t_2	false	6	false	true	true	false
t_3	false	4	true	true	false	true
t_4	false	0	true	true	false	false
t_5	true	4	true	false	false	true
t_6	false	6	false	true	true	false
t_7	true	4	true	false	false	false
t_8	false	2	false	true	true	false
t_9	true	4	true	false	false	false
t_{10}	false	6	false	true	true	false

表 5　属性情報
Table 5　Attributes information.

ID	属性	属性値
A_0	Milk	ブール値
A_1	# of Legs	非バイナリ値
A_2	Toothed	ブール値
A_3	Eggs	ブール値
A_4	Airborne	ブール値
A_5	Aquatic	ブール値

間にはつねに $od(t) \leq od_{max}(t)$ が成り立つ．

証明 6　性質 4，5 から $|{C_t}^+| \leq |{MC_t}^+|$ であるため，性質 1 より自明である．　□

極大頻出アイテム集合の集合 M は，頻出アイテム集合をマイニングする際に得ることができる．M を用いて，外れ値度の計算を行う前にすべてのレコードで外れ値度の上限 $od_{max}(t)$ を計算する．$od_{max}(t) < mod$ であるレコード t は，外れ値レコードになりえないため事前に刈り取る．一般に，$|M|$ は $|R|$ に比べて非常に小さい．

ことから，M を用いたレコードの絞り込みを行うことで処理速度の向上が期待できる．

すべてのレコードで極大相関性被覆を計算する時間計算量は，M に対して $O(n \log n \times |M| \times |DB|)$ である．$|M|$ は最悪でも $|F|$ に収まる．

ここで，例を用いて実際に外れ値度とその上限を計算する様子を示す．**表 4** は，各レコードが 1 匹の動物の生態を表すレコードデータである．このデータを構成する属性の情報が，

表 6　極大頻出アイテム集合
Table 6　Maximal frequent itemsets.

極大頻出アイテム集合
{(Milk,true), (# of Legs, 4), (Toothed, true)}
{(# of Legs, 6), (Eggs, true), (Airborne, true)}

表 5 に示されている．さらに**表 6** には，マッピングオプションを用いてすべてのブール値属性の値 $false$ を無視した場合に $msup = 30\%$に対して表 4 から得られるすべての極大頻出アイテム集合を示す．このときレコード t_3 に対応するアイテム集合は $map(t_3) = \{$(# of Legs, 4), (Toothed, true), (Eggs, true), (Aquatic, true)$\}$ である．ここから，$MC^+_{t_3}$ と $C^+_{t_3}$ は次のようになる．

$$\begin{aligned} MC^+_{t_3} = \{&(\#ofLegs, 4), (Toothed, true), (Eggs, true), \\ &(Aquatic, true), (Milk, true), (Airborne, true), \\ &(\#ofLegs, 6)\}. \\ C^+_{t_3} = \{&(\#ofLegs, 4), (Toothed, true), (Eggs, true), \\ &(Aquatic, true), (Milk, true), (Airborne, true)\}. \end{aligned}$$

よって，$od_{max}(t_3) = 0.43$ であり，$od(t_3) = 0.33$ である．

5.3　冗長ルールの除去

高確信度ルール集合 R の中には，相関性被覆の作成に対して冗長なルールが存在する．このようなルールは外れ値度の計算に必要ないので，あらかじめ除去する．

再び表 4 の例を用いて説明する．$msup = 30\%$と $mconf = 70\%$が与えられたとき，**表 7** に示されるような高確信度ルールの集合が得られる．ただし，確信度が 100%である高確信度ルールは非充足相関ルールになりえず，相関性被覆の成長に貢献する見込みがないため，表 7 には示していない．このとき，$rule_3$ があるため，$rule_1$ は相関性被覆の成長になんら貢献することがない冗長なルールである．同様に，$rule_2$ も $rule_3$ に対して冗長である．

定義 8　冗長ルール

R を高確信度ルールの集合とする．ルール $X \to Y \in R$ は，次の 2 つの条件のうち少なくともいずれか 1 つを満たすような他の異なるルール $Z \to W \in R$ が存在するとき，R において冗長ルールであるといえる．

(i) $Z \subset X \wedge Z \cup W = X \cup Y$.

(ii) $Z = X \wedge W \supset Y$.

© 2008 Information Processing Society of Japan

表 7　生成されるルール
Table 7　Generated rules.

ID	高確信度ルール
$rule_0$	{(Toothed,true)}→{(# of Legs,4)}
$rule_1$	{(# of Legs,4)}→{(Milk,true)}
$rule_2$	{(# of Legs,4),(Toothed,true)}→{(Milk,true)}
$rule_3$	{(# of Legs,4)}→{(Milk,true),(Toothed,true)}
$rule_4$	{(Eggs,true)}→{(Airborne,true)}

高確信度ルールを生成する際，冗長ルールをすべて除去し，残りの高確信度ルールだけを相関性被覆の作成に利用する．このような残りの高確信度ルールの集合を，ここでは**最小ルール集合**と呼ぶ．表 7 の場合，最小ルール集合は $\{rule_0, rule_3, rule_4\}$ である．

レコード t に対し，${C_t}^+$ を高確信度ルール集合 R で作られた相関性被覆，${C_{t,min}}^+$ を R に対する最小ルール集合 R_{min} から作られた相関性被覆とする．このとき必ず ${C_{t,min}}^+ = {C_t}^+$ である．

ある $Z \rightarrow W \in R_{min}$ に対して，最低でも $2^{|W|} - 2$ 個の高確信度ルール $X \rightarrow Y$ が冗長ルールとして R 内に存在する．R_{min} は R に比べてコンパクトになり，外れ値度計算の処理速度はかなり向上すると期待できる．

5.4　外れ値レコード検出アルゴリズム

これまでで説明したファストパスと 2 つの処理速度向上のための工夫を取り入れた提案アルゴリズムを**表 8** に示す．大きく次の 4 つのフェーズに分かれている．(i) 頻出アイテム集合マイニングフェーズ（1 行目），(ii) 候補外れ値レコード絞り込みフェーズ（6–10 行目），(iii) 最小ルール集合 R_{min} 生成フェーズ（14–22 行目），(iv) 外れ値検出フェーズ（23 行目）である．

フェーズ (i) では頻出アイテム集合をマイニングするとともに，すべての極大頻出アイテム集合と最大増分の集合濃度 $|I_{\geq 2}|$ を得る．頻出アイテム集合をマイニングしながら極大頻出アイテム集合を得るために，FP-growth アルゴリズム[25] を改良したものを実装した．これは，文献 26) で提案されている MFI-tree と subset-checking メソッドを利用したものである．まず，最初に DB を読み込み FP-tree[25] を構築する．この時間計算量は $O(n \log n \times |DB|)$ である[25]．最初の木の構築を終えると，再帰的に頻出アイテム集合をマイニングしながら，極大か否かチェックを行う．このとき，1 つの $X \in F$ に対して，1 つの局所的な FP-tree を構築する．この木は，最悪，$|DB| \times sup(X)$ 個のトランザクション（大きさ $n - |X| + 1$）のデータから木を構築するときと同じコストがかかる．よって，FP-growth アルゴリズムの計算コストは $O(\sum_{X \in F}((n - |X|) \log (n - |X|) \times |DB| sup(X)))$ である．提案手法では 1 つの $X \in F$ を発見すると極大か否か判定するので，$O(\sum_{X \in F}((n - |X|) \log (n - |X|) \times |DB| sup(X) + |X|))$ となる．最悪の場合でも，$O(n \log n \times |F| \times |DB|)$ である．

表 8　提案アルゴリズム
Table 8　Proposed algorithm.

入力：DB, $msup$, $mconf$, mod and n (or n_{max})
出力：外れ値レコードの完全集合 O

```
1.  頻出アイテム集合の集合 F，極大頻出アイテム集合の集
    合 M，および |I≥2| を得る;
2.  /* DB に外れ値が存在しない場合のファストパス */
3.  if(条件式 6 が真)
4.     exit;
5.  /* 外れ値の候補レコードの集合 C を得る */
6.  foreach(t ∈ DB) {
7.     t の極大相関性被覆 MC_t^+ を作成;
8.     if(od_max(t) ≥ mod)
9.        t を C に加える;
10. }
11. if(C = ∅)
12.    exit;
13. /* 最小ルール集合 R_min を得る */
14. foreach(l_k ∈ F (k ≥ 2)) {
    // l_k は濃度 k の頻出アイテム集合
15.    H_1 = {{h∈l_k}|conf((l_k−{h})→{h})≥mconf};
16.    R_min = R_min ∪ genMinRule(l_k, H_1);
17. }
18. foreach(マークのない r_1∈R_min と，異なる r_2∈R_min)
    {
19.    if(r_2 が定義 8 の条件 (ii) を満たす)
20.       r_1 にマークをつける;
21. }
22. マークのついているルールを R_min から除去;
23. O = getOutliers(C, R_min);
```

3，4 行目でファストパスの判定を行い，ファストパスを適用しない場合はフェーズ (ii) に移る．ここではすべてのレコードに対して外れ値度の上限を計算し，候補外れ値レコードの集合 C を獲得する．外れ値度の上限を計算する手順は，外れ値度を計算する手順と非常によく似ているため，ここでは説明を省く．先述のとおり，時間計算量は $O(n \log n \times |M| \times |DB|)$ である．

© 2008 Information Processing Society of Japan

表 9　関数 genMinRule
Table 9　Function genMinRule.

```
Function genMinrule（頻出アイテム集合 l_k,
                     集合濃度 m のアイテム集合の集合 H_m）
1.  if(k > m + 1) {
2.    TMP_1 = ∅;
3.    H_{m+1} = apriori-gen(H_m);
4.    foreach(h_{m+1} ∈ H_{m+1}) {
5.      r = h_{m+1} → (l_k − h_{m+1});
6.      if(conf(r) ≥ mconf)
7.        r を TMP_1 に加える;
8.      else H_{m+1} = H_m − h_{m+1};
9.    }
10.   TMP_2 = genMinRule(l_k, H_{m+1});
11.   foreach(マークのない r_1∈TMP_1 と，r_2∉TMP_2) {
12.     if(r_2 が定義 8 の条件 (i) を満たす)
13.       r_1 にマークをつける;
14.   }
15. }
16. return TMP_1 ∪ TMP_2;
```

フェーズ (iii) は，文献 3) の手法を基盤としたアルゴリズムとなっている．集合濃度 k の頻出アイテム集合 l_k に対して，関数 genMinRule（**表 9**）を再帰的に呼び出し，前提部と結論部の和が l_k である高確信度ルールを作る．この際，定義 8 の条件 (i) にあてはまる高確信度ルールを探し，冗長ルールであるとしてマークする．表 8 の 18 行目以降，再帰的処理を抜けると，今度は定義 8 の条件 (ii) にあてはまる高確信度ルールがないか調べ，もし存在していたら，それは冗長ルールであるとしてマークする．最後に，これまでの処理でマークされたルールを，冗長ルールであるとしてすべて削除し，残りの高確信度ルールの集合を最小ルール集合 R_{min} として得る．関数 genMinRule は l_k に対して $k-1$ 回呼び出される．1 度の呼び出しで，l_k の部分集合の数 $2^{k-1}-1$ に依存する時間コストがかかる．このコストは文献 3) のアルゴリズムのコストに準じている．また，冗長ルールの除去部分で，最悪 $|R|^2$ に依存するオーダがかかる．よって，全部で $O(n|F|+|R|^2)$ である．

フェーズ (iv) で，提案アルゴリズムは関数 getOutliers を呼び，候補として残ったレコードに対してのみ外れ値度をきちんと計算し，外れ値レコードの集合を得る．時間計算量は，getOutliers が受け取っている引数に基づき，$O(n \log n \times |R_{min}| \times |C|)$ である．

6. 実　　験

本稿では，提案手法が実際に外れ値として説得力のあるレコードを検出しうるか，関連手法と比べてどのくらいの検出精度を得られるか，ブルートフォースアルゴリズムと比べてどの程度処理速度の向上が認められるかを確認するための実験を行う．はじめに，実験に使用する実データについて説明し，それから外れ値レコードの検証，2 つの関連手法[22),24)] を比較対象とした検出精度の評価，ブルートフォースアルゴリズムを比較対象とした処理速度の高速性の評価の順で，実験結果を示す．

6.1　データセット

使用したのはすべて UCI Machine Learning Repository [27)] で配布されているデータで，ここではそれぞれ Zoo，Mushroom，KDD Cup 99 と呼ぶ．

Zoo データは動物の生態を表したデータベースである．元々は 15 個のブール値属性と 2 個の非バイナリ属性を持つ非対称性のあるデータである．各レコードが 1 匹の動物に対応しており，哺乳類，鳥類，爬虫類，両生類，魚類，昆虫，昆虫以外の節足動物の 7 クラスのうちの 1 つに分類される．この元データから，動物名属性とクラス属性を削除した 16 個の属性からなるデータを Zoo と名づけ，外れ値レコードの検証実験に使用する．さらに，処理速度の評価実験で処理時間を比較しやすくするため，Zoo のレコードを各 10,000 個ずつ持つデータセット Zoo10000 を用意する．ここで，Zoo10000 における各アイテム集合のサポート値は，Zoo におけるそれと等値である．Zoo データのレコード数は全部で 101 である．また，クラスの種類が多いため，後述する精度評価実験で十分信頼できる精度を測れないと判断した．よって，精度評価実験の対象には加えていない．

Mushroom はキノコの生態を表すレコードデータである．すべての属性が非数値で，各レコードは無毒クラスか有毒クラスのいずれかに分類されている．元データにおいては，無毒クラスに属するレコードは 4208 個，有毒クラスに属するレコードは 3916 個存在し，2 つのクラスの大きさはほとんど変わらない．この元データを Original Mushroom と呼び，外れ値レコードの検証実験に使用する．精度評価実験では，有毒クラスに属するレコード数が全レコード数の約 1%となるよう有毒クラスのレコードを減らしたデータを用意し，有毒クラスのレコードを正解外れ値と見なしている．この調整されたデータを Mushroom と呼び，検出精度評価実験に使用する．さらに，処理速度のの評価実験で処理時間を比較しやすくするため，Mushroom データのレコードを各 100 個ずつ持つデータセット Mushroom100 を用意する．

© 2008 Information Processing Society of Japan

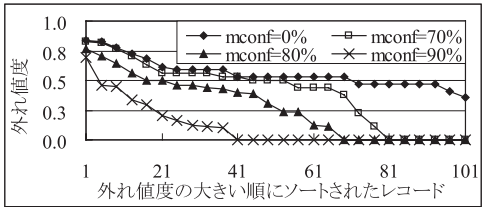
KDD Cup 99 は，ネットワークへのアクセスログデータである．元データにおいて，各レコードは正常クラスと，guess password クラスや warezmaster クラス等の不正侵入を表すクラスのいずれか 1 つに属している．ほとんどの属性が連続値であるため，それらの属性値は 5 つのレベルに区分しカテゴリ化した．不正侵入を表すクラスの 1 つである guess password に属するレコードを正解外れ値と見なし，97%の正常クラスに属するレコードと3%の外れ値レコードからなるデータセットを作成し，検出精度評価実験および処理速度比較実験で使用する．このデータを KDD Cup 99 とする．同様の手順で作られたデータが，精度比較の対象である Das らの手法[22] でも用いられている．

6.2　外れ値度の分布

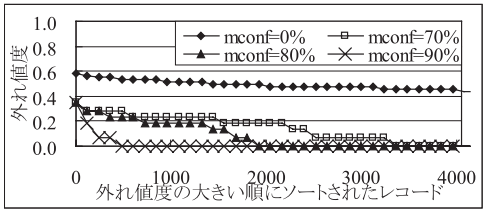
まず，提案外れ値度が各データでどのように分布するかを観察するため，各データのすべてのレコードの外れ値度を計算した．Zoo データに対しては $msup$ を 10%に固定し，ブール値属性の値 false に関するアイテムを無視するマッピングオプションを用いた．また Mushroom データと KDD Cup 99 データに対しては $msup$ を各々20%，95%と固定した．このとき，いくつかの $mconf$ 値を与えて，各レコードの外れ値度がどのように分布するかを観察した．**図 2** (a)–(c) の横軸は外れ値度の大きい順に左からソートされたレコードに対応している．縦軸が外れ値度を表している．3 つの図から，$mconf$ 値が大きくなるほど，より少数のレコードが比較的大きな外れ値度を持つようになっている様子が分かる．また，図 2 (b) では，$mconf = 0$%に対する外れ値度，すなわち外れ値度の上限が非常に高い値を示し，各レコード間でほとんど差がないことが分かる．Mushroom では各レコードの極大相関性被覆の濃度が，あらゆるアイテム（属性と属性値のペア）の数にほとんど等しくなっているためと考えられる．このことから，Mushroom データは得られる頻出アイテム集合の平均濃度が比較的大きい，密なデータであるといえる．また，図 2 (c) では，$mconf = 95$%に対する折れ線が，完全に $mconf = 0$%に対する折れ線と一致している．これは，KDD Cup 99 データにおいて，生成される相関ルールの確信度がどれも非常に大きいことを表している．このことから，KDD Cup 99 データは属性値間の相関性が非常に強いデータであることが分かる．

パラメータセンシティビティ

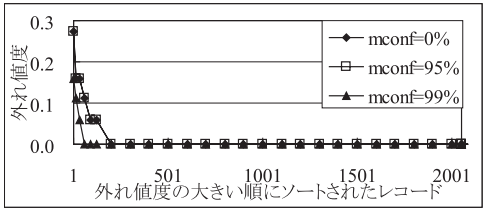
上述したデータセットの性質以外にも，図 2 からは，最小確信度を変えただけで外れ値度の分布が大きく変化することが分かる．このことから，与えられるパラメータセットによって提案手法の検出結果がセンシティブに変わることは容易に想像できる．このようなパラメータセンシティビティへの対策の 1 つとして，本稿ではファストパスを提案した．しかし，根本的な解決には最適なパラメータセットの推定方法を考えることが必要不可欠である．本稿ではこの問題を今後の研究課題と位置づけている．



(a) Zoo



(b) Mushroom



(c) KDD Cup 99

図 2　外れ値度の分布
Fig. 2　Outlier Degrees Destribution.

6.3　外れ値レコードの検証

ここでは提案手法を Zoo と Original Mushroom に適用し，外れ値として得られたレコードが真に外れ値として説得力のあるものであるかどうかを検証する．

表 10 は 6.2 節でも用いたマッピングオプションを利用し，Zoo から ($msup$, $mconf$) = (20%, 90%) で外れ値検出を行ったときの，外れ値度の大きい上位 3 件のレコードの内容を表している．1 列目にはレコードが表している動物の名前が，2 列目にはそのレコードの対

© 2008 Information Processing Society of Japan

表 10　Zoo における上位 3 件の外れ値レコード
Table 10　Top 3 outliers on Zoo data.

動物名	$map(t)$	典型的な非充足相関ルール
Crab	{Eggs, Aquatic, Predator, (# of Legs, 4)}	{(# of Legs, 4)}→{Toothed, Backbone, Breathes}
Housefly/Moth	{Hair, Eggs, Airborne, Breathes, (# of Legs, 6)}	{Hair}→{Milk, Backbone, Breathes}

応するアイテム集合 $map(t)$ が，3 列目には $map(t)$ の典型的な非充足相関ルールがリストされている．ブール値属性の値 false に関するアイテムはここでは無視されているので，スペースの都合上，ブール値属性の値 true に関するアイテムは，属性と属性値のペアという記述をせず，単に属性だけを記述する．最も外れ値度の大きかったのはカニ（Clab）を表すレコードである．カニは，本来 8 本足のはずだが，対応する $map(t)$ を見るとアイテム (# of Legs, 4) が出現している．ここでは,「4 対の足」という意味で (# of Legs, 4) が用いられていると考えられる．しかしながら，(# of Legs, 4) は通常哺乳類や爬虫類のような「4 本足」を意味するアイテムであり,「4 対の足」を意味する使い方は明らかに例外であるといえる．イエバエ（Housefly）とガ（Moth）は，それぞれの対応するアイテム集合が同一であるため，外れ値度が等しくなった．元々，イエバエやガのように昆虫クラスに属するレコードは，全体でも 8 種類しか存在しない．その中でも，アイテム Hair を持つ昆虫はイエバエとガ，そして 4 番目に外れ値度の大きいカリバチ（Wasp）のみである．Hair は，比較的大きなクラスである哺乳類や鳥類に属するレコードで頻出するアイテムである．哺乳類や鳥類の持つ「毛」と昆虫の持つ「毛」は本来異なるものであるのに，Zoo データでは同じ Hair で表されているため，少数派のクラスで Hair を持つレコードの逸脱性が大きくなった．

提案手法で検出されたこれらの外れ値レコードは，例外的な属性値の組合せを持つ珍しいレコードであるので，マッピングオプションを用いて Zoo データから検出されたレコードは，外れ値として説得力のあるものだといえる．

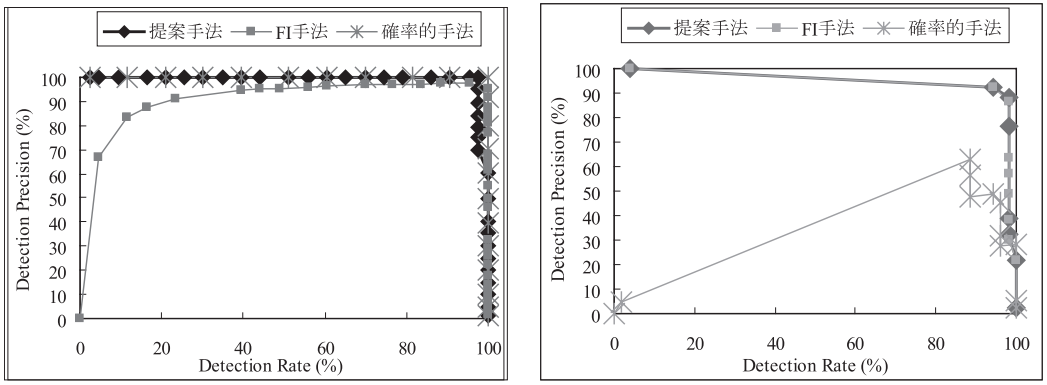
また，マッピングオプションを使わない場合は，最も高い外れ値度を示すレコードとしてタコ（Octopus）が検出された．典型的な非充足相関ルールに，{(Feathers, false), (Fins, false), (Catsize, true)} → {(Hair, true), (Eggs, false), (Milk, true), (Airborne, false), (Toothed, true), (Backbone, true), (Breathes, true), (Venomous, false)} があげられる．このルールの結論部は Zoo データの最大クラスである哺乳類の特徴が典型的に現れているものの，前提部は鳥類と魚類以外の 5 つのクラスに属するレコードで観測される情報であり，この非充足相関ルールは高い確信度を持つものの規則性としてあまり価値があるとはいえない．そのため，なぜタコが最も大きな外れ値度を示したのかが不明確となった．このことは，非対称性のあるデータの場合，マッピングオプションが外れ値の検出に有益に働いていることを示している．

さらに，提案手法を Original Mushroom データに $(msup, mconf) = (6\%, 90\%)$ で適用させた．外れ値度の大きい上位 112 件のレコード（全体の 1.4%）が，等値の外れ値度を示した．これらの外れ値のうち，93 個は無毒クラスに属しており，アイテム（Odor, none）を持たないレコードとなった．すなわち，何らかの香り（odor）を発する無毒なキノコである．このような無毒キノコは，データ全体でわずか 10%しか存在しない少数派である．さらに，生成された高確信度ルールの約半数は，アイテム（Odor, none）を前提部か結論部に持っている．実際，検出された外れ値レコードの相関性被覆は，まず {(Class, edible),...}→{(Odor, none),...} という形をしている非充足相関ルールによって成長し，次にアイテム（Odor, none）を前提部に持つ非充足相関ルールが成長に使われ大きく成長していった．同列 1 位である残りの 19 件の外れ値は，有毒クラスに属するレコードである．これらは皆，全体の約半数の高確信度ルールに含まれている (Stalk-shape, tapering) や (Stalk-root, bulbous) といったアイテムを持たなかったために，相関性被覆が大きくなっていった．Original Mushroom からも，より少数派の属性値の組合せを持つレコードが外れ値として検出されていることが分かった．

6.4 精度比較

ここでは提案手法と関連手法[22),24)] の検出精度を測定し，比較する．文献 24) はトランザクションデータを対象とした手法であるので，文献 24) に対しては，各データセットを属性と属性値のペアを 1 個のアイテムと見なしトランザクションデータに変換したデータを使用した．文献 22) の精度評価実験で用いられていたのと同様に，Detection Rate と Detection Precision を精度の尺度として利用する．Detection Rate は正解外れ値の数に対する，実際に検出された正解外れ値の数の比であり，Detection Precision は実際に外れ値として検出されたレコードの数に対する，実際に検出された正解外れ値の数の比である．各手法で用いられるパラメータセットが異なることから，公平を期するため，ここでは各手法に最適パラメータを与えて検出を行い，外れ値度（または正常度）の大きさの順にソートされたレ

© 2008 Information Processing Society of Japan



(a) Mushroom　　(b) KDD Cup 99

図 3　検出精度
Fig. 3　Accuracy comparison.

コード上位 l 件を外れ値と見なし，l に対する Detection Rate（dr_l）と Detection Precision（dp_l）を計算した．最適パラメータセットの決め方は次のとおりである．各手法でパラメータセットを変えながら，外れ値検出を行い，次式で得られる最大 F-measure F_{max} が最も大きくなるパラメータセットを最適と見なした．

$$F_{max} = max_l \frac{2dr_l dp_l}{dr_l + dp_l}$$

図 3 (a) は Mushroom データに対するグラフである．図中，FI 手法は He らの頻出アイテム集合を用いた手法[24] で，確率的手法は Das らの手法[22] である．提案手法にはパラメータ $msup = 20\%$，$mconf = 95\%$，$mod = 0.25$，FI 手法には $msup = 20\%$，確率的手法には $size = 2$ と $\alpha = 0.003$ をそれぞれ与えた．どの手法でも，非常に良い精度が得られている．FI 手法では，正解外れ値ではないレコードに最も小さな正常度が割り当てられたため，右肩上がりのグラフとなり，Detection Precision を見た場合，提案手法が FI 手法を上回った．しかし，FI 手法ではその後正解外れ値に小さな正常度が順当に割り当てられ，Detection Rate を見た場合，最終的には提案手法を若干上回る結果となった．確率的手法では，最高で Detection Rate と Detection Precision ともに 100%をマークした．しかし，Mushroom では提案手法と確率的手法の間に，検出精度の差はほとんどなかったといえる．

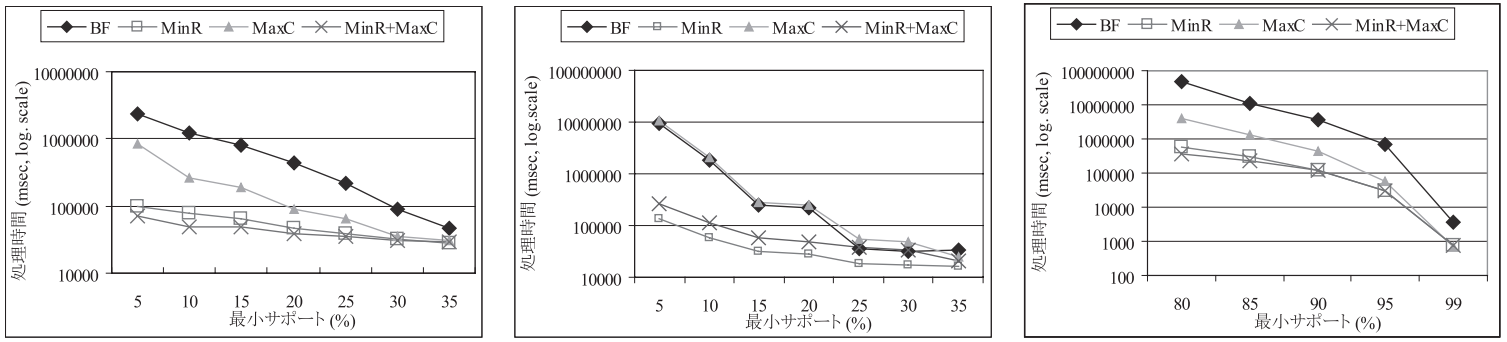
図 3 (b) は KDD Cup 99 に対する実験結果である．提案手法にはパラメータ $msup = 80\%$，$mconf = 80\%$，$mod = 0.15$，FI 手法には $msup = 80\%$，確率的手法には $size = 2$，$\alpha = 0.15$ をそれぞれ与えた．提案手法と FI 手法の折れ線はほとんど一致した．これは，KDD Cup 99 は属性値間の相関性が非常に強いデータで，頻出アイテム集合の集合と，その後生成される高確信度ルールの集合とで，情報としてほとんど差が生じなかったためである．確率的手法の精度が途中まで右肩上がりになっているのは，最も小さい r_value を示したレコード（すなわち，最も起こりえないと判断されたレコード）が，実際には正解外れ値ではなかったためである．このデータにおいては，提案手法が確率的手法をはるかに上回る検出能力を示した．

以上の結果から，提案手法は関連手法[22),24)] と比べて，同等以上の高い検出精度を得られていることが分かった．

6.5　処理速度比較

ここでは，提案アルゴリズムにより処理速度が向上していることを確認する．以下の 4 つの手法で処理速度を比較する．1 つは，表 2 に示されているブルートフォースアルゴリズム（BF）である．2 つ目は，最小ルール集合を利用した冗長ルールの除去工夫のみを取り入れたアルゴリズム（MinR），3 つ目は，極大相関性被覆を利用した候補外れ値レコードの絞り込み工夫を取り入れた手法（MaxC），そして最後に，冗長ルールを除去し，候補レコードを絞り込む 2 つの工夫を取り入れた手法（MinR+MaxC）である．Zoo10000 データに対

© 2008 Information Processing Society of Japan



(a) Zoo10000　(b) Mushroom100　(c) KDD Cup 99

図 4　最小サポートの変動に対する処理時間の比較
Fig. 4　Runtime comparison when the minimal support changes.

表 11　各データセットの対応するトランザクションデータの要素数
Table 11　Information on the corresponding transaction datasets.

データセット	$\|DB\|$	n	d
Zoo10000	1010000	11	22
Mushroom100	425800	13	120
KDD Cup 99	2053	17	258

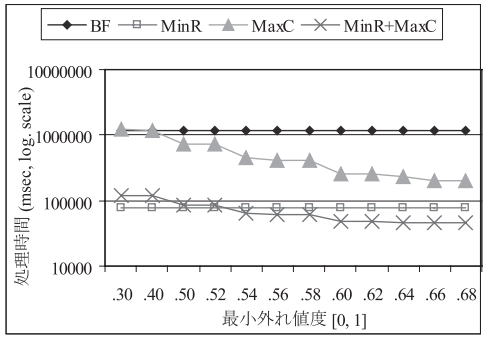
する検出では，ブール値属性の値 false に関するアイテムを無視するマッピングオプションを利用した．

この実験では Zoo10000，Mushroom100，KDD Cup 99 の 3 つのデータセットを使用した．処理速度に最も影響を与えるのは，入力データ DB の行数 $|DB|$ とトランザクションの大きさ n，およびアイテムのドメインの大きさ d である．d は頻出アイテム集合の集合の数 F と，高確信度ルールの数 R に影響を与える値である．処理速度の比較を解析しやすくするために，**表 11** に各データセットに対するこれらの情報をまとめる．ここで，マッピングオプションを利用した Zoo10000 データにおいて，n の大きさは各トランザクションで異なるが，表 11 では最大の値を記述する．

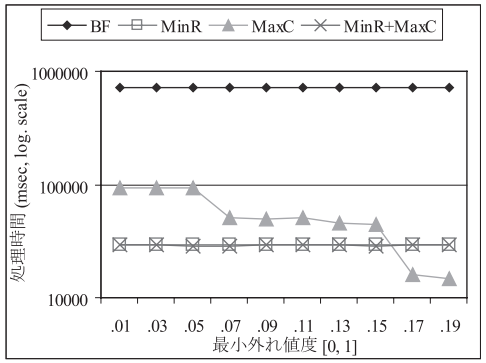
与えられる最小サポートの値によって，冗長ルールの除去工夫の有効性を左右する高確信度ルールの数と，候補外れ値レコードの絞り込み工夫の有効性を左右する極大頻出アイテム集合の数がそれぞれ変化する．2 つの工夫がそれぞれどれくらいきいているかを確かめるため，まずは最小サポートを変化させたときの処理時間の比較を行った．**図 4** は処理時間の変動の様子を，ログスケールで表したものである．Zoo10000 に対しては，パラメータ $(mconf, mod) = (90\%, 0.6)$ を与えた（図 4 (a)）．Mushroom100 に対しては，パラメータ $(mconf, mod) = (20\%, 0.3)$ を与えた（図 4 (b)）．KDD Cup 99 に対しては，パラメータ $(mconf, mod) = (95\%, 0.15)$ を与えた（図 4 (c)）．Zoo10000 データ，Mushroom100 データ，KDD Cup 99 データから，MinR が非常に高速に処理を行えている様子が分かる．MaxC は，Zoo10000 データと KDD Cup 99 データでは BF より高速に処理できているが，Mushroom100 データでは，逆に BF よりも遅くなる結果となった．図 2 (b) から明らかなように，Mushroom は，各レコードで外れ値の上限が非常に高いため，上限の計算による候補外れ値の絞り込みがほとんどできなかったと考えられる．このことから，候補外れ値の絞り込み工夫は，頻出アイテム集合の平均濃度が比較的大きい密なデータに対しては不向きであることが分かった．MinR+MaxC は，Zoo10000 データに対しては最小サポートが小さくなるほど高い有効性を示した．KDD Cup 99 データに対しては，若干 MinR よりも高速に処理できたが，グラフからは分からないほどの差であり，冗長ルールの除去工夫のみでも十分な高速性を得られることが分かった．全体的に，MinR は MaxC よりもずっと処理時間を削減できていた．

次に，最小外れ値度を変動させたときの各アルゴリズムの処理時間を測定した．**図 5** は各アルゴリズムの処理時間の動きを，ログスケールで表したものである．最小外れ値度の値に

© 2008 Information Processing Society of Japan



(a) Zoo10000



(b) KDD Cup 99

図 5　最小外れ値度の変動に対する処理時間の比較
Fig. 5　Runtime comparison when the minimal outlier degree changes.

有効性を左右されるのは，候補外れ値レコードの絞り込み工夫のみであるので，Mushroom データに対する実験結果は省略する．また，BF と MinR の処理速度は最小外れ値度の変動によって変化しない．Zoo データには $msup = 10\%$，$mconf = 90\%$を，KDD Cup 99 データには $msup = 80\%$，$mconf = 80\%$をそれぞれ与えた．どちらのデータからも，最小外れ値度を大きくするほど候補外れ値レコードの絞り込み工夫による処理速度の向上が認められた．

関連手法との速度比較

図 6 は，He らの FI 手法と Das らの確率的手法との処理速度の比較結果である．6.4 節の精度評価で最も良い F-measure の値が得られたときのパラメータセットを各手法に与え，処理時間を測定した．ここで使用したデータは，6.4 節で使用したデータに基づき，Mushroom100 と KDD Cup 99 である．

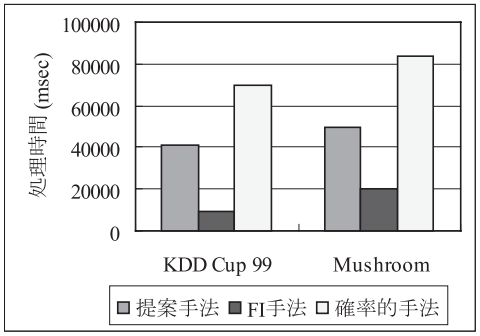


図 6　関連手法との処理速度比較
Fig. 6　Runtime comparison with related methods.

FI 手法で正常度を計算するための時間計算量は頻出アイテム集合の数 $|F|$ に対して $O(n \log n \times |F| \times |DB|)$ である．実験では FI 手法のための F の獲得に FP-growth を用いた．

確率的手法の正常度の計算では，レコード内の排他な 2 つのアイテム集合のあらゆるペアに対して確率の計算を行うので，$O(\sum_{k=2}^{d} {}_dC_k \cdot \sum_{j=1}^{k-1} {}_kC_j \times |DB|) = O((\frac{1}{2} \cdot 3^n - 2^n + \frac{1}{2}) \times |DB|)$ となる．また，この手法では集合濃度 $2 \cdot size$ 以下のあらゆるアイテム集合の出現数を学習データから数え上げる．実験では，この数え上げに FP-growth アルゴリズムを用いた．

7. 議　　論

X，Y（$X \cap Y = \emptyset$）を頻出アイテム集合，y_i を Y の要素アイテム（$y_i \in Y$，$1 \leq i \leq |Y|$）とすると，一般に次のことが成り立つ．

$conf(X \rightarrow Y) \geq mconf \Rightarrow$
$conf(X \rightarrow \{y_1\}) \geq mconf \wedge \cdots \wedge$
$conf(X \rightarrow \{y_{|Y|}\}) \geq mconf$

しかし，逆命題は必ずしも成り立つとは限らない．本研究における提案外れ値度の計算においては，結果的に，$conf(X \rightarrow Y) \geq mconf$ と $conf(X \rightarrow \{y_1\}) \geq mconf$

© 2008 Information Processing Society of Japan

$\wedge\cdots\wedge\ conf(X \rightarrow \{y_{|Y|}\}) \geq mconf$ を区別しない計算となっている．したがって，もし $conf(X \rightarrow \{y_1\}) \geq mconf \wedge\cdots\wedge\ conf(X \rightarrow \{y_{|Y|}\}) \geq mconf$ は成立するものの，$conf(X \rightarrow Y) \geq mconf$ は成り立たないデータの場合は，必ずしも妥当な外れ値度が得られない可能性がある．一方，このような簡略化を行うメリットの 1 つは，考慮すべきルールの数を削減できることである．実際，$X \rightarrow Y$ が高確信度ルールであった場合，このような簡略化を行わない場合は，$X \rightarrow Z$（$Z \subseteq Y$）をもすべて個別に考慮する必要があり，そのようなルールの数は $2^{|Y|} - 1$ となる．また，本提案では，最小確信度 $mconf$ は十分大きい値をとることを想定しており，その場合は，$conf(X \rightarrow \{y_1\}) \geq mconf \wedge\cdots\wedge\ conf(X \rightarrow \{y_{|Y|}\}) \geq mconf$ ならば，$conf(X \rightarrow Y)$ も十分に高い確信度を示すことが期待される．実際，6.4 節では $mconf = 80\%$ で精度評価実験を行っており，得られた結果が妥当であることが示されておる．

8. おわりに

本稿では，カテゴリ型レコードデータに対してより汎用的に利用可能な外れ値検出の提案を目的とした．まず，カテゴリ型レコードデータにおける外れ値とはどんなものであるか議論し，レコード内に出現している属性値の集合と本来ならば共起するはずの属性値が数多く観測できていないようなレコードを，外れ値と見なすこととした．属性間の相関性として高い確信度を持つ相関ルールを利用し，外れ値度の式を導出した．また，より高速な検出を実現するために，ブルートフォースアルゴリズムより効率的な検出アルゴリズムを提示した．この提案アルゴリズムは，与えられたパラメータセットに対して，データ中に外れ値が存在しない場合に，処理を途中で終了するファストパスと，外れ値度の単調増加生を利用した候補外れ値の絞り込み工夫，冗長ルールの除去工夫を取り入れたものである．実験では，提案手法が実際に外れ値として説得力のあるレコードを検出しうること，提案外れ値度が 2 つの関連手法と比較しても十分な検出精度を得る力があること，提案アルゴリズムがブルートフォースアルゴリズムに比べ，高速な検出処理を行えていることを示した．今後の課題としては，提案手法のパラメータセンシティビティを解消するために，最適なパラメータセットを発見する方法について検討する必要があると考えている．また，提案した計算処理効率化工夫の 1 つである候補外れ値の絞り込みは，頻出アイテム集合の平均濃度が比較的大きくなる性質のある密データに対しては十分な有効性を示せないという結果が得られたことから，さらなる計算効率化を目指して，アルゴリズムの洗練を行う予定である．

謝辞　貴重なコメントをいただいた査読者の方々に感謝します．本研究の一部は科学研究費特定領域研究（#19024006）の助成による．

参考文献

1) Han, J. and Kamber, M.: *Data Mining: Concepts and Techniques, 2nd ed.*, The Morgan Kaufmann Series in Data Management Systems (2006).
2) Agrawal, R., Mannila, H., Srikant, R., Toivonen, H. and Verkamo, A.I.: Fast Discovery of Association Rules, *Advances in Knowledge Discovery and Data Mining 1996*, pp.307–328 (1996).
3) Agrawal, R. and Srikant, R.: Fast Algorithms for Mining Association Rules in Large Databases, *VLDB*, pp.487–499 (1994).
4) Burdick, D., Calimlim, M. and Gehrke, J.: MAFIA: A Maximal Frequent Itemset Algorithm for Transactional Databases, *ICDE*, pp.443–452 (2001).
5) He, Z., Huang, J.Z., Xu, X. and Deng, S.: Mining Class Outliers: Concepts, Algorithms and Applications, *WAIM*, pp.589–599 (2004).
6) Hong, T.-P., Lin, K.-Y. and Wang, S.-L.: Fuzzy data mining for interesting generalized association rules, *Fuzzy Sets and Systems*, Vol.138, No.2, pp.255–269 (2003).
7) Pei, J., Han, J. and Mao, R.: CLOSET: An Efficient Algorithm for Mining Frequent Closed Itemsets, *ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, pp.21–30 (2000).
8) Srikant, R. and Agrawal, R.: Mining Generalized Association Rules, *VLDB*, pp.407–419 (1995).
9) Wang, J., Han, J. and Pei, J.: CLOSET+: searching for the best strategies for mining frequent closed itemsets, *KDD*, pp.236–245 (2003).
10) Zaki, M.J. and Hsiao, C.-J.: CHARM: An Efficient Algorithm for Closed Itemset Mining, *SDM* (2002).
11) Knorr, E.M. and Ng, R.T.: Algorithms for Mining Distance-Based Outliers in Large Datasets, *VLDB*, pp.392–403 (1998).
12) Ramaswamy, S., Rastogi, R. and Shim, K.: Efficient Algorithms for Mining Outliers from Large Data Sets, *SIGMOD Conference*, pp.427–438 (2000).
13) Aggarwal, C.C. and Yu, P.S.: Outlier Detection for High Dimensional Data., *SIGMOD Conference*, pp.37–46 (2001).
14) Arning, A., Agrawal, R. and Raghavan, P.: A Linear Method for Deviation Detection in Large Databases, *KDD*, pp.164–169 (1996).
15) Breunig, M.M., Kriegel, H.-P., Ng, R.T. and Sander, J.: LOF: Identifying Density-Based Local Outliers, *SIGMOD Conference*, pp.93–104 (2000).
16) Jagadish, H.V., Koudas, N. and Muthukrishnan, S.: Mining Deviants in a Time Series Database, *VLDB*, pp.102–113 (1999).
17) Knorr, E.M. and Ng, R.T.: Finding Intensional Knowledge of Distance-Based Out-

© 2008 Information Processing Society of Japan

liers, *VLDB*, pp.211–222 (1999).
18) Papadimitriou, S., Kitagawa, H., Gibbons, P.B. and Faloutsos, C.: LOCI: Fast Outlier Detection Using the Local Correlation Integral, *ICDE*, pp.315–326 (2003).
19) Rousseeuw, P.J. and Leroy, A.M.: *Robust Regression and Outlier Detection*, John Wiley and Sons (1987).
20) Zhu, C., Kitagawa, H. and Faloutsos, C.: Example-Based Robust Outlier Detection in High Dimensional Datasets, *ICDM*, pp.829–832 (2005).
21) Chan, P.K., Mahoney, M.V. and Arshad, M.H.: A Machine Learning Approach to Anomaly Detection, Technical report, Florida Institute of Technology (2003).
22) Das, K. and Schneider, J.G.: Detecting anomalous records in categorical datasets, *KDD*, pp.220–229 (2007).
23) Lee, W., Stolfo, S.J. and Mok, K.W.: A Data Mining Framework for Building Intrusion Detection Models, *sp*, Vol.00, p.0120 (1999).
24) He, Z., Xu, X. and Deng, S.: FP-outlier: Frequent pattern based outlier detection, Technical report, Harbin Institute of Technology (2002).
25) Han, J., Pei, J. and Yin, Y.: Mining Frequent Patterns without Candidate Generation, *Proc. 2000 ACM SIGMOD International Conference on Management of Data*, pp.1–12 (2000).
26) Grahne, G. and Zhu, J.: Efficiently Using Prefix-trees in Mining Frequent Itemsets., *FIMI* (2003).
27) Asuncion, A. and Newman, D.: UCI Machine Learning Repository, University of California, Irvine, School of Information and Computer Sciences (2007). http://www.ics.uci.edu/ ∼mlearn/MLRepository.html.

(平成 20 年 3 月 20 日受付)
(平成 20 年 7 月 14 日採録)

(担当編集委員　森本 康彦)

成田　和世

2004 年筑波大学第三学群情報学類を修士課程への飛び入学のため中途退学．2006 年同大学大学院理工学研究科理工学専攻にて修士号取得．同年同大学院システム情報工学研究科博士後期課程に入学，現在に至る．データマイニングに関する研究に従事．日本データベース学会学生会員．

北川　博之（フェロー）

1978 年東京大学理学部物理学科卒業．1980 年同大学大学院理学系研究科修士課程修了．日本電気（株）勤務の後，1988 年筑波大学電子・情報工学系講師．同助教授を経て，現在，筑波大学大学院システム情報工学研究科教授，ならびに計算科学研究センター教授．理学博士（東京大学）．異種情報源統合，XML とデータベース，データマイニング，センサデータベース，WWW データ管理等の研究に従事．著書『データベースシステム』(昭晃堂),『The Unnormalized Relational Data Model』（共著，Springer-Verlag）等．電子情報通信学会フェロー，日本データベース学会理事，ACM，IEEE-CS，日本ソフトウェア科学会各会員．

© 2008 Information Processing Society of Japan