DEWS2005 6C-o4

アイテムセットの時系列化による隠れアイテムセット抽出手法

平手　勇宇† 　岩橋　永悟† 　山名　早人††,†††

† 早稲田大学大学院理工学研究科　〒 169-8555 東京都新宿区大久保 3-4-1
†† 早稲田大学理工学術院　〒 169-8555 東京都新宿区大久保 3-4-1
††† 国立情報学研究所　〒 101-8430 東京都千代田区一ツ橋 2-1-2
E-mail: †{hirate,eigo}@yama.info.waseda.ac.jp, ††yamana@waseda.jp

あらまし　データマイニング分野における頻出アイテムセット抽出手法は，データベース中に頻繁に出現するアイテムセットを抽出する．こうした頻出アイテムセット抽出手法は，データベース中に格納されている過去のデータに対して適用されるため，過去には頻出でなかったアイテムセット，すなわち今後頻出となるであろうアイテムセットを抽出することは出来ない．これに対し本稿では,「アイテムの時系列上での出現タイミングが過去の頻出アイテムセットの出現タイミングと似ているアイテムセット」に着目する。このようなアイテムセットは、過去においては頻出ではなかったが将来頻出になる可能性を持つ、あるいは、スーパーマーケット等でこのようなアイテムセットを同じ棚に並べることで頻出に導くことができる可能性があると考える．そして今後頻出に導くことができる可能性のあるアイテムセットを，(1) トランザクションデータベースにはほとんど出現していないが，(2) 出現率が高いアイテムセットをサブセットとして含み，かつ (3) トランザクションデータベース中における時系列上での出現タイミングが似ているアイテムセットと定義し,「隠れアイテムセット」と呼ぶことにする．本稿では，(1) アイテムセットのサポート値の時系列化による，アイテムセットのクラスタリング手法，(2) クラスタリングしたアイテムセットを元に，隠れアイテムセットを抽出する手法を提案する．

キーワード　データマイニング，頻出アイテムセット抽出，時系列データ，クラスタリング

Hidden Itemset Mining Method based on the Clustering of Itemset Time Series

Yu HIRATE†, Eigo IWAHASHI†, and Hayato YAMANA††,†††

† Graduate School of Science and Enginnering, Waseda University　3-4-1 Okubo, Shinjuku-ku, Tokyo 169-8555 Japan
†† Science and Engineering, Waseda University　3-4-1 Okubo, Shinjuku-ku, Tokyo 169-8555 Japan
††† National Institute of Infomatices　2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo 101-8430 Japan
E-mail: †{hirate,eigo}@yama.info.waseda.ac.jp, ††yamana@waseda.jp

Abstract　In data mining researches, frequent itemsets mining extracts itemsets which appear frequently in databases. Since frequent itemsets mining refer to dataset which stores past data, it can't extract itemsets which is infrequent in the past, and which can make to be frequent in the future. In this paper, we focus on itemsets whose appearance pattern in time series are similar to appearance pattern of high support frequent itemsets in the past. And we suppose that even these itemsets is infrequent in the past, these itemsets can make to be frequent in the future by feed backing to the real world. We define itemsets ,which can make to be frequent in the future, as itemsets (1)which appear in database few times, (2)whose subsets are a part of high support itemsets, and (3)whose appearance time patterns are similar to other high support frequent itemsets, and we call these itemsets "Hidden Itemset". In this paper, we propose (1) itemsets clustering method based on time series data of itemset support, and (2)missing itemsets generation method based on itemset clustering method.

Key words　DataMining,Frequent Itemset Mining,Time Sereis, Clustering

1. はじめに

近年，ネットワーク環境の整備，記憶装置の低価格化・大容量化に伴い，大量のデータを蓄積することが一般的になってきた．しかし，集められた大量のデータは記号の列に過ぎない．大量のデータの中から新しい知識となる情報を抽出することは，人間では不可能であることから，データマイニング技術が注目されている．

データマイニング技術の重要な問題として，頻出アイテムセット抽出手法がある [1]．頻出アイテムセット抽出手法とは，データベースの中から頻出アイテムセット集合を抽出する手法である．頻出アイテムセット集合とは，ユーザが定義した最小サポート値よりも高い頻度でデータベース中に出現するアイテム集合である．頻出アイテムセット抽出問題は，93 年に Agrawal らによって定義され [1]，94 年に同じく Agrawal らによって頻出アイテムセットを高速に抽出する Apriori アルゴリズム [2] が提案されてから，さまざまな速度向上のための改良手法が提案されている [3] [4] [5] [6] [7].

頻出アイテムセット抽出手法は，データベースに格納されている過去のデータを見ることで，過去に顕著に現れていたアイテムセットを抽出する手法である．そのため，過去では非頻出であったが，これから頻出となる可能性のあるアイテムセットを抽出することは出来ない．したがって，従来，これから頻出となる可能性のあるアイテムセットは，頻出アイテムセット抽出によって得られた過去に頻出であったアイテムセットを基にして，ユーザが直接推定しなければならない．

ここで「**相関関係がある**」アイテム（セット）とは，相関ルール定義より「トランザクションデータベース中に**同時に出現する**」アイテム（セット）のことである．したがって,「**相関関係が強い**」可能性のあるアイテムとは,「トランザクションデータベース中に**同じような時期に出現する**」アイテム（セット）と考えることにする．もともとトランザクションは，時間の概念を持つ．つまり，ひとつのトランザクションは，ある時間にどのようなイベントが発生（トランザクション）したかを表現するものである．したがって,「トランザクションデータベース中に同じような時期に出現する」アイテム（セット）とは，時系列上で考えると，似たタイミングでトランザクションデータベース中に出現することに置き換えることが出来る．したがって，アイテム（セット）のサポート値を時系列化し，時系列化したサポート値に似た傾向があれば，それらのアイテム（セット）は相関関係が強い可能性があると判断する．たとえば，トランザクションデータベースをスーパーマーケットの POS データとして考えた場合，時系列化したサポート値が似ているということは，同じ時間帯に売れている商品となり，同じような客層によって購入されている可能性が高いと判断することができる．

本稿では，データベース中における時系列上での出現タイミングが似ているアイテムセットに含まれるアイテムを組み合わせることによって生成されるアイテムセットを，実社会へフィードバックさせることにより，頻出に導くことができる可能性があると仮定する．そして，対象とするデータベースには顕著に現れていないアイテムセットで，今後，頻出に導くことが出来る可能性があるアイテムセットを，

- 頻出アイテムセット集合のある要素をサブセットとして含み，
- 頻出アイテムセット抽出において抽出されにくい低いサポート値を持ち，
- かつ頻出アイテムセットとトランザクションデータベース中における出現タイミングが似ている

アイテムセットと定義し,「隠れアイテムセット」と呼ぶ．ここで出現タイミングが似ていることは，データベース中において同じような時間に多く出現し，同じような時間に出現数が減ることをさす．

たとえば，スーパーマーケットの POS データにおいてある時間帯に「大根」と「おでんの素」を買う顧客が多く，さらに同じ時間帯に,「大根」と「さんま」を買う顧客が多いことも分かったとする．しかし,「大根」,「おでんの素」,「さんま」を買う客はほとんどいないが,「大根」と「おでんの素」を買う顧客と同じ時間帯に増加したとする．ここで，この POS データに対して，従来の頻出アイテムセット抽出を行うと，当然，アイテムセット {”大根”,”おでんの素”} と，アイテムセット {”大根”,”さんま”} は出現頻度が高く抽出されるが，アイテムセット {”大根”,”おでんの素”,”さんま”} は出現頻度が低いので抽出されにくい．しかし頻出アイテムセット抽出を行うユーザ側からすれば，アイテムセット {”大根”,”おでんの素”,”さんま”} は，たとえば，特価品のチラシに,「大根」の近くに「おでんの素」と「さんま」を配置することによって，まとめ買いを促すような顧客への新しい提案が出来る可能性があるとして，有用であると考える．この例においての「隠れアイテムセット」は，{”大根”,”おでんの素”,”さんま”} のアイテムセットである．

隠れアイテムセットを抽出するためには，頻出アイテムセット集合のうち，トランザクションデータベースにおける時系列上で似たタイミングに出現する複数のアイテムセットを結合することにより生成することが可能である．

以上のような考えに基づき，本稿では，ある最小サポート値によって得られた頻出アイテムセット集合を基にして，隠れアイテムセットを抽出する効率の良い手法を提案する．提案手法は，頻出アイテムセットと相関性の可能性が高いアイテムもしくはアイテムセットを特定するための (1) **アイテムセットのサポート値の時系列化に基づくアイテムセットのクラスタリング手法**，そして (2) **クラスタリングしたアイテムセット集合から，隠れアイテムセットを生成する手法**の 2 つの手法で構成される．

以下，第 2 節では，アイテムセットのクラスタリング手法を提案する．第 3 節では，クラスタリングされたアイテムセットから隠れアイテムセットを生成する手法を提案する．第 4 節では，第 2 節，第 3 節で提案した手法を処理するアルゴリズムについて述べる．第 5 節では，第 2 節，第 3 節で提案した手法の性能評価を行い，最後に，第 6 節でまとめを行う．

2. アイテムセットのクラスタリング手法

アイテムセットのクラスタリング手法とは，ある最小サポー

ト値で抽出されたすべての頻出アイテムセット集合のサポート値を時系列化し，時系列化したサポート値をクラスタリングすることによって，アイテムセットのグループ化を行う手法である．ここで，時系列化したサポート値とは，対象となるトランザクションデータベースを時系列上で m 個に分割し，分割した m 個のトランザクションデータベースそれぞれにおいてのサポート値を並べたものである．この手法で同一クラスタに属するアイテムセット集合は，似たタイミングでトランザクションデータベースに出現する頻度が増減することを示す．したがって，同一クラスタに属するアイテムセットは，相関性が強い可能性が高いと判断することができる．

2.1 頻出アイテムセットの定義

アイテム集合を $I = \{i_1, i_2, \ldots, i_q\}$ とする．アイテムセット X は，アイテム集合 I のサブセットである．トランザクションデータベース（TDB）を $TDB = \{t_1, t_2, \ldots, t_n | t_i \in I, 1 \leqq i \leqq n\}$ とする．TDB の各要素 t_i をトランザクションとし，トランザクションは，発生した時間でソートされており，t_n がもっとも新しく発生したトランザクションとし，t_1 がもっとも古く発生したトランザクションとする．

TDB が与えられた場合，アイテムセット X のサポート $sup(X, TDB)$ は，TDB 全体に対して X を含むトランザクション数を表す．頻出アイテムセット集合 (FI) とは，ユーザが与えた最小サポート値（min_sup）以上のサポート値を持つアイテムセット (fi) の集合である．

2.2 頻出アイテムセットのサポート値の時系列データ化

ある最小サポート値を指定することによって抽出された p 個の頻出アイテムセット集合 FI を，

$$FI = \{fi_1, fi_2, \ldots, fi_p\} \tag{1}$$

とする．本節では，すべての頻出アイテムセットのサポート値を時系列データ化するプロセスを述べる．

まず，ウィンドウサイズ w を定義し，TDB を時系列上に m 個に分割する．ここで m は

$$m = \lfloor \frac{n}{w} \rfloor \tag{2}$$

である．セグメント化された TDB は，各々w 個のトランザクションによって構成される．

$$TDB = \{TDB_1, TDB_2, \ldots, TDB_m\} \tag{3}$$

のように m 個にセグメント化される．したがって，i 番目にセグメント化された TDB_i は，

$$TDB_i = \{t_{w\times(i-1)+1}, t_{w\times(i-1)+2}, \ldots, t_{w\times i}\} \tag{4}$$

のようになる．なお，TDB_m には TDB を w でセグメント化した端数となるトランザクションで構成されているため，w 以下のトランザクションで構成されている．

あるアイテムセット X の時系列化したサポート値を，$TSSup(X)$ (=Time Series Support) と表現し，

$$TSSup(X) = \{sup(X, TDB_1), sup(X, TDB_2), \ldots, sup(X, TDB_m)\} \tag{5}$$

と定義する．ここで，$sup(X, TDB_i)$ は，TDB_i を対象としたアイテム（セット）X サポート値のことである．したがって，

$$sup(X, TDB) = \sum_{i=1}^{m} sup(X, TDB_i) \tag{6}$$

となる．

ある最小サポート値で抽出された頻出アイテムセット集合 $\{fi | \forall fi \in FI\}$ の各々の要素に対し，$TSSup(fi)$ を生成する事で頻出アイテムセットのサポート値を時系列化する．

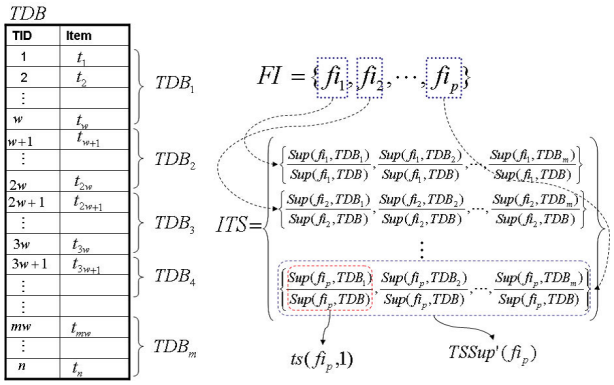

図 1　TDB のセグメント化と ITS の生成

2.3 時系列データの相対化

生成した $TSSup(fi)$ をひとつのオブジェクトとして，抽出されたすべての頻出アイテムセットをクラスタリングする．クラスタリングにあたっては，アイテムセットの出現頻度の相対的な変異が似ているものを同一クラスタに配置する．これは，$TSSup(fi)$ をそのままクラスタリングのオブジェクトにすると，$TSSup(X)$ の定義より，出現頻度が似ていても $TSSup(X)$ が絶対指標なため，正しくクラスタリングすることが出来ないからである．

$TSSup(X)$ の要素が X によらず相対的な値となるように，$TSSup'(X)$ を次式で定義する．

$$\begin{aligned} TSSup'(X) &= \{\frac{sup(X, TDB_1)}{sup(X, TDB)}, \frac{sup(X, TDB_2)}{sup(X, TDB)}, \\ &\quad \ldots, \frac{sup(X, TDB_m)}{sup(X, TDB)}\} \\ &= \{ts(X, 1), ts(X, 2), \ldots, ts(X, m)\} \end{aligned} \tag{7}$$

2.4 相対化した時系列データのクラスタリング

相対化したすべてのアイテムセットの時系列データを基に，クラスタリングを行う．クラスタリングの入力である ITS(=Itemset Time Series) は，一行をひとつの頻出アイテムセット fi の $TSSup'(fi)$ として，すべての頻出アイテムセットの $TSSup'(fi)$ $(1 \leqq i \leqq p)$ を並べたものであり，$(p \times m)$ の行列である．

$$ITS = \begin{pmatrix} TSSup'(fi_1) \\ TSSup'(fi_2) \\ \vdots \\ TSSup'(fi_p) \end{pmatrix}$$

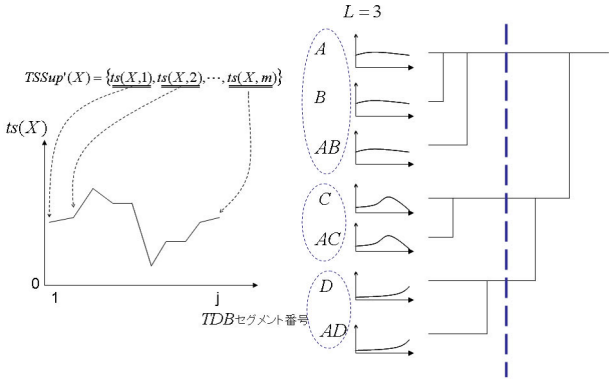

図 2　アイテムセットのクラスタリング

$$= \begin{pmatrix} ts(fi_1,1) & ts(fi_1,2) & \dots & ts(fi_1,m) \\ ts(fi_2,1) & ts(fi_2,2) & \dots & ts(fi_2,m) \\ \vdots & \vdots & \ddots & \vdots \\ ts(fi_p,1) & ts(fi_p,2) & \dots & ts(fi_p,m) \end{pmatrix} \quad (8)$$

図 1 に，TDB のセグメント化と ITS の生成方法を示す．

クラスタリングで用いるオブジェクト間の距離には，ユークリッド距離を用いる．すなわち任意のアイテムセット X，Y 間の距離は，

$$D(X,Y) = \sqrt{\sum_{j=1}^{m} \Big(ts(X,j) - ts(Y,j)\Big)^2} \quad (9)$$

である．

クラスタリングは，階層的手法 (hierarchical) の凝集型 (agglomerative) により行う [9]．この手法は，あらかじめユーザがいくつのクラスタ L に分類するかを指定して，L 個のクラスタに分割する方法である．この手法は，1 個のオブジェクトだけを含む p 個のクラスタがある状態から開始する．クラスタ間の距離関数に基づき，もっとも距離の近い 2 つのクラスタを逐次的に併合する．クラスタリングプロセスは，L 個のクラスタに併合された時点で終了する．

図 2 に，アイテムセットのクラスタリングを行ったときの例を示す．図 2 では，$FI = \{A,\ B,\ C,\ D,\ AB,\ AC,\ AD\}$ として，生成するクラスタ数を 3 にしてクラスタリングを行っている．クラスタリングの結果生成された 3 つのクラスタは，$\{A,\ B,\ AB\}$，$\{C,\ AC\}$，$\{D,\ AD\}$ となる．なお，後述する候補隠れアイテムセットの所属クラスタを計算するために，生成したクラスタのセントロイド（中心点）を計算する．クラスタ $Cluster_i$ に所属するアイテムセット数を N_i 個とすると，クラスタ $Cluster_i$ のセントロイド $Center_i$ は，

$$Center_i = \Big\{center_{i,1}, center_{i,2}, \cdots, center_{i,m}\Big\} \quad (10)$$

$$center_{i,j} = \frac{\sum_{X \in Cluster_i} ts(X,j)}{N_i} \quad (11)$$

と定義される．

3. 隠れアイテムセット生成手法

次に，前節で述べたクラスタリングしたアイテムセット集合を基にして，隠れアイテムセットを生成する．本手法は，(1) 同一クラスタ内の極大アイテムセットの生成，(2) 隠れアイテムセット候補の生成，(3) 生成した候補アイテムセットの時系列化したサポート値の生成，そしてどの (4) どのクラスタに属するかの検証の 4 つのステップで構成されている．以下，3.1〜3.4 で 4 つのステップを個別に述べた後，3.5 で 4 ステップをどのように実行するかを述べる．

3.1　同一クラスタ内の極大アイテムセットの生成

同一クラスタに分類されたアイテムセット集合は，要素となるアイテムセット間の相関関係が強いため，アイテムセット間にサブセット－スーパーセットの関係が数多く成立する．このため，このまま隠れアイテムセットを生成すると効率が悪い．そこで前処理として，同一クラスタ内に分類されたアイテムセットをまとめ挙げる．具体的には，同一クラスタに分類されたアイテムセットをひとつのトランザクションデータベース，任意のひとつのアイテムセットをひとつのトランザクションと見立てて，極大アイテムセット [8] を生成する．

ここで関数 $CL(mi)$ は，極大アイテムセット mi を生成したクラスタを返す関数と定義する．

3.2　隠れアイテムセット候補の生成

本プロセスでは，ある極大アイテムセットから，相関性のある可能性が高いアイテムを結合することにより，対象極大アイテムセットよりも要素数が 1 つ多い隠れアイテムセット候補を生成する．ここで，極大アイテムセット mi_a と相関性のある可能性が高いアイテム $HAPI(mi_a)$ (=High Associate Probabily Item) とは，対象極大アイテムセット mi_a と同一クラスタに分類されているアイテムセット集合のサブセットであり，

$$HAPI(mi_a) = \Big\{i \mid \forall i \in fi, fi \in CL(mi_a), i \notin mi_a\Big\} \quad (12)$$

と定義する．

極大アイテムセット mi_a に関連する候補隠れアイテムセット集合 $CHI(mi_a)$(=Candidate of Hidden Itemset) は，mi_a に $HAPI(mi_a)$ の各要素を結合することで生成し，

$$CHI(mi_a) = \Big\{mi_a \cap i \mid \exists i \in HAPI(mi_a)\Big\} \quad (13)$$

と定義する．生成した候補アイテムセット集合 $CHI(mi_a)$ は，頻出アイテムセット mi_a をサブセットとして含み，mi_a よりも要素数がひとつ多いアイテムセットで構成される．

3.3　候補隠れアイテムセットの $TSSup'(X)$ の生成

3.2 で生成した隠れアイテムセット候補は，隠れアイテムセット候補のサブセットとなる頻出アイテムセットと TDB においての出現傾向が似ていなくてはならない．したがって，生成した隠れアイテムセット候補が，同一クラスタに分類されるかどうかを検証するために，生成した隠れアイテムセット候補すべてに対し，$TSSup'(X)$ を生成する．

候補隠れアイテムセットの $TSSup'(X)$ の生成方法は，頻出アイテムセットの $TSSup'(X)$ の生成方法と同様で，2.2，2.3 と同じ手法で生成する．

3.4　分類されるクラスタの検証

本プロセスでは，ある隠れアイテムセット候補の $TSSup'(X)$

と，2.4 で生成したクラスタの中でもっとも近いクラスタに，隠れアイテムセット候補のサブセットとなる頻出アイテムセットが含まれるかどうかを確認する．もっとも近いクラスタに，サブセットとなる頻出アイテムセットが含まれる場合，対象隠れアイテムセット候補は，隠れアイテムセットとなる．

ある隠れアイテムセット候補 X とクラスタ cl との距離 $Distance(X, cl)$ は，ユークリッド距離で表し，

$$Distance(X, cl) = \sqrt{\sum_{k=1}^{m} \left(ts(X,k) - center_{cl,k}\right)^2} \quad (14)$$

とする．そして，隠れアイテムセット候補 X からもっとも近いクラスタ $Nearest(X)$ は，$1 \leqq Y \leqq L$ において，

$$Nearest(X) = cl \mid Distance(X, cl) = min(Distance(X, Y)) \quad (15)$$

とする．

クラスタ cl から抽出された極大アイテムセット mi_X から生成される隠れアイテムセット集合 HI は，

$$HI = \left\{h | h \in CHI(mi_X), Nearest(h) = CL(mi_X)\right\} \quad (16)$$

となる．

3.5 隠れアイテムセット生成手法

3.1〜3.4 を順番に実行することで生成されるアイテムセットは，対象クラスタから抽出された極大アイテムセットよりも要素数が 1 つ多い隠れアイテムセット集合である．

対象クラスタから抽出された極大アイテムセットよりも要素数が 2 つ多い隠れアイテムセットが，同一クラスタに分類されるのであれば，サブセットとなる極大アイテムセットよりも要素数が 1 つ多い隠れアイテムセットも同一クラスタに分類されるというアプリオリな考えの下，極大アイテムセットよりも要素数が 2 つ多い隠れアイテムセットを生成するためには，要素数が 1 つ多い隠れアイテムセットを極大アイテムセットとして，3.1〜3.4 のプロセスを実行する．

以下要素数が増えたときも同様の処理で，極大アイテムセットよりも要素数が k 個多い隠れアイテムセットを生成するためには，要素数が $k-1$ 個多い隠れアイテムセットを極大アイテムセットとして 3.1〜3.4 のプロセスを実行する．候補アイテムセットが生成されなくなったときに，隠れアイテムセット生成を終了する．

4. アルゴリズム

本節では，2 節，3 節で示した二つの手法を実際に処理するアルゴリズムを示す．

4.1 アイテムセットのクラスタリングアルゴリズム

アイテムセットのクラスタリングアルゴリズムを以下に示す．なお，本アルゴリズム中の関数 genITS の擬似コードを，図 3 に示す．このアルゴリズムは，TDB のスキャンを 1 回必要とする，

Input 頻出アイテムセット集合 FI，トランザクションデータベース TDB，セグメントサイズ w，分類するクラスタ数 L

Output アイテムセットのクラスタ $Cluster_i, (1 \leqq i \leqq L)$，各クラスタのセントロイド $Center_i, (1 \leqq i \leqq L)$

Method

（1） FI を読み込む

（2） ITS を生成する．(関数 genITS(FI,TDB,w) を実行する．)

（3） 生成された ITS を元にクラスタリングを行い，L 個のクラスタに分ける．

（4） 生成したすべてのクラスタのセントロイドを計算する．

```
// FI = 頻出アイテムセット集合
// TDB = トランザクションデータベース
// w = ウィンドウサイズ
function genITS(FI,TDB,w){
  SegNo=1; //セグメント番号
  Trans=1; //トランザクション番号
  // すべてのアイテムの TSSup を生成
  while(TDB から 1 トランザクション (=t) を読み込み){
    for each(fi ∈ FI){
      if(t が fi を含む){
        ts(fi, SegNo)++;
        sup(fi)++; // アイテムセット fi の番号 SegNo のセグメント TDB のサポート値
      }
    }
    Trans++;
    if(Trans % w ==1){
      //TDB セグメント番号をインクリメント
      SegNo++;
    }
  } // TSSup の生成終了
  // TSSup から TSSup′ への変換
  for each(fi ∈ FI){
    for(a=1;a<=SegNo;a++){
      ts(fi, a) = ts(fi, a)/Sup(fi);
    }
  } // TSSup′ への変換終了
}
```

図 3　ITS 生成擬似コード

4.2 隠れアイテムセット生成アルゴリズム

4.1 でクラスタリングしたアイテムセットから隠れアイテムセットを生成するアルゴリズムを示す．なお，同一クラスタに分類されるアイテムセット集合から生成された極大アイテムセット集合を基にして，隠れアイテムセットを生成する関数 genHI の擬似コードを図 4 に示す．

Input アイテムセットクラスタ $Cluster_i, (1 \leqq i \leqq L)$，各クラスタのセントロイド $Center_i, (1 \leqq i \leqq L)$，トランザクションデータベース TDB, セグメントサイズ w，クラスタ数 L

Output 隠れアイテムセット集合 HI

Method

（1） $ClusterNum$ = 1 とする．

（2） クラスタ $ClusterNum$ の極大アイテムセット

```
// ClusterNum = 対象タイムセットクラスタ番号
// MaxItem = 極大アイテムセット集合
// TDB = トランザクションデータベース
// Center = クラスタ番号 ClusterNum のセントロイド
// w = TDB セグメントサイズ
function genHI(ClusterNum,MaxItem, TDB, Center, w) {
  BaseItemsets = MaxItem;
    // 生成する隠れアイテムセットのサブセット集合
  CHI = φ; // 候補隠れアイテムセット集合
  HI = φ; // 隠れアイテムセット集合
  do {
    HAPI = φ; // 結合するアイテム
    BaseItemsets に含まれるすべてのアイテムを HAPI に挿入する
    for each(X ∈ BaseItemsets) {
      for each(Add ∈ HAPI) {
        if(Add ∉ X) {
          CHI にアイテムセット {X ∩ Add} を挿入
        }
      }
    }
    CHI の重複を取り除く
    CHI の TSSup'(X) を生成する // TDB と w を参照する.

    for each(X ∈ CHI) {
      アイテムセット X から一番近いセントロイドがクラスタ番号 ClusterNum であれば，X を HI に挿入する.
    }
    BaseItemsets = HI;
    CHI = φ;
  } while (BaseItemsets が 2 つ以上のアイテムセット);
  HI の重複を取り除く
  return HI
}
```

図 4　隠れアイテムセット生成擬似コード

$MaxItem$ を生成する.

（3） クラスタ $ClusterNum$ に属する隠れアイテムセット $HI_{ClusterNum}$ を生成する.
($HI_{ClusterNum}$=genHI($ClusterNum$,$MaxItem$,TDB,$Center$,w) を実行する.)

（4） HI に $HI_{ClusterNum}$ を挿入する.

（5） $ClusterNum$ をインクリメントする.

（6） $ClusterNum > L$ が成立する場合は,（3）に戻る. 成立しない場合は終了する.

5. 評　　価

本節では，2 節，3 節で提案手法を，Retail データセット [10] に適用することによって，アイテムセットのクラスタリング手法，隠れアイテムセット生成手法の有効性について評価する. Retail データセットとは，1999 年から 2000 年にかけての 5 ヶ月間に記録された，ベルギーのあるスーパーマーケットの購買履歴データである. トランザクション数は，88163 トランザクションである. Retail データセットには，アイテムとして商品の番号のみが記録されている.

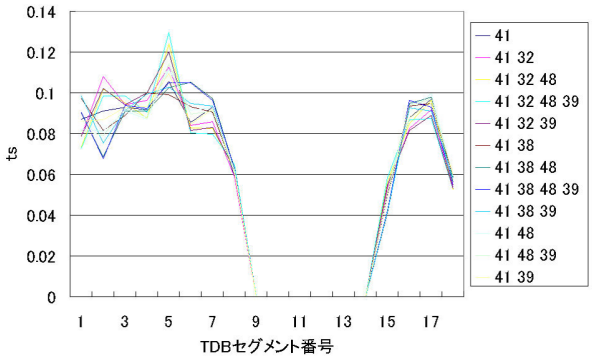

図 5　クラスタ ID7 番に属するアイテムセットの $TSSup'(X)$

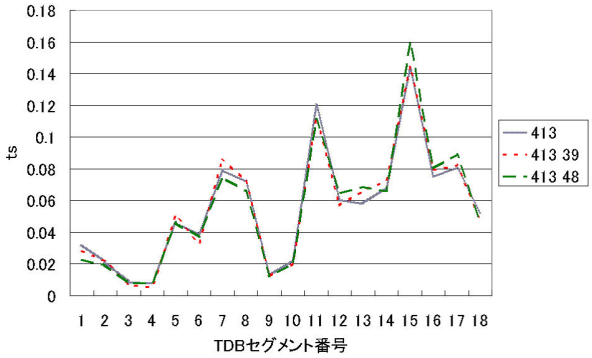

図 6　クラスタ ID8 番に属するアイテムセットの $TSSup'(X)$

5.1　アイテムセットのクラスタリング手法の評価

最小サポート値 min_sup=1000，セグメントサイズ w=5000，クラスタ数 L=20 とした場合のクラスタリングの結果を例に評価を行う. min_sup=1000 とした場合，頻出アイテムセットは 135 アイテムセット抽出された. クラスタ数 L=20 と指定しているため，20 個のアイテムセットクラスタが抽出される. 以下では，分類された頻出アイテムセット数 N が $3 \leqq N \leqq 15$ であり，グラフ化する上で視覚的に適当なクラスタを示す. クラスタ ID7 番に属するアイテムセットの $TSSup'(X)$ のグラフを図 5 に，同じくクラスタ ID8 番を図 6，クラスタ ID17 番を図 7，クラスタ ID18 番を図 8 に示す. 88163 トランザクションを w=5000 でセグメント化したため，セグメント数は 18 となり，各クラスタに分類されたアイテムセットの $TSSup(X)'$ が図 5〜8 に示されている.

図 5，図 6，図 7，図 8 に示すように，同一クラスタに属するアイテムセット集合は，サブセット－スーパーセットの関係が成立するアイテムセット集合が数多く存在する. さらに，クラスタを特徴付けるアイテムが存在することも確認できる. ここで「特徴付けるアイテム」とは，(1) 他のクラスタに分類された頻出アイテムセット集合の中には含まれておらず，かつ (2) 該当クラスタ中の頻出アイテムセット集合には多く出現するアイテムのことをさす.

ID7 番のアイテムセットクラスタは，アイテム”41”がすべて

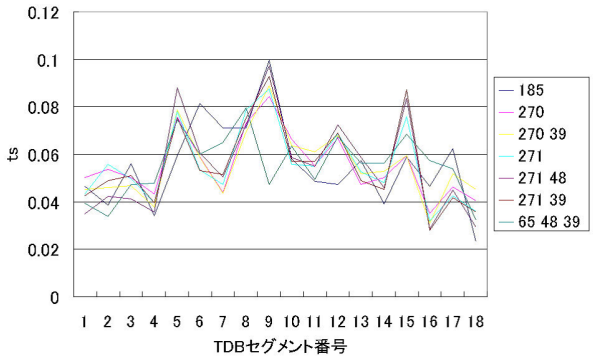

図 7　クラスタ ID17 番に属するアイテムセットの $TSSup(X)$

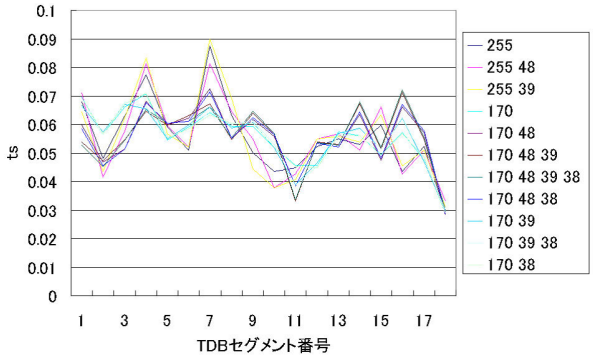

図 8　クラスタ ID18 番に属するアイテムセットの $TSSup'(X)$

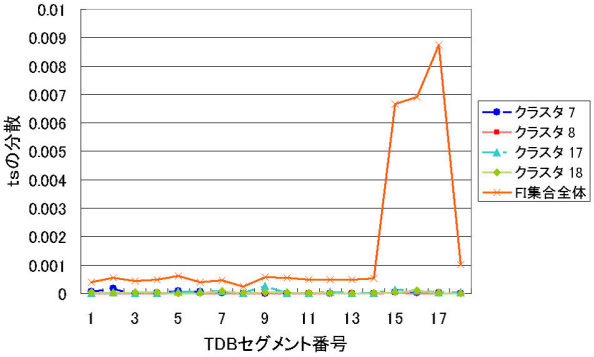

図 9　各セグメント TDB ごとの ts の分散

のアイテムセットに含まれており，ID7 番のクラスタを特徴付けている．同様に，ID8 番のアイテムセットクラスタはアイテム”413”番が，ID17 番のアイテムセットクラスタは，アイテム”270”と”271”番が，ID18 番のアイテムセットクラスタは，アイテム”225”と”170”番がそれぞれアイテムセットクラスタを特徴付けている．

また，セグメント TDB ごとに，ID7,8,17,18 番のクラスタに分類された頻出アイテムセットの ts の分散と，頻出アイテムセット集合 FI の ts の分散を比較したグラフを，図 9，10 に示す．図 9，10 は同じデータのグラフであるが，図 10 は Y 軸（ts の分散）のスケールを拡大したものである．

図 9，10 に示すように，頻出アイテムセット集合 FI の ts の分散は，各クラスタに分類された頻出アイテムセット集合の

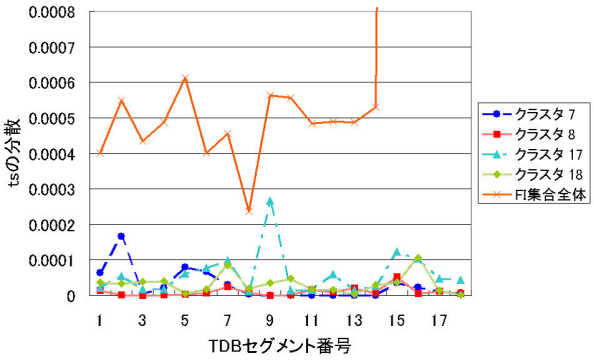

図 10　各セグメント TDB ごとの ts の分散

表 1　クラスタ ID7,8,17,18 番の極大アイテムセット

クラスタ ID	極大アイテムセット
7 番	{38 39 41 48}，{32 39 41 48}
8 番	{39 413}，{48 413}
17 番	{65 39 48}，{185}，{270 39}，{48 271}，{39 271}
18 番	{255 39}，{255 48}，{38 39 170 48}

分散よりも常に小さな値を持つ．また，TDB のセグメント番号 15〜17 番において頻出アイテムセット集合 FI の ts の分散は非常に大きく増加しているのにもかかわらず，各クラスタの ts の分散は大きく増加していない．これは，TDB のセグメント番号 15〜17 番において TDB に出現するトランザクションに変化があった，つまり，あるアイテムセットのサポート値は大きく増加し，あるアイテムセットのサポート値は大きく減少したにもかかわらず，同一クラスタに分類された頻出アイテムセットの TDB における出現傾向は同じであることを意味する．

以上より，頻出アイテムセットのクラスタリングを行うことによって，TDB における時系列上での出現傾向が似ているアイテムセットを同一クラスタに分類することが出来ることが確認できた．

5.2　隠れアイテムセット生成手法の評価

前節のアイテムセットクラスタリング手法の評価と同様に，最小サポート値 $min\_sup = 1000$，セグメントサイズ w=5000，クラスタ数 L=20 とした場合のクラスタリングの結果を例に評価を行う．

クラスタ ID7,8,17,18 番から抽出された極大アイテムセットは，表 1 のように抽出された．

表 1 に示す極大アイテムセット集合を基にして，隠れアイテムセットを抽出した結果，表 2 のように抽出された．クラスタ ID18 番からは，隠れアイテムセットは抽出できなかった．なお，表 2 で，アイテムセットの後に括弧内に示されている数字は，隠れアイテムセットのサポート値を示す．

また，隠れアイテムセットが抽出されたクラスタ ID7,8,17 番から抽出された隠れアイテムセットの，$TSSup'(X)$ と各クラスタのセントロイドの比較を行った．クラスタ ID7 番は図 11 に，クラスタ ID8 番は図 12 に，クラスタ ID17 番は図 13 に示す．

図 11，図 12，図 13 より，抽出された隠れアイテムセットの $TSSup'(X)$ は，各クラスタのセントロイドと比較して距離が

表 2　クラスタ ID7,8,17,18 番から生成された隠れアイテムセット

クラスタ ID	隠れアイテムセット (括弧内はサポート値)
7 番	{32 38 39 41 48}(448)
8 番	{39 48 413}(781)
17 番	{39 48 271}(827)，{39 270 271}(484)，{39 65 271}(106)，{48 270 271}(369)，{39 48 270 (733)}
18 番	ϕ

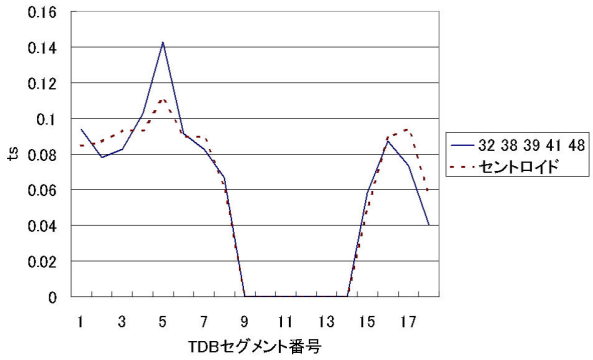

図 11　クラスタ ID7 番から抽出された隠れアイテムセットとセントロイドの比較

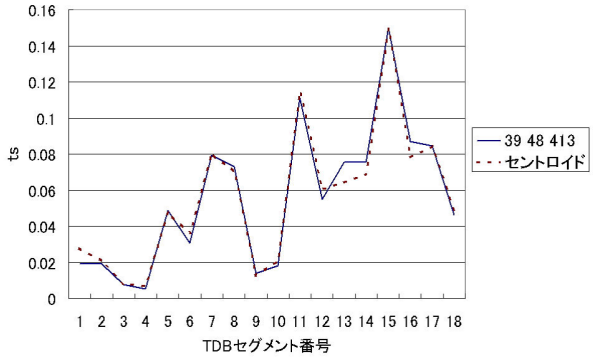

図 12　クラスタ ID8 番から抽出された隠れアイテムセットとセントロイドの比較

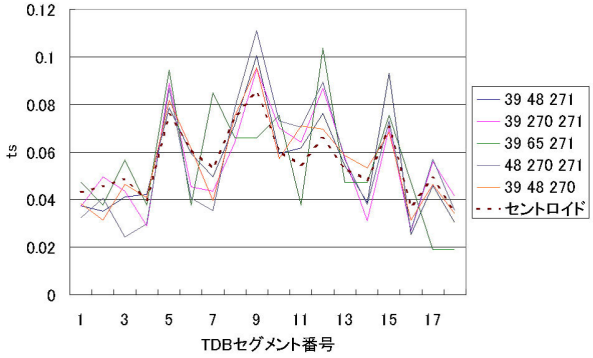

図 13　クラスタ ID17 番から抽出された隠れアイテムセットとセントロイドの比較

小さいことが確認される．すなわち，抽出された隠れアイテムセットは，頻出アイテムセット集合のある要素をサブセットとして含み，かつ最小サポート値よりも低いサポート値を持ち，かつトランザクションデータベース中における出現傾向が同じアイテムセットであることが確認される．

5.3　パラメータ設定

提案手法には，パラメータとしてウィンドウサイズ w とクラスタ数 L の 2 つが存在する．w と L の設定の違いによる結果の際について考察する．

a)　ウィンドウサイズ w

ウィンドウサイズ w が大きければ大きいほど，ひとつのセグメント化した TDB に存在するトランザクションの時間差が長くなる．そして w が小さければ小さいほど，ひとつのセグメント化した TDB に存在するトランザクションの時間差が短くなる．これは，w が大きければ大きいほど，抽出された隠れアイテムセットが長期的な視点に即した知識を表現するものであり，w が小さければ小さいほど，より短期的な視点に即した知識を表現するものになる．

b)　クラスタ数 L

クラスタ数 L が大きければ大きいほど，一つのクラスタに分類されたアイテムセットの時系列化したサポート値の分散が大きくなり，L が小さければ小さいほど，一つのクラスタに分類されたアイテムセットの時系列化したサポート値の分散が小さくなる．これは，L が大きければ大きいほど,「出現タイミングが似ている」という制約が強くなることを意味し，L が小さければ小さいほど,「出現タイミングが似ている」という制約が弱くなることを意味する．

5.4　既存手法との比較

従来，膨大な数の頻出アイテムセットを減らすために，極大頻出アイテムセットのみを抽出する手法 [8] や，飽和頻出アイテムセットのみを抽出する手法 [11] が提案されている．

ここで，アイテムセット X が頻出極大アイテムセットであるということは,「アイテムセット X のサポート値が最小サポート値以上であり，かつ X のスーパーセットである任意の X' が，最小サポート値未満のサポート値である」ことである．また，アイテムセット X が頻出飽和アイテムセットであるということは,「アイテムセット X のサポート値が最小サポート値以上であり，かつ X と同一のトランザクション上にある X の全てののスーパーセット X' が，最小サポート値未満のサポート値である」ことである．

両手法ともに，アイテムセット抽出高速化のために，さまざまなアルゴリズムが提案されている．極大頻出アイテムセット抽出手法では，Max-Miner [8]，FPmax [12] などが，飽和頻出アイテムセット抽出手法では，CLOSET [11],FP-close [13] などが提案されている．

ここで，本稿で定義した「隠れアイテムセット」をユーザが発見するために，ユーザが把握しなければならないアイテムセット数の比較を行うために，図 14 に，Retail データセット [10] を対象にして，頻出アイテムセット抽出手法 (All)，極大頻出アイテムセット手法 (Maximal)，飽和頻出アイテムセット手法 (Closed) を適用したときの，最小サポート値と抽出アイテムセット数の関係を示す．

図 14 に示すように，ユーザがこれら 3 手法によって本稿の提

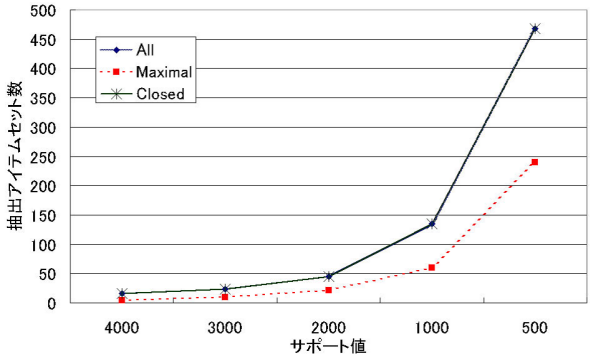

図 14　最小サポート値とアイテムセット数の関係

案手法で抽出した隠れアイテムセットを発見する場合には，膨大な抽出アイテムセット数から見つけなければならない．たとえば，クラスタ ID7 番から抽出された隠れアイテムセット {32 38 39 41 48}(448) を発見するためには，最小サポート値を 448 以下にして実行しなければならない．頻出アイテムセット抽出手法，飽和アイテムセット抽出手法では 470 以上の抽出アイテムセットから，極大頻出アイテムセット抽出手法では，250 以上の抽出アイテムセットの中の 1 つのアイテムセットである．

以上より，提案手法を利用することによって，本稿で定義した「隠れアイテムセット」を効率よくユーザは発見することができる．我々は，隠れアイテムセットは，新しい知識発見につながる可能性の高いアイテムの一部であると考えるため，提案手法はユーザにとって有効な手法の一つであることがいえる．

6. おわりに

本稿では，任意の最小サポート値によって抽出された頻出アイテムセットを自動的にクラスタリングすることによりアイテムセットをまとめ上げる手法を提案した．ユーザは，まとめ挙げられたアイテムセットを見ることで，より簡単に頻出アイテムセット集合を把握することができることを確認した．

さらに各クラスタに属する頻出アイテムセット集合に付随する隠れアイテムセットを抽出することにより，従来手法では，把握することが困難であったアイテムセット集合を抽出することができる手法を提案した．ユーザは，サポート値が低く，新しい知識発見につながる可能性の高いアイテムセットを従来手法に比べてより簡単に発見することが可能であることを確認した．

6.1　今後の課題

c)　TDB のセグメント化手法

本稿で提案したアイテムセットのクラスタリング手法は，TDB をセグメント化することにより任意のアイテムセットのサポート値を時系列化することでアイテムセットのクラスタリングを行う．セグメント化した TDB は，ウィンドウサイズ w を指定することにより，同一トランザクション数で構成されている．しかし，TDB にはトランザクションの発生の偏りが存在するため，セグメント化した TDB のトランザクションの疎密の違いが生じる．このような疎密の違いは，本稿で提案した隠れアイテムセットの定義のひとつである「出現タイミングが似ている」という制約を緩める要因になると考える．

TDB にトランザクションの発生時刻も記録されている場合，この問題を以下のように TDB のセグメント化手法を変えることで解決が出来ると考える．TDB をウィンドウサイズ w でセグメント化するのではなく，ある時間間隔 tw を与え，tw 間に発生するトランザクションをひとつのセグメント化 TDB としてセグメント化する．tw でセグメント化することによって，セグメント化 TDB のトランザクション数が変わってしまうが，トランザクションの疎密の問題を解決できると考える．今後の課題のひとつとして，時間間隔 tw によってセグメント化する手法の評価を行っていきたい．

d)　Stream Data Mining 手法への応用

本稿で提案した手法は，Stream Data Mining 手法に応用することができる．したがって，今後は本稿で提案した手法を Stream Data Mining 手法に対応させていくことを課題としたい．

謝　　辞

本研究の一部は，文科省 21 世紀 COE「プロダクティブ ICT アカデミア」及び科学技術振興費「e-Society」プロジェクトによるものである．

文　　献

[1] R,Agrawal, T.Imielinski, and A.N.Swami, ”Mining association rules between sets of item in large databases,” Proc. of Int. Conf. on ACM SIGMOD, pp.207-216, (1993).

[2] R.Agrawal, R.Srikant. “ Fast algorithms for mining association rules ”. In VLDB ’94, pp. 487-499, Santiago, Chile, Sept. 1994.

[3] J.S. Park, M.Chen, P.S. Yu, ”An effective hash-based algorithms for mining association rules,” Proc. of Int. Conf. on ACM SIGMOD, pp.175-186, (1996).

[4] A. Savasere, E. Omiecinski, and S. Navathe, ”An Efficient Algorithm for Mining Association Rules in Large Databases,” Proc. of Int. Conf. on VLDB, pp.423-444, (1995).

[5] J. Han, J. Pei, and P.S. Yu, ”Mining frequent Patterns without Candidate Generation,” Proc. of Int. Conf. on ACM SIGMOD, pp.1-12, (2000).

[6] J. Pei, J. Han, H. Lu, S. Nishio, S. Tang, and D. Yang, ”H-Mine: Hyper-Structure Mining of Frequent Patterns in Large Databases”, Proc. of Int. Conf. on IEEE ICDM ,pp.441-448, (2001).

[7] J. Liu, Ke Wang, and J. Han, ”Mining Frequent Item Sets by Opportunistic Projection,” Proc. of Int. Conf. on ACM SIGKDD, (2002).

[8] Bayard, R.J. ”Efficiently mining long patterns from databases”, In Proceedings of the ACM SIGMOD Conference on Management of Data, pp. 85-93, 1998.

[9] Jain, A. and Dubes, R. ”Algorithms for Clustering Data,” Prentice Hall, Englewood Cliffs, NJ, 1988

[10] Frequent Itemset Mining Implementations (FIMI) Repository, ”http://fimi.cs.helsinki.fi/”

[11] J.Pei, J.Han, and R.Mao, ”CLOSET: An efficient algorithm for mining frequent closed itemsets,” In DMKD’00, 2000.

[12] G.Grahne and J.Zhu, ”High performance mining of maximal frequent itemsets,” In SIAM’03 Workshop on High Performance Data Mining, May 2003.

[13] G. Grahne and J. Zhu, ”Efficiently Using Prefix-trees in Mining Frequent Itemsets,” Proc. of Int. Conf. on IEEE ICDM Workshop on Frequent Itemset Mining Inplementations, 2003.