■S3 群（脳・知能・人間）－ 3 編（人工知能と学習）

# 4 章 知識マイニング・発見

（章主任：鷲尾　隆）［2009 年 4 月 受領］

## ■概要■

本章では，データマイニングとその技術的立場から知識発見に関して説明する．計算機科学における知識発見は，データや規則から計算機を用いて人間にとって何らかの意味のある知識を発見する技術全般を指す．実用上の観点を除けば，技術的には発見された知識が人間にとって全く新しいものであるか，既知のものであるかは問わない．知識発見は，このような機能を持つ技術の総称であり，特定技術を指すわけではない．他章でも説明されているように，機械学習や計算機推論，統計的手法，情報検索，自然言語処理，データマイニングなど，種々の技術を用いて知識発見は実現される．このように知識発見は広範な技術を包含し，逆に言えばデータマイニングはその中の一つの技術分野であると言える．

では，知識発見技術の一つであるデータマイニングは，ほかの知識発見技術とはどのような関係にあるのであろうか．データマイニングは，大量データの中から人間にとって意味のある知識を発見する技術であると言われるが，それだけならば統計的手法や情報検索技術も同様である．しかし，統計的手法は，データに含まれる確率変数などの属性が，データ全体にわたって示す分布を問題にし，その分布の中からデータ全体ないしは局所的な傾向を把握しようとするアプローチである．情報検索技術は，逆にデータを構成する事例の中から，指定された条件に適合する個別事例を収集，優先順位付けするアプローチである．これらに対して，データマイニングは，確率分布などに限らず，何からの指標やパターンに基づいてデータを分割し，各分割データにおいて特徴的な傾向やパターンを探索し，データに埋もれた局所的かつ特徴的傾向を把握するアプローチである．

一方，機械学習の中には，決定木をはじめとして上記のデータマイニングの定義に合致する手法が多く存在する．したがって，データマイニングと機械学習を技術的に明確に区分することはできない．また，特にテキストデータを対象とする知識発見技術は，テキストマイニングと呼ばれるが，テキストデータが単語や文法，それらの背景に横たわる意味など，一般のデータに比して複雑かつ特徴的な情報を有するため，自然言語処理技術やほかの知識発見技術を取り入れつつ，独自の発達を遂げている．そのために，当初，テキストマイニングはデータマイニングの一部と考える研究者が多かったが，近年では両者を区別して扱うことも多くなった．しかし，技術的にはなおも多くの共通性を有している．

近年では統計的手法や情報検索，データマイニング，機械学習，自然言語処理の技術的境界に多くの新たな技術が生まれており，これらを技術的に明確に区分することはますます難しくなってきている．本章では，以上のように非常に広範かつ複雑な知識発見，データマイニング及び他技術間の関係，重なりのなかで，特にデータマイニング固有の考え方や枠組み，技術について取り上げ，説明する．また，独自の発展を遂げつつもデータマイニング技術を構成する大きな分野の一つと考えられるテキストマイニングについても説明する．

ほかの技術分野と異なり，データマイニング研究の歴史において特徴的なのは，当初から人間が計算機と共同して知識発見を行うプロセスの体系化を意図したことである．データマイニ

ング研究では，計算機が単独で自動的にデータから知識発見を行うことを目指すのではなく，そのユーザである人間と計算機が相互的に，かつ反復的に作用し合って，データから知識発見を行うという前提に立つ．そして，そのプロセスを念頭に置きながら，種々の技術を研究開発していく立場をとる．このプロセスは知識発見プロセスと呼ばれ，最初の節でこの概要について述べる．

更に，上記のような相互的，反復的知識発見プロセスの効率的実現を可能にするための計算機環境が重視され，そのためのデータマイニングプラットフォームの研究もなされている．プラットフォームとして多くの商用ツールが民間企業によって開発されているが，それ以外に研究者のテストプラットフォームとしてフリーでオープンソースのツールも普及している．その中でも，Weka というツールは最も有名であり，研究用のみならず，ある程度までのデータ規模とオフライン使用の範囲で，十分な実用性を備えている．2 番目の節では，このようなデータマイニングプラットフォームに求められる機能やデータ仕様，Weka の概要について述べる．

一方，データマイニング固有の代表的技術として，パターン列挙とその数え上げによる知識発見方法が挙げられる．これは，データ内の事例に含まれるための性質や規則を満たすパターンを探索的に列挙し，実際にそれがいずれの事例に含まれるかを数え上げることで，各事例の特徴やそれに含まれるパターンに関する知識を得る方法である．データマイニング研究の初期から研究されている方法論であり，その最も基本的な技術としてバスケット分析がある．これは，例えばコンビニエンスストアに訪れる顧客がいずれの品物の組合せをよく購入するかなど，イベントやアイテムの頻出共起パターンをデータから発掘するものである．3 番目の節では，この技術について概説する．

更に複雑なパターン列挙とその数え上げを行うデータマイニング固有の知識発見方法として，列挙による構造データマイニング技術が挙げられる．これは，系列や木，グラフなどの構造を持つ多数の事例データから，頻出する部分系列，部分木，部分グラフなどを発掘するものである．発掘対象とするパターンのカテゴリーによって，系列マイニング，木構造マイニング，グラフマイニングなどと呼ばれる．4 番目の節では，これらの技術について一般的に概説する．

最後の節では，データマイニング技術を構成する大きな分野の一つであり，かつ更に独自の発展を遂げつつあるテキストマイニング技術について概説する．テキストマイニング技術とデータマイニング技術の関連，データマイニング技術を用いた自然言語処理，大量文書から書き手にとってすら未知の知識を発見する可能性など，多面的な角度からテキストマイニング技術の内容とその可能性について述べる．

**【本章の構成】**

本章では以下について解説する．

4-1 知識発見プロセス
4-2 データマイニングプラットフォーム
4-3 バスケット分析
4-4 構造データマイニング
4-5 テキストマイニング

■S3 群－3 編－4 章

# 4-1 知識発見プロセス

（執筆者：平野章二）［2009 年 1 月 受領］

知識発見プロセスとは，蓄積された大量のデータから潜在的に有用と考えられる未知の知識（パターン）を抽出するための包括的な手続きである．データマイニングと同義として混用される場合も多いが，Fayyad らはデータマイニング（Data Mining）とデータベースからの知識発見（Knowledge Discovery in Databases：KDD）を明確に区別しており，前者は後者の一部であり，マイニング段階に相当すると位置づけている [1]．

1996 年に Fayyad が 9 段階から成る知識発見プロセスモデルを提唱して以来，Anand and Buchner の 8 段階モデル [2]，工業的な実用性を重視した Shearer らの CRISP-DM [3] など，様々なプロセスモデルが提案されてきた．各モデルは作業段階の数やその内容において差異が見られるが，いずれも最初の問題理解/目標設定から知識の整理/利用に至る作業を一連の段階的なプロセスと捉えること，また，各段階で必要に応じて人間（解析者，専門家など）が介入して内容を吟味しその結果を前段階にもフィードバックさせること，プロセスを全体的にも部分的にもループさせ繰り返し実行することが共通している．すなわち，マイニングアルゴリズムの盲目的（Blind）な適用は誤った結果を導く可能性があり，全体をプロセスとして捉え，そのなかで人間とデータが相互的（Interactive）に，かつ反復的（Iterative）に作用し合うことが有用な知識を効率的に発見するために不可欠であるとの認識に基づいている．各モデルの比較は文献 4）に詳しい．

以下では，後に広く影響を与えた Fayyad の知識発見プロセス [1] について説明する．このプロセスモデルは次の 9 段階から構成される．フローの概念図は**図 1･1** に示すとおりである．

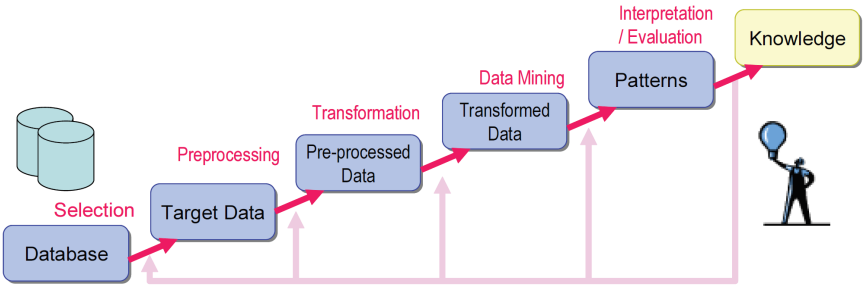



**図 1・1**　Fayyad による知識発見プロセスの概念図

1. **適用領域の理解と目標設定**：領域に関連する背景知識，ユーザの希望する到達目標などについて理解形成を行う．
2. **データセットの生成**：分析対象とするデータセットを定めるか，属性を選択してデータベースからデータ集合を抽出する．
3. **データ洗浄及び前処理**：基礎的分析を通じてノイズ，外れ値，欠損値，表記揺れなど分析上問題となる要素がデータセットに混入しているか確認し，混入している場合はその種類に応じて当該データもしくは属性の除外，事象のモデル化，代入，修正などの対応

を行う．

4. **データ削減と射影**：次元削減もしくは変換によって属性数を適度な数まで減少させるか，データの不変量による表現を探る．
5. **マイニングタスクの選択**：到達目標に鑑みて適切なマイニングのタスク（分類，クラスタ化，回帰，相関など）を定める．
6. **マイニングアルゴリズムの選択**：タスクとデータの性質（質的/量的，時系列，決定属性の有無など），投入可能なリソース（計算機，処理時間など）などに応じて適切なマイニングアルゴリズムと必要なパラメータを決定する．
7. **マイニングの実施**：選択したアルゴリズムを対象データに適用し，分類ルール，決定木，クラスタ，回帰モデル，相関ルールなどの形式で知識（パターン）を生成する．
8. **結果の解釈と評価**：マイニングアルゴリズムにより生成されるパターンをユーザが解釈しその内容について評価を行う．必要に応じて 1. ～7. の任意の段階を繰り返す．
9. **知識の整理**：獲得した知識のシステムへの実装，ドキュメント化，既存知識との整合性確認を行う．

上記のようなプロセスが一方向的でなく双方的に，かつユーザの評価を組み込んで繰り返し実行されることが特徴的である．例えば，到達目標については，結果の提示を受けたユーザがデータに対する理解を深化させ，初期の形からより具体化/展開していくことが現実的にしばしは発生するほか[3]，対象データや属性の拡充，前処理方法の変更なども比較的多くのケースで必要となる．

**■参考文献**


1) U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy：“From Data Mining to Knowledge Discovery: An overview,”In U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy(eds)：“Advances in Knowledge Discovery and Data Mining,”AAAI Press/MIT Press, 1-36, 1996.
2) S.S. Anand and A.G. Buchner：“Decision Support Using Data Mining,”Financial Times Pitman Publishers, 1998.
3) C. Shearer：“The CRISP-DM model: the New Blueprint for Data Mining,”Journal of Data Warehousing 5(4):13-22, 2000.
4) L.A. Kurgan and P. Musilek：“A survey of Knowledge Discovery and Data Mining process models,”The Knowledge Engineering Review, 21(1):1-24, 2006.
5) R.J. Brachman and T. Anand：“The Process of Knowledge Discovery in Databases: A Human-Centered Approach,”In U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy(eds)：“Advances in Knowledge Discovery and Data Mining,”AAAI Press/MIT Press 37-59, 1996.

■S3 群－3 編－4 章

# 4-2 データマイニングプラットフォーム

（執筆者：阿部秀尚）［2016 年 7 月 受領］

データマイニングプロセスの処理を実行するには，データマイニングツールと呼ばれるソフトウェアが必須である．データマイニングツールは蓄積されたデータを加工する処理・加工したデータの分析・得られた規則性（パターン）の検証を実行する環境である．ただし，組織としてのデータマイニングプラットフォームを考えるとき，データマイニングツールを中心とするコンピュータシステムに加えて人的な組織編成についても十分考慮する必要がある [2),3)]．以下では，データマイニングの各処理を統合的に実行する環境であるデータマイニングツールについて述べる．

## 4-2-1 データマイニングツールの定義

データマイニング（あるいは知識発見）プロセスはデータベース技術・解析技術・情報視覚化技術などを所定の目的のもとに統合した過程である．データマイニングツールは，これらの技術のうち複数の処理が「入力データや目的に応じて選択可能」であり，利用者による試行錯誤を伴う実行が可能なインタフェースを備えた統合環境を指す [1)*1]．狭義には，（主に表形式に加工した）データの解析技術について，統計解析手法や機械学習手法などのうち複数の手法が選択的に適用可能であるソフトウェアを指すこともある．データマイニングツールとして求められる機能は以下の通りである．

**データの前処理機能**：データ分析の前の段階として，データの加工処理ができること．

**データ分析機能**：複数のデータ分析手法が用意されていること．

**結果の後処理機能**：得られたパターンについて，任意のデータでの検証機能を有すること．

## 4-2-2 データマイニングツールの利用

データマイニングツールは，上述のような機能が必要なため，大規模なソフトウェアとなる．このため，高額な商用ツールが販売されている一方，定評のある機械学習ライブラリ Weka [4)] や統計解析ライブラリ R [5)] を取り入れることが可能な KNIME [6)]，RapidMiner [7)] といった非商用ツールが開発されている．利用者は，入力データの規模に応じて，多くの種類の分析手法，分かりやすい視覚化やシミュレーション結果を，データマイニングツールを利用することで得ることができる．

企業においては，データマイニングツールの導入にあたって多大な初期コストの投下は一般的に困難であることが考えられ，非商用のツールによる小規模なデータマイニングから規模を広げていくことが必要となる．その際，データ分析結果の検証については，十分に検討して一般化を行う必要がある [8)]．このような，小規模の活動から組織内の意思決定支援にデータ分析結果を取り込んでいくことにより，より大規模なクラスタ計算機と並列機械学習ライブラリーを利用したようなデータマイニング手法の活用 [9)] が可能になる．

統合的なデータマイニングツールで作成されたデータマイニングモデルは，PMML によって

*1 統合環境を伴ったデータマインングツールを KDnuggets では，“Suites”と呼んでいる．

標準化された入出力データを用いることで，OLAP ツールや情報検索，視覚化ツールもデータの前処理機能や結果の検証機能を実現することが可能な環境が整いつつある．

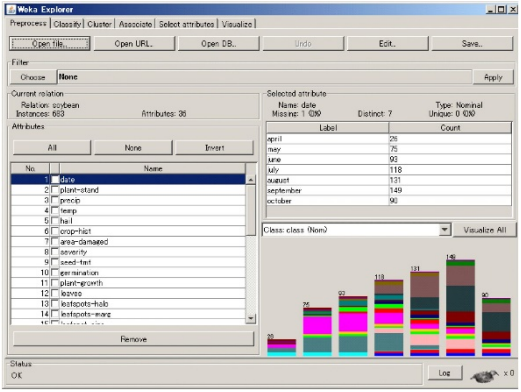
各種データマイニング処理に関係するライブラリーの対応状況はまちまちであるが，交換データの標準化が進めば，ツール間でのデータやモデルの交換，検証や視覚化の高度化，実システムへの適用といった各段階でのデータ中心の経営活動の支援が可能になると考えられる．

### 4-2-3 データマイニングツールの例

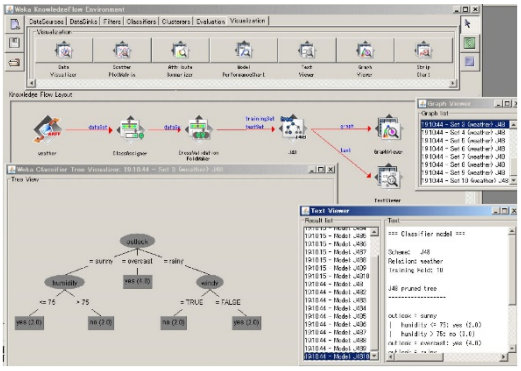
データマイニングツールに備えられた各機能の例として，Weka [4] を取り上げる．Weka に含まれる各機能は以下の通りである．

・**データの前処理機能**

属性フィルタ：離散化，二値化，欠損値処理，ダミー属性付加など
インスタンスフィルタ：サンプリング，条件付き除去など
属性選択：フィルタアプローチ，ラッパアプローチによる属性選択法




(a) Knowledge Explorer




(b) Knowledge Flow

**図 2・1**　Weka の GUI 画面

**・データ分析機能**

分類学習：確率モデル，ルール学習，決定木，Support Vector Machine，ニューラルネットワーク，ロジスティック回帰など

数値予測：線形回帰，回帰木，ニューラルネットワーク

アンサンブル学習スキーム：Boosting，Bagging，Stacking など複数の学習モデルやアルゴリズムを組み合わせることにより，分類学習や数値予測の精度を向上させる

クラスタリング：K-Means，EM アルゴリズムによる混合分布モデルによるクラスタリング，DBSCAN など

相関ルール：Apriori

**・結果の検証機能**

属性間の 2 次元プロット

マイニング結果（データ予測結果，モデル）の視覚化.

Weka の実行例を**図 2･1** に示す．“Knowledge Explorer（エクスプローラ）”は対話的に処理を適用するために用い，“Knowledge Flow（ナレッジフロー）”はデータの加工から分析，結果の取得までの流れを視覚的に表して各処理を適用する．

**■参考文献**


1) KDnuggets：http://www.kdnuggets.com/software/index.html
2) CRISP-DM：http://www.crisp-dm.org/
3) 酒井麻衣子：“小売業におけるデータマイニングの取り組み,” 人工知能学会誌, vol.17, no.5, pp.580-587, 2002.
4) Ian H. Witten and Eibe Frank：“Data Mining: Practical machine learning tools and techniques 2nd Edition,” Morgan Kaufmann, San Francisco, 2005.
5) The R Project for Statistical Computing：http://www.r-project.org/
6) KNIME：http://www.knime.org/
7) Rapid Miner：http://rapidminer.com/
8) 戸谷圭子:“金融サービス･マーケティングにおけるデータマイニングの諸問題,”人工知能学会誌, vol.19, no.5, pp.607-609, 2004.
9) DeNA：“Mobage を支える技術 ～ソーシャルゲームの舞台裏～,” 技術評論社, 2012.

■S3 群－3 編－4 章

# 4-3 バスケット分析

（執筆者：宇野毅明）［2008 年 11 月 受領］

バスケット分析とは，小売店での購買データなどに見られる組合せ的なデータの分析，及びその手法のことをいう．小売店に訪れた購買者それぞれが購入した品目が記録されたデータから，規則性，あるいは顕著な特徴を見つけ出すことによってマーケッティングを行う，といったものである．データとしては，1 つのレコードが 1 人の顧客の 1 回の購買行動に対応し，レコード 1 つにはその購買行動時に購入した商品の組合せが記録されているものを考える．このようなデータはトランザクションデータとも呼ばれ，各レコードはトランザクション（Transaction）とも呼ばれる．また，品目はアイテムと呼ばれ，商品の集合はアイテム集合とも呼ばれる．見つける特徴や規則は品目の組合せに基づくもの，例えば A と B と C は同時に購入されることが多い，A と B を買った人は B を買う可能性が高い，といったものである．つまり，各レコードが何らかのアイテムの部分集合となっているデータベースから，組合せ的な知識を発見する分析手法がバスケット分析である．

**図 3･1** で例を見てみよう．データ A では，商品 6 と 7 は一緒に買われる（共起性が高い）ことが多い．また，1 と 6 も，2 回一緒に買われている，という点では同様である．また，6,7 を購入した場合，1 を購入することが多いことも分かる．また，データ A では 6,7 という組合せが多いが，B には存在しない．このような，ある程度の割合の客，つまりレコードに共通して見られる商品（アイテム）の組合せの特徴を調べることが，バスケット分析なのである．

| データ A | データ B |
|---|---|
| 客 1：1, 4, 6, 7 | 客 A：2, 3, 6 |
| 客 2：1, 3, 6, 7 | 客 B：2, 3, 7 |
| 客 3：3, 4, 5 | 客 C：1, 4 |
| 客 4：1, 3, 5 | 客 D：1, 5 |
| 客 5：1, 3, 4 | 客 E：1, 2, 3, 4, 6 |

**図 3・1**　購買データの例

バスケット分析に用いるデータは購買データにとどまらない．1 ユーザの Web ページ閲覧記録は，そのユーザが閲覧したページの集合であるし，アンケート集計データの各レコードは，各記入者がチェックを付けた欄の集合となっている．共通して閲覧されることが多い Web ページ群や，関連する意見や感想のグループを見つけることが可能となる．数値を書き込む項目がある場合も，書き込まれた数値をいくつかの範囲に分類する（例えば 1 から 10 と，11 から 20 のように）ことで，チェックを付けた範囲の集合であるとみなせる．同様に，遺伝子の発現データのような各項目が実数値の値をとるデータであっても，トランザクションデータに変換が可能であり，特定の疾病を持つ患者に多く見られる遺伝子変異の組合せを見つけることができる．文章を単語の集合として見る，映像や画像をそれらが持つ特徴の組合せとして見る，XML のような構造データをそれらが含む部分構造の集合として見る，といったデータの変換により，多くのデータをトランザクションデータとしてみなすことができるため，バスケット分析の適

用範囲は広い.

バスケット分析はすべてのレコードに共通して現れる特徴を見つけるものではなく，一部のレコードが持つ特徴や規則性を見つける．これは機械学習やクラスタリングでの規則発見や分類器の生成問題とは異なるアプローチである．そのため，なんらかの意味で評価値の高い解を 1 つ見つける最適化的なアプローチはあてはまりが悪く，通常は列挙的なアプローチが用いられる．つまり，規則や特徴が満たすべき条件を定め，その条件を満たす解をすべて見つけるというものである．このようなアプローチはマイニングとも呼ばれる．

データから見つける特徴や規則としては以下のものがあげられる．

- **頻出集合**（Frequent Itemset）：多くのレコードに含まれるアイテム集合であり，単にルールとも呼ばれる．閾値を設定し，閾値以上のレコードに含まれるアイテム集合（例えばパンと牛乳とチーズのようなアイテムの組合せ）が頻出集合と呼ばれる．また，アイテム集合 X を含むレコードを X の出現（Occurrence），出現の数を X の頻出度（Frequency）と呼ぶ. 頻出度が閾値以上のアイテム集合が頻出集合である. 頻出度は支持度（Support）と呼ばれることもあり，X を含むレコード数をレコードの総数で割った値で定義されることもある．
- **アソシエーションルール**（Association Rule）：アイテム集合 X を含むレコードは，アイテム a（あるいはアイテム集合 Y）を含むことが多い，という形のルールである．このとき，X の頻出度を X の支持度（Support）と呼び，X を含むレコードの中で a（あるいは Y）を含むレコードの割合を信頼度（Confidence）と呼ぶ．支持度が高く，信頼度が高いアソシエーションルールが重要なルールとなる．
- **分類パターン**： 2 つのトランザクションデータ A と B があるとき，A の多くのレコードに含まれ B の少数のレコードにしか含まれないアイテム集合である. A での頻出度が閾値以上であり，かつ B での頻出度が閾値以下であるもの，あるいは A での頻出度を B での頻出度で割ったものが閾値以上であるものとして定義できる．

これらの逆，つまり頻出度の低いアイテム集合や，信頼度が低いルールを見つける問題もあるが，アイテム a を含まないレコードにアイテム a' を追加する，というデータの変換を行うことにより，頻出度・信頼度の高いアイテム集合，ルールを見つける問題に帰着できる．ある頻出集合を含むトランザクションの集合は，共通する特徴を持つものと考えられ，グループ，クラスタとみなすことができる．そのため，共通するリンク先を持つ Web ページ群を見つけることで Web コミュニティーを見つける研究も行われている．また，データのアイテムとトランザクションを逆転させたデータで大きさ 2 の頻出集合を見つけると，閾値以上の大きさの共通部分を持つトランザクションの組をすべて見つけることができる．

分類パターンを見つける問題は通常 NP 完全となるため，多項式時間性を持つアルゴリズムの解法を構築することは難しい．しかし，頻出集合やアソシエーションルールは，頻出度（支持度）・信頼度に関して単調性が成り立つ（頻出集合の部分集合は頻出集合となる）ため，発見は容易である．しかし，容易に発見できるものは，空集合や，大きさの小さいアイテム集合であるため，人間にとっては自明であり役に立たないものであることが多い．そのため，ある程度の大きさを持つ頻出集合，アソシエーションルールの発見が重要となるが，閾値以上の大きさを持つ頻出集合・アソシエーションルールの発見問題は NP 完全となる．そのため，頻出集合を列挙するアルゴリズムに改良を加えることで，大きな頻出集合や分類パターンを見つける

アプローチがとられる．アソシエーションルールについても，支持度の大きなアイテム集合を元にして見つけることが多い．

頻出集合の列挙は，頻出集合の単調性を利用し，空集合から出発して山登り的な探索で行われることが多い．この際，大きさ 1 の頻出集合をすべて見つけ，それらにアイテムを追加して大きさ 2 の頻出集合をすべて見つけ，と大きさごとに幅優先的に見つける方法と，アイテムを逐次的に追加し，頻出集合でなくなったらバックトラック（最後に追加したアイテムを除去すること）して異なるアイテムを追加し，探索を続ける，という方法がある．前者はアプリオリ（apriori）[1]，後者は深さ優先探索，あるいはバックトラックと呼ばれる [2]．歴史的にはまず apriori が開発され，その後深さ優先探索が開発された．一般に深さ優先探索のほうがメモリ効率が良いが，データをすべてメモリに読み込むことができない場合，apriori は外部メモリの参照回数が少ないという利点がある．

これらの列挙法により，頻出集合は安定した計算時間，つまり解の数に対して線形の時間で見つけることができる．ただし，素直にプログラミングすると頻出度の計算を行う際にデータ全体にアクセスすることとなり，解 1 つ当たりの計算時間は入力データの大きさに比例することとなる．そのため，絞り込み検索を用いた頻出度の計算，再帰的なデータ圧縮などを用いることで頻出度計算の高速化が行われており，現在では頻出集合 1 つ当たりほぼ定数時間，1 秒間に 10 万個程度の解を列挙できる（文献 3）を参照）．

大きさが小さくない非自明な頻出集合を見つけるには，閾値を小さく設定して多数の頻出集合を見つける．すると次に，多量の解をどのように処理するかという問題が発生する．この問題の解決法として，意味的な損失をなるべくせずに頻出集合の中から代表的なものだけを取り出すモデルが考えられている．その一つは極大な頻出集合，もう一つは飽和集合という，頻出度が同じ頻出集合の中で極大なものである．極大頻出集合は数がより少ないこと，単調族の極大元を列挙する手法で効率的に列挙できること [4] が利点である（ただし多項式時間の保証はない．出力数に対する多項式時間アルゴリズムの存在は未解決問題である）．飽和集合は，極大クリーク列挙に用いられる逆探索法という手法を使うことで，多項式時間でかつ実用的にも高速に列挙できること [5]，どの頻出集合に対しても，その頻出集合と出現の集合が同じ飽和集合が存在するため，出現集合の意味で損失がないことが利点である．これは分類パターンやアソシエーションルールを見つける際には，ある意味で完全性を保証していることになる．

頻出集合発見は，2003 年，2004 年にプログラミングコンテスト [3]，が行われたこともあり，比較的プログラムが手に入りやすい．文献 3）のサイトで多くのプログラムと論文が入手可能である．

**■参考文献**


1) R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, A.I. Verkamo：“Fast Discovery of Association Rules,”In “Advances in Knowledge Discovery and Data Mining,”MIT Press, pp.307-328, 1996.
2) R.J. Bayardo Jr.：“Efficiently Mining Long Patterns from Databases,”Proc. SIGMOD'98, pp.85-93, 1998.
3) B. Goethals：“the FIMI repository,”http://fimi.cs.helsinki.fi/, 2003.
4) J. Han, J. Pei, and Y. Yin：“Mining Frequent Patterns without Candidate Generation,”SIGMOD Conference 2000, pp.1-12, 2000.
5) T. Uno, T. Asai, Y. Uchida, and H. Arimura：“An Efficient Algorithm for Enumerating Closed Patterns,”in “Transaction Databases,”Lecture Notes in Artificial Intelligence 3245, pp.16-31, 2004.

■S3 群－3 編－4 章

# 4-4 構造データマイニング

（執筆者：猪口明博）［2008 年 11 月 受領］

本節では，グラフや木などの構造を持つデータを対象としたデータマイニング手法を紹介する．構造データマイニングはグラフマイニングとも呼ばれ，グラフの集合が与えられたとき，それらのグラフに頻繁に出現する部分グラフを列挙するデータマイニング手法である．例えば，**図 4･1** に示す 4 つのラベル付きグラフが与えられたとき，この手法はそれらのグラフの多くに含まれる部分グラフを出力する手法であり，網がけされた部分グラフなどが出力される．本節では，グラフ構造を定義し，グラフマイニングの問題を定式化する．更に出力される部分グラフの探索原理の説明し，代表的なアルゴリズムを紹介する．

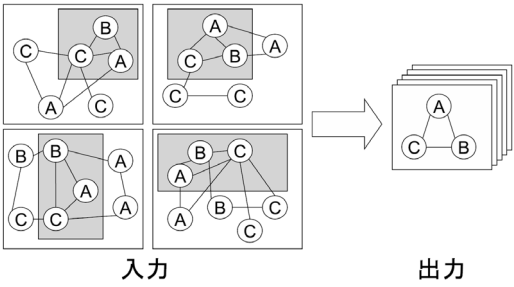



**図 4・1** グラフマイニングの例

## 4-4-1 ラベル付きグラフ

ラベル付きグラフ $g(V, E, L, l)$ は，頂点の集合 $V$，頂点ペアを結ぶ辺の集合 $E \subseteq V \times V$，頂点と辺のラベルの集合 $L$，頂点や辺にラベルを割り付ける関数 $l : V \cup E \rightarrow L$ で表される．2 つのグラフ $g(V, E, L, l)$ と $g'(V', E', L', l')$ が与えられ，$\forall v, v_1, v_2 \in V'$に対して，以下を満たす単射の写像 $\phi : V' \rightarrow V$ が存在するとき，$g'$ を $g$ の部分グラフと呼び，$g' \subseteq g$ と表す．

1. $\{\phi(v_1), \phi(v_2)\} \in E$ if $\{v_1, v_2\} \in E'$
2. $l'(v) = l(\phi(v))$
3. $l'(\{v_1, v_2\}) = l(\{\phi(v_1), \phi(v_2)\})$

グラフ $g$ の辺 $\{v_1, v_2\} \in E$ が順序を持つペアであるとき $g$ を有向グラフと呼び，向きの異なる辺を表すために頂点 $v_1$ と $v_2$ の間の辺 $(v_1, v_2)$ と $(v_2, v_1)$ を区別して書く．一方，グラフ $g$ の辺 $\{v_1, v_2\} \in E$ が順序を持たないとき $g$ を無向グラフと呼び，**図 4･2**(a)にその例を示す．頂点 $v_1$ から $v_2$ に至る辺の系列をパスという．連結グラフとは，辺の向きを除いたグラフの任意の頂点間にパスが存在するグラフである．始点と終点が同じ頂点であるパスを閉路と呼び，閉路を含まない有向グラフを非巡回有向グラフと呼び，図 4･2(b)にその例を示す．

閉路を含まない連結無向グラフを根無し木，あるいは自由木と呼び，図 4･2(c)にその例を示す．一方，以下を満たす非巡回有向グラフを根付き木と呼び，図 4･2(d)にその例を示す．

1. 連結グラフである．
2. 頂点 $r \in V$ からすべての頂点 $v \in \backslash\{r\}$に至るパスが存在する．

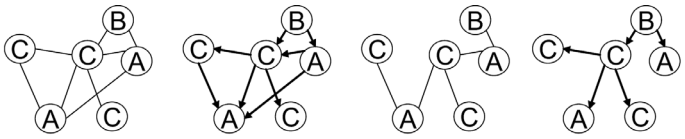



**図 4・2**　ラベル付きグラフ

根付き木において，頂点 $r \in V$ を根と呼ぶ．根 $r$ と頂点 $v$ を結ぶパスが辺 $(v', v'')$ を含むとき，頂点 $v$ を頂点 $v'$ の子孫，$v'$ を $v$ の祖先と呼ぶ．頂点 $v$ が $v'$ の子孫であり，$v$ と $v'$ の間に辺 $(v', v)$ が存在するとき，$v$ を $v'$ の子，$v'$ を $v$ の親と呼ぶ．頂点 $v$ と $v'$ が共通の親を持つとき，$v$ と $v'$ を兄弟と呼ぶ．根付き木 $g$ の兄弟である頂点間に順序関係があるとき $g$ を順序木と呼び，それらの頂点間に順序関係がないとき $g$ を無順序木と呼ぶ．2 つの無順序木 $g(V, E, L, l)$ と $g'(V', E', L', l')$ が与えられ，$\forall v, v_1, v_2 \in V'$ に対して，以下を満たす単射の写像 $\phi : V' \rightarrow V$ が存在するとき，$g'$ を $g$ の部分木と呼び，$g' \subseteq g$ と表す．

1. $(\phi(v_1), \phi(v_2)) \in E$ if $(v_1, v_2) \in E'$
2. $l'(v) = l(\phi(v))$
3. $l'(\{v_1, v_2\}) = l(\{\phi(v_1), \phi(v_2)\})$

根無し木の部分木は無向グラフの部分木と同様に定義され，順序木の部分木は無順序木の部分木の定義に兄弟間の順序関係を保存する条件を加えて定義される．上記の無順序木の部分木の定義において，条件 1 は頂点間の親子関係の保存，条件 2 と 3 はそれぞれ頂点と辺ラベルの一致の条件である．条件 1 を祖先-子孫関係の保存に変更し，すべての辺ラベルが同一であるとしたとき，$g'$ を $g$ の埋め込み木と呼ぶ．埋め込み木は，順序木や非巡回有向グラフにおいても同様に定義される．

### 4-4-2 頻出グラフマイニング

#### (1)　問題定義

グラフの集合 $G = \{g_1, g_2, \cdots, g_n\}$ が与えられたとき，グラフ $g \in G$ に含まれる部分グラフ $p$ の支持度を $\sigma(p) = |\{g \mid g \in G, p \subseteq g\}|$ と定義する．支持度の閾値である最小支持度 $\sigma'$ 以上の支持度を有する部分グラフを頻出部分グラフと呼ぶ．グラフの集合 $G$ と最小支持度 $\sigma'$ が入力として与えられたとき，頻出グラフマイニングの問題は，すべての頻出部分グラフ $F = \{p \mid g \in G, p \subseteq g, \sigma(p) \geqq \sigma'\}$ を列挙する問題である．頻出部分グラフ $p$ の候補は，頂点と辺，及びそれらのラベルの組合せから成るので，$g \in G$ が大きいとき，頻出部分グラフ $p$ の候補の探索空間は非常に大きくなる．また，頻出部分グラフ $p$ の候補がグラフ $g \in G$ の部分グラフであるかを判定する部分グラフ同型問題は NP 完全であることが知られている．

#### (2)　頻出部分グラフの探索原理

部分グラフ $p$ の支持度は，$p$ の部分グラフ $p'$ の支持度以下になる．すなわち，

$$p' \subset p \Rightarrow \sigma(p') \geq \sigma(p)$$

であり，この性質を支持度の逆単調性と呼ぶ．この逆単調性より，$p$ が頻出部分グラフであるためには，$p$ の部分グラフが頻出部分グラフであることが必要条件となる．**図 4・3**(b)は，図 4・

3(a)の 3 つのグラフと最小支持度$\sigma' = 2$ が与えられたときの探索空間である．紙面の都合上，連結な頻出部分グラフのみを描画している．図 4･3(b)では，頂点数が同じ頻出部分グラフを同じ高さに描いてあり，頻出部分グラフ $p'$ が頻出部分グラフ $p$ の部分グラフであるとき，それらの間に線を引いている．この探索空間の上部から頂点数を 1 つずつ追加しながら頻出部分グラフを探索していくと，支持度は単調に減少する．ある部分グラフ $p''$ の支持度が最小支持度を下回ると，$p''$ に頂点を追加しても支持度が増加することはないので，$p''$ 以下の探索空間を枝刈りすることが可能である．図 4･3(c)は，辺を 1 つずつ追加していく場合の探索空間であり，辺数が同じ頻出部分グラフを同じ高さに描画している．図 4･3(c)の場合も，図 4･3(b)の場合と同様に支持度は単調に減少する．

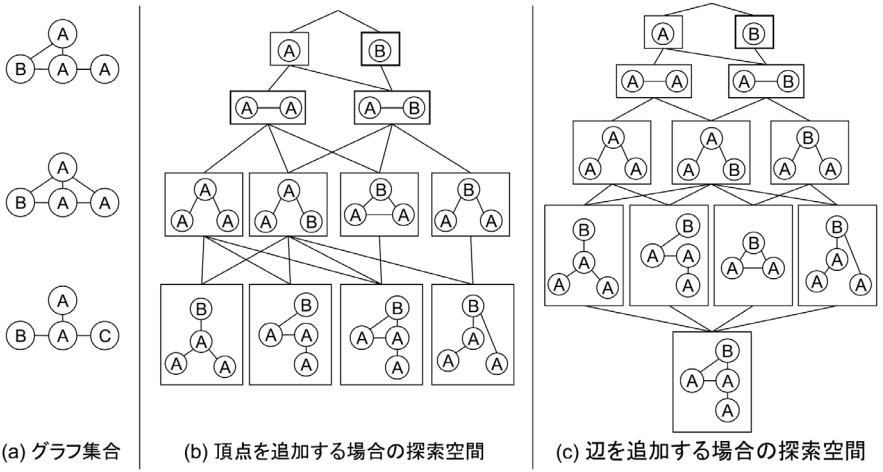



**図 4・3**　頻出グラフマイニングの探索空間

グラフを表現する手法の一つとして隣接行列があるが，隣接行列の行（列）の並び替えによって，複数の隣接行列が同型なグラフに対応する．したがって，探索空間中の部分グラフを隣接行列で扱った場合，同型なグラフが異なる隣接行列で表されている可能性があるので，既に探索した部分グラフであるのかを判定する必要がある．グラフの正準ラベルは，グラフを一意に表すグラフの不変量であり，同型なグラフは等しい正準ラベルを持ち，正準ラベルが等しければ同型なグラフとなる．グラフの正準ラベルを用いることで，グラフ表現の多様性や同型問題の探索空間を著しく削減することができ，後で紹介する代表的アルゴリズムの多くは正準ラベルを用いている．

### (3)　極大頻出部分グラフ，飽和頻出部分グラフ

上記の頻出グラフマイニングの問題では，非常に多くの頻出部分グラフが出力されるため極大頻出部分グラフや飽和頻出部分グラフのみを出力する手法も提案されている．極大頻出部分グラフとは，頻出部分グラフ $p$ に頂点，あるいは辺を加えると頻出でなくなる頻出部分グラフであり，極大頻出部分グラフの集合 $M$ は以下で定義される．

$$M = \{g \mid g \in F \ \wedge \ \nexists g' \in F \ s.t. \ g \subset g'\}$$

ここで，$F$ は頻出部分グラフの集合である．一方，飽和頻出部分グラフの集合 $C$ は，

$$C = \{g \mid g \in F \ \wedge \ \nexists g' \in F \ s.t. \ (g \subset g' \ \wedge \ \sigma(g) = \sigma(g'))\}$$

と表され，頻出部分グラフ $p$ に頂点，あるいは辺を加えると支持度が変わる頻出部分グラフである．マイニングする対象を極大頻出部分グラフや飽和頻出部分グラフに限定することで，極大頻出部分グラフ，あるいは飽和頻出部分グラフになる見込みのない部分グラフを探索の早期に探索空間から枝刈りすることができるので，探索の効率化を図ることが可能である．

**(4) 代表的なアルゴリズム**

頻出部分グラフを列挙するための代表的なアルゴリズムを紹介する．AGM は頂点数が少ない頻出部分グラフから始めてすべての頻出部分グラフを幅優先に探索する手法である．gSpan は，辺を追加しながら頻出連結部分グラフを深さ優先に探索する手法である．Gaston は実世界のグラフで表現されるデータの多くは疎グラフであるという仮定を用いて，枝分かれのない根無し木，枝分かれのある根無し木，閉路有り無向グラフの順に頻出連結部分グラフを探索していく．根無し木の正準ラベルは線形時間で計算でき，疎グラフに含まれる閉路有り無向グラフは非常に少ないので，Gaston は非常に高速に頻出部分グラフを列挙することができる．CloseGraph は gSpan の拡張であり，飽和頻出部分グラフを出力する手法である．SPIN は，極大な頻出連結部分グラフのみを出力する手法である．SiGraM は，1 つの巨大なグラフに頻出する連結部分グラフを列挙する手法である．入力のグラフが 1 つであるので，SiGraM の支持度の定義は前述とは異なる．Generalized AGM はラベルの間の階層関係を導入して頻出部分グラフを列挙する．FREQT や TreeMiner は順序木の集合から頻出部分順序木を出力するアルゴリズムであり，TreeMiner は埋め込み木を対象とする．uFreqt，uNot は無順序木の集合から頻出部分無順序木を列挙する手法である．FreeTreeMiner は，根無し木の集合から頻出部分根無し木を出力する手法である．CMTreeMiner は無順序木の集合から飽和頻出部分木，あるいは極大頻出部分木を列挙する手法である．頻出部分グラフの全探索ではなく，特徴的な部分グラフをヒューリスティックに探索する GBI，SUBDUE などの手法もある．

頻出部分グラフは，グラフ集合の分類やクラスタリングの属性生成の手法としても用いられている．具体的には，出力された頻出部分グラフ $p$ が，事例であるグラフ $g \in G$ に部分グラフとして含まれるとき属性値を 1 とし，含まれないとき 0 とする．事例 $g$ を 2 値ベクトルとして扱い，分類やクラスタリングなどの機械学習手法を適用する．

本節では様々なグラフを定義し，グラフマイニングの問題を紹介した．更に頻出部分グラフの探索原理の説明し，代表的なアルゴリズムを紹介した．更に，詳しく調べたい場合は，解説論文 [1), 2)] やチュートリアル資料 [3), 4)] を参照されたい．

**■参考文献**


1) T. Washio and H. Motoda：“State of the Art of Graph-based Data Mining.”SIGKDD Explorations, vol.5, no.1, pp.59-68, 2003.
2) 鷲尾　隆, 樋口知之, 井元清哉, 玉田嘉紀, 佐藤　健, 元田　浩：“グラフマイニングとその統計的モデリングへの応用,”統計数理, 第 54 巻, 第 1 号, pp.315-331, 2006.
3) J. Han, X. Yan, and P. S. Yu：“Mining and Searching Graphs and Structures,”The 12th ACM Conference on

Knowledge Discovery and Data Mining, チュートリアル資料, http://xifengyan.net/tutorial/KDD06GraphTuto.pdf

4) K. Borgwardt and X. Yan :“Graph Mining and Graph Kernels,” The 14th ACM Conference on Knowledge Discovery and Data Mining, チュートリアル資料, http://www.xifengyan.net/tutorial/KDD08 graph partI.pdf

■S3 群－3 編－4 章

# 4-5 テキストマイニング

（執筆者：新保　仁）［2008 年 12 月 受領］

「テキストマイニング（あるいはテキストデータマイニング）」はその名のとおりデータマイニングの一分野であるが，この用語は広い意味で大規模テキスト（文書）データからの情報抽出技術一般を指すことが多い．初期のデータマイニング法（バスケット解析）が対象とする*2（属性や関係といった構造が明確な）関係データベーステーブルと比べると，テキストマイニングが対象とする自然言語データは，明確な構造を持たないという大きな違いがある．

テキストマイニングではこのため，文字の列に過ぎないテキストを，自然言語解析技術を用いて一定の構造を付与した中間表現に変換し，そこから機械学習や統計的手法を用いて有用な情報（知識）を抽出する，という手順をとるのが一般的である．

中間表現としては従来，素朴に文章内の単語の列や単語集合，品詞が付与された単語列，などが用いられることが多かったが，近年の自然言語解析技術の発展にともない，基本句チャンク，句構造木，依存構造（係り受け）木，といったより複雑な表現も徐々に活用され始めている．

このような傾向は，解析精度の向上によるところも大きいが，同時にテキストマイニング技術への社会的需要の変化にともない，研究テーマが文書単位の（分類などの）処理から，段落や文といったより小さな単位の情報抽出へと拡大しつつあることを反映している．単語集合のような粗い表現では，こういった小さな単位の持つ情報の記述としては不充分であり，単語間の関係なども含めた，より詳細な表現が必要なタスクも多い．

テキストマイニング技術に対する社会的な需要は，その要素技術である自然言語処理の研究にも影響を与えており，両者にまたがる新しい研究タスクを生み出している．

## 4-5-1 大規模テキストデータへの情報アクセス

大量かつ更新の激しい Web 上のテキストデータを高速に処理し，個々の内容を要約，あるいは全体的な傾向を分析して提示する技術への需要はビジネス分野において大きく，これらはしばしばテキストマイニングの主要なアプリケーションとみなされている．

データマイニングの一つの目的が，統計的な基準に基づいて特徴的とみなせるパターンを抽出することだとすると，この目的は（対象こそデータベーステーブルと文書という違いはあるものの）従来より統計的自然言語処理が取り組んできたタスクと重なる面が多い．成句（コロケーション：“New”“York”など）抽出などがこれにあたる．こういった単語間の関連性抽出は，Web サービスなどで有用とされる，関連キーワード（タグ）の自動抽出に応用できる．

統計的自然言語処理分野で研究されてきた，文書クラスタリング，トピック抽出や，複数文書要約（重要文抽出）なども，大量の文書の傾向を要約するための有用な技術である．

このほかに注目を集めている新しい技術として，評価解析（Sentiment Analysis）が挙げられる．評価解析は，個々のテキストが，特定の事柄（出来事や品物など）について，肯定的な意

*2 近年のデータマイニング研究では，対象データが必ずしも関係データ，属性データの範疇に収まらない場合（系列，木構造，グラフなど）も多く，テキストマイニングにも応用可能な技術が多数提案されている．4-5-2 節を参照．

見の表明なのか，否定的な表明なのかを判定する技術であり，Web 上に多数見受けられる商品評価記事やブログの自動解析が研究者の想定する主要なアプリケーションである．

評価解析の近年の研究動向については，乾・奥村 [4] が包括的なサーベイを行っている．

### 4-5-2 データマイニング技術を応用した自然言語処理

近年，データマイニング研究の対象が関係データベースから木構造やグラフといったデータに拡大するにつれ，これらのデータに対する効率的なパターン列挙法が開発され，自然言語の構文解析データにも適用され成果をあげている．

工藤らの BACT[5] は，自然言語解析技術によって生成した多数の木構造（文の依存構造木など）と，それらに対する二値のラベル（教師信号：例えば，評判分類の場合には，肯定的な評価の文と，否定的評価の文）を訓練事例として受け取り，部分木を素性（分類の手がかり）とした分類規則（分類器）を出力する．BACT には，データマイニング分野における浅井ら [1] による部分木の効率的な数え上げ技術，及び Decision Stump（1 段のみからなる決定木）ブースティングにおける森下 [6] の分枝限定法が効果的に組み合わされて利用されている．

また，複数文書からの重要文抽出においても，リンク解析アルゴリズムを応用した手法が良い精度を出すと報告されている．文献 3）には，データマイニング技術からのテキストマイニングへのその他のアプローチについての解説がある．

### 4-5-3 テキストからの未知の知識の発見に向けて

テキストマイニングと統計的自然言語処理の区分は必ずしも明確ではないが，M.A. Hearst は 1999 年の講演 [2] のなかで，大量の文書から（書き手にとってすら）未知の知識を発見するタスクこそが，「発掘（マイニング）」という文字どおりの意味での「真のテキストマイニング」であると述べた．更に，文献からの科学的発見の例として著名な Swanson の研究 [7]（ただし，Swanson 自身の発見は主要な作業は人手によって行われているため，テキストマイニング技術を用いたものではない）を引用し，同様の知識発見過程の自動化，ないしは，そのための探索的データ解析過程を補助する技術が重要と説いた．

Hearst の言う知識発見は容易な問題ではなく，現状では，自然言語処理技術の大幅な進展なくしては実現困難と思われる．近年盛んに研究されている，文中に言及された事態間の含意関係（Textual Entailment）を自動認識する技術などは，これに向けた取り組みとみなせる．含意関係認識は，Hearst の定義からすると既知の情報を抽出しているに過ぎず，彼女の言うテキストマイニングとは合致しない．とはいえ，Swanson の科学的発見においては，因果関係の認識が重要なステップとなっており，その自動化のためには必要な技術である．4-5-1 節で紹介した研究なども，それら自体は新規の知識発見に直接結び付くわけではないが，探索的データ解析を補助するための有効な技術である．

**■参考文献**

1) Tatsuya Asai, Kenji Abe, Shinji Kawasoe, Hiroki Arimura, Hiroshi Sakamoto, and Setsuo Arikawa：“Efficient substructure discovery from large semi-structured data,”in “Proceedings of the Second SIAM International Conference on Data Mining,” Arlington, VA, USA, 2002.

2) Marti A. Hearst：“Untangling text data mining,”in Proceedings of the 37th Anuual Meetings of the Association for Computational Linguistics, pp.3-10, 1999.

3) 有村博紀：“テキストマイニング(2),”人工知能学会(編)：“人工知能学事典,”pp.676-677, 共立出版, 2005.
4) 乾　孝司, 奥村　学：“テキストを対象とした評価情報の分析に関する研究動向,”自然言語処理, vol.13, no.3, pp.201-241, 2006.
5) Taku Kudo, Kaoru Yamamoto, and Yuji Matsumoto：“A boosting algorithm for classification of semi-structured text,”in Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, pp.301-308, Barcelona, Spain, 2004.
6) Shinichi Morishita：“Computing optimal hypotheses efficiently for boosting,”in Progress in Discovery Science}, Lecture Notes in Computer Science 2281, pp.471-481, Springer, 2002.
7) Don R. Swanson：“Two medical literatures that are logically but not bibiliographically connected,”Journal of the American Society for Information Science, vol.38, no.4, pp.228-233, 1987.