03-01037

大規模文書からのアウトライヤー文書クラスター検出方法の研究

青　野　雅　樹　　豊橋技術科学大学工学部教授

1　研究調査の目的

Web 上の資源から有用なデータを発掘する「データマイニング」分野において，通常は，与えられた Web 資源データ（以降，単にデータと呼ぶ）に対し，データの全体的な傾向を把握することが主に研究されている。一方，全体的な傾向から逸れたアウトライヤー（外れ値）を検出する技術が最近のデータマイニングの重要な研究テーマのひとつとなってきている。このような異常値の中には，単純にノイズといえるものばかりでなく，新しいトレンドを示す重要なデータが含まれていることがある。このような異常値を検出する技術を「アウトライヤー（外れ値）検出」と呼ぶ。本研究調査の目的は，アウトライヤーを大規模文書データの中からできるだけ正確でかつロバストに検出でき，スケーラブルな手法の研究開発である。

アウトライヤー検出の応用分野のひとつに不正検出（fraud detection）がある。応用対象としては，ネットワークの不正侵入検出，携帯電話の成りすまし利用，クレジットカードの不正利用の検出，医療や保険業界における不正請求検出，電子商取引での違法行為の検出などの電気通信分野での応用があげられる。

また，Web 上のニュース記事等から新しいトレンドを検出し，新商品開発などの提案材料としての応用も期待できる。更に，論文や特許情報データなどから，稀少だが価値の高い文書の発見による企業や組織の基礎研究・特許戦略などへの応用も可能である。このような様々な応用分野に対処できるロバストでスケーラブルなアウトライヤー検出手法で，データが非常に高次元データ（たとえば 10,000 次元以上）である場合のアウトライヤー検出手法はまだ確立されておらず，本研究の目的は，このような応用分野に適用可能な高次元データからのアウトライヤー検出手法の研究・開発にある。

2　研究調査の方法

2.1　当初の調査方法

研究調査方法は，以下の手順で行う予定で開始した。

（1）データの収集

Web 上から英語および日本語文書データセットを少なくともそれぞれ 1 種類以上収集する。英語に関しては，NLM（http://www.ncbi.nlm.nih.gov/PubMed/）からダウンロードできる医療文献データのアブストラクトを収集する。日本語に関しては，特許庁からダウンロードできる特許データ，情報処理学会や電子情報通信学会などからダウンロードできる論文のアブストラクトデータ，もしくは，国立情報学研究所の NTCIR から入手できる多岐にわたる学術文献データなどから収集する。データ量は，英語，日本語それぞれ約 100GB 程度になると予想され，これをディスクなどの記憶装置に保持する。

（2）データの前処理とその環境整備

収集したデータのうち，英語文書データに関しえては，ニューヨーク大学が公開している“OAK”と呼ばれる形態素解析器を用いて，文書データからキーワードを抽出する。日本語文書データに関しては，奈良先端技術大学院大学が公開している「めかぶ」（または「茶筅」）と呼ばれる形態素解析器を用いて，文書データからキーワード抽出を行う。この際，データに応じて，不要語（stop word）やあいまい語，同義語辞書などを併用して，重要な（しかし出現頻度が低いかもしれない）キーワードが漏れないよう工夫する。この処理では，中間データ

が英語および日本語それぞれ元データの約1.5倍の記憶領域を必要とすると見積もられ，元データと中間データをあわせて 500GB 程度のデータが収まるディスクを用意しておく。CPU 時間としては，１文書あたり，平均200のキーワードがあると仮定し，Intel Pentium4 3GHz 相当の CPU と 3GB の主記憶装置を持つ PC で平均約１秒必要と見積もられ，100万件のデータでは，24時間稼動の PC を用い，約２週間で１サイクルの形態素解析が終了すると見積もられる。６ページの購入希望の研究用ワークステーションは，これらの処理を行うのに必要な最低水準のスペックの PC ワークステーションである。なお，メモリ 3GB をフルに利用するため，マイクロソフト社の Visual Studio .NET（アカデミック版）でのプログラムを利用する。

（3）国際会議での情報収集

８月22日から25日まで米国シアトルで開催される ACM SIGKDD（Special Interest Group on Knowledge Discovery and Data Mining）と呼ばれる国際会議に出席し，本研究に関連する最新技術動向を調査する。

（4）文書データのベクトル化

⑵の処理で取り出されたキーワードをもとに，文書×キーワード行列を構築する。ここで，キーワードの重みとしては，たとえば TF-IDF（Term Frequency-Inverse Document Frequency）と呼ばれる，全体から見ると出現頻度の少ないキーワードの重みが大きくなり，アウトライヤー文書を検出しやすいような重み付けモデルを用いる。各文書は，この行列の「行」に相当し，キーワードが「列」に対応する。各行のことを，その文書の「ベクトル」と呼ぶ。アウトライヤー検出の基本アルゴリズムを完成させる。

（5）学会への投稿・研究成果の公表

2004年秋以降は，基本モデルにサポートベクトルマシンによる教師つき学習を組み合わせ，メジャーなクラスタに，高い確率で分類される文書データをあらかじめ排除する手法を導入し，アウトライヤー検出の精度の向上を目指すと同時に，スケーラビリティ（大量データでの動作）テストの実証実験を行い，結果のデータをまとめ，国内学会発表の準備を行う。

2.2　実際の調査方法

実際の研究調査方法は，以下の手順で行った。

（1）データの収集

Web 上から英語および日本語文書データセットを少なくともそれぞれ１種類以上収集した。英語に関しては，NLM（http://www.ncbi.nlm.nih.gov/PubMed/）から医療文献データのアブストラクトを収集した。これは全部で100万分書以上あるが，専門用語などが多く，索引語抽出のため専門用語辞書が不可欠なため，現在はアーカイブとして保存している状態である。しかし，これとは別に，TREC（http://trec.nist.gov/）から Los Angeles Times のニュースデータ（約10万文書）を入手した。また，UCI Archive（http://kdd.ics.uci.edu/）から KDD カップ（コンペティションデータセット）と Reuters ニュースデータ（ベンチマーク約２万文書）をダウンロードした。Reuters データに関しては，**3 節**の**研究調査の結果**で述べるようにアウトライヤー文書の検出実験を行った。日本語に関しては，国立情報学研究所の NTCIR5 タスクに参加すると同時に，特許データを NTCIR 事務局（http://research.nii.ac.jp/~ntcadm/index-ja.html）から入手した。データ数は400万件程度（1993年から2002年の特許テキストデータ）ある。

（2）データの前処理と環境整備

収集したデータのうち，英語文書データの例として Reuters ニュースデータを，ニューヨーク大学が公開している“OAK”と呼ばれる形態素解析器を用いて，文書データからキーワードを13,300個抽出した。日本語文書データ（特許データ）に関しては，奈良先端技術大学院大学が公開している「茶筅」を用いて，文書データから38,400個のキーワード抽出を行った。日本語特許データに関しては，約12000語の単語からなる辞書（特に，化学物質名が多い）をマニュアルで作成した。特許データ400万件を保持するのに，ハードディスク約 150GB 消費した。これに茶筅での形態素解析結果を保持するのに１年あたり 5GB 程度の中間ファイルを出力し，10年分

で中間ファイルは約 50GB 程度となった。このため 400GB のディスクを数個（バックアップも含む）利用し，保持させた。この処理では，中間データが英語および日本語それぞれ元データの約1.5倍の記憶領域を必要とすると見積もられ，元データと中間データをあわせて 500GB 程度のデータが収まるディスクを用意して保持させた。なお，本研究用の PC やハードディスクは，当初の TAF への申し込み額から30万円ほどの削減があったため，大学の基盤経費で購入し，あとで述べる学会参加費や旅費を当助成金で割り当てることとした。

（3）国際会議での情報収集

2004年 8 月に米国ワシントン州シアトルで開催された ACM SIGKDD2004 (KDD=Knowledge Discovery and Data Mining) に参加して，データマイニングと知識発見の分野の最新動向を調査した。アウトライヤーの検出アルゴリズムも数件の発表があった。この海外出張に伴う情報収集の最大の収穫は，本報告の中心アルゴリズムのヒントとなる相互情報量の最大化に基づく，「共クラスタリング」（co-clustering）の発明者であるテキサス大学オースチン校の Inderjit Dhillon 先生とポスターセッションで出会ったことである。

（4）文書データのベクトル化

⑵の処理で取り出されたキーワードをもとに，文書×キーワード行列を構築した。ここで，キーワードの重みとしてTF-IDF (Term Frequency-Inverse Document Frequency) モデルを利用した。各文書は，この行列の「行」に相当し，キーワードが「列」に対応する。各行のことを，その文書の「ベクトル」と呼ぶ。アウトライヤー検出の基本アルゴリズムを完成させた。

（5）学会への投稿・研究成果の公表

2004年秋以降は，基本モデルに当初は，サポートベクトルマシンによる教師つき学習を組み合わせ，メジャーなクラスタに，高い確率で分類される文書データをあらかじめ排除する手法を導入する予定であったが，「教師付き学習モデル」では，訓練のため正解がわかっているデータ集合が必要であるため，予定を変更し，「教師無し」学習の中で当研究課題にもっとも有効であると思われる，⑶で述べた共クラスタリング・アルゴリズムをアウトライヤー検出のベースとして用いることとした。その上で，新しいアルゴリズムを考案し，電子情報通信学会，情報処理学会，および日本データベース学会共催の DBWS2005（2005年 7 月開催）に投稿し，発表した。また，英語の論文を作成し，これを IEEE ICDM2005 に投稿した。前者のタイトルは，「クラスタ粒度階層構造を用いたアウトライヤー文書の検出方法」（添付資料参照）で後者のタイトルは，“Detecting Outlier Documents Using a Hierarchy of Clusters”である。

3　研究調査の結果

3.1　本研究結果の概要

本テーマを行うにあたって，日本語の文書データとして特許データを，また英語の文書データとして，Reuters ニュースデータ，Los Angeles Times ニュースデータなどを収集した。収集後，前処理として日本語および英語の形態素解析器を用いて，索引語を抽出し，日本語特許文書では，38,400単語，英語の Reuters ニュースデータでは，13,300単語を抽出し，これらをもとに，文書をベクトル化した。これは，前者のデータは次元数が38,400次元，後者は13,300次元あることを意味する。アウトライヤー文書検出の過去の研究事例でこのような高次元データで行った事例は，筆者の知る限りない。（過去の事例では，高次元としているものも高々数十次元である。）検出アルゴリズムとしては，前段階で，共クラスタリング（co-clustering）を複数回，異なる粒度（粒度=生成するクラスター数）で行い，平滑化アルゴリズムと階層構築アルゴリズムを経て異粒度階層データ構造を構築した。その後，対象とする文書ベクトルデータを階層構造のノードに相当するクラスタ平均ベクトルの類似度を比較し，各文書データで，もっとも類似度の高いクラスタ平均ベクトルデータと，その平均ベクトルまでの距離の相対比で「アウトライヤー度」を定義した。全文書に関して，アウトライヤー度を計算し，これが最も大きかったものから適当な閾値までの文書を「アウトライヤー文書」として検出し，報告させる手法を開発した。これを国内の学会（こちらは，2005年 7 月に発表予定）と，国際会議（IEEE ICDM05）へ投稿した。

3.2 先行研究

統計学におけるアウトライヤー検出は，かなり古くから重要視されてきた技法であり，統計データに潜在的に含まれる「ノイズ」をアウトライヤーとしていかに除去するかという観点から研究されてきた。統計学的なアウトライヤーの話題（これは「(確率）分布ベースのアウトライヤー検出手法」と呼ばれることがある。分布ベースの手法では，多次元データに対しての検出方法に難点があるとされる）は，[5]に詳しく述べられている。一方，アウトライヤー検出の計算機科学的なアプローチの歴史は浅く，アウトライヤー検出の主目的は，電子商取引での不正検出，クレジットカードの不正使用検出，ネットアークへの不正侵入の検出，突然現れる潜在的に重要なニュースなどの早期発見に代表される知識処理・知識発見である。当分野でアウトライヤー検出法などの論文が発表されたのは近年になってからである。アウトライヤー検出技術を大別すると以下に分類できる。

(1) 距離ベースのアウトライヤー検出
(2) 密度ベースのアウトライヤー検出
(3) 確率分布ベースのアウトライヤー検出
(4) 低次元への射影ベースのアウトライヤー検出
(5) サンプリングベースのアウトライヤー検出
(6) 局所相関積分ベースのアウトライヤー検出
(7) クラスタリングベースのアウトライヤー検出

以下，これら7つの手法に関して概説する。

(1) 距離ベースのアウトライヤー検出

この範疇に含まれる研究としては，Knorr ら[12,13]により報告されている。Knorr らは，距離ベースのアウトライヤーを $DB(p,D)-outlier$ と定義した。ただし，p はデータ集合の全体のうちの割合を表し，D はあるオブジェクト（データ点）からの距離を表す。すなわち，あるオブジェクトが $DB(p,D)-outlier$ に属するとは，データの数の割合が少なくとも p 以上で，そのオブジェクトからの距離が D 以上はなれたデータのことをいう。データ点が正規分布する場合，アウトライヤーは 3σ（σ は標準偏差）以上はなれたデータに相当すると論じている。

距離ベースのアウトライヤー検出法では N-最近傍アルゴリズムか，範囲探索アルゴリズムがあればよい。このとき，アウトライヤーでない場合とは，$DB(p,D)-outlier$ の最大数をMとし，$DB(p,D)-outlier$ の数が $M+1$ 個以上になったものは，アウトライヤーでないとして棄却すればよい。すべてのデータ点を走査して，$DB(p,D)-outlier$ に含まれるオブジェクトの数が M 以下であれば，アウトライヤーであると判定する。計算時間は一般に $O\left(kN^2\right)$ である。ただし，k はデータの次元を表し，N はデータ総数を表す。

なお，距離ベースのアウトライヤー検出の最大のボトルネックは計算量であり，Ramaswamy ら[16]は，空間分割を利用した高速な近傍計算アルゴリズムを述べている。また，Angiulli ら[4]は，後述するサンプリングベースと距離ベースのアルゴリズムを融合させたヒルベルト曲線に代表される空間充填曲線を用いた手法でのアウトライヤー検出を報告している。

(2) 密度ベースのアウトライヤー検出

Breunig ら[7]は, LOF に基づく定量的なアウトライヤーを定義した。LOF とは，Local Outlier Factor（局所的なアウトライヤー量）を意味し，密度を用いて，アウトライヤーを量的に検出する手法を述べている。この手法の背景としては，**図1**のようにデータが分布しているとき，距離ベースのアウトライヤー検出手法では，図中のアウトライヤー o_1 は検出できるが，o_2 は検出できないという問題がある。

これに対して，LOF では，アウトライヤーを量的に定義できるだけでなく，クラスタ C_1, C_2 のばらつきや大きさに関係なく，ローカルなアウトライヤーを決定できるので，o_1 も o_2 も検出できるという利点がある。

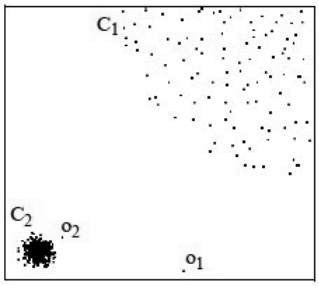

図1　密度の異なるクラスタにおけるアウトライヤーの分布のイラスト

（3）確率分布ベースのアウトライヤー検出

この範疇に含まれるアウトライヤー検出手法は，もともと統計学で用いられていた手法の延長線にある技術であるが，日本電気の山西ら[1,17]は，SmartSifter と呼ばれるエンジンを開発した。これは，与えられたデータがガウスの混合分布からなると仮定し，そのパラメータをオンラインで推定しながら，修正していくというものである。ただし，もともと非常に多くの仮定を必要とすること，高次元への対応が困難であることなどから，次元の小さいデータや応用分野に限られるという問題がある。なお，山西らは，上述の SmartSifter の改良版で，教師つき学習と教師なし学習を組み合わせる手法[18,19]も述べている。

（4）低次元への射影ベースのアウトライヤー検出

Aggarwal ら[2]は，高次元データの場合，「次元の呪い」のため，従来の方法では動作しない，あるいはアルゴリズムの適用自体が非常に困難であるという問題を指摘した。この問題を克服するために，彼らは密度ベースの手法を間接的に利用しながら，あるオブジェクトが低次元への射影にて「異常な」領域に含まれる場合，アウトライヤーであるとするアプローチを述べた。「異常な」領域は，k 次元の立方体格子を考え，各立方体での「疎率」（sparsity coefficient）を定義し，この値が負となる場合を異常と定義している。ただし，「異常な」領域を含む射影をいかにして効率的に発見するかがキーとなる。そこで筆者らは，遺伝的アルゴリズムを用いてアウトライヤーを含む射影を効果的に発見できる手法を述べている。

Ando ら[3]は，LSI（Latent Semantic Indexing）で提唱された特異値分解に基づく次元削減において，単純に LSI を 1 回適用して次元削減すると，メジャーなデータ（非アウトライヤーデータ）しか反映されない点に着目した。その上で，1 回の特異値分解で一番大きい特異値に対する特異値ベクトル（基底ベクトル）だけを取り出し，2 本目からは，それまでに取り出した基底ベクトルの影響を，その方向の残差（residual）をとることで，軽減できることを示した。この方法の問題は，計算量と記憶容量が膨大になるため，実用性に欠けることである。

（5）サンプリングベースのアウトライヤー検出

Kollios ら[11]は，与えられたデータが大規模である場合，サンプリングが大規模なデータからクラスタリングやアウトライヤー検出に有効な手段であることを述べている。ただし，通常のランダムサンプリングを実行すると，データの分布を反映できないので，データの分布（かたより）に応じたバイアス付きサンプリングを提案した。アウトライヤー検出方法としては，新しい手法を提案したわけではなく，Knorr らの距離ベースの方法を用い，一方バイアス付きサンプリングのために既存のクラスタリングを用いるとしている。また，密度に応じたバイアス付きサンプリングのために，カーネル方法で推定するのが速度的にも実用的であると報告している。

Bay ら[6]は，サンプリングと距離ベースのアウトライヤー検出法を用いた実用的にリアルタイムに近いアルゴリズムを提案している。手法としては，Knorr らが提案した距離ベースの最も素朴なアルゴリズムにランダムサンプリングと「刈り取り」（pruning）を加えるだけで劇的に性能が向上することを実証している。ただし，Bay らの手法は，データのランダム性とアウトライヤーが必ず存在することが前提となっており，そうでない場合は，性能が下がると警告している。

（6）局所相関積分ベースのアウトライヤー検出

Papadimitriou ら[15]は，MDEF（Multi-granularity Deviation Factor）という量を定義し，これを用いてアウトライヤーとなるデータ点，およびミクロなクラスタを $O(kN)$（k は次元，N はデータ数）で実現できる手法を報告している。MDEF は，相関積分（correlation integral）に関連する概念で，データのアウトライヤー性（“outlierness”）を量的に評価できると述べている。この点では，Breunig らの密度ベースの LOF に似ている。また，MDEF の近似版である aMDEF（a は“approximate”（近似）の意味）も提案している。なお，本手法は密度ベース，距離ベース，分布ベース，およびサンプリングによるアウトライヤー検出を組み合わせた手法と捉えることもできる。実際，データ点のアウトライヤーの判定には，分布ベースの手法で提案された標準偏差からの「3σ ルール」を用いている。同著者らは，MDEF に教師つき学習モデルの代表例である SVM（サポートベクトルマシン）を適用した，ユーザに得られた結果の positive（正例），negative（負例）の判定を行ってもらう手法も報告している[21]。

（7）クラスタリングベースのアウトライヤー検出

クラスタリングを用いてアウトライヤー検出を行う研究は，90年代後半から徐々に報告されている。ただし，メジャーなクラスタは容易に検出できるが，マイナーなクラスタ検出は困難とされており[14]，アウトライヤー検出は至難であると考えられてきた。Yu ら[20]は，ウェーブレットを利用して，メジャーなクラスタを排除することで，アウトライヤー検出する手法を述べている。He ら[10]は，データがすべてクラスタリングされると仮定して，小規模なクラスタをアウトライヤーとして定量的に検出する手法を述べている。

3.3 本研究結果の詳細な説明

提案手法は，クラスタリングを用いたアウトライヤー検出に分類される。しかしながら，クラスタリングを用いる場合には，注意すべき共通の問題があり，この点を考慮しないと，単にクラスタリングの結果を利用しただけでは，かえって逆効果となる場合もある。

なお，ここでの実験では，クラスタリング手法として，3.2の参照文献にあげた Dhillon ら[8,9]の提案した共クラスタリング（co-clustering）を用いている。以下では，クラスタ情報を効果的に利用するデータ構造を提案し，それに基づくアウトライヤー検出手法を論じる。

3.3.1 クラスタリングの問題点とその対応策

クラスタリングは，対象とするデータによっては有効であることが期待されるが，手法にかかわらず，クラスタリングの結果をそのまま利用すると，必ずしも応用分野が要求する性能を向上できない場合がある。代表的なのは，以下の 2 つの問題である。

① 生成したいクラスタの数（これは通常，ひとつのパラメータとしてクラスタリング・アルゴリズムに与える）によって，実際に生成されるクラスタの品質が変化してしまうという問題

② 初期条件（たとえば乱数の初期値や収束判定の閾値など）によって，クラスタリングの結果にバラつきが生じるという問題

この 2 つの代表的なクラスタリングの問題を緩和するため，以下のような対応策を講じた。

①の問題を緩和するために，クラスタ数を 2 のべき乗（たとえば16,32,64,128,...）で変更させ，粒度の異なるクラスタリングを実施することで，与えられたデータオブジェクト集合全体から判断して，ノイズとなるデータ以外は，どこかの粒度のクラスタで吸収させるようにした．また，アウトライヤー検出時に使用するデータ構造として，粒度の異なるクラスタ間で，適当な閾値以上の類似度を有するクラスタ同士にリンクをつけ，「クラスタ粒度階層構造」を作成した。

②の問題を緩和するために，①で述べたそれぞれの粒度で乱数の初期値を変更し，クラスタリングを複数回実行し，同一のクラスタに含まれる確率の高いデータ同士を最終的に同一クラスタに帰着するような平滑化アルゴリズム（**図 1** のアルゴリズム参照）を考案した．また，②で述べた平滑化アルゴリズムで，どこにも含まれない文書を潜在的なノイズとして，クラスタ集合に加えない方策をとった。

アウトライヤーの検出フェーズでは，各データ要素が与えられたとき，①で述べた粒度の粗い順に，クラスタ

を代表するベクトル（これを「クラスタ平均ベクトル」と呼ぶことにする）との類似度を比較し，その類似度がある閾値以上の場合のみ，階層構造を下向きにたどり，粒度のより細かいクラスタ平均ベクトルとの類似度計算を行い，階層構造の最も下部まで到達したときに得られる最も類似度の大きいクラスタ平均ベクトルとの類似度で，アウトライヤー度の計測に用いることとした。

3.3.2 アウトライヤー検出手法

前節の観察と，クラスタリングにおける問題点の対応策に基づき，共クラスタリングに基づく，アウトライヤー検出法を考案した．以下に提案手法を述べる。

提案手法では，まず異なる粒度 $M_1, M_2, \ldots, M_k$ で，それぞれ R 回クラスタリングを実行する。（「多粒度クラスタ平滑化アルゴリズムの [step1]」。

「多粒度クラスタ平滑化アルゴリズム」

[step1] クラスタ粒度 M，（乱数の）初期値を変更して R 回，クラスタリングを実行する。得られたクラスタを $\mathbb{D}^{\mathrm{r}}=\left\{\mathbf{D}_1^{\mathrm{r}}, \mathbf{D}_2^{\mathrm{r}}, \ldots, \mathbf{D}_{\mathrm{M}}^{\mathrm{r}}\right\}(\gamma-1, \ldots, R)$ と表現する。

[step2] クラスタベクトル $\mathbb{H}=\{\mathbf{H}_1, \mathbf{H}_2, \ldots, \mathbf{H}_{\mathrm{M}}\}$ を $\mathbb{D}^1$ で初期化する。

[step3] 後述のクラスタ平滑化アルゴリズムを実行する．ただし，$\mathbf{K}(\mathbf{j})$ は，γ 回目のクラスタリングで得られたクラスタ $\mathbf{D}_{\mathrm{j}}^{\mathrm{i}}$ の要素数を表す。

こうして得られたクラスタデータから，[step2] を適用して，拡大クラスタ $\mathbb{H}$ を得る。この後，[step3] でクラスタの平滑化アルゴリズムを適用する。下記に示す「クラスタ平滑化アルゴリズム」（ClusterSmoothing）の 7 行目の Similarity 関数は，クラスタベクトル H と個々のデータベクトル **D** との類似度を計算する。

類似度がある閾値 (δ) より大きければ， 8 行目にある Average 関数で，クラスタベクトルを，データベクトル **D** に基づき方向修正を（式１）で行う。

$$\begin{aligned} \mathbf{H}_{\mathrm{i}} &= \sum_{i \atop D(i)} (\mathbf{h}_{\mathrm{i}} + \lambda \mathbf{d}_{\mathrm{i}}) \\ d_i &= \sum_{j=1} \alpha_j \mathrm{w}_{\mathrm{j}} \\ \mathbf{H}_{\mathrm{i}} &= \frac{\mathbf{H}_{\mathrm{i}}}{\left\|\mathbf{H}_{\mathrm{i}}\right\|} \end{aligned} \qquad \text{（式１）}$$

```
Algorithm  Cluster Smoothing (ℍ, R,M)
1:  for γ =2 to R
2:    for j=1 to M
3:      D_j^r = {d_{j,1}^r, d_{j,2}^r, ..., d_{j,k(j)}^r};
4:      for i=1 to |ℍ_1|
6:        H = H_i ; D = D_j^r ;
7:        if (Similarity (H, D)>δ) then
8:          H_i = Average (H,D);
9:        else /* 追加 */
10:         ℍ=ℍ ∪D;
11:       endif
13:     end for
14:   end for
15: end for
```

クラスタ平滑化アルゴリズム

ただし，$D(i)$ は，クラスタ数の上限値を表し，α_j, λ は非負の実数パラメータである。乱数の初期値によらないクラスタの平滑化がなされた後，「クラスタ階層構造化アルゴリズム」により，異なる粒度で得られたクラス

タ間に階層関係を構築する。

「クラスタ階層構造化アルゴリズム」

[step1] 粒度の粗い順に，隣接する粒度 M_i と M_{i+1} の間で，対応する H_i と H_{i+1} の相互クラスタ類似度を，クラスタ平均ベクトルの類似度より求める。

[step2] もし，[step1]で，H_i の中のクラスタ $\mathbf{D}_i$ と H_{i+1} の中のクラスタ $\mathbf{D}_{i+1}$ の類似度がある閾値より大きければ，$\mathbf{D}_i$ と $\mathbf{D}_{i+1}$ の間にリンクをつける。

[step3] [step2]を最も粒度の細かいクラスタ H_k まで繰り返す。

図 2 は，実験4-2で用いた特許データに対して，「クラスタ階層構造化アルゴリズム」により得られたクラスタ階層の例である。一般に，粒度の粗いクラスタに含まれる個々のクラスタは，そのクラスタを構成するデータ集合の要素数が大きなクラスタであることが多く，たとえば，粒度=16では，「画像，画像データ」という単語に代表されるクラスタ，「細胞，酵素」という単語に代表されるクラスタなどが検出される。

粒度が細かくなると，粗い粒度では現れなかった構成データ要素数の少ないクラスタが表れるようになる。たとえば，粒度=16では現れなかった「遊技，パチンコ」という単語に代表されるクラスタが粒度=32で現れたり，粒度=64になると，「免振，免振装置」という単語に代表されるクラスタが新たに現れたりする。

一方，粗い粒度で検出されていたクラスタの一部は，2つ以上のサブクラスタに分割されて現れる場合がある。たとえば，粒度=64の「油圧，ピストン」という単語に代表されるクラスタは，粒度=128の「油圧，ブレーキ」という単語に代表されるクラスタと，「ピストン，シリンダ」という単語に代表されるサブクラスタに分割されて現れる。

逆に，粗い粒度の2つ以上のクラスタが細かい粒度のひとつのクラスタにリンクを持つ場合も生じる。たとえば，粒度=32の「画像，画像データ」という単語に代表されるクラスタと，同じ粒度の「画像，撮像」という単語に代表されるクラスタは，粒度=64の「画像，原稿」という単語に代表されるクラスタにともにリンクを有する。これは，本提案手法で得られる階層構造が，実際は木構造でなく，有向グラフ構造であり，特定の粒度のクラスタ（グラフのノード）の親クラスタ（親ノード）がある場合，それは必ずしもユニークでないことを意味する。一般に，クラスタ間の類似度が高い粒度の異なるクラスタ同士にリンクをつけると，**図 2** に例示するような階層構造が得られる。

「クラスタ粒度階層構造」が得られたら，各データ要素に対して，この階層グラフ構造を，粒度の粗い方から順にデータを表すベクトルとクラスタ平均ベクトルとの類似度を計算していく。類似度が特定の閾値を超えるノードクラスタが検出されたら，そのノードからリンクをたどり，下位ノードとの類似度計算を繰り返す。また，ある粒度で閾値を超える類似度が得られない場合は，下位粒度のクラスタで，新たにグラフの出発点となるノード集合と類似度計算を行う。これをグラフ全体で行うことにより，もっとも類似度の高いクラスタとその類似度

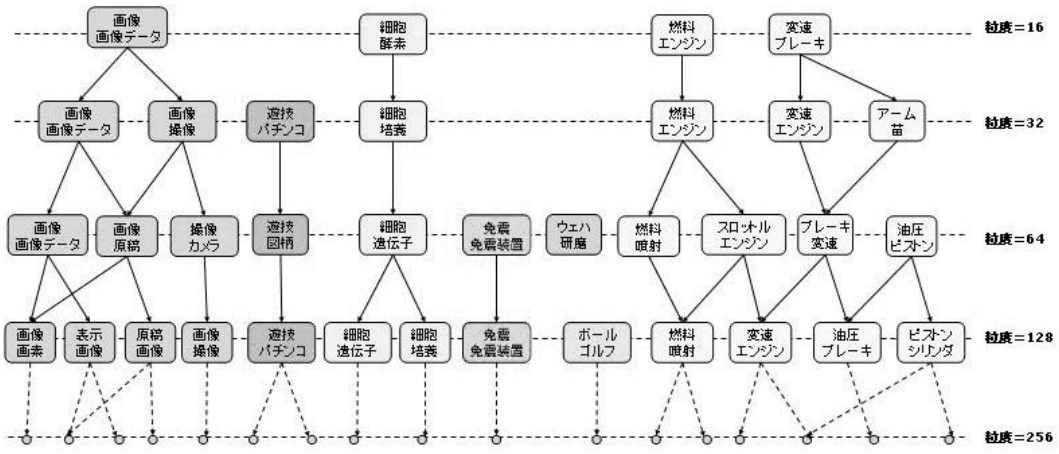

図 2　異粒度のクラスタリングを複数回行い，平滑化後，生成されたクラスタ粒度階層構造の一部

を保持しておく。この類似度をそのデータに関する最大類似度と呼び，対応するクラスタを最大類似クラスタと呼ぶ。

一方，**図 2** のような階層構造のノードに対応するクラスタにおいて，そのクラスタ平均ベクトルを求めると同時に，クラスタに属するデータのクラスタ平均ベクトル（クラスタの中心に対応する）との距離（類似度）の平均値（$\bar{d}$ とする）を求めておく。あるデータがアウトライヤーとするのは，そのデータと最大類似度を有するクラスタとの距離（d_i とする）が，そのクラスタでの平均値に比べて，ある閾値以上である場合とする。すなわち，

$$outlierness = \frac{d_i}{\bar{d}} \qquad \text{（式 2）}$$

が閾値 (δ) 以上の場合，アウトライヤーと判定する。

3.3.4 実験結果

実験では，人工的なデータと特許データからのアウトライヤー検出を試みた。手法としては，*k*NN (*k*-Nearest Neighbor)， LOF (Local Outlier Factor)，および本手法の 3 種類を用いた。

3.3.4.1 人工的なデータ

図 3 に示すような人工的なデータで 3 つのアウトライヤーが検出できるかどうかの実験を行った。*k*NN の場合，o1 と o3 はアウトライヤーとして検出できたが，o2 を検出することはできなかった。LOF では，もともとこのような場合でも検出できる方法として開発されたので，**図 4** に示すように，この 3 つの LOF 値だけが突出しており，正しく検出できた。

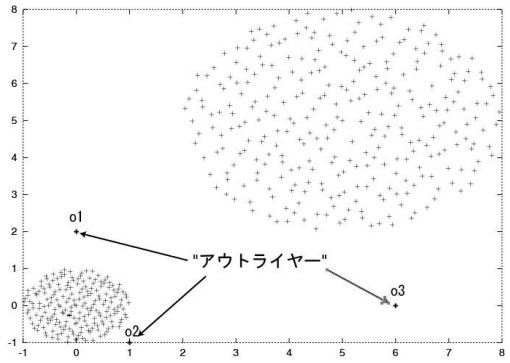

図 3　人工的なデータ，o1, o2, o3 がアウトライヤー

ただし，**図 4** の高さ軸（Z 軸）が LOF の値を示す。アウトライヤー以外のデータでは，LOF の値は，ほぼ 1.0となる。

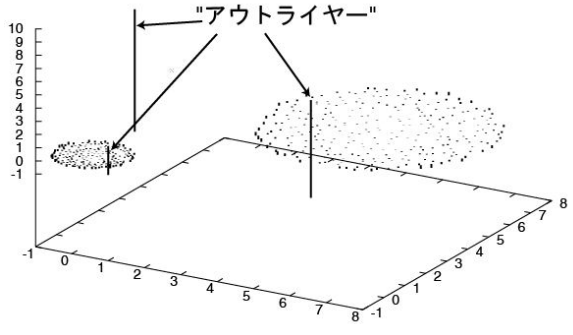

図 4　LOF での人工データからのアウトライヤー検出

本手法でクラスタリングを数回適用し，クラスタ平滑化を実行して，2つのクラスタを作成したところ，1つ目のクラスタ（クラスタ1）内のデータのクラスタ平均ベクトルからの平均距離は4.694，もうひとつのクラスタ（クラスタ2）の平均距離は0.540となった。これをすべてのデータ点に関して，（式2）を計算したところ，o1（クラスタ2）が 1.982，o2（クラスタ2）が 4.001，o3（クラスタ1）が 25.737 となった。$\delta = 3.0$ とすると，これら3点はいずれもアウトライヤーと検出され，他のデータ点でこの閾値を超えるものはなかった。

3.3.4.2 特許データ

ここでは，1998年の特許文書約33万件より，ランダムに選んだ4万文書を用いてアウトライヤー検出の実験を行った。特許文書には，IPC (International Patent Classification) と呼ばれる国際特許分類コードが付されているので，この値を手がかりに，同一の IPC（のサブクラス）を持つ特許が1件しかない特許，もしくは複数あっても内容が著しく他と異なる特許をアウトライヤーとした。特許データは，前処理として整形処理，品詞分解（茶筅）などを経て，最終的に単語 38,400 キーワードを抽出し，それぞれ，これらのキーワードと重みのペアからなる（38,400 次元の）ベクトルデータとした。

共クラスタリングの入力は，文書×単語の疎行列データ，キーワードデータ，および文書タイトルデータである。 これを複数回異なる粒度で実行し，クラスタ平滑化と階層化を「多粒度平滑化アルゴリズム」と「クラスタ階層構造化アルゴリズム」により実行し，**図2**に例示したようなクラスタ階層データを得た。

クラスタ粒度は，16, 32, 64, 128, 256, 512, 1024 の7レベルを用い，それぞれ5通りの異なる乱数の初期値で共クラスタリングを行い，クラスタ粒度階層構造を作成した。3つの手法，*k*NN（k=40），LOF（k=40），及び本手法において，アウトライヤー度の高いと判断された上位20位の結果は**表1**から**表3**までのとおりである。なお，実際にアウトライヤーであったもの（true positive）は，判定欄に○を，類似特許が2個以下のものは△を，それ以外は×（false positive）とした。

表1から**表3**までの結果より，各手法の特徴が出ていることがわかる。すなわち，*k*NN では，純粋に距離でアウトライヤー判定するので，正解判定されたものは，グローバルにユニークな特許となっている。しかし，kNN での正解率が100%とならないのは，データの次元が 38,400 次元と大きく，いわゆる「次元の呪い」のため，ある程度以上の距離のデータが密集していて，うまくアウトライヤーと判定できない現象が起こっているものと推察される。**表2**の LOF でアウトライヤーと判定されたものは，グローバルなものとローカルなものとが混在しており，たとえば「写真要素」に関する数件の特許は，G03C という IPC のサブクラスにおけるローカルなアウトライヤーと推察される。また，**表3**の本手法では，true positive として検出されるアウトライヤー

表1　*k*NN でのアウトライヤー検出結果上位20位

判定	筆頭 IPC	特許タイトル（抜粋）
△	A47J-43/24	洗米方法
○	A47G-1/08	額
○	C07D-201/04	ラウリルラクタムの製造方法
×	F16K-31/02	アクチュエータ
△	A01K-61/00	真珠およびその製造方法
○	B41C-1/055	印像を配分するための方法及び装置
○	G06F-17/60	観光案内情報提供方法
○	G01C-3/00	遠近距離認識装置
△	B42D-15/00	通訳ノート
○	E01C-19/48	ロードフィニシャ
○	C07C-69/96	トリメチロールアルカン
×	F16C-33/24	すべり軸受
○	H04N-9/47	SECAM信号処理回路
×	G07G-1/12	商品販売登録データ処理装置
×	G05B-19/05	プラント制御装置
○	A23L-1/317	フレッシュ風ソーセージ
×	E04G-17/075	型枠構築用の枠板緊結具
×	G06F-17/30	表示装置、表示方法のための記録媒体
×	A61B-6/03	断層撮影走査
△	G07B-1/00	証明書自動交付機

表 2　LOF でのアウトライヤー検出結果上位20位

判定	筆頭 IPC	特許タイトル（抜粋）
○	A61K-47/12	乳剤用保存剤および乳剤
○	G03C-5/29	現像活性化剤水溶液
○	C07F-9/72	モノアルキルアルシン
×	G03B-27/32	電力供給制御装置
×	G03C-5/26	ハロゲン化銀写真感光用固体処理剤
△	G03C-1/76	イメージング用支持体
×	G03C-7/407	発色現像液
○	G03G-21/00	純正交換部品消耗状態識別装置
○	G03C-5/31	写真浴
△	G03C-1/85	画像形成要素
×	A01N-59/16	防藻剤
△	G03C-1/79	積層基体及びそれを含む写真要素
○	G03C-1/81	反射写真感光材料
△	G03C-8/52	写真要素
△	G03G-15/11	液体現像剤の濃度測定装置
×	G03G-15/20	熱電温度制御を行う融解装置
○	A45C-5/12	パソコン用の鞄
○	A61C-15/02	つまようじ
△	G03G-15/11	液体濃度検出装置
×	G03D-3/00	ハロゲン化銀写真感光材用自動現像機

表 3　本手法でのアウトライヤー検出結果上位20位

判定	筆頭 IPC	特許タイトル（抜粋）
△	E04F-11/18	自在手摺り
△	E04C-3/10	構造材
○	G06F-17/60	死亡保険の効率的設計方法
×	G06F-13/00	リモートI／Oシステム
△	A61K-35/74	血糖降下剤
△	C09K-3/00	微粉体の低発塵化処理方法
○	B26B-21/44	石鹸付き折りたたみカミソリ
×	E05D-15/06	左右兼用引き戸式建具
△	G06F-17/60	生産計画システム
△	E03F-3/04	分割式カルバート
○	A23L-1/238	だし割り醤油の製造方法
△	B05C-21/004	塗工設備
△	G06F-17/6	訪問看護スケジューリングシステム
△	A47C-7/54	椅子の肘掛け装置
○	A47G-25/90	ストッキング着用補助具
○	B26B-13/22	目打つき握小鋏
△	B25J-17/02	コンプライアンス装置
△	G04B-19/22	電力制御装置
△	A47J-43/24	ざるを用いる洗米方法
○	G05B-13/02	PID調節計

は上位20件の中には少なかったが，数件のデータからなる△印のついた数件の類似データからなるアウトライヤーは多く検出されている。これは，クラスタにはいたらないが，非常に少数の類似特許からなるデータ集合がクラスタリング処理からは漏れたためと推察される。

上位20件までのアウトライヤー検出結果において○を 1.0，△ を0.5，×を 0.0 として，検出率を計算すると，*k*NN と LOF はいずれも55%で，本手法では60%となる。但し，いずれの方法においても共通に検出された正解アウトライヤーはなく，どの手法がベストであるということはいえない。

3.4 結果に基づく今後の研究活動計画

クラスタ粒度階層構造を利用した，大規模データからのアウトライヤー検出に対する手法を述べ，既知技術として *k*NN と LOF との比較実験を行った。データとして人工データの他，38,400 次元という高次元の特許データを利用したアウトライヤー検出を試みた。今後，この分野に要求されることとして，将来類似研究を行う研究者のために，高次元データ（10,000 次元以上）での大規模データセットおよび，その中での「正解となる」アウトライヤー文書をベンチマーク・データコーパスとして提供していく必要があると感じた。

クラスタ粒度に関しては，今回の実験では，2 のべき乗で変化させたが，そのような粒度で階層構造を作成するのが最良であるかは，改良の余地がある。

アウトライヤー検出技術の応用分野では今後は，時系列データからのアウトライヤー検出，また Web のような大きな構造体からのアウトライヤーコミュニティの発見，スケーラビリティ向上のための工夫などを手がけて生きたい。

参照文献

[1] 山西健司，“データ・テキストマイニングの最新動向—外れ値検出と評判分析を例に—,”応用数理, Vol. 12，No. 4 , pp. 241-356, December, 2002.

[2] C.C. Aggarwal and P.S. Yu, “Outlier Detection for High Dimensional Data”, *Proc. ACM SIGMOD 2001,* May, Santa Barbara, pp. 37-46, 2001.

[3] R. K. Ando and L. Lee, “Iterative Residual Rescaling: An Analysis and Generalization of LSI”, *Proc. ACM SIGIR 2001*, September, New Orleans, pp. 154-162, 2001.

[4] F. Angiulli and C. Pizzuti, “Outlier Mining in Large High-Dimensional Data Sets”, *IEEE Transactions on Knowledge and Data Engineering*, Vol. 17, No. 2, pp. 2030-215, 2005.

[5] V. Barnett and T. Lewis, *Outliers in Statistical Data*, John Wiley, 3rd edition, 1994.

[6] S. D. Bay and M. Schwabacher, “Mining Distance-Based Outliers in Near Linear Time with Randomization and a Simple Pruning Rule”, *Proc. ACM SIGKDD03*, August, Washington, pp. 29-38, 2003.

[7] M. M. Breunig et al. “LOF: Identifying density-Based Local Outliers”, *Proc. ACM SIGMOD 2000*, pp. 93-104, 2000.

[8] Inderjit S. Dhillon, S. Mallela, D. S. Modha, “Information-Theoretic Co-clustering”, *Proc. SIGKDD '03*, pp. 89-98, 2003.

[9] Inderjit S. Dhillon and Yuqiang Guan, “Information Theoretic Clustering of Space Co-Occurrence Data,” *Proc IEEE ICDM'03*, Melbourne, Florida, USA, pp.517-520, November 2003.

[10] Z. He, Z. Zu, and S. Deng, “Discovering cluster-based local outliers”, *Pattern Recognition Letters*, Vol. 24, pp. 1641-1650, 2003.

[11] G. Kollios, et al, “Efficient Biased Sampling for Approximate Clustering and Outlier Detection in Large Data Sets”, *IEEE Transactions on Knowledge and Data Engineering,* Vol. 15, No. 5, pp. 1170-1184, 2003.

[12] E. M. Knorr and R.T. Ng, “Algorithms for Mining Distance-Based Outliers in Large Datasets”, *Proc. VLDB,* pp. 393-403, New York, 1998.

[13] E. M. Knorr, *Outliers and Data Mining*, Ph.D. Dissertation, Department of Computer Science, The University of British Columbia, April, 2002.

[14] M. Kobayashi, M. Aono, et al, “Matrix computations for information retrieval and major and outlier cluster detection”, *Journal of Computational and Applied mathematics*, Vol. 149, pp. 119-129, 2002.

[15] S. Papadimitriou, H. Kitagawa, P.B. Gibbons, and C. Faloutsos, “LOCI: Fast Outlier Detection Using the Local Correlation Integral”, *Proc. International Conference on Data Engineering (ICDE)* 03, pp. 315-326, 2003.

[16] S. Ramaswamy, R. Rastogi, and K. Shim, “Efficient Algorithms for Mining Outliers from Large Data Sets”, *Proc. ACM SIGMOD 2000*, pp. 427-438, 2000.

[17] K. Yamanishi et al, “On-line Unsupervised Outlier Detection Using Finite Mixtures with Discounting Learning Algorithms”, *Proc. KDD2000*, pp. 320-324, 2000.

[18] K. Yamanishi and J. Takeuchi, “Discovering Outlier Filtering Rules from Unlabeled Data”, *Proc. ACM SIGKDD01*, San Francisco, pp. 389-394, 2001.

[19] K. Yamanishi et al, “On-Line Unsupervised Outlier Detection Using Finite Mixtures with Discounting Learning Algorithms”, *Data Mining and Knowledge Discovery*, Vol. 8, pp. 273-300, Elsevier, 2004.

[20] D. Yu, et al, “*FindOut*: Finding Outliers in Very Large Datasets”, *Knowledge and Information Systems*, Vol. 4, pp. 387-412, Springer, 2002.

[21] C. Zhu, H. Kitagawa, S. Papadimitriou, and C. Faloutsos, “OBE: Outlier by Example”, *PAKDD 2004*, LNAI 3056, pp. 222-234, 2004.

〈発 表 資 料〉

題　　名	掲 載 誌 ・ 学 会 名 等	発 表 年 月
クラスタ粒度階層構造を用いたアウトライヤー文書の検出方法	DBWS2005，電子情報通信学会，情報処理学会，日本データベース学会共催	2005年7月
Detecting Outlier Documents Using a Hierarchy of Clusters	IEEE ICDM05 投稿中，IEEE	2005年11月
A Method for Query Expansion by Using a Hierarchy of Clusters	AIRS2005 採録決定, Kist, Springer, ACM 他共催	2005年10月