2005－DBS－137（Ｉ）（1）
2005／7／13

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手法

青野 雅樹†

豊橋技術科学大学情報工学系 〒441-8580 愛知県豊橋市天伯町雲雀ヶ丘 1-1

E-mail: †aono@ics.tut.ac.jp

あらまし 大規模な文書から、稀少な分布を有する文書（アウトライヤー文書）を検出する手法を述べる．インターネットの普及にともない、文書データ、たとえば文献データ、特許データ、ニュースデータなどを比較的容易に収集できるようになった．これらのデータは年々増加する傾向にある．本報告では，文書データモデルとしてはベクトル空間モデルを用い，その後、前処理として「共クラスタリング」(Co-clustering)を複数回，異なるクラスタ粒度で実行する．これより「クラスタ粒度階層構造」を構築し、このクラスタ粒度階層構造を用いてアウトライヤーを検出する手法を述べる．同時に従来法との比較実験をあわせて報告する．

キーワード アウトライヤー，共クラスタリング，クラスタ階層構造

A Method for Detecting Outlier Documents Using a Hierarchy of Clusters

Masaki AONO†

Information of Computer Sciences Department, Toyohashi University of Technology

1-1 Hibarigaoka, Tempaku-cho, Toyohashi-shi, Aichi-ken, 441-8580 Japan

E-mail: †aono@ics.tut.ac.jp

Abstract Outlier detection is an important application in data mining community. We present an algorithm to generate a hierarchy of clusters for detecting “outlier” documents using the hierarchy. We also describe a comparative study of our proposed method with previously known methods.

Keyword Outlier detection, Co-clustering, Cluster hierarchy

1. はじめに

大規模なデータから、特異なデータ（アウトライヤー）を検出する研究は、統計学ではかなり古くから研究されてきた．これは、統計量の種々の解析において、ノイズに分類されるアウトライヤーの存在により、解析そのものに支障が生じるため、アウトライヤーを事前に見出し、これを削除した統計データで全体の傾向分析することで、より正確な知見が得られるためである．統計学では、このようにアウトライヤーを検出し、これを除去することが目的であった．

一方、Web ページ文書や Web 上のブログなどの書き込み、あるいは企業内・組織内の文書データベースやネットワークのアクセスログなどのデータに関しては、アウトライヤーを検出することは、それ自体が重要な研究対象となってきている．これは、ネットワークへの不正侵入や非常に希少で重要な文書、新製品などの発表によるニュース記事がもたらす新規ビジネス計画などの応用があり、それぞれ対象とするデータからアウトライヤーとして検出できるためである．

本報告では、日本語特許データを対象とし，ベクトル空間モデルで特許文書をモデル化し、前処理として「共クラスタリング」(co-clustering)を複数回，異なる「クラスタ粒度」で適用する．得られたデータから「クラスタ粒度階層構造」を作成し、このデータ構造を利用したアウトライヤー検出方法に関して述べる．クラスタ粒度に関しては，共クラスタリングで出力されるクラスタ数を 2 のべき乗（たとえば16, 32, 64, 128, ･･･）で変更しながら実行した．同時に、従来のアウトライヤー検出方法との比較実験を行ったのであわせて報告する．

2. アウトライヤー検出方法の関連技術

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 1998 年以降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のように分類できる．

（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

2.1. 距離ベースのアウトライヤー検出

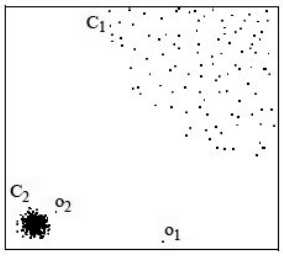
この範疇に含まれる研究としては、Knorr ら[12,13]により報告されている．Knorr らは、距離ベースのアウトライヤーを *DB(p,D)-outlier* と定義した．ただし、*p* はデータ集合の全体のうちの割合を表し、*D* はあるオブジェクト（データ点）からの距離を表す．すなわち、あるオブジェクトが *DB(p,D)-outlier* に属するとは、データの数の割合が少なくとも *p* 以上で、そのオブジェクトからの距離が *D* 以上はなれたデータのことをいう．データ点が正規分布する場合、アウトライヤーは 3σ（σ は標準偏差）以上はなれたデータに相当すると論じている．

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．

2.2. 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、下図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．



2.3. 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、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、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．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．

2.4. 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、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．このため、*k* 次元の立方体格子を考え、各立方体での「疎率」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．

2.5. 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．

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．

2.6. 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(*k*N)(*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．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].

2.7. 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. 提案手法

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、この点を考慮したクラスタリングに基づくアウトライヤー検出の先行研究は、筆者らの知る限りない．

なお、ここでの実験では、クラスタリング手法として、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．

3.1. 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つの場合である．

① 生成するクラスタの数を不適切(多すぎ，もしくは少なすぎ)に選択した場合

② クラスタ初期化における乱数の値により，生成されるクラスタの品質が著しく変化する場合

これらの代表的なクラスタリング誤使用を避けるため，我々は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$16, 32, 64, 128, \cdots$）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，アウトライヤー候補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,潜在的なアウトライヤーとして、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．

3.2. 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 $M_1, M_2, ..., M_k$ 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 $\mathbb{D}^r = \{\mathbf{D}_1^r, \mathbf{D}_2^r, ..., \mathbf{D}_M^r\} (r = 1, ..., R)$ と表現する．

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

[step3] 図１のクラスタ平滑化アルゴリズムを実行する．ただし，$\mathbf{K(j)}$ は，r 回目のクラスタリングで得られたクラスタ $\mathbf{D}_j^i$ 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．図１の７行目の Similarity 関数は，クラスタベクトル $\mathbf{H}$ と個々のデータベクトル $\mathbf{D}$ 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 $\mathbf{D}$ に基づき方向修正を（式１）で行う．

$$\mathbf{H}_i = \sum_i (\mathbf{h}_i + \lambda \mathbf{d}_i)$$

$$d_i = \sum_{j=1}^{D(i)} \alpha_j w_j \qquad \text{（式１）}$$

$$\mathbf{H}_i = \frac{\mathbf{H}_i}{\|\mathbf{H}_i\|}$$

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．

Algorithm ClusterSmoothing($\mathbb{H}, R, M$)

1: **for** $r = 2$ **to** R
2: **for** $j = 1$ **to** M
3: $\mathbf{D}_j^r = \{\mathbf{d}_{j,1}^r, \mathbf{d}_{j,2}^r, ..., \mathbf{d}_{j,K(j)}^r\}$;
4: **for** $i = 1$ **to** $|\mathbb{H}|$
6: $\mathbf{H} = \mathbf{H}_i$; $\mathbf{D} = \mathbf{D}_j^r$;
7: **if** (Similarity($\mathbf{H}, \mathbf{D}$) > δ) **then**
8: $\mathbf{H}_i$ = Average($\mathbf{H}, \mathbf{D}$);
9: **else** /* 追加 */
10: $\mathbb{H} = \mathbb{H} \cup \mathbf{D}$;
11: **endif**
13: **end for**
14: **end for**
15: **end for**

図１．クラスタ平滑化アルゴリズム

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．

「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 $\mathbb{H}_i$ と $\mathbb{H}_{i+1}$ 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，$\mathbb{H}_i$ の中のクラスタ $\mathbf{D}_i$ と $\mathbb{H}_{i+1}$ 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 $\mathbb{H}_k$ まで繰り返す．

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．

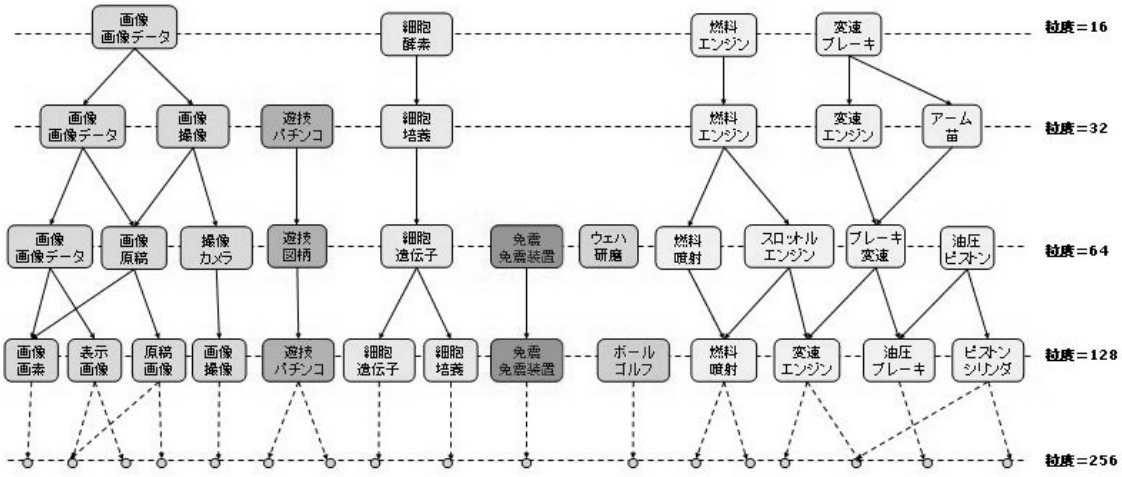


図２．異粒度クラスタリングと平滑化後，生成されたクラスタ粒度階層構造の一部.

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（$\overline{d}$ 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、

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

が閾値（δ）以上の場合、アウトライヤーと判定する.

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.

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.

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.

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.

4.2. 特許データ

ここでは、NTCIR-3[22]に含まれる 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（茶筌）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、キーワードと重みのペアからなる 38,400 次元のベクトルとした.

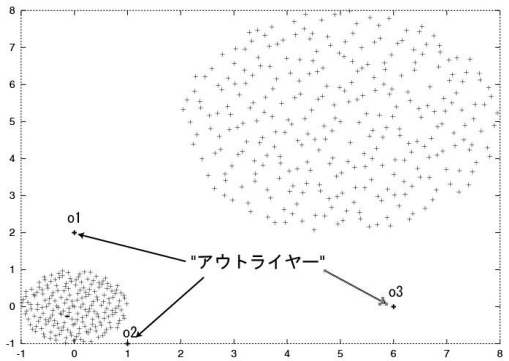


図 3. 人工的なデータ、o1, o2, o3 がアウトライヤー

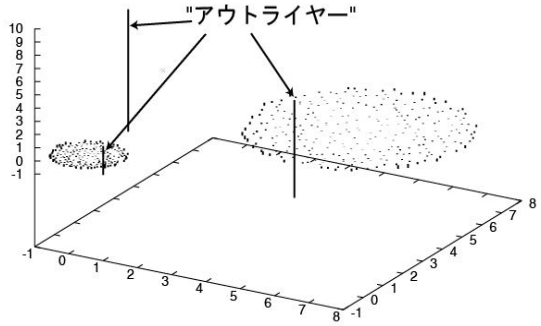


図 4. LOF での人工データからのアウトライヤー検出

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. *k*NN での検出結果上位 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
○	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用自動現像機

表 3. 本手法での検出結果上位 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

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、*k*NN 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．

計算時間としては、LOF が最もはやく、本手法がもっともおそかった．本手法を用いて表３のデータを得るまでに、クラスタリング時間を含め、Pentium 4 (3GHz) PC で約 8 時間要した．

5. おわりに

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．特許データでは、38,400 次元の高次元でのアウトライヤー検出を試みた．

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，今後の課題である．

今後は，時系列データからのアウトライヤー検出実験や手法のスケーラビリティ向上のための工夫などを行う予定である．

謝辞

本研究は、電気通信普及財団の援助を受けて行いました．

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 Trans. Knowledge and Data Engineering,* Vol. 17, No. 2, pp. 203-215, 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. KDD2003*, 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 Trans. 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. KDD2001*, 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.

[22] NTCIR (NII-NACSIS Test Collection for IR Systems), http://research.nii.ac.jp/ntcir/