

# Ajax マップ技術と MIGSOM を用いた 大量文書のクラスタリング・可視化に関する研究

中山 浩太郎<sup>†1</sup>

Google Maps などに代表される Ajax マップ技術は、大局的情報から局所的情報にシームレスに移動できるインタフェースを提供することで、ユーザエクスペリエンスを大きく向上させた。本論文では、大量文書をクラスタリングするアルゴリズム MIGSOM を利用し、大量データをクラスタリングし、Ajax マップ技術によって大局的情報と局所的情報をシームレスに可視化する手法について述べる。

## A Huge Scale Document Visualization Method based on Ajax Map Technique with MIGSOM Clustering

KOTARO NAKAYAMA<sup>†1</sup>

### 1. はじめに

インターネットの普及により、人間が日常生活の中で処理する情報の量は爆発的に増加した。特に、Wiki や SNS などの Web ベースのシステムでは、ユーザが分散的・自律的に情報を公開するため、膨大な情報が複雑なネットワーク構造上に配置される。この結果、誰も全体の大局的な情報を把握できなくなってしまう。これは、企業や組織など比較的クローズドな環境での情報共有でも同様の問題が起きている。

このような問題を解決するために有効なアプローチとして、情報のクラスタリングや分類が挙げられる。これらの手法は大量の情報をいくつかの単位にまとめることで、トップダウン

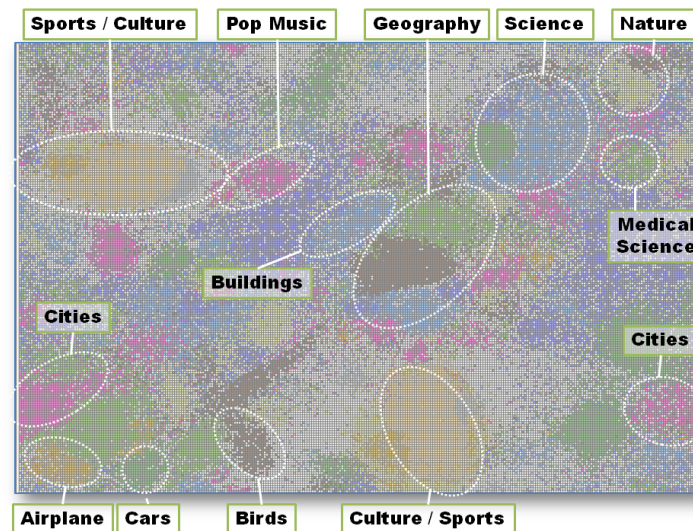


図 1 Discovered Clusters

ンに情報を俯瞰する方法を提供する。情報のクラスタリングの一手法として興味深いのは、自己組織化マップ (SOM) を利用する手法である。自己組織化マップは、多次元データを低次元データに写像する教師無し学習手法であり、多様なアプリケーションに適用されている。その中でも特に、情報の関連性を可視化し、大局的に情報を俯瞰する用途に良く利用される。

SOM の有効性は過去の研究で十分に証明されてきたが、大量の文書集合など、大規模なデータに対して SOM を適用する場合、精度とスケーラビリティの両立が問題であった。一般的に、大量の文書を SOM に適用する場合、LSI やランダムマッピングなどの次元圧縮手法を利用するが、次元圧縮はクラスタリングの精度低下を招く要因となる。そのため、この問題に対して、筆者は過去の研究において、神経細胞移動 (Neuronal Migration) にヒントを得た自己組織化アルゴリズム「MIGSOM」<sup>13),14)</sup> を提案した。

脳などの中枢神経系は、大量の情報の分類やクラスタリングなどに適した効率的・合理的な構造を持っているが、その構築方法などの詳しい仕組みは長い間研究者の興味の対象であった。特に中枢神経系の形成プロセスはこの 10 年程度で急速に解明されてきたが、その

<sup>†1</sup> 東京大学  
The University of Tokyo

中でも神経細胞移動 (Neuronal Migration) が重要な役割を果たすことが判明したことは脳科学における大きな進歩の一つであった。神経細胞移動とは、個々の神経細胞が自分に適合した場所を探しながら移動し、全体として効率的な情報ネットワークを構築する現象である。MIGSOMはこの神経細胞移動にヒントを得て作成した自己組織化アルゴリズムであり、精度とスケーラビリティの両立を目指した手法である。MIGSOMによって解析したWikipediaのデータを図1に示す。この可視化では、Wikipediaの全ページをページ内に出現するリンクによってベクトル化し、内容の近いページが近くに配置されるように学習している。

本稿では、MIGSOMの学習結果を可視化するために開発した「MIGSOM VIS」(図2について報告する。MIGSOM VISは、MIGSOMで解析したクラスタリングの結果をAjaxマップ技術を利用して可視化するシステムである。

MIGSOM VISでは、Ajax技術を用いることで、クラスタリングの結果を大局的な情報から局所的な情報までシームレスに解析することが可能なインタフェースを提供している。可視化に際しては、事前にk-meansクラスタリング手法によって予め文書を10のクラスタに分類し、それぞれクラスタに応じた色を定め、ノードを色づけしている。その際、いくつか調整可能なパラメータが存在する。例えば、1)クラスタの色づけ方法、2)色づけのシームレス化、3)色づけに利用するクラスタの数、などが該当する。本論文では、MIGSOMとVisualizerの概要および、これらのパラメータが与える可視化への影響について考察した結果を報告する。

## 2. 関連研究

### 2.1 KohonenのSOM

Kohonenの自己組織化マップ(SOM: Self Organizing Map)<sup>4)</sup>は、多次元データを低次元データに写像するために利用される教師無し学習手法である。KohonenのSOMは多くのSOM研究の原型であり、今日に至るまで様々なアルゴリズム改良が施されたが、基礎的な手順はほぼ同様である。以下にKohonenのSOMのアルゴリズムを示す。

- (1) マップ上の全ノードをランダムなベクトルで初期化。
- (2) 全レコードセット  $X$  からランダムに一つの入力レコード  $x$  を選択。
- (3)  $x$  をマップ上の全ノードと比較し、類似度の最も高い勝者ノード  $w$  を見つける
- (4)  $w$  の距離  $d$  内の周辺ノードのベクトルを、勝者ノード  $w$  に基づいて変更する。この

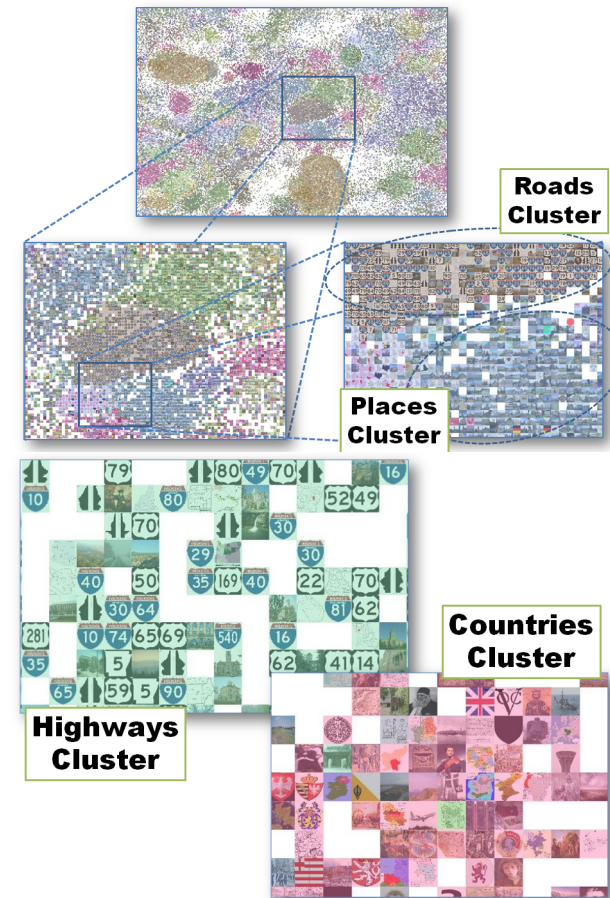


図2 Interactive Zooming Interface

時、 $w$  に近いノードほど強い影響を受ける。

- (5) 距離  $d$  の値を少し小さくする。
- (6) ステップ2に戻る。

ここで、ステップ 2 で選択される入力レコード  $x$  は、一回の反復では一つだけであるため、入力データ数が多くなっても、使用するメインメモリの容量は常に一定であり、スケーラビリティを確保できる。しかし、このスケーラビリティが保証されるためには、二つの前提条件がある。一つ目の前提条件は、扱うデータが小次元の密データであることである。SOM は、勝者ノードを発見する際と、周辺ノードのベクトルを修正する際に大量のベクトル比較とベクトル修正を必要とする。扱うデータが密ベクトルの場合は問題ないが、疎ベクトル集合が入力として与えられた場合、ステップ 4 で周辺ノードのベクトルを修正する際にベクトル情報が反復の度に大きくなるという問題がある。

二つ目の前提条件は、マップのサイズが十分に小さいことである。Kohonen の SOM ではマップサイズを自由に指定できるため、必ずしも大規模なデータに対して大きなマップが必要ではない。極論だが、数百万件の入力レコードに対して、2 行 x 2 列 (4 ノード) のマップを利用して学習することも可能である。しかし、文書をマップ上に配置して解析するためのドキュメントマップを作成するには、データ数に応じた十分な大きさのマップが必要となる<sup>5)</sup>。これは、大規模データを扱う場合、マップ領域が十分に大きくなければ一つのノードに多くのレコードが重複してマッピングされてしまう上に、十分な解析精度が得られないためである。

## 2.2 SOM の大規模化

文書のクラスタリングなどのアプリケーションを考えた場合、疎行列データへの SOM の適用は、重要な技術的課題であるため、各種の研究が行われてきた。従来研究の方向性としては、大きく分けて分散並列処理を利用した方法と次元圧縮を利用する方法の二つに分類される。

分散並列処理を利用する方法としては、Lawrence らの研究<sup>7)</sup> が有名である。Lawrence らは、MPI を利用した分散並列処理環境で、大規模疎行列の解析ベクトルを SOM で解析する方法を提案している。しかし、IBM SP2 の PC クラスタを利用するなど、特殊な解析環境が必要である。

一方、次元圧縮を使う方法で最も有名な研究の一つが、Lagus らの WebSOM<sup>3),6)</sup> である。WebSOM は、文書単語行列などの大規模疎行列を、LSI (Latent Semantic Indexing)<sup>2)</sup> やランダムマッピングなどの手法を用いて次元圧縮し、小規模密ベクトルに変換してから解析する方法を提案している。Scalable Self-Organizing Map (SSOM)<sup>11)</sup> は、疎行列データの 0 要素をデータ領域から除外し、ベクトル比較などで要素のある箇所だけ利用することで、高速化する手法を提案している。

WebSOM や SSOM 以外にも多くの研究において、大規模疎データに SOM を適用する場合は、なんらかの手法で次元圧縮手法を行うことが一般的であった。これらの研究によって次元圧縮の有効性は十分に証明されてきたが、SOM 自体を大規模な疎行列の解析に適用させる研究は少なかった。

## 3. MIGSOM

本節では MIGSOM のアルゴリズムを詳述する。アルゴリズムの説明に先立ち、MIGSOM の基礎モデルとして神経細胞移動について説明する。

### 3.1 神経細胞移動

脊髄や脳などから構成される中枢神経系は、情報の分析・判断・決定など、知的活動を担当する部位である。中枢神経系が持つ効率的・合理的な情報処理能力を解明することは、長い間多くの研究者の興味の対象であり、活発に研究が進められてきた。その中でも、神経細胞移動と呼ばれる現象が重要な役割を果たすことが判明している。

神経細胞 (ニューロン) は、様々な場所で生成されるのではなく、限られた場所で生成され、移動を繰り返して最適な場所へ定着する。このように、神経細胞が最適な場所を見つけるために移動する現象を神経細胞移動 (Neuronal Migration) と呼ぶ。神経細胞の移動方式としては、初期の中枢神経系によく見られる Translocation と後期によく見られる Locomotion の二つの方式が存在する<sup>8)</sup>。Translocation はニューロンが自身の軸索 (Axon) を利用して移動する方向を決定する方式であるのに対し、Locomotion はグリア細胞がニューロンをガイドすることで移動する方向を決定する。どちらも興味深く、効率的な計算モデルを実現する上で利用価値があると思われるが、今回の研究では、特に Translocation に注目をした。これは、Locomotion は Translocation の特殊形と見なすこともできるため、まずは Translocation をモデル化することで、将来的に Locomotion をモデルにしたアルゴリズムの構築に貢献できると考えたためである。

Translocation は、神経細胞が自身の軸索を様々な場所に伸ばし、移動すべき場所を見つけた後に、軸索を縮めて移動する方法である。Translocation の詳細な手順を図 3 に示す。

まず、ニューロンは生成された後、軸索を伸ばし始める。次に少しずつ軸索で周辺を探索をしながら、軸索を伸ばす方向を決める。そして、定着する場所を決めた後に、周辺ニューロンとの間にシナプス (結合部) を生成する。最後に軸索を縮めることで、ニューロンが移動する。

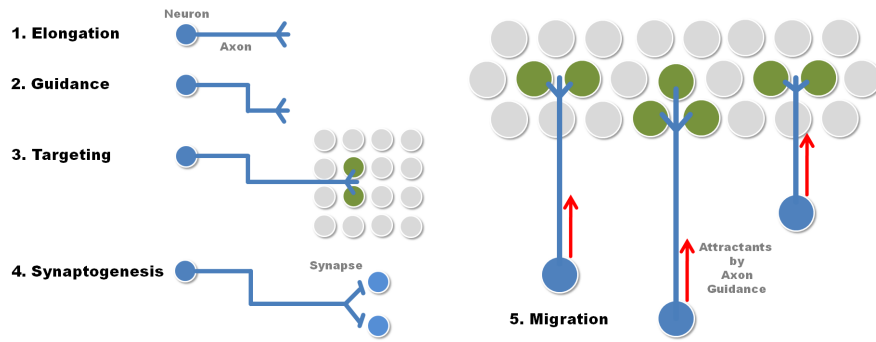


図 3 Translocation

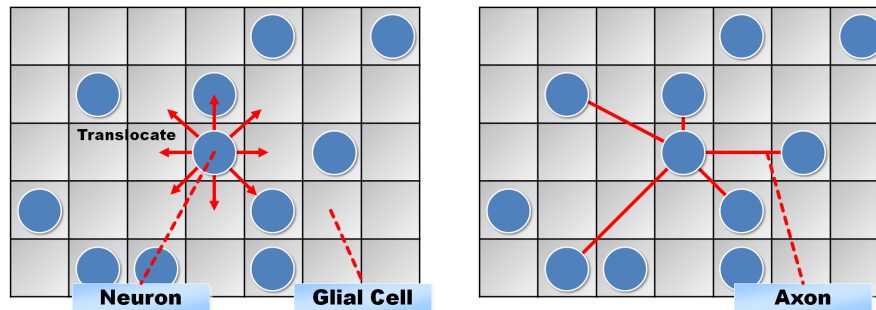


図 4 MIG-SOM basis

### 3.2 MIGSOM の基本動作

MIGSOM は、Kohonen の SOM と同様、反復処理による教師無し学習手法であるが、データの表現方法と学習対象の点で大きく異なる。Kohonen の SOM では、マップ上の各ノードが独自のベクトルを持ち、その値を変更（ベクトル修正）していくことで学習が行われる。この際、マップ上の一つのノードが必ずしも一つの入力レコードに対応するわけではない。これとは対照的に、MIGSOM では、マップ上のノードが入力レコードに対応する（図 4 上）。ノード上に配置された入力レコードをニューロンと呼び、ニューロンがマップ上を移動することでマップ全体の学習が行われる。また、入力レコードが割り当てられないノードには、ランダムに生成されたベクトルを持つグリア細胞が配置される。グリア細胞

は、ニューロンを埋める補助的なベクトルであり、ニューロンの移動をガイドする役割を果たす。ニューロンの移動方向は、ランダムに周辺に軸索を伸ばし（図 4 下）、自身のベクトルと類似しているノードが多く集まる方向を発見することで決定される。

MIGSOM では、まず全ての入力レコードをニューロンとしてマップ上のノードにランダムに配置する。次に、空いているノード上にグリア細胞としてランダムなベクトルを生成する。そして、次に示す処理を反復することで学習する。

---

#### Algorithm *train()* :

- 1 Randomly select  $g$  from  $G$
  - 2  $\vec{m}_g = \vec{0}$  #Initialize by null vector
  - 3  $N = GaussianSel(g)$
  - 4 **for each**  $n \in N$
  - 5      $dist = distance(n, g)$  #Distance in map
  - 6      $power = tanh(dist)$
  - 7      $\vec{m}_g = \vec{m}_g + U(n, g) \cdot power \cdot Sim(n, g)$
  - 8     if  $|\vec{m}_g| > t$
  - 9          $Migrate(g, \vec{m}_g)$
- 

- $G$ : マップ上の全ノード集合
- $\vec{m}_g$ :  $g$  の移動ベクトル
- $GaussianSel(g)$ : ノード  $g$  周辺のノードをガウシアン分布に従ってランダムに選択する関数
- $distance(n, g)$ : ノード  $n$  とノード  $g$  のユークリッド距離
- $tanh(dist)$ : 距離  $dist$  の逆正接
- $U(n, g)$ :  $g$  から  $n$  への単位ベクトル
- $Sim(n, g)$ :  $n$  と  $g$  が持つベクトルのコサイン類似度
- $Migrate(g, \vec{m}_g)$ : 移動ルーチン。もし  $|\vec{m}_g|$  ( $\vec{m}_g$  のノルム) が閾値  $t$  より大きい場合、 $g$  はステップ 9 で  $\vec{m}_g$  方向に移動 (Translocate)

本アルゴリズムでは、まずマップ上の全ノード集合  $G$  からランダムにノード  $g$  を一つ選択し、訓練用ニューロン（グリヤ細胞）として採用する。次に、 $g$  がどの方向に移動するかを示す  $\vec{m}_g$  の値をゼロベクトルで初期化する。そして、 $g$  周辺のノードからガウシアン分布に従ってランダムにノードを選択する。つまり、マップ上の距離が近いニューロン（やグリヤ細胞）ほど高確率に選択される。 $GaussianSel(g)$  は、ノード  $g$  周辺のノードをガウシアン分布に従って選択する関数である。

ランダムに選択された周辺ノード集合  $N$  の各ノード  $n$  に対し、 $g$  と  $n$  が持つベクトルの類似度とマップ上の距離を求める。そして、類似度の高いノードが存在した場合、類似度に応じてそのノードの方向へと移動ベクトル  $\vec{m}_g$  を修正する。最後に、移動ベクトルのノルム  $|\vec{m}_g|$  を求め、閾値以上であれば、移動する。この時、ニューロンが移動できる範囲は  $g$ （起点）の周辺 8 ノードのみとした。また、移動に伴い、移動先のニューロン（かグリヤ細胞）は  $g$  に移動する。つまり、移動とは厳密には移動先のニューロンと位置を入れ替えることである。

### 3.3 軸索キャッシュ

2 節で述べた通り、従来の SOM は疎行列を扱う際に大量のベクトル比較とベクトル修正が発生することが問題であった。この問題は、提案手法の MIGSOM も同様で、ステップ 7 において、周辺ノードのとの類似度を計算する箇所、大量のベクトル比較が必要となる。しかし、軸索をのぼす対象はガウシアン分布によって近いノードほど高確率で比較されるため、ステップ 9 でニューロンが隣接ノードへ移動したとしても、再び同じノードと比較される可能性が高い。つまり、ベクトル比較の「局所性」が存在する。そのため、ベクトル比較の結果をキャッシュしておくことで、処理の高速化が可能である。ベクトルの比較が再度必要になった場合、キャッシュの値を使うことで、ベクトルの比較に要する計算コストを大幅に削減できる。この仕組みは、直感的には、軸索がベクトル比較の値をキャッシュすると理解すると分かりやすい。そのため、この仕組みを「軸索キャッシュ」と呼ぶ。

ベクトル比較の結果をキャッシュする方法は、従来型の SOM には適用できない。これは、学習の過程でノードが持つベクトルの値が常に変動するため、比較の結果が一定ではないことに起因する。これに対し、MIGSOM ではニューロンやグリヤ細胞の持つベクトルの値は、学習の間修正されることが無い。そのため、ベクトル比較の値を保持することでベクトル比較が高速化できるのは MIGSOM の大きな特徴の一つである。

### 3.4 パフォーマンスに関する検証

MIGSOM のパフォーマンスを調査した結果、MIGSOM は特に大規模で疎なデータに対

して高い精度で解析できることがわかっている<sup>13),14)</sup>。実験では、生成されたマップ上で、相関の高いレコードがどれほど近くに配置されているかを数値化して精度とした。相関値では、MIGSOM のほうが高いスコアを達成できており、従来手法では、データ数が 100 の時は 0.5 程度の相関値があったものの、データサイズ（とそれに応じたマップサイズ）を増加させるとともに急激に値が下がり、データ数を 250 にした時点で 0.2~0.25 程度の相関値しか達成できなかった。一方、MIGSOM は収束に時間がかかるものの、データ数 100 の時に相関値 0.6 程度、データ数を 250 に増やしても 0.5 程度と、高い値を保持できていることがわかった。つまり、MIGSOM のほうが写像性能（ベクトルの類似度の高いレコード同士を近くに配置できる性能）が高く、データ数が増えても強い相関を保つと言える。

### 3.5 マップサイズに関する考察

マップサイズがメモリ使用量に及ぼす影響について評価実験を行った<sup>13),14)</sup> が、マップサイズを増加させると従来手法では急激にメモリ使用量が増えるのに対し、提案手法ではこのメモリ使用量の上昇を防ぐことができていることがわかった。これは、従来手法ではマップ上のノードが独自のベクトルを持つが、疎なデータを扱う場合、学習を繰り返すことでノードが持つベクトルが徐々に大規模な密データに近づき、疎データの圧縮ができなくなるためである。一方、MIGSOM は疎行列データを圧縮して保持する上に、ベクトル合成を一切行わないため、データ量が大きくなってメモリ使用量は線形に推移する。

## 4. MIGSOM VIS

本節では、MIGSOM の解析結果を可視化する MIGSOM VIS について説明する。MIGSOM VIS では、MIGSOM によって解析された結果に対して、ノード情報を表示し、クラスタリングの結果に応じて色づけする。クラスタリングのアルゴリズムには k-means クラスタリングの手法を利用した。実験では予め全てのレコードを k-means クラスタリングによって 10 のクラスタに分類しておく。クラスタ毎に特定の色を割り当て、そのクラスタに属するレコードはその色をノード情報にオーバーラップして表示する。

### 4.1 クラスタ境界のシームレス化

ここで、色づけにはクラスタ境界を明確にする方法と、クラスタの境界をシームレスに表示する方法が考えられる。まず、クラスタ境界を明確にする方法は、全レコードに対して、10 のクラスタのうち、最もベクトルの近いクラスタを特定し、その色を採用する方法である。この方法では、クラスタの領域が明確になり、大局的情報を閲覧している時に、どこにどのようなクラスタがあるのかが把握しやすくなるという利点がある。

一方、クラスタの境界をシームレスに表示する方法では、全てのクラスタに対して類似度を計算し、類似度に応じて色づけする方法である。この方法では、二つのクラスタに同程度類似しているようなレコードに対してこの結果、クラスタ同士の類似度などを視覚化できるという利点がある。

#### 4.2 Ajax マップ技術によるクラスタマップ

MIGSOM VIS の特徴の一つが、Ajax マップ技術を利用したクラスタリングの解析機能である。Ajax を利用することにより、大局的な情報を表示している時は非常に解像度の低い画像を利用し、詳細な部分をズームしていくことで、必要に応じて必要な画像の解像度を上げることが可能となる。この結果、大局的なクラスタリングの結果と、局所的なクラスタリングの結果をシームレスに移動しながら解析することが可能となった。

### 5. Wikipedia データへの適用

提案手法の実用性を示すために、実データとして英語版 Wikipedia<sup>\*1</sup>へ MIGSOM を適用し、可視化した事例を以下に紹介する。Wikipedia は Web ブラウザを利用して誰でも利用可能なオンライン百科事典であり、その規模や網羅性、実用性から、Web マイニングや自然言語処理、情報検索などの研究において基盤リソース（コーパス）として広く利用されている。Wikipedia の記事は、一つのエンティティ（概念）に対応しており、記事同士はハイパーリンクによって参照し合っている。Wikipedia は密なリンク構造を持っており、リンクの構造を解析することで記事（やエンティティ）の間の関係などを解析できることが過去の研究で示されている<sup>9)</sup>。Wikipedia のリンク情報を表現する隣接行列は典型的な大規模疎行列であり、本手法の有効性を検証する上で有効だと考えた。

本事例では、2009 年 4 月の英語版 Wikipedia（記事数 300 万程度）をデータセットとして利用した。まず、各記事を、記事内に出現するリンクでベクトル化し、300 万行 × 300 万列の隣接行列を作成した。次に、被リンク数が 10 件以下の記事など、ノイズデータを除外した後に、可視化で利用できる記事だけ（画像が含まれる記事だけ）抽出した。その結果、約 11 万 8 千件の記事が残った（つまり、11 万 8 千行 × 300 万列の隣接行列を作成）。

次に、サイズが 500 × 300、15 万ノードのマップを作成し、記事をランダムに配置した。その後、記事が割り当てられなかった 3 万 2 千個のノードにグリア細胞（ランダムに生成したベクトル）を配置し、MIGSOM で学習した。

次に、本解析結果を基に、各記事に含まれる画像を表示してズームして閲覧できるシステム「Wikipedia SOM Visualizer」を開発し、詳細に検証した結果を図 1 と図 2 に示す。Wikipedia SOM Visualizer は、各記事に含まれる画像をマップ上の記事の上に表示し、Ajax 技術を使ってインタラクティブな拡大縮小機能を実装することで、シームレスな分析を可能とした。本システムを利用して SOM の実行結果を解析すると、「文化・スポーツ」「科学」「自動車」などの明瞭なクラスタができていくのがわかる。次に、地理情報の箇所をズームしてみると、さらに「道」や「国」といったサブクラスタが存在することがわかる。また、意外な知見としては、「化学」に関するクラスタと、銃などの「兵器」に関するクラスタが密に交差しているということが判明した。境界領域を調べてみると、新型の火薬など新しい化学の技術を利用した兵器に関する情報が双方の中間ハブとしての役割を果たしているように見える。

SOM は、多様なアプリケーションへ適用されるアルゴリズムであるが、特に重要なアプリケーションの一つが、クラスタの分け方が不明瞭なデータセットに対して適用して可視化することである。これにより、クラスタの発見やクラスタ分けの指針を得ることができる。本実験により、このようなアプリケーションに対しても MIGSOM を適用できる可能性があると言える。

### 6. ま と め

本論文では、神経細胞移動モデルに基づく自己組織化マップ「MIGSOM」による可視化システム「MIGSOM VIS」を提案し、可視化に影響を与えるパラメータについて考察した。MIGSOM は、疎な大規模データに適用可能な SOM であり、実際に大規模疎行列データとして、Wikipedia のリンクデータに適用・可視化することで、実用性を示した。なお、Wikipedia データの可視化システムは、以下の URL にてアクセス可能である。

- Wikipedia SOM Visualization  
<http://sigwp.org/wikisom/>

### 7. 謝 辞

本研究の一部は、科学研究費補助金基盤研究 C(20500093)、科学研究費補助金基盤研究 B(21300032) の助成によるものである。ここに記して謝意を表す。

\*1 <http://wikipedia.org>

## 参 考 文 献

- 1) Beal, J.: Self-Managing Associative Memory for Dynamic Acquisition of Expertise in High-Level Domains, *Proc. of International Joint Conference on Artificial Intelligence (IJCAI)*, pp.998–1003 (2009).
- 2) Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K. and Harshman, R.: Indexing by latent semantic analysis, *Journal of the American Society for Information Science*, Vol.41, No.6, pp.391–407 (1990).
- 3) Kaski, S., Honkela, T., Lagus, K. and Kohonen, T.: WEBSOM - Self-organizing maps of document collections, *Neurocomputing*, Vol.21, No.1-3, pp.101–117 (1998).
- 4) Kohonen, T.: The self-organizing map, *Neurocomputing*, Vol.21, No.1-3, pp.1–6 (1998).
- 5) Kohonen, T.: Self Organization of a Massive Document Collection, *IEEE Transactions on Neural Networks*, Vol.11, No.3, pp.574–585 (2000).
- 6) Lagus, K., Kaski, S. and Kohonen, T.: Mining massive document collections by the WEBSOM method, *Information Science*, Vol.163, No.1-3, pp.135–156 (2004).
- 7) Lawrence, R.D., Almasi, G.S. and Rushmeier, H.E.: A Scalable Parallel Algorithm for Self-Organizing Maps with Applications to Sparse Data Mining Problems, *Data Mining and Knowledge Discovery*, Vol.3, No.2, pp.171–195 (1999).
- 8) Nadarajah, B., Brunstrom, J.E., Grutzendler, J., Wong, R.O. and Pearlman, A.L.: Two modes of radial migration in early development of the cerebral cortex., *Nature neuroscience*, Vol.4, No.2, pp.143–150 (online), DOI:10.1038/83967 (2001).
- 9) Nakayama, K., Hara, T. and Nishio, S.: Wikipedia Mining for An Association Web Thesaurus Construction, *Proc. of IEEE International Conference on Web Information Systems Engineering (WISE 2007)*, pp.322–334 (2007).
- 10) Rosenblatt, F.: The perceptron: a probabilistic model for information storage and organization in the brain, pp.89–114 (online), available from <http://portal.acm.org/citation.cfm?id=65669.104386> (1988).
- 11) Roussinov, D.G. and Chen, H.: A Scalable Self-organizing Map Algorithm for Textual Classification: A Neural Network Approach to Thesaurus Generation, *Communication Cognition and Artificial Intelligence, Spring*, Vol.15, pp.81–112 (1998).
- 12) Taniguchi, Y., Wakamiya, N. and Murata, M.: A traveling wave-based self-organizing communication mechanism for WSNs, *Proc. of International Conference on Embedded Networked Sensor Systems (SenSys)*, pp.399–400 (2007).
- 13) 中山浩太郎：Migsom:神経細胞移動モデルに基づく自己組織化マップ～大規模リンクドデータへの応用～, Web とデータベースに関するフォーラム (WebDB Forum 2010) (2010).
- 14) 中山浩太郎：神経細胞移動に着想を得た自己組織化マップによる wikipedia リンクデー

タの可視化, 日本データベース学会論文誌, Vol.9, No.3, pp.19–24 (2011).