

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

A Visualization Method for Wikipedia's Link Data with Self Organizing Map Algorithm Inspired by Neuronal Migration

中山 浩太郎*

Kotaro NAKAYAMA

SOM(自己組織化マップ)は、教師無し学習アルゴリズムの代表例であり、高次元データを低次元データ(通常は2次元マップ)に写像できる。SOMは汎用性とスケーラビリティの高さから可視化やクラスタリング、連想記憶など幅広いアプリケーションで利用されている。しかし、その高いスケーラビリティが発揮できるのは十分に低次元な密データを扱う場合に限定されており、単語文書行列やリンクデータを表現する隣接行列のような大規模疎行列データの扱いが技術的な課題であった。本論文では、神経細胞移動(Neuronal Migration)という、中枢神経系の効率的な構築に関する現象をモデルにした自己組織化マップアルゴリズム「MIGSOM」を提案すると共に、大規模な実データ(Wikipediaのリンクデータ)に適用・可視化することで、その実用性を示す。

SOM (Self Organizing Map) is one of the most popular unsupervised machine learning algorithms that maps high-dimensional vectors into low-dimensional data (usually 2-dimensional map). SOM is widely known as a "scalable" algorithm because of the capability of handling a significant number of records. However, it works out only when the length of vector data is enough small and dense. In other words, applying SOM for sparse high-dimensional data was a technical issue.

In this paper, we introduce MIGSOM, an yet another SOM algorithm inspired by "Neuronal Migration." We describe the detailed algorithm, implementation and the capability in this paper. Additionally, we apply the algorithm to actual huge scale linked data (Wikipedia's link data) in order to show the practicality.

1. はじめに

生物は複雑な動作や知的処理を、計算機に比べると驚くほど小規模な回路で実現し、小さなエネルギーで動作させることができる。このような合理性・効率性は何処から来るのであろうか。世界初の汎用計算機が開発されてから半世紀以上経ち、半導体技術の発展により、計算スピードという面ではコンピュータは目覚ましい進化を遂げた。しかし、その一方で生物が持つ情報処理能力の大局部性・ロバスト性・自律性・効率性などの面では、未だ遠く及ばないのが現実である。このような背景から、生物の情報処理の仕組みを解析し、合

* 会員 東京大学知の構造化センター
nakayama@cks.u-tokyo.ac.jp

理的・効率的な計算モデルを実現する「生体模倣技術（バイオミミクリー: Biomimicry）」に関する研究が注目を集めている。この研究分野では、生体システムが持つ合理的な機能がいかにして実現されているか分析し、その仕組みを模倣することで新しい技術を実現することを目的としている。生物の情報通信の仕組み（特に脳などの中枢神経系）を解明・模倣する取り組みも行われており、最近では螢の発光の同期現象にヒントを得た情報通信モデル[12]などの研究が行われている。また、小脳の情報選択の働きを模したローゼンプラッドのペーセプトロン[10]も生体模倣技術の一例と言える。

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。その中でも、中枢神経系の形成プロセスは特に注力されてきた分野であり、この10年程度で急速に解明されてきた。その中でも特に、神経細胞移動(Neuronal Migration)が重要な役割を果たすことが判明している。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。

本研究では、神経細胞移動をモデル化した自己組織化マップ(SOM)アルゴリズムの「MIGSOM」を提案する。SOMは、教師無し学習アルゴリズムの代表例であり、高次元データを低次元データ(通常は2次元マップ)に写像することが可能である。SOMは、その有用性と実用性から、データの可視化や文書分類、画像分類、連想記憶[1]など、様々なアプリケーションで利用されている。SOMは、一般的に「スケーラビリティの高い」アルゴリズムとして知られている。これは、SOMは反復アルゴリズム(Iterative Algorithm)の一種だが、一回のステップで利用するデータが一つの入力データに限られ、すべてのデータを一度に利用する必要が無いためである。この特性により、SOMは扱うデータ数が膨大になっても解析を行うことができるという特徴を持つ。しかし、このSOMの持つスケーラビリティは、大量のデータを扱うことが可能という意味している一方で、大規模な疎データ(疎行列データ)の扱いは、依然技術的課題であった。特に、Web文書のクラスタリングなど、文書内に出現する単語やリンクを利用して文書の特徴ベクトルとするようなアプリケーションでは、データは大規模な疎行列になるため、通常の方法ではSOMを適用することはできない。

MIGSOMは、従来型のSOMと同様に反復処理型の教師無し学習アルゴリズムであるが、その構造と処理プロセスは従来手法と大きく異なる。特に、疎な入力データ集合に対して高いパフォーマンスを実現できるのが特徴である。本論文では、アルゴリズムを詳述した後に、実験によってアルゴリズムの優位性を示す。さらに、大規模な実データ(Wikipediaのリンクデータ)に適用し、データを可視化することで、その実用性を示す。

2. 関連研究

2.1 Kohonen の SOM

Kohonenの自己組織化マップ(SOM: Self Organizing Map)[4]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x2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[5]。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[7]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[3, 6]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)[2]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)[11]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良く見られるTranslocationと後期によく見られるLocomotionの二つの方式が存在する[8]。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に注目した。これは、LocomotionはTranslocationの特殊形と見なすことができるため、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、移動する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1に示す。

まず、ニューロンは生成された後、軸索を延ばし始める。次に少しずつ軸索で周辺を探索をしながら、軸索を延ばす方向を決める。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。

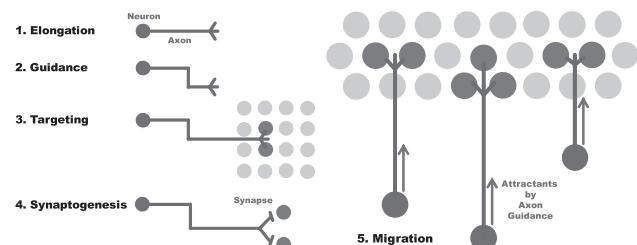


図1 神経細胞移動(Translocation)

Fig.1 Neuronal Migration(Translocation)

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2左）。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、ニューロンの間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2右），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。

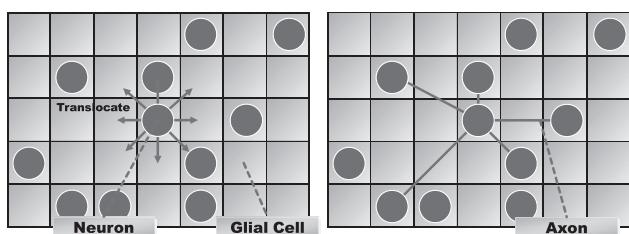


図2 MIGSOM の基本動作

Fig.2 MIG-SOM basis

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 $\vec{m}_g = \vec{O}$ #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（グリア細胞）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（やグリア細胞）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（かグリア細胞）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、周辺ノードとの類似度を計算する箇所で、大量のベクトル比較が必要となる。しかし、軸索をのば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。

軸索キャッシュの実装では、リンクドリストとハッシュテーブルを組み合わせた単純なFIFOにより実現した。ベクトル比較が行われる度にペア情報をリンクドリストの先頭へ登録し、ハッシュテーブルにその比較結果を保存する。キャッシュの容量を越えた場合、リンクドリストの末尾からデータを削除し、対応する比較値もハッシュテーブルから削除する。

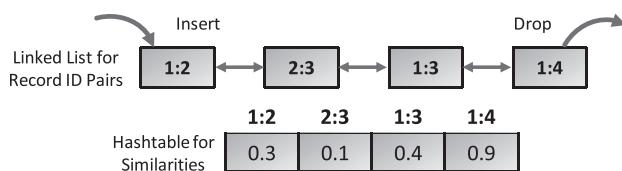


図3 軸索キャッシュの実装

Fig.3 Axonal cache implementation

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ア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。

4. 評価実験

4.1 パフォーマンスに関する検証

本実験では、提案手法とベースライン手法 (KohonenのSOM)について、生成されたマップがどれほど有効かを検証した。KohonenのSOMをベースライン手法として利用した理由は、2節で述べた通り、WebSOMなどの新しいSOMアルゴリズムであっても、基本原理はKohonenのSOMとほぼ変わらないためである。検証方法は以下の通りである。

- 全入力レコードの組み合わせのペア集合を生成
- 各ペアに対し、マップ上に写像された場所を特定
- 各ペアの（ベクトルの）コサイン類似度と、「マップ上での距離」を計算
- コサイン類似度集合と距離集合について、ピアソンの相関係数を求める

つまり、類似している入力レコードのペアが、マップ上でどの程度近くに配置（写像）されているかを検証した。マップサイズと写像性能の対応を検証するため、データ数を100, 150, 200, 250と変化させ、それに伴いマップサイズも増加させて実験した。ベクトルの次元数は、データ数と同様とした。なお、各データサイズにおいて、ランダムに10個の疎行列を生成・実験し、その平均値を実験結果として採用した。実験結果を図4に示す。横軸はCPUのサイクルタイム（処理時間）を示している。イテレーション数ではなく、CPUのサイクルタイムを利用した理由は、一回のイテレーションに必要な処理時間が手法によって大幅に異なるためである。図3に実験結果を示す。

まず、従来手法は、提案手法に比べて短い時間で収束することがわかった。従来手法は200~400CPUサイクル程度で収束しているのに対し、MIGSOMは1,600~1,800CPUサイクル程度で収束している。一方、相関値では、MIGSOMのほうが若干高いスコアを達成できていることが判明した。従来手法では、データ数が100の時は0.6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0.35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、データ数が多くなっても強い相関を保つと言える。

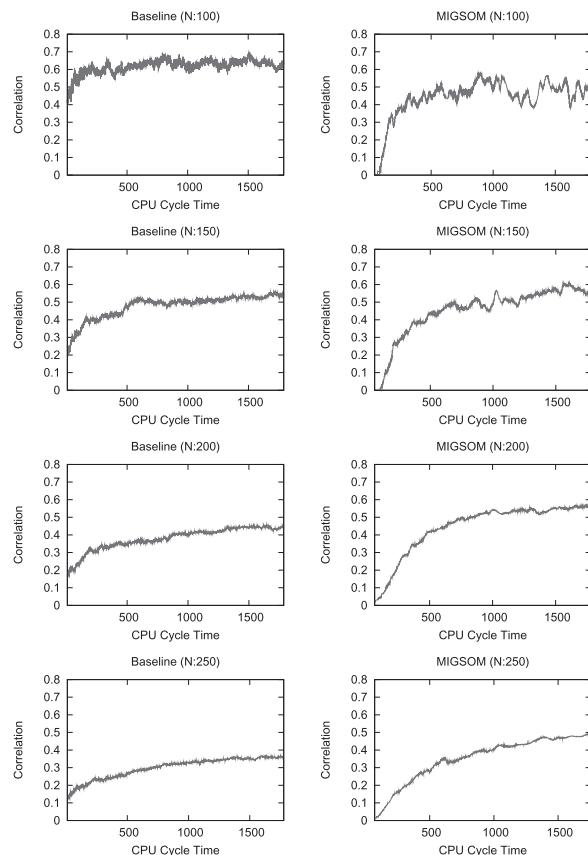


図4 相関値とデータスケールについての比較

Fig.4 Correlation by scale comparison

4.2 マップサイズに関する考察

次に、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。図5は、データ量とマップサイズを増加させたときに、各手法が性能ピーク時に使用するメモリ量を測定した結果である。データサイズを100~1,000に変化させ、データサイズに比例してマップサイズも増加させた。小規模密行列（次元数：100）と大規模疎行列（次元数：N）の二つのデータセットを作成した。それぞれランダムに10個の疎行列を各データ規模で生成・実験し、その平均値を実験結果として採用した。

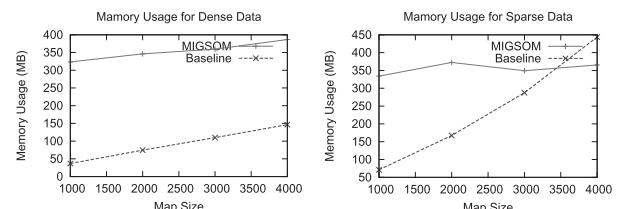


図5 メモリの使用状況

Fig.5 Memory Usage

まず、密行列データ（図5左）に関しては、従来手法のほうが少ないメモリ使用量で解析できているのがわかる。また、データ規模の大きさにメモリ使用量が影響を受けていないことがわかる。これは、小規模密行列では学習によるベクトル合成が行われても、メモリ使用量に影響しないためである。

一方、疎行列データ（図5右）では、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ことがわかる。これに対し、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る。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いためである。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てもメモリ使用量は線形に推移する。

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 Wikipedia¹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[9]。 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。なお、実行結果の解析を容易にするために、予め記事データをK-Meansクラスタリングによって10のクラスタに分類し、クラスタ毎に色分けした。解析結果を図6に示す。ここに示しているのは、Xeon 2.2GHzのマシンで解析開始から48時間後の様子である。 K-Meansで同一のクラスタ（同じ色）に分類された記事が、 MIGSOMでも近くに配置されており、MIGSOMのクラスタリング能力が機能していることを示している。なお、本マップは、マップの上下と左右が結合している円環状の空間である。マップを円環状にすることによって、全てのノードにおいて周辺ノード数と選択確率が一様となる。

¹ <http://wikipedia.org>

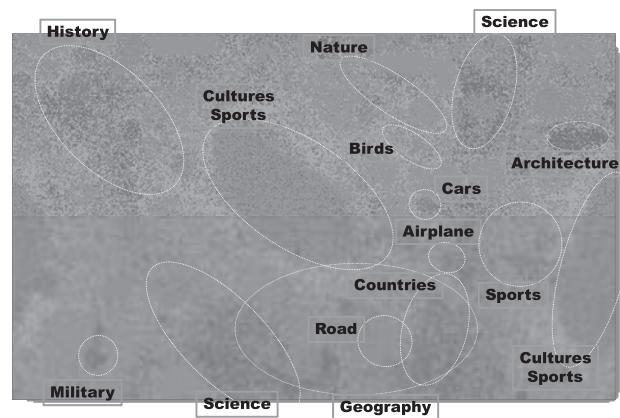


図6 発見されたクラスタ

Fig.6 Discovered Clusters

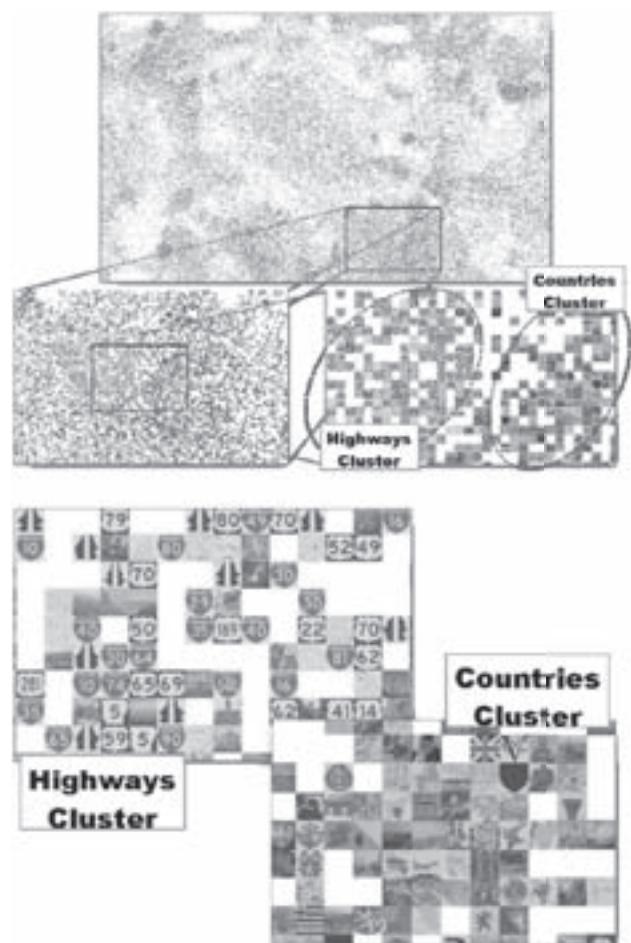


図7 インタラクティブなクラスタ解析のUI

Fig.7 Interactive Zooming Interface

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7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るのがわかる。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」を提案し、その有効性を実験によって評価した。実験の結果、従来手法に比べて収束に時間がかかるものの、高い写像性能を持つことを示した。さらに、データ数（とマップサイズ）が増加しても、使用するメモリサイズは線形に増加することを示した。また、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/>

今回の研究で、MIGSOMの実用性および、潜在的な研究の方向性を見出しができたが、課題も明らかになった。特に、計算速度は向上の余地がある。大規模データを解析する際の計算速度を向上させるための方法として、1) 分散処理への対応と、2) 軸索キャッシュの永続化が有望な研究の方向性だと考える。MIGSOMでは、マップ上をニューロンが移動することで学習を行うが、各ニューロンの移動は独立した動作であるため、マップの領域を複数のマシンに分割することで並列処理に対応できる。この結果、計算速度の向上が期待できる。また、各反復処理において、毎回ランダムに軸索をつなぎ直しているが、類似度の高い軸索の結合情報を保持しておき、一部分だけ変更するなどの処理を加えることで、さらなる性能向上が図れると考えられる。

【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。

【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 (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 (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).

中山 浩太郎 Kotaro NAKAYAMA

東京大学知の構造化センター 特任助教。2007 大阪大学大学院情報学研究科博士課程修了。情報科学博士。人工知能、データベース、Web / Wikipedia マイニングの研究・開発に従事。人工知能学会、情報処理学会、日本データベース学会、ACM, IEEE の会員。