自己組織化学習法則による多変量解析

樋口　勇夫
Isao Higuchi

博士(学術)

総合研究大学院大学
数物科学研究科 統計科学専攻

平成10年度
(1998)

目 次

第1章　Introduction

最近、ニューラルネットワーク理論の視点から、統計的推測の多くの新しい手法が提案されている。そのアプローチの性能は主にコンピュータによる数値実験によって確かめられている。しかし、これらの新しい手法に対する理論的な考察はまだ十分にはなされていない。このような新しい手法のひとつに、Xu and Yuilleによって提案された、自己組織化法則によるロバスト主成分分析がある。この場合もそのロバストネスはシミュレーションの結果によってのみ示されている。ロバストネスは影響関数の理論から示すことができる。この論文では、自己組織化法則による主成分分析で得られる統計量の影響関数を計算し、ロバストネスに理論的な裏付けを与える。

ニューラルネットワークの特徴としては、情報処理の並列性と、学習のプロセスがあげられる。ニューラルネットワークでは、複数のニューロンが結合して、それぞれのニューロンは個別に情報処理を行う。そのため、それぞれのニューロンにおける情報処理が低速であっても、ネットワーク全体では複雑な処理を高速に行うことができるのである。また、ニューラルネットワーク内では、情報は分散的に存在し、冗長性の高いものになる。これは、効率が悪くなるが、次のような長所がある。まず、学習の可能性である。分散的表現では、すべての文字をあらかじめ把握するのではなく、新しい文字に遭遇した時点で、それ以前に学習した文字との区別を学習していく。その過程で概念形成ができ、未知の事例への対応が可能になる。分散的表現では情報の類似性が表現に反映され、それぞれの概念の類似性を把握することが容易である。そのため、未知の

事例に出会っても、それに近い概念を見つけ出すことができる。さらに、情報の頑健性があげられる。効率的な情報表現では、1つのビットの損失でさえ、致命的であるが、分散的な情報表現では、一部のユニットがダメージを受けても情報の概略は保たれるのである。

ニューラルネットは個々のニューロンの学習というプロセスで結合比重の修正を行い、それによってネットワークの情報処理が決定される。代表的な学習法則として、「ニューロンが興奮したとき、入力の大きかったシナプスの結合を強化する」というヘブの学習法則がある。

ニューラルネットワークの学習のアルゴリズムは、目的関数の最適化のプロセスになっている。そのため、逆に目的関数の最適化によって行われる統計解析をニューラルネットワークモデルで実現し、ニューラルネットワークの学習理論を統計解析に適用できる。学習理論によって外れ値に対する最適化を導入したのがXu and Yuilleであり、この方法は他の統計解析への応用も期待できるものである。

主成分分析は情報圧縮と特徴抽出の基本的な手法として、統計データ解析、パターン認識、画像処理などに幅広く用いられている。主成分分析の古典的な方法では、最大固有値に対応する固有ベクトルを主成分ベクトルと呼び、データの主成分ベクトルへの射影をそのデータの主成分と呼ぶ。このとき主成分は、その分散が線形結合としては最大になるという意味で、データの特徴を抽出したものとなっている。

古典的な主成分分析は実データのように外れ値を含むデータを解析するとき、望まない結果が出てしまうことがしばしばあるということが報告されている。このことに対して2つの異なったアプローチがある。一つは、影響分析によってデータの外れ値を除去することである。影響関数を用いることで、統計量を大きく左右するデータを判別することができる。このとき、影響の大きなデータを外れ値とみなして、これらを取り除いたデータによって統計量を求めるというものである。もう一つは、古典的

な方法を別の方法でロバスト化することである。例えばMaronna(1976)は、データ共分散行列のM-推定量を用いたロバスト主成分分析を提起している。これは、平均から一定の距離以上離れているデータを自動的に切り落とした分散行列になっている。これら2つのアプローチは、観測データの特徴や、解析の目的に応じて選択される。ロバスト化の効果を評価するにはHampelの提唱した影響関数を用いるのが一般的である。Xu and Yuilleの方法はある重み付き分散行列の固有値問題に帰着し、その行列の影響関数からXu and Yuilleのアルゴリズムによる統計量の影響関数を求めることができる。Xu and Yuilleの方法が重み付き分散行列の固有値問題に帰着されるということから、その重みつき分散行列を用いた別のアルゴリズムも考えることができる。

$\{\boldsymbol{x}_i; i = 1, \cdots, N\}$ を n 次元ランダムサンプルとし、標本分散行列を、

$$S = \frac{1}{N}\sum_{i=1}^{N}(\boldsymbol{x}_i - \bar{\boldsymbol{x}})(\boldsymbol{x}_i - \bar{\boldsymbol{x}})^T \tag{1.1}$$

とする。ただし$\bar{\boldsymbol{x}}$は標本平均$\bar{\boldsymbol{x}} = \frac{1}{N}\sum \boldsymbol{x}_i$とする。従って主成分はデータと主成分ベクトルの内積として与えられる。古典的な主成分分析はSについての固有値問題をとくことに帰着される。ここで標本行列は外れ値の影響を強く受けるので、主成分分析はロバストではないのである。さらに、このような行列の固有値問題は、オンラインデータのように、次々とデータが追加される状況ではあまりよい方法ではない。なぜなら、データが追加されるとき、それまでの計算結果を有効に用いることができず、行列の計算から再びやり直さなければならないからである。

Amari(1977)は単一の線形ニューロンが自己組織化学習法則によって主成分分析を行うことを示した。ニューラルネットワークアプローチでは、オンラインデータに対する結合比重の修正法則が、学習の理論として研究されている。従って、線形ニューロンによる主成分分析も、オンラインデータと相性がよい。一つの単純な線形ニューロンを考えると、その出

力は入力と結合比重との内積となる。このことから、学習法則によって、ニューロンの結合比重ベクトルを主成分ベクトルに対応させることができる。

学習法則は統計物理学のアナロジーにおけるエネルギー関数の最小化アルゴリズムとして構成される。外れ値を記述するエネルギー関数は自己組織化のロバストネスを導入することで議論されて、最小化の過程で外れ値を除去するように作られる。自己組織化法則による主成分分析は主成分の方向に直交する方向の外れ値に対してロバストである。

第2章ではニューラルネットワークの自己組織化法則について説明する。主成分分析については第3章で説明する。第4章では主成分分析のロバスト化についてMaronnaによる方法や、Xu and Yuilleの方法を解説し、重み付き行列を用いた新しい方法も提起する。第5章ではこれらの方法について影響関数の理論を用いて議論する。そして、他の統計解析への応用の一例として自己組織化双対尺度法を第6章で解説する。

第2章　ニューラルネットワークと自己組織化法則

この章ではニューラルネットワークと、その学習法則、特に自己組織化法則について説明する。ニューラルネットワークは個々のニューロンが結合比重としきい値というパラメータを持っており、大きなネットワークになるとそのパラメータの数も多くなる。従ってこれらのパラメータを最適なものとするための学習理論が広く研究されている。種々の統計解析を行うニューラルネットワークも盛んに研究されているが、学習理論もそれとは別の形で度々統計解析に応用される。

2.1　ニューラルネットワーク

生物の脳の仕組みを調べると、数多くのニューロンと呼ばれる細胞が相互に結合していることがわかる。このようなニューロンのネットワークがニューラルネットワークである。各ニューロンは他のニューロンから受け取った信号をもとに情報処理を行い、それを他のニューロンに送る。各ニューロンが同時にこのような情報処理を繰り返すことで、ネットワークとして複雑な情報処理を高速に行うことができる。情報処理には直列と並列の大きく2つの原理がある。ニューラルネットワークは並列的情報処理の代表的なものである。

直列的情報処理は人間の思考や推論の過程に代表される情報処理である。チューリングは思考の原理を解明するために、手順に従って計算や

論理操作を行うアルゴリズムに目をつけた。そして、アルゴリズムを実行する機械であるチューリング・マシンを考案した。現代、広く用いられているコンピュータの多くははチューリング・マシンと同じ原理の直列的情報処理機械である。

しかし、生物学的な調査の結果、生物の脳は決してこのような直列的な情報処理でなく、ニューラルネットワークによる並列的情報処理を行っていることがわかった。コンピュータの発達は並列的な情報処理のシミュレートも可能にし、ニューラルネットに対する研究も急速的に発展していった。ニューラルネットの特徴は並列的な情報処理と学習による最適化である。本節では学習法則の一つである自己組織化法則について解説する。

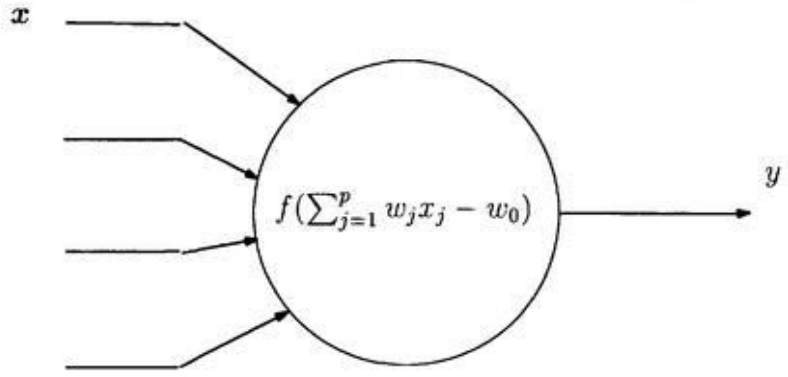
2.2 ニューラルネットワークの数理モデル

生物の脳は膨大な数のニューロンと呼ばれる細胞によって構成されている。それらは、互いに結合することによって高度な情報処理を行う。このような神経回路網における情報処理の特徴として、並列性があげられる。各ニューロンは多くの信号を受け取り一つの信号を出力する。出力された信号は別のニューロンに伝えられて入力となる。このような作用がすべてのニューロンで同時に行われることによって、神経回路網における情報処理が行われる。各ニューロンはシナプスと呼ばれる部分で入力信号を受け取る。このときシナプスでは各入力を重みつきで受け取る。これらの重みはシナプス結合と呼ばれる。

神経回路網に適切な情報処理をさせるためには、各ニューロンのシナプス結合比重を設定しなければならない。しかし、ユニットの数が多くなると、すべての結合比重を設定するのことは、困難になる。そのため、ニューロン自身が入力と出力からその結合比重を調整し、適切な処置を

行っていく学習能力が重要である。

ニューラルネットワークの数理モデルは、ニューロンという情報処理素子の結合したものとして考えられる。ニューロンは結合比重としきい値というパラメータをもち、これらによって入出力関係が決定する。



ニューロンのモデル

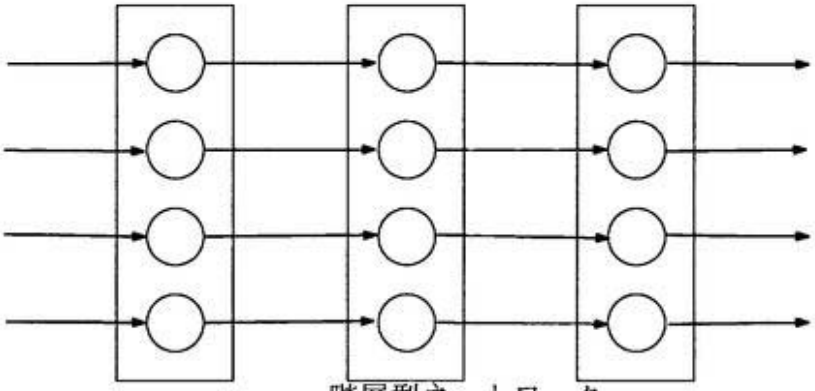
ニューロンのモデルでは、入力$\boldsymbol{x}$に対し出力yは次のような関係式で定義される。

$$y = f(\sum_{j=1}^{p} w_j x_j - h).$$

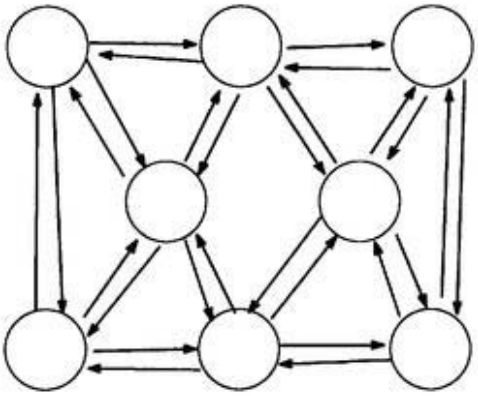
ここでw_jは結合比重であり、hはしきい値である。ここで関数$f(z)$としては、恒等関数、ステップ関数、シグモイド関数$f(z) = 1/\{1+\exp(-z)\}$、などが経験的に用いられている。

ニューラルネットワークは大きく階層型ネットワークと相互結合型ネットワークの2つにわけられる。階層型ネットワークはニューロンは複数の層をなして並び、入力層から出力層へ向かう方向の結合のみが存在するネットワークである。階層型ネットワークの最も基本的なものとして、パーセプトロンがある。これに対し、相互結合型ネットワークは、任意の2つのニューロンに双方向の結合があるようなものである。相互結合型ネットワークの代表的なものとして、連想記憶のモデルとなるホップ

フィールドネットワークや、各ニューロンが確率的に動作するボルツマンマシンなどがある。



階層型ネットワーク



相互結合型ネットワーク

2.3 パーセプトロン

階層型ネットワークの基本形であるパーセプトロンは1958年にRosenblattによって提唱された、学習するパターン識別ネットワークである。オリジナルのパーセプトロンはS(Sensory)層、A(Association)層、R(Response)層という3つの層からなり、S層からA層、A層からR層への一方向の結合を持つ。同じ層のニューロン間には結合が無い。さらに、S層からA

層への結合は固定されており、学習はA層からR層への結合を修正することによって行われる。S層は網膜の細胞に類比され、明暗パターンに応じた二値信号を出力し、その信号はA層への結合によって変換される。A層のニューロンはしきい素子で、S層の出力を結合の重み付き総和としてうけとり、それがしきい値より大きければ1を、そうでなければ0を出力する。R層も同様にしきい素子で、A層の出力の重み付き総和としきい値との比較により、0または1を出力するが、結合比重は可変であり、学習が可能になっている。

ここでは単純な一般化を行った次のようなネットワークをパーセプトロン型ネットワークと呼ぶ。パーセプトロン型ネットワークはいくつかの層からなる階層的ネットワークで、各層内の結合は無い。層間の結合は入力層から出力層へ向けての一方向の結合だけが存在するものとする。入力層を除く各層のニューロンは、前の層の出力の重み付き総和から、出力関数 $f(z)$ に従った出力を行う。すなわち、第 k 層の第 i ニューロンの入力の総和と出力をそれぞれ z_i^k, y_i^k と書くことにし、第 $k-1$ 層の第 j ニューロンから第 k 層の第 i ニューロンへの結合比重を $w_{j,i}^k$、第 k 層第 i ニューロンのしきい値を θ_i^k とすると、

$$\begin{aligned} z_i^k &= \sum_j w_{j,i}^k y_j^{k-1} - \theta_i^k \\ y_i^k &= f(z_i^k) \end{aligned} \tag{2.1}$$

となる。オリジナルのパーセプトロンでは出力関数 $f(z)$ はステップ関数

$$f(z) = \begin{cases} 1 & \text{if } z \geq 0 \\ 0 & \text{if } z < 0 \end{cases}$$

であるが、そのほかに区分線形関数、

$$f(z) = \begin{cases} 1 & \text{if } z \geq Z_0 \\ az & \text{if } Z_1 < z < Z_0 \\ 0 & \text{if } z \leq Z_1 \end{cases}$$

やシグモイド関数 $f(z) = 1/\{1 + \exp(-z)\}$、恒等関数 $f(z) = z$ などもよく用いられる。

さらに3層パーセプトロンの学習について説明する。パーセプトロンの学習は出力層とその一つ前の層との結合に対して行われる。その学習アルゴリズムは誤り訂正法と呼ばれるものである。パーセプトロンでは層内の結合がないので、学習について考えるときは出力層はただ一つのニューロンからなるとしてよい。このときネットワークは入力されたパターンを2種類に識別する。誤り訂正法はパターンを実際に入力してみて出力が誤っていたときに結合を修正する方法である。今入力パターンと正解のペア $\{(\boldsymbol{x}_s, y_s)\}(s = 1, \cdots, N)$ が与えられたとする。つまり時刻 t において、ネットワークに入力パターン $\boldsymbol{x}_s = (x_{s1}, x_{s2}, \ldots, x_{sn})$ が入力されたとき、出力ニューロンが出すべき正解を y_s とする。このとき A 層の第 i ニューロンの出力が、$\boldsymbol{y}_i^A(s)$、R 層出力が $y^R(s,t)$ であるとすると、結合 w_{ij} を次のように修正する。

$$w_{ij}(t+1) = w_{ij}(t) + (y_s - y^R(s,t))y_i^A(s).$$

ここで、正解 y_s を教師信号、$d_j(s,t) = y_s - y^R(s,t)$ を学習信号と呼ぶ。言いかえると、パーセプトロンの学習は正解と実際の出力との誤差を学習信号とする学習である。

この式からわかるように、ネットワークが正しく正解を出力している場合結合は変化しない。正解が1であるときにネットワークが0を出力していた場合、そのとき1を出力したニューロンとの結合が増加する。逆に正解が0であるときにネットワークの出力が1ならば結合が減らされる。これをいろいろな入力パターンに対して繰り返すことによってパーセプトロンは正しい識別を行うようになるわけである。

第3章　主成分分析

この章では主成分分析とそれを行うニューラルネットワークモデルについて説明する。主成分分析は多変量解析の基本的な手法であり、幅広く応用されているものである。主成分分析は比較的な単純なニューラルネットワークで実現でき、オンラインデータに対する主成分分析では非常に有用である。

3.1　主成分分析

n 次元のデータが N 個与えられた時、そのデータの構造、あるいは主な特徴を知りたいとする。そのためには高次元のデータを低次元に縮約するのが効果的である。縮約の方法はいくつかあるが、そのなかで最も簡単で、広く知られているものとして主成分分析がある。

データセットの主な特徴を捉えるため、主成分分析では、分散が最大になる方向を探し出す。つまり、そこへデータを射影した時に分散が最大となるような部分空間 $\mathcal{L}$ を見つける。もし $\mathcal{L}$ がユニットベクトル a で張られる直線ならば、射影 $P_{\mathcal{L}}\boldsymbol{x}$ は $P_a\boldsymbol{x} = aa^T\boldsymbol{x}$ で与えられ、その長さの2乗は $\|P_a\boldsymbol{x}\|^2 = (a^T\boldsymbol{x})^2 = a^T\boldsymbol{x}\boldsymbol{x}^Ta$ となる。従ってその直線に対する散らばりの平均は、a^TSa となる。ここで $S = \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{x}_i\boldsymbol{x}_i^T$ は標本分散行列である。主成分分析では、すべてのユニットベクトルの中で a^TSa が最大となるような a^* を見つける。もし $\hat{\lambda}_1 > \cdots > \hat{\lambda}_n > 0$ を S の固有値で、$\hat{\boldsymbol{\mu}}_1, \ldots, \hat{\boldsymbol{\mu}}_n$ に対応するものとすると、求めるべきベクトルは、$a^* = \hat{\boldsymbol{\mu}}_1$ である。

結論として主成分分析はデータの散らばりが最大となる方向を見つけることから始まる。それはデータ分散行列の第1固有ベクトル$\hat{\mu}_1$の方向である。縮約された第1の「特徴」は第1主成分$\hat{\mu}_1^T \boldsymbol{x}$である。

よって、主成分ベクトルは、標本分散行列の第1固有ベクトルとして導かれ、次のように最大固有値に対応する

$$S\hat{\mu}_1 = \frac{\sum(\hat{\mu}_1^T(\boldsymbol{x}_i - \bar{\boldsymbol{x}}))^2}{N\|\hat{\mu}_1\|^2}\hat{\mu}_1. \tag{3.1}$$

古典的手法による主成分ベクトル$\hat{\mu}$は$\boldsymbol{x}_i - \bar{\boldsymbol{x}}$からベクトル$\boldsymbol{m}$へ射影したベクトルの長さの二乗和をエネルギー関数$J(\boldsymbol{m})$として、それを最小化する。

$$J(\boldsymbol{m}) = \sum_{i=1}^{N} z(\boldsymbol{x}_i - \bar{\boldsymbol{x}}, \boldsymbol{m}), \tag{3.2}$$

ここで、

$$z(\boldsymbol{x}, \boldsymbol{m}) = \|\boldsymbol{x}\|^2 - \frac{(\boldsymbol{m}^T\boldsymbol{x})^2}{\|\boldsymbol{m}\|^2}. \tag{3.3}$$

である。

主成分分析はデータ共分散行列に基づくが、データ共分散行列は外れ値の影響を強く受ける場合がある。従って主成分分析もいくつかの外れ値の影響を受けて、本来求めるべきベクトルと大きく異なる結果が出てしまうこともありうる。

3.2 主成分分析を行うニューラルネットワーク

学習理論の一つとして、主成分分析を行うニューラルネットワークモデルも研究されている。ニューラルネットワークの学習法則はオンラインデータを扱う場合に容易に対応できるという特徴がある。

Amari(1977)は単一の線形ニューロンによる主成分分析が学習によって可能になることを示した。線形ニューロンとは入力$\boldsymbol{x}$に対して$\boldsymbol{m}^T\boldsymbol{x}$を出

力する情報処理素子であり、$\boldsymbol{m}$は結合比重である。ニューロンの学習はデータ集合 $\{\boldsymbol{x}_i\}$ に沿って$\boldsymbol{m}$を修正することによって行われる。

Hebb(1949) は、ニューロンの興奮が結合比重を増加させるルールを提起した。ニューロンの結合比重は、学習信号 $r(t)$ に比例して増加する。従って、比重の修正は次のような形に書ける。

$$\boldsymbol{m}(t+1) = \alpha \boldsymbol{m}(t) + r(t)$$

一般的に、学習信号 $r(t)$ は入力$\boldsymbol{x}(t)$、結合比重$\boldsymbol{m}$、出力$\boldsymbol{y}$と教師信号τの関数として決められる。$r_i(t) = x_i(t)\boldsymbol{y}(t)$ のとき、これは Hebbian rule である。また教師信号による学習法則は $r_i(t) = x_i(t)\tau(t)$ となる。教師信号は外から与えられるので、これは強制学習と呼べるだろう。ポテンシャル $u(t) = \sum_j \boldsymbol{m}_j(t)x_j(t)$ に対して、学習信号が $r_i(t) = x_i(t)u(t)$ で与えられる時、この学習はポテンシャルに基づく学習になる。

学習信号rは入力$\boldsymbol{x}$、結合比重$\boldsymbol{m}$、教師信号τの関数として、$r = r(\boldsymbol{x}, \boldsymbol{m}, \tau)$ と書くことができる。$\Delta\boldsymbol{m}(t)$ を時刻 t における結合比重の増分とする、

$$\Delta\boldsymbol{m}(t) = \boldsymbol{m}(t+1) - \boldsymbol{m}(t)$$

このとき結合比重の修正アルゴリズムは次のように書ける,

$$\Delta\boldsymbol{m}(t) = -c\boldsymbol{m}(t) + dr[\boldsymbol{m}(t), \boldsymbol{x}(t), \tau(t)],$$

ただし c, d は positive constant である。これは$\boldsymbol{m}(t)$ の修正の方向を示している。

結合比重の修正は、$\boldsymbol{x}(t)$ と$\tau(t)$ に依存している。そこで、$[\boldsymbol{x}(t), \tau(t)]$ がある分布からの独立な sample になる場合を考える。基本的なものは、k 個の入力$\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k$があり、$\boldsymbol{x}_\alpha$が確率あるいは相対頻度 p_αで発生する場合である。そして、$\boldsymbol{x}_\alpha$に対する教師信号τ_αを与えれば、これは連想の学習になる。

次に、

$$\frac{\partial Z}{\partial \boldsymbol{m}} = \boldsymbol{m} - \frac{d}{c} r(\boldsymbol{m}, \boldsymbol{x}, \tau)$$

となる Z が存在する場合を考える。このとき、

$$\Delta \boldsymbol{m} = -c \frac{\partial Z}{\partial \boldsymbol{m}}.$$

これは、$\boldsymbol{m}$の修正が Z の gradient 方向であることを示している。

ポテンシャルに基づく学習は、

$$Z(\boldsymbol{m}, \boldsymbol{x}) = \frac{1}{2}\left[\boldsymbol{m}\boldsymbol{m}^T - \frac{d}{c}(\boldsymbol{m}^T\boldsymbol{x})^2\right]$$

で定義される。ここで学習信号は $r = (\boldsymbol{m}^T\boldsymbol{x})\boldsymbol{x}$ である。また、$Z(\boldsymbol{m}, \boldsymbol{x})$ の $\boldsymbol{x}$に関する期待値を $J_x(\boldsymbol{m})$ とすると、

$$J_x(\boldsymbol{m}) = \frac{1}{2}\left[\boldsymbol{m}\boldsymbol{m}^T - \frac{d}{c}\boldsymbol{m}E[\boldsymbol{x}\boldsymbol{x}^T]\boldsymbol{m}^T\right]$$

である。しかし、関数 $J_x(\boldsymbol{m})$ は最小値を持たない。ゆえに上の学習法則では$\boldsymbol{m}(t)$ が収束しない。もし、結合比重に条件$\boldsymbol{m}\boldsymbol{m}^T = const.$、つまり $\boldsymbol{m}(t)$ が学習の各 Step で正規化されることにすれば、この条件の下で、$J_x(\boldsymbol{m})$ を最小化する。それは $E[\boldsymbol{x}\boldsymbol{x}^T]$ の最大固有値に対応する固有ベクトルの方向である。

主成分分析を行う線形ニューロンを考える場合、ランダムサンプル $\{\boldsymbol{x}_i\}$ に対し、$J_x(\boldsymbol{m})$ として、エネルギー関数 $J(\boldsymbol{m}) = \sum z(\boldsymbol{x}_i - \bar{\boldsymbol{x}}, \boldsymbol{m})$ が対応する。従って、自己組織化法則による主成分分析は、

$$\boldsymbol{m}(t+1) = \boldsymbol{m}(t) - \alpha(t)\sum_{i=1}^{N}\frac{\partial}{\partial \boldsymbol{m}} z(\boldsymbol{x}_i - \bar{\boldsymbol{x}}, \boldsymbol{m}(t))$$

で与えられる。

このアルゴリズムはデータがまとめて与えられる、いわゆるバッチ処理のアルゴリズムである。自己組織化法則では、オンラインデータに対

応したアルゴリズムはバッチ処理のアルゴリズムの簡単な修正によって実現できる。このアルゴリズムに対応するオンラインアルゴリズムは

$$\boldsymbol{m}(t+1) = \boldsymbol{m}(t) - \alpha(t)\frac{\partial}{\partial \boldsymbol{m}} z(\boldsymbol{x}(t) - \bar{\boldsymbol{x}}(t), \boldsymbol{m}(t))$$

となる。ここで$\bar{\boldsymbol{x}}(t) = ((t-1)\bar{\boldsymbol{x}}(t-1) + \boldsymbol{x}(t))/t$である。

第4章　自己組織化ロバスト主成分分析

この章では自己組織化法則の理論を用いた主成分分析のロバスト化について説明する。主成分分析のロバスト化には分散行列のロバスト推定量を用いる方法などがあった。主成分分析を行うニューラルネットワークモデルから、自己組織化法則によるロバスト化を考えることができる。さらに、このロバスト化はある重み付き行列の固有値問題と同値であり、この重み付き行列を分散行列のロバスト推定量とみなすことができる。

4.1　ロバスト主成分分析

ロバストな主成分分析のためにいくつかの方法が考えられてきた。例えば、Maronna(1976) は分散行列の M-推定量を用いた主成分分析を提起した。平均と分散行列のロバスト M-推定量$\tilde{\boldsymbol{x}}, \tilde{S}$は次の連立方程式の解として定義される。

$$\begin{aligned} n^{-1}\sum_{i=1}^{n} u[\{(\boldsymbol{x}_i-\tilde{\boldsymbol{x}})^T\tilde{S}^{-1}(\boldsymbol{x}_i-\tilde{\boldsymbol{x}})\}^{1/2}](\boldsymbol{x}_i-\tilde{\boldsymbol{x}}) &= \boldsymbol{0} \\ n^{-1}\sum_{i=1}^{n} u[\{(\boldsymbol{x}_i-\tilde{\boldsymbol{x}})^T\tilde{S}^{-1}(\boldsymbol{x}_i-\tilde{\boldsymbol{x}})\}^{1/2}](\boldsymbol{x}_i-\tilde{\boldsymbol{x}})(\boldsymbol{x}_i-\tilde{\boldsymbol{x}})^T &= \tilde{S} \end{aligned} \tag{4.1}$$

ここで $u(s) = -s^{-1}d[\log h(s)]/ds$ であり、$h(|\boldsymbol{x}|)$ は R^n 上の密度関数とする。

Xu and Yuille(1995) は、ニューラルネットの主成分分析アルゴリズムに、

外れ値を判定する役割を持つバイナリーフィールド$V = \{V_i : V_i = 0, 1\}$を導入することによって、ロバストな主成分分析のアルゴリズムを次のように提起した。

$$\boldsymbol{m}(t+1) = \boldsymbol{m}(t) - \alpha(t) \sum_{i=1}^{N} \gamma(t, \boldsymbol{x}_i - \bar{\boldsymbol{x}}, \boldsymbol{m}(t)) \frac{\partial}{\partial \boldsymbol{m}} z(\boldsymbol{x}_i - \bar{\boldsymbol{x}}, \boldsymbol{m}(t)). \tag{4.2}$$

ここで、

$$\gamma(t, \boldsymbol{x}, \boldsymbol{m}(t)) = \frac{1}{1 + \exp[\beta(t)\{z(\boldsymbol{x}, \boldsymbol{m}(t)) - \eta(t)\}]}. \tag{4.3}$$

である。ここでででてくる関数$\beta(t), \eta(t)$は次のようなものとする。

$$\lim_{t \to \infty} \beta(t) = \beta_0 \quad \text{and} \quad \lim_{t \to \infty} \eta(t) = \eta_0. \tag{4.4}$$

ここで、$\beta(t) > 0$は温度の逆数であり、$\eta(t)$はしきい値である。$beta(t)$が小さい時、すなわち温度が高い時は外れ値の判定はあまり sensitive でなく、逆に温度が低い時はzが少しでもしきい値$\eta(t)$を越えるとγはほとんど0となってしまう。

もしγがコンスタントならば、このアルゴリズムはAmari(1977)、Oja(1982)、で議論されている、古典的な主成分分析についてのアルゴリズムと一致する。

Xu and Yuille ではシミュレーションの結果によりこの手法がロバストであることを示している。

数値計算によって、Maronna の方法と、自己組織化アルゴリズムを比較する。45個のデータに5個の外れ値を加え、計算結果がどのくらい変化するかを調べる。いくつかのβおよびηに対して、外れ値を変えながら50回計算し、その結果と外れ値を加える前の主成分ベクトルとの内積の平均を取ったものが Table 1 である。Classical な方法と Maronnna の方法について同様の実験をしたところ、Classical では 0.9883、Maronna 方法

では0.9987であった。Table 1を見ると、$\eta = 8, 10$のとき、Maronnaよりも変化が少なく、ロバストであるといえる。この表で見るとβはロバストネスにあまり関係ないように見える。しかし、βが大きいときは、データによっては大きく外れてしまう。これは、外れ値の判定がsensitiveなため、本来データとみなすべきものを外れ値としてしまうケースがあるためである。

4.2 自己組織化ロバスト分散行列

自己組織化ロバスト主成分分析は次の重み付け分散行列の固有値問題に帰着される。

$$S_\gamma = \sum_{i=1}^{N} \gamma_0(\boldsymbol{x}_i)(\boldsymbol{x}_i - \bar{\boldsymbol{x}})(\boldsymbol{x}_i - \bar{\boldsymbol{x}})^T \tag{4.5}$$

ここで、

$$\gamma_0(\boldsymbol{x}) = \lim_{t\to\infty} \gamma(t, \boldsymbol{x} - \bar{\boldsymbol{x}}, \boldsymbol{m}(t)). \tag{4.6}$$

従って、古典的な主成分分析とここで議論する主成分分析の違いはSとS_γの挙動の違いとして説明できる。

議論を簡単にするため、経験分布関数を使ってアルゴリズムを書き直す。経験分布関数$\bar{F}_N$によってアルゴリズム(4.2)は次のように書き直される。

$$\begin{aligned}\boldsymbol{m}(t+1) =& \boldsymbol{m}(t) + \alpha(t) \int \gamma(t, \boldsymbol{x}^*, \boldsymbol{m}(t)) \Bigg[\{\boldsymbol{m}(t)^T \boldsymbol{x}^*\}\boldsymbol{x}^* \\ & - \frac{\{\boldsymbol{m}(t)^T \boldsymbol{x}^*\}^2}{\|\boldsymbol{m}(t)\|^2} \boldsymbol{m}(t) \Bigg] d\bar{F}_N(\boldsymbol{x}),\end{aligned} \tag{4.7}$$

ただし$\boldsymbol{x}^* = \boldsymbol{x} - \bar{\boldsymbol{x}}$である。$t \to \infty$の極限を取ると、

$$S_\gamma \mu_1 - \frac{\mu_1^T S_\gamma \mu_1}{\mu_1^T \mu_1} \mu_1 = 0,$$

ただし、$\mu_1 = \lim_{t\to\infty} m(t)$ である。従ってμ_1はS_γの第1主成分ベクトル$\tilde{\mu}$である。

アルゴリズム (4.7) は収束に非常に時間がかかる。そこで、もっと収束の早い別の方法を考えてみる。先に述べたように、自己組織化ロバスト主成分分析は、データ分散行列Sの代わりにS_γを用いた主成分分析とみなすことができる。そこで、S_γを反復計算で直接求める方法を提起する。$\boldsymbol{m}$の関数として、

$$S_\gamma^* = \sum_{i=1}^{N} \gamma(\boldsymbol{x}_i - \bar{\boldsymbol{x}}, \boldsymbol{m})(\boldsymbol{x}_i - \bar{\boldsymbol{x}})(\boldsymbol{x}_i - \bar{\boldsymbol{x}})^T$$

を考える。ただし、

$$\gamma(\boldsymbol{x}, \boldsymbol{m}(t)) = \frac{1}{1 + \exp[\beta_0\{z(\boldsymbol{x}, \boldsymbol{m}(t)) - \eta_0\}]}. \tag{4.8}$$

$\boldsymbol{m}(t+1)$を$S_\gamma^*(\boldsymbol{m}(t))$の第1固有ベクトルとして、反復計算を行う。すなわち、初期ベクトルを$\boldsymbol{m}(0)$として、次のように計算する。

$$\begin{aligned} &1) S_\gamma(t+1) = \sum_{i=1}^{N} \gamma(\boldsymbol{x} - \bar{\boldsymbol{x}}, \boldsymbol{m}(t))(\boldsymbol{x}_i - \bar{\boldsymbol{x}})(\boldsymbol{x}_i - \bar{\boldsymbol{x}})^T \\ &2) S_\gamma(t+1)\boldsymbol{m}(t+1) = \frac{\boldsymbol{m}(t+1)^T S_\gamma(t+1)\boldsymbol{m}(t+1)}{\|\boldsymbol{m}(t+1)\|^2}\boldsymbol{m}(t+1) \end{aligned} \tag{4.9}$$

を解く。

このとき、$\mu^* = \lim_{t\to\infty} \boldsymbol{m}(t)$とすると、

$$S_\gamma^*\mu^* - \frac{\mu^{*T} S_\gamma^* \mu^*}{\mu^{*T}\mu^*}\mu^* = 0$$

となる。

この方法について、数値計算の結果をしめす。正規乱数によってデータを生成し、そのデータの自己組織化分散行列をアルゴリズム (4.9) によって求める。まず、$N(0, \Sigma)$に従う50個の人工データについて実験する。η_0の設定は古典的なデータ分散行列の第1固有値λ_1をもとに、$\eta_0 = 2.7\lambda_1$と

する。データの分布が正規分布であるとき、90%以上のデータが含まれるようにη_0を設定した。

$$\Sigma = \begin{pmatrix} 9^2 & 0 & 0 \\ 0 & 4^2 & 0 \\ 0 & 0 & 1^2 \end{pmatrix} \tag{4.10}$$

の場合について50回計算した結果、Classicalな分散行列と第1固有ベクトルが一致した。従って、この方法は主成分分析の代用となりうると思われる。

さらに外れ値を加えた場合のこのアルゴリズムの有用性を検証するため、やはり人工的に主成分分析への影響の大きくなるデータを加えて実験する。データ分散行列の第1、第2固有ベクトルを$\boldsymbol{\mu}_1, \boldsymbol{\mu}_2$として、定数$a$に対し、$a(\boldsymbol{\mu}_1 + \boldsymbol{\mu}_2), -a(\boldsymbol{\mu}_1 + \boldsymbol{\mu}_2)$を新たに加えて上のアルゴリズムを適用する。ここで初期ベクトルとしては従来の分散行列の固有ベクトル$\boldsymbol{\mu}_1, \boldsymbol{\mu}_2$を用いるとする。$a = 5, 10, 15, 20$として実験したところ、$a = 5$では、従来のデータ分散行列でもあまり影響が出ないこともあり、特に優位性は確認できなかった。(Table 2) しかし、$a = 10, 15, 20$では従来のデータ分散行列は新たに加えたデータの影響を強く受けるのに対して自己組織化行列を用いる方法はデータを加える前とほぼ同じ固有ベクトルになる。ただし、従来のデータ分散行列の第1固有ベクトルを初期値とすると、そこに収束してしまうので、初期ベクトルとしてはデータ分散行列の第2固有ベクトル$\boldsymbol{\mu}_2$を用いるのがよい。

この方法のS-plusによるプログラムはList 1のようになる。

第5章　自己組織ロバスト主成分分析の影響関数

この章では、自己組織化法則によるロバスト主成分分析についてその影響関数を計算し、そのロバストネスについて理論的に検証する。さらに影響関数によって従来の他のロバスト化との比較も行う。

5.1　自己組織ロバスト主成分分析の影響関数

Hampel(1974) は統計量の局所的な挙動を調べるために、影響関数を提起した。ここでは統計量T_Nを汎関数$T(\bar{F}_N)$としてあらわす。ここで経験分布関数$\bar{F}_N$はディラック測度$\delta_y(\boldsymbol{x})$を用いて

$$\bar{F}_N(\boldsymbol{x}) = \frac{1}{N}\sum_{i=1}^{N}\delta_{\boldsymbol{x}_i}(\boldsymbol{x}) \tag{5.1}$$

とあらわされる。

たとえば、標本平均$\bar{\boldsymbol{x}}$は経験分布関数の汎関数として次のように書ける。

$$\bar{\boldsymbol{x}} \equiv \frac{1}{N}\sum_{i=1}^{N}x_i = \int \boldsymbol{x}d\bar{F}_N(\boldsymbol{x}). \tag{5.2}$$

そして標本分散行列も次のように書ける。

$$S \equiv \frac{1}{N}\sum_{i=1}^{N}(\boldsymbol{x}_i - \bar{\boldsymbol{x}})(\boldsymbol{x}_i - \bar{\boldsymbol{x}})^T = \int (\boldsymbol{x} - \bar{\boldsymbol{x}})(\boldsymbol{x} - \bar{\boldsymbol{x}})^T d\bar{F}_N(\boldsymbol{x}). \tag{5.3}$$

このようにして、主成分ベクトルも$\bar{F}_N$の汎関数としてあらわすことができる。影響関数は次のように定義されている。

$$IF(\boldsymbol{y}; \bar{F}_N, T) = \lim_{\varepsilon \to 0} \frac{T[(1-\varepsilon)\bar{F}_N + \varepsilon\delta_y] - T(\bar{F}_N)}{\varepsilon}. \tag{5.4}$$

従って影響関数は、局所的な感度の情報を与える。Nが充分に大きいと仮定する。このとき、

$$IF(\boldsymbol{y}; \bar{F}_N, T) \approx (N+1)(T(\bar{F}_{N+1}) - T(\bar{F}_N)), \tag{5.5}$$

ここで、$\bar{F}_{N+1} = \frac{N}{N+1}\bar{F}_N + \frac{1}{N+1}\delta_y$である。

従って影響関数は外れ値$\boldsymbol{y}$が加えられた後の統計量$T(\bar{F}_{N+1})$と元の統計量$T(\bar{F}_N)$との違いを漸近的に与える。もし影響関数が有界ならば、統計量は$\boldsymbol{y}$に対して柔らかな挙動を示す。逆に影響関数が非有界ならば、特定の$\boldsymbol{y}$はサンプルサイズが大きいとしても統計量に致命的な影響をおよぼす。

古典的な主成分分析では、第1主成分ベクトル$\hat{\mu}_1$の影響関数は、

$$IF_{\text{classic}} = IF(\boldsymbol{y}; \bar{F}_N, \hat{\mu}_1) = -\hat{a}_1(\boldsymbol{y}) \sum_{j=2}^{n} \hat{a}_j(\boldsymbol{y})(\hat{\lambda}_j - \hat{\lambda}_1)^{-1}\hat{\mu}_j, \tag{5.6}$$

ここで、$\boldsymbol{y} - \bar{\boldsymbol{x}} = \sum_j \hat{a}_j(\boldsymbol{y})\hat{\mu}_j$、書き直すと$\hat{a}_j(\boldsymbol{y}) = \hat{\mu}_j^T(\boldsymbol{y} - \bar{\boldsymbol{x}})$、そして$\hat{\mu}_j, \hat{\lambda}_j$は$S$の正規化された固有ベクトルと固有値$\hat{\lambda}_1 > \hat{\lambda}_2 > \cdots > \hat{\lambda}_n$である。(Critchley (1985)) もし$IF(\boldsymbol{y}; \bar{F}_N, \hat{\mu}_1) = 0$ならば、$\boldsymbol{y} - \bar{\boldsymbol{x}}$は$\hat{\mu}_1$によって張られる直線上にあるか、またはそれに直交する。さらに次のことがわかる。任意のξについて

$$IF(\xi\boldsymbol{y} + (1-\xi)\bar{\boldsymbol{x}}; \bar{F}_N, \hat{\mu}_1) = \xi^2 IF(\boldsymbol{y}; \bar{F}_N, \hat{\mu}_1).$$

これは古典的な主成分分析のセンシティビティを示している。なぜなら、$IF(\boldsymbol{y}; \bar{F}_N, \hat{\mu}_1) \neq 0$となる任意の$\boldsymbol{y}$に対して、

$$\|IF(\xi\boldsymbol{y} + (1-\xi)\bar{\boldsymbol{x}}; \bar{F}_N, \hat{\mu}_1)\| = O(\xi^2) \tag{5.7}$$

である。その意味で古典的な主成分分析は明らかにロバストではない。感度分析によるアプローチでは古典的な主成分分析の各観測値$\boldsymbol{x}_i$の影響を測度としてとる。

Section 2で紹介したように、ここではニューラルネットワークの方法による別のアプローチを用いる。Appendixで、(4.2)で定義した$\boldsymbol{m}(t)$がS_γの最大固有値$\tilde{\lambda}_1$に対応する主成分ベクトル$\tilde{\mu}_1$に収束することを示す。仮定(4.4)から、(4.6)で定義される極限$\gamma_0(\boldsymbol{y})$は次のようになる、

$$\gamma_0(\boldsymbol{y}) = \left(1 + \exp\left[\beta_0\left\{\sum_{j=2}^{n} \tilde{a}_j^2(\boldsymbol{y}) - \eta_0\right\}\right]\right)^{-1},$$

ただし$\boldsymbol{y} - \bar{\boldsymbol{x}} = \sum_j \tilde{a}_j(\boldsymbol{y})\tilde{\mu}_j$、すなわち$\tilde{a}_j(\boldsymbol{y}) = \tilde{\mu}_j^T(\boldsymbol{y} - \bar{\boldsymbol{x}})$である。ここで$\tilde{\mu}_1, \ldots, \tilde{\mu}_n$は$S_\gamma$の長さ1の固有ベクトルで、それぞれ固有値$\tilde{\lambda}_1 > \tilde{\lambda}_1 > \cdots > \tilde{\lambda}_n$に対応する。固有ベクトル$\tilde{\mu}_1$の影響関数は$S_\gamma$の影響関数から同様に計算できて、次のようになる、

$$IF_{\text{neural}} = IF(\boldsymbol{y}; F_N, \tilde{\mu}_1) = -\gamma_0(\boldsymbol{y})\tilde{a}_1(\boldsymbol{y}) \sum_{j=2}^{n} \frac{\tilde{a}_j(\boldsymbol{y})\lambda_j}{\tilde{\lambda}_j(\lambda_j - \lambda_1)}\tilde{\mu}_j. \qquad (5.8)$$

(5.6)と(5.8)の比較から、IF_{classic}とIF_{neural}との本質的な違いはスカラー関数$\gamma_0(\boldsymbol{y})$にあることがわかる。

(5.7)と同様に、

$$IF(\xi\boldsymbol{y} + (1-\xi)\bar{\boldsymbol{x}}; \bar{F}_N, \tilde{\mu}_1) = \frac{\xi^2}{1 + \exp(C_1\xi^2 - \beta_0\eta_0)} \frac{IF(\boldsymbol{y}; \bar{F}_N, \tilde{\mu}_1)}{\gamma_0(\boldsymbol{y})},$$

ただし$C_1 = \beta_0 \sum_{j=2}^{n} \tilde{a}_j^2(\boldsymbol{y})$である。従って$\xi\boldsymbol{y} + (1-\xi)\bar{\boldsymbol{x}}$に対する$\tilde{\mu}_1$の影響関数は有界であり、$\xi^2 \to \infty$のときゼロベクトルになる。スカラー関数$\gamma_0(\boldsymbol{y})$は$\tilde{\mu}_1$の方向の場合を除いて、$\|\boldsymbol{y}\|$が増加すると、急激に0に近づく。さらに$z = \sum_{j=2}^{n} \tilde{a}_j^2(\boldsymbol{y})$の関数として$\gamma_0(\boldsymbol{y})$の挙動を調べると、

$$\gamma_0(\boldsymbol{y}) = \frac{1}{1 + \exp\{\beta_0(z - \eta_0)\}}.$$

いくつかのβ_0とη_0についてzに関する$\gamma_0(\boldsymbol{y})$のグラフを示す。(Figure 1)ここで、β_0とη_0はそれぞれ(4.4)で与えられた、温度の逆数$\beta(t)$としきい値

$\eta(t)$ の極限である。これにより、温度としきい値は外れ値に対するニューラル主成分分析のロバストネスに関して直接的な役割を果たしていることがわかる。

5.2 数値計算例

この Section では、自己組織化の手法を土壌成分のデータに応用する。(Kendall (1975), Tables 2.1, 2.4) データは 4 次元で観測サンプルは 20 個である。主成分ベクトルは、

$$PC = (0.955785, 0.293681, 0.014972, 0.001317)$$

である。今回は$\alpha(t) = 0.001/t$、 $\eta(t) = 10$ (constant)、 and $\beta(t) = 10$ (constant) として計算した。その結果は、

$$\boldsymbol{m}_0 = (0.953688, 0.300199, 0.012510, 0.014235)$$

である。PC と$\boldsymbol{m}_0$の内積、すなわち間の角度のコサインは 0.99989008 であり。この 2 つのベクトルはほぼ等しいといえる。

Table 3 の示すように、Soil No. 4、13、17 に対して統計量 $|IF_{\text{classic}}|$ は大きくなっている。これは、これらのデータは主成分ベクトルへの影響が大きいことを示す。このことは固有値に対する影響関数についてもほぼ同様であることが、Critchley (1985) によって示されている。ここで統計量γ_0と $|IF_{\text{classic}}|$ との性能を比較する。統計量γ_0も $|IF_{\text{classic}}|$ と同様に Soil No. 4、13、17 を影響に強いデータとみなしている。さらに、No. 5 と 14 の 2 つのデータについてもγ_0は小さくなっている。このことから、γ_0を用いた感度分析も考えることができる。これはγ_0が、マスキング効果を識別している可能性がある。Table 4、Figure 2 によると、これら 2 つのデータの主成分は原点を挟んでほぼ対称な位置にあることがわかる。このようなデータは、単独では影響が少ないが複合することによって影響

が大き区なる場合がある。影響関数の代わりにγ_0を用いることで、このような外れ値も対象にした感度分析が可能になると期待される。

第6章　自己組織化双対尺度法

この章では自己組織化の他の統計解析法への応用として双対尺度法を示す。

6.1　双対尺度法

自己組織化法則による分散行列は、他の行列を用いた統計解析にも同様に応用できると考えられる。ここでは、その１つの例として双対尺度法に対しての応用について考える。

双対尺度法は 非計量的データを計量化することが目的である。双対尺度法では、同じグループに属するもののスコアの分散を小さく、異なるグループのスコアは分散を大きくするように非計量的データのそれぞれに重みを与える方法である。

データ行列を $F = \sum_{k=1}^{n} F_k$ とし、F_k は各個人の反応をあらわす行列とする。そして、$\boldsymbol{f}$ を F の列の周辺度数を要素とするベクトル、D を $\boldsymbol{f}$ の要素を対角項に持つ対角行列、D_n を F の行の周辺度数を対角項に持つ対角行列とする。また総反応数を f_t とする。

このとき全平方和は

$$\begin{aligned} SS_t &= \boldsymbol{x}^T D \boldsymbol{x} - \frac{\boldsymbol{x}^T \boldsymbol{f} \boldsymbol{f}^T \boldsymbol{x}}{f_t} \\ &= \boldsymbol{x}^T \left[D - \frac{\boldsymbol{f}\boldsymbol{f}^T}{f_t} \right] \boldsymbol{x} \end{aligned}$$

と表される。異なるグループ間のスコアの分散をあらわす級間平方和は

$$\begin{aligned} SS_b &= \boldsymbol{x}^T F^T D_n^{-1} F \boldsymbol{x} - \frac{\boldsymbol{x}^T \boldsymbol{f} \boldsymbol{f}^T \boldsymbol{x}}{f_t} \\ &= \boldsymbol{x}^T \left[F^T D_n^{-1} F - \frac{\boldsymbol{f} \boldsymbol{f}^T}{f_t} \right] \boldsymbol{x} \end{aligned}$$

となり、同一グループのスコアの分散である級内平方和は

$$\begin{aligned} SS_w &= SS_t - SS_b \\ &= \boldsymbol{x}^T D \boldsymbol{x} - \boldsymbol{x} F^T D_n^{-1} F \boldsymbol{x} \end{aligned}$$

となる。

従って、双対尺度法とは$\lambda = \frac{SS_b}{SS_t}$を最大化する$\boldsymbol{x}$を探すことである。

$\boldsymbol{f}^T \boldsymbol{x} = 0$を仮定すると、

$$SS_t = \boldsymbol{x}^T D \boldsymbol{x}$$

$$SS_b = \boldsymbol{x}^T F^T D_n^{-1} F \boldsymbol{x}$$

である。

さらに$\boldsymbol{m} = D^{1/2} \boldsymbol{x}$, $SS_t = f_t(\text{constant})$として$SS_b$を最大化する$\boldsymbol{m}$を求める。これは

$$C_0 = D^{-1/2} F^T D_n^{-1} F D^{-1/2}$$

の固有値問題に帰着される。

従って、

$$C_0 \boldsymbol{m} = \lambda \boldsymbol{m}$$

で、λを最大化する$\boldsymbol{m}$を求めれば良い。

ただし、自明な解$\boldsymbol{x} = \boldsymbol{1}$は$\lambda$の最大値$\lambda = 1$を与えるが、$\boldsymbol{f}^T \boldsymbol{x} = 0$を満たさない。

6.2 自己組織化双対尺度法

ニューラルネットの学習法則を用いて双対尺度法を行うには、エネルギー関数を、

$$J(\boldsymbol{m}) = \mathrm{tr}C_0 - \frac{\boldsymbol{m}^T C_0 \boldsymbol{m}}{\boldsymbol{m}^T \boldsymbol{m}}$$

とすれば良い。ここで、$F = \sum_{k=1}^{n} F_k$であるから、

$$J(\boldsymbol{m}) = \sum_{k=1}^{n} z(F_k, \boldsymbol{m})$$

となる。ただし、

$$z(F_k, \boldsymbol{m}) = \mathrm{tr}(D^{1/2} F_k^T D_n^{-1} F D^{1/2}) - \frac{\boldsymbol{m}^T D^{-1/2} F_k^T D_n^{-1} F D^{-1/2} \boldsymbol{m}}{\boldsymbol{m}^T \boldsymbol{m}}$$

である。$F = \sum_{k=1}^{n} F_k$より、

$$C_0 = \sum_{k=1}^{n} D^{-1/2} F_k^T D_n^{-1} F D^{-1/2}$$

と書くことができる。

自己組織化法則のウェイトは、

$$\gamma(F_k, \boldsymbol{m}) = \frac{1}{1 + \exp[\beta(t)\{z(F_k, \boldsymbol{m}) - \eta(t)\}]}$$

となり、自己組織化による双対尺度法は

$$C_1 = \sum_{k=1}^{n} \gamma(F_k, \boldsymbol{\mu}_2) D^{-1/2} F_k^T D_n^{-1} F D^{-1/2}$$

の固有ベクトルを求めることに帰着される。ただし、$\boldsymbol{\mu}_2$はC_1の第2固有値に対する固有ベクトルである。

しかし通常双対尺度法で扱うデータでは、$|z(F_k, \boldsymbol{m})|$が極端に大きくなることは稀である。従って、$z(F_k, \boldsymbol{m})$の大きさで、外れ値の判別を行う

ことは難しい。自己組織化では、求めるベクトルに直交する方向にあるデータのウェイトを減らすことになる。つまり、最も分散の大きい方向を強調することになる。データの傾向が2つのグループに分かれる場合、双対尺度法ではその2つを区別する方向をみつけだすが、そのグループが3つ以上の場合必ずしもうまくいかない。そのような場合に自己組織化を行うと、そのうちの2グループの判別が強調され、適切な重みづけが可能になると考えられる。(Figure 3)

第7章　まとめ

自己組織化ロバスト主成分分析は古典的な主成分分析の場合の外れ値による悪い影響を排除することができる。この方法の優位性はμ_1を中心とした円柱形の領域によって特徴づけることができる。平均から離れたデータは主成分分析に大きく影響を与える。特に主成分ベクトルμ_1から離れたデータは、結果を大きく変えてしまう。自己組織化ロバスト主成分分析では、これらのデータのウェイトを小さくすることで、ロバストネスを実現する。標本分散行列のロバスト推定値を用いる方法では (Maronna, 1976)、これは主成分分析のロバスト化として通常用いられるものだが、平均を中心とした楕円体状の領域で特徴づけられる。この通常の方法では、楕円形の外のデータが排除される。対照的に我々の方法では、楕円形の外にあっても、主成分ベクトルの方向に近い点については排除しない。このような点の影響は、実際には主成分ベクトルの方向を変化させない。そのためロバスト化のためにはこのような点を排除する必要はない。したがって我々の方法には、余分な点を排除しないという特徴がある。

我々の方法におけるデータの影響は円柱形の半径$\eta_0^{1/2}$に大きく依存している。もし、η_0が小さければ、多くのデータは外れ値とみなされてしまいこのアルゴリズムの結果はほとんど情報を持たないことになる。。逆に、η_0が大きくなると、このアルゴリズムはロバスト性を失ってしまう。そのため、η_0の選択は重要である。

Xu and Yuille のアルゴリズムは収束に時間がかかったが、ロバスト自己組織化分散行列を考えることにより、行列に対してのアルゴリズムで代用することができる。行列を用いるとステップ数は大幅に減少するが、

初期ベクトルへの依存性が高くなり、初期ベクトルの選択も問題になってくる。現在のところ、一度データ分散行列を求め、その固有値、固有ベクトルから、$\eta_0 = 2.7\lambda_1$とし第2固有ベクトル$\boldsymbol{\mu}_2$を初期ベクトルとするのが適当だと思われる。

自己組織化法則によるロバスト化は主成分分析以外の他の統計解析にも応用できる。その一例として双対尺度法を挙げた。双対尺度法ではロバスト化という意味合いはなく、特定のグループを強調することによる効率的な分類に効果を発揮すると思われる。

付 録A　影響関数の導出

ここでは、IF_{neural}を導く。すでに示したようにアルゴリズム

$$\begin{aligned}\boldsymbol{m}(t+1) =& \boldsymbol{m}(t) + \alpha(t) \int \gamma(t, \boldsymbol{x}^*, \boldsymbol{m}(t)) \Big[\{\boldsymbol{m}(t)^T \boldsymbol{x}^*\} \boldsymbol{x}^* \\ & - \frac{\{\boldsymbol{m}(t)^T \boldsymbol{x}^*\}^2}{\|\boldsymbol{m}(t)\|^2} \boldsymbol{m}(t) \Big] d\bar{F}_N(\boldsymbol{x}),\end{aligned}$$

ただし$\boldsymbol{x}^* = \boldsymbol{x} - \bar{\boldsymbol{x}}$、の収束先$\mu_1$は$S_\gamma$の固有ベクトルになる。外れ値の加わった分布

$$\bar{F}_{N,\varepsilon} = (1-\varepsilon)\bar{F}_N + \varepsilon\delta_y$$

について考えると、

$$\begin{aligned}\boldsymbol{m}_\varepsilon(t+1) = \quad & \boldsymbol{m}_\varepsilon(t) \\ & + \alpha(t) \Big[\quad (1-\varepsilon) \int \gamma(t, \boldsymbol{x}_\varepsilon^*, \boldsymbol{m}(t)) \frac{\partial}{\partial \boldsymbol{m}} z(\boldsymbol{x}_\varepsilon^*, \boldsymbol{m}_\varepsilon(t)) d\bar{F}_N(\boldsymbol{x}) \\ & \qquad + \varepsilon \gamma(t, \boldsymbol{y}_\varepsilon^*, \boldsymbol{m}(t)) \frac{\partial}{\partial \boldsymbol{m}} z(\boldsymbol{y}_\varepsilon^*, \boldsymbol{m}_\varepsilon(t)) \Big],\end{aligned}$$

ただし $\boldsymbol{x}_\varepsilon^* = \boldsymbol{x} - \bar{\boldsymbol{x}}_\varepsilon$, $\boldsymbol{y}_\varepsilon^* = \boldsymbol{y} - \bar{\boldsymbol{x}}_\varepsilon$, and $\bar{\boldsymbol{x}}_\varepsilon = (1-\varepsilon)\bar{\boldsymbol{x}} + \varepsilon \boldsymbol{y}$である。ここで、$\mu_{1,\varepsilon} = \lim_{t\to\infty} \boldsymbol{m}_\varepsilon(t)$ とすると、

$$IF_{\text{neural}} = \frac{\partial}{\partial \varepsilon} \mu_{1,\varepsilon}|_{\varepsilon=0} = \dot{\mu_{1,\varepsilon}}$$

である。同様にS_γの定義において$\bar{F}_N$を$\bar{F}_{N,\varepsilon}$で置き換えたものを$S_{\gamma,\varepsilon}$とすると、

$$\begin{aligned}S_{\gamma,\varepsilon} =& (1-\varepsilon) \left(S_\gamma + \int \frac{\partial \gamma(t, \boldsymbol{x}_\varepsilon^*, \boldsymbol{m}_\varepsilon(t)}{\partial \varepsilon} \boldsymbol{x}_\varepsilon^* \boldsymbol{x}_\varepsilon^{*T} d\bar{F}_n(\boldsymbol{x}) \right) \\ & + \varepsilon \lim_{t\to\infty} \gamma(\boldsymbol{y}, \boldsymbol{m}_\varepsilon(t)) \boldsymbol{y}^* \boldsymbol{y}^{*T} + O(\varepsilon^2),\end{aligned}$$

ただし$\boldsymbol{y}^* = \boldsymbol{y} - \bar{\boldsymbol{x}}$。$\mu_1$と同様に$\dot{S}_\gamma = \frac{\partial}{\partial \varepsilon} S_{\gamma,\varepsilon}|_{\varepsilon=0}$を考えると、$S_\gamma$の最大固有値$\lambda_1$に対して、、$S_\gamma \mu_1 = \lambda_1 \mu_1$が成り立つから、

$$(S_\gamma - \lambda_1 I)\dot{\mu}_1 = -(\dot{S}_\gamma - \dot{\lambda}_1 I)\mu_1$$

がいえる。従って、

$$\begin{aligned}
\dot{S}_\gamma \mu_1 &= -S_\gamma + \gamma_0(\boldsymbol{y})\boldsymbol{y}\boldsymbol{y}^T\mu_1 + \{\int \dot{\gamma}_0(\boldsymbol{x})\boldsymbol{x}\boldsymbol{x}^T dF(\boldsymbol{x})\}\mu_1 \\
&= -S_\gamma \mu_1 + \gamma_0(\boldsymbol{y})\boldsymbol{y}\boldsymbol{y}^T\mu_1 - \int \frac{\partial \gamma_0(\boldsymbol{y})}{\partial z} a_1^2 \boldsymbol{x}(\boldsymbol{x}^T\dot{\mu}_1) dF(\boldsymbol{x}) \\
&= -\lambda_1\mu_1 + a_1(\boldsymbol{y})\gamma_0(\boldsymbol{y})\boldsymbol{y}\boldsymbol{y}^T - \Gamma\Delta\Gamma^T\dot{\mu}_1
\end{aligned}$$

がいえる。ただし、

$$\Delta = \int \frac{\partial \gamma_0}{\partial z} a_1^2 a a^T dF(a)$$

さらに、

$$\dot{\lambda}_1 = \gamma_0(\boldsymbol{y})a_1^2(\boldsymbol{y}) - \lambda_1.$$

従って、

$$\dot{\tilde{\mu}}_1 = -\gamma_0(\boldsymbol{y})\tilde{a}_1(\boldsymbol{y}) \sum_{j=2}^{n} \frac{\tilde{a}_j(\boldsymbol{y})\lambda_j}{\tilde{\lambda}_j(\lambda_j - \lambda_1)} \tilde{\mu}_j.$$

関連図書

[1] 甘利俊一 (1989). 神経回路網モデルとコネクショニズム. 東京大学出版会.

[2] 甘利俊一 (1993). ニューラルネットの新展開. サイエンス社.

[3] AMARI, S.-I. (1977). Neural theory of association and concept formation. *Biol. Cybernetics* **26,** 175–185.

[4] 麻生英樹 (1988). ニューラルネットワーク情報処理. 産業図書.

[5] BALDI, F., & HORNIK, K. (1989) Neural networks and principal component analysis: Learning from examples without local minima. *Neural Networks,* **2,** 53–58

[6] BALDI, F., & HORNIK, K. (1995) Learning in linear neural networks: A survey. *IEEE Trans. on Neural Networks,* **6,** 837–858

[7] CAMERON, S., GROSSBERG, S., & GUENTHER, F. H. (1998) A self-organizing neural network architecture for navigation using optic flow. *Neural Computation,* **10,** 313–352.

[8] CHENG, B., & TITTERINGTON, D. M. (1994). Neural networks: A review from a statistical perspective. *Statistical Science* **9,** 2–54.

[9] CRITCHLEY, F. (1985). Influence in principal components analysis. *Biometrika* **72,** 627–636.

[10] DEVLIN, S. J., GNANADESIKAN, R., & KETTERNRING, J. R. (1981) Robust estimation of dispersion matrices and principal components. *J. Amer. Statist. Ass.* **69,** 383–393.

[11] HAMPEL, F. R. (1974). The influence curve and its role in robust estimation. *J. Amer. Statist. Ass.* **69,** 383–393.

[12] HIGUCHI, I., & EGUCHI, S (1998). The influence function of principal component analysis by self-organizing rule. *Neural Computation,* **10,** 1435–1444.

[13] HORNIK, K., & KUAN, C.-M. (1992) Convergence analysis of local feature extraction algorithms. *Neural Networks,* **5,** 229–240.

[14] HUANG, G.-B., BABRI, H. A., & LI, H.-T. (1998) Ordering of self-organizing maps in multidimensional cases. *Neural Computation,* **10,** 19–23

[15] HUBER, P. J. (1981). *Robust Statistics.* New York: Wiley.

[16] KENDALL, M. G. (1975). *Multivariate Analysis.* London: Griffin.

[17] KARHUNEN, J., & JOUTSENSALO, J. (1995). Generalizations of principal component analysis, optimization problems, and neural networks.*Neural Networks,* **8,** 549–562.

[18] LUO, F.L., UNBEHAUEN, R., & CICHOCKI, A. (1997) A minor component analysis algorithm. *Neural Networks,* **10,** 291–297.

[19] MARONNA, R. A. (1976). Robust M-estimators of multivariate location and scatter. *Annals of Statistics* **4,** 51–67.

[20] 西里静彦 (1982). 質的データの数量化. 朝倉書店.

[21] ニューロンネットグループ & 桐谷 滋 (1989). ニューロコンピュータ. 技術評論社.

[22] OJA, E. (1982). A simplified neuron model as a principal component analyzer. *J. Math. Biol.*, **15**, 267–273.

[23] OJA, E. (1983). *Subspace Methods of Pattern Recognition.* England: Reserch Studies Press. (邦題 : パターン認識と部分空間法. 小川英光, 佐藤誠 訳, 産業図書 (1986))

[24] OJA, E. (1992). Principal components, minor components, and linear neural networks. *Neural Networks,* **5**, 927–935.

[25] OJA, E., & KARHUNEN, J. (1985) On stocahastic approximation of the eigenvectors and eigenvalues of the expectation of a rondom matrix. *J. Math. Anal. Appl.,* **106, 69–84.**

[26] PLIMBLEY, M. D. (1995) Lyapunov functions for convergence of principal component algorithms. *Neural Networks,* **8**, 11–23.

[27] RUYMGAART, F. H. (1981). A robust principal component analysis. *J. Multivariate Anal.*

[28] SANGER, T. D. (1989). Optimal unsupervised learning in a single-layer linear feedforward neural network.*Neural Network,* **2**, 459–473

[29] SIBSON, R. (1979). Studies in the robustness of multidimensional scaling: perturbational analysis of classical scaling. *J. R. Statist. Soc. B* **41**, 217–229.

[30] 田中豊 (1983). 数量化法における感度分析. 数理科学 245, 32–37.

[31] 田中豊 (1992). 多変量解析における感度分析. 行動計量学 19, 3–17.

[32] 豊田秀樹 (1996). 非線形多変量解析. 朝倉書店.

[33] Xu, L. (1993) Least mean square error reconstruction principle for self-organizing neural-nets. *Neural Networks,* **6,** 627–648.

[34] Xu, L., & Yuille, A. (1995). Robust principal component analysis by self-organizing rules based on statistical physics approach. *IEEE Trans. on Neural Networks* **6,** 131–143.

[35] Xu, L., Oja, E., & Suen, C. Y. (1995). Modified Hebbian learning for curve and surface fitting. *Neural Networks* **5,** 441–457.

List 1

```
Sim3_function(DataArray=DataSet321,t=50,eta0=8)
{
r_matrix(0,t,2)
for (i in seq(t)){
  print(i)
  d_DataArray[i,,]
  S_var(d)
  ES_eigen(S)
  v1_ES$vectors[,1]
  v2_ES$vectors[,2]
  Sg_Sgst(data=d,pc=v1,eta0=eta0)
  m1_Sg$m0
  Sg_Sgst(data=d,pc=v2,eta0=eta0)
  m2_Sg$m0
  r[i,1]_crossprod(m1,v1)
  r[i,2]_crossprod(m2,v1)
}
return(r)
}

SimSgst_function(sv=c(9,4,1),eta0=8)
{
rst_list(m1=0,m2=0,v1=0,S=0)
d_t(matrix(rnorm(150,c(0,0,0),sv),3,50))
S_var(d)
v1_eigen(S)$vectors[,1]
v2_eigen(S)$vectors[,2]
Sg_Sgst(data=d,pc=v1,eta0=eta0)
rst$m1_Sg$m0
Sg_Sgst(data=d,pc=v2,eta0=eta0)
rst$m2_Sg$m0
rst$v1_v1
rst$S_S

return(rst)

}

Sgst_function(data=d0,eta0=8,beta0=1,ep=1e-8,pc=TRUE)
{
```

```
rst_list(m0=0,m=0,S=0,gm=0)
cl_ncol(data)
gm_Agam1(data)
Sg_var(data)
if(is.logical(pc)){m0_eigen(Sg)$vectors[,1]}
else{m0_pc}
mv_m0
flag_1
while(flag>ep){
  Sg_Sgs(data,eta0,beta0,m0)
  ma_eigen(Sg$S)$vectors[,1]
  mv_rbind(mv,ma)
  flag_sum(abs(ma-m0))
  m0_ma
  gm_rbind(gm,Sg$gm)
}
rst$m0_m0
rst$m_mv
rst$S_(Sg$S)
rst$gm_gm
return(rst)
}

Agam1_function(data=d0,eta0=8,beta0=1,pc=TRUE){
rw_nrow(data)
gm_rep(0,rw)
mx_apply(data,2,mean)
if(is.logical(pc)){pc_eigen(var(data))$vectors[,1]}
for (i in seq(rw)){
  vdata_data[i,]-mx
  z_crossprod(vdata,vdata)-(crossprod(vdata,pc)^2)
  gm[i]_1/(1+exp(beta0*(z-eta0)))
}
return(gm)
}

Sgs_function(data=d0,eta0=8,beta0=1,pc=TRUE){
r_list(S=0,gm=0)
rw_nrow(data)
cl_ncol(data)
mx_apply(data,2,mean)
S_matrix(0,cl,cl)
```

```
gm_Agam1(data,eta0,beta0,pc)
for(i in seq(rw)){
  vdata_data[i,]-mx
  S_S+gm[i]*vdata%o%vdata
}
S_S/sum(gm)
r$S_S
r$gm_gm
return(r)
}
```

Table 1:外れ値を加えることによる主成分ベクトルの変化

	β=2	4	6	8	10
$\eta = 4$	0.9861	0.9834	0.9806	0.9795	0.9801
6	0.9984	0.9968	0.9951	0.9950	0.9950
8	0.9998	0.9997	0.9997	0.9997	0.9997
10	0.9997	0.9996	0.9996	0.9996	0.9996

Table 2: 50 通りのデータセットに対して外れ値を加えない時の第 1 主成分ベクトルと、計算した結果との内積の平均

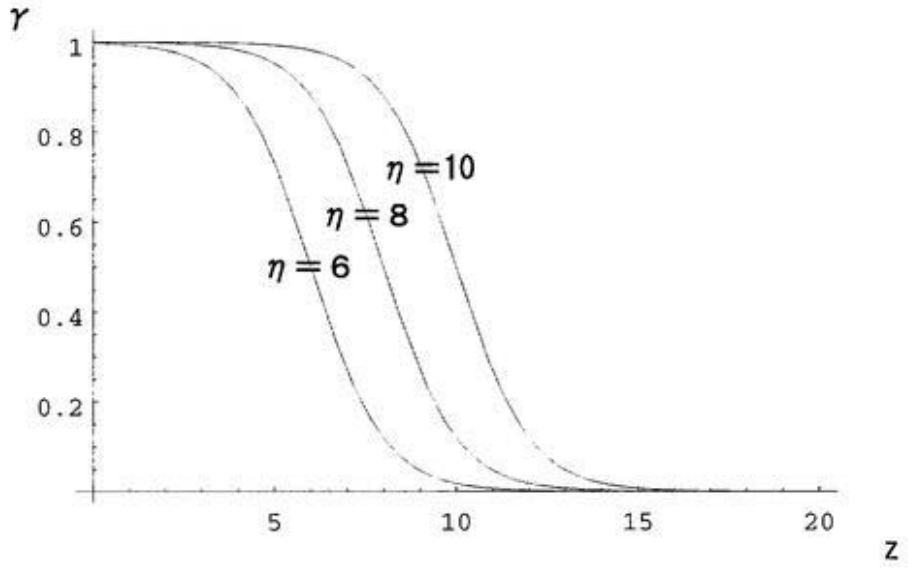
a	Classical	初期ベクトル μ_1	初期ベクトル μ_2
5	0.968	0.962	0.974
10	0.862	0.959	0.999
15	0.791	0.790	1.000

Table 3: Comparison between influence measures $|IF_{\text{classic}}|$ and γ_0 for soil composition data

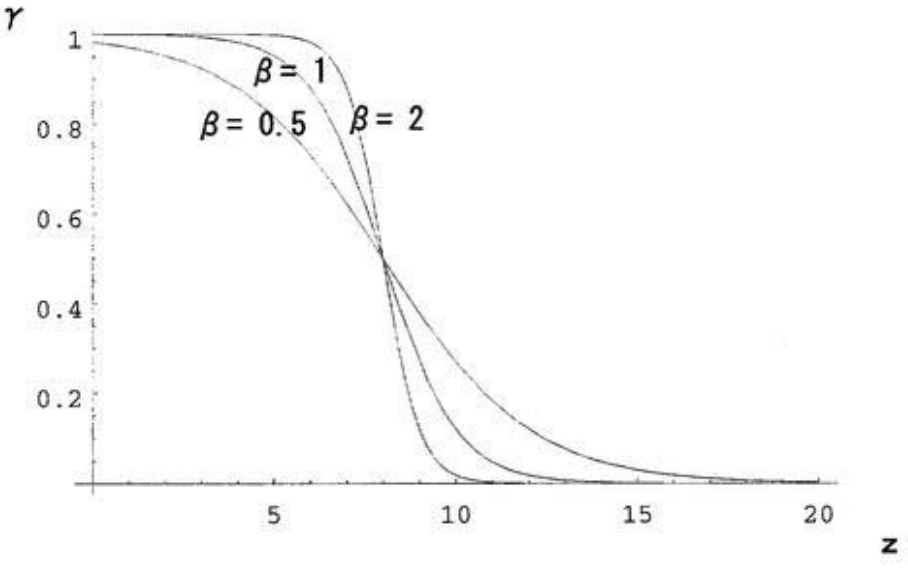
Soil	$\lvert IF_{\text{classic}}\rvert$	γ_0	Soil	$\lvert IF_{\text{classic}}\rvert$	γ_0
16	0.00162	1	18	0.16165	1
20	0.02565	1	9	0.18078	1
12	0.03021	1	11	0.19051	0.98581
3	0.04696	1	2	0.19452	1
5	0.05993	4.97×10^{-26}	8	0.20902	1
6	0.06288	1	7	0.23871	1
10	0.07236	1	1	0.27304	1
14	0.08564	1.49×10^{-39}	17	0.33459	1.02×10^{-7}
19	0.08949	1	13	0.45593	1.04×10^{-60}
15	0.15425	1	4	0.80959	1.72×10^{-44}

Table 4: Projected values of soil composition data by the classical PCA

Soil	1st PC	2nd PC	3rd PC	4th PC	Soil	1st PC	2nd PC	3rd PC	4th PC
1	-9.557	-2.203	0.587	-0.154	11	4.887	-3.090	-0.275	0.012
2	-13.070	-0.987	0.707	-0.034	12	-1.012	2.090	-1.209	0.183
3	-1.458	-2.536	-0.226	0.329	13	6.799	5.283	0.788	-0.162
4	13.073	-4.803	-0.692	-0.962	14	1.457	4.401	-1.658	0.416
5	-1.060	-4.233	-0.068	0.253	15	8.410	1.382	0.497	-0.096
6	-13.294	-0.122	0.067	0.381	16	-0.142	0.490	-0.632	-0.543
7	-10.997	1.708	-0.276	0.023	17	7.822	3.271	0.849	-0.555
8	14.680	-0.981	0.231	0.570	18	-10.269	0.861	0.630	-0.759
9	14.839	0.484	0.661	0.626	19	-12.242	0.195	0.104	0.586
10	3.337	-1.564	0.356	0.704	20	-2.203	0.355	-0.442	-0.816



a: β is fixed as 1.



b: η is fixed as 8.

Figure 1. Graphs of γ (z)

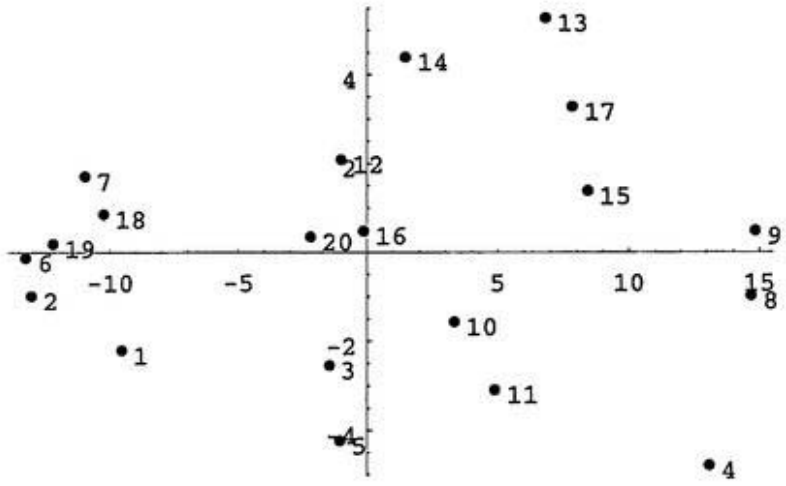
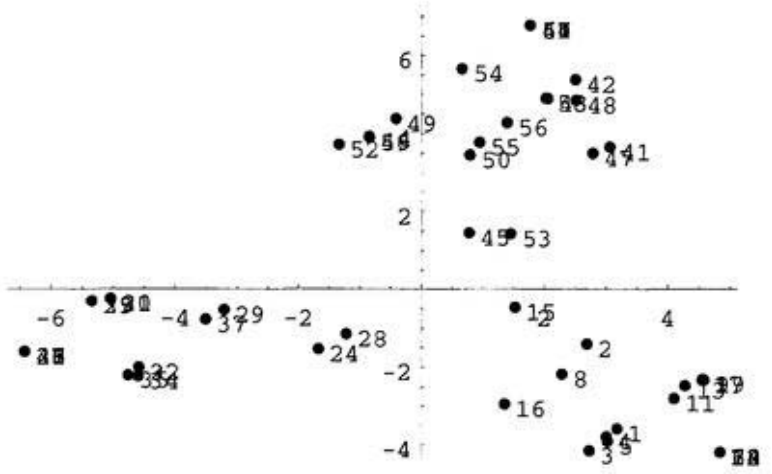
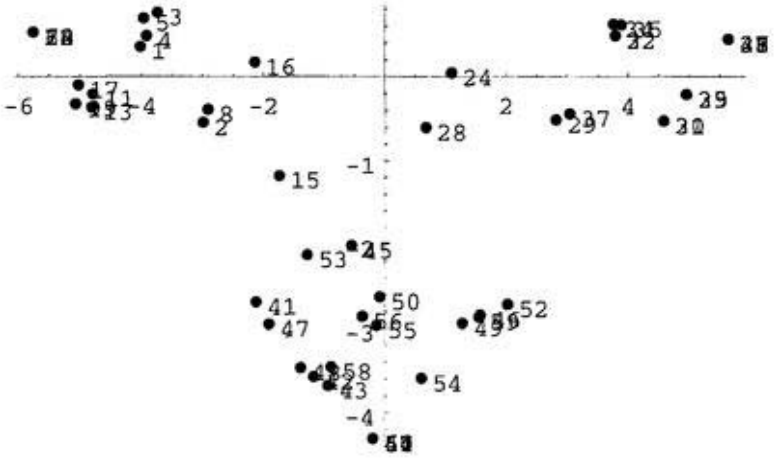


Figure 2



Classical Dual Scaling



Dual Scaling by Self-Organizing Rule

Figure 3