Deep Learning

TANIGUCHI Taichi

May 13, 2019

Contents

Part I
Notation

a: scalar
$\boldsymbol{a}$: Vector
$\boldsymbol{A}$: Matrix
$\tilde{\boldsymbol{A}}$: Tensors

ここでは、ベクトルも行列も体 R 上に定義されているものとする。ただし、C でも成立する場合には、ユニタリー行列や随伴行列が登場する。

Part II
Introduction

この章では、具体的な内容に入らず、AI や DEEP LEARNING のおおよそのイメージを作るためにある。読み飛ばしても問題はないと思われる。

長い間、人々は「考える機械」を作ることを夢見ていた。それは古代ギリシャまで遡る。今日では、artificial intelligence(AI) は多くの実用的な応用と活発な研究トピックで反映している分野である。

最近では、人間には知的に難しく、コンピューターには比較的簡単に解ける、数学的に表現される問題に急速に取り組まれている。しかし、人工知能は、人々が直観的に簡単に解ける問題で、解決するための形式的な問題説明が難しい問題が解決できなかった。この問題に有用なのが Deep Learning である。これは概念の階層という観点から経験を学び、世界を理解することを可能にする技術である。

概念がどのように相互に重なって構築されているかを示すグラフを描くと、グラフは深くなり、多くの層を持つようになる。これから、この技術は Deep Learning というように呼ばれる。

人口知能プロジェクトのいくつかに、正式な言語で世界についての知識をハードコーディングしようとしているものがある。コンピューターはこれらの正式な言語の記述について自動的に論理推論することができる。これを人口知能への知識ベースアプローチとして知られている。具体的な例としては、Cyc プロジェクトと呼ばれるものがある。これは、一般常識をデータベース化し、人間と同等の推論システムを構築することを目的としたプログラムが実施されている。

このようなハードコーディングされた知識に頼ったシステムが直面する困難は、容易に想像できる。作業の膨大さである。この困難を解決するために、AI システムが生データからパターンを抽出することによって、機械が知識を獲得する能力が必要になる。この技術をは Machine Learning(機械学習) と呼ばれる。簡単な機械学習のアルゴリズムの代表例は Logistic Regression(ロジスティック回帰) と呼ばれるものや、スパムメールのシステムに使われる naive Bayes と呼ばれるものがある。

1 Representation Learning

1.1 Future

機械学習の性能は、データの表現に大きく依存する。例えば、ロジスティック回帰は帝王切開の推薦に使われている。AI システムは直接は患者を検査できない。その代わり、ドクターは帝王切開に関連する情報をいくつかシステムに与える。このような情報を特徴という。ロジスティック回帰はこれらの相関のある特徴を学ぶ。ロジスティック回帰にはドクターのレポートより、MRI ののスキャンをデータとして与える方がよい予測結果を返す。他にも、話者の声の大きさの推定をする際、男性なのか、女性なのか、子供なのかという特徴に強く依存する。

多くの人工知能タスクには、特徴として何を選ぶかは非常に難しい。例えば、写真から車を検出するプログラムを書きたいとする。この時、特徴として何を選べばよいか。車にはタイヤがあるので、このタイヤを特徴として選べばよいかもしれない。しかし、正確にタイヤをピクセル値で表現することは難しい。タイヤはシンプルな幾何学的形状を有する。しかし、タイヤに落ちる影、タイヤの金属部分を照らす日光、タイヤの前衛に隠れているオブジェクト、などを理由に難しいことがわかる。

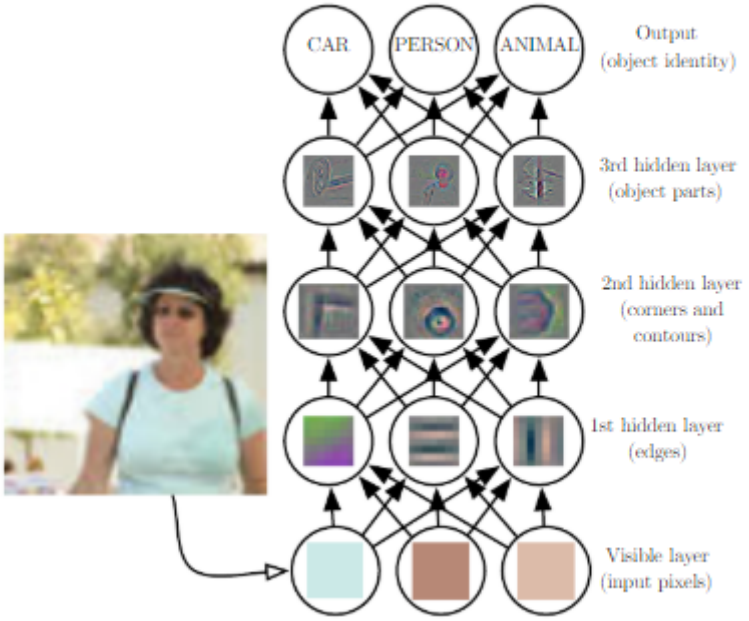
この問題を解決する方法として、データの表現から出力への写像だけでなく、表現自体も発見する方法がある。これを表現学習という。学習された特徴は、手動で設定するよりもよい結果を出すことが多い。Deep Learning は AI システムが最小限の人的介入で即座に新しいタスクに適応できることを可能にする。これまで、人々が手動で設定していた特徴もプログラムで可能にするのである。

1.2 表現学習

典型的な表現学習として、autoencoder というアルゴリズムがある。これは、データを別の表現に変換する encoder function と、新しい表現をオリジナルのフォーマットに戻す decoder function の結合からなる。autoencoder できる限り情報を保存するよに訓練されてから入力が encoder を通過し、次に decoder を通過する。その際、新しい表現に様々な優れた特性を持たせるように訓練されている。

特徴を学ぶためのアルゴリズムや、特徴を設計するとき、ゴールは観測データを説明する factors of variation を分けることである。ここでの factors は、影響限のことである。通常、factors は掛け算によって、結合されない。factors は直接観察されるものとは限らないが、factors は観察できるデータに影響を与えうるものである。

Deep Learning では、単純な概念から複雑な概念を構築することができる。以下の画像は Deep Learning が学習していく様子である。



単純な色の概念から複雑な概念を作り出していく様子である。

2 Model for Deep Learning

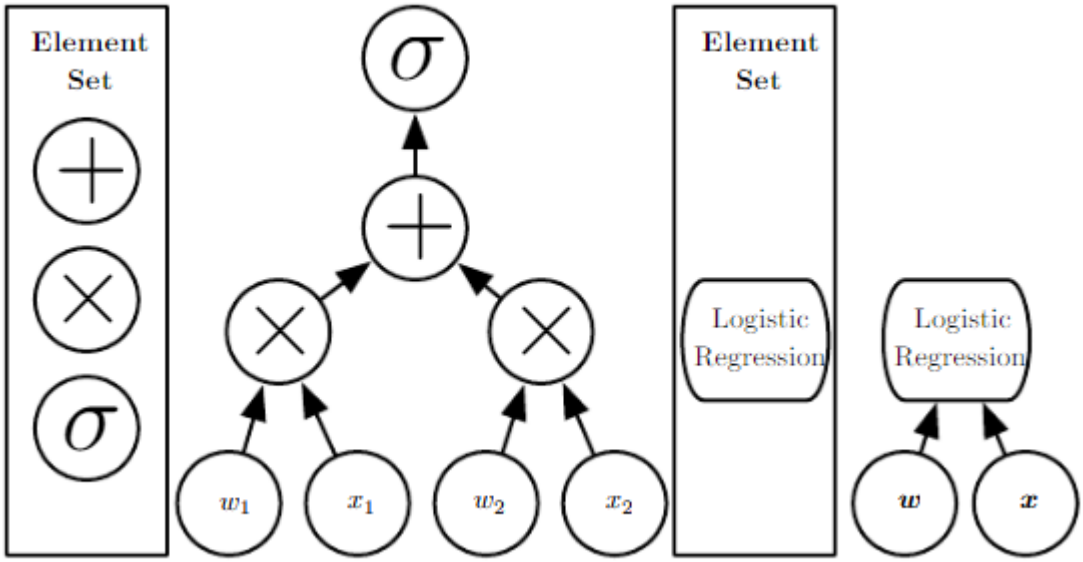
典型的な Deep Learning のモデルとして、feedforward deep network(フィードフォワードネットワーク)、multilayer perceptron(MLP) がある。MLP は、input のデータに対し、output の値を対応させる関数である。この関数はたくさんの簡単な関数で構成される。

Deep Learning では、上の画像を概念の層として理解することができる。それぞれ、Visible layer(入力層、出力層),hidden layer(隠れ層) からなる。隠れ層が二つ以上のものを Deep Learning と呼ぶ。

2.1 Depth of Deep Learning

モデルの深さの定義には、主に二つの方法が存在する。

まず第一に、深さを Computational graph の入力から出力までの最長経路の長さと定義することである。しかし、これは一つの計算ステップとして何を定義するかに依存する。

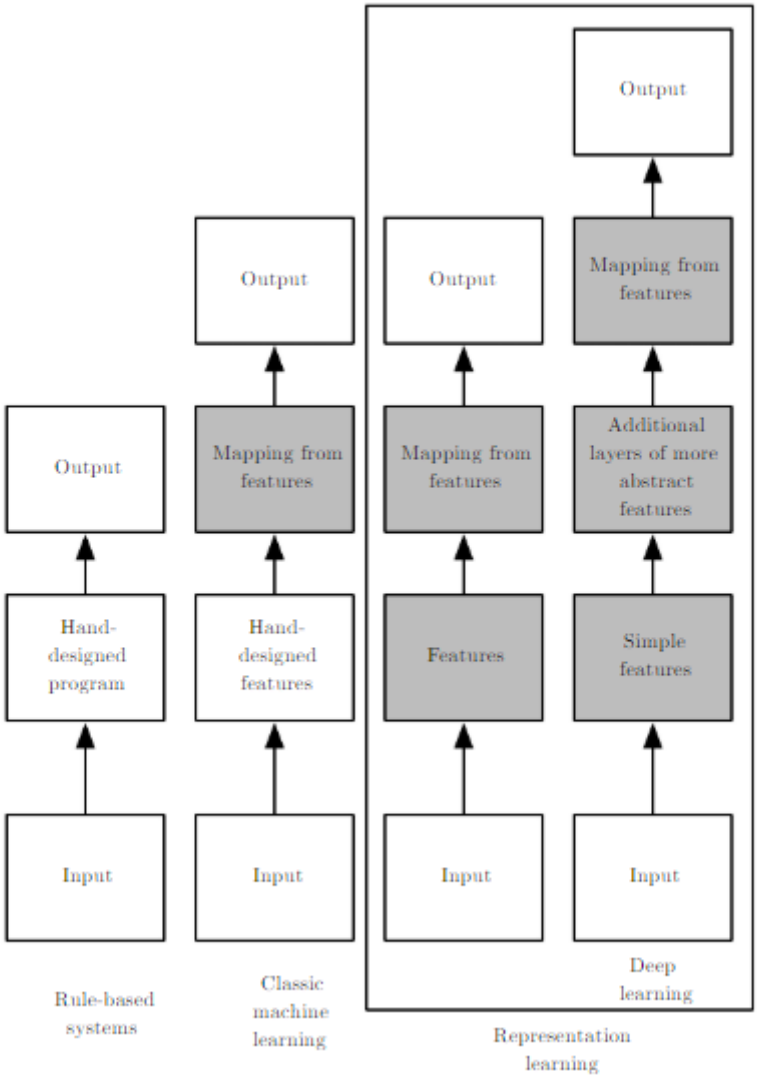


例えば、ロジスティック回帰で考えても、ロジスティック回帰を一つの要素として見るとモデルの深さは 1 になる。しかし、上の図の左図のように $w_1 \times x_1$、$w_2 \times x_2$ の計算でも一つの要素と考えると深さは 3 と見ることができる。

もう一つは、モデルの深さを計算グラフの深さではなく、概念が互いにどのように関連しているかを記述するグラフの深さであるとみなす。つまり各ノードが概念となっているグラフ (Probabilistic modeling graph) の深さと定義する。

どちらの定義を使うのが良いかは必ずしも明確ではない。正しいモデルの深さは一つではないのである。

Deep Learning とは AI への一種のアプローチであり、それは機械学習の一種でもある。具体的には、コンピューターが経験とデータによって向上することを可能にする手法である。Deep Learning は、現実世界を概念の入れ弧状の階層として表現し、各概念をより単純な概念に関連して定義し、抽象的な表現をより具体的なものに関して計算することによって、大きな力を発揮するアルゴリズムである。



この図では、Deep Learning の立ち位置を、ほかの機械学習と比べている。Rule-basedsystems では、ハードな実装 (手作業で、大量のデータを完全にデータベース化) を経て、output につながっている。Classicmachinelearning では、入力データに対し、特徴を手作業で設定する必要があった。表現学習、特に Deep Learning では、ハードな実装も必要とせず、特徴を自ら抽出し output につなげる技術である。

3 trend of Deep Learning

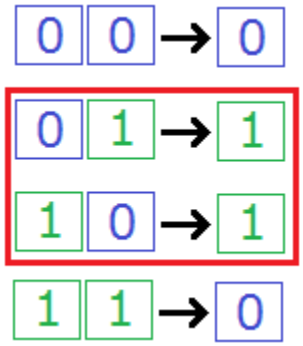
3.1 limitation of Linear model

$$f(\boldsymbol{x}, \boldsymbol{w}) = x_1 w_1 + x_2 w_2 + ... + x_n w_n$$

というモデルから、確率的勾配降下法によって、重み $\boldsymbol{w}$ を学習し、実数を $f(\boldsymbol{x})$ で予測するアルゴリズムを adaptive linear element(ADALINE) という。確率的勾配法を少し改良したものは最先端の Deep Learning 技術でも使われている (Adam など)。

ADALINE や perceptron は linear models(線形モデル) と呼ばれるようになった。線形モデルは現代でも比較的広い範囲で使われている有用な機械学習モデルである。

しかし、線形モデルには不便な点がある。それは、XOR 関数を学習できないことにある。XOR 関数とは、排他的論理和のことであり、以下の図の赤い部分のみ 1 を返すような関数である。



3.2 Reason of trend

Deep Learning はかなり昔からある技術だが、認められたのは最近のことである。なぜ、Deep Learning が認められなかったかは、様々な理由があるが、ここでは割愛する。

ここでは、最近になってなぜ、Deep Learning が認められたのかについて、二つの理由を挙げる。

- Big data の時代
- Model size の増加

3.2.1 Big data

Deep Learning には通常大量の学習データが必要になる。以前では、その学習データが少なく、実用的でなかった。しかし、現代では Big data の時代と呼ばれており、データ

が比較的多く手に入る時代にある。実際に企業でもデータは会社の財産として、高い価値のある物という印象を受ける。

bench mark set として、手書き文字のデータが何万もの数になる、MNIST というデータセットが有名である。

3.2.2 Model size

ニューラルネットワークはごく最近までニューロンの数が非常に少ない、つまり非常に小さいモデルでした。しかし、隠れユニットの導入以来、ニューラルネットワークのサイズは約 2.4 年ごとに倍増している。これは、より大きなメモリを備えた高速なコンピューターと、より大きなデータセットによって、作られたものである。

大規模なネットワークは、より複雑なタスクでも高い精度を達成する。
しかし、まだまだ最先端のネットワークでさえ、カエルのような比較的原始的な脊髄動物の神経系よりも小さい。
より早い CPU、汎用 GPU、分散コンピューティングなどの技術により、時間とともにモデルのサイズが大きくなることは明らかである。

4 Summarize

Deep Learning は過去数十年にわたって発展してきた人間の脳、統計学、応用数学に関する知識を重視した機械学習へのアプローチである。近年では、Deep Learning はより強力なコンピュータと、より大きなデータセット、およびより深い network をトレーニングするための技術の結果として、成長を遂げてきた。Deep Learning は、これをさらに改善し、新たなフロンティアにもたらすための課題と機会に満ち溢れている分野である。

Part III
Advance Preparation

ここでは、DEEP LEARNING に限らず、機械学習に必要な事前知識を解説する。各章は以下の通りになっている。まず初めに、線形代数、確率論、情報理論を解説する。そのあと、学習に必要になる数値計算、最後に機械学習の基本的な内容を解説する。

5 Linear Algebra

機械学習で扱うデータは必然的に高次元のデータ集合になる。よって、ベクトルや行列を用いて、目的関数と呼ばれるものを定義することが多い。線形代数を勉強することは、

機械学習を勉強するためには必須になるであろう。なお、スカラー、ベクトル、行列などは既知のものとする。

5.1 基本的な定義

行列の軸が二つより大きくしたいときがある。

定義 (Tensors)**.** 一般に軸が n 個ある配列を Tensors(テンソル) と定義します。$\tilde{\boldsymbol{A}}$ と書く。この場合、行列 $\boldsymbol{A}$ と表記を分けるために、チルダを付けることにする。

例えば、$n = 3$ の時、テンソル $\tilde{\boldsymbol{A}}$ の (i,j,k) 要素を $\tilde{\boldsymbol{A}}_{i,j,k}$ と書くことにする。

定義 (Hadamard product)**.** $\boldsymbol{M}(m,n)$ を m×n 行列全体の集合とする。この時、Hadamrd product という演算 $\odot : \boldsymbol{M}(m,n) \times \boldsymbol{M}(m,n) \to \boldsymbol{M}(m,n)$ を次のように定義する。

$$\forall \boldsymbol{A}, \boldsymbol{B} \in \boldsymbol{M}(m,n), \quad (\boldsymbol{A} \odot \boldsymbol{B})_{i,j} = \boldsymbol{A}_{i,j} \times B_{i,j}$$

定義 (Frobenius norm(フロベニウスノルム))**.** $\|\|_F : \boldsymbol{M}(n,m) \to \mathrm{R}$ を任意の $\boldsymbol{A} \in \boldsymbol{M}(n,m)$ に対して次のように定義する。

$$\|\boldsymbol{A}\|_F = \sqrt{\sum_{i,j} \boldsymbol{A}_{i,j}^2}$$

これは行列の Trace を使うと、以下のように定義することもできる。

$$\|\boldsymbol{A}\|_F = \sqrt{Tr(\boldsymbol{A}\boldsymbol{A}^T)}$$

5.2 Eigendecomposition(固有値分解)

固有値、固有ベクトルの定義については、省略する。

定理 **5.1** (固有値分解)**.** $\boldsymbol{A} \in \boldsymbol{M}(n,n)$ に対し、A の固有値 $\lambda_1, \lambda_2, ..., \lambda_n$ がすべて異なる時、行列 $\boldsymbol{A}$ を

$$\boldsymbol{A} = \boldsymbol{V} diag(\boldsymbol{\lambda}) \boldsymbol{V}^{-1}$$

と分解することができる。ただし、$\boldsymbol{V}$ は各列に固有ベクトルを並べたものであり、$\boldsymbol{\lambda}$ は固有値を並べたベクトルである。さらに、$diag(\boldsymbol{\lambda})$ はベクトル $\boldsymbol{\lambda}$ を対角成分に持つ行列である。

Proof. $\boldsymbol{A}\boldsymbol{V} = \boldsymbol{V} diag(\boldsymbol{V})$ となることと、各固有ベクトルが直行していることから明らかである。 □

固有値分解は、線形写像 $f(\boldsymbol{x}) = \boldsymbol{A}\boldsymbol{x}$ に対し、$\boldsymbol{A}$ の列空間の基底を、作用 $\boldsymbol{A}$ に対し、スカラー倍になるような基底に変換することに値する。

さらに、特異値分解というものがある。

定義 (特異値)**.** $\boldsymbol{A} \in \boldsymbol{M}(n, m)$ の特異値とは、

$$\boldsymbol{A}^T\boldsymbol{A}\text{もしくは、}\boldsymbol{A}\boldsymbol{A}^T\text{の固有値の正の平方根}$$

ここで、$\boldsymbol{A}^T\boldsymbol{A}$ は必ず正定置行列になるので、固有値は全て正。

定理 **5.2** (特異値分解定理)**.** $\boldsymbol{A} \in \boldsymbol{M}(m, n), \boldsymbol{U} \in \boldsymbol{M}(m, m), \boldsymbol{V} \in \boldsymbol{M}(n, n)$ とする。
この時、

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^T$$

とすることができる。

ただし、$\boldsymbol{D}$ は特異値を対角成分に並べた行列 *(*埋まらない対角成分は *0)* で、$\boldsymbol{U}$ の各列を左特異ベクトル、$\boldsymbol{V}$ の各列を右特異ベクトルという。
さらに、左特異ベクトルは $\boldsymbol{A}\boldsymbol{A}^T$ の固有ベクトルであり、右特異ベクトルは $\boldsymbol{A}^T\boldsymbol{A}$ の固有ベクトルである。通常、$\boldsymbol{U}, \boldsymbol{V}$

Proof. 初めに $m < n$ を仮定する。$\boldsymbol{A}^T\boldsymbol{A}$ の固有ベクトルを $\{\boldsymbol{u_1}, \boldsymbol{u_2}, ..., \boldsymbol{u}_n\}$, 固有値を $\lambda_1, \lambda_2, ..., \lambda_n$ とする。$\boldsymbol{A}^T\boldsymbol{A}$ は正定置対象行列であるので、固有値は必ず正で、二次形式は非負の値をとる。これは以下のように証明される。固有ベクトル、固有値の定義より、$\forall i \in \{1, 2, ..., n\}$,

$$\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{u}_i = \lambda_i\boldsymbol{u}_i$$

が満たされる。さらに、ここから両辺 $\boldsymbol{u}_i$ で内積をとると、

$$(\boldsymbol{u}_i, \boldsymbol{A}^T\boldsymbol{A}\boldsymbol{u}_i) = (\boldsymbol{u}_i, \lambda_i\boldsymbol{u}_i) = \lambda_i$$

任意の $\boldsymbol{x}$ について、

$$(\boldsymbol{x}, \boldsymbol{A}^T\boldsymbol{A}\boldsymbol{x}) = (\boldsymbol{A}\boldsymbol{x}, \boldsymbol{A}\boldsymbol{x}) = \|\boldsymbol{A}\boldsymbol{x}\|^2 \geq 0$$

よって、$\boldsymbol{A}^T\boldsymbol{A}$ は正定置対象行列である。
次に、$\lambda_i > 0,\ \ (i = 1, 2, ..., r)$ となる固有値と $\lambda_{r+1}, \lambda_{r+2}, ..., \lambda_n = 0$ となる固有値で分ける。
$\boldsymbol{v}_i \ \ for \ \ i = 1, 2, ..., r$ について、

$$\boldsymbol{v}_i = \frac{1}{\sqrt{\lambda_i}}\boldsymbol{A}\boldsymbol{u}_i$$

とすると $\boldsymbol{v}_i, \boldsymbol{v}_j$ は $i \neq j$ の時、直交する。(ここで、仮定 $m < n$) を用いている。よって、$\boldsymbol{v}_1, \boldsymbol{v}_2, ..., \boldsymbol{v}_r$ は m 次元ベクトル空間の r 次元部分空間の正規直交基底になっている。よって、$\boldsymbol{v}_1, ..., \boldsymbol{v}_m$ が m 次元ベクトル空間の正規直交基底になるように $\boldsymbol{v}_{r+1}, \boldsymbol{v}_{r+2}, ..., \boldsymbol{v}_m$

を加えることができる。次に、$\boldsymbol{V} = [\boldsymbol{v}_1, \boldsymbol{v}_2, ..., \boldsymbol{v}_m]$,$\boldsymbol{U} = [\boldsymbol{u}_1, \boldsymbol{u}_2, ..., \boldsymbol{u}_n]$ と定義する。
ここで、$\boldsymbol{V}^T\boldsymbol{A}\boldsymbol{U}$ という行列を考える。この行列の (i,j) 成分は $\boldsymbol{v}_i^T\boldsymbol{A}\boldsymbol{u}_j$ である。
$i = r+1, r+2, ..., n$ であれば、固有値は 0 なので、

$$\|\boldsymbol{A}\boldsymbol{u}_i\|^2 = (\boldsymbol{A}\boldsymbol{u}_i, \boldsymbol{A}\boldsymbol{u}_i) = (\boldsymbol{u}_i, \boldsymbol{A}^T\boldsymbol{A}\boldsymbol{u}_i) = \lambda_i\|\boldsymbol{u}_i\|^2 = 0$$

よって、$\boldsymbol{A}\boldsymbol{u}_i = 0$ となる。
$i, j = 1, 2, ..., r$ の場合、

$$\boldsymbol{v}_i^T\boldsymbol{A}\boldsymbol{u}_j = (\frac{1}{\sqrt{\lambda_i}}\boldsymbol{A}\boldsymbol{u}_i)^T\boldsymbol{A}\boldsymbol{u}_j = \frac{1}{\sqrt{\lambda_i}}\boldsymbol{u}_i^T\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{u}_j = \frac{\lambda_i}{\sqrt{\lambda_i}}\boldsymbol{u}_i^T\boldsymbol{u}_j = \sqrt{\lambda_i}\delta_{i,j}$$

ここで、$\delta_{i,j}$ はクロネッカーのデルタ。
i=r+1,r+2,...,m かつ j=1,2,...,r の場合、

$$\boldsymbol{v}_i^T\boldsymbol{A}\boldsymbol{u}_j = \boldsymbol{v}_i^T\sqrt{\lambda_j}\boldsymbol{v}_j = \sqrt{\lambda_j}(\boldsymbol{v}_i, \boldsymbol{v}_j) = 0$$

$n < m$ の時は、AA^T の固有ベクトル $v_1, .., v_n$ を使って同様に $u_1, ..., u_m$ を定めて同様の計算を行う。 □

5.3 一般逆行列

定理 **5.3** (一般逆行列)**.** $\boldsymbol{A} \in \boldsymbol{M}(m, n)$ に対して、次の四つの条件を満たす $\boldsymbol{A}^+ \in \boldsymbol{M}(n, m)$ がただ一つ存在する。

- $\boldsymbol{A}\boldsymbol{A}^+\boldsymbol{A} = \boldsymbol{A}$
- $\boldsymbol{A}^+\boldsymbol{A}\boldsymbol{A}^+ = \boldsymbol{A}^+$
- $(\boldsymbol{A}\boldsymbol{A}^+)^* = \boldsymbol{A}\boldsymbol{A}^+$
- $(\boldsymbol{A}^+\boldsymbol{A})^* = \boldsymbol{A}^+\boldsymbol{A}$

ただし、$\boldsymbol{A}^*$ は $\boldsymbol{A}$ の随伴行列を表す。

これは逆行列の一般化になっている。

定義 (Moore-Penrose Pseudoinverse(ムーンアペンローズ一般行列))**.** $\boldsymbol{A} \in \boldsymbol{M}(n, m)$ のムーンアペンローズ一般逆行列とは

$$\boldsymbol{A}^+ = \lim_{\alpha 0}(\boldsymbol{A}^T\boldsymbol{A} + \alpha\boldsymbol{I})^{-1}\boldsymbol{A}^T)$$

コンピューターに計算させるときは、アルゴリズム上定義より以下のような計算を行う。

$$\boldsymbol{A}^+ = \boldsymbol{V}\boldsymbol{D}^+\boldsymbol{U}^T$$

ここで、$\boldsymbol{U}, \boldsymbol{D}, \boldsymbol{V}$ は特異値分解によって、得られる行列である。
さらに、$\boldsymbol{D}^+$ は $\boldsymbol{D}$ の 0 以外の要素 (特異値) を逆数にした行列。$\boldsymbol{A}$ が行より列のほうが多い時、疑似逆行列を用いて、線形方程式を解くと、数ある解のうちの一つの解を得ることができる。さらにその解は、$\|x\|^2$ を最小にする解となる。
列よりも行のほうが大きいとき、疑似逆行列を用いて、線形方程式を解くと、$\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|$ を最小にする解が得られる。これはまさに最小二乗法の解である。

実際にこれが一般逆行列の条件を満たすのか確認しておく。

Proof. $\boldsymbol{A} \in \boldsymbol{M}(m,n) = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^{\boldsymbol{T}}$ に対して、$\boldsymbol{A} = \boldsymbol{V}\boldsymbol{D}^+\boldsymbol{U}^{\boldsymbol{T}}$ と置くと、

- $\boldsymbol{A}\boldsymbol{A}^+\boldsymbol{A} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^{\boldsymbol{T}}\boldsymbol{V}\boldsymbol{D}^+\boldsymbol{U}^{\boldsymbol{T}}\boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^{\boldsymbol{T}} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{D}^+\boldsymbol{D}\boldsymbol{V}^{\boldsymbol{T}} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^{\boldsymbol{T}}$
- $\boldsymbol{A}^+\boldsymbol{A}\boldsymbol{A}^+ = \boldsymbol{V}\boldsymbol{D}^+\boldsymbol{U}^{\boldsymbol{T}}\boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^{\boldsymbol{T}}\boldsymbol{V}\boldsymbol{D}^+\boldsymbol{U}^{\boldsymbol{T}} = \boldsymbol{V}\boldsymbol{D}^+\boldsymbol{D}\boldsymbol{D}^+\boldsymbol{U}^{\boldsymbol{T}} = \boldsymbol{V}\boldsymbol{D}^+\boldsymbol{U}^{\boldsymbol{T}}$
- $(\boldsymbol{A}\boldsymbol{A}^+)^T = (\boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^{\boldsymbol{T}}\boldsymbol{V}\boldsymbol{D}^+\boldsymbol{U}^{\boldsymbol{T}})^T = \boldsymbol{U}\boldsymbol{D}^+\boldsymbol{V}^{\boldsymbol{T}}\boldsymbol{V}\boldsymbol{D}\boldsymbol{U}^{\boldsymbol{U}}$
- $(\boldsymbol{A}^+\boldsymbol{A})^T = (\boldsymbol{V}\boldsymbol{D}^+\boldsymbol{U}^{\boldsymbol{T}}\boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^{\boldsymbol{T}}) = \boldsymbol{V}\boldsymbol{D}^+\boldsymbol{U}^{\boldsymbol{T}}\boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^{\boldsymbol{T}}$

□

5.4 Principal Components Analysis

ここでは、機械学習の簡単な例として Principal Components Analysis(PCA) を解説する。データポイント $\{\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, ..., \boldsymbol{x}^{(n)}\}$ を持っているとする。各 $\boldsymbol{x}^{(\boldsymbol{i})}$ は R^{n} の元とする。このデータを非可逆圧縮したい。つまり、次元圧縮がしたい。しかし、データの情報を失うことにもなるので、当然制度が落ちる可能性がある。制度をできるだけ落とさないような、次元圧縮を考える。

PCA は良い decoding function を選ぶ問題として定義される。特に、できるだけシンプルな decodeing function を選ぶことを考える。各、データポイント $\boldsymbol{x}_i \in \mathrm{R}^{\mathrm{n}}$ について、$\boldsymbol{c}_i \in \mathrm{R}^{\mathrm{l}}$ に対応させるような encodeing function,f と $\boldsymbol{x}_i \approx g(f(\boldsymbol{x}_{\boldsymbol{i}}))$ となるような decoding function,g を求めたい。シンプルな decoding function として、$g : \mathrm{R}^{\mathrm{l}} \to \mathrm{R}^{\mathrm{n}}$ として、$\forall \boldsymbol{c} \in \mathrm{R}^{\mathrm{l}},\ g(\boldsymbol{c}) = \boldsymbol{D}\boldsymbol{c}$ を選ぶことにする。問題を簡単にするために、$\boldsymbol{D}$ の各列ベクトルは互いに直行していると仮定する。

Input data$\boldsymbol{x}$ に対して、最適な $\boldsymbol{c}^*$ を対応させたい。一つの方法は、$\boldsymbol{x}$ と $g(\boldsymbol{c}^*)$ のノルムを最初にさせるような $\boldsymbol{c}^*$ を求めることである。つまり、以下の問題を考える。

$$\boldsymbol{c}^* \in \arg\min_{\boldsymbol{c} \in \mathrm{R}^{\mathrm{l}}} \|\boldsymbol{x} - g(\boldsymbol{c})\|$$

このままでは、$\boldsymbol{c}^*$ を求めにくいので、$\|\boldsymbol{x} - g(\boldsymbol{c})\|$ が非負で単調増加のことを利用して、以下の問題に書き換える。

$$\boldsymbol{c}^* \in \arg\min_{\boldsymbol{c}\in \mathrm{R}^{\mathrm{l}}} \|\boldsymbol{x} - g(\boldsymbol{c})\|^2$$

$$\begin{aligned} \|\boldsymbol{x} - g(\boldsymbol{c})\|^2 &= (\boldsymbol{x} - g(\boldsymbol{c}))^T(\boldsymbol{x} - g(\boldsymbol{c})) \\ &= \boldsymbol{x}^T\boldsymbol{x} - 2\boldsymbol{x}^T g(\boldsymbol{c}) + g(\boldsymbol{c})^T g(\boldsymbol{c}) \end{aligned}$$

$\boldsymbol{x}$ は定数なので、最適化問題は以下のように書き換えることができる。

$$\begin{aligned} & \boldsymbol{c}^* \in \arg\min_{\boldsymbol{c}\in \mathrm{R}^{\mathrm{l}}} -2\boldsymbol{x}^T g(\boldsymbol{c}) + g(\boldsymbol{c})^T g(\boldsymbol{c}) \\ = \ & \arg\min_{\boldsymbol{c}\in \mathrm{R}^{\mathrm{l}}} -2\boldsymbol{x}^T \boldsymbol{D}\boldsymbol{c} + (\boldsymbol{D}\boldsymbol{c})^T(\boldsymbol{D}\boldsymbol{c}) \\ = \ & \arg\min_{\boldsymbol{c}\in \mathrm{R}^{\mathrm{l}}} -2\boldsymbol{x}^T \boldsymbol{D}\boldsymbol{c} + \boldsymbol{c}^T\boldsymbol{D}^T\boldsymbol{D}\boldsymbol{c} \\ = \ & \arg\min_{\boldsymbol{c}\in \mathrm{R}^{\mathrm{l}}} -2\boldsymbol{x}^T \boldsymbol{D}\boldsymbol{c} + \boldsymbol{c}^T\boldsymbol{c} \end{aligned}$$

この最適化問題は、以下のようにして解くことができる。

$$\begin{aligned} \nabla(-2\boldsymbol{x}^T\boldsymbol{D}\boldsymbol{c} + \boldsymbol{c}^T\boldsymbol{c}) &= \boldsymbol{0} \\ -2\boldsymbol{D}^T\boldsymbol{x} + 2\boldsymbol{c} &= \boldsymbol{0} \\ \boldsymbol{c} &= \boldsymbol{D}^T\boldsymbol{x} \end{aligned}$$

以上より、encoding function,$f : \mathrm{R}^{\mathrm{n}} \to \mathrm{R}^{\mathrm{l}}$ を

$$\forall \boldsymbol{x} \in \mathrm{R}^{\mathrm{n}}, \quad f(\boldsymbol{x}) = \boldsymbol{D}^T\boldsymbol{x}$$

ここで、$\boldsymbol{r} : \mathrm{R}^{\mathrm{n}} \to \mathrm{R}^{\mathrm{n}}$ を次のように定義する。

$$\forall \boldsymbol{x} \in \mathrm{R}^{\mathrm{n}}, \qquad r(\boldsymbol{x}) = g(f(\boldsymbol{x})) = \boldsymbol{D}\boldsymbol{D}^T\boldsymbol{x}$$

次に、どのような $\boldsymbol{D}$(encoding matrix) にすればよいかを考える。
再び、$\boldsymbol{x}^{(i)}$ と $\boldsymbol{r}(\boldsymbol{x}^{(i)})$ をすべての i で近づけるような $\boldsymbol{D}^*$ を求める。すなわち、以下のような問題を考える。

$$\boldsymbol{D}^* \in \arg\min \sum_i \|\boldsymbol{x}^{(i)} - r(\boldsymbol{x}^{(i)})\|^2$$

$$sbject\ to\ \boldsymbol{D}^T\boldsymbol{D} = \boldsymbol{I}_l$$

$l = 1$ として、$\boldsymbol{D}$ を定義すると、$\boldsymbol{D}$ は縦ベクトルになる。よって、これを $\boldsymbol{d}$ と表記しなおすことにする。この時、上の式は以下のように書き直せる。

sbject to $\|\boldsymbol{d}^T\| = 1$ の元、

$$\begin{aligned}
& \boldsymbol{D}^* \in \arg\min \sum_i \|\boldsymbol{x}^{(i)} - r(\boldsymbol{x}^{(i)})\|^2 \\
= \ & \arg\min \sum_i \|\boldsymbol{x}^{(i)} - \boldsymbol{d}\boldsymbol{d}^T\boldsymbol{x}\|^2 \\
= \ & \arg\min \sum_i \|\boldsymbol{x}^{(i)} - \boldsymbol{d}^T\boldsymbol{x}\boldsymbol{d}\|^2 \\
= \ & \arg\min \sum_i \|\boldsymbol{x}^{(i)} - \boldsymbol{x}^T\boldsymbol{d}\boldsymbol{d}\|^2
\end{aligned}$$

ここで、データポイントを列に持つ行列 $\boldsymbol{X}$ を定義する。つまり、

$$\boldsymbol{X} = [\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, ..., \boldsymbol{x}^{(m)}] = \begin{bmatrix} \boldsymbol{x}_1^{(1)} & \boldsymbol{x}_1^{(2)} & \cdots & \boldsymbol{x}_1^{(m)} \\ \boldsymbol{x}_2^{(1)} & \boldsymbol{x}_2^{(2)} & \cdots & \boldsymbol{x}_2^{(m)} \\ \vdots & \vdots & \vdots & \vdots \\ \boldsymbol{x}_n^{(1)} & \boldsymbol{x}_n^{(2)} & \cdots & \boldsymbol{x}_n^{(m)} \end{bmatrix}$$

上の最適化問題は以下のように書き換えられる。

sbject to $\boldsymbol{D}^T\boldsymbol{D} = \boldsymbol{I}_l$ の元、

$$\begin{aligned}
& \boldsymbol{D}^* \in \arg\min \|\boldsymbol{X} - \boldsymbol{X}\boldsymbol{d}\boldsymbol{d}^T\|_F^2 \\
= \ & \arg\min Trace((\boldsymbol{X} - \boldsymbol{X}\boldsymbol{d}\boldsymbol{d}^T)^T(\boldsymbol{X} - \boldsymbol{X}\boldsymbol{d}\boldsymbol{d}^T)) \\
= \ & \arg\min Trace((\boldsymbol{X}^T - \boldsymbol{d}\boldsymbol{d}^T\boldsymbol{X}^{\boldsymbol{T}})(\boldsymbol{X} - \boldsymbol{X}\boldsymbol{d}\boldsymbol{d}^T)) \\
= \ & \arg\min Trace(\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{X} - \boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{X}\boldsymbol{d}\boldsymbol{d}^{\boldsymbol{T}} - \boldsymbol{d}\boldsymbol{d}^{\boldsymbol{T}}\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{X} + \boldsymbol{d}\boldsymbol{d}^{\boldsymbol{T}}\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{X}\boldsymbol{d}\boldsymbol{d}^{\boldsymbol{T}}) \\
= \ & \arg\min Trace(\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{X}) - Trace(\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{X}\boldsymbol{d}\boldsymbol{d}^{\boldsymbol{T}}) - Trace(\boldsymbol{d}\boldsymbol{d}^{\boldsymbol{T}}\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{X}) + Trace(\boldsymbol{d}\boldsymbol{d}^{\boldsymbol{T}}\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{X}\boldsymbol{d}\boldsymbol{d}^{\boldsymbol{T}}) \\
= \ & \arg\min -2Trace(\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{X}\boldsymbol{d}\boldsymbol{d}^{\boldsymbol{T}}) + Trace(\boldsymbol{d}\boldsymbol{d}^{\boldsymbol{T}}\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{X}\boldsymbol{d}\boldsymbol{d}^{\boldsymbol{T}}) \\
= \ & \arg\min -2Trace(\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{X}\boldsymbol{d}\boldsymbol{d}^{\boldsymbol{T}}) + Trace(\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{X}\boldsymbol{d}\boldsymbol{d}^{\boldsymbol{T}}\boldsymbol{d}\boldsymbol{d}^{\boldsymbol{T}}) \\
= \ & \arg\min -2Trace(\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{X}\boldsymbol{d}\boldsymbol{d}^{\boldsymbol{T}}) + Trace(\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{X}\boldsymbol{d}\boldsymbol{d}^{\boldsymbol{T}}) \\
= \ & \arg\min -Trace(\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{X}\boldsymbol{d}\boldsymbol{d}^{\boldsymbol{T}}) \\
= \ & \arg\max Trace(\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{X}\boldsymbol{d}\boldsymbol{d}^{\boldsymbol{T}}) \\
= \ & \arg\max Trace(\boldsymbol{d}^{\boldsymbol{T}}\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{X}\boldsymbol{d})
\end{aligned}$$

この最適化問題を解くために、$\boldsymbol{X}\boldsymbol{X}^{\boldsymbol{T}}$ の固有値分解を用いる。
すると、以下のように書き換えることができる。

$$\arg\max Trace(\boldsymbol{d}^{\boldsymbol{T}}\boldsymbol{V}\sum\boldsymbol{V}^{\boldsymbol{T}}\boldsymbol{d})$$

$$= \arg\max Trace((\boldsymbol{V^T d})^T \sum V^T d)$$

$$sbject\ to\ \boldsymbol{d^T d} = 1$$

$\boldsymbol{d}$ は m 次元ベクトル空間の元であり、$\boldsymbol{V}$ の各列はその正規直交基底を定める。よって、$\boldsymbol{d}$ を $\boldsymbol{v}_1, \boldsymbol{v}_2, ..., \boldsymbol{v}_m$ の一次結合で表す。よって、$a_1, a_2, ..., a_m \in \mathrm{R}$ を用いて、

$$\boldsymbol{d} = a_1\boldsymbol{v}_1 + a_2\boldsymbol{v}_2 + ... + a_m\boldsymbol{v}_n$$

と表す。すると、

$$\boldsymbol{V^T d} = \begin{bmatrix} a_1 \\ a_2 \\ \cdots \\ a_m \end{bmatrix}$$

であり、さらに、

$$\boldsymbol{d^T d} = a_1^2 + ... + a_m^2 = 1$$

以上より、

$$\arg\max Trace((\boldsymbol{V^T d})^T \sum V^T d)$$

$$= \arg\max \lambda_1 a_1^2 + ... + \lambda_m a_m^2$$

$$sbject\ to\ \boldsymbol{d^T d} = a_1^2 + ... + a_m^2 = 1$$

固有値分解をするとき、対応する固有値が大きい順になるように固有ベクトルを並べたとする。制約条件を a_1^2 について整理し、目的関数に代入すると、

$$\begin{aligned} &= \arg\max \lambda_1 a_1^2 + ... + \lambda_2 a_m^2 \\ &= \arg\max \lambda_1(1 - a_2^2 - a_3^2 - ... - a_m^2)a_1^2 + \lambda_2 a_2^2 + ... + \lambda_m a_m^2 \\ &= \arg\max \lambda_1 + (\lambda_2 - \lambda_1)a_2^2 + ... + (\lambda_n - \lambda_1)a_m^2 \end{aligned}$$

よって、$(\lambda_i - \lambda_1) \leq 0$ より、最大値は λ_1。よって、その時、

$$a_1 = 1, a_2 = 0, ..., a_m = 0$$

よって、

$$\boldsymbol{d} = \boldsymbol{v}_1$$

となることがわかる。つまり、$\boldsymbol{X^T X}$ の最大固有値に対応する固有ベクトルが最適解となる。

次に一般に $D : n \times l$ の時を考える。制約条件 $\boldsymbol{D^T D} = \boldsymbol{I}_n$ の元、

$$
\begin{aligned}
\boldsymbol{D}^* &\in \arg\min \sum_i \|\boldsymbol{x}^{(i)} - \boldsymbol{D}\boldsymbol{D}^{\boldsymbol{T}}\boldsymbol{x}^{(i)}\| \\
&= \arg\min \|\boldsymbol{X} - \boldsymbol{D}\boldsymbol{D}^{\boldsymbol{T}}\boldsymbol{X}\|_F^2 \\
&= \arg\min Trace((\boldsymbol{X} - \boldsymbol{D}\boldsymbol{D}^{\boldsymbol{T}}\boldsymbol{X})^T(\boldsymbol{X} - \boldsymbol{D}\boldsymbol{D}^{\boldsymbol{T}}\boldsymbol{X})) \\
&= \arg\min Trace((\boldsymbol{X}^{\boldsymbol{T}} - \boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{D}\boldsymbol{D}^T)(\boldsymbol{X} - \boldsymbol{D}\boldsymbol{D}^{\boldsymbol{T}}\boldsymbol{X})) \\
&= \arg\min Trace(\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{X} - \boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{D}\boldsymbol{D}^{\boldsymbol{T}}\boldsymbol{X} - \boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{D}\boldsymbol{D}^{\boldsymbol{T}}\boldsymbol{X} + \boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{D}\boldsymbol{D}^{\boldsymbol{T}}\boldsymbol{D}\boldsymbol{D}^{\boldsymbol{T}}\boldsymbol{X}) \\
&= \arg\min -Trace(\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{D}\boldsymbol{D}^{\boldsymbol{T}}\boldsymbol{X}) - Trace(\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{D}\boldsymbol{D}^{\boldsymbol{T}}\boldsymbol{X}) + Trace(\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{D}\boldsymbol{D}^{\boldsymbol{T}}\boldsymbol{D}\boldsymbol{D}^{\boldsymbol{T}}\boldsymbol{X}) \\
&= \arg\min -2Trace(\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{D}\boldsymbol{D}^{\boldsymbol{T}}\boldsymbol{X}) + Trace(\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{D}\boldsymbol{D}^{\boldsymbol{T}}\boldsymbol{X}) \\
&= \arg\min -Trace(\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{D}\boldsymbol{D}^{\boldsymbol{T}}\boldsymbol{X}) \\
&= \arg\max Trace(\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{D}\boldsymbol{D}^{\boldsymbol{T}}\boldsymbol{X}) \\
&= \arg\max Trace(\boldsymbol{D}^{\boldsymbol{T}}\boldsymbol{X}\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{D})
\end{aligned}
$$

再び、$\boldsymbol{X}\boldsymbol{X}^{\boldsymbol{T}}$ の固有値分解より、

$$
\begin{aligned}
&\arg\max Trace(\boldsymbol{D}^{\boldsymbol{T}}\boldsymbol{X}\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{D} \\
&= \arg\max Trace(\boldsymbol{D}^{\boldsymbol{T}}\boldsymbol{V}\sum\boldsymbol{V}^{\boldsymbol{T}}\boldsymbol{D}) \\
&= \arg\max Trace((\boldsymbol{V}^{\boldsymbol{T}}\boldsymbol{D})^{\boldsymbol{T}}\sum\boldsymbol{V}^{\boldsymbol{T}}\boldsymbol{D})
\end{aligned}
$$

$l = 1$ の時と同様に、$\boldsymbol{D}$ の各列ベクトルを $\boldsymbol{v}_1, \boldsymbol{v}_2, .., \boldsymbol{v}_m$ の一次結合で表す。

$$
\forall i \in 1, 2, .., m, \exists a_{i1}, ..., a_{in} \in \mathrm{R}, \qquad \boldsymbol{d}_i = a_{i1}\boldsymbol{v}_1 + a_{i2}\boldsymbol{v}_2 + ... + a_{in}\boldsymbol{v}_n
$$

上の時と同様に、

$$
\boldsymbol{V}^{\boldsymbol{T}}\boldsymbol{D} = \begin{bmatrix} \boldsymbol{v}_1^{\boldsymbol{T}}\boldsymbol{d}_1 & \boldsymbol{v}_!^{\boldsymbol{T}}\boldsymbol{d}_2 & \vdots & \boldsymbol{v}_1^{\boldsymbol{T}}\boldsymbol{d}_m \\ \cdots & \cdots & \cdots & \cdots \\ \boldsymbol{v}_m^{\boldsymbol{T}}\boldsymbol{d}_1 & \boldsymbol{v}_m^{\boldsymbol{T}}\boldsymbol{d}_2 & \vdots & \boldsymbol{v}_m^{\boldsymbol{T}}\boldsymbol{d}_m \end{bmatrix}
$$

よって、これの (i,j) 成分は $\boldsymbol{v}^{\boldsymbol{T}}\boldsymbol{d}_j = \boldsymbol{v}_i^{\boldsymbol{T}}(a_{j1}\boldsymbol{v}_1 + a_{j2}\boldsymbol{v}_2 + ... + a_{jm}\boldsymbol{v}_m) = a_{ji}$

以上より、

$$
Trace((\boldsymbol{V}^{\boldsymbol{T}}\boldsymbol{D})^{\boldsymbol{T}}\sum\boldsymbol{V}^{\boldsymbol{T}}\boldsymbol{D}) = \sum_{i_1}\lambda_{i_1}a_{1,i_1} + \sum_{i_2}\lambda_{i_2}a_{2,i_2} + ... + \sum_{i_m}\lambda_{i_m}a_{m,i_m}
$$

$l = 1$ のときで見たように、$\sum_{i_1}\lambda_{i_1}a_{1,i_1}$ を最大にする $\boldsymbol{d}_1$ は最大固有値に対応する固有ベクトル $\boldsymbol{v}_1$ だった。

よって、$\sum_{i_1}\lambda_{i_1}a_{1,i_1}$ を引いた目的関数を考えると、次に大きい固有値に対応する固有ベ

クトルが $\boldsymbol{d}_2$ の最適解となる。これを繰り返すことにより、m 個ある固有ベクトルのうち、大きい順に並べた固有値に対応する固有ベクトルを l 個並べたものが D である。

以上により、最適な D が求まったことになる。結局、$\boldsymbol{X}\boldsymbol{X}^T$ の固有値が大きい l 個の固有ベクトルを列に持つ行列を $\boldsymbol{D}^*$ に選ぶ。

6 Probability

確率変数や、確率密度、確率分布、期待値、分散、共分散などの定義は既知のものとする。

6.1 ベルヌーイ分布

ベルヌーイ分布とは、$\phi \in [0,1], k \in 0,1$ として、

$$P(X=k) = \phi^k (1-\phi)^{1-k}$$

という分布である。

期待値、分散は

$$\mathrm{E}_{\boldsymbol{x}}[x] = \phi$$

$$\mathrm{Var}_{\boldsymbol{x}[x]=\phi(1-\phi)}$$

6.2 ガウス分布

6.2.1 一変数ガウス分布

σ は標準偏差とすると一般的なガウス分布の確率密度関数は次のような式になる。

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{(x-\mu)^2}{2\sigma^2})$$

6.2.2 標準多変数ガウス分布

次に確率変数がいくつかある多変量正規分布を考える。確率変数ベクトルを以下とする。

$$X = \begin{pmatrix} X_1 \\ X_2 \\ X_3 \\ . \\ . \\ . \\ X_n \end{pmatrix}$$

この時それぞれの確率変数は独立に標準ガウス分布に従うので確率密度関数は

$$f(X_i) = \frac{1}{\sqrt[2]{2\pi}} \exp(-\frac{X_i^2}{2})$$

となる。さらに独立な確率分布に従う確率変数の同時密度関数は、それぞれの確率密度関数の積で表されるので

$$f(X_1, X_2, ..., X_n) = \prod_{i=1}^{n} \frac{1}{\sqrt[2]{2\pi}} \exp(-\frac{X_i^2}{2})$$

$$= \frac{1}{(\sqrt[2]{2\pi})^n} \exp(-\sum_{i=1}^{n} \frac{(X_i)^2}{2})$$

$$= \frac{1}{(\sqrt[2]{2\pi})^n} \exp(-\frac{X^T X}{2})$$

これが標準多変量ガウス分布となる。

6.2.3 多変量正規分布

次に多変量正規分布の定義は、ベクトル X が n 次元の標準多変量正規分布に従うとき X を一次変換し、定数ベクトルを足した行列が従う分布のことを標準正規分布という。つまり、

$$X = \begin{pmatrix} X_1 \\ X_2 \\ \vdots \\ X_n \end{pmatrix}, Z = \begin{pmatrix} Z_1 \\ Z_2 \\ \vdots \\ Z_n \end{pmatrix}, \mu = \begin{pmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_n \end{pmatrix}$$

A を正則で $n * n$ 行列とする。この時

$$Z = AX + \mu$$

を Z の従う分布を多変量正規分布という。$X = A^{-1}(Z - \mu)$ と変形すると

$$dx_1, dx_2, ...dx_n = |A|^{-1} dz_1, dz_2, ...dz_n$$

であるので $X = (Z - \mu)A^{-1}$ を標準多変量正規分布の密度関数に代入すると

$$f(X_1, X_2, ...X_n)dx_1dx_2...dx_n = \frac{1}{(\sqrt[2]{2\pi})^n} \exp(-\frac{1}{2}(z-\mu)^T(AA^T)^{-1}(z-\mu))|A|^{-1}dz_1dz_2...dz_n$$

これが多変量ガウス分布の密度関数になる。z 平均と共分散を求めると

$$E[Z] = AE[X] + \mu = 0 + \mu$$

$$V[Z] = AV[X]A^T + 0 = AI_nA^T = AA^T$$

分散を Σ とすると A が正値対象行列なので

$$||\Sigma|| = ||A|| * ||A^T|| = ||A||^2 \iff ||A|| = \sqrt{|\Sigma|}$$

多変量ガウス分布の密度関数は次になる

$$g(z_1, z_2, ..., z_n) = \frac{1}{(\sqrt[2]{2\pi})^n \sqrt{|\Sigma|}} \exp(-\frac{1}{2}(z-\mu)^T(\Sigma)^{-1}(z-\mu))$$

6.2.4 多変量ガウス分布から多変量標準ガウス分布へ

定理 **6.1.** ある $\mu \in \mathbf{R^n}$ と $\sum \in \mathbf{S^n}_{++}$ に対して $X \sim N(\mu, \sum)$ とすると、$B \in \mathbf{R^{nxn}}$ が存在し次を満たす、$Z = B^{-1}(X - \mu)$ とすると $Z \sim N(0, 1)$

上記の証明は次の 2 つのパートで構成される

- 共分散行列 $\sum$ がある正則行列 B を用いて $\sum = BB^T$ と分解できる (コレスキー分解)
- $Z = B^{-1}(X - \mu)$ を用いて X から異なるベクトルランダム変数 Z を作る

まずは共分散行列の分解を考える。

線形代数から対称行列に関する以下の 2 つの性質を思い出そう

1. 任意の実対称行列 $A \in \mathbb{R}^{nxn}$ は常に $A = U\Lambda U^T$ で表される、ここで U はフルランク直行行列であり、その列に A の固有ベクトルを持ち Λ は A の固有値を持つ対角行列である
2. もし A が正定値対称行列なら、すべての固有値は正である

共分散行列 $\sum$ は正定値なので上の 1 を使うことによりある適当な行列 U, Λ を用いて $\sum = U\Lambda U^T$ と書ける。上の 2 を使うことにより $\Lambda^{\frac{1}{2}} \in \mathbb{R}^{nxn}$ を Λ の各要素の二乗根に対応する要素として持つ対角行列とできる
$\Lambda = \Lambda^{1/2}(\Lambda^{1/2})^T$ なので

$$\sum = U\Lambda U^T = U\Lambda^{1/2}(\Lambda^{1/2})^T U^T = U\Lambda^{1/2}(U\Lambda^{1/2})^T = BB^T$$

を得る、ここで $B = U\Lambda^{1/2}$ である。この場合、$\sum^{-1} = B^{-T}B^{-1}$ なので多変量ガウス分布の式を次のように書き換えらえる

$$p(x; \mu, \sum) = \frac{1}{(2\pi)^{n/2}|BB^T|^{1/2}} \exp\{-\frac{1}{2}(x-\mu)^T B^{-T} B^{-1}(x-\mu)\}$$

次は変数変換について考える。

ベクトル値ランダム変数を $Z = B^{-1}(X - \mu)$ と定義する。確率論の基本式はベクトル値ランダム変数に関連した変数変換である

$X = [X_1 \cdots X_n]^T \in \mathbb{R}^{\kappa}$ が同時密度関数 $f_X : \mathbf{R^n} \to \mathbf{R}$ によるベクトル値ランダム変数と仮定する。もし $Z = H(X) \in \mathbf{R}$ ただし H は全単射で微分可能関数なら Z は次の同時密度関数 $f_Z : \mathbf{R^n} \to \mathbf{R}$ を持つ

$$f_Z(z) = f_X(x) \cdot |\det \begin{pmatrix} \frac{\partial x_1}{\partial z_1} & \cdots & \frac{\partial x_1}{\partial z_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial x_n}{\partial z_1} & \cdots & \frac{\partial x_n}{\partial z_n} \end{pmatrix}|$$

変数変換式を用いることでベクトル変数 Z が次の同時密度を持つことがわかる

$$p_Z(z) = \frac{1}{(2\pi)^{n/2}} \exp(-\frac{1}{2} z^T z)$$

これはまさしく $Z \sim N(0, 1)$ である

6.2.5 条件付きガウス分布

2 つの変数集合 x_a, x_b の同時分布がガウス分布に従うなら、もう一方の集合の条件付き分布もガウス分布になる。条件付き分布 $p(x_a|x_b)$ がガウス分布だと示すために、同時分布のガウス分布の指数部を見る。多変量ガウス分布は指数部分が二次形式なのでここ見れば良い。今、D 次元ベクトル x を多変量ガウス分布 $N(x|\mu, \Sigma)$ に従うものとする。これを 2 つの互いに素な部分集合 x_a, x_b に分割する。ただし x_a は最初の M 個の要素、x_b は残りの D-M 個の要素で構成されるとしても一般性は失われない。平均ベクトル、共分散行列も同様に定義する。

$$\boldsymbol{x} = \begin{pmatrix} x_a \\ x_b \end{pmatrix} \boldsymbol{\mu} = \begin{pmatrix} \mu_a \\ \mu_b \end{pmatrix} \Sigma = \begin{pmatrix} \Sigma_{aa} & \Sigma_{ab} \\ \Sigma_{ba} & \Sigma_{bb} \end{pmatrix}$$

なお共分散行列の対称性 $\Sigma^T = \Sigma$ より同様に Σ_{aa}, Σ_{bb} も対称であるが $\Sigma_{ba}^T = \Sigma_{ab}$ である。共分散の逆行列を考えると今後の話がスムーズになるので

$$\Lambda = \Sigma^{-1}$$

とし、またこれを精度行列という。上記のベクトル x の分割に対応する分割された精度行列を以下で定義する。

$$\Lambda = \begin{pmatrix} \Lambda_{aa} & \Lambda_{ab} \\ \Lambda_{ba} & \Lambda_{bb} \end{pmatrix}$$

対称行列の逆行列は対称行列なので $\Lambda_{aa}, \Lambda_{bb}$ はそれぞれ対称だが $\Lambda_{ab}^T = \Lambda_{ba}$ である。以上の表記を用いると

$$\begin{aligned} -\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu) &= -\frac{1}{2}(x_a-\mu_a)^T\Lambda_{aa}(x_a-\mu_a) - \frac{1}{2}(x_a-\mu_a)^T\Lambda_{ab}(x_b-\mu_b) \\ &\quad -\frac{1}{2}(x_b-\mu_b)^T\Lambda_{ba}(x_a-\mu_b) - \frac{1}{2}(x_b-\mu_b)^T\Lambda_{bb}(x_b-\mu_b) \\ &= -\frac{1}{2}x_a^T\Lambda_{aa}x_a + x_a^T\{\Lambda_{aa}\mu_a - \Lambda_{ab}(x_b-\mu_b)\} + const \qquad (1) \end{aligned}$$

(const は x_a と独立な項) では (6) を用いて条件付き分布の $\mu_{a|b}, \sum_{a|b}$ それぞれ求めよう。多変量ガウス分布の指数部分を平方完成すると

$$-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu) = -\frac{1}{2}x^T\Sigma^{-1}x + x^T\Sigma^{-1}\mu + const$$

なので (const は x と独立な項) 条件付き分布の場合は以下のようになる

$$-\frac{1}{2}(x_a-\mu_{a|b})^T\Sigma_{a|b}^{-1}(x_a-\mu_{a|b}) = -\frac{1}{2}x_a^T\Sigma_{a|b}^{-1}x_a + x_a^T\Sigma_{a|b}^{-1}\mu_{a|b} + const$$

(6) と係数を比較すると以下がそれぞれ得られる

$$\Sigma_{a|b} = \Lambda_{aa}^{-1}$$

$$\begin{aligned} \mu_{a|b} &= \Sigma_{a|b}\{\Lambda_{aa}\mu_a - \Lambda_{ab}(x_b-\mu_b)\} \qquad (2) \\ &= \mu_a - \Lambda_{aa}^{-1}\Lambda_{ab}(x_b-\mu_b) \qquad (3) \end{aligned}$$

ここからは $\Lambda_{aa}, \Lambda_{ab}$ を先ほどの分割 $\sum$ を使って表すことを目標にする。具体的には次の公式を用いる。

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix}^{-1} = \begin{pmatrix} M & -MBD^{-1} \\ -D^{-1}CM & D^{-1} + D^{-1}CMBD^{-1} \end{pmatrix}$$

ただし $M=(A-BD^{-1}C)^{-1}$ であり、M^{-1} を部分行列 D に関するシューア補行列という。この定理と以下の等式を利用すると

$$\begin{pmatrix} \Sigma_{aa} & \Sigma_{ab} \\ \Sigma_{ba} & \Sigma_{bb} \end{pmatrix}^{-1} = \begin{pmatrix} \Lambda_{aa} & \Lambda_{ab} \\ \Lambda_{ba} & \Lambda_{bb} \end{pmatrix}$$

次がすぐにわかる。

$$\mu_{a|b} = \mu_a + \Sigma_{ab}\Sigma_{bb}^{-1}(x_b - \mu_b) \qquad (*)$$

$$\Sigma_{a|b} = \Sigma_{aa} - \Sigma_{ab}\Sigma_{bb}^{-1}\Sigma_{ba} \qquad (**)$$

以上のまとめとしては 2 つの変数集合 x_a, x_b の同時分布がガウス分布に従うなら、もう一方の集合の条件付き分布 $p(x_a|x_b)$ もガウス分布になり、その平均と共分散はそれぞれ上記の $*, **$ で決定される。

6.2.6 周辺ガウス分布

同時分布 $p(x_a, x_b)$ がガウス分布ならば条件付き分布 $p(x_a|x_b)$ もガウス分布になることがわかった。ここではその周辺分布

$$p(x_a) = \int p(x_a, x_b)dx_b$$

も同様にガウス分布であることを示す。積分処理のために同時分布の x_b に関する項は

$$-\frac{1}{2}\Lambda_{bb}x_b + x_b^T m = -\frac{1}{2}(x_b - \Lambda_{bb}^{-1}m)^T\Lambda_{bb}(x_b - \Lambda_{bb}^{-1}m) + \frac{1}{2}m^T\Lambda_{bb}^{-1}m \qquad (***)$$

ただし m は

$$m = \Lambda_{bb}\mu_b - \Lambda_{ba}(x_a - \mu_a)$$

とする。この式 $(***)$ の右辺の第 1 項はガウス分布の標準的な二次形式部分に相当する。そして第 2 項は x_a には依存するが x_b には依存しない。よって第 1 項のみに注目し x_b で積分すると次の形式になる。

$$\int \exp\{-\frac{1}{2}(x_b - \Lambda_{bb}^{-1}m)^T\Lambda_{bb}(x_b - \Lambda_{bb}^{-1}m)\}dx_b$$

よって定数化できなかったつまり x_a に依存する項は

$$-\frac{1}{2}x_a^T\Lambda_{aa}x_a + x_a^T\{\Lambda_{aa}\mu_a + \Lambda_{ab}\mu_b\} + \frac{1}{2}m^T\Lambda_{bb}^{-1}m \qquad (*x_a*)$$

であり先ほどの m を代入して x_a に依存する項のみに注目し計算すると $(\Lambda_{ba}^T = \Lambda_{ab})$

$$\begin{aligned} *x_a* &= -\frac{1}{2}x_a^T\Lambda_{aa}x_a + x_a^T\{\Lambda_{aa}\mu_a + \Lambda_{ab}\mu_b\} \\ &\quad +\frac{1}{2}(\Lambda_{bb}\mu_b - \Lambda_{ba}(x_a - \mu_a))^T\Lambda_{bb}^{-1}(\Lambda_{bb}\mu_b - \Lambda_{ba}(x_a - \mu_a)) \\ &= -\frac{1}{2}x_a^T(\Lambda_{aa} - \Lambda_{ab}\Lambda_{bb}^{-1}\Lambda_{ba})x_a + x_a^T(\Lambda_{aa}\mu_a - \Lambda_{ab}\Lambda_{bb}^{-1}\Lambda_{ba}\mu_a) + const \end{aligned}$$

ただし const は x_a に依存しない定数を表す。
よって周辺分布 $p(x_a)$ の共分散は 2.3.1 と同様にして

$$\Sigma_a = (\Lambda_{aa} - \Lambda_{ab}\Lambda_{bb}^{-1}\Lambda_{ba})^{-1} \tag{$*1$}$$

$$\mu_a = \Sigma_a(\Lambda_{aa} - \Lambda_{ab}\Lambda_{bb}^{-1}\Lambda_{ba})\mu_a \tag{$*2$}$$

がわかる。Λ を Σ を使って書き直すため

$$\begin{pmatrix} \Lambda_{aa} & \Lambda_{ab} \\ \Lambda_{ba} & \Lambda_{bb} \end{pmatrix}^{-1} = \begin{pmatrix} \Sigma_{aa} & \Sigma_{ab} \\ \Sigma_{ba} & \Sigma_{bb} \end{pmatrix}$$

に対して先ほどの定理を用いると

$$\Sigma_{aa} = (\Lambda_{aa} - \Lambda_{ab}\Lambda_{bb}^{-1}\Lambda_{ba})^{-1}$$

ともとまるのでこれを $(*1), (*2)$ に代入すると周辺分布 $p(x_a)$ の平均と共分散はそれぞれ

$$E(x_a) = \mu_a$$

$$cov(x_a) = \Sigma_{aa}$$

と表され分割された精度行列について非常に簡単に表現されることがわかった。

6.2.7 ガウス変数に対するベイズの定理

今回は、条件付き分布、周辺分布がガウス分布だったときに同時分布がどのようになるか見ていく
今まで同時分布 p(x,y) がガウス分布だった時に条件付き分布 p(y|x) 、周辺分布 p(x) がどのようになるか計算してきた

あるガウス分布 $p(x)$ と、平均が x の線形関数で、共分散は x とは独立であるようなガウス条件付き分布 $p(y|x)$ が与えられているとする。これはのちに確認する線形ガウスモデルの一例となっている。このとき、周辺分布 $p(y)$ と条件付き分布 $p(x|y)$ を求める。
周辺分布と条件付き分布を

$$p(x) = N(x|\mu, \Lambda^{-1})$$

$$p(y|x) = N(y|Ax + b, L^{-1})$$

とする。ただし、μ, A, b は平均に関係したパラメータで Λ, L は精度行列である。x が M 次元で y が D 次元であるなら行列 A の大きさは $D \times M$ となる。
x と y 上の同時分布の表現を求める。このため次のよに定義する。

$$z = \begin{pmatrix} x \\ y \end{pmatrix}$$

そして、同時分布の対数を考える。

$$\begin{aligned} lnp(z) &= lnp(x) + lnp(y|x) \\ &= -\frac{1}{2}(x-\mu)^T\Lambda(x-\mu) \\ &\quad -\frac{1}{2}(y-Ax-b)^T L(y-Ax-b) + const \end{aligned} \tag{4}$$

ただし、const は x や y とは独立な定数である。そして ln は底を e とする対数である。また、以前に述べたように、これは z の要素の二次形式なので $p(z)$ もガウス分布となる。このガウス分布の制度行列を求めるために、(9) の二次の項について考察する。この項は次のようにる。

$$\begin{aligned} & -\frac{1}{2}(\Lambda + A^T LA)x - \frac{1}{2}y^T Ly + \frac{1}{2}y^T LAx + \frac{1}{2}x^T A^T Ly \\ = & -\frac{1}{2}\begin{pmatrix} x \\ y \end{pmatrix}^T \begin{pmatrix} \Lambda + A^T LA & -A^T L \\ -LA & L \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} \\ = & -\frac{1}{2}z^T Rz \end{aligned} \tag{5}$$

よって、z 上のガウス分布の精度行列（逆共分散行列）は

$$R = \begin{pmatrix} \Lambda + A^T LA & -A^T L \\ -LA & L \end{pmatrix}$$

になる。共分散行列は行列のシューア分解を適用して精度行列の逆行列を求めることで得られる。まずはシューア補行列を求めよう

$$\begin{aligned} M &= (\Lambda + A^T LA - (-A^T L)L^{-1}(-LA))^{-1} \\ &= (\Lambda + A^T LA - A^T LA)^{-1} = \Lambda^{-1} \end{aligned}$$

これより

$$\begin{aligned} cov(z) = R^{-1} &= \begin{pmatrix} \Lambda^{-1} & -\Lambda^{-1}(-A^T L)L^{-1} \\ -L^{-1}(-LA)\Lambda^{-1} & L^{-1} + L^{-1}(-LA)\Lambda^{-1} \end{pmatrix} \\ &= \begin{pmatrix} \Lambda^{-1} & \Lambda^{-1}A^T \\ A\Lambda^{-1} & L^{-1} + A\Lambda^{-1}A^T \end{pmatrix} \end{aligned} \tag{6}$$

同様に z 上のガウス分布の平均は (9) の線形の項を調べることで

$$x^T\Lambda\mu - x^TA^TLb + y^TLb = \begin{pmatrix} x \\ y \end{pmatrix}^T \begin{pmatrix} \Lambda\mu - A^TLb \\ Lb \end{pmatrix}$$

で与えられる、前回同様、多い変量ガウス分布の 2 次形式部分を平方完成して係数比較することで

$$\begin{aligned} E(z) &= R^{-1} \begin{pmatrix} \Lambda\mu - A^TLb \\ Lb \end{pmatrix} \\ &= \begin{pmatrix} \Lambda^{-1} & \Lambda^{-1}A^T \\ A\Lambda^{-1} & L^{-1} + A\Lambda^{-1}A^T \end{pmatrix} \begin{pmatrix} \Lambda\mu - A^TLb \\ Lb \end{pmatrix} \\ &= \begin{pmatrix} \Lambda^{-1}(\Lambda\mu - A^TLb) + \Lambda^{-1}A^TLb \\ A\Lambda^{-1}(\Lambda\mu - A^TLb) + (L^{-1} + A\Lambda^{-1}A^T)Lb \end{pmatrix} \\ &= \begin{pmatrix} \mu - \Lambda^{-1}A^TLb + \Lambda^{-1}A^TLb \\ A\Lambda^{-1}\Lambda\mu - A\Lambda^{-1}A^TLb + L^{-1}Lb + A\Lambda^{-1}A^TLb \end{pmatrix} \\ &= \begin{pmatrix} \mu \\ A\mu + b \end{pmatrix} = \begin{pmatrix} E(x) \\ E(y) \end{pmatrix} \end{aligned}$$

ここで (11) と共分散行列の定義より周辺分布 $p(y)$ の平均また共分散はそれぞれ以下のように表される

$$E(y) = A\mu + b$$
$$cov(y) = L^{-1} + A\Lambda^{-1}A^T$$

では最後に条件付き分布 $p(x|y)$ の平均と分散を求める。これは上で議論したように分割された精度行列によって条件付き分布は簡潔に表現できたことに注意すれば以下のものが簡単にもとまる。(分散は共分散行列の逆行列の (1,1) 成分の逆行列、ここでは R の (1,1) 成分の逆行列を指す)

$$cov(x|y) = (\Lambda + A^TLA)^{-1}$$

(7) を用いると

$$\begin{aligned} E(x|y) &= (\Lambda + A^TLA)^{-1}\{(\Lambda + A^TLA)\mu - (-A^TL)(y - A\mu - b)\} \\ &= (\Lambda + A^TLA)^{-1}\{\Lambda\mu + A^TLA\mu + A^TL(y - b) - A^TLA\mu\} \\ &= (\Lambda + A^TLA)^{-1}\{A^TL(y - b) + \Lambda\mu\} \end{aligned}$$

6.3 指数分布とラプラス分布

6.3.1 指数分布

$\lambda \in \mathrm{R}, x > 0$ とする。
指数分布とは、

$$p(x;\lambda) = \lambda \exp -\lambda x$$

という分布である。

6.3.2 ラプラス分布

ラプラス分布とは、$\mu, \gamma \in \mathrm{R}$ について

$$Laplace(x; \mu, \gamma) = \frac{1}{2\gamma} \exp - \frac{|x - \mu|}{\gamma}$$

という分布である。

6.4 ディラックのデルタ関数

6.4.1 デルタ分布

$x, \mu \in \mathrm{R}$ について、以下のようなものをディラックのデルタ分布という

$$p(x) = \delta(x - \mu)$$

ただし、$\int_{\mathrm{R}} \delta(x - \mu) f(x) dx = f(\mu)$

6.4.2 経験分布

$$p(x) = \frac{1}{m} \sum_{i=1}^{m} \delta(\boldsymbol{x} - \boldsymbol{x^{(i)}})$$

このような分布を経験分布という。

6.5 混合分布

確率分布 p がある時、k 個の確率分布が隠れていることがある。つまり、i 番目のクラスの確率分布が $p(x|C = i)$ で与えられているとする。この分布に重み $p(C = i)$ をつけると、

$$p(x) = \sum_{i=1}^{k} p(C = i) p(x|C = i)$$

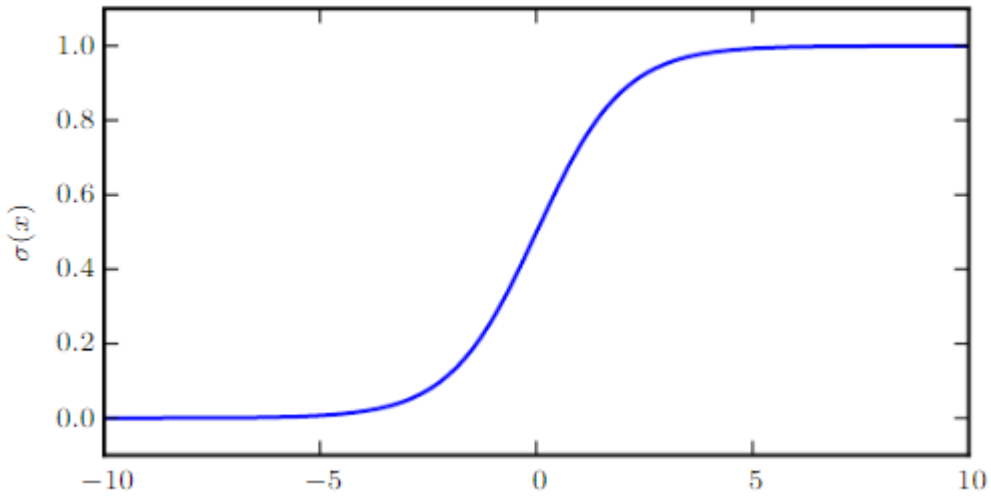
と表される。これを混合分布という。特に、正規分布の混合モデルなどが重要である。

6.6 sigmoid 関数

以下のような関数を logistic sigmoid 関数という。

$$\sigma(x) = \frac{1}{1 + \exp(-x)}$$

以下のようなグラフになる。



この関数は機械学習においてとても重要である。ロジスティックシグモイド関数には、対称性、値域が [0,1]、ほかにも微分にきれいな性質が存在する。

$$\zeta(x) = \log(1 + \exp(x))$$

を softplus function という

$\sigma(x)$ の微分を計算しておく。

$$\begin{aligned}
\frac{d}{da}\frac{1}{1+\exp(a)} &= \frac{\exp(-a)}{(1+\exp(-a))^2} \\
&= \frac{1}{1+\exp(-a)}\frac{\exp(-a)}{1+\exp(-a)} \\
&= \frac{1}{1+\exp(-a)}\{\frac{1+\exp(-a)}{1+\exp(-a)} - \frac{1}{1+\exp(-a)}\} \\
&= \sigma(a)(1-\sigma(a))
\end{aligned}$$

さらに、

$$\begin{aligned} 1-\sigma(x) &= \frac{1+\exp(-x)-1}{1+\exp(-x)} \\ &= \frac{\exp(-x)}{1+\exp(-x)} \\ &= \frac{1}{\exp(x)+1} = \sigma(-x) \end{aligned}$$

よって、

$$1-\sigma(x) = \sigma(-x)$$

という関係がある。さらに、

$$\begin{aligned} \log\sigma(x) &= \log(1) - \log(1+\exp(-x)) \\ &= -\zeta(-x) \end{aligned}$$

さらに、softplus function の微分は

$$\frac{d}{dx}\zeta(x) = \frac{\exp(x)}{1+\exp(x)}$$

$\sigma(x)$ は全単射であるので、その逆関数は

$$x = \frac{\exp(y)}{\exp(y)+1}$$

となる。y について、整理する。

$$\begin{aligned} x &= \frac{\exp(y)}{1+\exp(y)} \\ x(1+\exp(y)) &= \exp(y) \\ x + x\exp(y) &= \exp(y) \\ (1-x)\exp(y) &= x \\ \exp(y) &= \frac{x}{1-x} \\ y &= \log(\frac{x}{1-x}) \end{aligned}$$

この $\sigma(x)^{-1}$ は logit 関数と呼ばれる。

次に、$\zeta(x) = \log(1 + \exp(x))$ の逆関数は $x = \log(1 + \exp(y))$ である。y について整理すると、

$$\begin{aligned} x &= \log(1 + \exp(y)) \\ \exp(x) &= 1 + \exp(y) \\ \exp(y) &= 1 - \exp(x) \\ y &= \log(1 - \exp(x)) \end{aligned}$$

となる。

最後に以下の関係式を示しておく。

$$\begin{aligned} \zeta(x) - \zeta(-x) &= \log(1 + \exp(x)) - \log(1 + \exp(-x)) \\ &= \log \frac{1 + \exp(x)}{1 + \exp(-x)} \\ &= \log \frac{\exp(x) + \exp(x)^2}{\exp(x) + 1} \\ &= \log \exp(x) \frac{1 + \exp(x)}{\exp(x) + 1} \\ &= x \end{aligned}$$

よって、

$$\zeta(x) - \zeta(-x) = x$$

7 Information Theory

7.1 自己エントロピー

ある事象、「例えば明日大学の講義に X 分遅刻する」という事象を考える。

この事象に対する確率が P(X) が与えられているとします。P(1) は一分遅刻する確率である。この時確率分布 P(X) が持つ情報量はどれだけのものかとうことを考えたい。

明日の講義はテストを受けるとしよう。そのテストを受けないと単位を落としてしまうとする。しかし、テスト前日はすごく寝不足と仮定しよう。遅刻する確率が 99 パーセントとわかった時、ほとんどどうあがいても遅刻するのであれば単位を落とすのはほぼ確実である。

よって前日に徹夜で勉強するよりも、睡眠不足を解消するために寝る方がよっぽど効率的であることがわかる。しかし、遅刻をする確率が 50 パーセントとわかった時、前日にテスト勉強をすればよいのか、せずに睡眠をとればよいのか不明である。このように、確率が偏っているほど何が起こるか予測しやすく、対策を立てやすい。遅刻する確

率が 99 パーセントとわかる時は遅刻する確率が 50 パーセントとわかった時に比べて圧倒的に多いはずである。

確率 P(X) に対してこの情報量のことを P(X) の自己エントロピーという。
そして、自己エントロピーの期待値のことを平均エントロピーという。

ではこの情報量を数式で表うことを考える。。まず自己エントロピーには大事な性質が二つある。それが

- 互いに独立な確率変数の自己エントロピーはそれぞれの情報量の和で表される。
- 自己エントロピーは減少関数である。

の二つである。

7.1.1 自己エントロピーの加法性

互いに独立な確率変数の情報慮はそれぞれの情報量の和でなければならない。例えば「明日の講義が Y 分早く終わる」という事象を考える。この確率変数 Y はあなたが何分講義に遅刻しようが講義が何分早く終わるなんてことには関がない。よって確率変数 X と Y は独立である。ではもし「明日お前は 30 分遅刻し、講義は 30 分早く終わる」と未来の自分に教えてもらったとしよう。この時「明日自分は 30 分遅刻する」という事実と「明日の講義は 30 分早く終わる」という事実の二つの事実を知ったとして、それぞれの自己エントロピーの足し算と考えるのが自然である。

つまり、互いに独立な確率変数の同時に起こる確率変数の同時におこる確率はそれぞれの確率の積であるので P(X)P(Y)、情報量の関数を H() とするとき

$$H(P(X)P(Y)) = H(P(X)) + H(P(Y))$$

を満たしてほしいということになる。このような関数は対数関数である。

7.1.2 自己エントロピーは減少関数

自己エントロピーは上記の「直観的な話」の中でその事象が起こる確率が高ければ自己エントロピーは低く、確率が低ければ自己エントロピーは高いということを示した。
自己エントロピーにはどのような関数を定義すればよいか考える。

一つ目の性質を満たす関数といえば、対数関数だった。しかし、対数関数は単調増加関数である。そこで対数関数を-1 倍することで減少関数をし、二つ目の性質を満たすようにする。よって自己エントロピーの関数は確率 P(X) に対して

$$-log(P(X))$$

と定義される。

7.2 平均エントロピー

次に平均エントロピーを定義する。平均エントロピーは自己エントロピーの期待値であったので

$$H(P) = \int -P(X)log(P(X))dx$$

また、確率変数が離散（連続値でない）であれば

$$H(P) = \sum -P(X)log(P(X))$$

となる。平均エントロピーについての直観的理解を目指す。平均エントロピーとは自己エントロピーの期待値だった。

結論からゆうと平均エントロピーとはその分布の不確実性の大きさを示す。次の二つの確率分布について考え、そのことを確かめる。

- デルタ分布
- 一様分布

7.2.1 平均エントロピーの最小化

まずはデルタ分布の平均エントロピーである。
デルタ分布とはある確率変数で必ず確率 1 となり、それ以外の確率変数の確率は 0 となる確率分布のことである。

では [0,1] の値の中で 100 パーセントの確率で 0.5 の値をとるデルタ分布のエントロピーを求める。

この時の平均エントロピーは区間は $[0.5-\delta, 0.5+\delta]$ とし、この時 $\delta \to 0$ とすることで $\frac{1}{2\delta}log\frac{1}{2\delta}$ を積分し、求める。

つまり、[0.1] は連続な区間であり、デルタ分布（今回の）は 0.5 で 1 をとるので、積分するには底辺が区間 $0.5-\delta, 0.5+\delta]$ 高さ 1 の長方形を考え、上記のようにして長方形の横幅を狭めていくことで積分をします。計算は次のようになります。

$$lim_{\delta\to 0} - \int_{0.5-\delta}^{0/5+\delta} \frac{1}{2\delta} log\frac{1}{2\delta} = lim_{\delta\to 0}[-\frac{1}{2\delta}log\frac{1}{2\delta}]_{0.5-delta}^{0.5+\delta} = lim_{\delta\to 0} - \frac{1}{2\delta}log\frac{1}{2\delta}\times 2\delta = lim_{\delta\to 0} - log\frac{1}{2\delta} = -\infty$$

よってデルタ分布ではエントロピーは $-\infty$ となった。エントロピーを不確実性と考えるとデルタ分布では 100 パーセント何が起こるかわかっているという直観と一致する。

上記でデルタ分布はエントロピーを最小化させる分布であることがわかった。次にエントロピーを最大化するような分布は一様分布であることを示す。

一様分布とは区間 [a,b] に対してどの確率変数も $P(X) = \frac{1}{b-a}$ となる確率分布である。

7.2.2 平均エントロピーの最大化

次に平均エントロピーを最大化する。ここで 1,2,3,„,n のどれかをとる確率を $P = [p_1, p_2, , , , p_n]$ と表すことができ、この離散確率分布のエントロピーを最大化させることを考える。これにはラグランジュの未定乗数法という知識が必要になる。

$\sum_{i=1}^{n} p_i = 1$ の条件の下で $H = -\sum_{x \in X} p_i log p_i$ を最大化させる。
ラグランジュの未定乗数法を用いて $L(p, \lambda) = H + \lambda(\sum p_i - 1) \frac{\partial L}{\partial p_i} = 0$
より
$-log p_i - 1 + \lambda = 0 log p_i = \lambda - 1$
より
$p_i = exp(lambda - 1)$ 、これを条件式に代入すると
$nexp(\lambda - 1) = \lambda - 1 = log\frac{1}{n} = -logn$
これを先ほどの式に代入して
$p_i = exp(-logn) = \frac{1}{n}$ というふうに求まる。

これはつまり、$P = [p_1, p_2, , , , p_n]$ という確率分布を示し、これは、まさに一様分布を示す。つまり、エントロピーが最大になるのは一様分布であることがわかった。一様分布はどの確率変数に対しても $\frac{1}{b-a}$ 、今回でいえば $\frac{1}{n}$ をとるので何が一番起こりやすいのかわからないことを踏まえると平均エントロピーを最大化させる分布は一様分布であったので、平均エントロピーとは分布の不確実性の高さを表すということがわかる。

7.3 KL-ダイバージェンス

二つの確率分布の平均エントロピーの差を表す値を KL ダイバージェンスという。
式では次のように定義される。

$$KL(P||Q) = \int_{-\infty}^{\infty} P(X) log \frac{P(X)}{Q(X)}$$

離散の場合は

$$KL(P||Q) = \sum_i P(X_i) log \frac{P(X_i)}{Q(X)}$$

これは二つの分布の距離を表すとしてよく使われている。ではなぜ二つの分布間の距離をこのように定義できるのだろうか。

真の分布 P(X) が存在するとする。しかし、有限のデータから真の分布 P(X) を求めるのは不可能である。そこで、有限のデータから推定して得られた確率分布を Q(X）とする。真の分布 P(X）と推定した分布 Q(X) はどれだけ違うのだろうか。

ここで登場するのがエントロピーである。エントロピーはその分布の不確実性を示す値であった。

エントロピーが高いほど不確かなことが起こるとゆうことである。
P(X) のエントロピーとは $-\int_{-\infty}^{\infty} logP(X)$ である。

では推定した確率分布 Q(X）は確率分布 P(X) に対してどれだけ不確実性を持っているのだろうか。エントロピーとは情報量の期待値であった。確率分布 Q(X）が持つ情報量は $-logQ(X)$ である。この情報量を確率 P(X) で期待値をとる。

式は以下のようになる。

$$-\int_{-\infty}^{\infty} P(X) logQ(X)$$

この値と真の分布のエントロピーとの差を二つの分布間の差として定義する。式では以下のようになる。

$$-\int_{-\infty}^{\infty} P(X) logQ(X) - (-\int_{-\infty}^{\infty} P(X) logP(X)))$$

これを式変形すると

$$-\int_{-\infty}^{\infty} P(X)(logQ(X) - logP(X)) = \int_{-\infty}^{\infty} (logP(X) - logQ(X)) = \int_{-\infty}^{\infty} P(X) log \frac{P(X)}{Q(X)}$$

となる。これが KL-ダイバージェンスである。

7.4 クロスエントロピー

確率分布 $P(X), Q(X)$ のクロスエントロピーを

$$H(P,Q) = H(P) + KL(P||Q)$$

と定義する。
KL-ダイバージェンスを最小化する Q を選ぶことは、クロスエントロピーを最小化することに等しい。これは、$H(P)$ の項が Q に依存しないためである。

8 グラフィカモデル

機械学習では、多くの確率変数を扱う。各確率変数の関係をグラフと呼ばれるもので表したい。グラフとは、グラフ理論、もしくは離散数学と呼ばれる分野で使われるものである。

8.1 グラフ

node と呼ばれる集合 V と edge と呼ばれる集合 E に対して、写像 $\mathcal{G} : \mathrm{E} \to \mathrm{V} \times \mathrm{V}$ を定める。
この時、

$$(\mathrm{E}, \mathrm{V}, \mathcal{G})$$

で定められる空間をグラフという。

写像でノード間の関係を定めることにどういった意味があるのか考察する。
$\forall (v_1, v_2) \in \mathrm{V} \times \mathrm{V}$ の v_1 を始点、v_2 を終点ということにする。
写像の定義を思い出すと、すべての E の元に対して、たった一つの $\mathrm{V} \times \mathrm{V}$ の元を対応させる。つまり、各 edge に対して始点、終点がたった一つ存在すること意味する。よって、edge に始点がない、もしくは終点がないということは許されないことが写像の定義より表現できる。

8.1.1 有向グラフ

$\mathrm{V} \times \mathrm{V}$ の元 (v_1, v_2) の v_1 を始点、v_2 を終点ということにする。この時、$(\mathrm{E}, \mathrm{V}, \mathcal{G})$ を有向グラフという。

8.1.2 無効グラフ

$\mathrm{V} \times \mathrm{V}$ の元 (v_1, v_2) と (v_2, v_1) を同一とみた $(\mathrm{E}, \mathrm{V}, \mathcal{G})$ を無向グラフという。

無向グラフのクリークとは、頂点集合 $\mathrm{C} \subset \mathrm{V}$ のうち、C に属するあらゆる二つの頂点を繋ぐ辺が存在する場合に C をクリークという。

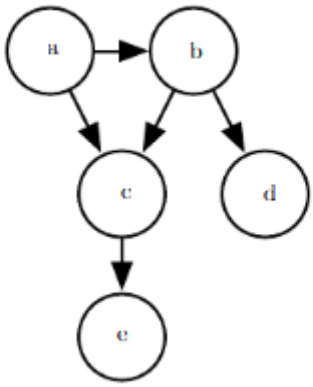
8.2 ベイジアンネットワーク

例えば、確率分布が $p(a,b,c,d,e) = p(a)P(b|a)p(c|a,b)p(d|b)p(e|c)$ のように表されているとする。

この時、確率変数 $\mathrm{V} = \{a,b,c,d,e\}$ として、$\mathrm{E} = \{(a,b),(a,c),(b,c),(b,d),(c,e)\}$ として、写像 $\mathcal{G} : \mathrm{E} \to \mathrm{V} \times \mathrm{V}$ を以下のように定める。

$$\mathcal{G}((i,j)) = (i,j)$$

これを可視化すると以下のようになる。



このように、条件付き確率を node と edge で表すことができる。このようなものをグラフィカモデルという。
例えば、ノード c には a,b という確率変数から edge が伸びている。これは、c の確率分布が a,b のパラメータのみで表される、ということを意味している。

これはグラフィカモデルの中でもこのような有向グラフをベイジアンネットワークという。

8.3 マルコフネットワーク

無向グラフ $(\mathrm{E}, \mathrm{V}, \mathcal{G})$、確率変数の集合 $\{X_v\}_{v\in\mathrm{V}}$ に対して、以下のマルコフ性を満たすとき、この無効グラフはマルコフネットワークと呼ばれる。

- ペアワイズマルコフ性：隣接しない任意の二変数が、ほかの全ての変数を与えられた時に条件付き独立になる。
- 局所マルコフ性：ある変数に直接つながっている変数が条件付けられた時、その変数が他の全ての変数と条件付き独立になる。
- 大域マルコフ性：確率変数の集合の任意の二つの部分集合が、その二つを分割するような部分集合を与えられた時、条件付き独立になる。

$\boldsymbol{X} = (X_v)_{v\in\mathrm{V}}$ が与えらえたとき、$\boldsymbol{X}$ の確率分布は $(X_v)_{v\in\mathrm{V}}$ の同時分布としてあらわされる。
この同時確率が次のように、グラフ $\mathcal{G}$ のクリークに分解可能であるとき、

$$P(\boldsymbol{X}) = \Pi_{\mathrm{C}\in cl(\mathcal{G})}(\phi_C(x_C))$$

クリーク分解可能であるといい、大域マルコフ性が成立し、$\boldsymbol{X}$ はグラフ $\mathcal{G}$ に対してマルコフネットワークを形成する。

9 Numerical Computation

機械学習では、ほとんど必ず numerical Computation(数値計算) を扱う必要がある。問題に合う目的関数と呼ばれるものを定義し、それを最小化、もしくは最大化する際に必ず必要になるからである。この章では数値計算の基本的なアルゴリズムの解説を行う。

初めに、コンピューターに関する問題点を扱う。具体的には、コンピューターでは ∞ や $-\infty$ を扱うことはできない。これにより、様々な弊害が起こりうる。この subsection ではこの問題に関して言及する。

後半では、具体的な最適化手法について解説する。

9.1 Overflow and Underflow

コンピューターで何かを計算するとき、大きすぎる値を扱おうとすると、Overflow という現象が起こる。同様に小さすぎる値を扱う、もしくは 0 に非常に近い値で割り算を行おうとすると、と underflow という現象が起こる。このような現象を解消することが必

要になる場合がある。

例えば、以下のような関数を考える。

$$softmax(\mathrm{x}_i) = \frac{\exp(x_i)}{\sum_{j=1}^{n} \exp(x_j}$$

この関数は softmax 関数 (ソフトマックス関数) と呼ばれている。機械学習では、ニクラスより大きい、クラス分類を行うときに、必要になる関数である。

例えば、x(ベクトル) のすべての要素が c という定数の時を考える。
c が非常に小さいとき、softmax 関数の分母 $\sum_{j=1}^{n} \exp(x_j$ は非常に小さい値 (0 に非常に近い値) になる。このままでは 0(に近い値) で割り算をすることになり、コンピューターはエラーを出す。

これを解決する方法がある。$c^{'} \in \mathrm{R}$ として、$\mathrm{z} = \mathrm{x} + c^{'}$ と置く。(x のすべての要素に $c^{'}$ を足す。)
すると、softmax 関数は

$$\begin{aligned}
softmax(\mathrm{z}_i) &= \frac{\exp(z_i)}{\sum_{j=1}^{n} \exp(z_j)} \\
&= \frac{\exp(x_i + c^{'})}{\sum_{j=1}^{n} \exp(x_j + c^{'})} \\
&= \frac{\exp(x_i)\exp(c^{'})}{\sum_{j=1}^{n} \exp(x_j)\exp(c^{'})} \\
&= \frac{\exp(x_i)\exp(c^{'})}{\exp(c^{'})\sum_{j=1}^{n} \exp(x_j)} \\
&= \frac{\exp(x_i)}{\sum_{j=1}^{n} \exp(x_j)} \\
&= softmax(\mathrm{x}_i)
\end{aligned}$$

このようにして、任意の定数を加えることで解決することができる。

9.2 Wilkinson の後退誤差解析

機械学習で扱うようなデータには誤差が含まれることが多い。では、誤差が小さければ、必ずしも安心できるのか、と言われればそうではない。これの代表例として、連立一次方程式 $\boldsymbol{Ax} = \boldsymbol{b}$ という問題を考える。
この時、各構成要素 $\boldsymbol{A}, \boldsymbol{x}, \boldsymbol{b}$ のそれぞれの誤差が、ほかの要素にどれだけ影響を与える

かを調べる。

まずは簡単な例から始め、少しの誤差が推定結果に大きな影響を与えることをイメージする。

9.2.1 敏感な方程式

$$\boldsymbol{A} = \begin{bmatrix} 1 & 100 \\ 0 & 1 \end{bmatrix}, \boldsymbol{A}^{-1} = \begin{bmatrix} 1 & -100 \\ 0 & 1 \end{bmatrix}$$

という行列を考える。この時、

$$\boldsymbol{b} = \begin{bmatrix} 100 \\ 1 \end{bmatrix}, \boldsymbol{b}' = \begin{bmatrix} 100 \\ 0 \end{bmatrix}$$

という二つのベクトルを用意して、$\boldsymbol{Ax} = \boldsymbol{b}, \boldsymbol{Ax}' = \boldsymbol{b}'$ という二つの方程式を解くと、

$$\boldsymbol{b} = \begin{bmatrix} 100 \\ 1 \end{bmatrix} \implies \boldsymbol{x} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$\boldsymbol{b}' = \begin{bmatrix} 100 \\ 0 \end{bmatrix} \implies \boldsymbol{x}' = \begin{bmatrix} 100 \\ 0 \end{bmatrix}$$

この例からもわかる通り、少しの差が解に大きな影響を与えている。倍率は 100^2 である。

もう一つ例を確認してみる。

$$\boldsymbol{A} = \begin{bmatrix} 0.780 & 0.563 \\ 0.913 & 0.659 \end{bmatrix}, \boldsymbol{b} = \begin{bmatrix} 0.217 \\ 0.254 \end{bmatrix}$$

とする。この時、$\boldsymbol{Ax} = \boldsymbol{b}$ の解は

$$\boldsymbol{x} = \begin{bmatrix} 1.00 \\ -1.00 \end{bmatrix}$$

である。ここで、

$$\Delta\boldsymbol{b} = \begin{bmatrix} 0 \\ 10^{-6} \end{bmatrix}$$

としたとき、$\boldsymbol{A}(\boldsymbol{x} + \boldsymbol{\Delta x}) = \boldsymbol{b} + \boldsymbol{\Delta b}$ の解 $\boldsymbol{x} + \Delta\boldsymbol{x}$ は

$$\boldsymbol{x} + \Delta\boldsymbol{x} = \begin{bmatrix} 0.440 \\ -0.220 \end{bmatrix}$$

となる。このように元の解から大きく外れた値が得られる。

ではこのような、少しの誤差によって、解が大きく変換するかを判断するような定理を確認していく。

具体的には condition number(条件数) と呼ばれるものを定義し、Banach の摂動定理から、行列の摂動、右辺の摂動に対する解の誤差を評価する。

9.2.2 行列のノルム

まずは、行列のノルムについて定義する。

定義 (行列のノルム).
$\|\cdot\| : \boldsymbol{M}(n,n) \ni \boldsymbol{A} \to \|\boldsymbol{A}\| \in \mathrm{R}$ が次の条件を満たすとき、これを $\boldsymbol{M}(n,n)$ のノルムという。
$\forall \boldsymbol{A}, \boldsymbol{B} \in \boldsymbol{M}(n,n),\ \forall c \in \mathrm{R}$ に対して、

- $\|\boldsymbol{A}\| \geq 0$, かつ $\boldsymbol{A} = 0 \iff \|\boldsymbol{A}\| = 0$
- $\|c\boldsymbol{A}\| = |c|\|\boldsymbol{A}\|$
- $\|\boldsymbol{A} + \boldsymbol{B}\| \leq \|\boldsymbol{A}\| + \|\boldsymbol{B}\|$
- $\|\boldsymbol{A}\boldsymbol{B}\| \leq \|\boldsymbol{A}\|\|\boldsymbol{B}\|$

一般的にノルムは最後の条件は要求しないが、行列には非可換な積が定義されているため、この条件を要求することにする。

一つの例として、行列の p 乗ノルムを定義する。

定義 (行列の p 乗ノルム).
$\forall \boldsymbol{A} \in \boldsymbol{M}(n,n)$ に対して、ノルム $\|\cdot\|_p$ を

$$\|\boldsymbol{A}\|_p = \sup_{\|\boldsymbol{x}\|_p \leq 1} \|\boldsymbol{A}\boldsymbol{x}\|_p$$

と定義する。

特に、p=2 の場合、これはスペクトル・ノルムと呼ばれる。
また、行列の p 上ノルムは以下のように書き換えられる。

定理 **9.1** (行列の p 乗ノルム).
$\boldsymbol{R}^n$ の任意のノルム $\|\cdot\|$, 任意の $\boldsymbol{A} \in \boldsymbol{M}(n,n)$ に対して、

$$\sup_{\|\boldsymbol{x}\| \leq 1} \|\boldsymbol{A}\boldsymbol{x}\| = \sup_{\|\boldsymbol{x}\| = 1} \|\boldsymbol{A}\boldsymbol{x}\| = \sup_{\|\boldsymbol{x}\| \neq 0} \frac{\|\boldsymbol{A}\boldsymbol{x}\|}{\|\boldsymbol{x}\|}$$

よって、

$$\|\boldsymbol{A}\|_p = \sup_{\boldsymbol{x} \neq 0} \frac{\|\boldsymbol{A}\boldsymbol{x}\|_p}{\|\boldsymbol{x}\|_p}$$

Proof.
$\|\boldsymbol{y}\| < 1$ となる $\boldsymbol{y} \in \boldsymbol{R}^n$ をとる。この時、

$$\boldsymbol{Ay} = \begin{bmatrix} row(A)_1 y \\ \vdots \\ row(A)_n y \end{bmatrix}$$

よって、明らかに $\|\boldsymbol{x}\| = 1$ のような $\boldsymbol{x} \in \mathrm{R}^n$ を選んだときのほうがノルムは大きい。
以上より題意が成り立つ。 □

無限次元ベクトル空間の作用素ノルムはこちらの定義を用いる。

定義 (ベクトルのノルムと両立する行列のノルム)**.**
R^n のノルム $\|\cdot\|$ と $\boldsymbol{M}(n,n)$ の行列ノルム $\|\cdot\|'$ が両立するとは任意の $\boldsymbol{A} \in \boldsymbol{M}(n,n)$,$\boldsymbol{x} \in \mathrm{R}^n$ に対して

$$\|\boldsymbol{Ax}\| \leq \|\boldsymbol{A}\|'\|\boldsymbol{x}\|$$

が成立することをいう。

定義より、行列の p 乗ノルムはベクトルの p 乗ノルムと両立する。

定義 (スペクトル半径)**.**
行列 $\boldsymbol{A}$ の固有値の最大値を $\boldsymbol{A}$ のスペクトル半径とよび、$\rho(\boldsymbol{A})$ で表す。
また、行列 $\boldsymbol{A}$ の固有値の集合を $\sigma(\boldsymbol{A})$ とする。

ベクトルのノルムと両立する行列ノルムはスペクトル半径以上である。つまり、行列のノルムを固有値の最大値で評価できる。

定理 **9.2** (行列ノルムの評価)**.**
R^n のノルム $\|\cdot\|$ と両立する $\boldsymbol{M}(n,n)$ のノルム $\|\cdot\|'$ について、任意の $\boldsymbol{A} \in \boldsymbol{M}(n,n)$

$$\rho(\boldsymbol{A}) \leq \|\boldsymbol{A}\|'$$

Proof. λ を $\boldsymbol{A}$ の任意の固有値とするとき、対応する固有ベクトル $\boldsymbol{x} \neq 0$ をとると、

$$|\lambda|\|\boldsymbol{x}\| = \|\lambda\boldsymbol{x}\| = \|\boldsymbol{Ax}\| \leq \|\boldsymbol{A}\|'\|\boldsymbol{x}\|$$

よって、

$$|\lambda| \leq \|\boldsymbol{A}\|'$$

以上より、

$$\rho(\boldsymbol{A}) \leq \|\boldsymbol{A}\|'$$

□

次の定理はこの後の照明で用いるものである。

定理 **9.3** (スペクトルノルムと固有値)**.**

$$\forall \boldsymbol{A} \in \boldsymbol{M}(n,n), \quad \|\boldsymbol{A}\|_2 = \sqrt{\rho(AA^T)}$$

つまりこれは $\boldsymbol{A}$ の最大特異値である。
特に A がエルミート行列（対称行列）の時は

$$\|\boldsymbol{A}\|_2 = \rho(\boldsymbol{A})$$

Proof.

$$\|\boldsymbol{A}\|_2 = \sup_{\boldsymbol{x}\neq 0} \frac{\|\boldsymbol{A}\boldsymbol{x}\|_2}{\|\boldsymbol{x}\|_2} = \sup_{\boldsymbol{x}\neq 0} \sqrt{\frac{(\boldsymbol{A}\boldsymbol{x}, \boldsymbol{A}\boldsymbol{x})}{(\boldsymbol{x},\boldsymbol{x})}} = \sup_{\boldsymbol{x}\neq 0} \sqrt{\frac{(\boldsymbol{A^T}\boldsymbol{A}\boldsymbol{x}, \boldsymbol{x})}{(\boldsymbol{x},\boldsymbol{x})}}$$

よって、Rayleigh の原理より、

$$\sup_{\boldsymbol{x}\neq 0} \sqrt{\frac{(\boldsymbol{A^T}\boldsymbol{A}\boldsymbol{x}, \boldsymbol{x})}{(\boldsymbol{x},\boldsymbol{x})}} = \sqrt{\rho(AA^T)}$$

□

ただし、Rayleigh の原理とは以下の定理である。まずは、Tayleigh 商を定義する。

定義 (Rayleigh 商)**.**
$\boldsymbol{A}$ をエルミート行列とする。$\boldsymbol{x} \neq 0$ とするとき、

$$R_{\boldsymbol{A}} = \frac{(\boldsymbol{A}\boldsymbol{x}, \boldsymbol{x})}{(\boldsymbol{x},\boldsymbol{x})}$$

を $\boldsymbol{A}$ の Rayleigh 商という。

定理 **9.4** (Rayleigh の原理)**.**

$$\min_{\boldsymbol{x}\neq 0} R_{\boldsymbol{A}}(\boldsymbol{x}) = \boldsymbol{A}\text{の最小固有値}$$

当然 max に変えると、$\boldsymbol{A}$ の最大固有値になる。

これは、PCA の章で確認した。

次に条件数を定義する。

9.2.3 Condition number(条件数)

条件数とは、一般的にはコンピュータでの数値計算がどれだけやりやすいかを表す尺度であり、条件数が小さい問題は「良条件」、大きい問題は「悪条件」という。ここでは、$\boldsymbol{Ax} = \boldsymbol{b}$ という連立一次方程式を解く際の条件数を扱う。

定義 (Condition number(条件数))**.**
任意の正則な $\boldsymbol{A} \in \boldsymbol{M}(n, n)$ と、任意の行列ノルム $\|\cdot\|$ に対して、条件数を

$$\|\boldsymbol{A}\|\|\boldsymbol{A}^{-1}\|$$

と定義する。
記号では、

$$cond(\boldsymbol{A}) = \|\boldsymbol{A}\|\|\boldsymbol{A}^{-1}\|$$

と書くことにし、特に p 乗ノルムを用いるときは $cond_p(\boldsymbol{A})$ と書くことにする。

条件数は行列のノルムに依存する定義であるので本来であれば、$cond_{\|\cdot\|_{(\boldsymbol{A}}}$ のような表記を用いるのが適切かもしれないが、ここでは p 乗ノルムを使う時以外は $cond(\boldsymbol{A})$ という書き方をすることにする。

条件数についての基本的な性質を示しておく。

定理 **9.5** (条件数の基本的な性質)**.**

- $cond(\boldsymbol{A}) \geq 1$
- $cond(\boldsymbol{A}) = cond(\boldsymbol{A}^{-1})$
- $\forall \alpha \in \mathrm{R}\ \{0\}, \quad cond(\alpha \boldsymbol{A}) = cond(\boldsymbol{A})$

Proof.
行列ノルムの定義からすべて明らか。 □

次に、後の計算に用いる簡単な性質を示す。

定理 **9.6.**
$\boldsymbol{A}, \boldsymbol{B}$ を n 次正方行列とする。この時

$$cond(\boldsymbol{AB}) \leq cond(\boldsymbol{A})cond(\boldsymbol{B})$$

条件数の定義から簡単に求められるが、念のため証明しておく。

Proof.

$$cond(\boldsymbol{AB}) = \|\boldsymbol{AB}\|\|(\boldsymbol{AB})^{-1}\| = \|\boldsymbol{AB}\|\|\boldsymbol{B}^{-1}\boldsymbol{A}^{-1}\| \leq \|\boldsymbol{A}\|\|\boldsymbol{B}\|\|(\boldsymbol{B})^{-1}\|\|(\boldsymbol{A})^{-1}\| = cond(\boldsymbol{A})cond(\boldsymbol{B})$$

□

定理 **9.7** ($cond_p(\boldsymbol{A})$ に関する性質)**.**
$\boldsymbol{a} =\in \mathrm{R}^n$ とする。この時、

- 任意の正則行列 $\boldsymbol{D}diag(\boldsymbol{a})$ $cond_p(\boldsymbol{D}) = \max \frac{|a_i|}{|a_i|} \quad (1 \leq p \leq \infty)$
- ユニタリー行列 $\boldsymbol{U}$ について $cond_2(\boldsymbol{U}) = 1$
- $\boldsymbol{T}$ を n 次ユニタリー行列とするとき、任意の n 次正則行列 $\boldsymbol{A}$ に対して、$cond_2(\boldsymbol{A}) = cond_2(\boldsymbol{AT}) = cond_2(\boldsymbol{TA}) = cond_2(\boldsymbol{T^TAT})$

Proof.
一つ目の証明
$1 \leq p < \infty$ の場合を示す。$p = \infty$ の時も同様である。行列の p 乗ノルムはベクトルの p 乗ノルムと両立するので、行列ノルムの評価より行列のノルムはスペクトル半径以上である。よって、

$$\max_{1\leq j\leq n} |a_j| = \rho(\boldsymbol{D}) \leq \|\boldsymbol{D}\|_p$$

これを用いると、

$$\|\boldsymbol{Dx}\| = (\sum_{j=1}^{n} |a_j x_j|^p)^{1/p} \leq (\sum_{j=1}^{n} (\max_{1\leq j\leq n} |a_j||x_j|)^p)^{1/p} = \max_{1\leq j\leq n} |a_j| \|\boldsymbol{x}\|_p$$

以上より、

$$\|\boldsymbol{D}\|_p \leq \max_{1\leq j\leq n} |a_j|$$

よって、

$$\|\boldsymbol{D}\| = \max_{1\leq j\leq n} |a_j|$$

$\boldsymbol{D}^{-1}$ についても同様に成立するので、

$$\|\boldsymbol{D}^{-1}\|_p = \max_{1\leq j\leq n} \frac{1}{|a_j|} = \frac{1}{\min_{1\leq j\leq n} |a_j|}$$

よって、

$$cond_p(\boldsymbol{D}) = \|\boldsymbol{D}\|_p \|\boldsymbol{D}^{-1}\|_p = \frac{\max_{1\leq j\leq n} |a_j|}{\min_{1\leq j\leq n} |a_j|}$$

□

Proof. 二つ目の証明
$\boldsymbol{U}$ をユニタリー行列とする。$\boldsymbol{U}^{-1}$ もユニタリー行列である。

$$\|\boldsymbol{U}\|_2 = \|\boldsymbol{U}^{-1}\|_2 = 1$$

よって、

$$cond_2(\boldsymbol{U}) = 1$$

□

Proof. 三つ目は略 □

次の定理は応用上重要である。

定理 **9.8** (スペクトル・ノルムに関する条件数の性質)**.**
n 次正則行列 $\boldsymbol{A}$ に対して、σ_{max} を $\boldsymbol{A}$ の特異値の最大値、σ_{min} を $\boldsymbol{A}$ の特異値の最小値とする。この時、

$$cond_2(\boldsymbol{A}) = \frac{\sigma_{max}}{\sigma_{min}}$$

特に、$\boldsymbol{A}$ がエルミート行列の時、λ_{max} を $\boldsymbol{A}$ の固有値の最大値、λ_{min} を $\boldsymbol{A}$ の固有値の最小値とすると、

$$cond_2(\boldsymbol{A}) = \frac{\lambda_{max}}{\lambda_{min}}$$

Proof.

$$\|\boldsymbol{A}\|_2 = \sigma_{max}, \|\boldsymbol{A}^{-1}\|_2 = \sigma_{min}$$

より、

$$cond_2(\boldsymbol{A}) = \frac{\sigma_{max}}{\sigma_{min}}$$

□

条件数を任意の両立する行列ノルムとベクトルのノルムに関して、評価する定理がある。

定理 **9.9** (最大固有値と最小固有値の比と条件数の関係)**.**
行列のノルムが少なくとも一つのベクトルのノルムと両立しているならば、その時条件数は $cond$ は $\lambda_{max}, \lambda_{min}$ を $\boldsymbol{A}$ の最大固有値、最小固有値とすると、

$$cond(\boldsymbol{A}) \geq \frac{|\lambda_{max}|}{|\lambda_{min}|}$$

Proof.
行列のノルムはスペクトル半径以上であったので、

$$\|\boldsymbol{A}\| \geq \rho(\boldsymbol{A})$$

同様に、

$$\|\boldsymbol{A}^{-1}\| \geq \rho(\boldsymbol{A}^{-1}) = \frac{1}{\rho(\boldsymbol{A})}$$

よって、

$$cond(\boldsymbol{A}) \geq \frac{\max_{\lambda\in\sigma(\boldsymbol{A})}|\lambda|}{\min_{\lambda\in\sigma(\boldsymbol{A})}|\lambda|}$$

□

これにより、固有値の絶対値の比が大きければ条件数も大きい。後に条件数が大きいとき、解の敏感性は高くなる。Banach の摂動定理の証明に必要な Neumann 級数 (ノイマン級数) についての定理を確認する。
ただし、関数解析の知識を仮定している。

9.2.4 バナッハの摂動定理

定理 **9.10** (Neumann 級数)**.**
$\boldsymbol{G} \in \boldsymbol{M}(n,n)$ が $\|\boldsymbol{G}\| < 1$ を満たすならば、

$$\boldsymbol{C} = \sum_{n=0}^{\infty} \boldsymbol{G}^n$$

はノルム収束し、

$$(\boldsymbol{I}-\boldsymbol{G})\boldsymbol{C} = \boldsymbol{C}(\boldsymbol{I}-\boldsymbol{G}) = \boldsymbol{I}$$

が成立する。
よって、$\boldsymbol{I}-\boldsymbol{G}$ は正則、$(\boldsymbol{I}-\boldsymbol{G})^{-1} = \boldsymbol{C}$ であり、

$$\|(\boldsymbol{I}-\boldsymbol{G})^{-1}\| \leq \frac{1}{1-\|\boldsymbol{G}\|}$$

Proof.

$$\|\boldsymbol{G}^n\| \leq \|\boldsymbol{G}\|^n \quad for\ n \in \mathrm{N}$$

である。$\sum_{n=0}^{\infty}\|\boldsymbol{G}^n\|$ は収束等比級数 $\sum_{n=0}^{\infty}\|\boldsymbol{G}\|^n$ を優級数に持つので収束する。よって、$\sum_{n=0}^{\infty}\boldsymbol{G}^n$ はノルム収束する。これは、$\sum_{n=0}^{\infty}\boldsymbol{G}^n$ の部分和がバナッハ空間上のコーシー列であるので、成立する。

$$\boldsymbol{C}(\boldsymbol{I}-\boldsymbol{G}) = \boldsymbol{C}-\boldsymbol{C}\boldsymbol{G} = \sum_{n=0}^{\infty}\boldsymbol{G}^n - (\sum_{n=0}^{\infty}\boldsymbol{G}^n)\boldsymbol{G} = \sum_{n=0}^{\infty}\boldsymbol{G}^n - \sum_{n=0}^{\infty}(\boldsymbol{G}^n\cdot\boldsymbol{G}) = \sum_{n=0}^{\infty}\boldsymbol{G}^n - \sum_{n=1}^{\infty}\boldsymbol{G}^n = \boldsymbol{I}$$

反対も同様に成立する。よって、

$$\|(\boldsymbol{I}-\boldsymbol{G})^{-1}\| = \|\boldsymbol{C}\| \leq \sum_{n=0}^{\infty} \|\boldsymbol{G}\|^n = \frac{1}{1-\|\boldsymbol{G}\|}$$

□

この Neumann 級数の定理より、次の Banach の摂動定理を証明する。

定理 **9.11** (Banach の摂動定理)**.**
正則行列 $\boldsymbol{A}$ と $\Delta\boldsymbol{A} \in \boldsymbol{M}(n,n)$, $\|\boldsymbol{A}^{-1}\Delta\boldsymbol{A}\| < 1$ とすれば、$\boldsymbol{A}+\Delta\boldsymbol{A}$ は正則で

$$\|(\boldsymbol{A}+\Delta\boldsymbol{A})^{-1}\| \leq \frac{\|\boldsymbol{A}^{-1}\|}{1-\|\boldsymbol{A}^{-1}\Delta\boldsymbol{A}\|}$$

Proof. $\boldsymbol{G} = \boldsymbol{A}^{-1}\Delta\boldsymbol{A}$ と置くと、仮定より $\|\boldsymbol{G}\| < 1$ であるので、Neumann 級数の定理から $\boldsymbol{I}+\boldsymbol{G}$ が正則で、

$$\|(\boldsymbol{I}+\boldsymbol{G})^{-1}\| \leq \frac{1}{1-\|\boldsymbol{G}\|}$$

となる。$\boldsymbol{A}+\Delta\boldsymbol{A} = \boldsymbol{A}(\boldsymbol{I}+\boldsymbol{A}^{-1}\Delta\boldsymbol{A}) = \boldsymbol{A}(\boldsymbol{I}+\boldsymbol{G})$ であるので、$\boldsymbol{A}+\Delta\boldsymbol{A}$ が正則で、

$$\|(\boldsymbol{A}+\Delta\boldsymbol{A})^{-1}\| \leq \|(\boldsymbol{I}+\boldsymbol{G})^{-1}\|\|\boldsymbol{A}^{-1}\| \leq \frac{\|\boldsymbol{A}^{-1}\|}{1-\|\boldsymbol{G}\|} = \frac{\|\boldsymbol{A}^{-1}\|}{1-\|\boldsymbol{A}^{-1}\Delta\boldsymbol{A}\|}$$

□

次に示す定理が、この章の目標のものである。

定理 **9.12** (右辺と係数行列の摂動)**.**
正則行列 $\boldsymbol{A}$,$\Delta\boldsymbol{A} \in \boldsymbol{M}(n,n)$,$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$,$(\boldsymbol{A}+\Delta\boldsymbol{A})(\boldsymbol{x}+\Delta\boldsymbol{x}) = \boldsymbol{b}+\Delta\boldsymbol{b}$,$\|\boldsymbol{A}^{-1}\Delta\boldsymbol{A}\| \leq 1$ とすれば、

$$\frac{\|\Delta\boldsymbol{x}\|}{\|\boldsymbol{x}\|} \leq \frac{cond(\boldsymbol{A})}{1-\|\boldsymbol{A}^{-1}\Delta\boldsymbol{A}\|}\left(\frac{\|\Delta\boldsymbol{A}\|}{\|\boldsymbol{A}\|} + \frac{\|\Delta\boldsymbol{b}\|}{\|\boldsymbol{b}\|}\right)$$

Proof. Banach の摂動定理より、$\boldsymbol{A}+\Delta\boldsymbol{A}$ が正則で、

$$\|(\boldsymbol{A}+\Delta\boldsymbol{A})^{-1}\| \leq \frac{\|\boldsymbol{A}^{-1}\|}{1-\|\boldsymbol{A}^{-1}\Delta\boldsymbol{A}\|}$$

$(\boldsymbol{A}+\Delta\boldsymbol{A})(\boldsymbol{x}+\Delta\boldsymbol{x}) = \boldsymbol{b}+\Delta\boldsymbol{b}$ から両辺 $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ を引くと

$$\Delta\boldsymbol{A}\boldsymbol{x} + (\boldsymbol{A}+\Delta\boldsymbol{A})\Delta\boldsymbol{x} = \Delta\boldsymbol{b}$$

$\Delta \boldsymbol{x}$ について解くと、

$$\Delta \boldsymbol{x} = (\boldsymbol{A} + \Delta \boldsymbol{A})^{-1}(\Delta \boldsymbol{b} - \Delta \boldsymbol{A}\boldsymbol{x})$$

両辺のノルムをとると、

$$\|\Delta \boldsymbol{x}\| \leq \|(\boldsymbol{A} + \Delta \boldsymbol{A})^{-1}\|\|\Delta \boldsymbol{b} - \Delta \boldsymbol{A}\boldsymbol{x}\| \leq \frac{\|\boldsymbol{A}^{-1}\|}{1 - \|\boldsymbol{A}^{-1}\Delta \boldsymbol{A}\|}(\|\Delta \boldsymbol{b}\| + \|\Delta \boldsymbol{A}\|\|\boldsymbol{x}\|)$$

$\|\boldsymbol{x}\|$ で割ると、

$$\frac{\|\Delta \boldsymbol{x}\|}{\|\boldsymbol{x}\|} \leq \frac{\|\boldsymbol{A}^{-1}\|}{1 - \|\boldsymbol{A}^{-1}\Delta \boldsymbol{A}\|}(\|\Delta \boldsymbol{A}\| + \frac{\|\Delta \boldsymbol{b}\|}{\|\boldsymbol{x}\|}) = \frac{\|\boldsymbol{A}\|\|\boldsymbol{A}^{-1}\|}{1 - \|\boldsymbol{A}^{-1}\Delta \boldsymbol{A}\|}(\frac{\|\Delta \boldsymbol{A}\|}{\|\boldsymbol{A}\|} + \frac{\|\Delta \boldsymbol{b}\|}{\|\boldsymbol{A}\|\|\boldsymbol{x}\|})$$

$\|\boldsymbol{b}\| \leq \|\boldsymbol{A}\|\|\boldsymbol{x}\|$ より、

$$\frac{\|\Delta \boldsymbol{x}\|}{\|\boldsymbol{x}\|} \leq \frac{\|\boldsymbol{A}\|\|\boldsymbol{A}^{-1}\|}{1 - \|\boldsymbol{A}^{-1}\Delta \boldsymbol{A}\|}(\frac{\|\Delta \boldsymbol{A}\|}{1 - \|\boldsymbol{A}^{-1}\Delta \boldsymbol{A}\|} + \frac{\|\Delta \boldsymbol{b}\|}{\|\boldsymbol{b}\|})$$

□

以上により、条件数が大きければそれだけ解の誤差は大きくなる可能性がある。

9.3 Optimization

定義 (最適化問題).

与えられた関数 $f : A \to \mathbb{R}$ と A の元 x についての条件 P(x) について、
$A_0 = \{x \in A | P(x)$が真$\} \subset A$ としたとき、

$$x_0 \in A_0 : \quad \forall x \in A_0,\ f(x_0) \leq f(x)$$

を満たす x_0 を求めよ

この問題を最適化問題という。以下は基本的な用語である。

- f: 目的関数
- P: 制約条件
- A_0: 実行可能領域
- $x \in A$: 実行可能解
- x_0: 最適解

9.3.1 再急降下法

関数 $f(\boldsymbol{x})$ の極小値を求めることを考える。ただし、制約条件はないものとする。
最も簡単な方法は再急降下法と呼ばれる。

$f : \mathbb{R}^n \to \mathbb{R}$ の勾配 ∇f を考える。
ベクトル $x \in \mathbb{R}^n$ に対して、最も f が変化する方向を求めたい。
$\nabla f(\boldsymbol{x})$ は位置 $\boldsymbol{x}$ においての、最大傾斜方向を示すベクトルになる。これを証明する。
$\boldsymbol{x} = [x_1, ..., x_n]$ が $d\boldsymbol{x} = [dx_1, ..., dx_n]$ だけずれたとする。この時、$f(\boldsymbol{x} + d\boldsymbol{x})$ の一次までのテイラー展開より、

$$f(\boldsymbol{x} + d\boldsymbol{x}) \approx f(\boldsymbol{x}) + \frac{\partial f}{\partial x_1} dx_1 + ... + \frac{\partial f}{\partial x_n} dx_n$$

よって、変化量 df は

$$\frac{\partial f}{\partial x_1} dx_1 + ... + \frac{\partial f}{\partial x_n} dx_n$$

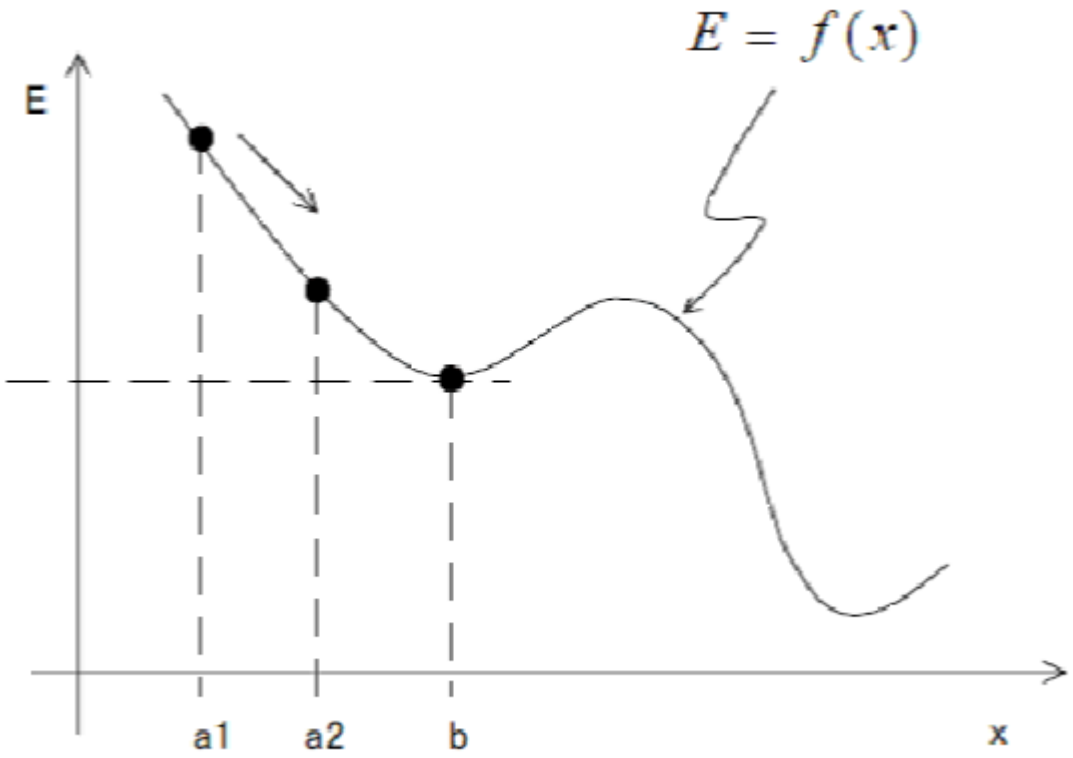
である。
よって、df は $[\frac{\partial f}{\partial x_1}, ..., \frac{\partial f}{\partial x_n}]$ と $[dx_1, ..., dx_n]$ の内積である。
f が変化しないとき、$df = 0$ なので、この内積は 0 になる。よって、この二つのベクトルが直交していることがわかる。$df = 0$ とは、$\boldsymbol{x}$ の移動先が同じ等高線上にあることを意味し、その方向に垂直な方向が最大傾斜方向であることがわかる。よって、勾配とは最大傾斜方向を表すベクトルであることがわかる。

適当な初期値から始め、その地点の最大傾斜方向にある定数をかけて、更新していくアルゴリズムを再急降下法と呼ぶ。

Algorithm 1 Gradient Decent

Require: ∇f,α(learning rate), x_0(init value)
  $\epsilon \leftarrow$ Convergence judgment value
  for k=0,...,n **do**
    $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \alpha \nabla f(\boldsymbol{x}_k)$
    if $|\boldsymbol{x}_{k+1} - \boldsymbol{x}_k| < \epsilon$ **then**
      break
    else
      continue
    end if
  end for

learning rate は適当な値を決める必要がある。これが小さすぎると収束が遅いし、大きすぎると、最小値を過ぎてしまって収束しない。



適切な learning rate を求める手法も存在する。最も簡単な方法は直線探索法である。

9.3.2 直線探索法

再急降下法の直線探索の場合、第 k ステップにおいて、$f(\boldsymbol{x}_k + \alpha d_k)$ を最小化させるような $\alpha \geq 0$ を探す。
一般的に、直線探索を用いた再急降下法は次のようなアルゴリズムになる。

Algorithm 2 Gradient Decent with linear search

Require: ∇f,α(learning rate), x_0(init value)
  $\epsilon \leftarrow$ Convergence judgment value
  for k=0,...,n **do**
    solve $\alpha_k \in \min_{\alpha \in \mathbb{R}_+} f(\boldsymbol{x}_k + \alpha \nabla f(\boldsymbol{x}_k))$
    $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \alpha_k \nabla f(\boldsymbol{x}_k)$
    if $|\boldsymbol{x}_{k+1} - \boldsymbol{x}_k| < \epsilon$ **then**
      break
    else
      continue
    end if
  end for

9.3.3 Hessian matrix

再急降下法では、局所最適解(極値)を求めることをした。これが大域最適解かどうかは、再急降下法だけでは判断できない。この時に、Hessian matrix と呼ばれる行列が重要になる。

定義 (Hessian matrix)**.** $f : \mathbb{R}^n \to \mathbb{R}$ にすべての二階偏微分が存在するとき、

$$\boldsymbol{H}(f)_i j = \frac{\partial^2 f}{\partial x_i \partial j}$$

を Hessian matrix という。

もちろん各偏微分はいつでも交換可能ではない。しかし、応用上多くの場合交換可能である。
二階偏微分関数が存在し、連続であれば、交換可能である。しかし、もう少し弱い十分条件が存在する。

定理 **9.13** (シュワルツの定理)**.** 二変数関数 $f(x, y)$ について、
(a, b) の近傍で f_x, f_y, f_{xy} が存在し、f_{xy} が (a, b) で連続 $\implies$ $f_{yx}(a, b)$ が存在し、(a, b) の近傍で $f_{xy} = f_{yx}$

Proof. $\phi(x,y) = f(x,y) - f(x,b)$ と置く。十分小さい h,k をとり、$\phi(x, b+k)$ に平均値の定理を用いて

$$\begin{aligned}\phi(a+h, b+k) - \phi(a, b+k) &= h\phi_x(a+\theta h, b+k) \\ &= h\{f_x(a+\theta h, b+k) - f_x(a+\theta, b)\}\end{aligned}$$

ただし、$0 < \theta < 1$ である。
さらに、$f_x(a+\theta h, y)$ に平均値の定理を用いて、

$$\phi(a+h, b+k) - \phi(a, b+k) = hkf_{xy}(a+\theta h, b+\theta^{'}k), \quad 0 < \theta^{'} < 1$$

k で割って、両辺 $\lim_{k\to 0}$ をとると、

$$f_y(a+h, b) - f_y(a, b) = h \lim_{k\to 0} f_{xy}(a+\theta h, b+\theta^{'}k)$$

よって、

$$\begin{aligned}f_{yx} &= \lim_{h\to 0}\{f_y(a+h, b) - f_y(a, b)\}/h \\ &= \lim_{h\to 0}\lim_{k\to 0} f_{xy}(a+\theta h, b+\theta^{'}k) \\ &= \lim_{(h,k)\to(0,0)} f_{xy}(a+\theta h, b+\theta^{'}k) \\ &= f_{xy}(a, b)\end{aligned}$$

□

点 $\boldsymbol{p}$ 上の Hessian matrix の固有値は、

10 Machine Learning Basis

References

[1] http://nalab.mind.meiji.ac.jp/ mk/labo/text/linear-eq-3.pdf