

計算機実験 I (講義 4)

藤堂眞治

wistaria@phys.s.u-tokyo.ac.jp

2023/07/05

- 1 密行列の対角化
- 2 疎行列に対する反復法
- 3 特異値分解
- 4 最小二乗法による回帰分析

出席: 本日の 13 時までに ITC-LMS でアンケートに回答

物理の問題にあらわれる行列演算

- 連立一次方程式・逆行列
 - ▶ 偏微分方程式の境界値問題
 - ▶ 非線形連立方程式に対するニュートン法
- 対角化・特異値分解
 - ▶ 固有値問題・行列関数
 - ▶ 最小二乗近似
- 計算機は大規模行列演算が得意
 - ▶ 直接法: $\sim 10^4$ 次元
 - ▶ 疎行列に対する反復解法: $\sim 10^9$ 次元
- 行列演算についてはライブラリがよく整備されている
 - ▶ それぞれの原理とその特徴を理解して正しく使うことが重要
 - ▶ 適切なライブラリを使うことで数十倍あるいはそれ以上速くなることも

時間依存しないシュレディンガー方程式

■ 井戸型ポテンシャル中の一粒子問題

$$\left[-\frac{\hbar^2}{2m} \frac{d^2}{dx^2} + V(x) \right] \psi(x) = E\psi(x)$$

$$V(x) = \begin{cases} 0 & a \leq x \leq b \\ \infty & \text{otherwise} \end{cases}$$

■ $\hbar^2/2m = 1$ 、 $a = 0$ 、 $b = 1$ となるように変数変換して

$$\left(\frac{d^2}{dx^2} + E \right) \psi(x) = 0 \quad 0 \leq x \leq 1$$

を境界条件 $\psi(0) = \psi(1) = 0$ のもとで解けば良い

シュレディンガー方程式の行列表示

■ シュレディンガー方程式

$$\left[-\frac{d^2}{dx^2} + V(x)\right]\psi(x) = E\psi(x)$$

■ 連立差分方程式を行列の形で表す ($\psi(x_0) = \psi(x_n) = 0$)

$$\begin{bmatrix} \frac{2}{h^2} + V(x_1) & -\frac{1}{h^2} & & & \\ -\frac{1}{h^2} & \frac{2}{h^2} + V(x_2) & -\frac{1}{h^2} & & \\ & -\frac{1}{h^2} & \frac{2}{h^2} + V(x_3) & -\frac{1}{h^2} & \\ & & \ddots & \ddots & \\ & & & -\frac{1}{h^2} & \frac{2}{h^2} + V(x_{n-1}) \end{bmatrix} \begin{bmatrix} \psi(x_1) \\ \psi(x_2) \\ \psi(x_3) \\ \vdots \\ \psi(x_{n-1}) \end{bmatrix} = E \dots$$

■ $(n-1) \times (n-1)$ の疎行列の固有値問題

- ▶ 固有値: 固有エネルギー
- ▶ 固有ベクトル: 波動関数

実対称行列 (エルミート行列) の性質

- $n \times n$ 実対称行列 $A (= A^T)$ の固有値問題

$$Ax = \lambda x$$

- n 個の固有値 $(\lambda_1, \lambda_2, \dots, \lambda_n)$ は全て実。固有ベクトル $(\xi_1, \xi_2, \dots, \xi_n)$ は互いに正規直交するようにとることができる。行列 U を

$$U = [\xi_1 \ \xi_2 \ \cdots \ \xi_n]$$

と定義すると、 U は直交 (ユニタリ) 行列 ($U^T U = U^{-1} U = E$)

- A の固有分解 (固有値分解)

$$A = U \Lambda U^T \quad \Lambda = \text{diag}(\lambda_1, \dots, \lambda_n)$$

行列のべき乗・指数関数

■ 行列のべき乗

$$\begin{aligned} A^p &= (U\Lambda U^T)(U\Lambda U^T) \cdots (U\Lambda U^T) \\ &= U\Lambda^p U^T \quad \Lambda^p = \text{diag}(\lambda_1^p, \dots, \lambda_n^p) \end{aligned}$$

■ 行列の指数関数

$$\begin{aligned} e^{xA} &= \sum_{k=0}^{\infty} \frac{1}{k!} (xA)^k = U \left[\sum_{k=0}^{\infty} \frac{1}{k!} (x\Lambda)^k \right] U^T \\ &= U e^{x\Lambda} U^T \quad e^{x\Lambda} = \text{diag}(e^{x\lambda_1}, \dots, e^{x\lambda_n}) \end{aligned}$$

- [逆行列 $A^{-1} = U\Lambda^{-1}U^T$] → 逆行列をあらわに求めるかわりに連立方程式を解く
- [行列式 $|A| = \prod_i \lambda_i$] → 対角化ではなく LU 分解を使う
- [対角和 (トレース) $\text{tr}A = \sum_i \lambda_i$] → 対角化不要

行列の数値対角化

- 一般的に次元が 5 以上の行列の固有値は、あらかじめ定まる有限回の手続きでは求まらない
 - ▶ 必ず何らかの反復法 (+収束判定) が必要となる
- 密行列向きの方法
 - ▶ Jacobi 法
 - ▶ Givens 変換・Householder 法 (三重対角化) + QR 法など
- 疎行列向きの方法
 - ▶ べき乗法
 - ▶ Lanczos 法 (三重対角化) + QR 法など
- 固有ベクトル
 - ▶ QR 法で求めたものを逆変換
 - ▶ 逆反復法で精度改善

基本方針

- やってはいけない方法: 特性方程式

$$|\lambda E - A| = 0$$

の係数を求めて、代数方程式として解く

- ▶ 数値的に不安定 (代数方程式の解は係数の誤差に対して敏感)
- ▶ 計算コスト大 [$\sim O(N!)$]
- スタンダードな方法: 行列を次々に直交変換して、対角行列 (あるいは三重対角行列) に近づけていく

$$A \rightarrow U_1^T A U_1 \rightarrow U_2^T (U_1^T A U_1) U_2 \rightarrow U_3^T (U_2^T (U_1^T A U_1) U_2) U_3 \rightarrow \cdots$$

- 固有値は変換された行列の固有値、固有ベクトルは変換後の行列の固有ベクトルに左から $U_1 U_2 U_3 \cdots$ を掛けたもの

Jacobi 法

- 直交行列 U_{pq} を以下のように選ぶ ($(p, p), (p, q), (q, p), (q, q)$ 成分を除くと単位行列)

$$U_{pq} = \begin{bmatrix} 1 & & & & & & & \\ & \ddots & & & & & & \\ & & 1 & & & & & \\ & & & \cos \theta & & \sin \theta & & \\ & & & & 1 & & & \\ & & & & & 1 & & \\ & & & -\sin \theta & & \cos \theta & & \\ & & & & & & 1 & \\ & & & & & & & \ddots \\ & & & & & & & & 1 \end{bmatrix}$$

Jacobi 法による相似変換

- $B = U_{pq}^{-1} A U_{pq}$ により、 A の p 行、 q 行、 p 列、 q 列のみが変更を受ける

$$b_{pk} = b_{kp} = a_{pk} \cos \theta - a_{qk} \sin \theta \quad k \neq p, q$$

$$b_{qk} = b_{kq} = a_{pk} \sin \theta + a_{qk} \cos \theta \quad k \neq p, q$$

$$b_{pp} = \frac{a_{pp} + a_{qq}}{2} + \frac{a_{pp} - a_{qq}}{2} \cos 2\theta - a_{pq} \sin 2\theta$$

$$b_{qq} = \frac{a_{pp} + a_{qq}}{2} - \frac{a_{pp} - a_{qq}}{2} \cos 2\theta + a_{pq} \sin 2\theta$$

$$b_{pq} = b_{qp} = \frac{a_{pp} - a_{qq}}{2} \sin 2\theta + a_{pq} \cos 2\theta$$

- $b_{pq} = b_{qp} = 0$ とするには、 θ を次のように選べば良い

$$\tan 2\theta = -\frac{2a_{pq}}{a_{pp} - a_{qq}}$$

Jacobi 法の収束

- 相似変換により対角和は不変に保たれるので

$$\mathrm{tr} A^T A = \mathrm{tr} B^T B \Rightarrow \sum_{i,j} a_{ij}^2 = \sum_{i,j} b_{ij}^2$$

- 一方、この変換で

$$b_{pp}^2 + b_{qq}^2 = b_{pp}^2 + 2b_{pq}^2 + b_{qq}^2 = a_{pp}^2 + 2a_{pq}^2 + a_{qq}^2$$

すなわち、変換により、対角成分の二乗和は増加する \Rightarrow 非対角成分の二乗和は単調減少

- 全ての非対角成分が十分小さくなるまで繰り返す
- 固有値=対角成分、固有ベクトル $= U_1 U_2 U_3 \cdots$

3重対角化

- 対角化は有限回の手続きでは行えない
- 3重対角化であれば、 $O(n^3)$ の有限回の計算で決定論的に行える
- Givens 変換: Jacobi 変換と同じ相似変換を利用
 - ▶ U_{32} で $(3,1)$ と $(1,3)$ を消去 $\Rightarrow U_{42}$ で $(4,1)$ と $(1,4)$ を消去 $\Rightarrow U_{52}$ で $(5,1)$ と $(1,5)$ を消去 $\Rightarrow U_{62}, \dots, U_{n,2} \Rightarrow U_{43}, U_{53}, \dots, U_{n,3} \Rightarrow \dots \Rightarrow U_{n,n-1}$ で $(n, n-2)$ と $(n-2, n)$ を消去
 - ▶ $(4/3)n^3$ 回の乗算と $(2/3)n^3$ 回の加減算で 3 重対角化される
- Householder 変換: $U = E - 2ww^T/|w|^2$
 - ▶ $(2/3)n^3$ 回の乗算と加減算で 3 重対角化される
 - ▶ Givens 変換に比べ少し効率的なので、こちらが広く使われている

QR 法

■ QR 分解

- ▶ 行列 A を直交 (ユニタリー) 行列 Q と上三角行列 R の積に分解:

$$A = QR$$

- ▶ Gram-Schmidt の直交化と等価

■ QR 法による固有値と固有ベクトルの計算

- ▶ 行列 A_1 を QR 分解 ($A_1 = Q_1 R_1$) $\rightarrow A_2 = R_1 Q_1$
- ▶ 行列 A_2 を QR 分解 ($A_2 = Q_2 R_2$) $\rightarrow A_3 = R_2 Q_2$
- ▶ 行列 A_k を QR 分解 ($A_k = Q_k R_k$) $\rightarrow A_{k+1} = R_k Q_k$
- ▶ 繰り返していくと対角より下の全ての成分は零に収束し、対角成分は固有値に収束する (証明略)

■ 連続した直交変換: $A_{k+1} = R_k Q_k = Q_k^{-1} Q_k R_k Q_k = Q_k^{-1} A_k Q_k$

- ▶ A_1 が対称 (エルミート) 三重対角行列の場合、 A_k も対称 (エルミート) 三重対角
- ▶ 密行列に対して最初から QR 法を適用するより、Householder 法で三重対角化した後で使う方が効率がよい

LAPACK の対角化ルーチン

- 様々な対角化ルーチンが準備されている

- ▶ 倍精度実対称行列の対角化 dsyev http://www.netlib.org/lapack/explore-html/dd/d4c/dsyev_8f.html
- ▶ Fortran による関数宣言

```
subroutine dsyev(character JOBZ, character UPLO,  
integer N, double precision, dimension(lda, *) A,  
integer LDA, double precision, dimension(*) W,  
double precision, dimension(*) WORK,  
integer LWORK, integer INFO)
```

- 他にも dsyevd、dsyevr、dsyevx などがある
3 重対角化までは同じ。3 重対角行列の対角化が異なる
- 単精度版の ssyev、複素 (エルミート行列) 版の zheev など
- dsyev の使用例: [diag.c](#)
 - ▶ コンパイル方法: `cc -o diag diag.c -llapack -lblas -lm`
 - ▶ 実行方法: `./diag matrix1.dat`

反復法

- 疎行列の場合、行列ベクトル積は高速に行える
- Givens 変換、Householder 変換などを行うと疎行列性が失われる
- 行列ベクトル積のみを用いる反復法が効果的
 - ▶ べき乗法
 - ▶ Lanczos 法

べき乗法 (Power Method)

- 適当なベクトル v_1 から出発する
- v_1 が最大固有ベクトル ξ_1 と直交していないとすると

$$v_1 = c_1\xi_1 + c_2\xi_2 + c_3\xi_3 + \cdots + c_n\xi_n$$

と展開できる ($c_1 \neq 0$)。この両辺に A を次々と掛けていくと

$$v_2 = Av_1 = c_1\lambda_1\xi_1 + c_2\lambda_2\xi_2 + c_3\lambda_3\xi_3 + \cdots + c_n\lambda_n\xi_n$$

$$v_3 = A^2v_1 = c_1\lambda_1^2\xi_1 + c_2\lambda_2^2\xi_2 + c_3\lambda_3^2\xi_3 + \cdots + c_n\lambda_n^2\xi_n$$

$$\vdots$$

$$v_{k+1} = A^k v_1 = c_1\lambda_1^k\xi_1 + c_2\lambda_2^k\xi_2 + c_3\lambda_3^k\xi_3 + \cdots + c_n\lambda_n^k\xi_n$$

$$= c_1\lambda_1^k \left[\xi_1 + \sum_{i=2}^n \frac{c_i}{c_1} \left(\frac{\lambda_i}{\lambda_1} \right)^k \xi_i \right] \approx c_1\lambda_1^k \xi_1$$

べき乗法の収束

■ べき乗法による固有値

$$\frac{v_{k+1}^T v_{k+1}}{v_{k+1}^T v_k} = \lambda_1 + O\left(\left(\frac{\lambda_2}{\lambda_1}\right)^{2k}\right)$$

■ 誤差の収束

$$\frac{v_{k+1}^T v_{k+1}}{v_{k+1}^T v_k} \approx \lambda_1 + e^{-2k \ln(\lambda_1/\lambda_2)}$$

- $1/\ln(\lambda_1/\lambda_2)$ 程度の反復が必要
- λ_1 と λ_2 が近い場合には、反復回数が非常に多くなる

第2固有値・第3固有値...

- 第1固有ベクトル ξ_1 の成分を行列から差し引く (減次)

$$A_1 = A - \lambda_1 \xi_1 \xi_1^T$$

この行列は、固有値 $0, \lambda_2, \lambda_3, \dots, \lambda_n$ を持つ

- 行列 A_1 に対してべき乗法を使うと、第2固有値 λ_2 と対応する固有ベクトル ξ_2 が得られる
- 第 k 固有値まで求まっている場合

$$A_k = A - \sum_{i=1}^k \lambda_i \xi_i \xi_i^T$$

- 実際には数値誤差のため、ベクトルの直交性は厳密ではない
- 大きい方から数個程度を求めるのが限界

Rayleigh-Ritz の方法

- $n \times n$ 行列 A について、互いに正規直交するベクトル v_1, v_2, \dots, v_m ($m < n$) が張る部分空間の中で「最良の」固有ベクトルを求めたい
- $n \times m$ 行列

$$V = [v_1 v_2 \cdots v_m]$$

を定義すると、 $V^T V = E_m$ が成り立つ (ただし $V V^T \neq E_n$)

- 部分空間内のベクトルを $w = \sum_i a_i v_i$ と表すと、 $\frac{w^T A w}{w^T w}$ が極大値を取る (本当の固有ベクトルにできるだけ平行になる) 条件は、

$$\frac{\partial}{\partial a_i} \frac{w^T A w}{w^T w} \sim \sum_j H_{ij} a_j - \lambda a_i = 0$$

Rayleigh-Ritz の方法

- $m \times m$ 行列

$$H = V^T A V$$

に対する固有値問題 $Ha = \lambda a$

- λ : もとの行列の近似固有値 (Ritz 値)
- Va : もとの行列の近似固有ベクトル (Ritz ベクトル)
- 最大固有値に対する近似固有値が欲しい場合、最大固有ベクトルになるべく近い (しかし互いに直交する) ベクトル v_1, v_2, \dots, v_m を選べばよい

Lanczos 法

- 初期 (ランダム) ベクトル v_1 に A を掛けて生成される

$$v_1, Av_1, A^2v_1, \dots, A^{m-1}v_1$$

を正規直交化して v_1, v_2, \dots, v_m を作る (Krylov 部分空間)

$$\mathcal{K}_m(A, v_1) = \text{span}\{v_1, Av_1, A^2v_1, \dots, A^{m-1}v_1\}$$

- 部分空間での Ritz 値を固有値の近似値とする
- $A^k v_1$ はどんどん最大固有ベクトルに近づいていくので、 $m \ll n$ でも良い近似固有値が得られると期待される

Lanczos 法

- 正規化された初期 (ランダム) ベクトル v_1 から出発する
- v_2, v_3, \dots を生成する

$$v_2 = (Av_1 - \alpha_1 v_1) / \beta_1$$

$$v_3 = (Av_2 - \beta_1 v_1 - \alpha_2 v_2) / \beta_2$$

$$\vdots$$

$$v_{m+1} = (Av_m - \beta_{m-1} v_{m-1} - \alpha_m v_m) / \beta_m$$

ここで、 α_i と β_i は以下のように選ぶ

$$\alpha_i = v_i^T A v_i$$

$$\beta_i = \|Av_i - \beta_{i-1} v_{i-1} - \alpha_i v_i\|, \beta_0 = 0$$

Lanczos 法

- $v_1, v_2, v_3, \dots, v_{m+1}$ は正規直交
- 漸化式を書き換えると

$$Av_1 = \alpha_1 v_1 + \beta_1 v_2$$

$$Av_2 = \beta_1 v_1 + \alpha_2 v_2 + \beta_2 v_3$$

$$Av_3 = \beta_2 v_2 + \alpha_3 v_3 + \beta_3 v_4$$

$$\vdots$$

$$Av_m = \beta_{m-1} v_{m-1} + \alpha_m v_m + \beta_m v_{m+1}$$

Lanczos 法

■ 行列で表現すると

$$A[v_1 v_2 \cdots v_m] = [v_1 v_2 \cdots v_m v_{m+1}] \begin{bmatrix} \alpha_1 & \beta_1 & & & & & \\ \beta_1 & \alpha_2 & \beta_2 & & & & \\ & \beta_2 & \alpha_3 & \beta_3 & & & \\ & & \beta_3 & \alpha_4 & \beta_4 & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & \beta_{m-1} & \alpha_m & \\ & & & & & \beta_m & \end{bmatrix}$$

両辺に左から $[v_1 v_2 \cdots v_m]^T$ をかけると

$$[v_1 v_2 \cdots v_m]^T A [v_1 v_2 \cdots v_m]$$

は 3 重対角行列となることがわかる

Lanczos 法

- 原理的には、 n ステップ目で $\beta_n = 0$ となり、3 重対角化が完了する
- 実際には、数値誤差のため $v_1, v_2, v_3 \cdots$ の直交性が崩れていく
 - ▶ m を大きくしすぎると、おかしい固有値が出てくる
 - ▶ 多くの固有値・固有ベクトルが欲しい場合には Householder 法を使うべき
- Lanczos 法では、大きな固有値に対応する固有ベクトルにできるだけ近いものから部分空間を作っていく
 - ▶ 100 万次元以上の行列の場合でも $m = 100 \sim 200$ 程度で最初の数個の固有値は精度良く求まる
- 必要な操作は、行列とベクトルの積、ベクトルの内積・スケーリング・和のみ
 - ▶ 疎行列の場合、非常に効率が良い

一般の非正方行列の場合

- 特異値分解 (SVD: Singular Value Decomposition)
- 任意の $m \times n$ 実行列 A は

$$A = U\Lambda V^T$$

の形に (一意に) 分解できる ($k = \min(m, n)$)

- U : ($m \times k$) 行列 (列ベクトルは互いに正規直交)
 V : ($n \times k$) 行列 (列ベクトルは互いに正規直交)
 $\Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_k)$ ($\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_k \geq 0$) 特異値
- ベクトル表示 (行列をランク 1 の行列で分解)

$$A = \sum_{i=1}^k \lambda_i u_i v_i^T$$

特異値分解の例

$$\begin{pmatrix} 1 & 2 & 3 \\ 6 & 4 & 5 \\ 8 & 9 & 7 \\ 10 & 11 & 12 \end{pmatrix} = \begin{pmatrix} -0.14 & -0.62 & -0.05 \\ -0.34 & 0.37 & 0.81 \\ -0.55 & 0.54 & -0.58 \\ -0.75 & -0.44 & 0.06 \end{pmatrix} \\ \times \begin{pmatrix} 25.35 & 0 & 0 \\ 0 & 2.15 & 0 \\ 0 & 0 & 1.71 \end{pmatrix} \begin{pmatrix} -0.56 & -0.59 & -0.59 \\ 0.68 & 0.09 & -0.73 \\ 0.48 & -0.81 & 0.35 \end{pmatrix}$$

完全特異値分解 (full SVD)

$$\begin{pmatrix} 1 & 2 & 3 \\ 6 & 4 & 5 \\ 8 & 9 & 7 \\ 10 & 11 & 12 \end{pmatrix} = \begin{pmatrix} -0.14 & -0.62 & -0.05 & -0.77 \\ -0.34 & 0.37 & 0.81 & -0.29 \\ -0.55 & 0.54 & -0.58 & -0.29 \\ -0.75 & -0.44 & 0.06 & 0.48 \end{pmatrix} \\
 \times \begin{pmatrix} 25.35 & 0 & 0 \\ 0 & 2.15 & 0 \\ 0 & 0 & 1.71 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} -0.56 & -0.59 & -0.59 \\ 0.68 & 0.09 & -0.73 \\ 0.48 & -0.81 & 0.35 \end{pmatrix}$$

- U, V が直交行列になるように、足りない基底ベクトルを追加
- こちらを「SVD」、もともとの分解を「thin SVD」と呼ぶこともある

LAPACK による特異値分解

- 倍精度実行列の特異値分解 dgesvd http://www.netlib.org/lapack/explore-html/d8/d2d/dgesvd_8f.html
- Fortran による関数宣言

```
subroutine dgesvd(character JOBU, character JOBVT,  
    integer M, integer N,  
    double precision, dimension(lda, *) A,  
    integer LDA, double precision, dimension(*) S,  
    double precision, dimension(ldu, *) U, integer LDU,  
    double precision, dimension(ldvt, *) VT,  
    integer LDVT,  
    double precision, dimension(*) WORK, integer LWORK,  
    integer INFO)
```

- dgesvd の使用例: [svd.c](#) (SVD), [full_svd.c](#) (完全 SVD)
 - ▶ コンパイル方法: `cc -o svd svd.c -llapack -lblas -lm`
 - ▶ 実行方法: `./svd matrix2.dat`

固有値分解との関係

- 半正定値実対称行列の場合
「固有値分解」と「特異値分解」は等価
- 一般の (非正方) 実行列 A の場合
 - ▶ 完全 SVD $A = U\Lambda V^T$ を考えると
 - ▶ $B = A^T A = V\Lambda U^T U\Lambda V^T = V\Lambda^2 V^T$ は実対称行列
固有値: λ_i^2 、固有ベクトル V
 - ▶ $C = AA^T = U\Lambda V^T V\Lambda U^T = U\Lambda^2 U^T$ も実対称行列
固有値: λ_i^2 、固有ベクトル U

特異値分解が役に立つ例

- 連立一次方程式の最小二乗解
- 行列の低ランク近似
- 画像圧縮

特異な (特異に近い) 連立一次方程式

- ランク r の $m \times n$ 行列 A の full SVD を考える

$$A = (U_1 U_2) \begin{pmatrix} \Lambda_1 & 0 \\ 0 & 0 \end{pmatrix} (V_1 V_2)^T = U_1 \Lambda_1 V_1^T$$

U_1 : $m \times r$, U_2 : $m \times (m - r)$, Λ_1 : $r \times r$, V_1 : $n \times r$, V_2 : $n \times (n - r)$

- $r \neq n$ のとき、方程式 $Ax = b$ は無限個の解をもつ、あるいは解なし
- 無限個の解をもつ場合 ($U_1 U_1^T b = b$) の一般解

$$x = V_1 \Lambda_1^{-1} U_1^T b + V_2 z$$

z は $(n - r)$ 次元の任意のベクトル

一般化逆行列と最小二乗解

- V_1 と V_2 の列ベクトルは全て直交する。一般解のうちノルム $|x|^2$ が最小となるのは $z = 0$ のとき

$$x = A^\dagger b \quad A^\dagger = V \begin{pmatrix} \Lambda^{-1} & 0 \\ 0 & 0 \end{pmatrix} U^T$$

- A^\dagger : Moore-Penrose の一般化逆行列 (1/0 を 0 とおくことに相当)
- $Ax = b$ が解を持たない場合にも最良解を与える
- 零に非常に近い特異値がある場合も、それらを零とみなすことで数値的に安定した解がえられる

行列の低ランク近似

- ランク r ($r < k$) の行列のうち、行列 A を「最も良く近似」する \tilde{A} を選ぶ
- 「最も良く近似」 = フロベニウスノルム $\|A - \tilde{A}\|_F$ を最小化

$$\|X\|_F^2 \equiv \sum_{ij} x_{ij}^2$$

- 特異値のうち大きなものから r 個とり、残りを零とした

$$\tilde{\Lambda} = \text{diag}(\lambda_1, \dots, \lambda_r, 0, \dots, 0)$$

を使い

$$\tilde{A} = U\tilde{\Lambda}V^T$$

を作れば良い (Eckart-Young の定理)

行列の低ランク近似の例

$$\begin{pmatrix} -0.14 & -0.62 & -0.05 \\ -0.34 & 0.37 & 0.81 \\ -0.55 & 0.54 & -0.58 \\ -0.75 & -0.44 & 0.06 \end{pmatrix} \begin{pmatrix} 25.35 & 0 & 0 \\ 0 & 2.15 & 0 \\ 0 & 0 & 0 \end{pmatrix} \\
 \times \begin{pmatrix} -0.56 & -0.59 & -0.59 \\ 0.68 & 0.09 & -0.73 \\ 0.48 & -0.81 & 0.35 \end{pmatrix} = \begin{pmatrix} 1.04 & 1.93 & 3.03 \\ 5.33 & 5.12 & 4.52 \\ 8.47 & 8.21 & 7.34 \\ 9.95 & 11.1 & 12.0 \end{pmatrix}$$

- それなりに悪くない近似が得られる
- 誤差 (フロベニウスノルム) = 1.71 (無視した特異値の二乗和の平方根)

Eckart-Young の定理の証明

- A を近似する行列 X (ランク $\leq r$) を考えると

$$\|A - X\|_F^2 = \sum_{ij} (a_{ij} - x_{ij})^2 = \text{tr}[(A - X)(A - X)^T]$$

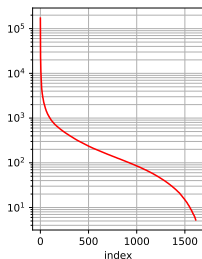
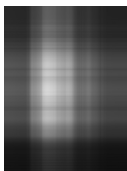
を最小化すればよい。完全 SVD について $UU^T = E_m$, $VV^T = E_n$ より

$$\begin{aligned} \|A - X\|_F^2 &= \text{tr}[UU^T(A - X)VV^T(A - X)^T] \\ &= \text{tr}[(\Lambda - G)(\Lambda - G)^T] \quad (G \equiv U^T X V) \\ &= \sum_i^k (\lambda_i - g_{ii})^2 + \sum_i \sum_{j \neq i} g_{ij}^2 \end{aligned}$$

ランク r 以下で、 $\|A - X\|_F^2$ を最小化するには、 $g_{ii} = \lambda_i$ ($i = 1 \cdots r$)、それ以外は全て零とすればよい

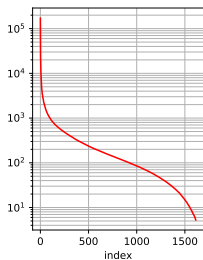
特異値分解による画像圧縮

■ 特異値の分布とランク r 近似 (1614 × 2178 グレースケール写真)

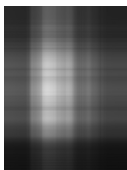
 $r = 1$  $r = 2$  $r = 3$  $r = 4$ 

特異値分解による画像圧縮

■ 特異値の分布とランク r 近似 (1614 × 2178 グレースケール写真)



$r = 1$



$r = 2$



$r = 3$



$r = 4$



$r = 5$



$r = 10$



$r = 20$



$r = 50$



最小二乗法によるフィッティング

- 説明変数 (例: 電圧): $x_1, x_2, x_3, \dots, x_n$
- 観測値 (例: 電流): $y_1, y_2, y_3, \dots, y_n$
- 単回帰モデル: $y = a + bx + \epsilon$ (ϵ : ノイズ)
- 未知母数: a, b
- 最小二乗法: 残差 $R(a, b) = \sum_i^n (y_i - (a + bx_i))^2$ を最小化

$$\frac{\partial R}{\partial a} = -2 \sum_i^n (y_i - (a + bx_i)) = 0$$

$$\frac{\partial R}{\partial b} = -2 \sum_i^n (y_i - (a + bx_i))x_i = 0$$

回帰分析の一般化

- 基底関数: $\phi_j(x)$ ($j = 1 \cdots m$)

- モデル: $y(x) = \sum_j^m \phi_j(x)w_j + \epsilon$

- 残差: $R(\mathbf{w}) = \sum_i^n \left[y_i - \sum_j^m \phi_j(x_i)w_j \right]^2$

$$\frac{\partial R}{\partial w_k} = -2 \sum_i^n \left[y_i - \sum_j^m \phi_j(x_i)w_j \right] \phi_k(x_i) = 0 \quad (k = 1 \cdots m)$$

- 計画行列 (design matrix) $\Phi_{ij} = \phi_j(x_i)$ を導入すると

$$R(\mathbf{w}) = \|\mathbf{y} - \Phi\mathbf{w}\|^2$$

- 最小二乗解: $\Phi^t\Phi\mathbf{w} = \Phi^t\mathbf{y} \Rightarrow \mathbf{w} = (\Phi^t\Phi)^{-1}\Phi^t\mathbf{y}$

リッジ回帰 (Ridge Regression)

- 基底関数が線形独立でない (例: $\phi_1(x) = 1$, $\phi_2(x) = x$, $\phi_3(x) = 1 + x$) と、Gram 行列 ($\Phi^t \Phi$) が特異になる
- 基底関数の数 (例: 多項式の次数) を増やしすぎると過学習 (over-fitting) が生じる
- 正則化最小二乗法 (λ は非負の定数)

$$\begin{aligned} R(\mathbf{w}) &= \sum_i^n \left[y_i - \sum_j^m \phi_j(x_i) w_j \right]^2 + \lambda \sum_j^m w_j^2 \\ &= |\mathbf{y} - \Phi \mathbf{w}|^2 + \lambda \mathbf{w}^t \mathbf{w} \end{aligned}$$

- 最小二乗解

$$(\Phi^t \Phi + \lambda I) \mathbf{w} = \Phi^t \mathbf{y} \Rightarrow \mathbf{w} = (\Phi^t \Phi + \lambda I)^{-1} \Phi^t \mathbf{y}$$

講義日程 (予定)

- 全 8 回 (水曜 2 限 10:25-12:10)
 - ▶ 4 月 5 日 講義 1: 講義の概要・基本的なアルゴリズム
 - ▶ 4 月 19 日 演習 1: 環境整備・C 言語プログラミング・図のプロット
 - ▶ 4 月 26 日 講義 2: 常微分方程式
 - ▶ 5 月 10 日 演習 2 (グループ 1): 基本的なアルゴリズム・常微分方程式
 - ▶ 5 月 17 日 演習 2 (グループ 2): 基本的なアルゴリズム・常微分方程式
 - ▶ 5 月 24 日 休講
 - ▶ 5 月 31 日 休講
 - ▶ 6 月 7 日 講義 3: 連立方程式
 - ▶ 6 月 14 日 演習 3 (グループ 1): 連立方程式
 - ▶ 6 月 21 日 演習 3 (グループ 2): 連立方程式
 - ▶ 6 月 28 日 休講
 - ▶ 7 月 5 日 講義 4: 行列の対角化
 - ▶ 7 月 12 日 演習 4 (グループ 1): 行列の対角化
 - ▶ 7 月 19 日 演習 4 (グループ 2): 行列の対角化
- 2 回目以降の演習はクラスを 2 グループに分けて実施 (グループ 1: 学生証番号が奇数、グループ 2: 偶数)

計算機環境

- 教育用計算機システム
- 知の物理学クラスタ (ai.phys.s.u-tokyo.ac.jp)
 - ▶ 卒業まで利用可 (希望すれば大学院でも)
 - ▶ バッチシステムを使えば、かなり大規模な計算も可能
- 計算機利用・シミュレーションに関する質問は今後も歓迎
 - ▶ Slack
 - ▶ メール computer@phys.s.u-tokyo.ac.jp
 - ▶ Twitter...

計算機実験 II

- 3 年冬学期、金曜 5 限
- 全 8 回程度
- 計算機実験 I で身に付けた知識をもとに、より高度な数値計算手法・アルゴリズムを学び、物理学における具体的な問題への応用を通して実践的な知識と経験を身につける。
 - ▶ モンテカルロ法 (乱択アルゴリズム、モンテカルロ積分、マルコフ連鎖モンテカルロ)
 - ▶ 偏微分方程式の初期値問題
 - ▶ 多体系の量子力学 (横磁場イジング模型、時間発展、量子コンピュータ)
 - ▶ 少数多体系・分子動力学
 - ▶ 最適化とその応用 (共役勾配法、シミュレーテッドアニーリング、他)
 - ▶ ...