電 301 数値解析
第 9 回
関数近似 (3)

- 今回の講義の典拠はおもに
  - ▷ 斎藤, 数値解析入門, 東京大学出版会, 2012
  - ▷ 杉原, 室田, 数値計算法の数理, 岩波書店, 1994
  - ▷ 森, 数値解析, 第 2 版, 共立出版, 2002.

最良多項式と Chebyshev 近似 (1)

- 有界閉区間 $[a, b]$ で定義された連続関数 f を多項式 p によって近似したい場合には，「有限の標本点で $f(x)$ と $p(x)$ を一致させる」のは必ずしも良い方法ではない．
- $\|\cdot\|$ を何らかのノルムとしたとき，$\|f - p\|$ を最小とするような多項式を求めた方が良いこともある．

最良多項式と Chebyshev 近似 (2)

- P_n を n 次の多項式の全体とする.
  $p^* = \arg\min_{p \in P_n} \|f - p\|$ を, f の $\|\cdot\|$ に関する n 次の最良近似多項式という.

- 記号 $\arg\min(\arg\max)$ は, 与えられた評価関数が最小 (最大) となる変数をあらわす記号である.

最良多項式と Chebyshev 近似 (3)

- $p^* = \arg\min_{p \in P_n} \|f - p\|$ という式では, $\|f - p\|$ が評価関数である. また, この問題では, 「変数」として動くのは多項式 p である (多項式 p が n 次の多項式全体の集合 P_n の中を動く).
- $\arg\min$ および $\arg\max$ はよく使われる記号であるが, 明確に定義されずに用いられることが多い記号でもある.

  たとえば `http://reference.wolfram.com/language/ref/ArgMin.ja.html` 参照).

最良多項式と Chebyshev 近似 (4)

- $\|\|\|_\infty$ に関する n 次の最良近似多項式を n 次の Chebychev 近似という.

- Chebychev はロシア人で, ロシア語人名の英語表記が一定でないため, Chebychev の綴りは文献によって異なることがある.

最小二乗近似 (1)

- l_2 ノルムに関する最良近似多項式の計算は比較的簡単．応用上もよく用いられる．これを最小二乗近似という．
- 有界閉区間 $[a, b](\text{ただし } b > a)$ において連続な実数値関数 f を近似する n 次の多項式を構成する問題を考える．

最小二乗近似 (2)

- まず, 基底を $\{1, x, \ldots, x^n\}$ とし, f をもっとも良く近似する $p(x) = \sum_{k=0}^{n} a_k x^{n-k}$ を求める問題を考える. この問題を解くには, $\{a_0, a_1, \ldots, a_n\}$ を求めればよい.
- ここで言う「基底」は関数空間の基底であり, 関数である.

最小二乗近似 (3)

- 基底という言葉が用いられている理由は, n 次の多項式全体の集合 P_n を実数体上のベクトル空間と見たとき, $\{1, x, \ldots, x^n\}$ がこのベクトル空間の基底のひとつだから.
- 関数 f と g の内積を $(f, g) = \int_a^b f(x)g(x)dx$ とする (第 7 回の定義と同じ). 実数値関数を考えているので複素共役は不要.

最小二乗近似 (4)

- $g_{kl} = (x_k, x_l)$, $h_k = (f, x^k)$ とする.
- $\boldsymbol{G} = (g_{kl})_{0 \leq k \leq n, 0 \leq l \leq n}$ を, g_{kl} が定める $(n+1)$ 次の正方行列とする. 任意の多項式 p に対し, $\int_a^b p(x)dx > 0$ だから, $\boldsymbol{G}$ は正定対称行列である.
- $\|f-p\|_2^2 = (f-p, f-p)$ を計算すると, $\|f-p\|_2^2 = (f,f) - 2\sum_{k=0}^{n} a_k h_k + \sum_{k=0}^{n}\sum_{l=0}^{n} a_k a_l g_{kl}$ となる.

最小二乗近似 (5)

- $\boldsymbol{h} = (h_0, \ldots, h_n)^T$, $\boldsymbol{\alpha} = (a_0, \ldots, a_n)^T$ とおくと, 上の式は以下のように書き直される.
  $\|f - p\|_2^2 = (f, f) - 2\boldsymbol{h}^T\boldsymbol{\alpha} + \boldsymbol{\alpha}^T\boldsymbol{G}\boldsymbol{\alpha}$.
- G は正定対称行列なので, $\frac{\partial \|f-p\|_2^2}{\partial \boldsymbol{\alpha}} = \boldsymbol{0}$ を満たす $\boldsymbol{\alpha}$ を用いることで, 最良近似が得られる.

最小二乗近似 (6)

- 解くべき方程式は $-\boldsymbol{h}^T + \boldsymbol{\alpha}^T \boldsymbol{G} = \boldsymbol{0}$ であるが, $\boldsymbol{G}$ が対称行列なので転置して書き直すと, $\boldsymbol{G}\boldsymbol{\alpha} = \boldsymbol{h}$ となる. この連立一次方程式を正規方程式という (後に別の形の正規方程式も出て来るので注意).
- 正規方程式を解くことにより, 最小二乗近似多項式の係数が得られる.

最小二乗近似 (7)

- ただし, 数値計算の誤差を考慮した場合には, 基底 $\{1, x, \ldots, x^n\}$ が定める行列 $\boldsymbol{G}$ は, a, b の値によっては条件数が大きくなることがあり, 好ましくない.
- このような問題を回避するため, 基底として別の多項式系を使うことがある.
- $\{r_0(x), \ldots, r_n(x)\}$ を P_n のひとつの基底 (線形独立な高々n 次の多項式系) とする.

最小二乗近似 (8)

以下の典拠は伊理, 線形代数汎論, 朝倉書店, 2009 および杉原, 室田, 線形計算の数理, 岩波書店, 2009.

- 行列 $\boldsymbol{A}$ に対し, $\|\boldsymbol{A}\|\|\boldsymbol{A}^{-1}\|$ を $\boldsymbol{A}$ の条件数という.
- ノルムが l_2 誘導ノルムである場合には, $\boldsymbol{A}$ の条件数は $\boldsymbol{A}$ の最大特異値を最小特異値で割ったものになる.
- 条件数が大きい行列では, 線形方程式の解の誤差が大きくなることがあり得る.
- 行列 $\boldsymbol{A}$ の特異値とは, $\boldsymbol{A}^H\boldsymbol{A}$ の零でない固有値の平方根である. $\boldsymbol{A}^H$ は行列 $\boldsymbol{A}$ の Hermite 転置.

最小二乗近似 (9)

- $\{r_0(x), \ldots, r_n(x)\}$ が内積 (f, g) に関して正規直交基底であれば, $\boldsymbol{G}$ は単位行列となり, 正規方程式は簡単に解ける.
- 正規性までは要求しなくても, $\{r_0(x), \ldots, r_n(x)\}$ が内積 (f, g) に 関して 直交していて, $\forall k, (r_k, r_k) > 0$ であれば, $\boldsymbol{G}$ が正則な 対角行列となるから, 正規方程式は容易に解ける.

最小二乗近似 (10)

- さらに一般化して, (f, g) のかわりに, 重み付き内積 $(f, g)_w$ を考えることがある.
- $(f, g)_w = \int_a^b f(x)g(x)w(x)dx$ である.
- $w(x)$ は重み関数と呼ばれる関数である. これは, 有限個の点を除いて零にならず, $\int_a^b w(x)dx$ が有限となるような, 非負の連続関数である.

最小二乗近似 (11)

- $\|f-p\|_w = \sqrt{(f-p, f-p)_w}$ と定義する.
- 区間 $[a,b]$ で定義された連続関数 $f(x)$ を $p(x) = \sum_{k=0}^{n} a_k r_k(x)$ で重み付き最小二乗近似する問題を考える. これは, $\|f-p\|_{2,w}$ が最小となるような多項式 p を求める問題である.
- 基底を $\{r_0(x), \ldots, r_n(x)\}$ とする.

最小二乗近似 (12)

- $g_{kl} = (r_k, r_l)_w$, $h_k = (f, r_k)_w$ とし, $\boldsymbol{G} = (g_{kl})_{0 \leq k \leq n, 0 \leq l \leq n}$ を, g_{kl} が定める $(n+1)$ 次の正方行列とする. また, $\boldsymbol{h} = (h_0, \ldots, h_n)^T$, $\boldsymbol{\alpha} = (a_0, \ldots, a_n)^T$ とする.

- 基底関数が $\{1, x, \ldots, x^n\}$ の場合と同様に, $\|f - p\|_{2,w}^2$ の $\boldsymbol{\alpha}$ による偏微分が零であれば, 対応する $p(x)$ は最良近似となる.

最小二乗近似 (13)

- このための条件は, 先ほどと同様に $\boldsymbol{G\alpha} = \boldsymbol{h}$(正規方程式) である.
- まとめると, 基底や内積の取り方を変えると, 正規方程式を構成する行列 $\boldsymbol{G}$ およびベクトル $\boldsymbol{h}$ は変わるが, 正規方程式を解くことで最小二乗近似ができるという事実は不変である.

直交多項式系 (1)

- 基底が直交していれば正規方程式が簡単に解けることは既に述べた.
- 関数系 $(r_1(x), r_2(x), \ldots)$ において $k \neq l$ なら $(r_k, r_l)_w = 0$ となるとき, これを重み w に関する直交関数系という (岩波数学辞典).

上記の直交関数系の定義は $\forall k, (r_k, r_k)_w = 0$ となる場合を許容しており不合理であるが, この用語はすでに定着しているようである.

直交多項式系 (2)

- Fourier 級数が使いやすい理由のひとつに, $\{1, \sin x, \cos x, \sin 2x, \cos 2x, \ldots\}$ が直交関数系になっていることが挙げられる.
- 多項式の列 $(r_1(x), r_2(x), \ldots)$ が直交関数系になっていて, かつ $\forall k,\ (r_k, r_k)_w > 0$ となるとき, これを重み w に関する直交多項式系という.

直交多項式系 (3)

- $\{1, x, x^2, \ldots\}$ に Gram-Schmidt の直交化法を適用することで直交系が得られるが (Legendre 多項式, http://mathworld.wolfram.com/LegendrePolynomial.html), 他にもよく使われる直交多項式系がある.
- 直交多項式系を使うときには, どの区間で積分を計算するか, どのような重みを使うかも併せて定める.

直交多項式系 (4)

▷ Legendre 多項式

- Legendre 多項式 (系) は, 以下の式によって定められる多項式 $p_n(x)$ を集めたものである.

$$p_n(x) = \frac{1}{2^n n!} \frac{d^n}{dx^n}(x^2 - 1)^n, \quad n = 0, 1, 2, \ldots.$$

Legendre 多項式の最初の数項は, $p_0(x) = 1$, $p_1(x) = x$, $p_2(x) = (3/2)x^2 - (1/2)$ である.

直交多項式系 (5)

- Legendre 多項式は区間 $[-1, 1]$ で利用される. 重みは 1 である. よって, 内積は次のように定義される.

$$(p_k, p_j) = \int_{-1}^{1} p_k(x) p_j(x) dx$$

直交多項式系 (6)

- 帰納法と部分積分により, $k \neq j$ であれば $(p_k, p_j) = 0$ であることが示せる.
- また, 同じく帰納法と部分積分を用い, ガンマ関数とベータ関数の性質を使うと, $(p_k, p_k) = 2/(2k+1)$ であることが示せる.
- 証明は [斎藤], pp. 157–158 を参照. ガンマ関数とベータ関数については杉浦, 解析入門 I, 東京大学出版会, 1980 を参照.

直交多項式系 (7)

▷ Laguerre 多項式

- Laguerre 多項式 (系) は, 以下の式によって定められる多項式 $l_n(x)$ を集めたものである.

$$l_n(x) = \frac{e^x}{n!}\frac{d^n}{dx^n}\left(e^{-x}x^n\right), \quad n = 0, 1, 2, \ldots.$$

直交多項式系 (8)

- Leguerre 多項式が多項式であることは一見では明らかではないが, 定義式の右側の e^{-x} は何回微分しても消えることはないので, 最終的に定義式の左側の e^{x} と打ち消し合い, 多項式だけが残る.
- Leguerre 多項式は量子力学で利用される ([物理学辞典]).

直交多項式系 (9)

- Leguerre 多項式は区間 $[0, \infty)$ で利用される. 重みは e^{-x} である. よって, 内積は次のように定義される.

$$(l_k, l_j) = \int_0^\infty l_k(x) l_j(x) e^{-x} dx$$

直交多項式系 (10)

- $k \neq j$ であれば $(l_k, l_j) = 0$ であることと, $(l_k, l_k) = (k!)^2$ であることが帰納法と部分積分によって示せる (ガンマ関数の性質も使う). 証明は [斎藤], pp. 278–279 を参照.
- Fourier 級数や Legendre 多項式と異なり, Leguerre 多項式は半無間開区間 $[0, \infty)$ で定義された関数の近似に使える.

直交多項式系 (11)

▷ Chebychev 多項式

- Chebychev 多項式 (系) は, 以下の式によって定められる多項式 $t_n(x)$ を集めたものである.

$$t_n(x) = \cos(n \arccos x), \quad n = 0, 1, 2, \ldots.$$

直交多項式系 (12)

Chebychev 多項式が実際に多項式になっていることを確認する.

- $n = 0$ とおくと, $t_0(x) = \cos 0 = 1$ となる.
- $n = 1$ とおくと, $t_1(x) = \cos(\arccos x) = x$ となる.
- 以下, $k \leq n$ に対して t_k が x の高々 k 次の多項式になっているならば, $t_{n+1}(x)$ は x の高々 $n+1$ 次の多項式になることを確認する.

直交多項式系 (13)

- $\arccos x = \theta$ とおき，三角関数の公式 $\cos(\alpha+\beta)+\cos(\alpha-\beta) = 2\cos\alpha\cos\beta$ において $\alpha = n\theta$, $\beta = \theta$ とおくと，$\cos((n+1)\theta))+\cos((n-1)\theta) = 2\cos n\theta\cos\theta$ だから，これに $x = \cos\theta$ を代入し，t_n の定義を思い出すと，$t_{n+1}(x)+t_{n-1}(x) = 2xt_n$ となる．したがって，$t_{n+1}(x)$ は x の高々 $n+1$ 次の多項式である．

直交多項式系 (14)

- Chebychev 多項式は区間 $[-1, 1]$ で利用される．重みは $1/\sqrt{1-x^2}$ である．よって，内積は次のように定義される．

$$(t_k, t_j) = \int_{-1}^{1} t_k(x) t_j(x) \frac{dx}{\sqrt{1-x^2}}.$$

直交多項式系 (15)

- $\theta = \arccos x$ $(x = \cos\theta)$ と変数変換すると, $t_k = \cos k\theta, t_l = \cos l\theta, \sqrt{1-x^2} = \sin\theta, dx = -\sin\theta d\theta$ だから,

$$(t_k, t_j) = \int_{\pi}^{0} \cos k\theta \cos l\theta \frac{-\sin\theta d\theta}{\sin\theta}$$

となる. よって, Fourier 級数展開と同じ計算によって, $k \neq j$ なら $(t_k, t_j) = 0$ で, $(t_0, t_0) = \pi$, $k \geq 1$ なら $(t_k, t_k) = \pi/2$ となることがわかる.

直交多項式系 (16)

- Chebychev 多項式 $t_k(x)$ は, $x_j = \cos\dfrac{j\pi}{k}$, $0 \leq j \leq k$ において極値を取り, すべての x_j で $t_k(x_j)$ の絶対値が 1 で, かつ $t_k(x_j)$ と $t_k(x_{j+1})$ は符号が異なるという特徴がある. この特徴が数値計算で有用であるため, Chebychev 多項式は数値解析でよく用いられる.

直交多項式系 (17)

- 直交多項式はこれ以外にもあるが, この講義ではこれ以上述べない.
- 各種の直交多項式にはそれに適した用途があるが, 一般的な周期関数の近似には Fourier 級数を使った方が無難.

Scilab における Chebychev 多項式 (1)

- Scilab がサポートしている直交多項式は Chebychev 多項式のみである． 関数 `chepol` によって Chebychev 多項式を求めることができる．
- 関数 `chepol` は, 2 個の引数 (第一引数は整数, 第二引数は文字) を受け取り, 第一引数で指定された次数の第二引数を変数とする Chebychev 多項式を生成する．
- 使用例を次ページに示す．

```
-->chepol(3,'x')
 ans  =
           3
  - 3x + 4x

-->chepol(5,'y')
 ans  =
             3     5
    5y - 20y + 16y
```

注意

- -->はScilabのプロンプトである.
- 以下, プロンプトと書くことと書かないことがあるが, Scilabでこれ (-->) を入力するわけではないので注意.

Chebychev 補間

- Chebychev 多項式の零点を標本点とした Lagrange 補間を Chebychev 補間という.
- Lagrange 補間における標本点の取り方は任意であり, 関数近似の精度は標本点の取り方に依存する. この誤差の下限を見積ることができるが, Chebychev 補間は下限に近い値を達成することが知られている ([杉原, 室田]).

最小二乗法による曲線のあてはめ (1)

以下しばらく J. Stoer, R. Bulirsch, Introduction to Numerical Analysis, 3/e, Springer, 2002 および R. Butt, Introduction to Numerical Analysis using MATLAB, Infinity Science Press, 2008 に基づいて説明する.

- 実験によって 得られた n 個の測定点 $\{(x_k, y_k) : 1 \leq k \leq n\}$ に曲線 (あるいは直線) をあてはめる, という問題を考える.
- 補間多項式によってこの目的は達成されるが, 測定値に誤差が含まれる場合には, 「曲線が測定点を通過する」ことは必ずしも合理的とはいえない.

最小二乗法による曲線のあてはめ (2)

- 代替案として, パラメータ $\boldsymbol{\alpha}$ を含む非線形関数 $g(x, \boldsymbol{\alpha})$ を準備し, $\sum_{k=1}^{n} (y_k - g(x_k, \boldsymbol{\alpha}))^2$ が最小となる $\boldsymbol{\alpha}$ を求めるという方法が考えられる. これを (非線形) 最小二乗法と呼ぶことがある. 補間とも最良近似とも違う考え方なので注意.
- 関数 $g(x, \boldsymbol{\alpha})$ をどう選ぶべきかは問題による. 線形関数と多項式 (教科書 121~126 ページ) はふつうに用いられる.

最小二乗法による曲線のあてはめ (3)

- 非線形最小二乗法は, 物理現象の数学モデルのパラメータを実験的に定めるときにも用いられる. この場合には, 関数 $g(x, \boldsymbol{\alpha})$ は物理法則から導かれる.
- $\boldsymbol{\alpha}$ の次元はデータの点数とは無関係だが, 次元がデータ点数より大きいと $\boldsymbol{\alpha}$ を一意的に定められない可能性が高い.

最小二乗法による曲線のあてはめ (4)

- f が非線形関数の場合には, 解くべき問題は非線形最小化問題

$$\min_{\boldsymbol{\alpha}} \sum_{k=1}^{n} \left(y_k - g(x_k, \boldsymbol{\alpha})\right)^2$$

である．この問題は非線形最小化問題であり，一般には反復解法によって数値的に解く以外の解法はない.

最小二乗法による曲線のあてはめ (5)

- $g(x)$ が既知の関数の線形結合であらわされている場合, すなわち $g(x) = \sum_{k=1}^{m} \alpha_k r_k(x)$ となっていて, $r_k(x)$ は既知であるが α_k は未知である場合には, 問題はもう少し簡単になる.
- $\boldsymbol{y} = (y_1, \ldots, y_n)^T$, $\boldsymbol{r}(x) = (r_1(x), \ldots, r_m(x))$, $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_m)$ とする. 列ベクトルと行ベクトルが混在していること, 一般に $m \neq n$ であることに注意.

最小二乗法による曲線のあてはめ (6)

- $\boldsymbol{R} = \begin{pmatrix} \boldsymbol{r}(x_1) \\ \vdots \\ \boldsymbol{r}(x_n) \end{pmatrix}$ とする．$\|\boldsymbol{y} - \boldsymbol{R\alpha}\|_2^2$ が最小となる $\boldsymbol{\alpha}$ を求めれば, 曲線のあてはめができる．
- $\|\boldsymbol{y}-\boldsymbol{R\alpha}\|_2^2 = \boldsymbol{y}^T\boldsymbol{y}-2\boldsymbol{y}^T\boldsymbol{R\alpha}+\boldsymbol{\alpha}^T\boldsymbol{R}^T\boldsymbol{R\alpha}$ である．
- $\frac{\partial\|\boldsymbol{y}-\boldsymbol{R\alpha}\|_2^2}{\partial\boldsymbol{\alpha}}$ が零となる $\boldsymbol{\alpha}$ が最適解の候補．

最小二乗法による曲線のあてはめ (7)

- $\mathbf{0}^T = -\boldsymbol{y}^T\boldsymbol{R}+\boldsymbol{\alpha}^T\boldsymbol{R}^T\boldsymbol{R}$, あるいはこれを $\boldsymbol{y}^T\boldsymbol{R}$ を移項して転置した, $\boldsymbol{R}^T\boldsymbol{R}\boldsymbol{\alpha} = \boldsymbol{R}\boldsymbol{y}$ を解けばよい.
- 方程式 $\boldsymbol{R}^T\boldsymbol{R}\boldsymbol{\alpha} = \boldsymbol{R}\boldsymbol{y}$ も正規方程式と呼ばれる. 関数の最小二乗近似で出てきた正規方程式とは形が違うので注意.
- 非線形最小二乗法では, 正規方程式に解がないこともあり得る.

Scilab による線形回帰 (1)

- Scilab で直線のあてはめをこなう関数は `reglin` (教科書 125 ページ).
- この関数は, スカラーに関する直線回帰だけでなく, $\boldsymbol{x}$ と $\boldsymbol{y}$ がベクトルで, $\boldsymbol{y} \simeq \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}$ という線形回帰をおこないたい場合にも使える. ただし, 記号 $\simeq$ を近似の意味で用いている.
- $(\boldsymbol{x}_i, \boldsymbol{y}_i)_{i=1,\ldots,Ns}$ が Ns 個の標本で (列ベクトルとする), $\dim \boldsymbol{x} = n_x$, $\dim \boldsymbol{x} = n_y$ とする.

Scilab による線形回帰 (2)

- 線形回帰をおこなう際に次元を合わせる必要があるので, $\boldsymbol{A} \in \mathbb{R}^{n_y \times n_x}$, $\boldsymbol{b} \in \mathbb{R}^{n+y}$ となる.
- `reglin` を使うときには, $\boldsymbol{x}_i$ を横に Ns 個並べた行列 (仮に `xmat` とする) と, $\boldsymbol{y}_i$ を横に Ns 個並べた行列 (仮に `ymat` とする) を作り, `[A,b,s]=reglin[xmat,ymat]` とする.
- `A`, `b` には回帰の結果得られる行列およびスカラーが, `s` には回帰残差の標準偏差が返される.

Scilab による線形回帰 (3)

x と y がスカラーの場合の直線回帰: 次に, $x = 1, 2 \ldots, 100$, $y = x +$ (正規分布する乱数) として, 線形回帰をおこなった結果を示す. 実行結果から, a $\simeq 1$, b $\simeq 0$ となっていることが確認できる.

```
x=1:100;
y=x+rand(1,100,'normal');
[a,b,s]=reglin(x,y);

-->[a,b,s]
 ans  =
    0.9997142   - 0.0073729     1.0491074
```

Scilab による線形回帰 (4)

xmat が 2 行 100 列, ymat が 3 行 100 列の乱数行列 (正規分布) の場合の線形回帰の例を次に示す. 空白を減らして表示しているので注意.

```
xmat=rand(2,100,'normal');
ymat=rand(3,100,'normal');
[A,b,s]=reglin(xmat,ymat);
```

```
-->A
A   =
-0.0940391  -0.0510797
-0.0298481   0.0738067
-0.0712583   0.0198992
```

```
-->b
b   =
-0.0986652
-0.0063689
-0.0370478
```

```
-->s
s   =
 1.0819806  0.0205290  0.0777786
 0.0205290  1.0073556  0.1069899
 0.0777786  0.1069899  0.9121303
```

Scilab による曲線のあてはめ (1)

- Scilab でデータに曲線をあてはめるには非線形最小二乗法問題を解く関数 `leqstsq` を使うが, 使い方に若干の工夫が必要である.
- 例として, 曲線 $y = f_p(a, x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3$ にデータをあてはめる問題を考える. x が -1 から 1 まで 0.1 刻みで動き, $(a_0, a_1, a_2, a_3) = (1, 1, 1, 1)$ で, y は $f_p(a, x)$ に $[-0.1, 0.1]$ の範囲に一様分布する雑音を加えたものとする.

Scilab による曲線のあてはめ (2)

- `leqstsq` には引数として最小化したい関数, データ, パラメータの初期値を与える. 初期値次第では適切なあてはめができないことがあるので注意. その場合には乱数等を用いた再試行が必要.
- `leqstsq` はパラメータ初期値と同じ次元のパラメータを探索するので, 初期値の次元が実際のパラメータより大きくても, とりあえず動く.
- 次のページに Scilab のプログラムを示す.

Scilab による曲線のあてはめ (3)

```
function y=fp(a,x) //パラメータを含む関数を定義
  y=a(1)+a(2).*x+a(3).* (x.^2) + a(4).* (x.^3)
endfunction

function e=fopt(a,xmat,ymat) //最小二乗法で使う関数
  e=(norm(fp(a,xmat)-ymat))^2;
endfunction

aT=[1 2 3 4]; //パラメータの真値
xmat=-1:.1:1; //x 軸上の標本点
ymat=fp(aT,xmat)+0.1*(rand(xmat)-0.5); //y の値を生成

a0=[1 1 1 1]; //パラメータ推定値の初期値
[f,aEst]=leastsq(list(fopt,xmat,ymat),a0); //最小化
```

Scilab による曲線のあてはめ (4)

実行結果の例:

```
->aEst
 aEst  =
    1.0050031     1.9672115     2.9836197     4.0484874
```

- 上記のようにすると, f には最小化された関数値が, aEst にはパラメータ推定値が格納される.
- leqstsq にはより多くのパラメータを与えることができるが, この講義では略す. 興味がある者は Scilab のオンラインマニュアルを参照すること.
- 関数定義における x.^3 などの記述は, 変数が観測値を集めたベクトルである場合に対応するためのものである. こうしないとエラーになる.

Scilab による曲線のあてはめ (5)

- 関数 `datafit` もデータへの曲線をおこなう関数である．初期値に関する注意は `leqstsq` と同様.

- `leqstsq` との違いは, パラメータを列ベクトルにしなければならないことと, データとして x 軸と y 軸の値をまとめた行列を与えることである．次ページに Scilab のプログラムを示す (先ほどと同じ問題).

Scilab による曲線のあてはめ (6)

```
function y=fp(a,x) //パラメータを含む関数を定義，前と同じ
 y=a(1)+a(2).*x+a(3).*x.^2+a(4).*x.^3
endfunction

function e=fopt(a,z) //誤差関数，x 軸と y 軸の値がペアであることを想定
 e=(z(2)-fp(a,z(1)))^2;
endfunction

aT=[1;2;3;4];
xmat=-1:.1:1;
ymat=fp(aT,xmat)+0.1*rand(xmat,'normal');

a0=[1;1;1;1]; //パラメータは列ベクトル
[aEst,err]=datafit(fopt,[xmat;ymat],a0); //返却値はパラメータ推定値と誤差
```

Scilab による曲線のあてはめ (7)

- 次に, $\boldsymbol{x}$ が 2 次のベクトル, y がスカラーで, $y \simeq a_0 + \boldsymbol{a}_1^T \boldsymbol{x} + \boldsymbol{x}^T \boldsymbol{A}_2 \boldsymbol{x}$ という 2 次関数にデータをあてはめたいという問題を考える. 標本点 $\boldsymbol{x}$ の各成分は区間 $[0, 1]$ に値を取る一様乱数から生成されているものとする.

- 次ページに Scilab のプログラムを示す.

leastsq の実行例,
[f,aEst] = leastsq (list (fopt,xm,ym), a0); という部分が非線形最小二乗法を実行している部分で, 残りは関数定義や標本の準備. 初期値 (a0 しだいで収束しないことがあるので注意.

```
function y=fp(a,x)
 y=a(1)+a(2:3)'*x+x'*[a(4:5) a(6:7)]*x;
endfunction

function e=fopt(a,xm,ym)
 e=0;
 for i=1:size(xm,2)
   e=e+(ym(i)-fp(a,xm(:,i)))^2;
 end
endfunction

aT=[1;1;1;1;0;0;1];
samples=100;
xm=rand(2,samples);
ym=[];
for i=1:samples
 ym=[ym fp(aT,xm(:,i))];
end

a0=rand(7,1);
[f,aEst]=leastsq(list(fopt,xm,ym),a0);
```

datafit の実行例,
[aEst,err] = datafit (fopt, [xm;ym], a0); という部分が非線形最小二乗法を実行している部分で, 残りは関数定義や標本の準備. こちらも初期値 (a0 しだいで収束しないことがあるので注意.

```
function y=fp(a,x)
 y=a(1)+a(2:3)'*x+x'*[a(4:5) a(6:7)]*x;
endfunction

function e=fopt(a,z)
 e=norm(z(3)-fp(a,z(1:2)))^2;
endfunction

aT=[1;1;1;1;0;0;1];
samples=100;
xm=rand(2,samples);
ym=[];
for i=1:samples
  ym=[ym fp(aT,xm(:,i))];
end

a0=rand(7,1);
[aEst,err]=datafit(fopt,[xm;ym],a0);
```

Scilab による曲線のあてはめ (10)

- 続いて, $\boldsymbol{x}$ と $\boldsymbol{y}$ がともに 2 次のベクトルで, $\boldsymbol{y} \simeq \boldsymbol{a}_0 + \boldsymbol{A}_1\boldsymbol{x} + \boldsymbol{a}_2(\boldsymbol{x}^T\boldsymbol{x})$ という 2 次関数に $\boldsymbol{a}_0$ と $\boldsymbol{a_2}$ は 2 次のベクトル, $\boldsymbol{A}_1$ は 2 次の正方行列である. データをあてはめたいという問題を考える. 標本点 $\boldsymbol{x}$ の各成分は区間 $[0, 1]$ に値を取る一様乱数から生成されているものとする.

- 次ページに Scilab のプログラムを示す.

leastsq の実行例, 先ほどと同様に, [f,aEst] = leastsq (list (fopt,xm,ym), a0); という部分が非線形最小二乗法を実行している部分で, 残りは関数定義や標本の準備. 要するに, 関数定義を工夫すれば, 標本がベクトルの場合にも柔軟に対応できる.

```
function y=fp(a,x)
 y=a(1:2)+[a(3:4) a(5:6)]*x+a(7:8)*x'*x;
endfunction

function e=fopt(a,xm,ym)
 e=0;
 for i=1:size(xm,2)
   e=e+norm(ym(:,i)-fp(a,xm(:,i)))^2;
 end
endfunction

aT=[1;1;1;0;0;1;1;1];
samples=100;
xm=rand(2,samples);
ym=[];
for i=1:samples
 ym=[ym fp(aT,xm(:,i))];
end

a0=rand(8,1);
[f,aEst]=leastsq(list(fopt,xm,ym),a0);
```

datafit の実行例, こちらも [aEst,err] = datafit (fopt, [xm;ym], a0); という部分が非線形最小二乗法を実行している部分で, 残りは関数定義や標本の準備. やはり, 関数定義を工夫すれば, 標本がベクトルの場合にも柔軟に対応できる.

```
function y=fp(a,x)
 y=a(1:2)+[a(3:4) a(5:6)]*x+a(7:8)*x'*x;
endfunction

function e=fopt(a,z)
 e=norm(z(3:4)-fp(a,z(1:2)))^2;
endfunction

aT=[1;1;1;0;0;1;1;1];
samples=100;
xm=rand(2,samples);
ym=[];
for i=1:samples
 ym=[ym fp(aT,xm(:,i))];
end

a0=rand(8,1);
[aEst,err]=datafit(fopt,[xm;ym],a0);
```