電 301 数値解析
第 7 回
関数近似 (1)

関数近似とは何か (1)

- 関数近似 (関数の近似) という言葉は, いくつかの教科書で, 未定義で用いられている.
- この講義では, 仮に, 関数を, 性質が既知の関数 (系) を使って (近似的に) 表現することと定義しておく.

関数近似とは何か (2)

- 実用上は, もう少し意味を狭めて, 関数を, 性質が既知の関数 (系) の線形結合によって (近似的に) 表現することとした方が使いやすい. 今後は, この講義では, 線形結合に限定して議論を進める.
- 表現が厳密であることも, 誤差を含む近似表現であることもあり得る.

関数近似とは何か (3)

- $f(x)$ が周期 2π の周期関数であるとき,
  $$\frac{a_0}{2} + \sum_{n=1}^{\infty} (a_n \cos nx + b_n \sin nx)$$
  という表現を $f(x)$ の Fourier 級数展開という (岩波数学入門辞典).

  ただし, x は実数で, $f(x)$ は実数値関数とする. また,
  $a_n = \dfrac{1}{\pi}\displaystyle\int_{-\pi}^{\pi} f(x)\cos nx\,dx$, $b_n = \dfrac{1}{\pi}\displaystyle\int_{-\pi}^{\pi} f(x)\sin nx\,dx$ である.

関数近似とは何か (4)

- 関数 $f(x)$ を Fourier 級数展開したものを $F.S.[f]$ と書くことにする

  (K. B. Howell, Principles of Fourier Analysis, Chapman& Hall/CRC, 2001 で使われている機能．

- Fourier 級数は関数近似の一種である．関数 $f(x)$ の性質は必ずしも明らかでないのに対し，三角関数 $\sin nx$ および $\cos nx$ の性質はよくわかっている．

関数近似とは何か (5)

- 関数が Fourier 展開できる (Fourier 級数が収束する) ための十分条件としては, たとえば $f(x)$ が区分的に連続, というものがある (Howell, 2001.).
- 関数が区分的に連続とは, 関数が有限 $[-\pi, \pi]$ 内に持つ不連続点の個数は高々有限個で, α が f の不連続点であるときには, x が α に右から近付いたときの極限 (右極限) と左から近付いたときの極限 (左極限) の両方が有限値となることをいう.

関数近似とは何か (6)

- Fourier 級数の収束条件としては上記以外のものも知られている (岩波数学辞典).
- 区間を $[-\pi, \pi]$ と取ったのは便宜上の措置であり, 長さが 2π であればどのように取ってもよい.

関数近似とは何か (7)

- 周期が 2π でなく T の場合には,
  $$a_n = \frac{2}{T}\int_0^T f(x)\cos 2\pi\frac{nx}{T}\,dx,$$
  $$b_n = \frac{2}{T}\int_0^T f(x)\sin 2\pi\frac{nx}{T}\,dx$$ となる.
- $\sin nx$ と $\cos nx$ のかわりに e^{inx} を使うこともよく行われる (複素 Fourier 級数).

関数近似とは何か (8)

- $f(x)$ が区分的に連続のとき, その Fourier 級数は, $f(x)$ の連続点では $f(x)$ に収束するが, 不連続点ではその点における右極限と左極限の平均値に収束する.
- フーリエ級数を第 N 項まで打ち切ったもの (部分和) も応用上よく使われる.

関数近似とは何か (9)

- 関数の Fourier 級数展開は近似とは限らないが, Fourier 級数が無限個の係数が零でない項を持つときには, 部分和は関数の近似になっている.
- Fourier 級数展開で三角関数を使って他の関数を表現するのは, 三角関数が周期現象の表現や線形回路網の解析に適しているから. とはいえ, 絶対に三角関数を使わねばならないというわけではない.

関数近似とは何か (10)

- 三角関数以外で, もっとも一般的なのは, 多項式を使った展開である. 関数の Taylor 展開は多項式による展開である.
- 関数 $f(x)$ の Taylor 展開が無限個の係数が零でない項を持つとき, それを最初の N 項で打ち切ったものは, 関数近似になる.

関数近似とは何か (11)

- 関数を近似するときには, 単に既知の関数系で展開するよりは, 直交関数系と呼ばれる関数系で展開した方が, 近似に関する議論がやりやすい (直交性の意味は後述). 三角関数は直交関数系になっているが, 多項式は直交関数系になっていない.
- このため, 多項式のかわりに, 直交化した多項式がよく用いられる.

関数近似とは何か (12)

- 応用で使われる直交多項式には, Legendre 多項式, Chebychev 多項式, Jacobi 多項式, Laguerre 多項式, Hermite 多項式といったものがある (杉原, 室田, 数値計算法の数理, 岩波書店, 1994; これらをすべてこの講義で取り扱うわけではない)
.

直交性のメリット (1)

- 先の議論で「直交」という言葉が出て来た.
- 直交性を導入することの利点を, 2 次元数ベクトル空間を例に取って説明する.
- $\boldsymbol{a}_1 = (a_{11}, a_{21})^T$, $\boldsymbol{a}_2 = (a_{12}, a_{22})^T$ を線形独立なベクトル, $\boldsymbol{b}$ を 2 次の数ベクトルとする.

直交性のメリット (2)

- $\boldsymbol{b}$ を $\boldsymbol{a}_1$ と $\boldsymbol{a}_2$ の線形結合で書き表すには, $\boldsymbol{a}_1 x_1 + \boldsymbol{a}_2 x_2 = \boldsymbol{b}$ という方程式を解かねばならない.
- $\boldsymbol{a}_1$ の成分と $\boldsymbol{a}_2$ の成分を横にならべて行列 $\boldsymbol{A}$ を作ると, 上記は連立一次方程式 $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ を解くことに相当する.

直交性のメリット (3)

- 一方, $(\boldsymbol{a}_1, \boldsymbol{a}_2)$ が正規直交基底になっているときには, 内積 $(\boldsymbol{a}_i, \boldsymbol{b})$ $(i = 1, 2)$ を計算することにより, $\boldsymbol{b} = (\boldsymbol{b}, \boldsymbol{a}_1)\boldsymbol{a}_1 + (\boldsymbol{b}, \boldsymbol{a}_2)\boldsymbol{a}_2$ となり, 計算はずっと簡単 (連立一次方程式を解く必要がない).

直交性のメリット (4)

- また, n 次のベクトル空間において, ベクトル $\boldsymbol{e}_1, \ldots, \boldsymbol{e}_n$ が (正規) 直交基底となっていれば, ベクトル $\boldsymbol{x}$ のベクトル $\boldsymbol{e}_1, \ldots, \boldsymbol{e}_k$(ただし $k \leq n$) が張る部分空間への直交射影は, ベクトル $\boldsymbol{x}$ のその部分空間における最良近似になっていることが示せる.

関数空間 (1)

- 線形代数は, 有限次元ベクトル空間における線形写像の理論であった. (ベクトル空間の公理については電気数学 I の内容なのでこの講義では述べない).
- 線形代数における定理を無限次元空間に拡張しようとする試みが 20 世紀初頭に始まった.

関数空間 (2)

- 以下しばらく, 便宜上, 関数の定義域を $\mathbb{R}$ とする.
- たとえば「勝手に取られた関数をなるべくうまく連続関数で近似したい」といった問題を取り扱うときには, 最適解を探す空間 (この例には連続関数の全体) がどのような構造を持っているかが問題となる.

関数空間 (3)

- $\mathbb{R}$ で定義された実数値関数の全体を V とする．$f, g \in V$ と $\alpha \in \mathbb{R}$ に対し，$f+g$ および αf を次のように定義する：

$$
\begin{aligned}
f+g: &\quad \mathbb{R} \ni x \mapsto f(x)+g(x) \in \mathbb{R} \\
\alpha f: &\quad \mathbb{R} \ni x \mapsto \alpha f(x) \in \mathbb{R}
\end{aligned}
$$

関数空間 (4)

- 前ページの記法には不慣れな者が多いと思うが,
  - $f+g$ は各点 x において $f(x)+g(x)$ という値を取る関数
  - αf は各点 x において $\alpha f(x)$ という値を取る関数

と定義する, という意味である.

関数空間 (5)

- $\mathbb{R}$ で定義された実数値関数の全体 V は, 以上にようにょって定義された加算およびスカラー倍の演算によって, ベクトル空間となる.
- 適切に加算およびスカラー倍の演算を定める (導入する) ことで関数全体の集合をベクトル空間にするというのは要注意点.

関数空間 (6)

- 関数全体がつくるベクトル空間を関数空間という (岩波数学入門辞典)
- 以下の講義では, 関数を関数空間というベクトル空間の要素として取り扱う．ただし, f を太字にすることはない (かえって混乱を招くから).

関数空間 (7)

- 連続関数を操作して, もとの関数を近似するのだから, 連続関数の加算とスカラー倍くらいはできなければお話にならない.
- 換言すると, 連続関数の全体がベクトル空間になっていないと困る.
- 幸い, 連続関数の全体はベクトル空間をなす.

関数空間 (8)

- 上述の事実は, 連続関数全体は関数全体が作るベクトル空間の部分ベクトル空間であると言い換えられる.
- ところで, 近似について論ずるのであれば, 「関数 f と関数 g の違い」を定量的に定められなければならない.

関数空間 (9)

- ベクトル空間 V にノルム (記号 $\|\cdot\|$ であらわす) が定まっているときには, $\|f - g\|$ によって「f と g の違い」を測ることができる.
- ノルムが定められたベクトル空間をノルム空間という (岩波数学入門辞典).

関数空間 (10)

- ノルムとは, 以下の 3 条件 (ノルムの公理) を満たす実数値関数なら, 何でもよい.

  1. $\|0\| = 0$,
  2. $\forall \alpha \in \mathbb{C}, \forall f \in V, \|\alpha f\| = |a| \|f\|$,
  3. $\forall f, g \in V, \|f + g\| \leq \|f\| + \|g\|$.

関数空間 (11)

- 「関数列 $(g_k)_{k\in\mathbb{N}}$ を f に収束させるようなアルゴリズムを作りたい」といった問題を考えるときには,「収束」「極限」といった概念を明確にしておく必要がある．このためには,位相構造 (開集合) が定まっていればよい.

関数空間 (12)

- 解析学では「コーシー列は収束する」という性質が必要となることが多い．この性質を完備あるいは完備性という．関数空間は完備とは限らないが，応用上は完備な関数空間の方が使いやすい．
- 線形代数において，ベクトルを別のベクトルに移す写像のうち，もっともよく使われる (使いやすい) ものは線形写像である．関数空間でもこれは同様で，線形写像が重要な役割を果たす．

関数空間 (13)

- 関数空間を特徴付けるキーワードは
  1. ベクトル空間　2. ノルム
  3. 線形写像　4.(完備性)
- 関数空間は無限次元ベクトル空間であることが普通.

関数空間 (14)

- 位相構造 (開集合系) が定まっている空間を位相空間, 2 点間の距離が定まっている空間を距離空間, ノルムが定まっているベクトル空間がノルム空間である.
- ノルム空間では, ノルムを使って 2 要素 (ベクトル) のあいだの距離を定めることができる.

関数空間 (15)

- 距離空間では, 距離を使って開集合を定めることができる.
- したがって, 「ノルムによって距離を定める」「距離によって位相を定める」という操作をすることより, 以下の包含関係が成り立つ. この包含関係で等号は成立しない.

$$\boxed{\text{ノルム空間} \subset \text{距離空間} \subset \text{位相空間}}$$

関数空間 (16)

- 以上のような議論をした理由は, 収束などの議論をする際に重要なのは位相であって, ノルムそのものではないから.
- 異なるノルムが同一の位相構造を定めることも, そうでないこともある.

次に述べる事実の典拠は, 前者が R. E. Megginson, An Introduction to Banach Space Theory, Springer, 1998, 後者が黒田, 関数解析, 共立出版, 1980.

関数空間 (17)

- 同じ位相構造を定めるノルムを, 同値なノルムという.
- ノルム空間 V に 2 個のノルム $\|\cdot\|_A$ と $\|\cdot\|_B$ が定義され,
  $\exists a, b > 0, \forall x \in V, a\|x\|_A \leq \|x\|_B \leq b\|x\|_A$ となれば, $\|\cdot\|_A$ と $\|\cdot\|_B$ は同値である.

関数空間 (18)

- ベクトル空間 V に 2 個のノルム $\|\cdot\|_A$ と $\|\cdot\|_B$ が定められているものとする.
- $\|\cdot\|_A$ と $\|\cdot\|_B$ が同値でないときには, 集合としての V は共通でも, $(V, \|\cdot\|_A)$ と $(V, \|\cdot\|_B)$ とは位相空間としては別物である.

関数空間 (19)

- 一方, $\|\cdot\|_A$ と $\|\cdot\|_B$ が同値であれば, $(V, \|\cdot\|_A)$ と $(V, \|\cdot\|_B)$ とは, ノルムの値が違っても位相空間としては同じである.
- 同値なノルムが複数あるときには, 収束などの議論をする際に, 計算がもっとも簡単なノルムを選べる, という利点がある.

関数空間 (20)

- V が有限次元ノルム空間であるとき, V で定義されたノルムはすべて同値であることが証明できる. また, 有限次元ノルム空間は完備であることも証明できる ([黒田]).
- 以下しばらく, V は有限次元で, V の要素 x が基底を用いて $(x_1, \ldots, x_n)$ と成分表示されているものとする.

関数空間 (21)

有限次元ベクトル空間で 応用上 よく使われるノルム (下付き添字に注意)

$$
\begin{aligned}
&l_1\text{ノルム} && \|x\|_1 && = \textstyle\sum_{k=1}^n |x_k|, \\
&l_2\text{ノルム} && \|x\|_2 && = \left(\textstyle\sum_{k=1}^n |x_k|^2\right)^{1/2} \\
&l_p\text{ノルム} && \|x\|_p && = \left(\textstyle\sum_{k=1}^n |x_k|^p\right)^{1/p}
\end{aligned}
$$

$(1 \le p < \infty, l_p$ノルムは $p = 1, 2$ の場合を含む$)$.

$$
\text{無限大ノルム} \quad \|x\|_\infty = \max\{|x_k|, 1 \le k \le n\}
$$

関数空間 (22)

- Scilab で有限次元ベクトルの l_p ノルムを計算する関数は `norm` で, p を指定しないと l_2 ノルムが計算される.
- 線形写像のノルムの定め方については次回.
- $0 < p < 1$ に対し, $(\sum_{k=1}^{n} |x_k|^p)^{1/p}$ はノルムにならないが, $x, y \in V$ に対し,
  $d(x, y) = (\sum_{k=1}^{n} |x_k - y_k|^p)$ は距離を定める.

典拠: J. Yeh, Real Analysis, 3rd ed., World Scientific, 2014.

関数空間 (23)

V を実数の無限列 $(x_k)_{k\in\mathbb{N}}$ の全体が作る集合とする．V は加算とスカラー倍を成分ごとに定義することで無限次元ベクトル空間となる．この空間でよく使われるノルム (に対応する関数) は以下の通り．

$$
\begin{array}{lll}
l_1\text{ノルム} & \|x\|_1 & = \sum_{k=1}^{\infty} |x_k|, \\
l_2\text{ノルム} & \|x\|_2 & = \left(\sum_{k=1}^{\infty} |x_k|^2\right)^{1/2} \\
\text{無限大ノルム} & \|x\|_\infty & = \sup\{|x_k|, 1 \le k < \infty\}
\end{array}
$$

関数空間 (24)

- ノルムは実数値関数でなければならなかったから, V からノルム空間を作るためには, ノルム関数の値が無限大の要素を除去しなければならない.

関数空間 (25)

- 上述のノルム関数が定めるノルム空間は以下の通り.

  ▷ $l_1 = \{x \in V : \|x\|_1 < \infty\}$,

  ▷ $l_2 = \{x \in V : \|x\|_2 < \infty\}$,

  ▷ $l_\infty = \{x \in V : \|x\|_\infty < \infty\}$

関数空間 (26)

- $1 \le p < \infty$ に対し，$\|x\|_p = (\sum_{i=1}^{\infty} |x_i|^p)^{1/p}$ とし，$l_p = \{x \in V : \|x\|_p < \infty\}$ とする．$\|\cdot\|_p$(を l_p に制限したもの) は p-ノルムと呼ばれる．
- $0 < p < 1$ に対し，$(\sum_{k=1}^{\infty} |x_k|^p)^{1/p}$ はノルムにならない．ただし，$x, y \in V$ に対し，$d(x, y) = (\sum_{k=1}^{\infty} |x_k - y_k|^p)$ とすると，これは距離を定める．

典拠: J. Yeh, Real Analysis, 3rd ed., World Scientific, 2014.

関数空間 (27)

- 以下, 数学的に正確に定義のみ述べるが, 工学系の学生に馴染みがない用語が多数出てくるので, 理解でなくてもよい.
- V を, ほとんど至るところで有限値を取る可測関数全体とする. V はベクトル空間であることが証明できる ([Yeh]). $f \in V$ とする. $\|f\|_p = \left(\int (f(x))^p \, d\mu\right)^{1/p}$ と定義する ($d\mu$ は定義域の測度に関する積分).

関数空間 (28)

- $\|\cdot\|_p$ はノルムの公理を一部満たさない ($f \neq 0$ であっても $\|f\|_p = 0$ となることがある) が, ほとんど至るところ零の関数に関する同値類を作ることで, ノルムの公理をすべて満たすようにできる.
- $\mathcal{L}_p = \{f \in V : \|f\|_p < \infty\}$ とし, L_p を $\mathcal{L}_p$ の上記の同値関係による商空間としたとき, $\|\cdot\|_p$ は L_p におけるノルムになる.

関数空間 (29)

- L_∞ は, essential supremum という概念を用いて定義される (この講義ではこれ以上述べない).
- 無限次元空間では, $p \neq q$ であるとき, l_p と l_q は, ノルム空間としては同値にならない. 同様に, L_p と L_q も同値にならない.

内積 (1)

- V を複素数体上のベクトル空間とする. $V \times V$ で定義された複素関数 $g(\cdot,\cdot)$ が $u, v, w \in V$ および $\alpha \in \mathbb{C}$ をどのように取っても次の 5 条件を満たすとき, これを内積という [黒田].
  1) $g(v,u) = \overline{g(u,v)}$, 2) $g(\alpha u, v) = \alpha g(u,v)$, 3) $g(u+v,w) = g(u,w)+g(v,w)$, 4) $g(u,u) \geq 0$, 5) $g(u,u) = 0$ iff $u = 0$.

内積 (2)

- 内積が定義されたベクトル空間を内積空間または前 Hilbert 空間という.
- 以下では, $g(u, v)$ を単に (u, v) と書く.
- 内積は, 一般的な関数空間に直交性を導入するために必要となる.

内積 (3)

- iff は if and only if の略である.
- $\overline{\cdot}$ は複素共役をあらわす.
- V が実数体上のベクトル空間のときには, 第一の条件は $g(v, u) = g(u, v)$ に変わる.

内積 (4)

- l_2 における内積は, たとえば $u = (u_k)_{k\in\mathbb{N}}$, $v = (v_k)_{k\in\mathbb{N}}$ としたとき, $(u, v) = \sum_{k=1}^{\infty} u_k \overline{v_k}$ によって与えられる ($\overline{v_k}$ は v_k の複素共役)[黒田].
- L_2 における内積は, たとえば $(f, g) = \displaystyle\int f(t)\overline{g(t)}d\mu$ によって与えられる [黒田]. f や g が Riemann 積分可能のときには, 積分を Riemann 積分で置き換えてよい. ($\overline{g(t)}$ は $g(t)$ の複素共役)

Banach 空間と Hilbert 空間

- 完備なノルム空間を Banach 空間という.
- 完備な内積空間を Hilbert 空間という.
- Banach 空間と Hilbert 空間は関数空間の代表格. これ以外にもよく使われる関数空間があるが, この講義では述べない.

補間 (1)

- 補間は, 関数近似の一種.
- 関数 $f(x)$ が未知であるが, その n 個の点 $x_1, \ldots, x_n$ における値 $y_1, \ldots, y_n$ が知られているとき, $g(x_1) = y_1, \ldots, g(x_n) = y_n$ となるような関数 $g(x)$ を求めて $f(x)$ の代用とすることを補間または内挿といい, g を補間公式または内挿公式という (岩波数学入門辞典).

補間 (2)

- 補間とは, 関数 $f(x)$ が未知であるが, その n 個の点 $x_1, \ldots, x_n$ における値 $y_1, \ldots, y_n$ が知られているとき, $x_1, \ldots, x_n$ 以外の点における $f(x)$ の値を推測しようとすることをいう. 1 次元の補間問題において, $\min\{x_1, \ldots, x_n\} \leq x \leq \max\{x_1, \ldots, x_n\}$ となっているときに補間, そうでないときに補外と呼ぶ ([伊理]).

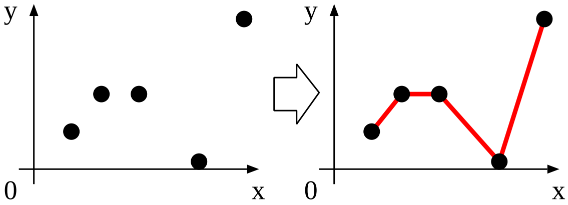
補間 (3)

- 補間とは関数近似の一種ではあるが, 与えられた n 個の点 (観測データと思ってもよい) $x_1, \ldots, x_n$ における値 $y_1, \ldots, y_n$ に対し, すべての (x_i, y_i) を通る曲線 (関数) を求めようとする点において, 単なる関数近似より狭義.
- ただし, 一般的な関数近似の意味で「補間」という言葉が使われる場合があるので要注意.

補間 (4)

- もっとも簡単な補間は線形補間．これは要するに「折れ線近似」である．
- Scilab で線形補間をおこなう関数は `interpln` この関数は補間および補外をおこなう．
- 線形補間 (Linear Interpolation) の例を次ページに示す．

補間 (5)



Linear Interpolation

補間 (6)

- 多項式による補間も一般的．多項式補間には何種類もやり方がある．この講義では Lagrange 補間の導入を述べる (詳細は次回).
- n 個の実数の組 $(x_1, y_1), \cdots, (x_n, y_n)$ が与えられ, $x_1, \cdots, x_n$ の中には同一の値のものはないと仮定する (関数の近似であれば, こうなっている筈である).

補間 (7)

- 2 点 (x_1, y_1), (x_2, y_2) を通る 1 次の多項式 (すなわち直線) は,

$$y_1 \frac{x - x_2}{x_1 - x_2} + y_2 \frac{x - x_1}{x_2 - x_1}$$

で与えられる.

補間 (8)

- 3 点 (x_i, y_i), $i = 1, 2, 3$ を通る 2 次の多項式は,

$$
y_1 \frac{(x - x_2)(x - x_3)}{(x_1 - x_2)(x_1 - x_3)} + y_2 \frac{(x - x_1)(x - x_3)}{(x_2 - x_1)(x_2 - x_3)} + y_3 \frac{(x - x_1)(x - x_2)}{(x_3 - x_1)(x_3 - x_2)}
$$

で与えられる.

補間 (9)

- n 点の場合も同様だが詳細は次回．上記のようにすればうまくいく理由については教科書 104 ページ.