CS 450: Numerical Anlaysis

Lecture 20

Chapter 7 Interpolation

Chebyshev Interpolation

Edgar Solomonik

Department of Computer Science
University of Illinois at Urbana-Champaign

April 4, 2018

Orthogonal Polynomials

- Recall that good conditioning for interpolation is achieved by constructing a well-conditioned Vandermonde matrix, which is the case when the columns (corresponding to each basis function) are orthonormal. To construct robust basis sets, we introduce a notion of *orthonormal functions*:

  To compute overlap between basis functions, use a w-weighted integral as inner product,

  $$\langle p, q \rangle_w = \int_{-\infty}^{\infty} p(t)q(t)w(t)dt$$

  orthonormality is then defined in the usual way, $\{\phi_i\}_{i=1}^n$ are orthonormal with respect to the above inner product if

  $$\langle \phi_i, \phi_j \rangle_w = \delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$$

  the corresponding norm is given by $||f|| = \sqrt{\langle f, f \rangle_w}$.

Legendre Polynomials

- The Gram-Schmidt orthogonalization procedure can be used to obtain an orthonormal basis with the same span as any given arbitrary basis:
  Given orthonormal functions $\{\hat{\phi}_i\}_{i=1}^{k-1}$ *obtain* k*th function from* ϕ_k *via*

$$\hat{\phi}_k(t) = \frac{\hat{\psi}_k(t)}{||\hat{\psi}_k||}, \quad \psi_k(t) = \phi_k(t) - \sum_{i=1}^{k-1} \langle \phi_k(t), \hat{\phi}_k(t) \rangle_w \hat{\phi}_k(t)$$

- The Legendre polynomials are obtained by Gram-Schmidt on the monomial basis, with normalization done so $\hat{\phi}_i(1) = 1$ and $w(t) = \begin{cases} 1 : -1 \leq t \leq 1 \\ 0 : \text{otherwise} \end{cases}$
  For example, $\{\hat{\phi}_i(t)\}_{i=1}^{3} = \{1, t, (3t^2 - 1)/2\}$ *since*

$$\psi_1(t) = 1, \quad \psi_2(t) = t - \frac{1}{2}\int_{-1}^{1} t dt = t$$

$$\psi_3(t) = t^2 - \frac{1}{2}\int_{-1}^{1} t^2 dt - t\int_{-1}^{1} t^3 dt = t^2 - 1/3$$

Basis Orthogonality and Conditioning

- To obtain perfectly conditioned Vandermonde system, want inner products of different columns to be zero:

$$\langle \boldsymbol{a}_i, \boldsymbol{a}_j \rangle = \sum_{k=1}^{n} \phi_i(t_k)\phi_j(t_k) = \delta_{ij}$$

- These inner products should be close to zero, if they are a suitable quadrature rule for our weighted functional inner product:

$$\sum_{k=1}^{n} \phi_i(t_k)\phi_j(t_k) \approx \int_{-\infty}^{\infty} w(t)\phi_i(t)\phi_j(t)dt$$

For example above holds as equality if we choose

$$w(t) = \begin{cases} \infty & : t = t_k \text{ for some } k \\ 0 & : \text{otherwise} \end{cases}$$

Chebyshev Basis

- *Chebyshev polynomials* $\phi_j(t) = \cos(j \arccos(t))$ and *Chebyshev nodes* $t_i = \cos\left(\frac{2i-1}{2n}\pi\right)$ provide a way to pick *nodes* $t_1, \dots, t_n$ along with a basis, to yield perfect conditioning:

  The Chebyshev basis functions are orthonormal with respect to

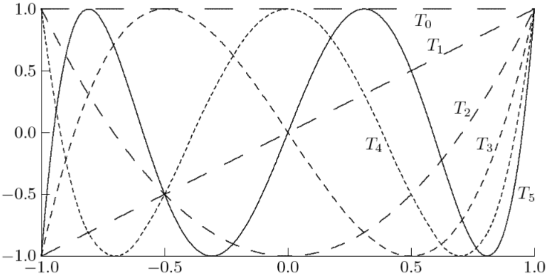
  $$w(t) = \begin{cases} 1/(1-t^2)^{1/2} & : -1 \leq t \leq 1 \\ 0 & : \textit{otherwise} \end{cases}.$$

  The Chebyshev nodes ensure orthogonality of the columns of $\boldsymbol{A}$, since

  $$\sum_{k=1}^{n} \phi_l(t_k)\phi_j(t_k) = \sum_{k=1}^{n} \cos\left(\frac{l(2k-1)}{2n}\pi\right)\cos\left(\frac{j(2k-1)}{2n}\pi\right)$$

  is zero whenever $j \neq l$ due to periodicity of the summands.

Chebyshev Nodes Intuition



- Note *equi-alteration* property, successive extrema of T_k have opposite sign:
- Set of k Chebyshev nodes of are given by zeros of T_k

Orthogonal Polynomials and Recurrences

- The Newton polynomials could be obtained by a two-term recurrence
- Legendre and Chebyshev polynomials also satisfy three-term recurrence, for Chebyshev

$$\phi_{i+1}(t) = 2t\phi_i(t) - \phi_{i-1}(t)$$

- In fact all orthogonal polynomials satisfy some recurrence of the form,

$$\phi_{i+1}(t) = (\alpha_k t + \beta_k)\phi_i(t) - \gamma_k \phi_{i-1}(t)$$

Error in Interpolation

Given degree n polynomial interpolant $\tilde{f}$ of f induction on n shows that $E(x) = f(x) - \tilde{f}(x)$ has n zeros $x_1, \ldots, x_n$ and there exist $y_1, \ldots, y_n$ such that

$$E(x) = \int_{x_1}^{x} \int_{y_1}^{w_0} \cdots \int_{y_n}^{w_{n-1}} f^{(n+1)}(w_n) dw_n \cdots dw_0 \tag{1}$$

$$E(x) = E(x_1) + \int_{x_1}^{x} E'(w_0) dw_0 \tag{2}$$

Now note that for each of $n-1$ consecutive pairs x_i, x_{i+1} we have

$$\int_{x_i}^{x_{i+1}} E'(x) dx = E(x_{i+1}) - E(x_i) = 0$$

and so there are $z_i \in (x_i, x_{i+1})$ such that $E'(z_i) = 0$. By inductive hypothesis

$$E'(w_0) = \int_{z_1}^{w_0} \int_{y_2}^{w_1} \cdots \int_{y_n}^{w_{n-1}} f^{(n+1)}(w_n) dw_n \cdots dw_1 \tag{3}$$

Substituting (3) *into* (2)*, we obtain* (1) *with $y_1 = z_1$.*

Interpolation Error Bounds

- The error bound implies that

$$|E(x)| \leq \frac{\max_{s\in[x_1,x_n]} |f^{(n+1)}(s)|}{n!} \prod_{i=1}^{n}(x - x_i) \quad \text{for} \quad x \in [x_1, x_n]$$

Note that the Choice of Chebyshev nodes decreases this error bound at the extrema, equalizing it with nodes that are in the middle of the interval.

- Letting $h = x_n - x_1$ (often also achieve same for h as the node-spacing $x_{i+1} - x_i$), we obtain

$$|E(x)| \leq \frac{\max_{s\in[x_1,x_n]} |f^{(n+1)}(s)|}{n!} h^n = O(h^n) \quad \text{for} \quad x \in [x_1, x_n]$$

Suggests that higher-accuracy can be achieved by

- *adding more nodes (however, high polynomial degree can lead to unwanted oscillations)*
- *shrinking interpolation interval (suggests piecewise interpolation)*

Piecewise Polynomial Interpolation

- The kth piece of the interpolant is a polynomial in $[x_i, x_{i+1}]$
  - *Typically low-degree polynomial pieces used, e.g. cubic.*
  - *Degree of piecewise polynomial is the degree of its pieces.*
  - *Continuity is automatic, differentiability can be enforced by ensuring derivative of pieces is equal at knots (nodes at which pieces meet).*
- *Hermite* interpolation ensures consecutive interpolant pieces have same derivative at each *knot* x_i:

  Hermite interpolation ensures overall interpolant is once differentiable. Various further constraints can be placed on the interpolant if its degree is at least 3, since otherwise the system is underdetermined.

Spline Interpolation

- A *spline* is a $(k-1)$-time differentiable piecewise polynomial of degree k:
  Cubic splines are different from Hermite cubics
  - $2(n-1)$ *equations needed to interpolate data*
  - $n-2$ *to ensure continuity of derivative*
  - $n-2$ *to ensure continuity of second derivative for cubic splines*

  Overall there are $4(n-1)$ *parameters in the interpolant.*
- The resulting interpolant coefficients are again determined by an appropriate generalized Vandermonde system:

  A natural spline obtains $4(n-1)$ *constraints by forcing* $f''(x_1) = f''(x_n) = 0$.
  Given pieces $p(x) = \sum_{i=0}^{3} \alpha_i x^i, q(x) = \sum_{i=0}^{3} \beta_i x^i$ *the generalized Vandermonde system looks like (3 similar constraints on* β *not shown)*

$$\begin{bmatrix} 1 & x_1 & x_1^2 & x_1^3 & 0 & 0 & 0 & 0 \\ 1 & x_2 & x_2^2 & x_2^3 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2x_2 & 3x_2^2 & 0 & -1 & -2x_2 & -3x_2^2 \\ 0 & 0 & 2 & 6x_2 & 0 & 0 & -2 & -6x_2 \\ 0 & 0 & 2 & 6x_1 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \alpha_0 & \cdots & \alpha_3 & \beta_0 & \cdots & \beta_3 \end{bmatrix}^T = \begin{bmatrix} y_1 \\ y_2 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

B-Splines

B-splines provide an effective way of constructing splines from a basis:

- The basis functions can be defined recursively with respect to degree.

$$v_i^k(t) = \frac{t - t_i}{t_{i+k} - t_i}, \quad \phi_i^0(t) = \begin{cases} 1 & t_i \leq t \leq t_{i+1} \\ 0 & \textit{otherwise} \end{cases}$$

$$\phi_i^k(t) = v_i^k(t)\phi_i^{k-1}(t) + (1 - v_{i+1}^k(t))\phi_{i+1}^{k-1}(t)$$

- The ith degree k polynomial piece is positive on $[t_i, t_{i+k+1}]$ and zero everywhere else
- All possible splines of degree k with notes $\{t_i\}_{i=1}^n$ can be represented in the basis.
- The resulting interpolant coefficients are again determined by an appropriate generalized Vandermonde system: