# CSE276C - Functional Interpolation and Approximation

Henrik I. Christensen

Computer Science and Engineering
University of California, San Diego
http://cri.ucsd.edu

October 2022

## Outline

## Introduction

- Last time we spoke about direct use of data point / simple models
- What if we want an explicit functional approximation to data?
- Approximating a function/data by a class of simpler functions
- Two main motivations
  1. Decomposition of a complicated function into constituent simpler functions to simplify further work
  2. Recover a function from partial or noisy information
- Applications:
  1. Signal compression / reconstruction (Fourier would be an example)
  2. Data fitting (line, plane, manifold, ...)
  3. Recovery of a model say CAD recovery



## Material

- Numerical Recipes: Chapter 3.4-3.5
- Numerical Renaissance: Chapter 5

## Outline





## Uniform approximation by polynomials

- Looking at polynomial again
- What is the best uniform approximation?
- Given a function f: $[a, b] \rightarrow R$ and a polynomial p we can measure the error by the $L_\infty$ norm, i.e.,

$$||f - p||_\infty = \max_{a<x<b} |f(x) - p(x)|$$

- A good approximation is one where the norm is small
- Remember Weierstrass' theorem.

## Polynomial approximation

- Lets restrict the degree of the polynomial - n
- Lets set $\pi_n$ be all the polynomials degree at most n
- Let uniform distance of f from $\pi_n$ be the smallest error achievable using polynomials from $\pi_n$ denoted by

$$d(f, \pi_n) = min_{p \in \pi_n} ||f - p||_\infty$$

- How can we make it happen?



## Polynomial approximation - getting help

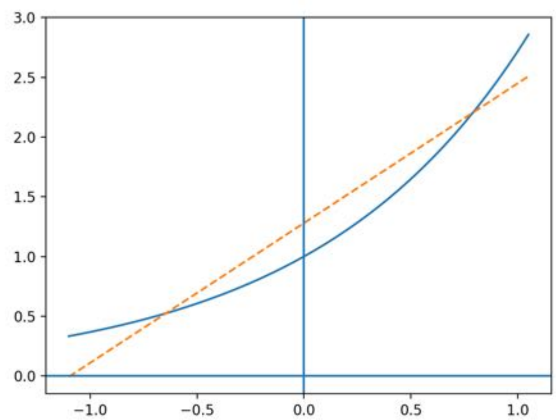
- We have a theorem:
  - A function f continuous in $[a, b]$ has exactly one best solution from $\pi_n$
  - The polynomial $p \in \pi_n$ of f across $[a, b]$ iff
  - there are n+2 point $a \leq x_0 \leq \ldots \leq x_n + 1 \leq b$ such that

$$(-1)^i[f(x_i) - p(x_i)] = \epsilon ||f - p||_\infty$$

  where $\epsilon = signum[f(x_0) - p(x_0)]$
- By alternating signs at n+2 points the different between f and p is precisely equal to the $L_\infty$

## Putting theorem to work

- Can we use the theorem to build a strategy?
- Lets consider $f(x) = e^x$ on $[-1, 1]$
- What would be the best 1st order approximation, i.e., $\pi_1$






## Fitting the line

- So we have three points
- $x_0 = -1$, $x_1 = ?$ and $x_2 = 1$
- at which the error is $f(x) = p(x)$
- So what is $x_1$?

## Fitting the line

- So we have three points
- $x_0 = -1$, $x_1 = ?$ and $x_2 = 1$
- at which the error is $f(x) = p(x)$
- So what is $x_1$?
- we can write $p(x) = a + bx$
- We can compute the error at the three points:

$$\begin{array}{llll} e(x_0) & = f(x_0) - p(x_0) & = f(-1) - p(-1) & = \frac{1}{e} - a + b \\ e(x_1) & = f(x_1) - p(x_1) & & = e^{x_1} - a + bx_1 \\ e(x_2) & = f(x_2) - p(x_2) & = f(1) - p(1) & = e - a - b \end{array}$$

- Given $e(x_0) = e(x_2)$

$$\begin{array}{ccc} \frac{1}{e} - a + b & = & e - a - b \\ 2b & = & e - \frac{1}{e} \\ b & = & 1.1752 \end{array}$$

The slope is equal to the average change



## Fitting the line (cont)

- How do we find a?
- The difference (positive / negative) should be symmetric
- The error function should at an extrema at $x_0, x_1, x_2$ but with alternate signs
- $e(x) = f(x) - p(x) = e^x - a - bx$ so
- $e'(x) = e^x - b \Rightarrow e^{x_1} - b = 0$
- $x_1 = ln\ b$
- $x_1 \approx 0.16144$

## Fitting the line (cont)

- How do we find a?
- The difference (positive / negative) should be symmetric
- The error function should at an extrema at $x_0, x_1, x_2$ but with alternate signs
- $e(x) = f(x) - p(x) = e^x - a - bx$ so
- $e'(x) = e^x - b \Rightarrow e^{x_1} - b = 0$
- $x_1 = ln\ b$
- $x_1 \approx 0.16144$
- $e(x_1) = -e(x_2) \Rightarrow e^{x_1} - a - bx_1 = -e + a + b$
- $a = \frac{e - bx_1}{2} \approx 1.2643$
- $p(x) \approx 1.2643 + 1.1752x$
- The maximum error would be $e(x_1) = ||f(x_1) - p(x_1)||_\infty \approx 0.2788$



## Approximation - Discussion

- Example showed a way to construct a solution.
- What if we did not know the appropriate n?
- If we make n too small there is a lack of fit
- If we make n too large the fit will be poor (too much wiggle)
- Could we estimate $d(f, \pi_n)$?
- Maybe not, but a lower bound might be possible

## Divided Differences

- Slight detour
- Divided differences are frequently used to compute coefficients in interpolation polynomials.
- Recursive formulation. Given a set of data points $(x_0, y_0), \ldots, (x_k, y_k)$

$$[y_v, \ldots, y_{v+j}] = \frac{[y_{v+1}, \ldots, y_{v+j}] - [y_v, \ldots, y_{v+j-1}]}{x_{v+j} - x_v}$$

  and

$$[y_v] = y_v \ \ v \in \{0, \ldots, k\}$$

- The recursive formulation is computationally effective
- The first few terms

$$\begin{aligned}
[y_0] &= y_0 \\
[y_0, y_1] &= \frac{y_1 - y_0}{x_1 - x_0} \\
[y_0, y_1, y_2] &= \frac{[y_1, y_2] - [y_0, y_1]}{x_2 - x_0} = \frac{\frac{y_2 - y_1}{x_2 - x_1} - \frac{y_1 - y_0}{x_1 - x_0}}{x_2 - x_0} \\
&= \frac{y_2 - y_1}{(x_2 - x_1)(x_2 - x_0)} - \frac{y_1 - y_0}{(x_1 - x_0)(x_2 - x_0)}
\end{aligned}$$



## Estimating a lower bound

- Assume we have a function $f : [a, b] \to R$
- We will use divided differences to compute bounds
- Lets assume we have three points $x_0, x_1, x_2$ as p is linear

$$p[x_0, x_1, x_2] = 0$$

  i.e. the gradient does not vary
- we can also write

$$f[x_0, x_1, x_2] = \frac{f(x_0)}{(x_0 - x_1)(x_0 - x_2)} + \frac{f(x_1)}{(x_1 - x_0)(x_1 - x_2)} + \frac{f(x_2)}{(x_2 - x_0)(x_2 - x_1)}$$

  so

## Estimating lower bound (cont.)

$$
\begin{aligned}
f[x_0, x_1, x_2] &= f[x_0, x_1, x_2] - p[x_0, x_1, x_2] \\
&= (f - p)[x_0, x_1, x_2] \\
&= \frac{f(x_0) - p(x_0)}{(x_0 - x_1)(x_0 - x_2)} + \frac{f(x_1) - p(x_1)}{(x_1 - x_0)(x_1 - x_2)} + \frac{f(x_2) - p(x_2)}{(x_2 - x_0)(x_2 - x_1)} \\
&= \frac{f(x_0) - p(x_0)}{w'(x_0)} + \frac{f(x_1) - p(x_1)}{w'(x_1)} + \frac{f(x_2) - p(x_2)}{w'(x_2)}
\end{aligned}
$$

where

$$w'(x) = (x - x_0)(x - x_1)(x - x_2)$$



## Estimating lower bound (cont.)

- We can then estimate a bound

$$|f[x_0, x_1, x_2]| \le ||f - p||_\infty \left( \frac{1}{|w'(x_0)|} + \frac{1}{|w'(x_1)|} + \frac{1}{|w'(x_2)|} \right)$$

or

$$||f - p||_\infty \ge \frac{|f[x_0, x_1, x_2]|}{\frac{1}{|w'(x_0)|} + \frac{1}{|w'(x_1)|} + \frac{1}{|w'(x_2)|}}$$

- the polynomial on left hand side is arbitrary so $d(f, \pi_n) = min_{p \in \pi_n} ||f - p||_\infty$
- right hand side is purely based on f and three points, so we can estimate the value

## Back to our example

- Lets use $f(x) = e^x$ in the interval $[-1, 1]$.
- Pick say -1, 0, 1 as our points

$$f[x_0, x_1, x_2] = \frac{1}{2} f(-1) - f(0) + \frac{1}{2} f(1)$$

and

$$\frac{1}{|w'(x_0)|} + \frac{1}{|w'(x_0)|} + \frac{1}{|w'(x_0)|} = \frac{1}{2} + 1 + \frac{1}{2} = 2$$

thus

$$d(f, \pi_1) \geq \frac{f(-1) - 2f(0) + f(1)}{4}$$

- the bound is then $d(f, \pi_1) = 0.2715$, which is not too far away from 0.2788 that was achieved.
- the lower bounds says that we cannot estimate $e^x$ much better than .3 in the interval -1,1 with a linear approximation, which is very valuable.



## Outline

## Chebyshev polynomials

- Chebyshev polynomials are sequences of polynomials that are defined recursively.
- The first kind of a Chebyshev polynomial is denoted $T_N(x)$ and given by

$$T_N(x) = \cos(n \arccos x)$$

looks trigonometric but can be used to general polynomials. I.e

$$\begin{aligned} T_0(x) &= 1 \\ T_1(x) &= x \\ T_2(x) &= 2x^2 - 1(\text{as } cos(2\theta) = 2cos^2(\theta) - 1) \\ T_3(x) &= 4x^3 - 3x \\ T_{N+1}(x) &= 2xT_N(x) - T_{N-1}(x), \text{ for } n \geq 1 \end{aligned}$$



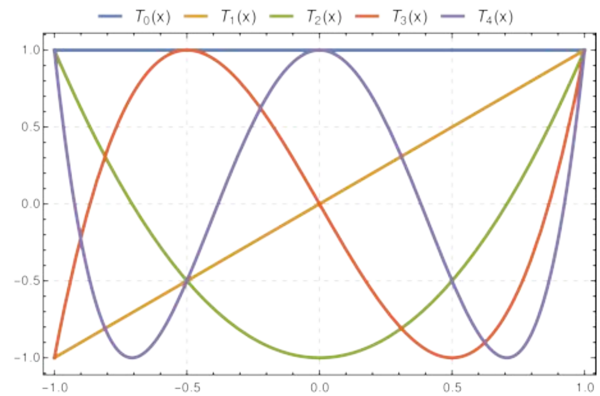
## Chebyshev Polynomials

- The polynomials are orthogonal over the interval $[-1, 1]$ over a weight of $(1 - x^2)^{-1/2}$ so that

$$\int_{-1}^{1} \frac{T_i(x)T_j(x)}{\sqrt{1 - x^2}} dx = \begin{cases} 0 & i \neq j \\ \frac{\pi}{2} & j = j \neq 0 \\ \pi & i = j = 0 \end{cases}$$

# Chebyshev Polynomials

- The polynomial $T_N(x)$ has N zeros in the internal $[-1, 1]$ at the points $x = \cos(\frac{\pi(k+\frac{1}{2})}{N})$ for $k \in 0, \ldots, N-1$
- There is a similar set of extrema at $x = \cos(\frac{\pi k}{N})$






# Chebyshev Approximation

- For periodic functions. f(x), over the interval $[-1, 1]$ an N coefficient approximation is

$$\begin{aligned} c_j &= \frac{2}{N} \sum_{k=0}^{N-1} f(x_k) T_J(x_k) \\ &= \frac{2}{N} \sum_{k=0}^{N-1} f\left(\cos\frac{\pi(k+\frac{1}{2})}{N}\right) \cos\frac{\pi(k+\frac{1}{2})}{N} \end{aligned}$$

- The approximation is then

$$f(x) \approx p(x) = \left[\sum_{k=1}^{N-1} c_k T_k(x)\right] - \frac{1}{2} c_0$$

- which is an exact match in terms of zero crossings
- the errors are uniformly distributed over $[-1, 1]$

# Warping coordinated

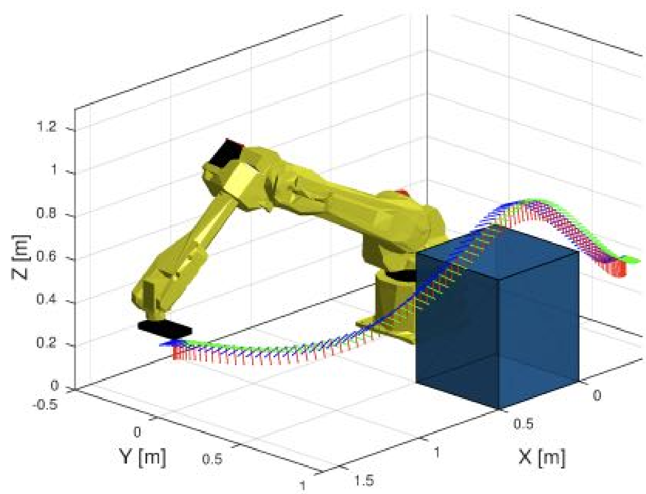
- If the domain is different from $[-1, 1]$ the variable can be changed from $[a, b]$

$$y = \frac{x - \frac{1}{2}(b-a)}{\frac{1}{2}(b-a)}$$

the approximated can be mapped forward / back as needed



# Example of using Checyshev Points for Control




(a) Test case 1

## Outline





## Truncated Power Series

- The uniform error of the Chebyshev functions/series implies that one can use a limited number of terms
- Say you have a series

$$f(x) = \frac{1}{2} - \frac{x}{4} + \frac{x^2}{8} - \frac{x^3}{16} + \ldots$$

- fitting a polynomial function and trying to achieve $\epsilon < 10^{-9}$ would require more than 30 terms
- If we use a Chebyshev approximation
  1. Compute enough terms to have $\epsilon < T$ across series
  2. Change variable to $[-1, 1]$
  3. Find Chebyshev series that satisfy error
  4. Truncate series using $c_k T_k(x)$ as an estimated error residential
  5. Convert back to polynomial form
  6. Convert back to original coordinate range
- For the example the reduction is from 30 to 9 terms

# Outline

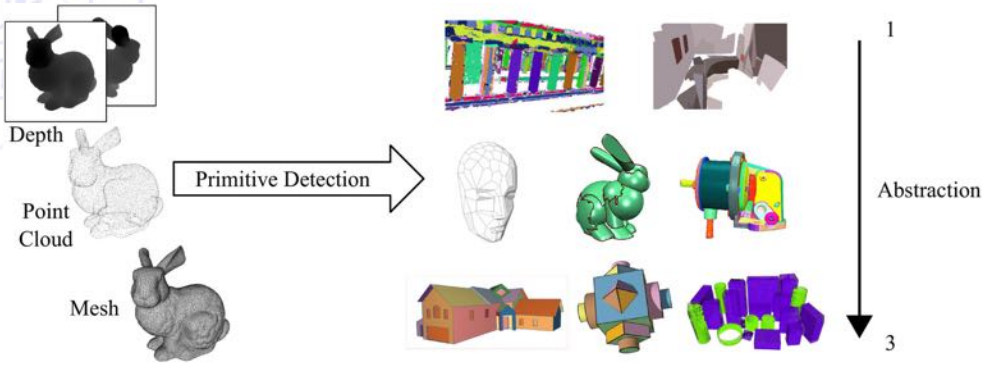




# Functional approximation and interpolation

- Frequently using a functional approximation is much more effective and it adds semantic information (a class) to the data approximation
- The are quite a few functional approximation forms
- Giving a few examples from polynomial, $\pi_n$, form to periodic function
- A key consideration is what domain knowledge is available to guide model selection

# Small example






# Questions

Questions