CS368 MATLAB Programming

Lecture 7

Young Wu

Based on lecture slides by Michael O'Neill and Beck Hasti

March 9, 2022

Coordination Game

Quiz

Schedule

Admin

Polynomials

Math

- A degree n polynomial, $c_n x^n + c_{n-1} x^{n-1} + c_{n-2} x^{n-2} + \ldots + c_2 x^2 + c_1 x + c_0$ is specified by a list of coefficients $c_n, c_{n-1}, \ldots, c_1, c_0$ usually stored in a vector.
- The computation can be done efficiently using Horner's method: $((((c_n x + c_{n-1})\, x + c_{n-2})\, x + \ldots + c_2)\, x + c_1)\, x + c_0.$

Exponential and Trigonometry Functions

Math

- Many elementary function can be evaluated and approximated using polynomials.

1. $e^x \approx 1 + x + \frac{x^2}{2} + \frac{x^3}{6} + \frac{x^4}{24} + \frac{x^5}{120} + \dots$
2. $\sin(x) \approx x - \frac{x^3}{6} + \frac{x^5}{120} + \dots$
3. $\cos(x) \approx 1 - \frac{x^2}{2} + \frac{x^4}{24} + \dots$

- They are also called Taylor polynomials.

Polynomial Evaluation

Code

- Suppose c and x are row vectors with length n.
- *dot(c, x .^ ((n − 1):−1:0))* evaluates the polynomial with coefficients c at x.
- *polyval(c, x)* evaluates the same polynomial but faster.

Polynomial Evaluation, Trig

Quiz

Lagrange Polynomial

Math

- A set of n points (with distinct x values) uniquely determine a degree $n-1$ polynomial that passes through all the points.
- The coefficients of the leading terms can be zero, so technically the degree of the polynomial can be less than $n-1$.

Lagrange Polynomial Diagram

Math

Vandermonde Matrix

Math

- The coefficients can be found by solving a system of linear equations, $\begin{bmatrix} x_1^{n-1} & x_1^{n-2} & \dots & 1 \\ x_2^{n-1} & x_2^{n-2} & \dots & 1 \\ \dots & \dots & \dots & \dots \\ x_n^{n-1} & x_n^{n-2} & \dots & 1 \end{bmatrix} \begin{bmatrix} c_{n-1} \\ c_{n-2} \\ \dots \\ c_0 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \dots \\ y_n \end{bmatrix}$.

Polyfit

Code

- Suppose x, y are two row vectors with length n.
- *repmat(x', 1, n) .^ ((n − 1):−1:0)* is the Vandermonde matrix, and *vander(x)* creates the same matrix.
- *vander(x) \ y'* solves for the coefficients using the Vandermonde matrix.
- *polyfit (x, y, n − 1)* finds the same coefficients.

Spline

Math

- A set of n points (with distinct x values, and ordered according to x) can define $n-1$ cubic polynomial segments satisfying the following conditions.

1. The i-th polynomial segment connects point i and point $i+1$.
2. The segments form a smooth curve. Technically, C^2 smoothness is desired: the first and second derivatives should be continuous.

Spline Diagram

Math

Tridiagonal Matrix

Math

- With n points, there are $4n-4$ coefficients to compute. Suppose the coefficient for segment i connecting (x_i, y_i) and (x_{i+1}, y_{i+1}) is $c_{i1}, c_{i2}, c_{i3}, c_{i4}$.

1. $c_{i1}x_i^3 + c_{i2}x_i^2 + c_{i3}x_i + c_{i4} = y_i$ for $i = 1, 2, \ldots, n-1$ so that the segment passes through the left end point.
2. $c_{i1}x_{i+1}^3 + c_{i2}x_{i+1}^2 + c_{i3}x_{i+1} + c_{i4} = y_{i+1}$ for $i = 1, 2, \ldots, n-1$ so that the segment passes through the right end point.
3. $3c_{i,1}x_{i+1}^2 + 2c_{i2}x_{i+1} + c_{i3} = 3c_{i+1,1}x_{i+1}^2 + 2c_{i+1,2}x_{i+1} + c_{i+1,3}$ for $i = 1, 2, \ldots, n-2$ so that the first derivative is continuous.
4. $6c_{i1}x_{i+1} + 2c_{i2} = 6c_{i+1,1}x_{i+1} + 2c_{i+1,1}$ for $i = 1, 2, \ldots, n-2$ so that the second derivative is continuous.

- There are $4n-6$ equations. 2 more equations are needed.

End Point Conditions

Math

- There are many ways to specify the 2 extra equations.

1. If the third derivative for the first and last two segments are continuous, $c_{1,1} = c_{2,1}$ and $c_{n-1,1} = c\left(n-2,1\right)$, the resulting spline is called a not-a-knot spline.
2. If the second derivative at the end points are 0, or $6c_{1,1}x_1 + 2c_{1,2} = 0$ and $6c_{n-1,1}x_n + 2c_{n-1,2} = 0$, the resulting spline is called a natural spline.

Spline

Code

- Suppose x, y are two row vectors with length n.
- *spline (x, y). coefs* finds the coefficients of the cubic spline that interpolates (x, y) while satisfying the not-a-knot condition.
- *spline (x, y, x0)* evaluates the cubic spline interpolating (x, y) at $x = x_0$.
- *csape(x, y, ' variational '). coefs* finds the coefficients of the natural cubic spline. It requires MATLAB's Curve Fitting Toolbox.

Local Coefficient

Code

- *spline(x, y).coefs* are local coefficients meaning the equation for segment i and $x \in [x_i, x_{i+1}]$ is $c_{i1}(x - x_i)^3 + c_{i2}(x - x_i)^2 + c_{i3}(x - x_i) + c_{i4}$.
- The polynomial can be expanded to find the coefficients of the original spline problem.

Interpolation, Polyfit

Quiz

Interpolation, Polyfit Too

Quiz

Interpolation, Spline

Quiz

Interpolation, Spline Too

Quiz

Polynomial Regression

Math

- *polyfit (x, y, d)* with $d < n$, finds the polynomial that is the closest to the points (x, y).
- The total distance from the points to the polynomial is defined as the sum of the squared differences between the y value at x and the value of the polynomial at x,

$$
\begin{aligned}
D &= \sum_{i=1}^{n} \left(y_i - \sum_{j=0}^{d} c_i x_i^j \right)^2 \\
&= \sum_{i=1}^{n} \left(y_i - c_d x_i^d - c_{d-1} x_i^{d-1} - \ldots - c_1 x_i - c_0 \right)^2 .
\end{aligned}
$$

- *polyfit* finds the coefficients of the polynomial that minimizes D. These coefficients are also called regression coefficients.

Regression Diagram

Math

General Linear Regression

Math

- *regress (y', X)* finds the regression coefficients for $c_n x_n + c_{n-1} x_{n-1} + \ldots + c_1 x_1 + c_0 x_0$, where $x_0, x_1, \ldots, x_n$ are the columns of X. It requires MATLAB's Statistics and Machine Learning Toolbox.
- *regress (y', X)* solves $\left(X^T X\right) c = X^T y$ instead of $Xc = y$. This is because $X^T X$ is always invertible (meaning $\left(X^T X\right)^{-1}$ always exists), so $c = \left(X^T X\right)^{-1} X^T y$.
- When X is the degree d Vandermonde matrix for x, meaning $x_i = x^i$ for $i = d, d-1, \ldots, 0$, *regress (y, X)* finds the same coefficients as *polyfit (x, y, d)*.

Non-Polynomial Regression

Code

- *polyfit (f(x), y, 1)* finds the regression coefficients for $c_1 f(x) + c_0$, for example, *polyfit (sin (x), y, 1)* finds c_1, c_0 such that the total distance from the points to the function $y = c_1 \sin(x) + c_0$ is minimized.
- *regress (y', [f1(x') f2(x') ...])* finds the regression coefficients for $c_1 f_1(x) + c_2 f_2(x) + \ldots$, for example, *regress (y', [sin (x') cos(x') ones(length (x), 1)])* finds c_2, c_1, c_0 such that the total distance from the points to the function $y = c_2 \sin(x) + c_1 \cos(x) + c_0$ is minimized.

Regression, Polynomial

Quiz

Regression, Unknown Degree

Quiz

Regression, Non-Polynomial

Quiz

Blank Slide