Math 636 - Mathematical Modeling

Linear and Polynomial Modeling

Joseph M. Mahaffy,
⟨jmahaffy@mail.sdsu.edu⟩

Department of Mathematics and Statistics
Dynamical Systems Group
Computational Sciences Research Center
San Diego State University
San Diego, CA 92182-7720

http://jmahaffy.sdsu.edu

Fall 2017

Outline

Snowy Tree Cricket

Snowy Tree Cricket (*Oecanthulus niveus*)

Chirping Crickets and Temperature

Simplest Mathematical Model is the **Linear Model**

- Folk method for finding temperature (Fahrenheit)
  Count the number of chirps in a minute and divide by 4, then add 40
- In 1898, A. E. Dolbear [1] noted that
  "crickets in a field [chirp] synchronously, keeping time as if led by the wand of a conductor"
- This gives the **Linear Model**

$$T = \frac{N}{4} + 40$$

[1] A. E. Dolbear, The cricket as a thermometer, American Naturalist (1897) 31, 970-971

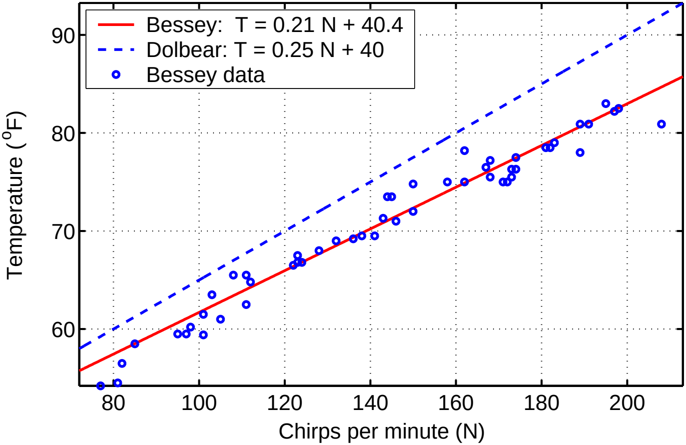
Data Fitting Linear Model

- Mathematical models for chirping of snowy tree crickets (*Oecanthulus fultoni*) are **Linear Models**
- Data from C. A. Bessey and E. A. Bessey [2] (8 crickets) from Lincoln, Nebraska during August and September, 1897 (shown on next slide)
- The ***least squares best fit line*** to the data is

$$T = 0.215476\,N + 39.7441$$

[2] C. A. Bessey and E. A. Bessey, Further notes on thermometer crickets, American Naturalist (1898) 32, 263-264

Bessey Data and Linear Models



Cricket Equation as a Linear Model

The line creates a mathematical model

- The **temperature,** T as a **function** of the rate snowy tree crickets chirp, **Chirp Rate,** N

There are several Biological and Mathematical questions about this ***Linear Cricket Model***

There is a complex relationship between the biology of the problem and the mathematical model

Biological Questions – Cricket Model 1

How well does the line fitting the Bessey & Bessey data agree with the Dolbear model given above?

- Graph shows Linear model fits the data well
- Data predominantly below **Folk/Dolbear** model
- Possible discrepancies
  - Different cricket species
  - Regional variation
  - Folk only an approximation
- Graph shows only a few °F difference between models

Biological Questions – Cricket Model

When can this model be applied from a practical perspective?

- Biological thermometer has limited use
- Snowy tree crickets only chirp for a couple months of the year and mostly at night
- Temperature needs to be above 50°F
- Not very accurate

Does this model give any insight into what is happening biologically?

Mathematical Questions – Cricket Model 1

Over what range of temperatures is this model valid?

- Biologically, observations are mostly between 50°F and 85°F
- Thus, limited **range** of temperatures, so limited **range** on the **Linear Model**
- **Range** of **Linear functions** affects its **Domain**
- From the graph, **Domain** is approximately 50–200 **Chirps/min**

Mathematical Questions – Cricket Model 2

How accurate is the model and how might the accuracy be improved?

- Data closely surrounds **Bessey Model** – No more than about 3°F away fom line
- **Dolbear Model** is fairly close though not as accurate – Sufficient for rapid temperature estimate (casual)
- Observe that the temperature data trends lower at higher chirp rates – compared against linear model
- Better fit with **Quadratic function** – Is this really significant?

Linear Least Squares 1

Consider a set of $n+1$ data points:

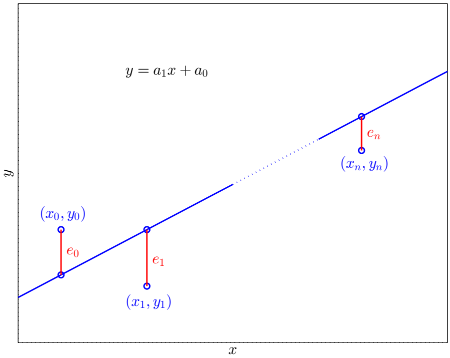
$$(x_0, y_0), (x_1, y_1), ..., (x_n, y_n).$$

If these points appear to form a line, then assume a ***linear model*** of the form

$$y(x) = a_1 x + a_0.$$

- Must find a slope, a_1, and an intercept, a_0
- The ***linear model*** in some sense best fits the data

Linear Least Squares



The ***least squares best fit*** minimizes the square of the error in the distance between the y_i values of the data points and the y value of the line

$$y(x) = a_1x + a_0.$$

Linear Least Squares

The ***error*** between the data points and the line is

$$e_i = y_i - y(x_i) = y_i - (a_1 x_i + a_0), \quad i = 0, ..., n,$$

which depends on a_0 and a_1.

The ***absolute error*** between the data points and the line satisfies:

$$|e_i| = |y_i - y(x_i)| = |y_i - (a_1 x_i + a_0)|, \quad i = 0, ..., n.$$

The ***sums of square errors*** function depends on the slope a_1 and intercept a_0 of the ***linear model***:

$$E(a_0, a_1) = \sum_{i=0}^{n} e_i^2 = \sum_{i=0}^{n} \left(y_i - (a_1 x_i + a_0)\right)^2$$

The ***Least Squares Best Fit Line*** is the ***minimum*** of the function $E(a_0, a_1)$

Linear Least Squares

Minimizing $E(a_0, a_1)$ is a classic problem in multivariable Calculus

The best fitting values of a_1 and a_0 are found in most elementary statistics texts

Define the ***mean*** of the x values:

$$\bar{x} = \frac{x_0 + x_1 + ... + x_n}{n+1} = \frac{1}{n+1}\sum_{i=0}^{n} x_i$$

The formulas for a_1 and a_0, assuming data points $(x_i, y_i), i = 0, ..., n$, and a ***linear model***, $y = a_1 x_i + a_0$, are the slope

$$a_1 = \frac{\sum_{i=0}^{n}(x_i - \bar{x})y_i}{\sum_{i=0}^{n}(x_i - \bar{x})^2}$$

and the intercept

$$a_0 = \frac{1}{n+1}\sum_{i=0}^{n} y_i - a_1\bar{x} = \bar{y} - a_1\bar{x}.$$

Linear Least Squares

Computer Software Packages

- Virtually all software packages have this formula
- In **Excel**, if a data set is entered into a spreadsheet, then graphing the data allows application of its ***Trendline*** package to obtain the ***linear least squares model***
- **MatLab** has the program `polyfit`, which can readily find the best fitting slope and intercept to a set of data
  - Data is stored as $x = [x_0, ..., x_n]^T$ and $y = [y_0, ..., y_n]^T$
  - The best fitting coefficients are found with the command `a = polyfit(x,y,1)`

Linear Least Squares

Minimization Problem (Multivariable Calculus):

Find the ***Least Squares best fit*** to the ***linear model***,

$$p_1(x) = a_0 + a_1 x$$

Minimize the error function:

$$E(a_0, a_1) = \sum_{i=0}^{n} \left[(a_0 + a_1 x_i) - y_i \right]^2 ,$$

so the first partial derivatives with respect to a_0 and a_1 are ***zero*** at the ***minimum***:

$$\begin{aligned} \frac{\partial}{\partial a_0} E(a_0, a_1) &= 2 \sum_{i=0}^{n} \left[(a_0 + a_1 x_i) - y_i \right] &= 0 \\ \frac{\partial}{\partial a_1} E(a_0, a_1) &= 2 \sum_{i=0}^{n} x_i \left[(a_0 + a_1 x_i) - y_i \right] &= 0. \end{aligned}$$

Linear Least Squares

The partial derivatives are rearranged to give the ***normal equations***

$$\begin{aligned}
\sum_{i=0}^{n} a_0 + \sum_{i=0}^{n} a_1 x_i &= \sum_{i=0}^{n} y_i \\
\sum_{i=0}^{n} x_i a_0 + \sum_{i=0}^{n} x_i a_1 x_i &= \sum_{i=0}^{n} x_i y_i.
\end{aligned}$$

The only unknowns in these ***normal equations*** are a_0 and a_1.

The ***normal equations*** form the 2×2 system of equations:

$$\begin{pmatrix} (n+1) & \displaystyle\sum_{i=0}^{n} x_i \\ \displaystyle\sum_{i=0}^{n} x_i & \displaystyle\sum_{i=0}^{n} x_i^2 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \end{pmatrix} = \begin{pmatrix} \displaystyle\sum_{i=0}^{n} y_i \\ \displaystyle\sum_{i=0}^{n} x_i y_i \end{pmatrix},$$

which is easily solved.

Actual and Absolute Error

- Error analysis is important for testing validity of a model
- Let X_e be an experimental measurement or the ***worst value*** from a model being tested
- Let X_t be a theoretical value or the ***best value*** from actual data
- The **Actual Error** is

$$\textbf{Actual Error} = X_e - X_t$$

- The **Absolute Error** is appropriate when only the magnitude of the error is needed

$$\textbf{Absolute Error} = |X_e - X_t|$$

Relative and Percent Error

- Relative and Percent error allow a better comparison of the error between data sets or within a data set with large differences in the numerical values
- Again let X_e be an experimental measurement or the ***worst value*** from a model being tested and X_t be a theoretical value or the ***best value*** from actual data
- The **Relative Error** is

$$\textbf{Relative Error} = \frac{X_e - X_t}{X_t}$$

- The **Percent Error** is the most common and divides the **Relative error** by the best expected value

$$\textbf{Percent Error} = \frac{X_e - X_t}{X_t} \times 100\%$$

Polynomial Discrete Least Squares

The m^{th} degree polynomial, $p_m(x)$, evaluated at the data points x_i:

$$a_0 + a_1 x_i + a_2 x_i^2 + \cdots + a_m x_i^m = y_i, \quad i = 0, \dots, n$$

is the product of an $(n+1) \times (m+1)$ matrix, A and the $(m+1) \times 1$ vector $\mathbf{a}$ and the result is the $(n+1) \times 1$ vector $\mathbf{y}$, where usually $n \gg m$:

$$\underbrace{\begin{pmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^m \\ 1 & x_1 & x_1^2 & \cdots & x_1^m \\ 1 & x_2 & x_2^2 & \cdots & x_2^m \\ 1 & x_3 & x_3^2 & \cdots & x_3^m \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^m \end{pmatrix}}_{A} \underbrace{\begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_m \end{pmatrix}}_{\mathbf{a}} = \underbrace{\begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{pmatrix}}_{\mathbf{y}}.$$

Polynomial Discrete Least Squares

The ***linear system*** below is not immediately solvable:

$$A\mathbf{a} = \mathbf{y},$$

as A is a rectangular matrix $(n+1)\times(m+1)$, $m \neq n$.

We generate a solvable system by multiplying both the left- and right-hand-side by A^T, *i.e.*,

$$A^T A\mathbf{a} = A^T\mathbf{y}$$

The matrix A^TA is a square $(m+1)\times(m+1)$ matrix, and $A^T\mathbf{y}$ an $(m+1)\times 1$ vector, which is a solvable ***linear system***.

NOTE: This is a solvable ***linear system*** because the coefficients of a polynomial appear linearly. Other ***nonlinear methods*** are needed for models with ***nonlinear parameters***.

Polynomial Discrete Least Squares

A closer look at A^TA,

$$\begin{bmatrix} 1 & 1 & 1 & 1 & \dots & 1 \\ x_0 & x_1 & x_2 & x_3 & \dots & x_n \\ x_0^2 & x_1^2 & x_2^2 & x_3^2 & \dots & x_n^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_0^m & x_1^m & x_2^m & x_3^m & \dots & x_n^m \end{bmatrix} \begin{bmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^m \\ 1 & x_1 & x_1^2 & \cdots & x_1^m \\ 1 & x_2 & x_2^2 & \cdots & x_2^m \\ 1 & x_3 & x_3^2 & \cdots & x_3^m \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^m \end{bmatrix}$$

$$= \begin{bmatrix} n+1 & \sum_{i=0}^n x_i^1 & \cdots & \sum_{i=0}^n x_i^m \\ \sum_{i=0}^n x_i^1 & \sum_{i=0}^n x_i^2 & \cdots & \sum_{i=0}^n x_i^{m+1} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{i=0}^n x_i^m & \sum_{i=0}^n x_i^{m+1} & \cdots & \sum_{i=0}^n x_i^{2m} \end{bmatrix}.$$

and $A^T\mathbf{y}$,

$$\begin{bmatrix} 1 & 1 & 1 & 1 & \dots & 1 \\ x_0 & x_1 & x_2 & x_3 & \dots & x_n \\ x_0^2 & x_1^2 & x_2^2 & x_3^2 & \dots & x_n^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_0^m & x_1^m & x_2^m & x_3^m & \dots & x_n^m \end{bmatrix} \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ \vdots \end{bmatrix} = \begin{bmatrix} \sum_{i=0}^n y_i \\ \sum_{i=0}^n x_i y_i \\ \sum_{i=0}^n x_i^2 y_i \\ \vdots \\ \sum_{i=0}^n x_i^m y_i. \end{bmatrix}$$

Polynomial Discrete Least Squares

From the previous slide, we have recovered the ***Normal Equations***:

$$A^T A\mathbf{a} = A^T \mathbf{y},$$

which is a solvable $(m+1)$ system of ***linear equations***.

Thus, given the data set $\mathbf{x} = [x_0, x_1, \dots, x_n]^T$ and $\mathbf{y} = [y_0, y_1, \dots, y_n]^T$, the best polynomial fit is readily found for any specified polynomial degree.

Let $\mathbf{x}^j$ be the vector $[x_0^j, x_1^j, \dots, x_n^j]^T$. To compute the best fitting polynomial of degree 3, $p_3(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3$, define:

$$A = \begin{bmatrix} | & | & | & | \\ \tilde{\mathbf{1}} & \mathbf{x} & \mathbf{x}^2 & \mathbf{x}^3 \\ | & | & | & | \end{bmatrix}, \quad \text{and compute} \quad \mathbf{a} = (A^T A)^{-1}(A^T \mathbf{y}).$$

This direct computation is not necessarily the most efficient.

Return to Cricket Thermometer

C. A. Bessey and E. A. Bessey collected data on eight different crickets that they observed in Lincoln, Nebraska during August and September, 1897. The number of chirps/min was N and the temperature was T.

Create matrices

$$A_1 = \begin{pmatrix} 1 & N_1 \\ 1 & N_2 \\ \vdots & \vdots \end{pmatrix} \qquad A_2 = \begin{pmatrix} 1 & N_1 & N_1^2 \\ 1 & N_2 & N_2^2 \\ \vdots & \vdots & \vdots \end{pmatrix}$$

$$A_3 = \begin{pmatrix} 1 & N_1 & N_1^2 & N_1^3 \\ 1 & N_2 & N_2^2 & N_2^3 \\ \vdots & \vdots & \vdots & \vdots \end{pmatrix} \qquad A_4 = \begin{pmatrix} 1 & N_1 & N_1^2 & N_1^3 & N_1^4 \\ 1 & N_2 & N_2^2 & N_2^3 & N_2^4 \\ \vdots & \vdots & \vdots & \vdots & \end{pmatrix}$$

Cricket A_n Matrices

How do we efficiently create the A_n matrices from the previous slide?

The data for the number of chirps/min stored as a vector,

$$N = [N_1, N_2, ..., N_m]^T,$$

so we use the MatLab function below with $x = N$ and n entered as the degree of the polynomial fit desired

```
function A = vanA(x,n)
%Least Squares Matrix for x and n poly
A = [ones(length(x),1)];
for i = 1:n
    A = [A,x.^i];
end
end
```

The output forms the matrices on the previous slide

Cricket Linear Model

As noted before, the best ***linear model***, $T(N) = a_1N + a_0$, is found by solving the ***linear system***:

$$A_1^T A_1 \mathbf{a} = A_1^T \mathbf{T}$$

Courses in **Numerical Linear Algebra** give the best way to solve this system.

MatLab efficiently computes this with the backslash operation:

$$\mathtt{A}_1 \backslash \mathtt{T}$$

and gives the parameters for best ***linear model***

$$T_1(N) = 0.215476\, N + 39.7441.$$

Cricket Polynomial Thermometer

We can find any best ***polynomial model***, $T(N) = a_n N^n + a_{n-1} N^{n-1} + ... + a_0$, by solving the ***linear system***:

$$A_n^T A_n \mathbf{a} = A_n^T \mathbf{T}$$

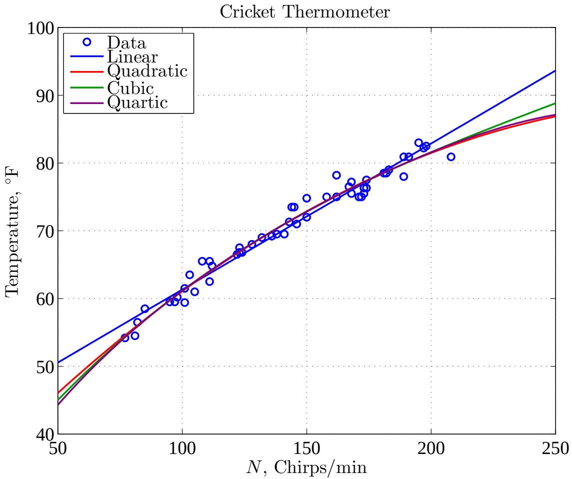
For the best **quadratic**, **cubic**, and **quartic models**, we use our program `vanA` to find our matrices A_2, A_3, and A_4, then apply the backslash operation in **MatLab** to efficiently compute the best ***polynomial coefficients***:

$$\texttt{A}_2\backslash\texttt{T} \qquad \texttt{A}_3\backslash\texttt{T} \qquad \texttt{A}_4\backslash\texttt{T}$$

This gives the best ***polynomial models***

$$\begin{aligned} T_2(N) &= -0.00064076\,N^2 + 0.39625\,N + 27.8489, \\ T_3(N) &= 0.0000018977\,N^3 - 0.001445\,N^2 + 0.50540\,N + 23.138, \\ T_4(N) &= -0.00000001765\,N^4 + 0.00001190\,N^3 - 0.003504\,N^2 \\ &= +0.6876\,N + 17.314. \end{aligned}$$

Graphs of Cricket Thermometer



Graph of the Bessey brother data and the best fitting polynomials of order 1, 2, 3, and 4.

Best Cricket Model

So how does one select the best model?

Visually, one can see that the linear model does a very good job, and one only obtains a slight improvement with a quadratic. Is it worth the added complication for the slight improvement.

It is clear that the sum of square errors (SSE) will improve as the number of parameters increase.

From statistics, it is hotly debated how much penalty one should pay for adding parameters.

Best Cricket Model - Analysis

Bayesian Information Criterion

Let n be the number of data points, SSE be the sum of square errors, and let k be the number of parameters in the model.

$$BIC = n\ln(SSE/n) + k\ln(n).$$

Akaike Information Criterion

$$AIC = 2k + n(\ln(2\pi SSE/n) + 1).$$

Best Cricket Model - Analysis Continued

The table below shows the by the Akaike information criterion that we should take a quadratic model, while using a Bayesian Information Criterion we should use a cubic model.

	Linear	Quadratic	Cubic	Quartic
SSE	108.8	79.08	78.74	**78.70**
BIC	46.3	33.65	**33.43**	37.35
AIC	189.97	**175.37**	177.14	179.12

Returning to the original statement, we do fairly well by using the folk formula, despite the rest of this analysis!