Math 484: Nonlinear Programming[1] Mikhail Lavrov

Chapter 4, Lecture 1: Interpolation and Best-Fit Lines

February 20, 2019 *University of Illinois at Urbana-Champaign*

1 Polynomial interpolation

Suppose we are given points $(x_1, y_1), (x_2, y_2), \dots, (x_k, y_k)$, and we want a polynomial f that passes through all the points: $f(x_i) = y_i$ for $1 \le i \le k$. What can we do?

One way to think of this problem is as a system of linear equations. If the polynomial $f(x)$ is written as $a_n x^n + a_{n-1}x^{n-1} + \dots + a_1 x + a_0$, then we have k constraints on the coefficients:

$$\begin{aligned} a_n x_1^n + a_{n-1} x_1^{n-1} + \dots + a_0 &= y_1 \\ a_n x_2^n + a_{n-1} x_2^{n-1} + \dots + a_0 &= y_2 \\ &\dots \\ a_n x_k^n + a_{n-1} x_k^{n-1} + \dots + a_0 &= y_k. \end{aligned}$$

We can rewrite this in matrix form as

$$\begin{bmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^n \\ 1 & x_2 & x_2^2 & \cdots & x_2^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_k & x_k^2 & \cdots & x_k^n \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_k \end{bmatrix}.$$

This strongly suggests that we should take $n = k - 1$. We might be worried, though, that occasionally this matrix is singular for $n = k - 1$, in which case we would have infinitely many polynomials for some values of $y_1, y_2, \dots, y_k$ and no polynomials for other values.

This doesn't happen, provided all the x-values are different.

Theorem 1.1. *Assuming $x_i \ne x_j$ for all i, j between 1 and k, there is a unique polynomial of degree at most $k - 1$ that passes through the points $(x_1, y_1), (x_2, y_2), \dots, (x_k, y_k)$.*

Proof. Write the matrix equation above as $M\mathbf{a} = \mathbf{y}$, where M is the matrix of powers of $x_1, \dots, x_k$, $\mathbf{a}$ is the vector of coefficients we want to find, and $\mathbf{y}$ is the vector of y-values.

First, consider the case where $\mathbf{y} = \mathbf{0}$. Then we are asking for a polynomial of degree $k - 1$ which is 0 at the points $x_1, x_2, \dots, x_k$. But a nonzero polynomial of degree $k - 1$ can have at most $k - 1$ roots, so only the zero polynomial can satisfy this condition.

This means that when the system of equations above is homogeneous (we are solving $M\mathbf{a} = \mathbf{0}$) there is a unique solution $\mathbf{a} = \mathbf{0}$. By the usual theory of linear equations, this means that there can

[1] This document comes from the Math 484 course webpage: https://faculty.math.illinois.edu/~mlavrov/courses/484-spring-2019.html

always be at most one solution to the general case: if we had two solutions $\mathbf{a}^{(1)} \neq \mathbf{a}^{(2)}$ to the system $M\mathbf{a} = \mathbf{y}$, then we'd have $M(\mathbf{a}^{(1)} - \mathbf{a}^{(2)}) = \mathbf{0}$, obtaining a nontrivial solution to the homogeneous equation.

At this point, we can finish the proof with an argument by dimensions: since the null space of M (the set of solutions to $M\mathbf{a} = \mathbf{0}$) has dimension 0 and the matrix is $k \times k$, the column space of M (the set of vectors $\mathbf{y}$ for which $M\mathbf{a} = \mathbf{y}$ has a solution) has dimension k, which means it must be all of $\mathbb{R}^k$. So there is a unique solution $\mathbf{a}$ for any choice of $\mathbf{y}$.

But we also have a more explicit construction for the polynomial that works, called the Lagrange interpolation formula. We will first construct k polynomials $\ell_1, \ell_2, \dots, \ell_k$, each of degree $k-1$, such that $\ell_i(x_i) = 1$ and $\ell_i(x_j) = 0$ when $i \neq j$. (For example, $\ell_1(x_1) = 1$ and $\ell_1(x_2) = \cdots = \ell_1(x_k) = 0$.)

Setting $\ell_1(x) = (x - x_2)(x - x_3) \cdots (x - x_k)$ would almost work: it is 0 in the right places, and has the right degree, but it has the wrong value when $x = x_1$. So we fix this by making

$$\ell_1(x) = \frac{(x - x_2)(x - x_3) \cdots (x - x_k)}{(x_1 - x_2)(x_1 - x_3) \cdots (x_1 - x_k)}$$

and in general we set $\ell_i(x) = \displaystyle\prod_{j \neq i} \frac{x - x_j}{x_i - x_j}$.

To finish the construction, let $f(x) = y_1 \cdot \ell_1(x) + y_2 \cdot \ell_2(x) + \cdots + y_k \cdot \ell_k(x)$. When we evaluate $f(x_i)$, the i^{th} term becomes $y_i \cdot \ell_i(x_i) = y_i$, and all other terms vanish. So $f(x_i) = y_i$, as desired, for each i. $\square$

So we have at least two approaches to find a polynomial that works: we can set up the system $M\mathbf{a} = \mathbf{y}$ and solve it, or we can use the Lagrange interpolation formula and simplify. Sometimes one approach works better than the other: for example, if all we want to do is evaluate the polynomial at some other point x_{k+1}, the Lagrange interpolation formula will probably be easier to deal with.

2 Lines of best fit

Now suppose that we still have points $(x_1, y_1), (x_2, y_2), \dots, (x_k, y_k)$, but instead of allowing a polynomial of degree up to $k - 1$, we're only willing to put up with a linear equation $y = ax + b$.

Of course, in this case, we can't possibly hope to hit all the points exactly. Instead, we want to minimize the errors in approximating each y_i by a prediction $ax_i + b$.

But just saying this doesn't fully specify the problem: we haven't said how to aggregate the errors. When are we willing to put up with a larger error for y_i in exchange for a smaller error for y_j?

One approach is to just add up the absolute errors:

$$\text{Error}(a, b) = |ax_1 + b - y_1| + |ax_2 + b - y_2| + \cdots + |ax_k + b - y_k|.$$

This is sometimes a reasonable thing to do. One drawback of this method is that the function is not differentiable, so straightforward methods will not work here. (It is still convex, though, so we can still optimize it with techniques we'll learn later in this class. Also, minimizing this error function is a linear program, so if you've taken a linear programming class, you already know how to do it.)

For now, we'll consider a different aggregation method:

$$\text{Error}(a,b) = (ax_1 + b - y_1)^2 + (ax_2 + b - y_2)^2 + \cdots + (ax_k + b - y_k)^2.$$

This is the squared norm $\|a\mathbf{x} + b\mathbf{1} - \mathbf{y}\|^2$, where $\mathbf{x}, \mathbf{y} \in \mathbb{R}^k$ are the vectors of x-values and y-values, and $\mathbf{1} \in \mathbb{R}^k$ is the all-ones vector. Minimizing the squared norm is equivalent to minimizing the norm, which helps make sense of this method: we are minimizing the distance between the vector of predicted values, $a\mathbf{x} + b\mathbf{1}$, and the vector of actual values, $\mathbf{y}$.

We'll develop some theory for this problem in the next lecture, but for now, we'll solve it in the straightforward way: by finding the critical points.

Here, a and b are the variable, so we compute $\frac{\partial}{\partial a}\text{Error}(a,b)$ and $\frac{\partial}{\partial b}\text{Error}(a,b)$.

For the first one, we have

$$\frac{\partial}{\partial a}(ax_i + b - y_i)^2 = 2(ax_i + b - y_i) \cdot x_i$$

and therefore

$$\frac{\partial}{\partial a}\text{Error}(a,b) = \sum_{i=1}^{k} 2(ax_i + b - y_i) \cdot x_i = 2(a\mathbf{x} + b\mathbf{1} - \mathbf{y}) \cdot \mathbf{x}.$$

For the second, we have

$$\frac{\partial}{\partial b}(ax_i + b - y_i)^2 = 2(ax_i + b - y_i)$$

and therefore

$$\frac{\partial}{\partial b}\text{Error}(a,b) = \sum_{i=1}^{k} 2(ax_i + b - y_i) = 2(a\mathbf{x} + b\mathbf{1} - \mathbf{y}) \cdot \mathbf{1}.$$

To find the critical point of $\text{Error}(a,b)$, we set both of these to 0. Then we have

$$\begin{cases} (a\mathbf{x} + b\mathbf{1} - \mathbf{y}) \cdot \mathbf{x} = 0 \\ (a\mathbf{x} + b\mathbf{1} - \mathbf{y}) \cdot \mathbf{1} = 0 \end{cases} \iff \begin{cases} (\mathbf{x} \cdot \mathbf{x})a + (\mathbf{1} \cdot \mathbf{x})b = \mathbf{y} \cdot \mathbf{x} \\ (\mathbf{x} \cdot \mathbf{1})a + (\mathbf{1} \cdot \mathbf{1})b = \mathbf{y} \cdot \mathbf{1} \end{cases}$$

which gives us a 2×2 system of equations to solve for a and b:

$$\begin{bmatrix} \mathbf{x} \cdot \mathbf{x} & \mathbf{1} \cdot \mathbf{x} \\ \mathbf{1} \cdot \mathbf{x} & \mathbf{1} \cdot \mathbf{1} \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} \mathbf{x} \cdot \mathbf{y} \\ \mathbf{1} \cdot \mathbf{y} \end{bmatrix}.$$

This critical point is a global minimizer, because $\text{Error}(a,b)$ is convex. Testing this with the Hessian matrix is not too bad, but a quick way is to note that $f(t) = t^2$ is convex (by the second derivative test), so $(ax_i + b - y_i)^2$ is convex for each i (this is just a linear substitution), and therefore $\text{Error}(a,b)$ is convex (as a sum of convex functions).

3 Examples

Suppose we just have the three points $(-1, 1)$, $(3, 1)$, $(4, 8)$.

With $k = 3$ points, there should be a polynomial through them of degree $k - 1 = 2$: a quadratic polynomial. To find it, we can set up the system

$$\begin{bmatrix} 1 & -1 & (-1)^2 \\ 1 & 3 & 3^2 \\ 1 & 4 & 4^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 8 \end{bmatrix}$$

and solve, or we can use Lagrange interpolation. If we do, we get

$$\ell_1(x) = \frac{(x-3)(x-4)}{(-1-3)(-1-4)}, \quad \ell_2(x) = \frac{(x+1)(x-4)}{(3+1)(3-4)}, \quad \ell_3(x) = \frac{(x+1)(x-3)}{(4+1)(4-3)}$$

and the polynomial

$$\ell_1(x) + \ell_2(x) + 8\ell_3(x) = \frac{1}{20}(x-3)(x-4) - \frac{1}{4}(x+1)(x-4) + \frac{8}{5}(x+1)(x-3)$$

which simplifies to $\frac{7}{5}x^2 - \frac{14}{5}x - \frac{16}{5}$.

To find a linear approximation, we take the matrix equation

$$\begin{bmatrix} \mathbf{x} \cdot \mathbf{x} & \mathbf{1} \cdot \mathbf{x} \\ \mathbf{1} \cdot \mathbf{x} & \mathbf{1} \cdot \mathbf{1} \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} \mathbf{x} \cdot \mathbf{y} \\ \mathbf{1} \cdot \mathbf{y} \end{bmatrix}$$

which becomes

$$\begin{bmatrix} (-1)^2 + 3^2 + 4^2 & -1+3+4 \\ -1+3+4 & 3 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} (-1) \cdot 1 + 3 \cdot 1 + 4 \cdot 8 \\ 1+1+8 \end{bmatrix} \iff \begin{bmatrix} 26 & 6 \\ 6 & 3 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} 34 \\ 10 \end{bmatrix}.$$

Solving, we get $a = 1$ and $b = \frac{4}{3}$, representing the line $y = x + \frac{4}{3}$.