EECS 16A Designing Information Devices and Systems I
Fall 2018 Lecture Notes Note 23

1.1 Introduction: Least Squares

In our previous exploration of linear algebra, we used techniques like Row Reduction and Matrix Inversion to solve explicit systems of equations in multiple variables. However, what happens when we introduce noise or imprecision into the system? How do we solve our system of equations in such a way that we minimize error, and get as precise a solution as we possibly can?

For example, in our GPS system, we measure the distance between the user and three beacons. However, some of the measurements might have errors. How is it possible to determine which distance reading is correct? It turns out that if we add more beacons to get more data, we can make more robust estimate of our final position.

To do this, we will explore **Least Squares**, a method through which we can minimize error by solving overdetermined systems of equations (more equations than variables). Least squares is the fundamental idea behind *data fitting* and *machine learning*: In data fitting, we find lines or curves that best match the data. In machine learning, we use a best-fit curve to make predictions about new, unseen data.

1.2 Least Squares

To explain least squares, we'll start with an example where we try to find the best fit line through n points in 2D. In later sections, we'll explore how to use the same concepts to fit more complex curves.

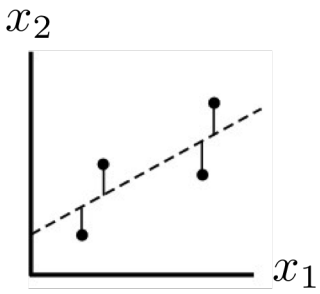
Suppose you have n points in $\mathbb{R}^2$, $(a_1,b_1),\cdots,(a_n,b_n)$ and want to fit a straight line to these points. If all these points lie on the line, then they will all satisfy the equation $b_i = x_1 a_i + x_2$ for some unknown x_1 and x_2. Written in matrix form:

$$\begin{bmatrix} a_1 & 1 \\ a_2 & 1 \\ \vdots & \vdots \\ a_n & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

However not all points may lie on the same straight line due to measurement error or model mismatch. In this case we have n equations and 2 unknowns. If $n > 2$, the system of equations may have no solutions. A general principle to apply when there's more equations than unknowns is to introduce more unknowns so that the equations can be solved. In this case, we introduce an unknown error term, e_i, for each equation, so that the equations become:

$$b_i + e_i = x_1 a_i + x_2 \tag{1}$$

This corresponds to the vertical distance between the best-fit line and points in the following diagram:



We then try to find the line that minimizes the squared vertical distance between each point and the line.

1.2.1 Least Squares Derivation

Let's think about how we would set up a system in which we have more equations (n) than unknowns (m).

$$a_{11}x_1 + a_{12}x_2 + ... + a_{1m}x_m = b_1 + e_1 \tag{2}$$
$$a_{21}x_1 + ... + ... + a_{2m}x_m = b_2 + e_2 \tag{3}$$
$$a_{n1}x_1 + ... + ... + a_{nm}x_m = b_n + e_n \tag{4}$$

With the above equations, we are trying to approximate a solution to the system with a range for the error. Essentially, we know b (in the above example it would be the distance readings), but we already know that they are a little bit off, so we capture the error with the e term.

We can put the above equations into a matrix and attempt to generate an $\vec{x}$ such that $\|\vec{e}\|$ is as small as possible. In this case, we know that the b's are noisy, so we try to separate the b's out from the noise:

$$\vec{e} = A\vec{x} - \vec{b}. \tag{5}$$

Hence, we would want to choose an $\vec{x}$ that minimizes $||\vec{e}||^2$, which is equivalent to solving the following optimization problem:

$$\min_{\vec{x}} \|\vec{e}\|^2 = \min_{\vec{x}} \left(\left\| A\vec{x} - \vec{b} \right\|^2 \right). \tag{6}$$

Here, the notation $\min_{\vec{x}} \|e\|^2$ means to find the $\vec{x}$ that gives the minimum $\|e\|^2$. To illustrate, let's use a simple example where $\vec{x}$ is one-dimensional and $n = 2$. In this case, let's use a slight abuse of notation and treat x as a scalar.

$$a_{11}x = b_1 + e_1 \tag{7}$$
$$a_{21}x = b_2 + e_2 \tag{8}$$

Hence, we have the following optimization problem:

$$\min_x [(a_{11}x - b_1)^2 + (a_{21}x - b_2)^2]. \tag{9}$$

To calculate the minimum of a function we use the idea of finding critical points, i.e. points where the slope of the function is zero. For this, we take the derivative of a function and set it equal to zero to find the critical

point. Then, we can check whether the point corresponds to a maximum or minimum by using the second derivative. A positive second derivative means that it is a minimum.

Defining $f(x)$ to be the expression we are trying to minimize, first we calculate the derivative.

$$f(x) = (a_{11}x - b_1)^2 + (a_{21}x - b_2)^2$$
$$\frac{df}{dx} = 2a_{11}(a_{11}x - b_1) + 2a_{21}(a_{21}x - b_2)$$

Then we set the derivative equal to zero and find the critical point:

$$\begin{aligned} \frac{df}{dx} &= 0 \\ 2a_{11}(a_{11}x - b_1) + 2a_{21}(a_{21}x - b_2) &= 0 \\ 2a_{11}^2 x + 2a_{21}^2 x &= 2a_{11}b_1 + 2a_{21}b_2 \\ x &= \frac{b_1a_{11} + b_2a_{21}}{(a_{11})^2 + (a_{21})^2} \end{aligned}$$

Since the second derivate of $f(x)$ is always positive, the following is the minimum of x in the 2D case:

$$x = \frac{b_1a_{11} + b_2a_{21}}{(a_{11})^2 + (a_{21})^2} = \frac{\vec{a} \cdot \vec{b}}{\|\vec{a}\|^2}$$

Note that in 2D, we know that the orthogonal projection is the shortest distance from a point onto a line. This principle generalizes to higher dimensions. (In 3D the shortest distance from a point to a plane is the orthogonal projection of the point onto the plane.)

Based on this strategy, we can try to generalize this to n dimensions.

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1m} \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ a_{n1} & \dots & \dots & a_{nm} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \dots \\ b_n \end{bmatrix} + \vec{e}$$

In order to achieve the same result as with 2 dimensions, we need $\vec{e}$ to be orthogonal to all the columns in the matrix (in order to minimize e).

$$\begin{bmatrix} a_{11} & a_{21} & \dots & a_{n1} \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \\ \dots \\ e_n \end{bmatrix} = 0$$

This can essentially be simplified to:

$$(A^T)(\vec{e}) = 0$$

Since $\vec{e}$ is just $A\vec{x} - \vec{b}$ we have the following:

$$A^T(A\vec{x} - \vec{b}) = 0$$

$$A^T A\vec{x} - A^T\vec{b} = 0$$
$$A^T A\vec{x} = A^T\vec{b}$$
$$\vec{x} = (A^T A)^{-1}A^T\vec{b} \tag{10}$$

We have just derived the least squares algorithm for the general case! The $\vec{x}$ given by Eq. (10) is the optimal solution (in the least squares sense).

1.3 Application of Least Squares

It turns out Gauss used this technique to predict where certain planets would be in their orbit. A scientist named Piazzi made 19 observations over the period of a month in regards to the orbit of Ceres (can be viewed as equations). Gauss used some of these observations. He also knew the general shape of the orbit of planets due to Kepler's laws of planetary motion. Gauss set up equations like so:

$$\alpha x^2 + \beta xy + \gamma y^2 + \delta x + \varepsilon y = \phi$$

If one divides the whole equation by ϕ, nothing significant happens so we can ignore the denominator and treat the right side of the equation as 1.

$$\alpha x^2 + \beta xy + \gamma y^2 + \delta x + \varepsilon y = 1$$

We can set up a matrix like so:

$$\begin{bmatrix} x_1^2 & x_1y_1 & y_1^2 & x_1 & y_1 \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ x_n^2 & \dots & \dots & \dots & y_n \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \\ \gamma \\ \delta \\ \varepsilon \end{bmatrix} = \begin{bmatrix} 1 \\ \vdots \\ \vdots \\ 1 \end{bmatrix}$$

Here, the x's and y's are known: they are the coordinates of the measured positions of Ceres. The unknowns are $\alpha, \dots, \varepsilon$. We write the above equation with matrix/vector notation as follows

$$A\vec{v} = \vec{b}$$

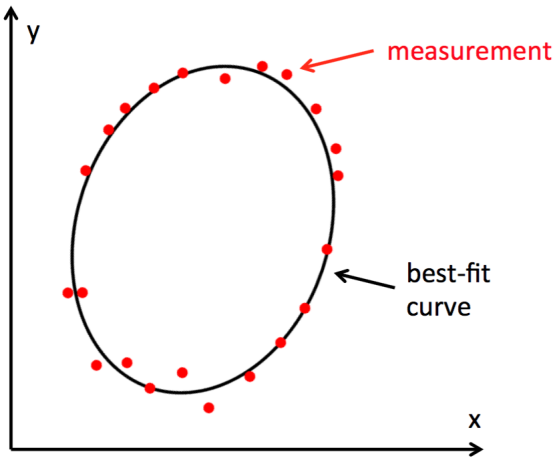
where define

$$A = \begin{bmatrix} x_1^2 & x_1y_1 & y_1^2 & x_1 & y_1 \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ x_n^2 & \dots & \dots & \dots & y_n \end{bmatrix} \qquad \vec{v} = \begin{bmatrix} \alpha \\ \beta \\ \gamma \\ \delta \\ \varepsilon \end{bmatrix} \qquad \vec{b} = \begin{bmatrix} 1 \\ \vdots \\ \vdots \\ 1 \end{bmatrix}$$

Now we can use the least squares formula to estimate the unknown coefficients in $\vec{v}$. From Eq. (10) derived earlier:

$$\vec{v} = (A^T A)^{-1}A^T\vec{b}$$
$$\begin{bmatrix} \alpha \\ \beta \\ \gamma \\ \delta \\ \varepsilon \end{bmatrix} = (A^T A)^{-1}A^T\vec{b}$$

Once solved, we can now translate the coefficients, $\alpha, \ldots, \varepsilon$, back into an ellipse using the original equation $\alpha x^2 + \beta xy + \gamma y^2 + \delta x + \varepsilon y = 1$. A possible result might look like this:



As we can see, the least squares method is useful for fitting noisy or approximate measurements to a curve, provided that we know the general shape of the curve. In addition, this example shows least squares is not limited to lines.

1.4 Practice Problems

These practice problems are also available in an interactive form on the course website.

1. True or False: Least squares is a method for solving an underdetermined system of linear equations.
2. Find the least squares solution to $\begin{bmatrix} 1 & 2 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \vec{x} = \begin{bmatrix} 2 \\ 0 \\ -8 \end{bmatrix}$.
3. True or False: Least squares always has a unique solution given by $\vec{x} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\vec{b}$.
4. True or False: We can use the least squares method to perform regression with a sine function, solving for $y = a \cdot \sin(bx)$. Make sure to also understand the reasoning why.
5. Let's say that we have a scenario where we have a set of data which includes information about a given set of patients. We have their height, weight, age, and white blood cell count. We are trying to create a predictor function for white blood cell count by solving an equation of the form $\mathbf{A}\vec{x} = \vec{b}$. What information should be in the $\mathbf{A}$ matrix?
   (a) The white blood cell counts.
   (b) The height, weight, age, and white blood cell count of each patient.
   (c) The unknown parameters $\alpha_1, \alpha_2, \alpha_3$.
   (d) The height, weight, and age for each patient.

6. True or False: Let $\vec{x} = \text{proj}_{\text{Col}(\mathbf{A})}\vec{b}$ be the projection of $\vec{b}$ onto the column space of a matrix $\mathbf{A}$. Then, $\mathbf{A}^T(\vec{b}-\vec{x}) = \vec{0}$.

7. True or False: The projection of a vector $\vec{b}$ onto a set of vectors $\{\vec{a}_1, \vec{a}_2, \ldots, \vec{a}_k\}$ is equal to $\frac{\langle \vec{b}, \vec{a}_1 \rangle}{\langle \vec{a}_1, \vec{a}_1 \rangle}\vec{a}_1 + \frac{\langle \vec{b}, \vec{a}_2 \rangle}{\langle \vec{a}_2, \vec{a}_2 \rangle}\vec{a}_2 + \cdots + \frac{\langle \vec{b}, \vec{a}_k \rangle}{\langle \vec{a}_k, \vec{a}_k \rangle}\vec{a}_k$.

8. True or False: Given an arbitrary cost function, the error vector corresponding to the best approximation of a vector $\vec{b}$ to the column space of $\mathbf{A}$ is always orthogonal to $\text{Col}(\mathbf{A})$.

9. Find the best approximation $\hat{x}$ to the system of equations $\begin{cases} a_1 x = b_1 \\ a_2 x = b_2 \end{cases}$ given the cost function $\text{cost}(x) = 2(b_1 - a_1\hat{x})^2 + (b_2 - a_2\hat{x})^2$.

   (a) $\frac{a_1 b_1 + a_2 b_2}{a_1^2 + a_2^2}$

   (b) $\frac{2a_1 b_1 + a_2 b_2}{a_1^2 + a_2^2}$

   (c) $\frac{2a_1 b_1 + a_2 b_2}{2a_1^2 + a_2^2}$

   (d) $\frac{a_1 b_1 + 2a_2 b_2}{a_1^2 + 2a_2^2}$