CS 15-859: Algorithms for Big Data Fall 2020

Lecture 1.1 — 9/10/2020

Prof. David Woodruff *Scribe: Bailey Miller*

Massive data sets are becoming ubiquitous; they show up everywhere from internet traffic logs to financial data. Classical algorithms are no longer a feasible tool for working with these data sets, as we often need run times that are nearly linear or better. Randomized approximations provide an alternative that has the efficiency we seek at the cost of approximated results with probabilistically bounded error.

1 Regression Analysis

Regression is a statistical method that studies the dependencies between variables in the presence of noise. In these notes, we focus on *linear regression* which models the linear dependencies between variables.

We can estimate the resistance R of a circuit element using linear regression. By Ohm's law $V = R \cdot I$, so if we collect the voltage for several given currents.

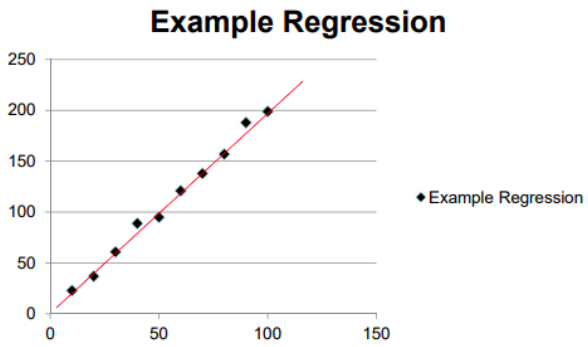

Figure 1: An example of regression studying Ohm's law

We can then use linear regression to find the linear function that best fits our data.

1.1 Standard Setting for Linear Regression

When performing linear regression, we assume that a *measured variable* b is the value of a linear function

$$b = x_0 + a_1 x_1 + \ldots + a_d x_d + \epsilon \tag{1}$$

where $a_1, \ldots, a_d$ are a set of *predictor variables* and $x_0, \ldots, x_d$ are *model parameters*. We add an ϵ term which represents *noise*.

When we have multiple observations of predictor and measured variables, we use a matrix representation. For n observations and d predictor variables we have an $n \times d$ matrix A of predictor variables and an $n \times 1$ vector b of measured variables. Ideally we find x that satisfies the linear equation

$$Ax = b. \tag{2}$$

We focus on the *over constrained* case of A where there are more observations than predictor variables $n \gg d$. In the over constrained setting, there may not be an exact solution of $Ax = b$. Instead we find x such that Ax and b are close.

1.2 Least Squares Method

Least squares is the most common technique for choosing an optimal x in the over constrained case. Least squares finds the x^* that minimizes the euclidean distance between Ax and b, that is

$$\min_x \|Ax - b\|_2^2 = \sum (b_i - \langle A_{*i}, x \rangle)^2 \tag{3}$$

where A_{*i} is the i^{th} column of A.

Least Squares has desirable statistical properties and is the maximum likelihood estimator of the model parameters if the noise ϵ is Gaussian and independent. Additionally, Least squares has interesting geometry. We can rewrite Ax as a weighted sum of columns

$$A_{1^*}x_1 + A_{2^*}x_2 + \ldots + A_{d^*}x_d \tag{4}$$

which is a linear d-dimensional subspace. Then, we can reinterpret least squares $\min_x \|Ax - b\|_2^2$ as finding the projection of b onto $\text{Col}(A)$.

1.3 Solving Least Squares via Normal Equations

When A has linearly independent columns, the closed-form solution of the least squares method $\min_x \|Ax - b\|_2^2$ is

$$x = (A^T A)^{-1} A^T b. \tag{5}$$

To show that this is a valid solution, we derive the normal equation. Express b as

$$b = Ax' + b' \tag{6}$$

where $b' \perp \text{col}(A)$. Then, we can rewrite the cost function

$$\|Ax - b\|_2^2 = \|Ax - Ax' - b'\|_2^2 = \|A(x - x') - b'\|_2^2 = \|A(x - x')\|_2^2 + \|b'\|_2^2 \tag{7}$$

where the Pythagorean theorem is used for the last equality. The critical observation here is that for $x = x'$ we will minimize the cost function.

We claim that x is optimal if and only if the following is true:

$$A^T(Ax - b) = A^T(Ax - Ax' - b') = A^T(Ax - Ax') = A^T(A(x - x')) = 0. \tag{8}$$

We drop the b' term since $b' \perp \text{Col}(A)$ and so $A^T b' = 0$. When x is optimal, it follows that $A^T(Ax - b) = 0$ which gives us the normal equation

$$A^T Ax = A^T b. \tag{9}$$

From the normal equation, we can construct an expression for the optimal x

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

If A does not have linearly independent columns, then $(A^T A)^{-1}$ does not exist and we must use an alternative approach involving the Moore-Penrose Pseudoinverse.

Definition. (Singular Value Decomposition) $A = U\Sigma V^T$ where

- U has orthonormal columns
- Σ is diagonal with non-increasing non-negative entries down the diagonal
- V^T has orthonormal rows

Definition. (Moore-Penrose Pseudoinverse)$A^- = V\Sigma^{-1}U^T$

where Σ^{-1} is a diagonal matrix with i^{th} diagonal entry equal to $\frac{1}{\Sigma_{ii}}$ if $\Sigma_{ii} \geq 0$ and is 0 otherwise.

We can use the Moore-Penrose Pseudoinverse to construct an optimal solution $x = A^- b$ to the least squares problem, regardless of whether the columns of A are linearly independent.

Claim 1. $x = A^- b$ is an optimal least squares solution.

Proof. We show $x = A^- b$ satisfies the normal equation

$$A^T Ax = A^T b \tag{11}$$
$$A^T AA^- b = A^T b \tag{12}$$
$$(V\Sigma U^T)(U\Sigma V^T)(V\Sigma^{-1}U^T)b = A^T b \tag{13}$$
$$(V\Sigma(U^T U)\Sigma(V^T V)\Sigma^{-1}U^T)b = A^T b \tag{14}$$

U, V have orthonormal columns and U^T, V^T have orthonormal rows, that means $U^T U = I$ and $V^T V = I$

$$(V\Sigma\Sigma\Sigma^{-1}U^T)b = A^T b \tag{15}$$

since $\Sigma\Sigma^{-1}$ is a diagonal matrix where entry $\Sigma\Sigma^{-1}_{i,i} = 1$ if $\Sigma_{i,i} > 0$ and 0 otherwise. Therefore, $\Sigma(\Sigma\Sigma^{-1}) = \Sigma$ and

$$(V\Sigma U^T)b = A^T b$$
$$A^T b = A^T b.$$

■

If the columns of A are not linearly independent, however, then $x = A^- b$ will not be a unique solution although it will have the minimum norm.

Claim 2. $x = A^{-}b$ has the minimum norm of all optimal solutions

Proof. Let $(V')^T$ corresponds to the rows i of V^T for which $\Sigma_{i,i} > 0$ and $(V'')^T$ correspond to the rows i of V^T for which $\Sigma_{i,i} = 0$

$$A = U \begin{bmatrix} \Sigma_{1,1} & & & & & \\ & \ddots & & & & \\ & & \Sigma_{r,r} & & & \\ & & & 0 & & \\ & & & & \ddots & \\ & & & & & 0 \end{bmatrix} \begin{bmatrix} \frac{(V')^T}{(V'')^T} \end{bmatrix} \tag{16}$$

where r is the rank of A. Then $(I - V'V'^T)$ is a projection onto the kernel space of A and so $A(I - V'V'^T)z = 0$. Therefore, $A^{-}b + (I - V'V'^T)z$ is still a solution and the solutions form a $(d - \text{rank}(A))-$dimensional affine space. Since the solutions still solve the normal equations, they are all optimal. Finally, since $A^{-}b$ is in $col(V')$ we can apply the Pythagorean theorem to the general solution form

$$\|A^{-}b + (I - V'V'^T)z\|_2^2 = \|A^{-}b\|_2^2 + \|(I - V'V'^T)z\|_2^2 \geq \|A^{-}b\|_2^2. \qquad \blacksquare$$

The most computationally expensive part of solving least squares is computing $x = A^{-}b$ which takes $O(nd^2)$ time naively. With fast matrix multiplication, the time complexity can be reduced to $O(nd^{1.376})$. We still have efficiency issues since $n \gg d$. We need much better time complexity in order to handle massive data sets.

2 Sketch and Solve Paradigm

To improve the efficiency of solving least squares, we consider computing an approximate solution x'. In particular, we want to find x' such that

$$\|Ax' - b\|_2 \leq (1 + \epsilon) \min_x \|Ax - b\|_2. \tag{17}$$

The *Sketch and Solve* method has the following steps

- Draw S from a $k \times n$ random family of matrices, for a $k << n$
- Compute SA and Sb
- Output solution x' to $\min_{x'} \|(SA)x - (Sb)\|_2$

The time complexity of solving $\min_{x'} \|(SA)x - (Sb)\|_2$ is $O(kd^2)$ where $k \ll n$ which is a major improvement over computing $A^{-}b$. We still have to compute SA which takes $O(knd)$ time. We can choose random families of matrices that are very structured so that SA is easier to compute.

Lots of random family of matrices will work, though we choose S to be a $\frac{d}{\epsilon^2} \times n$ matrix of i.i.d. Normal random variables, $S_{i,j} \sim N(0, \frac{1}{k})$. In this next set of lecture notes, we'll introduce the notion of subspace embedding to show why this random family of matrices works.