1 Linear Least Squares Regression

In this lecture we will take a new look at the fundamental problem of *linear least-squares regression*. Given data points $a_1, a_2, \ldots, a_n \in \mathbb{R}^d$ and values $b_1, \ldots, b_,$ one often hopes to find a *linear* relationship between the a_i's and b_i's. Namely, to find a linear function $f : \mathbb{R}^d \to R$ such that

$$f(a_i) = b_i$$

for all $i = 1, 2, \ldots, n$. Remember, a linear function is one where $f(a) = \sum_{i=1}^{d} x_i a_i$ for some vector $x = (x_1, x_2, \ldots, x_d)$. Linear functions have many desirable properties, and thus linear regression is often the first step that is taken when trying to understand a relationship between data points. However, because of noise in the observations of the data, it may be the case that no such f exists. Nevertheless, we can still hope to find a linear function that closely *approximates* the data. This is precisely the goal of *least-squares* linear regression. We can form the data points $a_1, \ldots, a_n \in \mathbb{R}^d$ into a matrix matrix $A \in \mathbb{R}^{n \times d}$, and form the b_i's into a vector $b \in \mathbb{R}^n$. Then the least squares regression problem is to find a vector $x \in \mathbb{R}^d$ that minimizes the following objective function:

$$\min_x \sum_{i=1}^{n} (a_i^\intercal x - b_i)^2$$

I.e., we sum the squares of the errors between the prediction of the linear function on data point a_i (namely $a_i^\intercal x$) and the actual value b_i. This can be rewritten as

$$\min_x \|Ax - b\|^2 \tag{1}$$

where for a vector $y \in \mathbb{R}^n$, the squared Euclidean length is $\|y\|^2 = \sum_{i=1}^{n} y_i^2$. So while there may not exist any x with $Ax = b$, regression seeks to find the $x \in \mathbb{R}^d$ that *best fits* the observed data, where best fit means minimizing the sum-of-squares objective function above. The figure below an example of linear regression for the case of $n = 4, d = 1$.

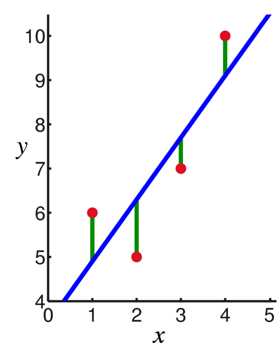

Figure 1: In linear regression, observations (red) are assumed to be the result of some deviations (green) from an underlying relationship (blue) between a dependent variable (b) and an independent variable (x). We want to recover the true $x \in \mathbb{R}^d$ (image from Wikipedia).

In this lecture, we are focused on the setting where the quantity of data is *enormous* (big data!). Namely, when there are many more data points n (the rows of A) than control variables d (coordinates of x). This is referred to as the *over-constrained case*, where $n \gg d$. Because of the size of n, our goal will be to solve regression with small runtime with respect to n.

1.1 A Strawman Solution

In the homework, you showed that for symmetric square matrices A, any optimal solution x^* satisfies the *normal equations*. Namely, if $x^* = \arg\min_x \|Ax - b\|^2$, and if A is an $n \times n$ symmetric matrix, then it must be that $A^2 x^* = Ab$. In fact, the normal equations extend to non-symmetric matrices as follows. For any $A \in \mathbb{R}^{n\times d}$ and $b \in \mathbb{R}^n$, if $x^* = \arg\min_x \|Ax - b\|$ then

$$A^\intercal A x^* = A^\intercal b$$

(Check this!). For simplicity, we assume the columns of A are linearly independent. In this case, $(A^\intercal A)$ is invertible, thus we can solve x^* via

$$x^* = (A^\intercal A)^{-1} A^\intercal b.$$

The formula looks a bit daunting at first, but as you can see the ideas are very simple.[1] However, computing $(A^\intercal A)^{-1}$ requires $O(\min\{n^2 d, d^2 n\}) = O(nd^2)$ time, which can be prohibitively large for large n.

In these notes, we will see how, if we allow our solution to the regression problem (1) to be *approximately* optimal, we can achieve running time $O(\text{nnz}(A) + \text{poly}(d/\epsilon))$, where $\text{nnz}(A)$ is the number of non-zero entries in A (note that $\text{nnz}(A) \le nd$, so the runtime is always an improvement over using the normal equations), and ϵ is a accuracy parameter. We assume that A has no non-zero rows, so $\text{nnz}(A) \ge n$.

2 Approximate Regression

We first formalize what it means to approximately solve the linear regression problem (1) on an input matrix $A \in \mathbb{R}^{n\times d}$ and vector $b \in \mathbb{R}^n$.

Definition 1 (Approximate Linear Regression) *Given $A \in \mathbb{R}^{n\times d}, b \in \mathbb{R}^n$, and $\epsilon > 0$, the ϵ-approximate regression problem is to find $x' \in \mathbb{R}^d$ such that*

$$\|Ax' - b\|^2 \le (1+\epsilon) \min_{x\in\mathbb{R}^d} \|Ax - b\|^2.$$

We now describe an approach to solve this problem in time $O(\text{nnz}(A) + \text{poly}(d/\epsilon))$. For $n \gg d$, this represents a substantial improvement over the earlier $O(nd^2)$ running time. Our approach is known as the *sketch-and-solve* approach, which is as follows.

[1] If A does not have full column rank, the optimal solution is given by the Moore Penrose pseudo-inverse. This is a very nice idea, but we will skip over this concept for now.

Sketch-and-Solve Paradigm:

1. First, we choose a matrix $S \in \mathbb{R}^{k \times n}$ for $k \ll n$, where the entries of S are drawn *randomly* from some distribution that we will soon specify. The matrix S is known as a *sketching matrix.*

2. Then, we compute $\mathbf{A} := SA$ and $\mathbf{b} := Sb$. Note now that the matrix $\mathbf{A} = SA \in \mathbb{R}^{k \times d}$ and the vector $\mathbf{b} = Sb \in \mathbb{R}^k$, so the dimension n has disappeared, and the matrices are now much smaller.

3. Optimally solve (via the normal equations) the optimization problem $x' = \arg\min_x \|\mathbf{A}x - \mathbf{b}\|$. Output this solution $x' = (\mathbf{A}^\intercal \mathbf{A})^{-1} \mathbf{A}^\intercal \mathbf{b} \in \mathbb{R}^d$.

Runtime. Note that once the sketches $\mathbf{A}, \mathbf{b}$ have been computed, the last step requires only $\min\{k^2 d, d^2 k\}$ time. Thus, if we set k to be at most some polynomial in d and $1/\epsilon$, the total running time to solve $\min_x \|\mathbf{A}x - \mathbf{b}\|$ (once $\mathbf{A}, \mathbf{b}$ are computed) will be $O(\text{poly}(d/\epsilon))$. In fact, we will show that $k = \Theta(d^2/\epsilon^2)$ suffices, so the running time would be $O(d^4/\epsilon^2)$. Furthermore, we will choose S from a family of matrices so that the products $\mathbf{A} = SA, \mathbf{b} = Sb$ can be computed in $O(\text{nnz}(A))$ time. Taken all together, the whole procedure can then be carried out in $O(\text{nnz}(A) + \text{poly}(d/\epsilon))$ time as claimed.

2.1 The Count-Sketch Matrix

We now (re-)introduce the *count-sketch* matrix S (which we saw in Lecture 22 as well). This is the sketching matrix we will use. For an integer $n \geq 0$, let $[n] = \{1, 2, \ldots, n\}$.

Definition 2 *Fix k, n, and let $S \in \mathbb{R}^{k \times n}$ be defined as follows. First, we pick a 2-wise independent hash function $h : [n] \to [k]$ and a 4-wise independent hash function $s : [n] \to \{1, -1\}$. Then we define S via:*

$$S_{i,j} = \begin{cases} s(j) & \text{if } h(j) = i \\ 0 & \text{otherwise} \end{cases}$$

Note then that S is a matrix consisting only of the values $\{0, 1, -1\}$. Moreover, every column of S has exactly one non-zero value, which is placed in a random row (chosen via the hash function h), and is given a random sign (chosen via the hash function s). E.g., here is what S may look like.

$$\begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 & -1 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

One advantage of this matrix is that S can be applied to the matrix A in $\text{nnz}(A)$ time.

Claim 3 *The matrix product $\mathbf{A} = SA$ can be computed in $O(nnz(A))$ time.*

Proof: We show that for any column vector $v \in \mathbb{R}^n$, we can compute Sv in $O(\text{nnz}(v))$ time, which will complete the proof since $\mathbf{A} = SA$ consists of d such products. To see this, note that each non-zero entry of v_i of v effects only the coordinate $(Sv)_{h(i)}$, since each column i of S has a single non-zero value $s(i)$, which is in row $h(i)$. Thus, to compute Sv, we can simply intialize a vector $y = 0 \in \mathbb{R}^k$. Then, for each non-zero entry v_i of v, we update $(Sv)_{h(i)} \leftarrow (Sv)_{h(i)} + s(i)v_i$, which requires $O(1)$ time per non-zero entry of v. ■

Thus, we have now shown we can carry the sketch-and-solve steps outlined above, with count-sketch S as the sketching matrix, in the desired runtime. It remains now to show correctness, namely that we obtain an $(1 + \epsilon)$ approximate solution to the regression problem.

2.2 Correctness of the Algorithm

Our analysis will crucially rely on the definition of a *subspace embedding.*

2.2.1 Subspace Embeddings

Loosely speaking, a sketching matrix S is a subspace embedding for A if the length of *every* vector in the column span of A is approximately preserved after multiplication by S on the left.

Definition 4 *Fix any matrix A, and let $\mathcal{V} \subseteq \mathbb{R}^n$ be the column span of A. Then a matrix $S \in \mathbb{R}^{k\times n}$ is said to be a ϵ-*subspace embedding *for $\mathcal{V}$ (or for matrix A) if for all vectors $x \in \mathcal{V}$ we have*

$$\|Sx\| \in (1 \pm \epsilon)\|x\|$$

Equivalently, S is a subspace embedding for A if for all vectors $x \in \mathbb{R}^d$, we have

$$\|SUx\| \in (1 \pm \epsilon)\|Ux\|$$

where $U \in \mathbb{R}^{n\times d}$ is an orthonormal basis for the column span of A.

Here, for $a, b \in \mathbb{R}$ we use the notation $a \in (1 \pm \epsilon)b$ to denote $(1 - \epsilon)b \leq a \leq (1 + \epsilon)b$.

The main technical challenge will now be to show that, with good probability, if S is a randomly generated instance of count-sketch, then it is a subspace embedding for $[A, b]$ when $k = \Omega(d^2/\epsilon^2)$. Here $[A, b]$ is the matrix A with an additional column b appended. Before we do this, we first show how S being a subspace embedding for $[A, b]$ implies the correctness of the sketch-and-solve routine.

Claim 5 (Subspace Embeddings imply Correctness) *If S is a subspace embedding for $[A, b]$, then the sketch-and-solve routine solves the ϵ-approximate regression problem.*

Proof: For all vectors $x \in \mathbb{R}^d$, since $Ax - b$ is in the column span of $[A, b]$, we have $\|S(Ax - b)\| = (1 \pm \epsilon)\|Ax - b\|$ for all $x \in \mathbb{R}^d$, thus in particular we have $\min_x \|S(Ax - b)\| \leq (1+\epsilon) \min_x \|Ax - b\|$. So solving for the optimal x' that minimizes $\|S(Ax - b)\|$ yields a solution to the ϵ-approximate regression problem. ■

2.2.2 Count-Sketch gives Subspace Embeddings (Optional)

To show that S is a subspace embedding, we will first show that S satisfies a property known as *approximate matrix product.* We introduce some notation. For a count-sketch $S \in \mathbb{R}^{k\times n}$ and any $(i, j) \in [k] \times [n]$, let $\delta_{i,j} = 1$ if $S_{i,j} \neq 0$, and let $\delta_{i,j} = 0$ otherwise. So $\delta_{i,j}$ simply indicates whether $S_{i,j}$ is zero or not. For any column $j \in [n]$, let $\sigma(j) \in \{1, -1\}$ denote the sign of the non-zero entry in the j-th column of S. Finally, we define the *Frobenius norm* of a matrix (or vector):

$$\|A\|_F = \sqrt{\sum_{i,j} A_{i,j}^2}.$$

Note that the Frobenius norm of a vector is the same as the Euclidean norm.

The approximate matrix product theorem says what you may expect: if you take two matrices A and B, and then sketch them down to get $\mathbf{A} = SA$ and $\mathbf{B} = SB$, then the product $\mathbf{A}^\intercal \mathbf{B}$ is close to the actual product $A^\intercal B$—the error is small with high probability.

Lemma 6 (Approximate Matrix Product) *Let $S \in \mathbb{R}^{k \times n}$ be a count sketch matrix such that $k \geq \Omega(\frac{1}{\epsilon^2 \delta})$. Let A, B be any two matrices with n rows, and let $\mathbf{A} := SA$ and $\mathbf{B} := SB$. Then we have*

$$\Pr\left[\quad \|\mathbf{A}^\intercal \mathbf{B} - A^\intercal B\|_F \quad \leq \quad \epsilon \|A\|_F \|B\|_F \quad \right] \geq 1 - \delta.$$

Proof: Let $\mathbf{C} = \mathbf{A}^\intercal \mathbf{B}$. The approach of this proof is simple:

#1. look at the random variable $\|\mathbf{C} - A^\intercal B\|_F^2 = \sum_{u,u'}((\mathbf{C} - A^\intercal B)_{u,u'})^2$, and upper bound its expected value as follows:

$$\mathbb{E}\left[\|\mathbf{C} - A^\intercal B\|_F^2\right] \leq \frac{2\|A_u\|^2 \|B_{u'}\|^2}{k}. \tag{2}$$

#2. Then we can apply Markov's inequality to bound the probability that $\|\mathbf{C} - A^\intercal B\|_F^2$ is too much larger than its expectation. Specifically, if $k \geq \frac{2}{\epsilon^2 \delta}$, then Markov's inequality gives:

$$\Pr\left[\|\mathbf{C} - A^\intercal B\|_F^2 \geq \epsilon^2 \|A\|_F^2 \|B\|_F^2\right] \leq \delta$$

as desired.

So it just remains to prove (2), and bound the expectation, which is what the rest of the proof does. First, observe that for any entry (u, u') in $\mathbf{C}$, we can write:

$$\mathbf{C}_{u,u'} = \sum_{t=1}^{k} \sum_{i,j \in [n]} \sigma(i)\sigma(j)\delta_{t,i}\delta_{t,j} A_{i,u} B_{j,u'} = \sum_{t=1}^{k} \sum_{i \neq j \in [n]} \sigma(i)\sigma(j)\delta_{t,i}\delta_{t,j} A_{i,u} B_{j,u'} + (A^\intercal B)_{u,u'}$$

Now since $\sigma(i)\sigma(j)$ are independent for $i \neq j$, we have $\mathbb{E}[\sigma(i)\sigma(j)] = 0$, so $\mathbb{E}[\mathbf{C}_{u,u'}] = (A^\intercal B)_{u,u'}$, namely that the desired property holds in expectation. We now consider the variance: $\mathbb{E}[((\mathbf{C} - A^\intercal B)_{u,u'})^2]$. We have

$$\begin{aligned} ((\mathbf{C} - A^\intercal B)_{u,u'})^2 = \sum_{t_1,t_2=1}^{k} \sum_{i_1 \neq j_1, i_2 \neq j_2 \in [n]} & \sigma(i_1)\sigma(i_2)\sigma(j_1)\sigma(j_2) \cdot \delta_{t_1,i_1}\delta_{t_1,j_1}\delta_{t_2,i_2}\delta_{t_2,j_2} \\ & \cdot A_{i_1,u} A_{i_2,u} B_{j_1,u'} B_{j_2,u'} \end{aligned} \tag{3}$$

For a given term in the summation to have a non-zero expectation, it must be the case that $\mathbb{E}[\sigma(i_1)\sigma(i_2)\sigma(j_1)\sigma(j_2)] \neq 0$. Since the random signs $\sigma(\cdot)$ are 4-wise independent, the expectation is always 0 unless each of the indicies i_1, i_2, j_2, j_2 appear in even multiplicity in the term $\sigma(i_1)\sigma(i_2)\sigma(j_1)\sigma(j_2)$. Since the signs are ± 1 variables, the expectation must be 1 if it is not zero. Thus, for the expectation to be non-zero, one of the following cases must occur: either 1) we have $i_1 = i_2$ and $j_1 = j_2$, or 2) we have $i_1 = j_2$ and $j_1 = i_2$. We first show that the total contribution of the terms where $i_1 = i_2$ and $j_1 = j_2$ is bounded by $\frac{\|A_u\|_2^2 \|B_{u'}\|_2^2}{k}$, where A_u is the u-th column of A. Note that if $t_1 \neq t_2$, we always have $\delta_{t_1,i_1}\delta_{t_2,i_2} = 0$, since the non-zero entry in the $i_1 = i_2$

column of S cannot be in two distinct rows at once (there is only one such non-zero entry). Note moreover that for distinct $t_1 \neq t_2$, since the hash function h was pairwise independent, we have $\mathbb{E}[\delta^2_{t_1,i_1}\delta^2_{t_1,j_1}] = \mathbb{E}[\delta_{t_1,i_1}\delta_{t_1,j_1}] = \frac{1}{k} \cdot \frac{1}{k} = 1/k^2$, since this is the probability that $h(i_1) = t_1$ and $h(i_2) = t_2$. Keeping this in mind, then for a fixed $i_1 = i_2$ and $j_1 = j_2$, we have

$$\begin{aligned}
&\mathbb{E}\left[\sum_{t_1,t_2=1}^{k} \sigma(i_1)\sigma(i_2)\sigma(j_1)\sigma(j_2) \cdot \delta_{t_1,i_1}\delta_{t_1,j_1}\delta_{t_2,i_2} \cdot A_{i_1,u}A_{i_2,u}B_{j_1,u'}B_{j_2,u'}\right] \\
&= \mathbb{E}\left[\sum_{t_1=1}^{k} \delta^2_{t_1,i_1}\delta^2_{t_1,j_1} A^2_{i_1,u}B^2_{j_1,u'}\right] \\
&= \frac{A^2_{i_1,u}B^2_{j_1,u'}}{k}
\end{aligned} \tag{4}$$

Summing over all possible values of i_1, j_1, we get the desired upper bound of $\frac{\|A_u\|^2\|B_{u'}\|^2}{k}$. The case where $i_1 = j_2$ and $j_1 = i_2$ is analogous, where we can obtained the same upper bound of $\frac{\|A_u\|^2\|B_{u'}\|^2}{k}$ on the expectation of these terms. This shows (2), and hence completes the proof. ■

Now that we have shown that S satisfies the approximate matrix product property, we are finally ready to prove that S is a subspace embedding for $[A, b]$ with good probability when $k = \Omega(d^2/\epsilon^2)$. To do this, we will need the well-known (and highly useful!) Cauchy-Schwarz inequality.

Lemma 7 (Cauchy-Schwarz) *Let $v, u \in \mathbb{R}^n$ be vectors. Then*

$$|\langle v, u\rangle| \leq \|u\|\|v\|,$$

where $\langle u, v\rangle = \sum_{i=1}^n u_i v_i$ is the inner product. Moreover, if A is a matrix, then

$$\|Av\| \leq \|A\|_F\|v\|.$$

Proof: If θ is the angle between the vectors v, u (that is, the angle between the two vectors in the plane they span), then the dot product over Euclidean space satisfies $\langle u, v\rangle = \|u\|\|v\|\cos(\theta)$. The desired inequality follows from the fact that $\cos(\theta) \leq 1$ for all θ. The second claim follows from application of Cauchy-Schwarz on each coordinate of Av, which itself is an inner product between a row of A and v. ■

Theorem 8 (Subspace Embedding) *Let $\mathcal{V}$ be any fixed d-dimensional subspace. Then if $k \geq \Omega(\frac{d^2}{\epsilon^2\delta})$, then with probability at least $1-\delta$, we have for all $x \in \mathcal{V}$ simultaneously:*

$$\|Sx\| = (1 \pm \epsilon)\|x\|$$

Proof: Let $U \in \mathbb{R}^{n\times d}$ be an orthonormal basis for the subspace $\mathcal{V}$. Since U is orthonormal, we have $U^\intercal U = I_d$ (because the columns of U are orthogonal and normal, i.e., of unit length) and $\|U\|_F^2 = d$ (because orthogonal matrices have column norm 1 for every column). Also note that since the columns of U are orthogonal, for any vector $x \in \mathbb{R}^d$ we have $\|Ux\|^2 = \|x\|^2$ (orthogonal matrices preserve distances).

Use $\mathbf{U}$ to denote the sketch SU, and let $\epsilon' = \epsilon/d$. Then since $k \geq \Omega(\frac{d^2}{\epsilon^2\delta}) \geq \Omega(\frac{1}{\epsilon'^2\delta})$, by the approximate matrix product property from Lemma 6, with probability at least $1-\delta$ we have

$$\|\mathbf{U}^\intercal\mathbf{U} - U^\intercal U\|_F \leq \epsilon'\|U\|_F^2 \quad\implies\quad \|\mathbf{U}^\intercal\mathbf{U} - I_d\|_F \leq \left(\frac{\epsilon}{d}\right)\cdot d \leq \epsilon. \tag{5}$$

Now for any vector $x \in \mathbb{R}^d$,

$$\begin{aligned}\|\mathbf{U}x\|^2 - \|Ux\|^2 &= (\mathbf{U}x)^\intercal(\mathbf{U}x) - (Ux)^\intercal(Ux)\\ &= x(\mathbf{U}\mathbf{U} - I_d)x\end{aligned}$$

But now we can apply Cauchy-Schwarz twice, to say

$$\begin{aligned}&\leq \|x\|\|(\mathbf{U}^\intercal\mathbf{U} - I_d)x\| && \text{(by Cauchy-Schwarz)}\\ &\leq \|x\|\|(\mathbf{U}^\intercal\mathbf{U} - I_d)\|_F\|x\| && \text{(again by Cauchy-Schwarz)}\\ &\leq \epsilon\|x\|^2 && \text{(by (5))}\\ &= \epsilon\|Ux\|^2.\end{aligned}$$

This implies that $\|\mathbf{U}x\|^2 \leq (1+\epsilon)\|Ux\|^2$ for all $x \in \mathbb{R}^d$; taking square roots and using that $\sqrt{(1+\epsilon)} \leq (1+\epsilon)$ gives us $\|\mathbf{U}x\| \leq (1+\epsilon)\|Ux\|$. A similar calculation shows that $\|\mathbf{U}x\| \geq (1-\epsilon)\|Ux\|$.

Thus $\|SUx\| = \|\mathbf{U}x\| = (1\pm\epsilon)\|Ux\|$ for any $x \in \mathbb{R}^d$. Since any vector $y \in \mathcal{V}$ can be written as $y = Ux$ for some $x \in \mathbb{R}^d$, we have $\|Sy\| = (1\pm\epsilon)\|y\|$ for all $y \in \mathcal{V}$, which completes the proof. ■

Finally, since the subspace $\mathcal{V}$ spanned by $[A, b]$ is at most $d+1$ dimensional, we conclude that setting $k = \Theta(\frac{d^2}{\epsilon^2\delta})$ is sufficient for S to be a subspace embedding with probability at least $1-\delta$, which completes the proof of the approximate linear regression algorithm.