Lecture 8 — January 30

Lecturer: Mert Pilanci *Scribe: Raymond Ye Lee*

To motivate randomized least squares, it's useful to think of randomized least squares as approximating one objective function with another that is cheaper to compute. We see this idea often in other fields: for example, in constructing generalization bounds for machine learning algorithms, we approximate population loss using the sample/empirical loss.

8.1 Least Squares Problems and Random Projection

Recall the least squares problem: given $A \in \mathbb{R}^{n\times d}$ and $b \in \mathbb{R}^n$, we want to find the best approximation $x \in R^d$ such that $Ax \approx b$, i.e.,

$$x_{LS} = \arg\min_{x\in\mathbb{R}^d} \|Ax - b\|_2^2$$

If A has full column rank, then the solution x_{LS} is given by $x_{LS} = (A^TA)^{-1}A^Tb$. Let $A = U\Sigma V^T$ be a singular value decomposition. As the pseudoinverse of A is given by $A^+ = V\Sigma^+U^T$, we can also write the least squares solution as

$$x_{LS} = A^+b = V\Sigma^+U^Tb,$$

so that $b \in \mathcal{R}(A)$ and $b = Ax_{LS} + b^\perp$. We'll see that expressing b in this way gives us a tool to prove unbiasedness of randomized least squares.

8.2 Faster Least Squares Optimization: Random Projection

One way to obtain a faster least squares solution is random projection. Suppose we have a random projection matrix $S \in \mathbb{R}^{m\times n}$ and instead work with the problem

$$\min_{x\in\mathbb{R}^d} \|SAx - Sb\|_2^2.$$

Choosing S so that $\mathbb{E}[S^TS] = I$, we have

$$\mathbb{E}\left[\|SAx - Sb\|_2^2\right] = \mathbb{E}\left[(Ax-b)^TS^TS(Ax-b)\right] = (Ax-b)^T(Ax-b),$$

which was our original objective. This is the left-sketching approach.

This sketched problem can be solved using any classical method with direct method complexity $O(md^2)$. Of the direct methods, Cholesky decomposition is the most commonly used, while conjugate gradient is the most commonly used of the indirect methods. On the other hand, gradient descent with sparse A and dense S could actually take longer on the sketched problem than on the original one, as the matrix SA is usually dense.

We'll see that $m \gg d$ in practice, or else the variance with respect to our sketched solution will blow up, i.e., d is a computational lower limit on m, analogous to channel capacity.

8.3 Approximation Result

Consider $A \in \mathbb{R}^{n\times d}$ with $n \gg d$ and let $S \in \mathbb{R}^{m\times d}$ be a Johnson-Lindenstrauss embedding. Then in lieu of the classical least squares problem

$$x_{LS} = \arg\min_{x\in\mathbb{R}^d} \|Ax - b\|_2^2 ,$$

we instead solve the problem

$$\tilde{x} = \arg\min_{x\in\mathbb{R}^d} \|SAx - Sb\|_2^2 .$$

Denote $f(x_{LS}) = \|Ax - b\|_2^2$ and $f(\tilde{x}) = \|SAx - Sb\|_2^2$. We have the follow result.

Lemma 1. If $m \geq \text{constant} \cdot \frac{\text{rank}(A)}{\epsilon^2}$, then

$$f(x_{LS}) \leq f(\tilde{x}) \leq (1+\epsilon^2) f(x_{LS})$$

and

$$\|A(x_{LS} - \tilde{x})\|_2^2 \leq \epsilon^2$$

with high probability.

So we have both upper and lower bounds on $f(\tilde{x})$, as well as a probabilistic upper bound on the prediction difference $Ax_{LS} - A\tilde{x}$ in the 2-norm sense.

8.4 Application: Streaming data

Consider a setting where we need to update our least squares solution in each time period. For example, suppose $A \in \mathbb{R}^{n\times d}$ is our feature matrix.

A naive approach would be to update the entries of A in each time period re-solve the least squares problem; this would require $O(nd^2)$ time. In seeking a better solution, we can look to the normal equations for insight:

$$A^T A = A^T b.$$

Notice that $A^T A \in \mathbb{R}^{d\times d}$. With $n \gg d$, this is a relatively smaller problem, and therefore certainly less expensive to solve.

However, updates to $A^T A$ are in general expensive. For example, an update $A_{t+1} = A_t + \Delta_t$ at time $t+1$ for some $\Delta_t \in \mathbb{R}^{n\times d}$ would result in additional matrix multiplications on top of the multiplication required to construct $A_t^T A_t$:

$$(A_t + \Delta_t)^T (A_t + \Delta_t) = A_t^T A_t + A_t^T \Delta_t + \Delta_t^T A_t + \Delta_t^T \Delta_t$$

as well as additional memory. Linear sketching, on the other hand, requires only $O(md)$ memory to store the data, so using the random projection matrix to track A and b—i.e., $SA_{t+1} = SA_t + S\Delta_t$ and $Sb_{t+1} = Sb_t + S\Delta_t$—provides a more memory-efficient solution in this dynamic setting.

8.5 Gaussian Sketch

Now we examine a particular formulation of S: the Gaussian sketch. That is, let S have i.i.d. Gaussian entries i.e., $S_{ij} = \frac{1}{\sqrt{m}} N(0,1)$ with $\mathbb{E}[S^T S] = I$. Recall that the sketched least squares solution is given by

$$\tilde{x} = \arg\min_{x \in \mathbb{R}^d} \|SAx - Sb\|_2^2 .$$

As S is random, an important question is whether $\mathbb{E}[\tilde{x}]$ is equal to x_{LS}, so that in expectation we obtain the true least squares solution.

Assuming that $A^T S^T SA$ is invertible (which will hold with high probability when $m \gg d$), we can write

$$\tilde{x} = (A^T S^T SA)^{-1} A^T S^T Sb$$

which, rewriting $b = Ax_{LS} + b^\perp$ where $b^\perp \perp \text{Range}(A)$, we can write

$$\begin{aligned} \tilde{x} &= (A^T S^T SA)^{-1} A^T S^T S(Ax_{LS} + b^\perp) \\ &= x_{LS} + (A^T S^T SA)^{-1} A^T S^T Sb^\perp \end{aligned}$$

Then for $\mathbb{E}[\tilde{x}] = x_{LS}$ to hold, we need

$$\mathbb{E}[(A^T S^T SA)^{-1} A^T S^T Sb^\perp]$$

to vanish. To see that this will happen, note that SA and $Sb^\perp$ are uncorrelated, which then implies independence. Then we can rewrite the expectation

$$\begin{aligned} \mathbb{E}[\tilde{x}] &= \mathbb{E}_{SA}[(A^T S^T SA)^{-1} A^T S^T] \cdot \mathbb{E}_{Sb^\perp}[Sb^\perp] && \text{(independence)} \\ &= \mathbb{E}_{SA}[(A^T S^T SA)^{-1} A^T S^T] \cdot 0 && (Sb^\perp \text{ has zero mean}) \\ &= 0 \end{aligned}$$

and so we see that indeed, the expectation of the randomly projected solution is equal to the true solution using the Gaussian sketch.

8.6 Gaussian Sketch: Variance

We also want to analyze the variance

$$\mathbb{E}[\|A\tilde{x} - Ax_{LS}\|_2^2].$$

One thing to note is that we can achieve a lower variance when the objective value given by $f(x_{LS}) = \|Ax - b\|_2^2$ is small, analogous, for example, to the variance being low when the training loss is small when fitting machine learning models.

Something else to note is that throughout, we assume that the inverse of $A^T S^T SA$ exists. In fact, if the eigenvalues of this matrix are small, i.e., $A^T S^T SA$ is nearly not invertible, then the variance will increase accordingly. However, we have some control over this via construction of S.

In analyzing the variance, we first condition on SA to obtain a distribution and then relax this assumption. Note that by fixing SA, the only randomness in $\tilde{x}$ comes from $Sb^\perp$, which we can write as:

$$Sb^\perp = \begin{bmatrix} S_1^T b^\perp \\ S_2^T b^\perp \\ \vdots \end{bmatrix} = \begin{bmatrix} \vdots \\ \sum_j S_{ij} b_j^\perp \\ \vdots \end{bmatrix}.$$

Then

$$\begin{aligned} \operatorname{Var}\left[\sum_j S_{ij} b_j^\perp\right] &= \sum_i \left(b_j^\perp\right)^2 \cdot \frac{1}{m} \qquad \text{(since } S_i j \sim \frac{1}{\sqrt{m}} \text{ Gaussian)} \\ &= \left\|b^\perp\right\|_2^2 \cdot \frac{1}{m} \\ &= \|b - Ax_{LS}\|_2 \cdot \frac{1}{m} \\ &= f(x_{LS}) \cdot \frac{1}{m} \end{aligned}$$

so we have that $Sb^\perp \sim N\left(0, \frac{f(x_{LS})}{m} I\right)$ and

$$\tilde{x} \sim N\left(x_{LS}, \frac{f(x_{LS})}{m}(A^T S^T SA)^{-1}\right)$$

from which it follows that

$$A\left(\tilde{x} - x_{LS}\right) \sim N\left(0, \frac{f(x_{LS})}{m} A\left(A^T S^T SA\right)^{-1} A^T\right).$$

Now consider SA no longer fixed. Though $\mathbb{E}[A^T S^T SA]$ is unbiased for $A^T A$, $\mathbb{E}[(A^T S^T SA)^{-1}]$ is biased for $(A^T A)^{-1}$ with factor $\frac{m}{m-d-1}$ (we'll see later where this factor comes from). That is, for $m > d+1$, we have

$$\mathbb{E}\left[\left(A^T S^T SA\right)^{-1}\right] = \left(A^T A\right)^{-1} \frac{m}{m-d-1}$$

Therefore, if $m = d-1$ for instance, the variance will blow up.
Note that for $z \sim N(0, K)$, we have

$$\mathbb{E}[\|z\|_2^2] = \mathbb{E}[\operatorname{tr} zz^T] = \mathbb{E}[\operatorname{tr} K],$$

so we can write

$$\begin{aligned} \mathbb{E}\left\|A\left(\tilde{x} - x_{LS}\right)\right\|_2^2 &= \mathbb{E}\left[\frac{f\left(x_{LS}\right)}{m} \operatorname{tr} A\left(A^T S^T SA\right)^{-1} A^T\right] \\ &= \frac{f\left(x_{LS}\right)}{m-d-1} \operatorname{tr} A\left(A^T A\right)^{-1} A^T \end{aligned}$$

Now notice that $A\left(A^TA\right)^{-1}A^T = AA^+$ projects onto the column space of A.
Let $A = U\Sigma V^T$ be an SVD of A.

$$\begin{aligned}
\operatorname{tr} A\left(A^TA\right)^{-1}A^T &= \operatorname{tr} UU^T && \text{(equivalent projection matrices)}\\
&= \operatorname{tr} U^TU && \text{(cyclic property of trace)}\\
&= \operatorname{tr} I_{d\times d}\\
&= d\\
&= \operatorname{rank}(A)
\end{aligned}$$

So we conclude

$$\mathbb{E}\left\|A\left(\tilde{x}-x_{LS}\right)\right\|_2^2 = f\left(x_{LS}\right)\frac{d}{m-d-1}.$$

where $f(x_{LS})\frac{m}{m-d-1}$ gives us an idea of how to set m in constructing S in order to obtain a certain expected error.

8.7 Expected Inverse of a Random Matrix

We examine the expression

$$\mathbb{E}\left[\left(A^TS^TSA\right)^{-1}\right] = (A^TA)^{-1}\frac{m}{m-d-1}$$

in greater detail. Let $A = U\Sigma V^T$ be a compact SVD of A. Then we can write

$$\begin{aligned}
(A^TS^TSA)^{-1} &= (V\Sigma U^TS^TSU\Sigma V^T)^{-1}\\
&= V^T\Sigma^{-1}(U^TS^TSU)^{-1}\Sigma^{-1}V
\end{aligned}$$

and since S is i.i.d $\sim N(0, \frac{1}{\sqrt{m}}$, SU will also be i.i.d. Gaussian as U is orthogonal and therefore a rotation. Then

$$\mathbb{E}[(U^TS^TSU)^{-1}] = I\cdot\text{constant}$$

so

$$\begin{aligned}
\mathbb{E}[V^T\Sigma^{-1}(U^TS^TSU)^{-1}\Sigma^{-1}V] &= V^T\Sigma^{-2}V\cdot\text{constant}\\
&= (A^TA)^{-1}\cdot\text{constant}
\end{aligned}$$

where the constant above is equal to the factor $\frac{m}{m-d-1}$ we saw earlier. This constant is the expected value of a certain χ^2 random variable, a result from random matrix theory.

8.8 Which sketching matrices are good?

We saw that with random sketching, some conditions needed to be fulfilled in order to guarantee approximate optimality (e.g., recall that the Gaussian sketch requires A^TS^TSA to be invertible). We can also consider deterministic formulations of the sketching matrix S. Let $A = U\Sigma V^T$ be an SVD of A in compact form.

Option 1: $S = U^T$

Suppose we choose $S = U^T$, i.e, the matrix containing the left singular vectors in its rows as our sketching matrix.

$$\begin{aligned}
\tilde{x} &= (SA)^+ Sb \\
&= (U^T U \Sigma V^T)^+ Sb \\
&= (\Sigma V^T)^+ Sb \\
&= V \Sigma^{-1} Sb \\
&= V \Sigma^{-1} U^T b \\
&= A^+ b \\
&= x_{LS}
\end{aligned}$$

So we've recovered the least squares solution exactly using $S = U^T$! However, a prerequisite for doing so requires the costly computation of the left singular vectors (takes $O(nd^2)$ time).

Option 2: $S = A^T$

Now suppose we choose $S = A^T$.

$$\begin{aligned}
\tilde{x} &= (SA)^+ Sb \\
&= (A^T A)^+ A^T b \\
&= V \Sigma^{-2} V^T V \Sigma U^T b \\
&= V \Sigma^{-1} U^T b \\
&= x_{LS}
\end{aligned}$$

So again we've recovered the least squares solution exactly, but we wanted to avoid the cost of computing $A^T A$ in the first place (which, again, takes $O(nd^2)$ time).