Randomized Algorithms 2021A – Lecture 10 (second part)
Regression via OSE, Importance Sampling*

Robert Krauthgamer

1 Least Squares Regression

Problem definition: In *Least Squares Regression*, the input is a matrix $A \in \mathbb{R}^{n\times d}$ and a vector $b \in \mathbb{R}^n$, and the goal is to find $\operatorname{argmin}\{\|Ax^* - b\| : x^* \in \mathbb{R}^d\}$.

Informally, when solving a system $Ax^* = b$ that is over-constrained ($n \gg d$), we do not expect to find an exact solution, and we want to minimize the sum of squared errors $\sum_i (A_i x^* - b_i)^2$.

We shall consider $(1+\varepsilon)$-approximation, i.e., finding $x' \in \mathbb{R}^d$ such that

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

Theorem: Let $S \in \mathbb{R}^{s\times n}$ be an $(\varepsilon, \delta, d+1)$-OSE matrix. Then for every regression instance $A \in \mathbb{R}^{n\times d}$ and $b \in \mathbb{R}^n$, with high probability, an optimal solution x' (or even $(1+\varepsilon)$-approximation) to the regression instance $\langle SA, Sb\rangle$ is a $(1+O(\varepsilon))$-approximation to the instance $\langle A, b\rangle$, i.e., such x' satisfies (1).

This theorem essentially reduces a regression problem with n constraints to regression with s constraints, but we should take into account also the time to compute SA.

Proof: As explained in class, it follows from applying the OSE guarantee to the linear subspace spanned by the columns of A and by b (total of $d+1$ vectors), and then

$$(1-\varepsilon)\|Ax' - b\| \le \|SAx' - Sb\| = \min_{x\in\mathbb{R}^d} \|SAx - Sb\| \le (1+\varepsilon) \min_{x^*\in\mathbb{R}^d} \|Ax^* - b\|.$$

2 Importance sampling

It's a tool to reduce variance when sampling. The idea is to sample, instead of uniformly, in a "focused" manner that roughly imitates the contributions, and then "factor out" the bias in this sample.

*These notes summarize the material covered in class, usually skipping proofs, details, examples and so forth, and possibly adding some remarks, or pointers. The exercises are for self-practice and need not be handed in. In the interest of brevity, most references and credits were omitted.

Setup: We want to estimate $z = \sum_{i \in [s]} z_i$ without reading all the z_i values. The main concern is that the z_i are unbounded, and thus most of the contribution might come from a few unknown elements, but we have a "good" lower bound on each element, intuitively $p_i \approx \frac{z_i}{z}$.

Theorem 1 [Importance Sampling]: Let $z = \sum_{i \in [s]} z_i$, and $\lambda \geq 1$. Let $\hat{Z}$ be an estimator obtained by sampling a single index $\hat{i} \in [s]$ according to distribution $(p_1, \ldots, p_n)$ where $\sum_{i \in [s]} p_i = 1$ and each $p_i \geq \frac{z_i}{\lambda z}$, and setting $\hat{Z} = z_{\hat{i}}/p_{\hat{i}}$. Then

$$\mathbb{E}[\hat{Z}] = z \quad \text{and} \quad \sigma(\hat{Z}) \leq \sqrt{\lambda}\,\mathbb{E}[\hat{Z}].$$

Proof: was seen in class.

Exer: Show that averaging $O(\lambda/\varepsilon^2)$ independent repetitions of the above approximates z within factor $1 \pm \varepsilon$ with success probability at least $3/4$.

Hint: use Chebyshev's inequality.

Exer: Prove a variant of Theorem 1, where each z_i is read independently with probability $q_i \geq \min\{1, t\frac{z_i}{z}\}$, in which case it contributes $\frac{z_i}{q_i}$ (and otherwise contributes 0). Show that with high probability, the number of values read is $O(\sum_i q_i)$ and the estimate is $(1 \pm O(1/\sqrt{t}))z$.

Hint: The difference is here we read each z_i independently, while in Theorem 1 we see in each step exactly one value (the value of z_i with probability p_i).

Exer: Let $z = \sum_{i \in [s]} z_i$ and suppose that for each z_i we already have an estimate within factor $b \geq 1$, i.e., some $z_i \leq y_i \leq b z_i$. How many z_i values we need to sample/read into order to estimate z within factor $1 \pm \varepsilon$ (with success probability at least $3/4$)?

Learn the next section for next class

3 Counting DNF solutions via Importance Sampling

Problem definition: The input is a DNF formula f with m clauses $C_1, \ldots, C_m$ over n variables $x_1, \ldots, x_n$, i.e., $f = \vee_{i=1}^m C_i$ where each C_i is the conjunction of literals like $x_2 \wedge \bar{x}_5 \wedge x_n$.

The goal is the estimate the number of Boolean assignments that satisfy f.

Theorem 2 [Karp and Luby, 1983]: Let $S \subset \{0,1\}^n$ be the set of satisfying assignments for f. There is an algorithm that estimates $|S|$ within factor $1 + \varepsilon$ in time that is polynomial in $m + n + 1/\varepsilon$.

3.1 A first attempt

Random assignments: Sample t random assignments, and let Z count how many of them are satisfying. We can estimate $|S|$ by $Z/t \cdot 2^n$.

Formally, we can write $Z = \sum_{i=1}^{t} Z_i$ where each Z_i is an indicator for the event that the i-th sample satisfies f. Then $Z = \frac{1}{t}\sum_i (Z_i \cdot 2^n)$. We can see it is an unbiased estimator:

$$\mathbb{E}[Z \cdot 2^n/t] = \sum_{i=1}^{t} \mathbb{E}[Z_i] \cdot 2^n/t = |S|.$$

Observe that $\mathrm{Var}(Z) = \frac{1}{t^2}\sum_i \mathrm{Var}(Z_i \cdot 2^n) = \frac{1}{t}\mathrm{Var}(Z_1 \cdot 2^n)$. But even though we can use Chernoff-Hoeffding bounds since Z_i are independent, it's not very effective because the variance could be exponentially large.

Exer: Show that the standard deviation of Z (for $t = 1$) could be exponentially large relative to the expectation.

3.2 A second attempt

Idea: We can bias the probability towards the assignments that are satisfying, but then we will need to "correct" the bias.

Let $S_i \in \{0,1\}^n$ be all the assignments that satisfy the i-th clause, hence $|S_i| = 2^{n-\mathrm{len}(C_i)}$.

Remark: The naive approach does not use the DNF structure at all. We can use this structure by writing $S = \cup_i S_i$, which can be expanded using the inclusion-exclusion formula, but it would be too complicated to estimate efficiently.

Algorithm E:

1. Choose a clause C_i with probability proportional to $|S_i|$ (namely, $|S_i|/M$ where $M = \sum_i |S_i|$).
2. Choose at random an assignment $a \in S_i$.
3. Compute the number y_a of clauses satisfied by a.
4. Output $Z = \frac{M}{y_a}$.

Exer: Prove the following two claims.

Claim 2a: $\mathbb{E}[Z] = |S|$.

Claim 2b: $\sigma(Z) \le m \cdot \mathbb{E}[Z]$.

Exer: Show that $|S|$ can be approximated within factor $1 \pm \varepsilon$ with success probability at least 3/4, by averaging $O(m^2/\varepsilon^2)$ independent repetitions of the above.

Exer: Show how to improve the success probability to $1-\delta$ by increasing the number of repetitions by an $O(\log \frac{1}{\delta})$ factor.

Exer: Explain this DNF counting algorithm using the importance sampling theorem.

Hint: Assignments a that satisfy no clause are chosen with zero probability.