Randomized Algorithms 2025A – Lecture 12*
Importance Sampling, Counting DNF solutions, and Coresets for Clustering

Robert Krauthgamer

1 Importance Sampling

Sampling is often used to estimate a sum. When the variance is too large, this method can reduce the variance, by sampling not uniformly, but rather in a biased manner that roughly imitates the contributions, while making sure to "factor out" the bias.

Setup: We want to estimate $z = \sum_{i\in[n]} z_i$ 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. If we happen to have a "good enough" lower bound on each element z_i, then we can sample with probability $p_i \geq \Omega(\frac{z_i}{z})$.

Theorem 1 [Importance Sampling]: Let $z = \sum_{i\in[n]} z_i$, and $\lambda \geq 1$. Let $\hat{Z}$ be an estimator obtained by sampling a single index $\hat{i} \in [n]$ according to distribution $(p_1, \ldots, p_n)$ where $\sum_{i\in[n]} p_i = 1$ 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 $t = O(\lambda/\varepsilon^2)$ 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 decide whether to read each z_i independently, while in Theorem 1 we read at each step exactly one value z_i.

*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.

1.1 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 $m + n + 1/\varepsilon$.

1.2 Counting: 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. We can easily see it is an unbiased estimator:

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

Observe that its standard deviation is $\sigma(Z/t \cdot 2^n) = \sigma(Z_1 \cdot 2^n)/\sqrt{t}$. 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 larger than the expectation.

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

1.3 Counting: 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 $|S_i| = 2^{n-\text{len}(C_i)}$.

Remark: Our goal is to estimate $|S| = |\cup_i S_i|$. We can expand this 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 $Z = \frac{M}{y_a}$.

Claim 2a: $\mathbb{E}[Z] = |S|$ (i.e., this algorithm's output is unbiased).

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

The proofs are straightforward and were seen in class.

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.

Hint: use Chebyshev's inequality.

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: Think what is the relative contribution of each assignment $\hat{a}$ to $|S|$.

2 Coresets for Clustering

Let $\text{dist}(\cdot,\cdot)$ denote the Euclidean distance in $\mathbb{R}^d$, including distance between a point and set $\text{dist}(x, C) = \min_{c \in C} \|x - c\|$.

Geometric Clustering: In the *k-median problem* the input is a set of n data points $X = \{x_1, \ldots, x_n\} \subset \mathbb{R}^d$, and the goal is to find a set of k centers $C = \{c_1, \ldots, c_k\} \subset \mathbb{R}^d$ that minimizes the objective function

$$f(X, C) := \sum_{x \in X} \text{dist}(x, C) = \sum_{i \in [n]} \min_{j \in [k]} \|x_i - c_j\|_2.$$

Note that the centers are not required to be from X (the version with this requirement is called discrete centers or k-medoid).

The *k-means problem* is similar but using squared distances.

Notation: We shall omit the subscript from all norms, as we always use ℓ_2 norms.

Observe that points need not be distinct, i.e., we consider multisets, which is equivalent to giving every point an integer weight, and admits a succinct representation. We thus would like to reduce the number of *distinct* points, denoted throughout by $|X|$.

Strong Coreset: Let $\epsilon \in (0, 1/2)$ be an accuracy parameter. We say that $S \subset \mathbb{R}^d$ is a strong ε-coreset of X (for objective f, which in our case is k-median) if

$$\forall C = \{c_1, \ldots, c_k\} \subset \mathbb{R}^d, \qquad f(S, C) \in (1 \pm \varepsilon) f(X, C).$$

Note: A weak coreset is similar, except the above requirement is only for the optimal centers for the coreset, i.e., C' that minimizes $f(S, C')$.

Goal: We want to construct small coresets. If done without computing an optimal solution C^*, then it would be useful for computing a near-optimal solution, because it suffices to solve k-median

on the smaller instance S. If the construction requires computing C^*, it could still be useful when sending (communicating) or storing the data.

We focus henceforth on existence (of coresets of a certain size), the algorithmic implementation and applications are usually straightforward.

2.1 Coresets via Geometric Decomposition

Theorem 3 [Har-Peled and Mazumdar, 2004: Every set X of n points in $\mathbb{R}^d$ admits an ε-coreset S of cardinality $|S| = O(k(9/\varepsilon)^d \log n)$.

Idea: Discretize the space to create a small set $\hat{S}$, and "snap" every point in X to its nearest neighbor in $\hat{S}$. Throughout, the (closed) ball of radius $r > 0$ about $c \in \mathbb{R}^d$ is defined as

$$B(c, r) = \{z \in \mathbb{R}^d : \|z - c\| \leq r\}.$$

Lemma 3a (ε-Ball Cover): For every $\varepsilon \in (0, 1)$, the unit ball $B = B(\vec{0}, 1)$ in $\mathbb{R}^d$ can be covered by $(3/\varepsilon)^d$ balls of radius ε.

The conclusion is that every point in the unit ball can be "approximated" by one of those $(3/\varepsilon)^d$ centers, with additive error ε. This argument immediately extends to a ball of radius $r > 0$, except that the discretization error is now εr.

We already saw a proof of this lemma (for the unit sphere instead of unit ball, but the proof is the same).

Proof of Theorem 3: Was seen in class.

Exer: Modify the above proof to be algorithmic, by using an $O(1)$-approximation to the minimum cost (meaning a set C' such that $f(X, C') \leq O(1) \cdot f(X, C^*)$), which can be computed in polynomial time.

Exer: Extend this argument to k-means using the following generalized triangle inequality: For every $a, b, c \in \mathbb{R}^d$ and $\varepsilon \in (0, 1)$,

$$\left| \|a - c\|^2 - \|b - c\|^2 \right| \leq \tfrac{12}{\varepsilon} \|a - b\|^2 + 2\varepsilon \|a - c\|^2.$$