Randomized Algorithms 2019A – Lecture 7
Importance Sampling and Coresets for Clustering*

Robert Krauthgamer

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

1.1 A first attempt

Random assignments: Sample t random assignments, and let Z count how many of them are satsifying. 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_1 (and thus Z) could be exponentially large relative to the expectation.

*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.

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

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 $Z = \frac{M}{y_a}$.

We proved in class the following two claims.

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

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

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.

1.3 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.

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 3 [Importance Sampling]: Let $z = \sum_{i \in [s]} z_i$, and $\lambda \geq 1$. Let $\hat{Z}$ be an estimator computed by sampling a single index $i \in [s]$ with probability p_i and setting $\hat{Z} = z_i/p_i$, where each $p_i \geq \frac{z_i}{\lambda z}$ and $\sum_{i \in [s]} p_i = 1$. 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: 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 $z_i \leq y_i \leq bz_i$. How many samples are needed to compute, with probability at least 3/4, a $1 \pm \varepsilon$ factor estimate for z?

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

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

2 Coresets for Clustering

Let $D(\cdot,\cdot)$ denote the Euclidean distance in $\mathbb{R}^d$.

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

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

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

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

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

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 Geometric Decomposition

Idea: Discretize the space to create a small set $\hat{S}$, and "snap" every point in X to its nearest neighbor in 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 4 (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 any ball of radius $r > 0$, except that the additive error is now εr.

Exer: Prove this lemma.

Hint: Construct the covering iteratively, and use the volume estimate $\mathrm{vol}(B(c, r)) = r^d \cdot \mathrm{vol}(B(\vec{0}, 1))$.

Theorem 5: Every set X of n points in $\mathbb{R}^d$ admits an ε-coreset S of cardinality $|S| = O(k(9/\varepsilon)^d \log n)$.

Proof: 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 $\varepsilon \in (0, 1)$,

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