COMP 480/580 — Probabilistic Algorithms and Data Structure March 21, 2018

Lecture 17

Lecturer: Anshumali Shrivastava *Scribe By: Shabnam Daghaghi and Edward Feng*

1 Sampling in High Dimensions

In the previous lectures, we saw that rejection sampling methods fail in high dimensions due to the curse of dimensionality. For instance, suppose we want to estimate the area of the irregular shape a in Figure **??**. By randomly sampling from area S, we can estimate the area of a using the same procedure we used in the previous lectures to compute the area of a circle. The proportion of samples from area a to the samples from square S is (in expectation) equivalent to the proportion of the area of these two regions. But this method does not work in high dimensions. One odd result in high-dimensional spaces is that most of the volume of a sphere or cube is concentrated at the edge of the shape. In high-dimensional spaces, $\frac{a}{A} \to 0$. In other words, by sampling from area A, we almost never get any samples that are in area a, leading to a terribly inefficient sampling procedure. This counterintuitive behavior is often referred to as the curse of dimensionality.

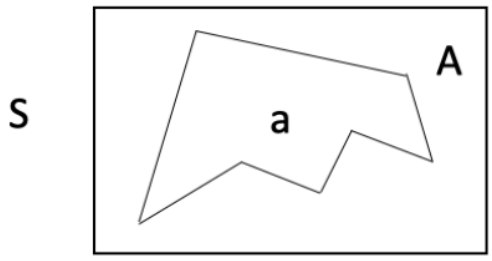

Figure 1: In low dimensions, it is possible to efficiently estimate the area or volume of a. However, in high dimensions the problem is much more difficult to solve via sampling

Monte Carlo offers a solution for sampling in higher dimensions. We generate N samples $x_1, x_2, ..., x_N$ uniformly from A (area in S). Assume I_i is the indicator variable where $I_i = 1$ if x_i is a sample from the region of interest (a). It is clear that $E(I_i) = \frac{a}{A} = (\frac{1}{factor})^d$, and for high dimensions this value decreases exponentially in dimension.

Lemma: Let $I_1, I_2, ..., I_N$ be *iid* indicator variables and $\mu = E(I_i)$, then one can show (using the Chernoff bound) that $Pr(\frac{1}{N}\sum_{i=1}^{N} I_i - \mu \geq \epsilon\mu) \leq \delta$ if $N \geq \frac{3\ln\frac{2}{\delta}}{\epsilon^2\mu}$.

In other words, we require a number of samples that is exponential in the number of dimensions to have a given failure probability. This naive approach requires $O(\frac{A}{a})$ or equivalently $O(\frac{1}{\mu})$ or $O(factor^d)$ samples.

Now consider Figure **??**, where the aim is to compute the volume of the irregular shape K. Further, suppose that K and the sphere lie in a high-dimensional space. It is possible to sample

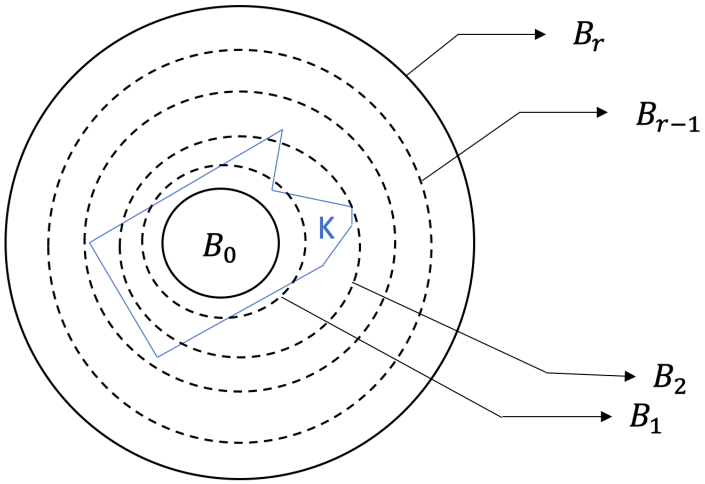

Figure 2: Sampling polynomial in dimension

uniformly from a d-ball, even in high dimensions. Therefore, to efficiently drawe samples, we consider a set of balls around the region K. Our approach is to compute the volume of K by partitioning the ball into sub-balls. Essentially, we create a set of balls and approximate the volume of K intersected with each ball in our set. We gradually increase the radius of the ball until we cover the entirety of K.

$$Vol(K) = Vol(K \cap B_r) = \frac{Vol(K \cap B_r)}{Vol(K \cap B_{r-1})} \times \frac{Vol(K \cap B_{r-1})}{Vol(K \cap B_{r-2})} \times .... \times \frac{Vol(K \cap B_1)}{Vol(K \cap B_0)} \times Vol(B_0) \quad (1)$$

To solve equation (**??**), $Vol(B_r)$ and $Vol(B_{r-1})$ should not differ a lot, otherwise the first issue, curse of dimensionality, appears again. With this assumption each term (fraction) in (**??**) can be estimated by sampling. This assumption is formalized in equation (**??**). If (**??**) is valid, $Vol(B_r)$ and $Vol(B_{r-1})$ are comparable even in high dimensions.

$$rad_{B_r} = \left(1 + \frac{1}{d}\right) rad_{B_{r-1}} \quad (2)$$

A natural question is: how many balls are needed? We do not want the number of balls to be large, as we do not wish to use Monte Carlo to estimate too many ratios, but we need sufficiently small gaps between the balls for the number of Monte Carlo samples to be relatively low. We can balance these requirements and find that a good choice for the radius of ball r (B_r) is $\text{rad}(B_r) = (1 + \frac{1}{d})\text{rad}(B_{r-1})$. This choice of radius does not cause large gaps between consecutive spheres while keeping the number of ratios relatively small. Under this scheme, the sampling is $O(d^4)$, which is only polynomial in dimension. This is a substantial improvement over the first approach, which was exponential in dimension. This concludes our discussion of sampling in high dimensions.

2 DNF Counting

2.1 Disjunctive Normal Forms (DNF)

In this section, we switch to a review of some basic results in logic in preparation for our next problem. A *disjunctive normal form* (DNF) is a boolean expression that is structured as an OR of clauses $C_1 \vee C_2 \vee ... \vee C_m$, where each clause C_i is an AND of literals $X_1 \wedge X_2 \wedge X_3 \wedge ... \wedge X_d$. For example, consider the following DNF.

$$(x_5 \wedge \bar{x_4} \wedge x_3) \vee (x_1 \wedge x_2 \wedge x_4) \vee (\bar{x_{18}} \wedge x_{48}) \vee ... \vee (x_{72} \wedge x_{45}) \tag{3}$$

It is clear that if any one of the clauses in DNF is satisfied, the whole DNF is satisfied. Therefore, we can easily find an assignment of the literals that satisfies the DNF - we simply have to find an assignment to satisfy one of the clauses. We will refer to an assignment that satisfies the DNF as a satisfiability assignment. Suppose we wish to count the number of satisfiability assignments. Since we know the total number of possible assignments is 2^n - each of the n literals can be either true or false, this problem is hard.

It should be noted that conjunctive normal forms (CNF)[1] have the opposite situation - finding a satisfiability assignment is difficult (NP hard), but determining the number of satisfiability assignments is easy.

2.2 Monte Carlo for DNFs

We start with a naive approach to counting the number of satisfiability assignments for a DNF.

Algorithm 1 Naive Approach: randomized Monte Carlo based solution to count number of satisfiability assignments for a DNF

Input: DNF
Output: Number of satisfiability assignments for given DNF
$Count \leftarrow 0$
for i= 1 : M **do**
  Generate random assignment s_i
  if *random assignment satisfies DNF* **then**
    $Count$++
return $(\frac{Count}{M})2^n$

In this Monte Carlo scheme, the estimated count should be the fraction of satisfying assignments in the sample multiplied by 2^n. To see this, note that the expected value of the indicator variable is the number of satisfiability assignments over 2^n: $\mathbb{E}(I_i) = \frac{N(s)}{2^n}$. Therefore, the number of samples needed is $\frac{1}{\mathbf{E}(I_i)} = \frac{2^n}{N(s)}$. This is not desirable because $N(s)$ is usually small in comparison with 2^n. In the above algorithm, the proposal and target are not comparable.

2.3 Work-Around

In order to solve this issue, we define a universal set: $U = \{(i, a); a \in SC_i\}$ where SC_i includes the satisfiability assignments that satisfy clause i in DNF. Note that $N(s) = \bigcup_i SC_i$ and does not include repeated terms (common terms in SC_is). However U has the repetitions, therefore $|U| = \sum_i |SC_i| \leq |N(s)|$. If $|U|$ is not much bigger that $|N|$ and can be sampled randomly,

[1] Conjunctive normal forms are the "opposite" of DNFs - they are ANDs of OR clauses

then the number of satisfiability assignments can be estimated efficiently. Of course, we must be able to sample U uniformly. This will be discussed in the next lecture.