(CE7456 Lecture Notes 3) Set Union Sampling

Yufei Tao

Let us start with a rudimentary lemma (recall that if $x \geq 1$ is an integer, $[x]$ represents the set $\{1, 2, ..., x\}$):

Lemma 1 (Probability through Expectation)**.** *Fix any integer $m \geq 1$. Let X be a random variable taking values from $[m]$, and Y be a uniformly random variable over $[m]$. If X and Y are independent, then $\mathbf{Pr}[Y \leq X] = \frac{1}{m}\mathbf{E}[X]$.*

Proof.

$$\begin{aligned}
\mathbf{Pr}[Y \leq X] &= \sum_{x=1}^{m}\sum_{y=1}^{x}\mathbf{Pr}[Y = y] \cdot \mathbf{Pr}[X = x] \\
&= \sum_{x=1}^{m}\sum_{y=1}^{x}\frac{1}{m} \cdot \mathbf{Pr}[X = x] \\
&= \frac{1}{m}\sum_{x=1}^{m} x \cdot \mathbf{Pr}[X = x] \\
&= \frac{1}{m}\mathbf{E}[X].
\end{aligned}$$

□

We will witness the lemma's power by using it to solve the *set union sampling* problem:

Set Union Sampling: Let $S_1, S_2, ..., S_m$ be $m \geq 2$ sets of elements drawn from a certain domain. Each S_i ($i \in [m]$) supports three operations in constant time:

1. (*size*) return $|S_i|$;
2. (*membership*) check whether a given element is in S_i;
3. (*sampling*) return an element of S_i chosen uniformly at random.

The goal is to sample an element from $\bigcup_{i=1}^{m} S_i$ uniformly at random using only these operations. The sample obtained each time must be independent of all previous samples.

1 A Slow Algorithm

Define

$$\begin{aligned}
n &= |\bigcup_{i=1}^{m} S_i| \\
N &= \sum_{i=1}^{m} |S_i|.
\end{aligned}$$

For each element $y \in \bigcup_{i=1}^{m} S_i$, define its *inverted set* as $\text{InvS}(y) = \{i \in [m] \mid y \in S_i\}$, i.e., the "ids" of the sets containing y. The *degree* of y is

$$\deg(y) = |\text{InvS}(y)|.$$

Algorithm. Draw a random value $X \in [m]$ such that $\mathbf{Pr}[X = i] = |S_i|/N$ for each $i \in [m]$. Then, use the sampling operation to draw an element Y from S_X. Finally, carry out an *acceptance step*:

Acceptance Step: Accept Y with probability $1/\deg(Y)$.

If accepted, Y is returned; otherwise, the algorithm repeats from scratch.

The algorithm correctly returns a uniform sample of $\bigcup_{i=1}^{m} S_i$. To see why, fix an arbitrary element $y \in \bigcup_{i=1}^{m} S_i$. This element is returned if and only if three conditions are satisfied:

- $X \in \text{InvS}(y)$, which occurs with probability $|S_X|/N$ for each $X \in \text{InvS}(y)$;
- $Y = y$, which occurs with probability $1/|S_X|$ conditioned on X;
- y is accepted, which occurs with probability $1/\deg(y)$ conditioned on y.

Hence, the algorithm outputs y with probability

$$\sum_{X \in \text{InvS}(y)} \frac{|S_X|}{N} \cdot \frac{1}{|S_X|} \cdot \frac{1}{\deg(y)} = \frac{1}{N} \tag{1}$$

which is identical for all $y \in \bigcup_{i=1}^{m} S_i$. As a corollary, in each repeat, the algorithm *succeeds* in returning a sample with probability n/N. Thus, N/n repeats are needed in expectation.

Running Time. The random value X can be easily obtained in $O(m)$ time. One way to do so is to build an alias structure on $|S_1|, |S_2|, ..., |S_m|$ — namely, the m set sizes — on the fly in $O(m)$ time (using the size operation), after which X can be extracted from the structure in $O(1)$ time. The time to obtain Y is $O(1)$ (using the sampling operation). The troublemaker, however, is the acceptance step. A simple approach is to query the membership of Y in each S_i ($i \in [m]$), which costs $O(m)$ time. This renders the overall sample time $O(mN/n)$ in expectation.

We will refer to the above algorithm as the *baseline* method.

2 An Optimal Algorithm

Our objective is to implement the acceptance step in $O(mn/N)$ expected time — note that this is faster than $O(m)$ by a factor of N/n, which eventually allows us to bring the expected sample time from $O(mN/n)$ down to $O(m)$.

New Idea: Probability through Expectation. By Lemma 1, implementing the acceptance step boils down to:

Given $x \in [m]$ and $y \in S_x$, generate a random variable $\Gamma \in [m]$ with $\mathbf{E}[\Gamma] = m/\deg(y)$.

Later, we will explain how to achieve the above purpose in $O(m/\deg(y))$ expected time. Once done, we can perform the acceptance step as follows (recall that, prior to this, the baseline method in Section 1 has obtained the values of two random variables X and Y):

- (i) generate a uniformly random variable U over $[m]$ in constant time;
- (ii) generate the aforementioned random variable Γ (setting x and y to the values of X and Y, respectively) in $O(m/\deg(Y))$ expected time;
- (iii) accept if $U \leq \Gamma$.

The acceptance probability is $\frac{1}{m}\mathbf{E}[\Gamma] = 1/\deg(Y)$, as desired.

For a particular $x \in [m]$ and a particular $y \in S_x$, we have $\mathbf{Pr}[X = x, Y = y] = \frac{|S_x|}{N}\frac{1}{|S_x|} = 1/N$. Thus, the expected cost of the acceptance step will be at the order of

$$\begin{aligned}\sum_{x\in[m]}\sum_{y\in S_x}\mathbf{Pr}[X=x,Y=y]\frac{m}{\deg(y)} &= \sum_{x\in[m]}\sum_{y\in S_x}\frac{1}{N}\frac{m}{\deg(y)} \\ &= \frac{m}{N}\sum_{y\in\cup_{i=1}^{m}S_i}\sum_{x\in\mathrm{InvS}(y)}\frac{1}{\deg(y)} \\ &= \frac{mn}{N} \qquad (2)\end{aligned}$$

where the last step used $\deg(y) = |\mathrm{InvS}(y)|$ and $n = |\bigcup_{i=1}^{m} S_i|$.

Generation of Γ. Next, we will concentrate on the Γ-generating task defined earlier. Recall that the generation is based on a given set "id" $x \in [m]$ and an element $y \in S_x$. Consider the procedure below:

find-2nd (x, y)
1. $F \leftarrow 0$ and $I \leftarrow [m] \setminus \{x\}$
2. **while** $I \neq \emptyset$ **do**
3. sample without replacement (WoR) an integer $i \in I$
   /* note: the WoR-sampling operation removes i from I chosen uniformly at random */
4. **if** $y \in S_i$ **then return** F **else** increase F by 1
5. **return** F /* note: F must be $m - 1$ here */

In plain words, if $\deg(y) \geq 2$, the value F returned is a random variable giving the number of *failed* WoR-sampling operations before finding *another* set $S_i \neq S_x$ containing y; otherwise, $F = m - 1$.

Proposition 1. $\mathbf{E}[F] = \frac{m}{deg(y)} - 1$.

Proof. If $\deg(y) = 1$, then F is deterministically $m - 1$, in which case the lemma clearly holds.

The rest of the proof considers $\deg(y) \geq 2$. In that scenario, F can be rephrased in the following conventional WoR-sampling setup. Suppose that we have $m-1$ balls, among which $\deg(y)-1$ ones are red and the rest are white. Uniformly sample a ball WoR until seeing a red ball, and F gives the number of white balls sampled in this process. It is well known (e.g., see [2]) that F follows the Negative Hypergeometric distribution with $\mathbf{E}[F] = \frac{m}{\deg(y)} - 1$. □

Combining the above with the obvious fact that $F \in [0, m - 1]$, we can now define

$$\Gamma = F + 1$$

as the desired random variable satisfying $\Gamma \in [m]$ and $\mathbf{E}[\Gamma] = m/\deg(y)$. The WoR-sampling operation at Line 3 can be implemented in $O(1)$ time (think: how?). Thus, the cost of **find-2nd**

is proportional to the value F returned. It follows from Proposition 1 that the expected cost of **find-2nd** is $O(\mathbf{E}[F]) = O(m/\deg(y))$.

Total Time of Set-Union Sampling. Recall that, in the first algorithm (Section 1), each repeat takes $O(m)$ time; as the number of repeats is N/n expected, the total cost of taking one sample from $\bigcup_{i=1}^{m} S_i$ is $O(mN/n)$. By applying our remedy, one repeat of the baseline algorithm is now carried out in $O(mn/N)$ expected time (see the analysis in (2)). The remedy does not affect the expected number of repeats (i.e., N/n), which suggests that the overall expected sample time should be $O(\frac{mn}{N} \cdot \frac{N}{n}) = O(m)$. Indeed, if $\mathbf{E}[T_{total}]$ represents the expected time for our algorithm to acquire a sample from $\bigcup_{i=1}^{m} S_i$, we can write

$$\mathbf{E}[T_{total}] = O(mn/N) + (1 - n/N)\,\mathbf{E}[T_{total}]$$

which solves to $\mathbf{E}[T_{total}] = O(m)$.

We thus have proved that we can sample from $\bigcup_{i=1}^{m} S_i$ in $O(m)$ expected time. This is optimal (in the expected sense) because clearly at least one operation needs to be performed on each S_i ($i \in [m]$). (Think: why?)

3 Remark

The algorithm in Section 2 was developed in [1].

References

[1] Jinchao Huang, Yufei Tao, and Sibo Wang. Acyclic join sampling under selections: Dichotomy, union sampling, and enumeration. In *ICDT*, 2026.

[2] Norman L Johnson, Adrienne W Kemp, and Samuel Kotz. *Univariate Discrete Distributions*. John Wiley & Sons, 2005.