Crypto Engineering
Discrete probability

2023-09-19

Exercise 1: (multi-)collisions

In all of this exercise we let $\mathcal{S}$ be an arbitrary finite set of size N, and we denote by $X \twoheadleftarrow \mathcal{S}$ the process of drawing a random variable X from $\mathcal{S}$ uniformly at random, and independently of any other process.

Let $X \twoheadleftarrow \mathcal{S}$, $Y \twoheadleftarrow \mathcal{S}$, $Z \twoheadleftarrow \mathcal{S}$.

1. Compute $\Pr[(X = x) \wedge (Y = y)]$ for any fixed $x, y \in \mathcal{S}$.
2. Compute $\Pr[X = Y]$.
3. Compute $\Pr[X = Y = Z]$.

Exercise 2: (non-)uniform masks

Let X and Y be two independent random variables drawn from $\mathbb{F}_2$ with a uniform law for X and an unknown arbitrary law for Y.

1. What is the distribution of $X + Y$? (That is, compute $\Pr[X + Y = 0]$)

We now draw X and Y independently from a finite commutative group $(\mathbb{G}, +)$ of size N.

2. What is (again) the distribution of $X + Y$? (Note that the distribution of $X + Y$ is given here by the discrete convolution of the distributions of X and Y).

Remark. The result shown in those two questions is essential in cryptography, and is used to justify the security of many constructions.

We go back to X and Y being drawn independently over $\mathbb{F}_2$, but consider this time arbitrary laws for both of them. We write c_X the *correlation bias* of X defined as $c_X = |2\Pr[X = 0] - 1|$, and the same for c_Y.

3. Compute c_{X+Y}, the correlation bias of $X + Y$.
4. By induction, give a formula for the correlation bias of the sum $X_1 + \cdots + X_N$ of N independent variables of correlation biases $c_1, \ldots, c_N$.

Remark. This last result is known in (symmetric) cryptography as the *piling-up lemma*.

We go back to X and Y being uniform from a finite commutative group (and still take them to be independent), and let $Z := X + Y$.

5. Show that X and Z are independent, but that X, Y, Z are not mutually independent.

Remark. One may show the general result that if $\vec{X}$ is a vector of n mutually independent *uniform* random variables over a finite field $\mathbb{F}_q$, $\boldsymbol{M} \in \mathbb{F}_q^{k\times n}$, $k \leq n$, then the n random variables in $\vec{Y} := \vec{X}\boldsymbol{M}$ are mutually independent iff. $\boldsymbol{M}$ is of full rank. This result is at the core of the construction of *linear secret sharing schemes.*

Exercise 3: For my birthday I got a coupon for a pair of socks

Let again $\mathcal{S}$ be an arbitrary finite set of size N, which we sample repeatedly by drawing $X_1, \ldots, X_q$ uniformly and independently. A (non-trivial) *collision* for those random variables is a pair $(X_i, X_{j\neq i} = X_i)$.

Q.1 (*Pigeonhole principle*, or *lemme des chaussettes*): How many samples q are necessary to ensure (with probability 1) that there is *at least one* collision among $X_1, \ldots, X_q$?

Q.2 (*Birthday paradox*):

1. Compute the probability p^q_{unq} that there are *no* collisions among $X_1, \ldots, X_q$.
2. Using the union bound, give an upper bound for $p^q_{\text{col}} := 1 - p^q_{\text{unq}}$, the probability that there *is* a collision.

   Hint: Introduce some new random variables $C_{i,j}$ that indicate if their corresponding pair (X_i, X_j) forms a collision.
3. Compute the expected number of collisions in function of q.

   Hint: Use the linearity of expectations.

Remark. By suitably upper-bounding p^q_{unq}, one may show that for small enough values of q, $p^q_{\text{col}} \geq q(q-1)/4N$, cf. https://membres-ljk.imag.fr/Bruno.Grenet/IntroCrypto/BirthdayBounds.pdf.

Q.3 (*Coupon collector's problem, cf. Figure 1*):

1. For all $\alpha \in \mathbb{R}$, $\alpha > 1$, compute an upper-bound on the number of samples q necessary to ensure that the probability that there is some a in $\mathcal{S}$ s.t. none of the X_i's evaluated to a (i.e. the probability that not all coupons were collected) is less than $1/\alpha$.

   Hint: Apply the union bound to suitable random variables, and use $(1 - 1/N)^{kN} \leq e^{-k}$ (for $k > 1$).
2. Compute the expected number of samples q needed to collect all coupons.

   Hint: Use the linearity of expectations and the fact that the number of samples needed to pick a new coupon after k have been collected follows a geometric distribution of parameter $\frac{n-k}{n}$.

Exercise 4: (close-to) uniform permutations ⋆

We consider the following algorithm to generate a random permutation of $[\![1, N]\!]$ (or more generally, of N arbitrary elements): 1) build a list of N pairs (r_i, i), where $r_i \twoheadleftarrow \mathbb{Z}/q\mathbb{Z}$; 2) sort the list according to the first element of the pairs; 3) return the list of the second element of the pairs in the sorted order.

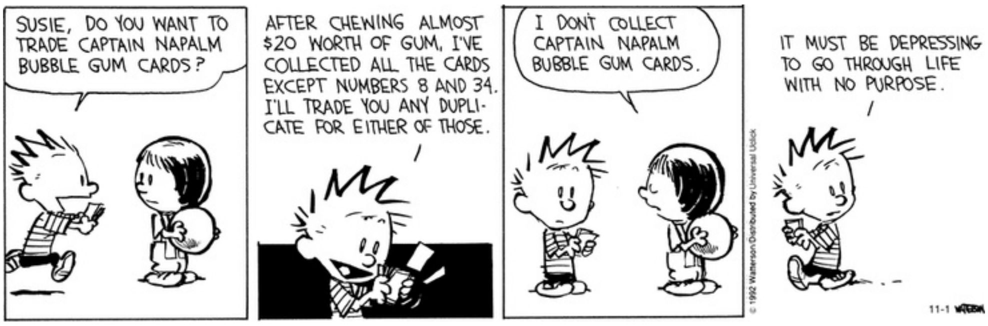

Figure 1: The coupon collector's problem: a Calvin & Hobbes illustration

Q.1 : Compute the number of sorted lists of N elements of $\mathbb{Z}/q\mathbb{Z}$.
Hint: Map all such possible lists to paths from $(0, 1)$ to (N, q) in the 2-dimensional discrete grid, where only horizontal and vertical steps are allowed.

Q.2 :

1. For every possible permutation generated by the algorithm, compute a non-trivial upper-bound for the number of drawings for $(r_1, \ldots, r_N)$ that lead to it.
2. What is then an upper-bound for the probability of occurence of any permutation?
3. Express this probability as $\delta/N!$ for δ of the form $\prod_{i=1}^{N-1}(1 + x_i/q)$.
4. For a fixed N, give an approximative criterion on q for δ to be close to 1 (for instance using the approximation (for "large" x) $\left(1 + \frac{1}{x}\right)^x \approx e$).

We now consider a variant of the algorithm, where one is interested in drawing a random combination of weight w. This is done as follows: 1) build a list of N pairs $(r_i, [i \leq w])$, where $r_i \twoheadleftarrow \mathbb{Z}/q\mathbb{Z}$ (and $[i \leq w]$ is 1 if $i \leq w$, and 0 otherwise); 2) sort the list according to the first element of the pairs; 3) return the list of the second element of the pairs in the sorted order.

Q.3 :

1. For every possible combination generated by the algorithm, compute a non-trivial upper-bound for the number of drawings for $(r_1, \ldots, r_N)$ that lead to it.

   Hint: For any fixed combination, count the number of permutations that lead to it. (This can be counted as the number of permutations that leave a given combination invariant.)
2. What is then an upper-bound for the probability of occurence of any combination?
3. Express this probability as $\delta/\binom{N}{w}$ for δ of the form $\prod_{i=1}^{N-1}(1 + x_i/q)$.
4. How could this have been found directly by using the result of **Q.2**?

Remark. Generating (close-to) uniform permutations and combinations is an important step in code- and lattice-based cryptosystems. The quantity δ computed above corresponds to a *divergence* between the uniform distribution and the one obtained with the above algorithm. This exercise is based on: https://ntruprime.cr.yp.to/divergence-20180430.pdf.