1 Indistinguishability

We'll now start a major unit on *indistinguishability* and *pseudorandomness*. These concepts are a cornerstone of modern cryptography, underlying several foundational applications such as pseudorandom generators, secure encryption, "commitment" schemes, and much more.

For example, our most immediate application of indistinguishability will be to construct cryptographically strong *pseudorandom bit generators*. These are algorithms that produce many "random-looking" bits, while using very little "true" randomness. (In particular, the bits they output will necessarily be "very far" from truly random, in a statistical sense.) One easy-to-imagine application would be to use the pseudorandom bit stream as a one-time encryption pad, which would allow the shared secret key to be much smaller than the message. But what does it *mean* for a string of bits to be "random-looking"? And how can we be confident that using such bits does not introduce any unforeseen weaknesses in our system?

More generally, the motivating question for our study is:

When can two (possibly different) objects be considered *effectively the same*?

The answer:

When they can't be told apart!

Though seemingly glib, this answer encapsulates a very powerful mindset that will serve us well as we go forward.

1.1 Statistical Indistinguishability

We use probability theory to model (in)distinguishability. If two distributions are identical, then they certainly should be considered indistinguishable. We relax this condition to define *statistical* indistinguishability, for when the *statistical distance* between the two distributions is negligible. The statistical distance between two distributions X and Y over a domain Ω is defined as

$$\Delta(X,Y) := \sup_{A \subseteq \Omega} |X(A) - Y(A)|.$$

We view A as a statistical "test" — $X(A) = \sum_{w \in A} \Pr[X = w]$ being the probability that a draw from X lands in A, and likewise likewise $Y(A)$. Note that A and $\bar{A}$ are effectively the same test, since

$$|X(\bar{A}) - Y(\bar{A})| = |1 - X(A) - (1 - Y(A))| = |Y(A) - X(A)| = |X(A) - Y(A)|.$$

Lemma 1.1. *For distributions X, Y over a finite domain Ω,*

$$\Delta(X,Y) = \frac{1}{2} \sum_{w \in \Omega} |X(w) - Y(w)|.$$

Proof. Let the test $A = \{w \in \Omega \ : \ X(w) > Y(w)\}$. This makes $X(A) - Y(A)$ as large as possible, so $\Delta(X,Y) = X(A) - Y(A) = \sum_{w \in A} |X(w) - Y(w)|$. As noted above, we also have $\Delta(X,Y) = Y(\bar{A}) - X(\bar{A}) = \sum_{w \in \bar{A}} |X(w) - Y(w)|$. Summing the two equations, we have $2\Delta(X,Y) = \sum_{w \in \Omega} |X(w) - Y(w)|$, as desired. □

Statistical distance is very robust, which enhances its usefulness. Using Lemma 1.1, the following facts are straightforward to prove.

Lemma 1.2. *Let f be a (randomized) function on the domain of X, Y. We have $\Delta(f(X), f(Y)) \leq \Delta(X, Y)$.*

In other words, statistical distance cannot be increased by the application of a (randomized) function.

Lemma 1.3. *Statistical distance is a metric; in particular, $\Delta(X, Z) \leq \Delta(X, Y) + \Delta(Y, Z)$.*

Statistical distance lets us say when two (sequences of) distributions are "essentially the same," in an asymptotic sense.

Definition 1.4. Let $\mathcal{X} = \{X_n\}_{n\in\mathbb{N}}$ and $\mathcal{Y} = \{Y_n\}_{n\in\mathbb{N}}$ be sequences of probability distributions, called *ensembles*. We say that $\mathcal{X}$ and $\mathcal{Y}$ are *statistically indistinguishable*, written $\mathcal{X} \overset{s}{\approx} \mathcal{Y}$, if

$$\Delta(X_n, Y_n) = \mathrm{negl}(n).$$

Example 1.5. Let X_n be the uniform distribution over $\{0,1\}^n$, and let Y_n be the uniform distribution over the nonzero strings $\{0,1\}^n \backslash \{0^n\}$. An optimal test A is the singleton set $A = \{0^n\}$, yielding $\Delta(X_n, Y_n) = 2^{-n} = \mathrm{negl}(n)$, so $\mathcal{X} \overset{s}{\approx} \mathcal{Y}$. (This can also be seen by calculating the summation in Lemma 1.1.) The analysis extends similarly to any Y_n that leaves out a $\mathrm{negl}(n)$ fraction of $\{0,1\}^n$. From this we can say that such ensembles $\mathcal{Y}$ are "essentially uniform," or *statistically pseudorandom*.

Question: is a statistically peudorandom generator possible? This depends on the definition of "generator" (which we give below), but for any meaningful definition of the term, it isn't possible! This can be shown by explicitly demonstrating a subset (test) containing all the points outside the image of the generator, which make up a significant fraction of the range.

1.2 Computational Indistinguishability

We can define a natural analogue of statistical distance in the computational setting, where the "test" is implemented by an *efficient algorithm*. Namely, for distributions X and Y and an algorithm $\mathcal{A}$ (possibly randomized), define $\mathcal{A}$'s *distinguishing advantage* between X and Y as

$$\mathbf{Adv}_{X,Y}(\mathcal{A}) = |\Pr[\mathcal{A}(X) = 1] - \Pr[\mathcal{A}(Y) = 1]|.$$

(The output of $\mathcal{A}$ can be arbitrary, but we interpret 1 as a special output indicating that the test implemented by $\mathcal{A}$ is "satisfied," and any other output as "not satisfied.") We extend this to ensembles $\mathcal{X}$ and $\mathcal{Y}$, making $\mathbf{Adv}_{\mathcal{X},\mathcal{Y}}(\mathcal{A})$ a function of $n \in \mathbb{N}$.

Definition 1.6. Let $\mathcal{X} = \{X_n\}$ and $\mathcal{Y} = \{Y_n\}$ be ensembles, where X_n and Y_n are distributions over $\{0,1\}^{l(n)}$ for $l(n) = \mathrm{poly}(n)$. We say that $\mathcal{X}$ and $\mathcal{Y}$ are *computationally indistinguishable*, written $\mathcal{X} \overset{c}{\approx} \mathcal{Y}$, if $\mathbf{Adv}_{\mathcal{X},\mathcal{Y}}(\mathcal{A}) = \mathrm{negl}(n)$ for all non-uniform PPT algorithms $\mathcal{A}$. We say that $\mathcal{X}$ is (computationally) *pseudorandom* if $\mathcal{X} \overset{c}{\approx} \{U_{l(n)}\}$, the ensemble of uniform distributions over $\{0,1\}^{l(n)}$.

The basic facts about statistical distance also carry over to computational indistinguishability, where all functions/tests are restricted to be efficient.

Lemma 1.7 (Composition lemma)**.** *Let $\mathcal{B}$ be a non-uniform PPT algorithm. If $\{X_n\} \overset{c}{\approx} \{Y_n\}$, then $\{\mathcal{B}(X_n)\} \overset{c}{\approx} \{\mathcal{B}(Y_n)\}$.*

Proof. Let $\mathcal{D}$ be any non-uniform PPT algorithm attempting to distinguish $\{\mathcal{B}(X_n)\}$ from $\{\mathcal{B}(Y_n)\}$; we wish to show that its advantage must be negligible. Consider an algorithm $\mathcal{A}$ that, given input x, runs $\mathcal{D}(\mathcal{B}(x))$ and outputs whatever $\mathcal{D}$ outputs. Clearly $\mathcal{A}$ is non-uniform PPT. By construction, we have

$$\mathbf{Adv}_{X_n,Y_n}(\mathcal{A}) = \mathbf{Adv}_{\mathcal{B}(X_n),\mathcal{B}(Y_n)}(\mathcal{D}).$$

The left-hand side is $\mathrm{negl}(n)$ by hypothesis, hence so is the right-hand side, as desired. □

Lemma 1.8 (Hybrid lemma)**.** *Let $\mathcal{X}^i = \{X_n^i\}$ for $i \in [m]$, where $m = \mathrm{poly}(n)$. If $\mathcal{X}^i \stackrel{c}{\approx} \mathcal{X}^{i+1}$ for every $i \in [m-1]$, then $\mathcal{X}^1 \stackrel{c}{\approx} \mathcal{X}^m$.*

Proof. Let $\mathcal{D}$ be any non-uniform PPT algorithm attempting to distinguish $\mathcal{X}^1$ from $\mathcal{X}^m$. Let $p_i = p_i(n) = \Pr[\mathcal{D}(X_n^i) = 1]$. By the triangle inequality, we can write $\mathcal{D}$'s advantage as

$$\mathbf{Adv}_{\mathcal{X}^1,\mathcal{X}^m}(\mathcal{D}) = |p_1 - p_m| \leq \sum_{i\in[m-1]} |p_i - p_{i+1}| = \sum_{i\in[m-1]} \mathbf{Adv}_{\mathcal{X}^i,\mathcal{X}^{i+1}}(\mathcal{D}).$$

Now by assumption, each $\mathbf{Adv}_{\mathcal{X}^i,\mathcal{X}^{i+1}}(\mathcal{D}) = \nu_i(n)$, where $\nu_i(n)$ is a negligible function—which may be different for each i. The sum of $\mathrm{poly}(n)$-many negligible functions is indeed negligible: letting $\nu(n) = \sum_i \nu_i(n)$, we need to show that for all $c > 0$, there exists some n_0 such that $\nu(n) \leq n^{-c}$ for all $n \geq n_0$. By assumption, we know that for each i, there is some n_i such that $\nu_i(n) \leq n^{-c}/m$ for *all* $n \leq n_i$. Letting n_0 be the largest of these, it follows that $\nu(n) \leq n^{-c}$ for all $n \geq n_0$, as desired. □

Remark 1.9. In the proof of the lemma, we used the triangle inequality on the quantities $\mathbf{Adv}_{\mathcal{X}^i,\mathcal{X}^{i+1}}(\mathcal{D})$ to conclude something about $\mathbf{Adv}_{\mathcal{X}^1,\mathcal{X}^m}(\mathcal{D})$. Syntactically this is unremarkable, but observe closely what we have done: even though $\mathcal{D}$'s "goal in life" is to distinguish between $\mathcal{X}^1$ and $\mathcal{X}^m$, by referring to the quantities $\mathbf{Adv}_{\mathcal{X}^i,\mathcal{X}^{i+1}}(\mathcal{D})$, we are implicitly considering how $\mathcal{D}$ behaves on *all* the hybrid ensembles $\mathcal{X}^i$ — these are distributions on which $\mathcal{D}$ was never "designed" to run! Yet because $\mathcal{D}$ is "just an algorithm," we can run it and use it for whatever purposes we like. The hybrid lemma says that in order for $\mathcal{D}$ to distinguish between $\mathcal{X}^1$ and $\mathcal{X}^m$, it must also distinguish between $\mathcal{X}^i$ and $\mathcal{X}^{i+1}$ for some i, which is impossible by hypothesis.

2 Pseudorandom Generators

Definition 2.1. A *deterministic* function $G : \{0,1\}^* \to \{0,1\}^*$ is a *pseudorandom generator* (PRG) with output length $\ell(n) > n$ if:

- G can be computed by a polynomial-time algorithm,
- $|G(x)| = \ell(|x|) > |x|$ for all $x \in \{0,1\}^*$, and
- the ensemble $\{G(U_n)\}$ is (computationally) pseudorandom.

This last property essentially says that $\{G(U_n)\}$ and $\{U_{\ell(n)}\}$ are computationally indistinguishable. By the composition lemma, it follows that $G(U_n)$ can be used in place of $U_{\ell(n)}$ in *any* (efficient) application! (*Exercise:* Show that a PRG cannot exist if we demand *statistical* pseudorandomness.)

2.1 Expansion of a PRG

From the definition, it is easy to see that the "weakest" PRG we could ask for would be one that stretches its input by just 1 bit, i.e., $\ell(n) = n + 1$. Is there an upper limit on how much a PRG can stretch? The following theorem says that there is (effectively) *no limit*: if you can stretch by even just 1 bit, then you can stretch by essentially any (polynomial) amount!

Theorem 2.2. *Suppose there exists a PRG G with expansion $\ell(n) = n + 1$. Then for any polynomial $t(\cdot) = \mathrm{poly}(n)$, there exists a PRG $G_t : \{0,1\}^n \to \{0,1\}^{t(n)}$.*

We will prove this theorem in the next lecture.

Remark 2.3. This theorem says something extremely strong. Observe that the image $\{G_t(s) : s \in \{0,1\}^n\}$ of G_t is an extremely small fraction $2^{n-t(n)}$ of its range set $\{0,1\}^{t(n)}$. Yet no computationally bounded algorithm can distinguish a random element from this small subset, from a truly random one over the whole space!