Lecture 13

Probabilistic Complexity

There are many instances of problems with efficient randomized or probabilistic algorithms for which no good deterministic algorithms are known. In the next few lectures we take a complexity-theoretic look at probabilistic computation. We define a simple model of randomized computation, the *probabilistic Turing machine*, define some basic probabilistic complexity classes, and outline the relationship of these classes to conventional time and space classes. Our main result, which we prove next time, is that the class BPP of sets accepted by polynomial-time probabilistic algorithms with error probability bounded below $\frac{1}{2}$ is contained in $\Sigma_2^{\mathrm{p}} \cap \Pi_2^{\mathrm{p}}$ [112].

Many probabilistic algorithms have only a one-sided error; that is, if the input string is in the set, then the algorithm accepts with high probability; but if the string is not in the set, then the algorithm rejects always. The corresponding probabilistic complexity class is known as RP and is called *random polynomial time*.

Discrete Probability

Before we begin, let us recall some basic concepts from discrete probability theory.

Law of Sum The *law of sum* says that if $\mathcal{A}$ is a collection of pairwise disjoint events, that is, if $A \cap B = \varnothing$ for all $A, B \in \mathcal{A}$, $A \neq B$, then the

probability that at least one of the events in $\mathcal{A}$ occurs is the sum of the probabilities:

$$\Pr(\bigcup \mathcal{A}) \quad = \quad \sum_{A \in \mathcal{A}} \Pr(A).$$

Expectation The *expected value* $\mathcal{E}X$ of a discrete random variable X is the weighted sum of its possible values, each weighted by the probability that X takes on that value:

$$\mathcal{E}X \quad = \quad \sum_{n} n \cdot \Pr(X = n).$$

For example, consider the toss of a coin. Let

$$X \quad = \quad \begin{cases} 1, & \text{if the coin turns up heads} \\ 0, & \text{otherwise.} \end{cases} \tag{13.1}$$

Then $\mathcal{E}X = \frac{1}{2}$ if the coin is unbiased. This is the expected number of heads in one flip. Any function $f(X)$ of a discrete random variable X is a random variable with expectation

$$\mathcal{E}f(X) \quad = \quad \textstyle\sum_n n \cdot \Pr(f(X) = n) \quad = \quad \sum_m f(m) \cdot \Pr(X = m).$$

It follows immediately from the definition that the expectation function $\mathcal{E}$ is linear. For example, if X_i are the random variables (13.1) associated with n coin flips, then

$$\mathcal{E}(X_1 + X_2 + \cdots + X_n) \quad = \quad \mathcal{E}X_1 + \mathcal{E}X_2 + \cdots + \mathcal{E}X_n,$$

and this gives the expected number of heads in n flips. The X_i need not be independent; in fact, they could all be the same flip.

Conditional Probability and Conditional Expectation The *conditional probability* $\Pr(A \mid B)$ is the probability that event A occurs given that event B occurs. Formally,

$$\Pr(A \mid B) \quad = \quad \frac{\Pr(A \cap B)}{\Pr(B)}.$$

The conditional probability is undefined if $\Pr(B) = 0$.

The *conditional expectation* $\mathcal{E}(X \mid B)$ is the expected value of the random variable X given that event B occurs. Formally,

$$\mathcal{E}(X \mid B) \quad = \quad \sum_{n} n \cdot \Pr(X = n \mid B).$$

If the event B is that another random variable Y takes on a particular value m, then we get a real-valued function $\mathcal{E}(X \mid Y = m)$ of m. Composing

this function with the random variable Y itself, we get a new random variable, denoted $\mathcal{E}(X \mid Y)$, which is a function of the random variable Y. The random variable $\mathcal{E}(X \mid Y)$ takes on value n with probability

$$\sum_{\mathcal{E}(X|Y=m)=n} \Pr(Y = m),$$

where the sum is over all m such that $\mathcal{E}(X \mid Y = m) = n$. The expected value of $\mathcal{E}(X \mid Y)$ is just $\mathcal{E}X$:

$$\begin{aligned}
\mathcal{E}(\mathcal{E}(X \mid Y)) &= \sum_m \mathcal{E}(X \mid Y = m) \cdot \Pr(Y = m) \\
&= \sum_m \sum_n n \cdot \Pr(X = n \mid Y = m) \cdot \Pr(Y = m) \\
&= \sum_n n \cdot \sum_m \Pr(X = n \wedge Y = m) \qquad (13.2) \\
&= \sum_n n \cdot \Pr(X = n) \\
&= \mathcal{E}X
\end{aligned}$$

(see [39, p. 223]).

Independence and Pairwise Independence A set of events $\mathcal{A}$ are *independent* if for any subset $\mathcal{B} \subseteq \mathcal{A}$,

$$\Pr(\bigcap \mathcal{B}) = \prod_{A \in \mathcal{B}} \Pr(A).$$

They are *pairwise independent* if for every $A, B \in \mathcal{A}$, $A \neq B$,

$$\Pr(A \cap B) = \Pr(A) \cdot \Pr(B).$$

For example, the probability that two successive flips of a fair coin both come up heads is $\frac{1}{4}$.

Pairwise independent events need not be independent. Consider the following three events:

- The first flip gives heads.
- The second flip gives heads.
- Of the two flips, one is heads and one is tails.

The probability of each pair is $\frac{1}{4}$, but the three cannot happen simultaneously.

If A and B are independent, then $\Pr(A \mid B) = \Pr(A)$.

Inclusion–Exclusion Principle It follows from the law of sum that for any events A and B, disjoint or not,

$$\Pr(A \cup B) = \Pr(A) + \Pr(B) - \Pr(A \cap B).$$

More generally, for any collection $\mathcal{A}$ of events,

$$\Pr(\bigcup \mathcal{A}) = \sum_{A \in \mathcal{A}} \Pr(A) - \sum_{\substack{\mathcal{B} \subseteq \mathcal{A} \\ |\mathcal{B}|=2}} \Pr(\bigcap \mathcal{B}) + \sum_{\substack{\mathcal{B} \subseteq \mathcal{A} \\ |\mathcal{B}|=3}} \Pr(\bigcap \mathcal{B}) - \cdots \pm \Pr(\bigcap \mathcal{A}).$$

This equation is often used to estimate the probability of a join of several events. The first term alone gives an upper bound and the first two terms give a lower bound:

$$\Pr(\bigcup \mathcal{A}) \leq \sum_{A \in \mathcal{A}} \Pr(A)$$

$$\Pr(\bigcup \mathcal{A}) \geq \sum_{A \in \mathcal{A}} \Pr(A) - \sum_{\substack{A,B \in \mathcal{A} \\ A \neq B}} \Pr(A \cap B).$$

Probabilistic Turing Machines

Intuitively, we can think of a probabilistic Turing machine as an ordinary deterministic TM, except that at certain points in the computation it can flip a fair coin and make a binary decision based on the outcome. The probability of acceptance is the probability that its computation path, directed by the outcomes of the coin tosses, leads to an accept state.

Formally, we define a *probabilistic Turing machine* to be an ordinary deterministic TM with an extra semi-infinite read-only tape containing a binary string called the *random bits*. The machine runs as an ordinary deterministic TM, consulting its random bits in a read-only fashion. We write $M(x, y)$ for the outcome, either accept or reject, of the computation of M on input x with random bits y. We say that M is $T(n)$ time bounded (respectively, $S(n)$ space bounded) if for every input x of length n and every random bit string, it runs for at most $T(n)$ steps (respectively, uses at most $S(n)$ worktape cells).

In this model, the probability of an event is measured with respect to the uniform distribution on the space of all sequences of random bits. This is the measure that would result if a fair coin were flipped infinitely many times with the ith random bit determined by the outcome of the ith coin flip.

In practice, we consider only time-bounded computations, in which case the machine can look at only finitely many random bits. This makes the

calculation of the probabilities of events easier. For example, if M is $T(n)$ time bounded, then the probability that M accepts its input string x is

$$\Pr_y(M(x,y) \text{ accepts}) \quad = \quad \frac{|\{y \in \{0,1\}^k \mid M(x,y) \text{ accepts}\}|}{2^k},$$

where k is any number exceeding $T(|x|)$. The notation $\Pr_y(E)$ refers to the probability of event E with a bit string y chosen uniformly at random among all strings of length k.

Randomness can be regarded as a computational resource, much like time and space. One can measure the number of random bits consulted in a computation. We show some examples of this in Lectures 18 to 20.

The following are two basic complexity classes defined for probabilistic Turing machines.

Definition 13.1 *A set A is in RP if there is a probabilistic Turing machine M with polynomial time bound n^c such that*

- *if $x \in A$, then $\Pr_y(M(x,y)$ accepts$) \geq \frac{3}{4}$; and*
- *if $x \notin A$, then $\Pr_y(M(x,y)$ accepts$) = 0$.*

The definition of BPP is the same, except we replace the second condition with:

- *if $x \notin A$, then $\Pr_y(M(x,y)$ accepts$) \leq \frac{1}{4}$.*

Equivalently, a set A is in BPP if there is a probabilistic Turing machine M with time bound n^c such that for all inputs x,

$$\Pr_y(M(x,y) \text{ errs in deciding whether } x \in A) \quad \leq \quad \tfrac{1}{4}.$$

We have used $\frac{1}{4}$ and $\frac{3}{4}$ in the definition of RP and BPP, but actually any $\frac{1}{2} - \varepsilon$ and $\frac{1}{2} + \varepsilon$ will do. It matters only that the probabilities be bounded away from $\frac{1}{2}$ by a positive constant ε independent of the input size.

Also, as previously observed, the length of the random bit string is not important; any set of strings of sufficient length will do, as long as the machine has access to as many random bits as it needs.

It is easy to see that $P \subseteq RP \subseteq NP$, $RP \subseteq BPP$, and BPP is closed under complement. We show next time that $BPP \subseteq \Sigma_2^{\mathrm{p}} \cap \Pi_2^{\mathrm{p}}$.

Other classes such as $RPSPACE$ and RNC can be defined similarly.

Probabilistic Tests with Polynomials

Here is an example of a probabilistic test for which no equally efficient deterministic test is known: determining whether a given multivariate polynomial $p(x_1, \ldots, x_n)$ of low degree with integer coefficients is identically 0.