1 Random complexity classes

To formalize the idea of randomized computation, we need the notion of a probabilistic Turing Machine (PTM).

A probabilistic Turing Machine is a TM with two transition functions $\delta_0$ and $\delta_1$. At every step of an execution with input $x$, we apply $\delta_0$ with probability $1/2$, and $\delta_1$ otherwise. For a PTM $M$, its output $M(x)$ is now a random variable. We say $M$ runs in time $T(n)$ if for any input of length $n$, $M$ halts within $T(n)$ time regardless of the random choices made.

The standard class for efficient randomized computation is called BPP (bounded-error probabilistic polynomial-time).

**Definition 1.** A language $L$ is in BPP if there exists a PTM $P$ and a polynomial $p(\cdot)$, such that $P$ runs in time $p(n)$ and \[ \Pr[P(x) = 1] \geq 3/4. \]

The constant $3/4$ in Definition 1 is not essential, as long as it is strictly larger than $1/2$. This is captured by a notion called “amplification”. On the other hand, if it is $1/2$, then the class defined would be (under standard assumptions) more powerful. That class is called PP, probabilistic polynomial-time. Historically PP is defined earlier than BPP, and as it turns out, PP is the wrong definition for efficient randomized computation.

We also have the “one-sided” error version of BPP, called RP (randomized polynomial-time).

**Definition 2.** A language $L$ is in RP if there exists a PTM $P$ and a polynomial $p(\cdot)$, such that $P$ runs in time $p(n)$ and

- if $x \in L$, then $\Pr[P(x) = 1] \geq 3/4$;
- if $x \notin L$, then $\Pr[P(x) = 0] = 1$.

In other words, an RP algorithm never errs if $x \notin L$, but it may reject some $x \in L$. The aforementioned Bit-PM algorithm by Lovász is indeed an RP algorithm.

The complementary class coRP is defined in the usual way:

\[ \text{coRP} := \{ L \mid \overline{L} \in \text{RP} \}. \]

There is also a “zero-error” version, called ZPP.

**Definition 3.** A language $L$ is in ZPP if there exists a PTM $P$ and a polynomial $p(\cdot)$, such that the expected running time of $P$ is at most $p(n)$ and if $P$ halts on input $x$, $P(x) = L(x)$.
In fact \( \text{ZPP} = \text{RP} \cap \text{coRP} \). To show \( \text{ZPP} \subseteq \text{RP} \), we just run the ZPP algorithm, and answer 0 if the we have used, say, 4 times the expected running time. The probability for that to happen is at most 1/4, due to Markov’s inequality.

**Lemma 1** (Markov’s inequality). *Let \( X \) be a non-negative random variable. For any \( a > 0 \),
\[
\Pr[X \geq a] \leq \frac{\mathbb{E}[X]}{a}.
\]*

*Proof.* Let \( I \) be an indicator variable for the event \( X \geq a \). Namely
\[
I = \begin{cases} 
1 & \text{if } X \geq a; \\
0 & \text{if } X < a.
\end{cases}
\]

Then, \( \mathbb{E}[I] = \Pr[X \geq a] \). Note that \( I \leq \frac{X}{a} \) in all circumstances. (This is called stochastic domination.) Then
\[
\Pr[X \geq a] = \mathbb{E}[I] \leq \mathbb{E}\left[\frac{X}{a}\right] = \frac{\mathbb{E}[X]}{a}.
\]

By a similar argument, \( \text{ZPP} \subseteq \text{coRP} \), implying that \( \text{ZPP} \subseteq \text{RP} \cap \text{coRP} \).

On the other hand, the ZPP algorithm for a language in \( L \in \text{RP} \cap \text{coRP} \) is to run both the \( \text{RP} \) and the \( \text{coRP} \) algorithms simultaneously. If one correct answer is obtained, then we accept it. Otherwise we restart and do it again. It is easy to check that the restarting probability is at most 1/4, and thus the total expected running time \( T \) satisfies
\[
T \leq \frac{3}{4}T_0 + \frac{1}{4}(T + T_0),
\]
where \( T_0 \) is the maximum running time of the \( \text{RP} \) and \( \text{coRP} \) algorithms. Solving it gives us \( T \leq 4T_0/3 \), which is still a polynomial.

## 2 Amplification of BPP

The constant 3/4 in Definition 1 can be replaced with any constant > 1/2. In fact, even just barely larger than 1/2 is enough (but not 1/2). This is captured by the following amplification theorem. In fact, the error probability can be amplified to exponentially small.

**Theorem 2.** *Let \( L \) be a language and suppose that there is a polynomial-time PTM \( M \) and a constant \( c \) such that for every \( x \), \( \Pr[M(x) = L(x)] \geq 1/2 + |x|^{-c} \). Then for every constant \( d > 0 \), there is a polynomial-time PTM \( M' \) such that for every \( x \), \( \Pr[M(x) = L(x)] \geq 1 - \exp(-|x|^d) \).*

The idea is simple: \( M' \) will just run \( M \) many times and then take a majority vote. In order to bound the error probability, we will need the so-called “Chernoff bound”.

\(^1\)\(\exp(x)\) is a shorthand for \( e^x \).
Lemma 3 (Chernoff Bounds). Let $X_1, \ldots, X_n$ be mutually independent, identically distributed (i.i.d.) random variables over $\{0, 1\}$, with probability $p$ to be 1. For any $0 < \varepsilon \leq 1$,

$$\Pr \left[ \sum_{i=1}^{n} X_i \geq (1+\varepsilon)pn \right] \leq \exp \left( -\frac{\varepsilon^2 pn}{3} \right);$$

$$\Pr \left[ \sum_{i=1}^{n} X_i \leq (1-\varepsilon)pn \right] \leq \exp \left( -\frac{\varepsilon^2 pn}{3} \right).$$

Note that the form above is not the strongest possible, but it is useful, neat, and easy to remember. A more general form and proofs can be found in [AB09, Theorem A.14]. Lemma 3 is proved via applying Markov’s inequality Lemma 1 on a suitable exponential random variable.

What Lemma 3 is saying is that if we have a number of independent random variables, then their sum cannot deviate from the mean by too much. In particular, it implies that with constant probability, the deviation is within $O(\sqrt{n})$.

Proof of Theorem 2. As said earlier, $M'$ will run $M$ for $s = 24n^{2c+d}$ times and then take a majority vote, where $n = |x|$ is the input length. Let $X_i$ be the output of the $i$th run and $X = \sum_{i=1}^{s} X_i$. Let $p = 1/2 + n^{-c}$. If the majority vote went wrong, then roughly $n^{-c}s$ of all these runs are wrong, which would have very small probability due to Lemma 3 since we set $s \gg n^{2c}$.

To be more precise, let us assume that $x \in L$. (The other case is completely symmetric.) Then, the output is wrong, if and only if $X < s/2$. Namely, we can set $\varepsilon = n^{-c}/2$ in Lemma 3, so that $(1 - \varepsilon)p > 1/2$, and

$$\Pr[X < s/2] \leq \Pr[X < (1 - \varepsilon)ps] \leq \exp \left( -\frac{n^{-2c}(1/2 + n^{-c})s}{12} \right) \leq \exp \left( -\frac{n^{-2c}s}{24} \right) = \exp (-n^d).$$

This finishes the proof. \hfill $\Box$

Remark (Bibliographic). Relevant chapters are [AB09, Chapter 7.3, 7.4] and [Pap94, Chapter 11].

References
