1 Statistical Distance, Algorithms & Complexity

1.1 Statistical Distance

Definition 1 (Statistical Distance $\Delta$) Given two random variables $X, Y$ distributed according to $D_1, D_2$ respectively with $V = [D_1] = [D_2]$ we define the statistical distance as

$$\Delta[X, Y] = \frac{1}{2} \sum_{\nu \in V} |\text{Prob}[X = \nu] - \text{Prob}[Y = \nu]|$$

![Graph of statistical distance]

Figure 1: The shaded area is the statistical distance of the two random variables $X, Y$

Definition 2 ($\epsilon$-close) Two random variables $X, Y$ are said to be $\epsilon$-close if $\Delta[X, Y] \leq \epsilon$.

1.2 An Example

Consider the following two probability distributions:

- $D_1$: is the uniform distribution over $[0, A)$. Note: $A > 2^{n-1}$
- $D_2$: is the uniform distribution over all $n$-bit numbers.

**Question:** What is the statistical distance of $D_1, D_2$?

Recall that
\[
\text{Prob}_{D_1}[x] = \frac{1}{A} \\
\text{Prob}_{D_2}[x] = \frac{1}{2^{n-1}}
\]

To facilitate the comparison we set \(\forall x \in [0, A] \setminus [0, 2^{n-1}] \) \(\text{Prob}_{D_2}[x] = 0\) (so that the two distributions are defined over the same sample space). Suppose now that \(X, Y : [0, A] \leftarrow [0, A]\) with \(X(x) = Y(x) = x\) are two random variables distributed according to \(D_1, D_2\) respectively. We have:

\[
\Delta[X, Y] = \frac{1}{2^n} \sum_{x \in [0, A)} |\text{Prob}_{D_1}[x] - \text{Prob}_{D_2}[x]| = \frac{1}{2^n} \left[ \sum_{x \in [0, 2^{n-1})} \left| \frac{1}{A} - \frac{1}{2^{n-1}} \right| + \sum_{x \in [2^{n-1}, A)} \frac{1}{A} \right] = \\
\frac{1}{2^n} \left[ 2^{n-1} \frac{1}{2^{n-1}} + (A - 2^{n-1}) \frac{1}{A} \right] = \frac{1}{2^n} \left[ 2^{n-1} \frac{4 - 2^{n-1}}{2^{n-1}} + \frac{4 - 2^{n-1}}{A} \right] = \\
\frac{1}{2^n} \frac{4 - 2^{n-1}}{A} = \frac{d = A - 2^{n-1}}{2^{n-1} + d} = \frac{1}{2^{n-1} + 1}.
\]

What is the significance of this?
If \(A\) is somewhat close to \(2^{n-1}\), e.g. \(d = 2^{n-1}\), then \(\Delta[X, Y] = \frac{1}{2^{n-1} + 1}\).

**Definition 3 (Negligible Function)** We call a function negligible if \(\forall c \exists n_0\) such that \(f(n) < \frac{1}{n^c} \forall n \geq n_0\).

**Definition 4 (Statistically Indistinguishable)** If \(X, Y\) are two random variables parameterized by \(n\), they are called statistically indistinguishable if \(\Delta[X, Y]\) is a negligible function in \(n\).

### 1.3 Algorithms & Complexity

Various formal models can be employed to describe the algorithms:

- Turing Machine
- RAM (random access machine)

**Definition 5 (Time Complexity of an algorithm)** \(T(n) = \max. \# \text{ of "steps" required to halt for any input of size } n\).

**Definition 6 (Space Complexity of an algorithm)** \(S(n) = \max. \# \text{ of additional memory units utilized in the course of the computation for any input of size } n\) (additional means that we exclude the memory units required to store the input).

**Definition 7 (Efficient Algorithm)** An algorithm is efficient if \(T(n) = O(n^c)\) for some \(c \in \mathbb{N}\).

### 1.4 Probabilistic Algorithms

Probabilistic Algorithms have access to a "true random bit generator". They can use the instruction \(x \leftarrow R\) to assign a random bit to the variable \(x\).
1.4.1 Example I

Consider the algorithm:
Input: $1^v$
Select $x_0, \cdots, x_{v-1} \leftarrow \{0, 1\}$
if $\sum_{l=0}^{v-1} 2^l x_l \geq 2^v - 1$
then output 1
else output 0

Clearly the output of this algorithm is a random variable. We call this algorithm $A$.

$$\operatorname{Prob}[A(1^v) = 1] = \frac{\{b \in \{0, 1\}^v | A \text{ with random bit selection = b output 1}\}}{2^v} = \frac{2^v - 1}{2^v} = \frac{1}{2}$$

1.4.2 Example II

Algorithm $A_1$:
Given $\epsilon$ (empty string)
repeat forever
   $x \leftarrow \{0, 1\}$
   if $x = 1$ output 1; halt

Probability that this algorithm halts approaches 0 with exponenational rate. Nevertheless the above algorithm may not terminate for any fixed finite amount of time. This makes the above algorithm a bit hard to analyze. To assist us in the analysis the following transformation of the above algorithm can be useful:
Algorithm $A_2$:
input $1^v$
choose $x_1, \cdots, x_v \leftarrow \{0, 1\}$
repeat for $i = 1, \cdots, v$
   $x = x_i$
   if $x = 1$ the output 1; halt
output fail

Probability calculation:
$$\operatorname{Prob}[A_2(1^v) = 1] = 1 - \frac{1}{2^v}$$
$$\operatorname{Prob}[A_2(1^v) = \text{FAIL}] = \frac{1}{2^v}$$

Is it frequently useful to draw the tree representation of a probabilistic algorithm. See figure 4.

1.4.3 Exercise

Let $A$ be a $n$-bit number with most significant bit $A = 1$
Goal: Select a random number in $[0, A)$
given A:
1: $n := \lceil \log A \rceil$
2: choose: $x_0, x_1, \cdots, x_{n-1} \leftarrow \{0, 1\}$
3: if $\sum_{l=0}^{n-1} 2^l x_l < A$, output $\sum_{l=0}^{n-1} 2^l x_l$ else goto 2:
Figure 2: Algorithm \(A_1\)

Figure 3: Algorithm \(A_2\)

Figure 4: Tree representations of probabilistic algorithms

Homework until Thursday 02/05/04: Investigate the output probability distribution of this Algorithm.

1.5 An Alternative definition of Statistical Distance

The proof of the following two theorems is easy and it is recommended that you work out the details:

**Theorem 8** For any deterministic algorithm \(A\) with output \(\{0, 1\}\) (sometimes also called a predicate or test) and two random variables \(X, Y\) following the distributions \(D_1, D_2\) respectively it holds that

\[
\Delta[X, Y] \geq |\text{Prob}[A(X) = 1] - \text{Prob}[A(Y) = 1]| \]

**Theorem 9** Using the same notation as above,

\[
\Delta[X, Y] = \max_A |\text{Prob}[A(X) = 1] - \text{Prob}[A(Y) = 1]| \]

**Proof.** Set \(A\) to be as follows: \(A : V \rightarrow \{0, 1\}\) where \(V = [D_1] = [D_2]\), with \(A(v) = 1\) if and only if \(\text{Prob}[X = v] < \text{Prob}[Y = v]\). Work out the details.