1. Suppose our goal is to sample the value of a fair coin ("heads" having probability 1/2, and "tails" having probability 1/2) but unfortunately we only have a faulty coin, which returns "heads" with some unknown probability \( p \), and 1\( - p \) otherwise. Suppose we do not know the value of \( p \), or even whether \( p > 1/2 \) or \( p < 1/2 \). We do assume \( p \in (0, 1) \) (neither 0 nor 1). Design a simple algorithm which uses a number of coin flips to return a value ("heads" or "tails") in such a way that "heads" and "tails" are each returned with exact probability 1/2.

(a) Describe the algorithm. [5 marks]

(b) Argue/prove that "heads" and "tails" are equally likely to be returned. [5 marks]

(c) Prove that the expected number of coin-flips that will be used by your algorithm is \( p^{-1}(1-p)^{-1} \). [5 marks]

2. Consider the following balls-and-bin game. We start with one black ball and one white ball in a bin. We repeatedly do the following: choose one ball from the bin uniformly at random, and then put the ball back in the bin with another ball of the same colour. We repeat until there are \( n \) balls in the bin. Show that the number of white balls is equally likely to be any number between 1 and \( n - 1 \). [10 marks]

3. Let \( Y \) be a nonnegative integer-valued random variable with (strictly) positive expectation. Prove that

\[
\frac{(E[Y])^2}{E[Y^2]} \leq \Pr[Y \neq 0] \leq E[Y].
\]
4. Suppose we have \( n \) independent Bernoulli random variables, and we are actually given the values of their parameters as the list \( p_1, p_2, \ldots, p_n \), where \( p_i = \Pr[x_i = 1] \). Our interest is in \( \sum_{i=1}^{n} x_i \); we are given a threshold value \( T \) and asked for a polynomial-time algorithm (polynomial in \( n \) and \( \log(\min_i p_i) \)) to decide whether \( \Pr[\sum_{i=1}^{n} x_i \geq T] \geq 1/2 \) or not.

Your algorithm should be deterministic (not using any sampling). Also, you should not assume any special facts about the programming language being used, just basic arithmetic operations, iteration, and array declarations being available.

5. Consider the example of a product distribution \( P = P(p_1, \ldots, p_n) \) for generating tuples over \( \{0,1\}^n \), where we generate a sample \( \bar{x} \in \{0,1\}^n \) by independently generating the value \( x_i \) as the Bernoulli distribution for \( p_i \) ("1" having probability \( p_i \), "0" otherwise), and setting \( \bar{x} \) to be the product of all the \( x_i \)s.

Suppose now that we are told that we will be shown a sequence of \( \bar{x} \) tuples generated independently (with replacement) from a product distribution with hidden \( p_i \) values. Our challenge is to come up with the parameters \( p'_1, \ldots, p'_n \) of an hypothesis distribution \( P' \) which is within total variation distance of \( \epsilon \) to the true distribution. Specifically, with variation distance defined as

\[
\text{d}_{TV}(P, P') = \sum_{\bar{x} \in \{0,1\}^n} |\Pr_P(\bar{x}) - \Pr_{P'}(\bar{x})| 
\]

for a particular given \( \epsilon \in (0,1) \). Note that \( \Pr_P(\bar{x}) \) denotes the probability of \( \bar{x} \) in the distribution \( P \), \( \Pr_{P'}(\bar{x}) \) defined analogously.

Suppose we consider the very simple algorithm whereby we draw \( \ell \) samples from \( P \), then for each of the \( i \in [n] \), set \( p'_i \) to be the fraction of samples which have 1 in position \( i \).

We will prove that with high probability, this algorithm can be an efficient (polynomial in \( n \), inverse \( \epsilon^{-1} \) of error, inverse \( \delta^{-1} \) of failure) "PAC-learning" algorithm for a product distribution.

(a) Prove that if the values \( p'_1, \ldots, p'_n \) satisfy \( |p_i - p'_i| \leq \frac{\epsilon}{2n} \) for every \( i \in [n] \), then it is the case that

\[
\sum_{\bar{x} \in \{0,1\}^n} |\Pr_P(\bar{x}) - \Pr_{P'}(\bar{x})| \leq \epsilon.
\]

(b) Let \( i \in [n] \). Prove that if we take \( \ell > 2(\frac{n}{\epsilon})^2 \log(\frac{2}{\delta}) \) samples from \( P \) to estimate \( p'_i \), then with probability at least \( 1 - \delta \), we will have \( |p_i - p'_i| \leq \frac{\epsilon}{2n} \).

(c) Combine (a) and (b) to argue why, with probability at least \( 1 - \delta \), our algorithm will return a distribution within variation distance \( \epsilon \) of \( P \).
6. Consider a more general situation than that of 5., where we now have two product distributions $P, Q$ over $\{0, 1\}^n$, defined by parameters $p_1, \ldots, p_n$ and $q_1, \ldots, q_n$ respectively.

Suppose there is also a choice probability $\gamma \in (0, 1)$, where we draw samples from distribution $P$ with probability $\gamma$, or from distribution $Q$ with probability $1 - \gamma$.

Note that with this “mixed” process, that for every $i \in [n],$

$$\Pr_{a \sim P + (1-a) \sim Q} [x_i = 1] = \gamma p_i + (1 - \gamma)q_i.$$

Prove the following:

(a) For every $i, j \in [n], i \neq j,$ \hspace{1cm} [5 marks]

$$\text{Cov}[x_i, x_j] = \gamma(1 - \gamma)(p_i - q_i)(p_j - q_j).$$

(b) Suppose that in the “mixed” distribution, we have at least three different $i, j$ pairs (different meaning non-identical, that any two pairs have a intersection of size at most 1) such that $\text{Cov}[x_i, x_j] \neq 0$.

Prove that under this assumption, that for every $i \in [n]$ which has one “partner” index $i^* \neq i$ such that $\text{Cov}[x_i, x_{i^*}] \neq 0$, that there is also at least one “triplet” index $\hat{i} \in [n] \setminus \{i, i^*\}$ such that $\text{Cov}[x_i, x_{\hat{i}}], \text{Cov}[x_i, x_{i^*}]$ and $\text{Cov}[x_{i^*}, x_{\hat{i}}]$ are all non-zero.

(c) Suppose the condition of (b) holds (at least three different pairs satisfying $\text{Cov}[x_i, x_j] \neq 0$), and define a measure $w(i)$ on $[n]$, in the following way: \hspace{1cm} [10 marks]

$$w(i) = \begin{cases} 
0 & \text{if } \text{Cov}[x_i, x_j] = 0 \text{ for all } j \in [n] \setminus \{i\} \\
-\frac{1}{2} \ln \left( \frac{\text{Cov}[x_{i^*}, x_{\hat{i}}]}{\text{Cov}[x_i, x_{i^*}]} \right) & \text{for any } i^* \neq i, \hat{i} \in [n] \setminus \{i, i^*\} \text{ satisfying (b) above}
\end{cases}$$

Prove that $w(i)$ is well-defined, taking the same value regardless of which $i^*, \hat{i}$ pair (satisfying condition (b)) are chosen.

(d) Again suppose the condition of (b) holds, and let $w(i)$ be defined as in (c). For any pair $i, j \neq i$, define $w(i, j) = -\ln(\text{Cov}[x_i, x_j]).$

Show that $w$ satisfies the property of additivity, that for every $i, j \neq i, that$

$$w(i, j) = w(i) + w(j).$$

Mary Cryan, 6th February 2017