princeton univ. F'18 cos 521: Advanced Algorithm Design

Homework 5: Select Topics (**60pts**)

Due: **Friday, January 11th, 2019, 11:59pm**

Collaboration is allowed on this problem set, but solutions must be written-up individually. Please list collaborators for each problem separately, or write "No Collaborators" if you worked alone. Collaboration is not allowed on bonus problems.

Please prepare your problem sets in LaTeX and compile to a PDF for your final submission. A LaTeX template is available on the course webpage.

§1 **(10 pts)** In class it was mentioned that there exist simpler compressed sensing schemes that work when noise/numerical precision is not an issue. Let $q_1, \ldots, q_n \in \mathbb{R}^n$ be any set of *distinct* numbers. E.g. we could choose $[q_1, \ldots, q_n] = [1, \ldots, n]$. Consider the sensing matrix $A \in \mathbb{R}^{2k \times n}$:

$$A = \begin{bmatrix} 1 & 1 & 1 & \ldots & 1 \\ q_1 & q_2 & q_3 & \ldots & q_n \\ (q_1)^2 & (q_2)^2 & (q_3)^2 & \ldots & (q_n)^2 \\ \vdots & \vdots & \vdots & & \vdots \\ (q_1)^{2k-1} & (q_2)^{2k-1} & (q_3)^{2k-1} & \ldots & (q_n)^{2k-1} \end{bmatrix}$$

Show that, if $x \in \mathbb{R}^n$ is a k sparse vector – i.e. $\|x\|_0 \leq k$ – then we can recover x from Ax, which is a vector with length $2k$. You don't need to give an efficient algorithm. Just argue that for any given $y \in \mathbb{R}^{2k}$, there is at most one k-sparse x such that $y = Ax$. (Hint: Use that a non-zero degree d polynomial can't have more than d roots.)

§2 **(10 pts)** Assume that for the previous problem we have an algorithm $Decode(y)$ which returns x if $y = Ax$ for some k-sparse x. If $y \neq Ax$ for some k-sparse x, $Decode(y)$ can return anything.

Show how to construct a matrix $B \in \mathbb{R}^{O(\log n) \times n}$ (using a randomized algorithm) such that *for any* x (i.e. not necessarily sparse) it is possible to recover a single index/value pair (i, x_i) with $x_i \neq 0$ from Bx with constant probability (e.g. with success probability 9/10). Your algorithm can return *any* (i, x_i) as long as $x_i \neq 0$.

§3 **(10 pts)** Suppose Alice holds a subset of elements $S_A \subseteq \{1, \ldots, n\}$. Bob holds another subset $S_B \subseteq \{1, \ldots, n\}$. Using as little communication as possible, Alice wants to figure out if she or Bob hold any unique elements – i.e. if there is any $j \in A \cup B - A \cap B$.

Show that, for some constant c, Bob can send Alice a single message of $O(\log^c n)$ bits that allows her to find such a j if one exists, with constant success probability.

You can assume that Alice and Bob decide on a strategy in advance, and that they have access to an unlimited source of shared random bits (e.g. that are published by some third party).

§4 **(10 pts)**

(a) Let M be the transition matrix of a ergodic Markov Chain with mixing time t_0. Let $M' = 1/2(I + M)$ be the "lazy" version of this Markov Chain. Show that the mixing time of M' is at most $10t_0$. It's fine to have any constant (rather than 10) in this bound.

(b) Let M be the transition matrix of a random walk on an undirected graph G on n vertices that defines an ergodic Markov Chain with stationary distribution π. In the class, we defined the mixing time of this Markov Chain as the smallest integer t_0 such that for every distribution x on the vertices of G, $\|M^{t_0}x - \pi\|_1 \leq 1/4$. Justify this definition by arguing that the distance to stationary distribution shrinks exponentially: i.e., show that after kt_0 steps, $\|M^{kt_0}x - \pi\|_1 \leq 2^{-k}$.

§5 **(10 pts)** Let M be the transition matrix of a lazy random walk on the n-cycle, that is, at any vertex, the random walk stays at that vertex with probability 1/3 and moves to one of the two neighbors with probability 1/3 each. Show that the mixing time of this Markov Chain is $O(n^2 \log n)$. (Hint: lower bound the spectral gap of this Markov Chain. It might be helpful to guess a form for the eigenvectors of this transition matrix. One might expect the eigenvectors to be "periodic".)

§6 **(10 pts)** Suppose we have access to a string of n independent but biased random bits $X \in \{0,1\}^n$, where each entry of X is 1 with probability $p \leq 1/2$ and 0 with probability $1 - p$. We would like to use these bits to generate *unbiased* coin flips.

In particular, we wish to construct some function $F : \{0,1\}^n \to \{0,1\}^m$ for some $m \leq n$ such that $F(X)$ looks like m fair coin flips. In particular, any $Y \in \{0,1\}^m$ should appear as the output of F with probability $1/2^m$.

(a) Show that if F is a deterministic function, $m \leq n \log_2(\frac{1}{1-p})$.

Consider a richer set of strategies for generating unbiased random bits where, instead of outputting exactly m bits, we can output a variable number of bits j when given a sample of X. The output bits still need to be unbiased, meaning that for all j, the probability of outputting any string in $\{0,1\}^j$ is the same as the probability of outputting any other string in $\{0,1\}^j$.

(b) Show that for any deterministic function F, the expected number of uniform random bits output is upper bounded by $\mathbb{E}[j] \leq H(p)n$, where $H(p)$ is the binary entropy function.