Assignment 5
CS289: Algorithmic Machine Learning, Fall 2016
Due: December 7, 10PM

Guidelines for submitting the solutions:

- The assignments need to be submitted on Gradescope. Make sure you follow all the instructions - they are simple enough that exceptions will not be accepted.
- To save my eyes (and perhaps, more importantly, a few of your points) please use a good scanner and/or digitize the scan as per the instructions on the class web page.
- Start each problem or sub-problem on a separate page even if it means having a lot of white-space and write/type in large font.
- The solutions need to be submitted by 10 PM on the due date. No late submissions will be accepted.
- Please adhere to the code of conduct outlined on the class page.

1. In class we looked at an algorithm to approximate the number of 1's in a stream of 0's and 1's within error ε using $O((\log\log n)/\varepsilon^2)$ space. Here you will show a different way to obtain a similar guarantee. Consider the following algorithm for some number $0 \leq \alpha \leq 1$:

   (a) Initialize a counter $X = 0$.

   (b) For each 1 in the stream, increment X with probability $1/(1+\alpha)^X$.

   (c) Output some function $g(X)$ of the counter.

   (Taking $\alpha = 1$ and $g(X) = 2^X - 1$ gives the base algorithm we analyzed in class). Let t_n be the total number of 1's in the stream.

   (a) What should the function $g(X)$ be so that the expectation of the output of the above algorithm is exactly t_n? [1 point]

   (b) How small must α be to guarantee that $\mathsf{Pr}[|g(X) - t_n| \geq \varepsilon n] \leq 1/3$? [1 point]

   (c) Derive a bound on the space of the algorithm $S(n, \varepsilon)$ used to get the above guarantee. The dependence on n should be $O(\log\log n)$ (along with some dependence on ε) for full credit. [1 point]

2. Suppose you have a data stream $x_1, \ldots, x_n$ of items from some domain $[U]$. In class we saw one algorithm to estimate the frequency counts $f_u = |\{i : x_i = u\}|$ for all $u \in U$. Here we will develop a different algorithm with a different guarantee. As in class, let $h : [U] \to [s]$ be a uniformly random hash function. Let $z \in \{1, -1\}^U$ be uniformly random sequence of signs of length U. Consider the following algorithm:

   (a) Initialize s counters $C[\ell] = 0$ for $1 \leq \ell \leq s$.

   (b) For each item x_i in the stream, set $C[h(x_i)] = C[h(x_i)] + z_{x_i}$.

   For each u, the estimate for f_u is $\tilde{f}_u = z_u \cdot C[h(u)]$. Show the following properties of the sketch:

   (a) For every u, $\mathbb{E}[\tilde{f}_u] = f_u$. Here, the expectation is over the randomness of the hash function h and the string z. [1 point]

   (b) Show that for every u, $Var(\tilde{f}_u) = \left(\sum_{v \neq u} f_v^2\right)/s$. [2 point]

   (Hint: $Var(\tilde{f}_u) = \mathbb{E}[(\tilde{f}_u - f_u)^2]$. Now, as in our analysis for count-min sketch, the difference $\tilde{f}_u - f_u$ is dictated by collisions, i.e., $v \neq u$ such that $h(v) = h(u)$. To exploit this, for $v \in [U]$, let Y_v be the indicator random variable that is 1 if $h(v) = h(u)$ and 0 otherwise. Show that $\tilde{f}_u - f_u = z_u \cdot \sum_{v \neq u} f_v z_v Y_v$. You can then compute $\mathbb{E}[(\tilde{f}_u - f_u)^2]$ by first evaluating the expectation with respect to z and then with respect to h.)

3. Consider a setup just as above, but now we are trying to just estimate $\|f\|_2^2$ (instead of the individual frequencies). Show that the expected output of the following algorithm is exactly $\|f\|_2^2$. What is the space used by the algorithm? [2 points]

   (a) Initialize $X = 0$. Choose a random string $z \in \{1, -1\}^U$.

   (b) For each item x_i of the stream, set $X = X + z_{x_i}$.

   (c) Output X^2.

4. Suppose you have a data stream $x_1, \ldots, x_n$, where x_i is an element of $\{A, B\} \times [U]$; that is, each x_i is either (A, u) or (B, u) for some $u \in [U]$. An example would be $(A, 1), (B, 4), (A, 2), (B, 5), (A, 1)$. We can think of the stream as specifying two sets $A, B \subseteq [U]$ by viewing each $x_i \in \{A, B\} \times [U]$ of the stream as asking you to add the second coordinate of x_i to one of A or B as specified by the first coordinate . For instance, for the stream in the example, $A = \{1, 2\}$ and $B = \{4, 5\}$.

   Give an algorithm to compute the Jaccard similarity $J(A, B)$ of the sets specified by the stream within accuracy ε with probability at least $1/2$. For full credit, your algorithm should make only one pass over the stream and must use space at most $O((\log U)/\varepsilon^2)$ space. You don't need to prove the algorithm works. [2 points]