Problem Set 3 Solution

by Yuan Zhou

1. (a) $\sigma(u(S)v(S) + u(T)v(T) + u(U)v(U))$.

   (b) Since $\|u\|_2 = \|v\|_2 = 1$, we have $\|u\|_1, \|v\|_1 \le \sqrt{n}$. Therefore, there are at most $2\sqrt{n}/\delta$ possibilities for each of $u(S), v(S), u(T), v(T), u(U), v(U)$. Therefore, there are at most $(2\sqrt{n}/\delta)^6$ possible $f(S, T, U)$ vectors needed for the purpose of approximation.

   (c) We maintain a list $\mathcal{L}_i$ of $f(S, T, U)$ vectors for the first i vertices. We start from $\mathcal{L}_0 = \{(0, 0, 0, 0, 0, 0)\}$, and at each of the n iterations, we derive $\mathcal{L}_i$ from $\mathcal{L}_{i-1}$, where $1 \le i \le n$. For each element $(a, b, c, d, e, f, g) \in \mathcal{L}_{i-1}$, we consider the new vectors $(a + u_i, b + v_i, c, d, e, f), (a, b, c + u_i, d + v_i, e, f), (a, b, c, d, e + u_i, d + v_i)$ (corresponding to adding vertex i to S, T, U). Round the three new vectors to the nearest multiple of δ' (which will be chosen later), and add them to $\mathcal{L}_i$.

   Finally, $\mathcal{L}_n$ is the desired set of approximation vectors.

   Now that at each iteration, we might introduce a δ' additive error. There might be a $n\delta'$ additive error in the final approximation vectors. Therefore, we need to set $\delta' = \delta/n$, and the list size is upper bounded by $(2\sqrt{n}/\delta')^6 = O(n^{1.5}/\delta)^6$.

   (d) We use the natural extension of the dynamic programming described above, getting a list of at most $O(n^{1.5}/\delta)^{6k}$ approximating vectors (at precision δ). By choosing $k = O(1/\epsilon)$, the additive error introduced in the SVD step can be upper bounded by $\epsilon n^2/2$. The rest of the error is upper bounded by (for every partition S, T, U)

$$\Big| \sum_{t=1}^{k} \sigma_t(u_t(S)v_t(S) + u_t(T)v_t(T) + u_t(U)v_t(U)) - \sum_{t=1}^{k} \sigma_t((u_t(S) + \delta_{t,1})(v'_t(S) + \delta_{t,2}) + (u_t(T) + \delta_{t,3})(v'_t(T) + \delta_{t,4}) + (u_t(U) + \delta_{t,5})(v_t(U) + \delta_{t,6}))\Big|,$$

where $|\delta_{t,j}| \le \delta$ are the error terms. The value above is upper bounded by

$$\sum_{t=1}^{k} \sigma_t \Big(|u_t(S)v_t(S) - (u_t(S) + \delta_{t,1})(v_t(S) + \delta_{t,2})| + |u_t(T)v_t(T) - (u_t(T) + \delta_{t,3})(v_t(T) + \delta_{t,4})| + |u_t(U)v_t(U) - (u_t(U) + \delta_{t,5})(v_t(U) + \delta_{t,6}))| \Big)$$

$$= \sum_{t=1}^{k} \sigma_t \Big(|\delta_{t,1}v_t(S) + \delta_{t,2}u_t(S) + \delta_{t,1}\delta_{t,2}| + |\delta_{t,3}v_t(T) + \delta_{t,4}u_t(T) + \delta_{t,3}\delta_{t,4}| + |\delta_{t,5}v_t(U) + \delta_{t,6}u_t(U) + \delta_{t,5}\delta_{t,6}| \Big)$$

$$\le \sum_{t=1}^{k} \sigma_t \left(\delta(|u_t(S)| + |v_t(S)| + |u_t(T)| + |v_t(T)| + |u_t(U)| + |v_t(U)|) + 3\delta^2 \right)$$

$$\le \sum_{t=1}^{k} \sigma_t \left(\delta \cdot 2\sqrt{n} + 3\delta^2 \right) \qquad (\text{since } \|u\|_1, \|v\|_1 \le \sqrt{n})$$

$$\le \sum_{t=1}^{k} \sigma_t \cdot 3\sqrt{n}\delta \qquad (\text{for large enough } n)$$

$$\begin{aligned}&\leq k\sigma_1 \cdot 3\sqrt{n}\delta \\ &\leq kn^2 \cdot 3\sqrt{n}\delta.\end{aligned}$$

Therefore, we can upper bound this value by $\epsilon n^2/2$ by choosing $\delta = \epsilon/(6k\sqrt{n}) = \Omega(\epsilon^2/\sqrt{n})$. This would give an algorithm with ϵn^2 additive error which runs in time $n^{O(1)} \cdot O(n^{1.5}/\delta)^{6k} = (n/\epsilon)^{O(1/\epsilon)}$.

2. The probability that at least one of the x_i's is one is

$$1 - \prod_{i=1}^{n}(1 - \Pr[x_i = 1]) \leq 1 - (1 - (1-\epsilon)/l)^l \approx 1 - 1/e^{1-\epsilon},$$

for large enough l.

Now back to our problem of estimating the number of distinct elements. Suppose we want a $(1+\epsilon)$ approximation and there are l distinct elements. To get an estimation within $l(1 \pm \epsilon)$ for the min-hash method, at least one of the l elements should be mapped to the first $1/(l(1-\epsilon))$ fraction of the hash buckets (which happens with probability $1/(l(1-\epsilon)) \approx (1+\epsilon)/l$). Even when the hash function is l-wise independent (i.e., the l elements are hashed in a fully independent way), by the exercise above, the probability that at least one of the l elements mapped to the first $1/(l(1-\epsilon))$ fraction of the hash buckets is at most $1 - 1/e^{1+\epsilon}$. Therefore, with constant probability, we are not able to get a $(1+\epsilon)$ approximation.

3. (a) The different f_s's might cancel each other due to difference in their signs.

   (b) By solving the equation

$$\int_{t=0}^{x} 2 \cdot \frac{1}{\pi} \cdot \frac{dt}{1+t^2} = \frac{1}{2},$$

   we get the median value of $|\Lambda|$ is $x = 1$.

   (c) Let z_1, z_2 be the value such that

$$\Pr[Z \leq z_1] = 1/2 - \epsilon, \Pr[Z \leq z_2] = 1/2 + \epsilon.$$

   Now, we only need to prove that,

$$\Pr[z_1 \leq M \leq z_2] \geq 1 - \delta.$$

   We are going to show that $\Pr[z_1 \leq M] \geq 1 - \delta/2$. Similarly, we can show that $\Pr[M \leq z_2] \geq 1 - \delta/2$. By a union bound, we prove the desired statement.
   To prove $\Pr[z_1 \leq M] \geq 1 - \delta/2$, we note that

$$\Pr[z_1 \leq M] \geq \Pr[\text{more than half of } s_i\text{'s are no less than } z_1].$$

   Since each s_i is an independent sample of Z and therefore is no less than z_1 with probability $1/2 + \epsilon$ (by the definition of z_1). By a Chernoff bound, we know that as long as $k = C\log(1/\delta)/\epsilon^2$ for some large enough C, we have

$$\Pr[\text{more than half of } s_i\text{'s are no less than } z_1] \geq 1 - \delta/2,$$

   which implies that $\Pr[z_1 \leq M] \geq 1 - \delta/2$.

(d) We are going to show that

$$\int_{1-10\epsilon}^{1} 2 \cdot \frac{1}{\pi} \cdot \frac{dx}{1+x^2} > \epsilon,$$
$$\int_{1}^{1+10\epsilon} 2 \cdot \frac{1}{\pi} \cdot \frac{dx}{1+x^2} > \epsilon,$$

which would imply the desired statement.
Note that for $x \in [1-10\epsilon, 1+10\epsilon]$ and small enough ϵ, we have $2 \cdot \frac{1}{\pi} \cdot \frac{1}{1+x^2} \geq \frac{2}{\pi} \cdot \frac{1}{3} \geq 1/6$. Therefore,

$$\int_{1-10\epsilon}^{1} 2 \cdot \frac{1}{\pi} \cdot \frac{dx}{1+x^2} \geq \int_{1-10\epsilon}^{1} \frac{dx}{6} = \frac{10}{6} \cdot \epsilon > \epsilon,$$

and

$$\int_{1}^{1+10\epsilon} 2 \cdot \frac{1}{\pi} \cdot \frac{dx}{1+x^2} \geq \int_{1}^{1+10\epsilon} \frac{dx}{6} = \frac{10}{6} \cdot \epsilon > \epsilon.$$

(e) Let $k = C\log(1/\delta)/\epsilon^2$ as defined in part (c). Take ks independent samples of Λ : $\{X_i^{(t)}\}_{i \leq s, t \leq k}$. Now we keep k running sums $S_t = \sum_{i=1}^{s} a_i X_i^{(t)}$, and return the value $\text{median}(|S_1|, |S_2|, \cdots, |S_k|)$.
Note that the algorithm runs in sub-linear space: only keeps $k = C\log(1/\delta)/\epsilon^2$ values (if not considering the samples from Λ).
Now we are going to analyze the performance of the algorithm. Observe that each S_i is independently distributed as $\sum_{i=1}^{s} |a_i|\Lambda$. By part (c), we know that for an independent Λ, with probability at least $1-\delta$, we have

$$1/2 - \epsilon \leq \Pr\left[\left(\sum_{i=1}^{s} |a_i|\right) |\Lambda| \leq \text{median}(|S_1|, |S_2|, \cdots, |S_k|)\right] \leq 1/2 + \epsilon.$$

Now, by part (c), we know that $(1-10\epsilon)\left(\sum_{i=1}^{s} |a_i|\right) \leq \text{median}(|S_1|, |S_2|, \cdots, |S_k|) \leq (1+10\epsilon)\left(\sum_{i=1}^{s} |a_i|\right)$. I.e., the algorithm gives a $(1+O(\epsilon))$ approximation with probability at least $1-\delta$.

4. (a) For $(i_1, i_2) \neq (j_1, j_2)$, we have

$$\left\langle v^{(i_1,i_2)}, v^{(j_1,j_2)} \right\rangle = \sum_{a \in C} (-1)^{a_{i_1} + a_{i_2} + a_{j_1} + a_{j_2}}.$$

Note that by 4-wise independence of C, this value is 0 as long as there is an element (from $[n]$) which appears exactly once in i_1, i_2, j_1, j_2, while this is true for $(i_1, i_2) \neq (j_1, j_2)$ and $i_1 < i_2, j_1 < j_2$.

(b) For any set of coefficients $\{\alpha^{(i_1,i_2)}\}_{1 \leq i_1 < i_2 \leq n}$, we have

$$\|\sum_{i_1,i_2} \alpha^{(i_1,i_2)} v^{(i_1,i_2)}\|^2 = \sum_{i_1,i_2} \left(\alpha^{(i_1,i_2)}\right)^2 \|v^{(i_1,i_2))}\|^2 = n \cdot \sum_{i_1,i_2} \left(\alpha^{(i_1,i_2)}\right)^2,$$

where the first equality is because of part (a). Therefore, if $\sum_{i_1,i_2} \alpha^{(i_1,i_2)} v^{(i_1,i_2)} = \mathbf{0}$, we have $\alpha^{(i_1,i_2)} = 0$ for all $1 \leq i_1 < i_2 \leq n$. This means that the vectors $\{v_{i_1,i_2}\}_{1 \leq i_1 < i_2 \leq n}$ are linearly independent over reals.

(c) Since the vectors $\{v_{i_1,i_2}\}_{1 \leq i_1 < i_2 \leq n}$ are $|C|$-dimensional vectors. There can be at most $|C|$ of them. Therefore, we have $\binom{n}{2} \leq |C|$, i.e. $|C| = \Omega(n^2)$.