Sid Banerjee (sbanerjee@cornell.edu)

Problem 1: (Chernoff Bounds via Negative Dependence - from MU Ex 5.15)

While deriving lower bounds on the load of the maximum loaded bin when n balls are thrown in n bins, we saw the use of *negative dependence*. We now consider another example, where this technique can be used to derive Chernoff-style bounds for the number of empty bins.

Suppose n balls are thrown in n bins, and let $\{X_i\}_{i\in[n]}$ be a collection of indicator r.v.s indicating whether bin i is empty (i.e., $X_i = 1$ iff bin i has 0 balls). On the other hand, let $\{Y_i\}_{i\in[n]}$ be a set of i.i.d. Bernoulli r.v.s which are 1 with probability $(1 - 1/n)^n$.

Part (a)

For any $k \geq 1$, show that $\mathbb{E}[X_1X_2 \ldots X_k] \leq \mathbb{E}[Y_1Y_2 \ldots Y_k]$.

Solution: $X_1X_2 \ldots X_k = 1$ if and only if the first k bins are empty. Note that there are $(n-k)^n$ ways to throw n balls into $n-k$ bins, and there are n^n ways to throw n balls into n bins. Therefore, we have:

$$\mathbb{E}[X_1X_2 \ldots X_k] = \frac{(n-k)^n}{n^n} = \left(1 - \frac{k}{n}\right)^n.$$

Since $\{Y_i\}_{i\in[n]}$ are i.i.d. Bernoulli r.v.s which are 1 with probability $(1 - 1/n)^n$, we have:

$$\mathbb{E}[Y_1Y_2 \ldots Y_k] = \left(1 - \frac{1}{n}\right)^{kn}.$$

Now for any positive integers k, n, we have that $1 - \frac{k}{n} \leq \left(1 - \frac{1}{n}\right)^k$

Hence, $\mathbb{E}[X_1X_2 \ldots X_k] \leq \mathbb{E}[Y_1Y_2 \ldots Y_k]$.

Part (b)

Let $X = \sum_{i=1}^n X_i$ and $Y = \sum_{i=1}^n Y_i$. Using the above result, prove that for any $\theta \geq 0$, we have:

$$\mathbb{E}[e^{\theta X}] \leq \mathbb{E}[e^{\theta Y}]$$

Hint: Think of the Taylor series of the exponential function.

Solution: Using the Taylor series of the exponential function and linearity of expectation, we can write:

$$\mathbb{E}[e^{\theta X}] = \sum_{i=0}^{\infty} \frac{t^i}{i!}\mathbb{E}[X^i]$$

$$\mathbb{E}[e^{\theta Y}] = \sum_{i=0}^{\infty} \frac{t^i}{i!}\mathbb{E}[Y^i]$$

Now, let's show that for each i, $\mathbb{E}[X^i] \le \mathbb{E}[Y^i]$.
Let $Z = \sum_{i=1}^n Z_i$ be an indicator random variable which is the sum of i.i.d. indicators Z_i. Then,

$$\mathbb{E}[Z^j] = \mathbb{E}\left[\left(\sum_{i=1}^n Z_i\right)^j\right].$$

After expanding out the jth power, we get a sum of terms where each term is a product of various Z_i's to powers c_i, i.e. $Z_1^{c_1} Z_2^{c_2} \cdots Z_n^{c_n}$. Since the Z_i's are indicators, each term simplifies to:

$$Z_1^{c_1} Z_2^{c_2} \cdots Z_n^{c_n} = \prod_{i=1}^n Z_i^{I\{c_i \neq 0\}} = \prod_{i=1}^k Z_i,$$

where $I\{c_i \neq 0\}$ indicates that $c_i > 0$, and $k = \sum_{i=1}^n I\{c_i \neq 0\}$.
Noting that both X and Y satisfy the properties of Z, and using the result from part a), we can write:

$$\mathbb{E}[X^j] = \mathbb{E}\left[\left(\sum_{i=1}^n X_i\right)^j\right] = \sum_{c=(c_1,\cdots,c_n)} \mathbb{E}\left[\prod_{i=1}^k X_i^{c_i}\right] = \sum \mathbb{E}\left[\prod_{i=1}^k X_i\right] \le \sum \mathbb{E}\left[\prod_{i=1}^k Y_i\right] = \mathbb{E}[Y^j]$$

Which means that we have shown that $\mathbb{E}[e^{\theta X}] \le \mathbb{E}[e^{\theta Y}]$.

Part (c)

Finally, using this result, state a Chernoff bound for $\mathbb{P}[X \ge (1+\epsilon)\mathbb{E}[X]]$.
(You can use bounds you know from before without re-deriving them).

Solution: Note that $\mathbb{E}[X] = \sum \mathbb{E}[X_i] = \sum (1 - 1/n)^n = \mathbb{E}[Y]$. Now using Markov inequality and part b), we can write:

$$\mathbb{P}[X \ge x] \le \frac{\mathbb{E}[e^{tX}]}{e^{tx}} \le \frac{\mathbb{E}[e^{tY}]}{e^{tx}}.$$

If we take $x = (1+\epsilon)\mathbb{E}[X]$, and use the fact that $\mathbb{E}[e^{tY}] \le e^{\mathbb{E}[Y](e^t - 1)}$, we will get:

$$\mathbb{P}[X \ge (1+\epsilon)\mathbb{E}[X]] \le \frac{e^{\mathbb{E}[Y](e^t-1)}}{e^{t(1+\epsilon)\mathbb{E}[X]}}.$$

Therefore,

$$\mathbb{P}[X \ge (1+\epsilon)\mathbb{E}[X]] \le \left(\frac{e^{(e^t-1)}}{e^{t(1+\epsilon)}}\right)^{\mathbb{E}[Y]}.$$

Setting $t = \ln(1+\epsilon)$, we get:

$$\mathbb{P}[X \ge (1+\epsilon)\mathbb{E}[X]] \le \left(\frac{e^{\epsilon}}{(1+\epsilon)^{(1+\epsilon)}}\right)^{\mathbb{E}[Y]}.$$

Problem 2: (Bucket Sort)

Suppose we are given $n = 2^m$ elements, each of which are k-bit sequences drawn uniformly at random from $U = \{0,1\}^k$ (where $k \geq m$). We'll now consider a simple deterministic algorithm for sorting these, that takes $O(n)$ time on average. First, we place place each element in one of 2^m buckets, where the j^{th} bucket ($j \in \{0, 1, \ldots, 2^m - 1)$ is used to place all elements whose first m bits correspond to the number j. Next, we use any sorting algorithm with quadratic running time (for example, a simple bubble sort or insertion sort) to sort the elements in each bucket, and then merge the buckets. Prove that the expected running time of this algorithm is $O(n)$.
Hint: Recall the analysis of the FKS hashing scheme.

Solution: Let X_b be the number of elements that land in the b^{th} bucket. As we are using a sorting algorithm with quadratic running time, the time to sort the b^{th} bucket is then at most cX_b^2 for some constant c. The expected time spent sorting in the second stage is at most

$$\mathbb{E}\left[\sum_{b=1}^{2^m} cX_b^2\right] = c\sum_{b=1}^{2^m} \mathbb{E}[X_b^2].$$

On the other hand, for pairs of items i, j, let Y_{ij} be the indicator that they 'collide', i.e., fall in the same bucket (because they have the same first m bits). By the principle of deferred decisions, we have that $\mathbb{E}[Y_{ij}] = 1/2^m = 1/n$, and also that there are $\binom{n}{2} = \frac{n^2-n}{2}$ such pairs – thus $\mathbb{E}[\sum_{i,j} Y_{ij}] = \frac{n-1}{2}$. On the other hand, we also have that $\sum_{i,j} Y_{ij} = \sum_{b=1}^{2^m} \binom{X_b}{2}$ (since $n = 2^m$). Further, since $\sum_b X_b = n$, we can simplify to get $\sum_{i,j} Y_{ij} = \frac{\sum_{b=1}^{n} X_b^2 - n}{2}$. Taking expectation, we get:

$$\sum_{b=1}^{n} X_b^2 = 2n - 1$$

Thus the expected running-time is $\mathbb{E}\left[\sum_{b=1}^{2^m} cX_b^2\right] \leq 2cn = O(n)$.

Problem 3: (Open Addressing)

In class, we saw the *chaining* technique for designing hash tables for answering exact set-membership (i.e., without allowing for false-positives). Another common approach is that of *open-addressing*, where given a set S of m items, we hash the elements in a single array of length $> m$. Each entry in the array either contains an element from S, or is empty. The hash function defines for each element $x \in U$, a probe sequence $\{h(x,1), h(x,2), \ldots\}$. To insert an element x in the array, we first check position $h(x,1)$ – if this is occupied, we try to insert it in $h(x,2)$, and so on till we find an open cell in the array.

Part (a)

Suppose we use an array of length $2m$ to store m items, and suppose each hash-function $h(x,i)$ is independent and uniform over $\{0, 1, \ldots, 2m-1\}$. Show that for any of the first m elements to be inserted, the insertion required more than k probes with probability $\leq 2^{-k}$ – hence show that the probability that the i^{th} insertion (for $i \leq m$) took more than $2\log_2 m$ probes is less than $1/m^2$.

Solution: Note that even after the first m elements have been hashed, the number of unoccupied spots is m. Thus, for any of the first m elements, the probability that each hash function maps to an empty position is at least $1/2$ – moreover, since each probe is independent, the probability that the i^{th} insertion needs more than k probes is at most 2^{-k}. Now if we set $k = 2\log_2 m$ in this bound, then we get that for any of the first m elements, the probability that it needs more than $2\log_2 m$ probes is less than $1/m^2$.

Part (b)

Next, let X be the *maximum* number of probes required by an item during insertion of the first m items. Show that X is less than $2\log_2 m$ with probability at least $1 - 1/m$. Using this, also show that the $\mathbb{E}[X]$ is $O(\log m)$.

Solution: For any element $i \in \{1, 2, \ldots, m\}$, from above, we know that the probability it needs more than $2\log_2 m$ probes is less than $1/m^2$. By the union bound, we have that:

$$\begin{aligned}\mathbb{P}[X > 2\log_2 m] &= \mathbb{P}[\cup_{i=1}^m \{\text{Number of probes needed by } i^{th} \text{ element} > 2\log_2 m\}] \\ &\leq m\mathbb{P}[\text{Number of probes needed by } i^{th} \text{ element} > 2\log_2 m] \leq \frac{1}{m}\end{aligned}$$

Next, note that in the worst case, the number of probes required is m. Now, for $\mathbb{E}[X]$, we have:

$$\begin{aligned}\mathbb{E}[X] &\leq (2\log_2 m)\mathbb{P}[X \leq 2\log_2 m] + m\mathbb{P}[X > 2\log_2 m] \\ &\leq (2\log_2 m) \times 1 + m \times \frac{1}{m} = O(\log m)\end{aligned}$$

Problem 4: (Extensions of Bloom Filters)

In class we saw the basic Bloom filter, where we used k independent random hash-functions $\{h_1, h_2, \ldots, h_k\}$ to hash a set S of m elements into an array A of n bits. Recall that in order to get a false-positive rate of $\delta = O(1)$, we chose $n = cm$, for some constant c , and $k\ c\ln 2$ (in particular, for false-positive rate of 2%, we used $c = 8$ and $k = 6$). We now see how this basic structure can be modified in various ways.

Part (a)

In order to support item deletions in addition to insertions and look-ups, we can replace each bit $A[i]$ in A with a counter – when an element is hashed to bucket i, we increment $A[i]$, and to delete an element x, we decrement the counter for each $A[i]$ corresponding to $\{h_1(x), h_2(x), \ldots, h_k(x)\}$. As before, if we use $n = O(m)$ and *fixed-size counters* of b-bits. What is the probability that counter $A[i]$ overflows after inserting m elements? Also argue that $O(\log\log m)$-bit counters are necessary and sufficient to prevent overflow in any counter (with high probability).

Solution: Note that a b bit counter can count up to 2^b elements. For counter $A[i]$ overflow, we need at least 2^b+1 elements to be hashed to bucket i – this happens with probability $\sum_{k=2^b+1}^{m} \binom{m}{k} \frac{1}{(cm)^k} \left(1 - \frac{1}{cm}\right)^{m-k}$ For $m >> 2^b$, we can use $\binom{m}{k} \approx m^k/k!$ to get that the probability of overflow $\approx \frac{1}{c^{2^b}(2^b)!}$.

Moreover, from class you know that if m balls are dropped uniformly at random into $\theta(m)$ bins, then with high probability, the maximum loaded bin has $\Theta\left(\frac{\log m}{\log\log m}\right)$ balls. Thus the bucket with the maximum number of hashed items has $\Theta\left(\frac{\log m}{\log\log m}\right)$ items, and hence needs a counter of size $\Theta\left(\log\left(\frac{\log m}{\log\log m}\right)\right) = \Theta\left(\log\log m\right)$ bits.

Part (b)

Suppose we use the same hash functions $\{h_1, h_2, \dots, h_k\}$ to hash two separate sets S_1 and S_2 (both of size m) – let the resulting Bloom filters (each of n bits) be A_1 and A_2 respectively. Suppose we create a new Bloom filter A_{OR} by taking the bit-wise OR of the bits of A_1 and A_2. Is this the same as the Bloom filter constructed by adding the elements of $S_1 \cup S_1$ one at a time?

Solution: Recall that in a Bloom filter, every cell is set to 1 if any of the elements are hashed to it – this can be thought of representing each element x in terms of a fingerprint, which has 1s in all the k positions where x is hashed, and then taking the OR of all the fingerprints. Thus, taking the bitwise OR of two Bloom filters obtained from S_1 and S_2 does give the same Bloom filter as that created by adding each element of $S_1 \cup S_2$.

Part (c)

Suppose we create another new Bloom filter A_{AND} by taking the bit-wise AND of the bits of A_1 and A_2. Argue that this is not the same as the Bloom filter constructed by adding the elements of $S_1 \cap S_2$ one at a time. However, also argue that A_{AND} can be used to check if $x \in S_1 \cap S_2$ with one-sided error (i.e., give an algorithm that always returns TRUE if $x \in S_1 \cap S_2$), and explain how we can get false-positives.

Solution: First, note that since we use the same hash functions $\{h_1, h_2, \dots, h_k\}$, hence for any $x \in S_1 \cap S_2$, the positions corresponding to $h_1(x), h_2(x), \dots, h_k(x)$ are set to 1 in both A_1 and A_2, and thus in the bitwise-AND of A_1 and A_2. However, additional positions in A_{AND} may also be falsely set to 1 – in particular, a position b, where $b \neq h_i(x)$ for any $x \in S_1 \cap S_2$ and $i \in [k]$, can be set to 1 if there exists elements $y \in S_1 \setminus S_2$, $z \in S_2 \setminus S_1$ such that $h_i(y) = h_j(z)$ for some $i, j \in [k]$. Note that there are $|S_1 \setminus S_2| \times |S_2 \setminus S_1|$ pairs of such false collisions, and each one may collide with probability $1/n$.

Problem 5: (Similarity functions with no linear-LSH family)

In class we discussed locality sensitive hashing for the Hamming and Jaccard similarity functions Recall that for a ground set $\mathcal{U}$ and subsets $A, B \subseteq \mathcal{U}$, these two distances corresponded to:

$$s_{Hamming}(A,B) = 1 - \frac{A\Delta B}{|\mathcal{U}|} \quad , \quad s_{Jaccard}(A,B) = \frac{|A\cap B|}{|A\cup B|},$$

where $A\Delta B$ is the symmetric difference between sets A and B (i.e., $A\Delta B = (A\cup B)\setminus(A\cap B)$) Moreover, in both cases, we obtained families of hash-functions H satisfying:

$$\mathbb{P}[h(x) = h(y)] = s(x,y)$$

A natural question to ask is if such *linear-LSH families* exists for other similarity functions, in particular, for two other natural subset-similarity measures – the *Overlap* and *Dice* similarities:

$$s_{Overlap}(A,B) = \frac{|A\cap B|}{\min\{|A|,|B|\}} \quad , \quad d_{dice}(A,B) = \frac{2|A\cap B|}{|A|+|B|}$$

Part (a)

As in class, suppose we define a distance function $d : \mathcal{U}\times\mathcal{U} \to [0,1]$ corresponding to a similarity function as $d(x,y) = 1 - s(x,y)$. Show that for a given similarity function s, if we have a linear-LSH family H, i.e., whose hash functions satisfy $\mathbb{P}[h(x) = h(y)] = s(x,y)$, then the distance functions must obey the triangle inequality, i.e., for any $x,y,z \in \mathcal{U}$, we must have:

$$d(x,y) + d(y,z) \geq d(x,z)$$

Solution: Consider x,y,z distinct elements in $\mathcal{U}$. Note that we have $\mathbb{P}[h(x) \neq h(y)] = d(x,y)$, and similarly for x,z and y,z. Now we have:

$$\begin{aligned}
\mathbb{P}[h(x) \neq h(y)] &= \mathbb{P}[h(x) \neq h(y), h(x) = h(z)] + \mathbb{P}[h(x) \neq h(y), h(x) \neq h(z)] \\
&\leq \mathbb{P}[h(y) \neq h(z)] + \mathbb{P}[h(x) \neq h(z)] \\
&= d(y,z) + d(x,z)
\end{aligned}$$

Part (b)

Using the above result, prove that the Overlap and Dice similarity functions can not have a linear-LSH family.

Solution: We just need to show via examples that the Overlap and Dice similarities do not obey the triangle inequality. For example, consider the ground set $\mathcal{U} = \{1,2,\ldots,8\}$, and the sets $A = \{1,2,3,4,5\}, B = \{1,2,6,7,8\}, C = \{1,2,3,4,6,8\}$. Now we have $s_{Overlap}(A,B) = 2/5, s_{Overlap}(B,C) = s_{Overlap}(A,C) = 4/5$, and hence $d_{Overlap}(A,B) = 3/5 > 1/5 + 1/5 = d_{Overlap}(B,C) = d_{Overlap}(A,C)$. Similarly, we have $s_{Dice}(A,B) = 2/5, s_{Dice}(B,C) = s_{Dice}(A,C) = 8/11$, and hence $d_{Dice}(A,B) = 3/5 > 3/11 + 3/11 = d_{Dice}(B,C) = d_{Dice}(A,C)$