COMPSCI 514: Problem Set 2

Released: 9/28.

Due: Friday 10/11 by 8:00pm in Gradescope. Submissions will be accepted until Sunday 10/13 at Midnight, without penalty.

Instructions:

- Each group should work together to produce a single solution set. One member should submit a solution pdf to Gradescope, marking the other members as part of their group.
- You may talk to members of other groups at a high level about the problems but not work through the solutions in detail together.
- You must show your work/derive any answers as part of the solutions to receive full credit.

1. Some Useful Inequalities (6 points)

There are a number of very useful inequalities that come up over and over again in randomized and approximation algorithm analysis, and more generally in statistical/probabilistic analysis that are worth knowing well.

1. (1 point) Show that for any $\epsilon \in [0,1)$, $\frac{1}{1-\epsilon/2} \leq (1+\epsilon)$.
2. (1 point) Show that for any $\epsilon \geq 0$, $\frac{1}{1+\epsilon} \geq (1-\epsilon)$.
3. (1 point) Show that for any $\epsilon \in [0,1)$, $(1+\epsilon)^2 \leq 1+3\epsilon$.
4. (1 point) Show that for any $\epsilon \in [0,1)$ and any integer $t \geq 1$, $(1-\epsilon)^t \geq 1-t\epsilon$.
5. (2 points) For *any* x, $1+x \leq e^x$. Use this to show that for any $x, c, b > 0$ with $\frac{c}{x} \leq 1$, $\left(1-\frac{c}{x}\right)^{x \cdot b} \leq e^{-c \cdot b}$. Use this show that, if I flip a random coin which is heads with probability $1/k$ and tails with probability $1-1/k$, that the probability I see at least $c \cdot k \ln k$ tails in a row is $\leq \frac{1}{k^c}$ for any $c \geq 0$.

2. Set Similarities and Locality Sensitive Hash Functions (11 points)

You would like to use shingling and locality sensitive hashing to identify possible plagiarism in student essays. One possibility is to compare an essay A with a publication B in your database using shingling and Jaccard similarity. Another possibility is to use shingling, but then to measure similarity with cosine similarity, where the shingle sets are viewed as binary vectors.

1. (1 point) Given an essay A with 1000 words in it an a book B with 10000 words in it, what is the maximum number of 7 word shingles that may appear in A? What is the maximum that may appear in B?

In the following questions, consider an essay A with 1000 unique shingles in it and a publication B with 3000 unique shingles, let S_A and S_B be the shingle sets for A and B respectively. So $|S_A| = 1000$ and $|S_B| = 3000$. Assume the essay A was fully copied from a portion of B.

2. (1 point) What is the Jaccard similarity between these sets? What is $\Pr(MinHash(S_A) = MinHash(S_B))$?

3. (1 point) Let m be the number of all possible shingles. Represent S_A and S_B by length m binary vectors x_A and x_B, with a 1 in every position corresponding to a shingle they contain. What is the cosine similarity between x_A and x_B (again assuming A was fully copied from a portion B)? What is $\Pr(SimHash(x_A) = SimHash(x_B))$?

   Hint: Use that for any two vectors z, w, $\Pr(SimHash(z) = SimHash(w)) = 1 - \frac{\theta}{\pi}$ where θ is the angle between z and w in radians.

4. (2 points) The number of all possible shingles m will generally be huge. Describe why $SimHash(x_A)$ and $SimHash(x_B)$ can be computed in $O(|S_A|)$ and $O(|S_B|)$ time respectively instead of $O(m)$ time. How does this compare to the big-O runtime of MinHash?

5. (2 points) Consider another essay C also with 1000 unique shingles that is not copied and only shares 100 shingles with B. Compute $\Pr(MinHash(S_C) = MinHash(S_B))$ and $\Pr(SimHash(x_C) = SimHash(x_B))$.

6. (3 points) For both MinHash and SimHash, find a signature length r and repetition parameter t such that the fully copied essay A is identified with LSH-based similarity search with probability $\geq .95$ and the non-copied essay C is identified with probability $\leq .05$. Focus on minimizing the space complexity (i.e., the number of hash tables t used). By 'identified', we mean that the essay falls in the same bucket as B in at least one of the t hash tables.

   Hint: It may be helpful (although is not required) to write a very simple program to help solve this one.

7. (1 point) Given the above, which similarity metric and hash function would you pick for the plagiarism detection task?

3. A Better Count-Min Sketch (5 points)

Consider the count-min sketch algorithm: when processing a stream of inputs $x_1, \ldots, x_n$, we store t length-m count arrays $A_1, \ldots, A_t$ and chose t random hash functions $\mathbf{h}_1, \ldots, \mathbf{h}_t$ mapping the universe of possible items to $[m]$. When input x_i comes in we increment each count $A_1[\mathbf{h}_1(x_i)], \ldots, A_t[\mathbf{h}_t(x_i)]$. Once the stream has been processed, we estimate the frequency of any element x in the stream, denoted $f(x)$, by $\tilde{f}(x) = \min_{j \in [t]} A_j[\mathbf{h}_j(x)]$.

1. (4 points) Consider a variation on count-min sketch: instead of incrementing each counter $A_1[\mathbf{h}_1(x_i)], \ldots, A_t[\mathbf{h}_t(x_i)]$ when x_i comes in, we compute $M = \min_{j \in [t]} A_j[\mathbf{h}_j(x_i)]$. Then we only increment $A_j[\mathbf{h}_j(x_i)]$ if $A_j[\mathbf{h}_j(x_i)] = M$. Show that the estimate output by this variation *is at least as good or better* than the estimate of the count-min sketch algorithm presented in class.

2. (1 point) Why might you still use the original count-min sketch algorithm? Note that the updates for the original version are slightly faster, but we're looking for a more significant reason.

4. Randomized Dimensionality Reduction for Clustering (10 points)

One of the most popular clustering objectives is k-*means* clustering. Given a set of n data points $X = \{x_1, \ldots, x_n\}$ all in $\mathbb{R}^d$ the goal is to partition $[n]$ into k sets (clusters) $\mathcal{C} = \{C_1, \ldots C_k\}$ minimizing:

$$cost(\mathcal{C}, X) = \sum_{j=1}^{k} \sum_{i \in C_j} \|x_i - \mu_j\|_2^2 \tag{1}$$

where $\mu_j = \frac{1}{|C_j|} \sum_{i \in C_j} x_i$ is the *centroid* of cluster C_j (i.e., the mean of the points in that cluster.)

1. (3 points) Show that $cost(\mathcal{C})$ can be equivalently written as:

   $$cost(\mathcal{C}, X) = \sum_{j=1}^{k} \frac{1}{2|C_j|} \sum_{i_1 \in C_j} \sum_{i_2 \in C_j} \|x_{i_1} - x_{i_2}\|_2^2. \tag{2}$$

   Intuitively, what does this reformulation of the cost function mean?

   Hint: Show that both (1) and (2) can be rewritten as $\sum_{j=1}^{k} \left[\left(\sum_{i \in C_j} \|x_i\|_2^2 \right) - |C_j| \cdot \|\mu_j\|_2^2 \right]$. This will require some vector algebra. It will be helpful recall use that for any vector z, $\|z\|_2^2 = \sum_{i=1}^{d} z(i)^2 = \langle z, z \rangle$, as well as to use the linearity of inner product.

2. (4 points) Suppose that $\mathbf{\Pi} \in \mathbb{R}^{m \times d}$ is a random projection matrix with each entry chosen independently as $\frac{1}{\sqrt{m}} \mathcal{N}(0, 1)$. For each x_i in the dataset, let $\tilde{x}_i = \mathbf{\Pi} x_i$ and let $\tilde{X} = \{\tilde{x}_1, \ldots, \tilde{x}_n\}$ denote our set of sketched data points in $\mathbb{R}^m$.

   Given parameters $\epsilon, \delta \in (0, 1)$ how large must we set m so that, with probability $\geq 1 - \delta$, for all clusterings $\mathcal{C}$,

   $$(1 - \epsilon) cost(\mathcal{C}, X) \leq cost(\mathcal{C}, \tilde{X}) \leq (1 + \epsilon) cost(C, X).$$

   That is, how much can we reduce the dimension (from d to m) and still approximately preserve all cluster costs with high probability. Give the answer in big-O notation.

3. (3 points) Use the above to show how large we must set m such that, if we find an *optimal clustering* for the dimension reduced dataset $\tilde{x}_1, \ldots, \tilde{x}_n$, then with probability $\geq 1 - \delta$ this clustering will have at most $(1 + \epsilon)$ of the optimal cost on the original data set $x_1, \ldots, x_n$. Again, use big-O notation in your answer.