“Industrial” sketching

Piotr Indyk
MIT
Sketching

• A.k.a. simultaneous communication model
  – Alice and Bob send messages to Referee
  – Referee reports the output

• Almost all streaming algorithms can be used for sketching:
  – $L_2$, $L_0$, $L_1$, $L_k$ norms
  – Heavy hitters/sparse approximations/comp. sensing
  – Geometric MST, matching, etc
  – Core-sets
  – Sparse certificates
Industrial Application I

To be or not to be ...

(... , 1, ..., 4, ..., 2, ..., 2, ...)
(... , 6, ..., 1, ..., 3, ..., 6, ...)
(... , 1, ..., 3, ..., 7, ..., 5, ...)
(... , 2, ..., 2, ..., 1, ..., 1, ...)
Industrial Application II

• Google’s Map-Reduce
  – Data element → (key, value)
  – Aggregate all keys together
  – For each key, compute the “sum” of the values
Today

• Two sketching algorithms:
  – Min-wise hashing [Broder et al, ‘97-’98]
  – Random hyper-plane [Goemans-Williamson’94, Charikar’02]
  – Nice comparison in [Henzinger’07]

• Application to lower bounds
Min-wise hashing

• In many applications, the vectors tend to be quite sparse (high dimension, very few 1’s)
• Easier to think about them as sets
• For two sets \( A, B \), define the Jaccard coefficient:
  \[
  J(A,B) = \frac{|A \cap B|}{|A \cup B|}
  \]
  – If \( A=B \) then \( J(A,B)=1 \)
  – If \( A, B \) disjoint then \( J(A,B)=0 \)
• How to compute short sketches of sets that preserve \( J(.) \) ?
Hashing

• Mapping:
  \[ g(A) = \min_{a \in A} h(a) \]
  where \( h \) is a random permutation of the elements in the universe

• Fact: \( \Pr[g(A) = g(B)] = J(A, B) \)

• Proof: Where is \( \min( h(A) \cup h(B) ) \) ?
Min Sketching

• Define $Sk(A)=(g_1(A), g_2(A), \ldots, g_t(A))$
• By Chernoff bound, for $t=C \log(1/P)/ [J(A,B) \varepsilon^2]$ we have
  $$1-||Sk(A) - Sk(B)||_0 / t = J(A,B) (1 \pm \varepsilon)$$
  with probability at least $1-P$

• Another application: approximate near neighbor
Random hyperplane

• Let \( u, v \) be unit vectors in \( \mathbb{R}^n \)
• Angular distance:
  \[ A(u, v) = \text{angle between } u \text{ and } v \]
• Sketching:
  – Choose a random unit vector \( r \)
  – Define \( s(u) = \text{sign}(u^*r) \)
Probabilities

• What is the probability of \( \text{sign}(u*r) \neq \text{sign}(v*r) \) ?
• It is \( \frac{A(u,v)}{\pi} \)
• Usual Chernoff:
  – Set \( t = C \log(1/P)/[A(u,v) \varepsilon^2] \)
  – Define \( S_k(u) = [s_1(u), \ldots, s_t(u)] \)
  – With prob. \( 1-P \) we have
    \[ ||S_k(u) - S_k(v)||_0 / t = A(u,v) (1 \pm \varepsilon)/\pi \]
Applications to Lower Bounds

• Gap Dot Product Recap
  – (Gap) parameter $\Delta = 1/(m/2)^{1/2}$
  – Alice: a vector $u \in \mathbb{R}^m$, $\|u\|_2 = 1$
  – Bob: a vector $v \in \mathbb{R}^m$, $\|v\|_2 = 1$
  – Goal:
    • If $u^*v = 0$, return 0
    • If $u^*v \geq \Delta$, return 1

• Theorem: the randomized one-round c.c. of GDP is $\Omega(m)$
Angular formulation

• Conditions:
  – If $u^*v=0$ then $A(u,v)=\pi/2$
  – If $u^*v\geq \Delta$ then $A(u,v)<\pi/2-c\Delta$
    (c=c(\Delta)\approx 1$ for small $\Delta$)

• Sketching with $t=O(1/\Delta^2)$
  – If $u^*v=0$ then $||Sk(u)-Sk(v)||_0/t=1/2 \pm \Delta/8$
  – If $u^*v\geq \Delta$ then $||Sk(u)-Sk(v)||_0/t= 1/2-\Delta/\pi \pm \Delta/8$

• Corollary: the randomized one-round c.c. of $(1+\Delta/8)$-approximating the Hamming distance between vectors $x,y\in\{-1,1\}^{O(m)}$ is $\Omega(m)$

• Corollary II: $(1+1/m^{1/2})$-approximating $L_0$ norm in a stream requires $\Omega(m)$ space
  [Woodruff’04, Jayram-Kumar-Sivakumar ’07]
A bonus “war story”

• Once upon a time (in 1999), we (A+T+P) used min-hashing + LSH to cluster a large set of web pages
  – Documents = sets of words
  – Cluster together pairs of similar documents

• Problem: the home page of T’s advisor got clustered with porn websites

• Problem II: our algorithm was provably correct – the probability of failure was $10^{-6}$ (we calculated it exactly)
What happened?

• Implementation:
  – Consider a word $x$
  – We implemented $g(A) = \min_{a \in A} h(a)$ using
    $h(x) = (ax \mod P) \mod 2^8$
    • $P = 2^{64} - 59$ (more or less)
    • $a$ randomly chosen
  – To speed up the process, we kept only words $x$ which were
    divisible by 8

• What happened?
  – Implementation bug: $ax$ was computed modulo $2^{64}$
  – $\mod P$ had essentially no effect
  – $x$ divisible by 8 $\Rightarrow$ $(ax)$ divisible by 8 $\Rightarrow$ $(ax) \mod 2^8$ divisible by 8
  – 3 lowest bits of $h(x)$ were zero, so the actual range was $2^5$ not $2^8$
  – Enough for word collisions to occur…
Moral

• Do your hashing right, or you might never graduate …