Lower Bounds

T.S. Jayram (IBM Almaden)

MADALGO Summer School
Lecture I

Algorithms for Massive Data Sets

- Traditionally, “efficient” computation is identified with polynomial time
  - P vs NP
  - Clearly, not adequate for massive data sets
- Is there a simple characterization of efficient computation over massive datasets?

A Single Theory?

- Modern computing systems are complex and varied
  - Memory + I/O architectures
  - Distributed computing e.g. Map-Reduce
  - Randomization
  - Etc.

- Difficult to capture all these aspects in a single model

Many Paradigms

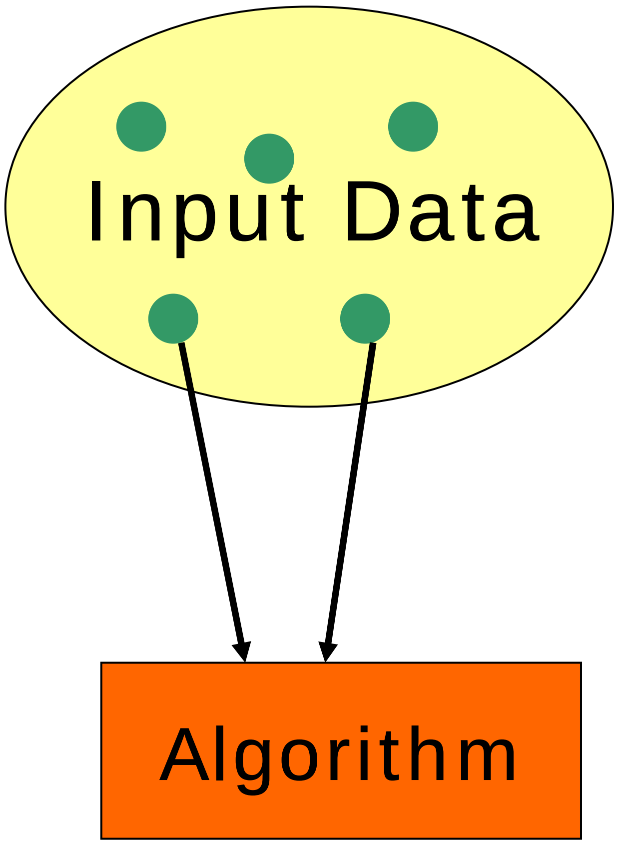
- Sampling
- Sketching
- Data Streams
- Read-write streams
- Stream-sort
- Map-reduce
- External memory algorithms

… and many more yet to come!

Lower Bounds

- This is a fertile ground for proving unconditional results
- Many successes ☺
- Certain problems seem to be fundamental
- Reductions play a big role

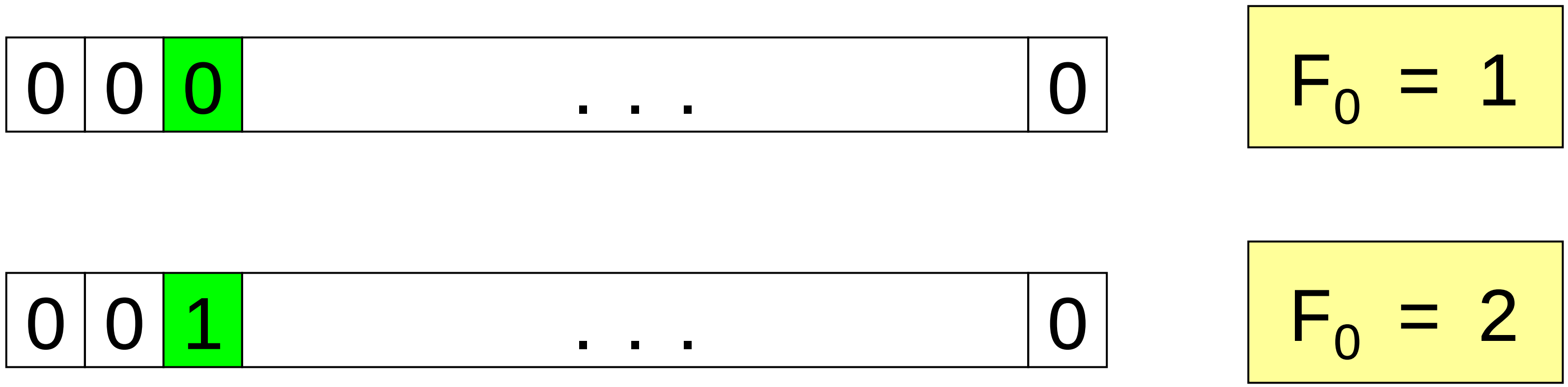
Sampling



- Query input at random locations
- Can query adaptively
- Key Measure: #queries

Warm-up: distinct elements (F_0)

- "Needle-in-a-haystack"
- Create 2 inputs



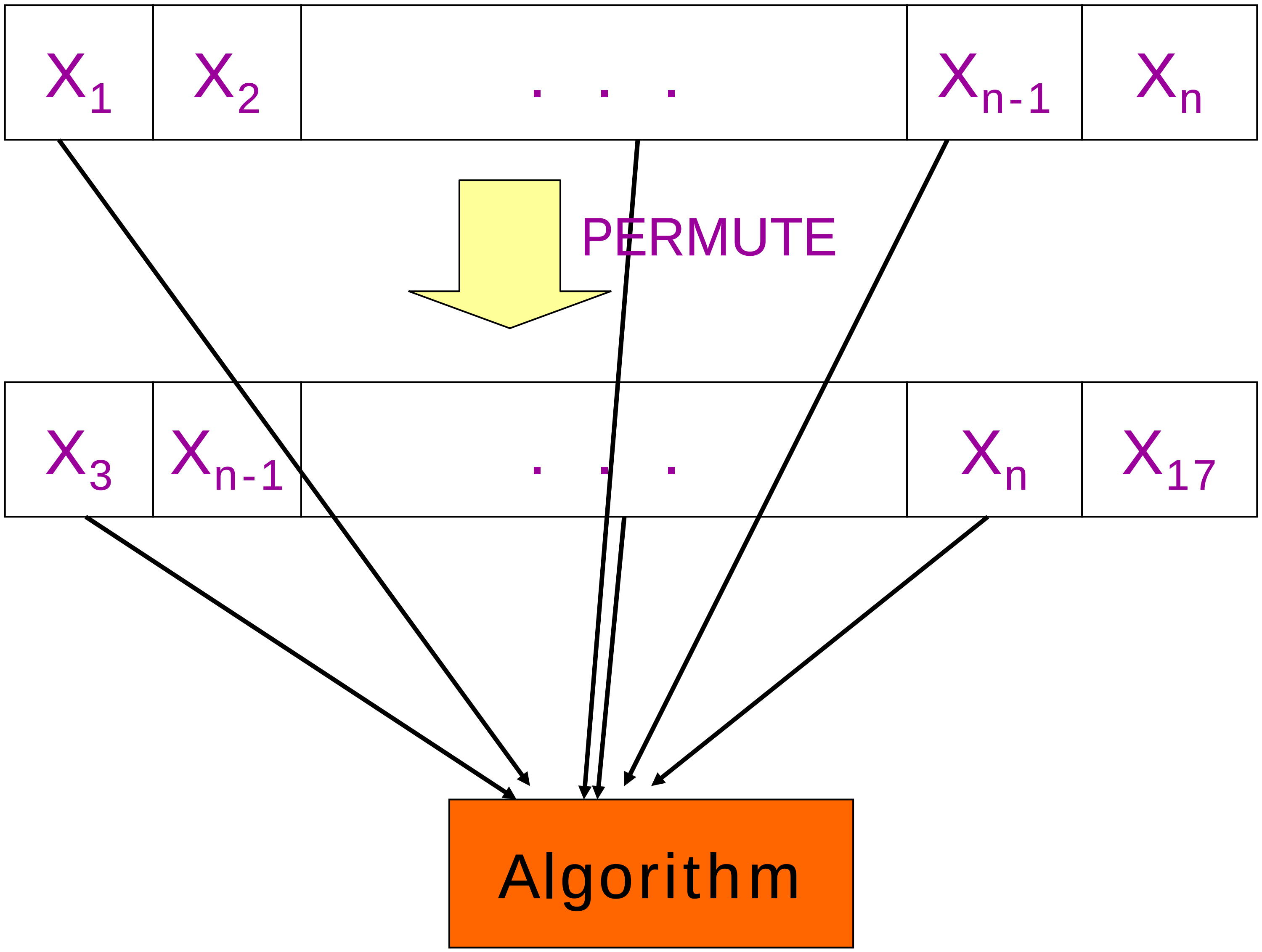
- 2-approximation needs space $\Omega(n)$

Sampling LBs for Symmetric Functions

Theorem [Bar-Yossef, Kumar, Sivakumar]

For symmetric functions, uniform sampling is the best possible.

Proof



Lower Bounds for Uniform Sampling

Tools:

- block sensitivity — Combinatorics [Nisan]
- Hellinger distance — Statistics [Bar-Yossef et al.]
- Kullback-Leibler divergence
- Jensen-Shannon divergence

(Kullback-Leibler divergence and Jensen-Shannon divergence:) Information theory [Bar-Yossef]

Example

- Find the mean of n numbers in $[0,1]$
- Exercise: Show that $O(1/\varepsilon^2)$ samples suffice to approximate mean additively within ε
- Lower Bound proof using Hellinger distance

Step 1: Approximation ➔ Promise

- Let
  - a : $\frac{1}{2} + \varepsilon$ 0's and $\frac{1}{2} - \varepsilon$ 1's
  - b : $\frac{1}{2} - \varepsilon$ 0's and $\frac{1}{2} + \varepsilon$ 1's
  - Promise: Input $x \in \{a,b\}$

- Any sampling algorithm for Mean can distinguish whether $x=a$ or $x=b$
  - as long as additive error is $\varepsilon/4$

Step 2: Create distributions

- P_a: distribution induced by taking a uniform sample from input a
- P_b: sampling uniformly from input b

- Compute Hellinger distance $h^2(P_a,P_b)$
  - For discrete distributions P, Q

    $$h^2(P,Q) = (½) \|\sqrt{P} - \sqrt{Q}\|^2$$

    $$= (½) \Sigma_x (\sqrt{P(x)} - \sqrt{Q(x)})^2$$

  - $h(P_a,P_b) = O(\varepsilon)$ (Exercise)

Lower bound via Hellinger Distance

Theorem.

Any uniform sampling algorithm needs

$$k=\Omega(1/\varepsilon^2)$$

samples to distinguish input a from input b

Proof

- Initially: $h(P_a,P_b) = O(\varepsilon)$
- Finally: $h((P_a)^k,(P_b)^k) = \Omega(1)$

  [By distinguishability]
- Key Idea: multiplicative property of Hellinger distance

  $1 - h^2(P^k,Q^k) = (1 - h^2(P,Q))^k$

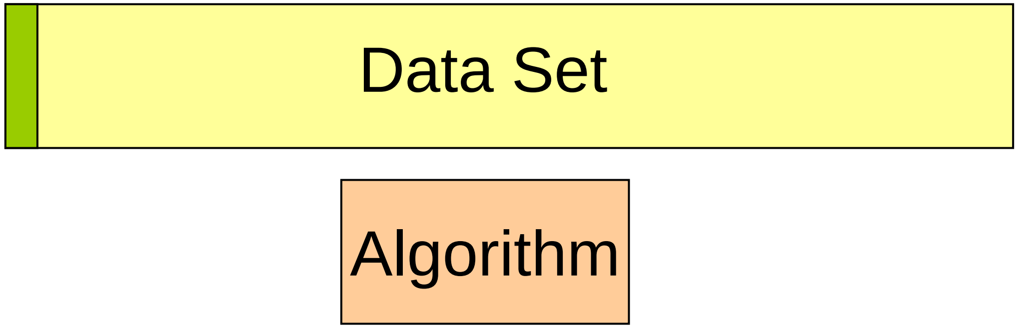
  (Exercise)
- So $O(1) = (1- O(\varepsilon^2))^k \rightarrow k = \Omega(1/\varepsilon^2)$

Summary

- Identify 2 hard inputs such that
  - The outputs are different
  - Sampling from the inputs creates close distributions
- Apply Theorem to get LB on samples
- Extensions [Bar-Yossef] : more than 2 inputs, multi-output answers etc.

Data Streams

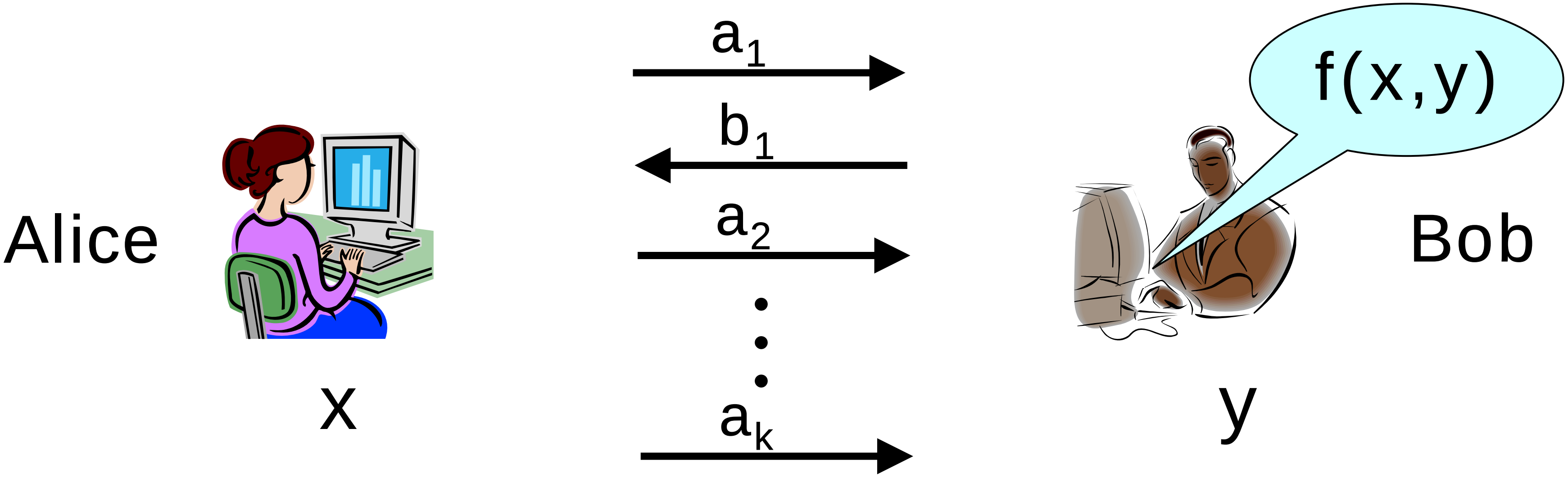
- Stream through the data in a one-way fashion
  - limited main memory storage
  - Also allow multiple passes



Lower Bounds for Data Streams

- Idea is to somehow bound the flow of information (yields space lower bounds)
- Model is too fine-grained to prove lower bounds directly
- Instead, we consider more powerful models (hopefully simpler to tackle)

Communication complexity (C.C.)



Resources:

\# bits = $\sum_i |a_i| + \sum_j |b_j| + |f(x,y)|$

\# rounds

See book by Kushilevitz & Nisan

Extensions to multiple parties

Transcripts

- Issue: Answer is too long!
- Solution: let last player output some more bits instead of the answer
  - Contributes to bit cost
  - Does not increase #rounds
- Transcript: string describing the entire communication + last player's output
  - Output is a function of the transcript alone

Data Streams → C.C.

Theorem.

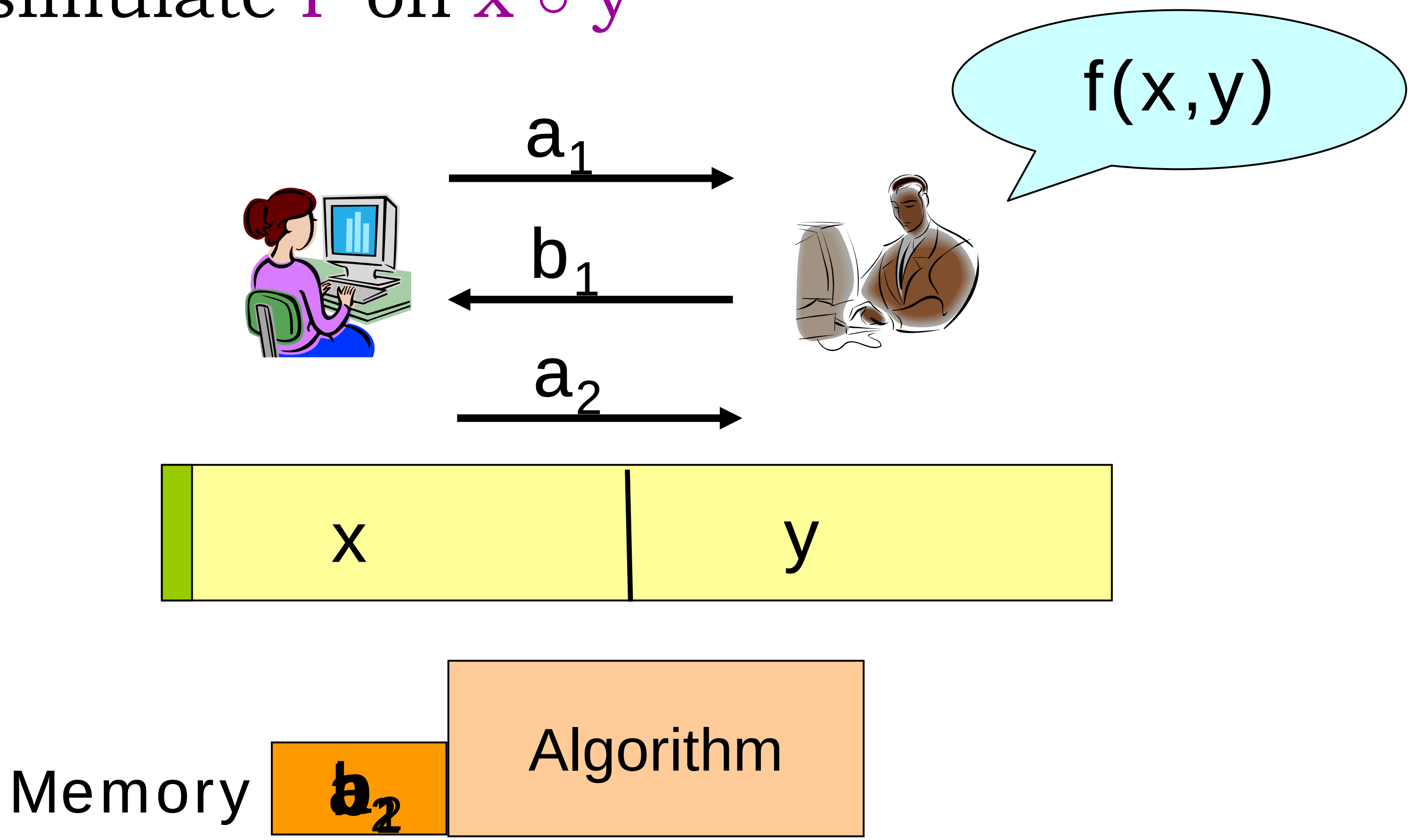
Data stream algorithm for $f(x \circ y)$
Space s
Passes k

⇓

C.C. protocol for $f(x,y)$
Bits $O(2ks)$
Rounds $2k-1$

Proof

- Alice gets x and Bob gets y
- Given data stream algorithm P, Alice and Bob simulate P on $x \circ y$



One-pass Data Stream

- Data stream algorithm for $f(x \circ y)$
  - Space s

➔ $O(s)$, 1-round protocol for $f(x,y)$

- One-round communication protocols are worthy of study!

Caveat

- C.C. usually deals with decision problems
- Data stream problems involve approximate computations
- Usual reduction techniques yield promise problems in C.C.

The Equality Function

- $EQ: U \times U \rightarrow \{0,1\}$
- $EQ(x,y) = 1$ iff $x = y$

Theorem.

Deterministic C.C. of EQ equals log |U|

Proof Warmup: One-way

- Suppose Alice sends fewer than $\log|U|$ bits
- #messages of Alice $< 2^{\log|U|} = |U|$
- By pigeonhole principle, there exist distinct $x, x' \in U$ s.t. Alice sends the same message for both x and x'
- Suppose Bob's input is x.
- Then protocol gives same answer on both (x,x) and (x',x).
- Contradiction.

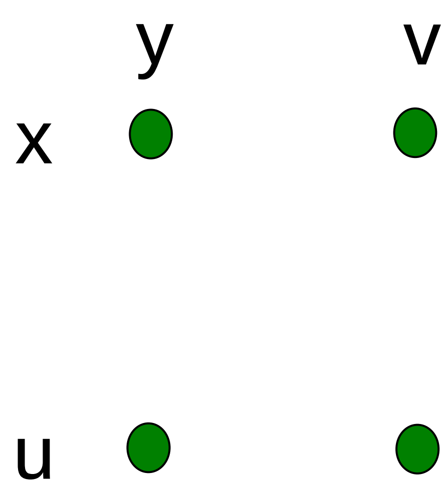
Proof for General Protocols

Fundamental Theorem of C.C.

Let P(x,y) denote the transcript of a det. communication protocol P. Then,

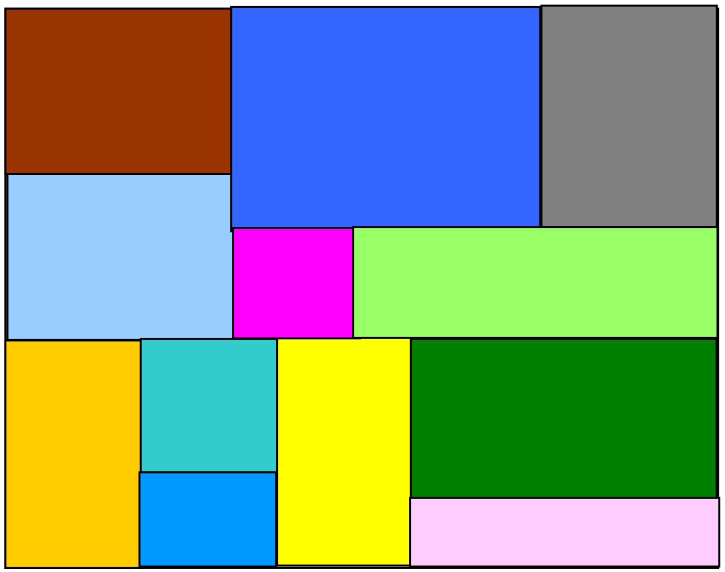
$$P(x,y) = t = P(u,v)$$

$$\Rightarrow P(x,v) = t = P(y,v)$$



Rectangle Property of C.C.

- View $P(x,y)$ as a matrix of transcripts
  - Rows/Columns indexed by inputs to Alice/Bob resp.
- Every transcript is a combinatorial rectangle in P
  - of the form $A \times B$
  - A: subset of rows
  - B: subset of columns



Fooling Set Method for EQ

- Consider the set of inputs
  $F = \{ (x,x) : x \in \{0,1\}^n \}$ (YES instances)

- No two inputs in F can generate the same transcript in a protocol P. Why?
- Suppose $P(x,x) = t = P(x',x')$, $x \neq x'$
- By fundamental theorem, $P(x,x') = t$
- Protocol errs on (x,x'). Contradiction!

- # of transcripts $\geq 2^n$

Gap Hamming Distance (GHD)

- $x, y \in \{0,1\}^n$
- $|x| = |y| = n/2$

Promise problem (with parameter $\Delta > 0$):

- $GHD_\Delta(x,y)$
  $= 1$ if $d_H(x,y) \geq (1 + \Delta)n/2$
  $= 0$ if $d_H(x,y) \leq n/2$

Exercise: Show the connection between GHD_Δ and distinct elements (F_0)

Reduction from EQ to GHD

- Idea: use a binary error-correcting code
  - Encoder E maps n bits to $N = \Theta(n)$ bits
  - Each codeword has weight $N/2$
  - Relative distance $\Delta = \Theta(1)$
  - Such codes exist; need not be constructive!

- Given inputs x,y to EQ
  - Construct $x' = E(x) \circ 0^{N/2}\ 1^{N/2}$
  - Construct $y' = E(y) \circ 1^{N/2}\ 0^{N/2}$

  - $|x'| = |y'| = N$
  - $d_H(x',y')$ is either N or $(1+\Delta)N$

  - → $GHD_\Delta(x',y')$ satisfies the right properties

Summary

- Deterministic LBs are easier to handle*
- LB problem gets considerably harder for randomized data stream algorithms
  - Randomization is powerful
  - Exercise: show O(1)-bit protocol for equality (Hint: Use error-correcting codes)
- Will see how to handle randomized protocols in the next lecture

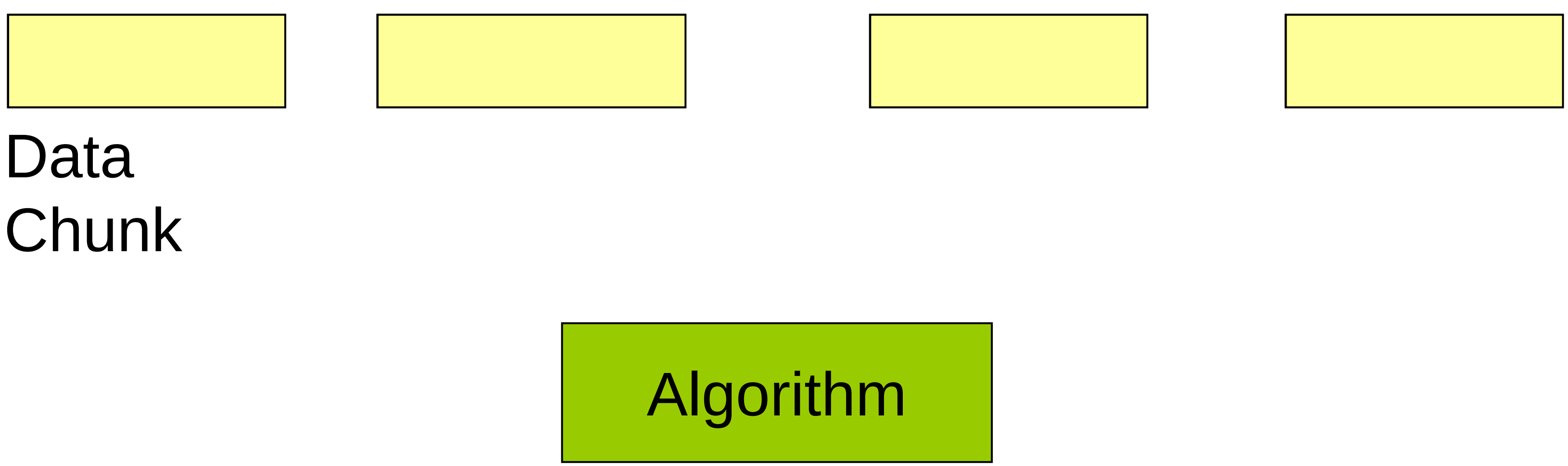
Summary

- C.C. is a well-developed field with many tools and ideas, so offers hope for streaming LBs

- But the problems that arise from streaming are difficult
  - promise problems
  - randomized computation

Sketching

- Function-specific data compression
- Lossy data compression
  - function is usually only approximable
- Data is distributed over several chunks
  - Chunks are compressed into small sketches
  - Function is computed over the sketches



Indexing (IND)

- Input: a binary string x of length n

- Can we sketch it so that any bit can be retrieved w.h.p.?

Theorem.

The sketching complexity of IND is $\Omega(n)$.

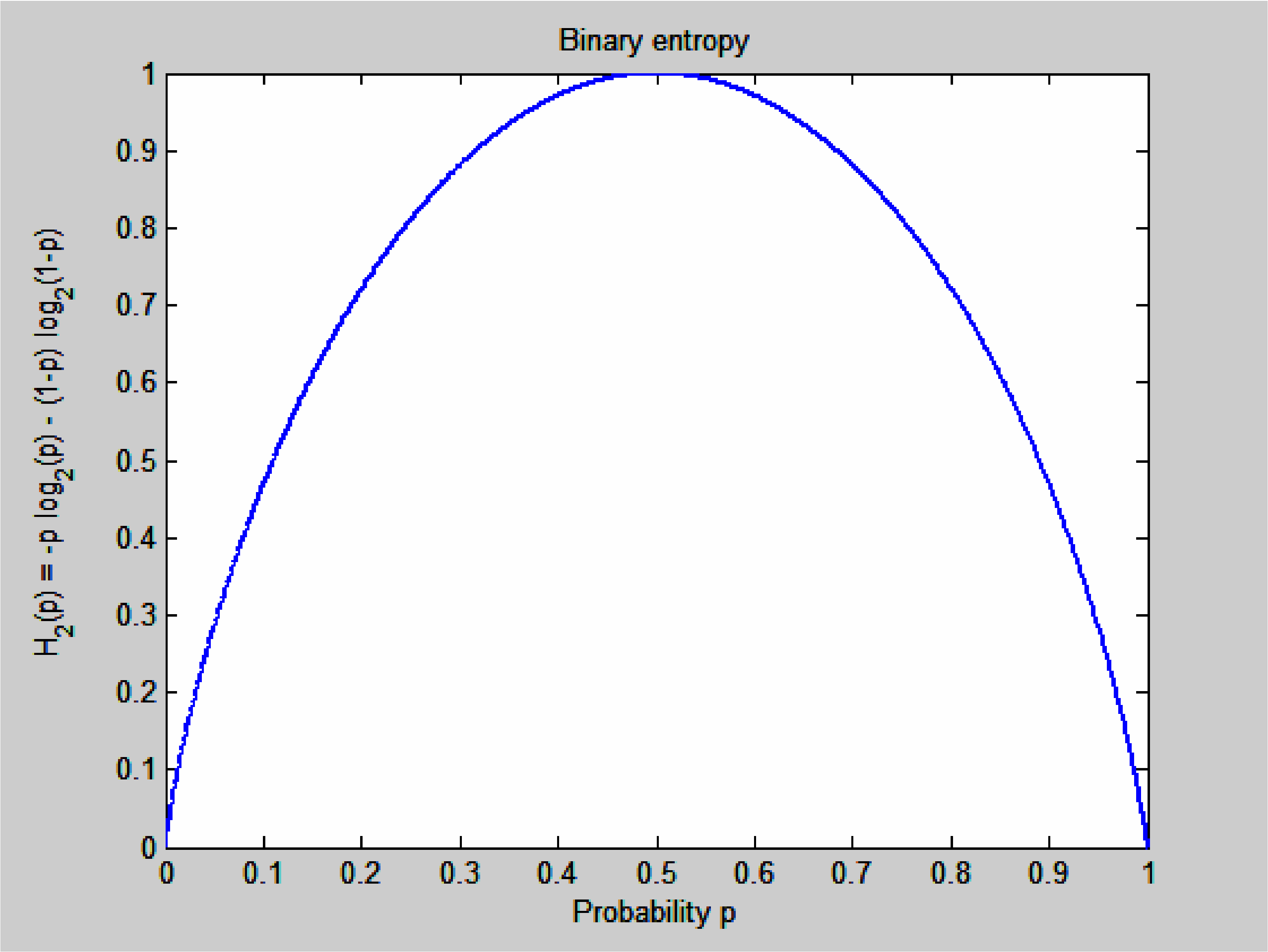
Information Theory Primer

Entropy of a random variable X

$$H(X) = -\sum_{x} \Pr[X = x] \log \Pr[X = x]$$

- amount of "uncertainty" in X (in bits)
- X is constant → H(X) = 0
- X is uniform → H(X) = log(|range(X)|)
  - largest value possible

Binary Entropy: $H_2(\cdot)$



Conditional Entropy

Conditional entropy of X given Y

$$H(X \mid Y) = H(X, Y) - H(Y)$$

- amount of uncertainty left in X after knowing Y
- $H(X \mid X) = 0$
- If X,Y are independent, $H(X \mid Y) = H(X)$

Mutual Information

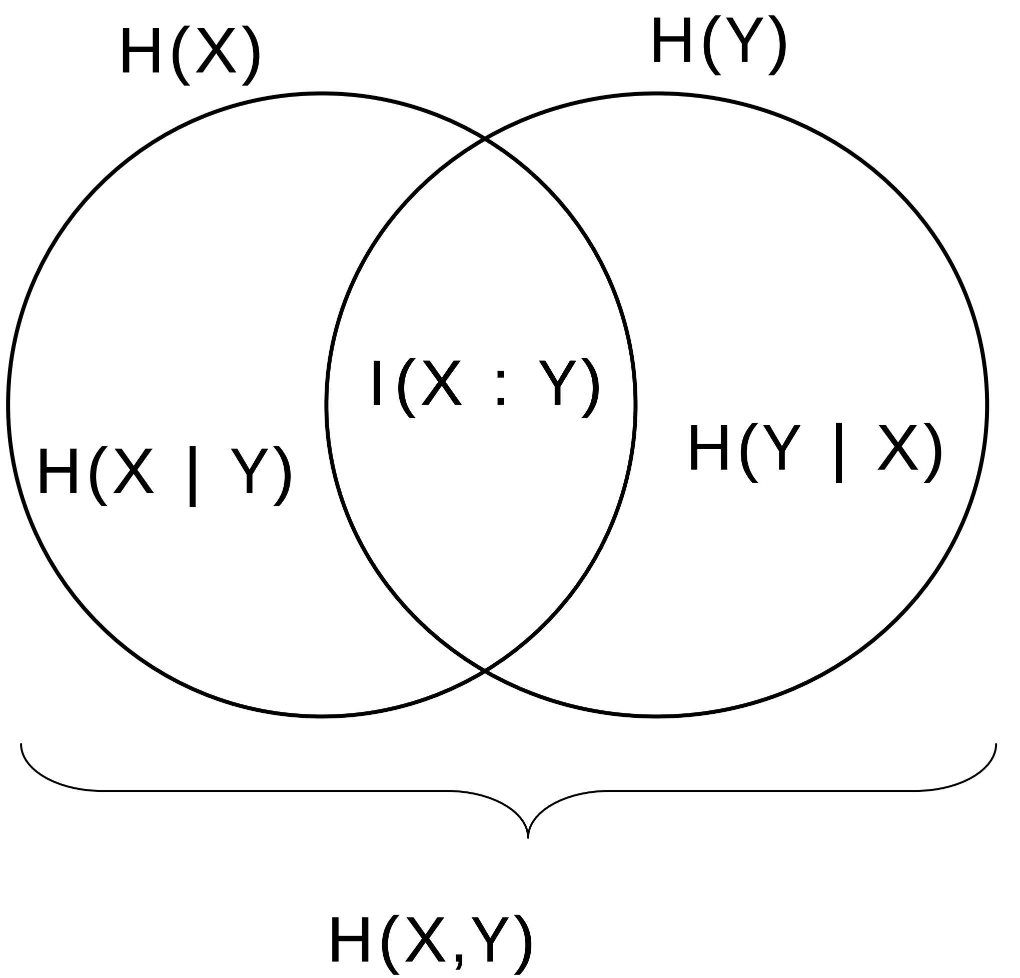
Mutual information between X and Y:

$$\begin{aligned} I(X:Y) &= H(X) - H(X \mid Y) \\ &= H(Y) - H(Y \mid X) \end{aligned}$$

Conditional mutual information:

$$I(X:Y \mid Z) = H(X \mid Z) - H(X \mid Y, Z)$$

Basic Relationships



Sub-additivity

Entropy is sub-additive

$H(X,Y) \leq H(X) + H(Y)$.

- Equality iff X, Y independent
- ➔ $H(X \mid Y) \leq H(X)$
- ➔ $H(X \mid Y,Z) \leq H(X \mid Z)$

Fano's Inequality

- X: a binary random variable
- Y: a predictor of X
  - g(Y) is a "guess" of X, for some function g
- E: indicator r.v. for error event "$g(Y) \neq X$"

Then, $H(X \mid Y) \leq H(E)$

- If $\Pr[E] \leq \delta \leq ½$, then $H(E) \leq H_2(\delta)$

Indexing

- Input: a binary string x of length n
- Output: a sketch of x so that any bit of x can be retrieved w.h.p.

Theorem.

The sketching complexity of indexing is $\Omega(n)$.

Proof

- Let $s(x,R)$ be the sketch of x
  - R is a public coin
- Let X be uniformly chosen in $\{0,1\}^n$
- Let $S = s(X,R)$
- We will show that $H(S)$ is large
  → sketch size must be large

Proof (cont.)

$$H(S)$$

$$\geq H(S \mid R)$$

$$\geq H(S \mid R) - H(S \mid X,R)$$

$$= I(X : S \mid R)$$

$$= \underbrace{H(X \mid R)} - \underbrace{H(X \mid S,R)}$$

$$H(X \mid R) = H(X) = n$$

Proof (cont.)

$H(X \mid S,R)$

$= H(X_1,X_2,\ldots,X_n \mid S,R)$

$\leq \sum_i H(X_i \mid S,R)$

[by sub-additivity]

$\leq n \cdot H_2(\delta)$

[by Fano's inequality]

Concluding,

$H(S) \geq n - n \cdot H_2(\delta) \geq n \cdot (1- H_2(\delta))$

Summary

- Information-theoretic arguments provide a general LB template
- Can be used to prove lower bounds for other functions, e.g., set disjointness
- In some cases, refined tools are needed to understand the structure
  - E.g., Statistics, Fourier analysis

- Open problem: prove good lower bounds on the sketching of edit distance