Information Theory and Applications, Feb. 2014

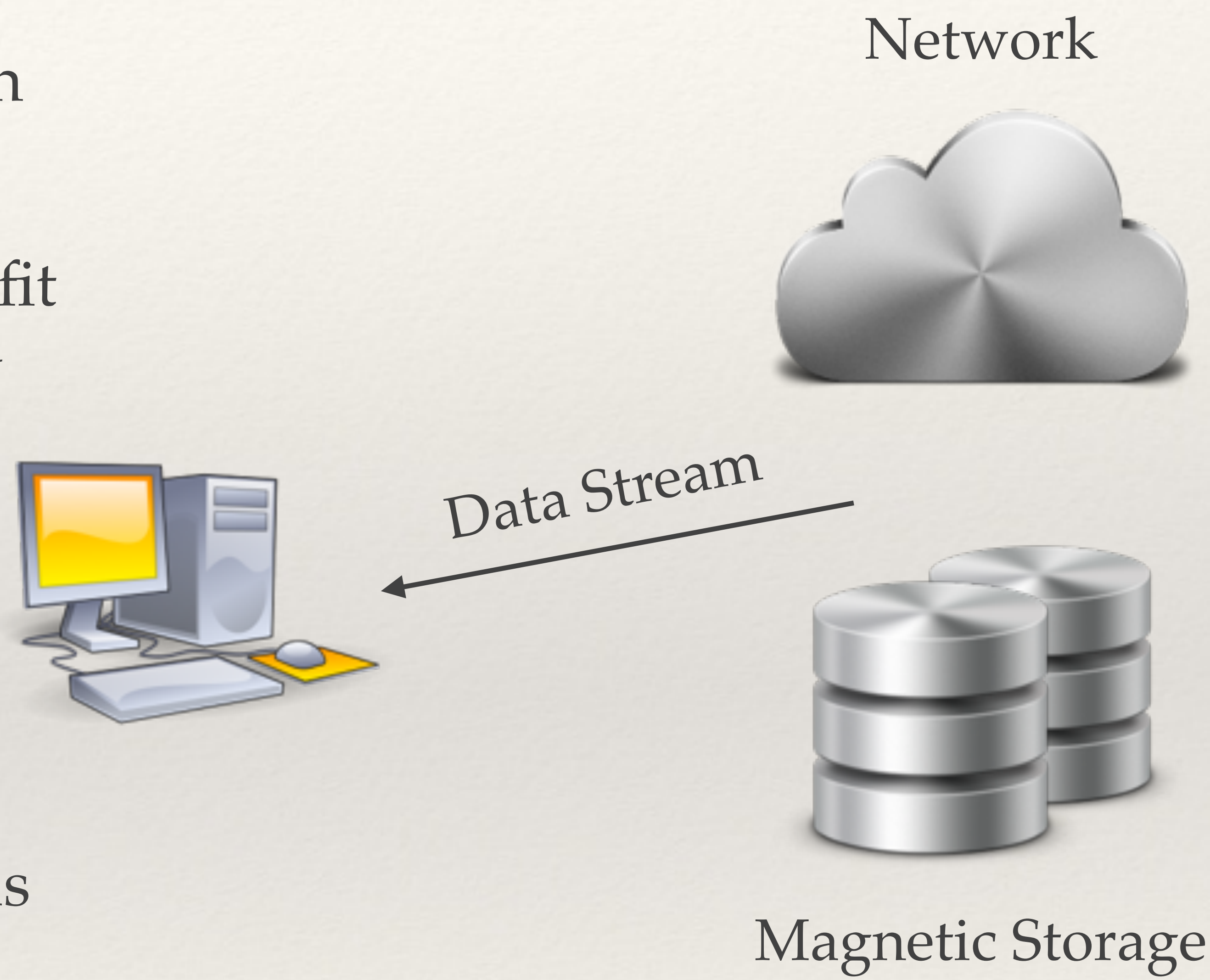
Sorting Big Data with Small Memory

Farzad Farnoud
Eitan Yakoobi
Jehoshua Bruck

Caltech

Sorting with Limited Memory

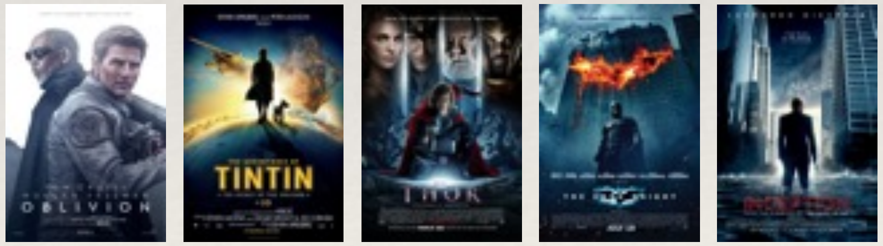
- Sorting is a fundamental operation in data processing
- Data maybe so large that it does not fit in memory and must be sequentially accessed:
  - Streamed data from network
  - Data stored on magnetic storage
- Not to rearrange data but to approximate its ordering as closely as possible
- Study of relationship between quality of sorting and available memory



Learning Preference Rankings

- Same model for obtaining a user's ranking of objects presented one by one
- User's ranking is useful for recommendation and collaborative filtering
- User can remember only a small number of movies she watched

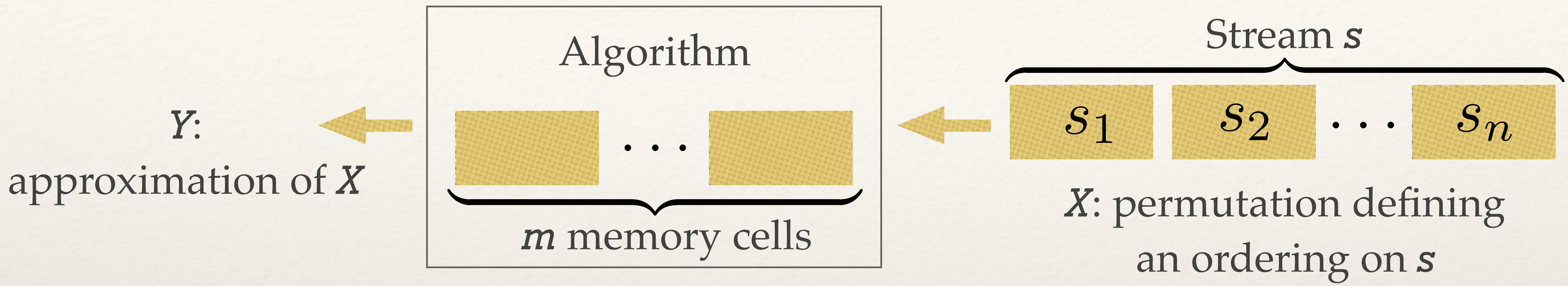
Ranking of movies ⟵ 

⟵ 

Learning Preference Rankings

- Same model for obtaining a user's ranking of objects presented one by one
- User's ranking is useful for recommendation and collaborative filtering
- User can remember only a small number of movies she watched

Problem Statement



- If i appears before j in X, then $s_i < s_j$
- To store stream elements, m cells are available; no limitation on other types of memory
- Algorithm can compare any two elements residing in memory
- Deterministic algorithms, X is a random permutation
- Performance measure: *Mutual information* and *distortion* between X and Y

Related Work

- J. Munro and M. Paterson. Selection and sorting with limited storage. Theoretical Computer Science, 12(3):315–323, 1980.
- G. S. Manku, S. Rajagopalan, and B. G. Lindsay. Approximate medians and other quantiles in one pass and with limited memory. ACM SIGMOD 1998
- Sudipto Guha and Andrew McGregor. Approximate quantiles and the order of the stream. In Proc. 25th ACM Symposium on Principles of Database Systems, pp. 273– 279, 2006.
- A. Chakrabarti, T. S. Jayram, and M. Patrascu. Tight lower bounds for selection in randomly ordered streams. SODA 2008

Universal Bounds: Mutual Information

Theorem: For any algorithm, if $m=n^b$, we have

$$I(X;Y)/H(X) \leq b(1+o(1)),$$

where I is mutual information and H is entropy.

Proof outline:

- Algorithms may only compare elements in memory
- Mutual information between X and Y cannot be larger than entropy of solutions of comparisons

Kendall Distortion

- To measure agreement between input and output we use Kendall tau and weighted Kendall distances
- *Kendall tau* distance:
  - Counts the number of *pairwise mistakes*
  - *# transpositions of adjacent elements* needed to take one permutation to another
  - Example: $d_\tau(312,123)=2$ since 312→132→123
- *Weighted Kendall* distance:
  - Weight w_i for transposing ith and $(i+1)$st elements
  - Can be used to penalize mistakes in higher positions more
  - Example: Let $w_1=2$ and $w_2=1$. $d_w(312,123)=3$ since 312→132→123

Universal Bounds: Kendall Distortion

Theorem: For any algorithm with memory μn and average Kendall distortion δn,

$$\mu \geq 1/(e^2\delta)\ (1+O(\mathrm{Log}\ n\ /n)+O(1/\delta)).$$

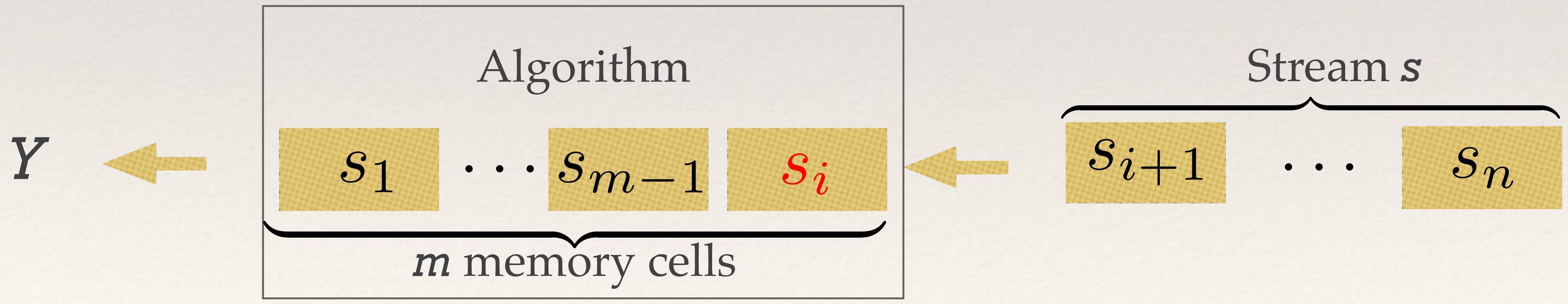
Proof outline:

- Bound number of outputs of alg. by counting solutions to comparisons
- Set of outputs can be viewed as a covering code
- Use rate-distortion on permutations [Wang et al. 2013, Farnoud et al. 2014]

See paper for non-asymptotic result in δ.

Algorithm

- A simple algorithm:
  - Store the first m-1 elements of the stream as *pivots*
  - Sort the set {1,2,...,m-1} based on the ordering $s_1, s_2, \ldots, s_{m-1}$
  - Compare each new element with pivots
  - Put the index of new element in its proper position in Y



Algorithm: Performance

Theorem: In terms of mutual information, the algorithm is asymptotically optimal.

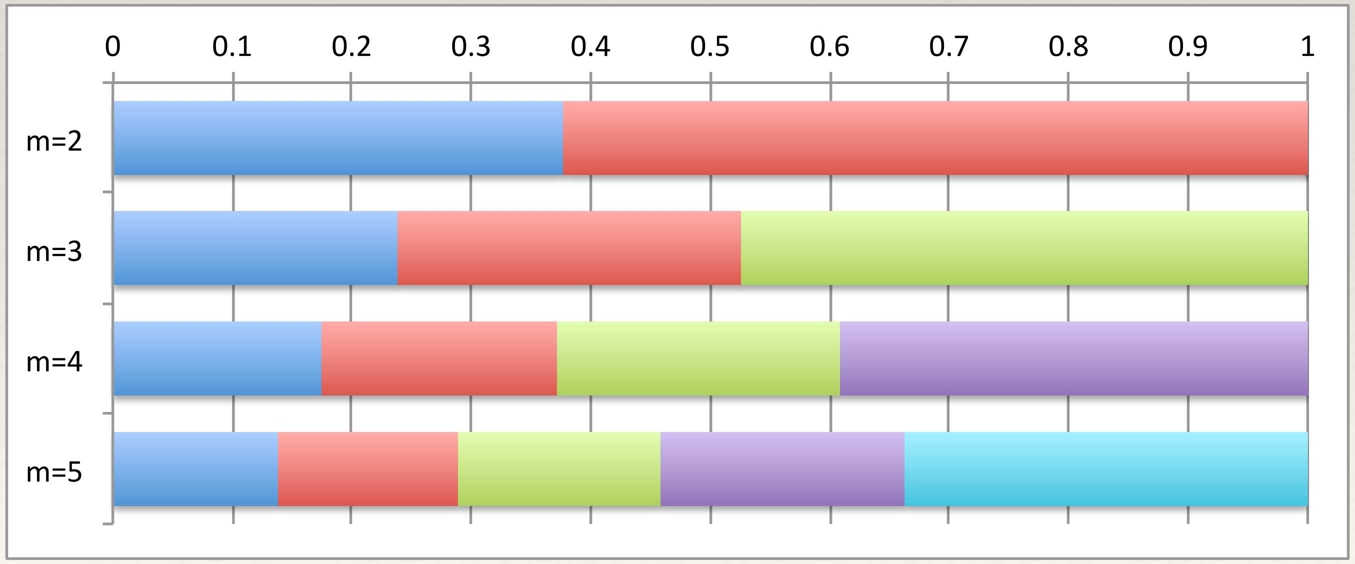
Theorem: Suppose the algorithm has memory μn and average Kendall distortion δn. We have

$$\mu \leq 1/(2\delta)\ (1+O(1/n)+O(1/\delta)).$$

To provide the same distortion as an optimal algorithm, we need $e^2/2 \cong 3.7$ times as much memory.

Distortion with Weighted Kendall

- What should be the ranks of pivots if errors in higher positions are to be penalized more?
- Use weighted Kendall to model non-uniform importance
- Linearly decreasing weight function: $w_i = 1 + c\ (n\text{-}i\text{-}1)$:



Remembering last *m* elements

- Finding the best ranking is closely related to the #P-complete problem of *counting the number of linear extensions of a poset*
- Simple algorithm: rank each group of *m* elements and interleave

Theorem: In terms of mutual information, the algorithm is asymptotically optimal. That is, with $m=an^b$, a fraction b of information in X is recovered.

- Better algorithm needed for distortion