Sorting with Limited Storage

Source

Data Stream

Sensor data: Tbs/day

Sensor

User with limited storage

Application

Rank Correlation Testing (Spearman’s, Kendall’s)

Preference Learning

Netflix, Amazon servers

Movies: 1/week
Problem Statement

Stream $s$

$S_n \ldots S_2 S_1$

$X$: permutation defined by the ordering of $s$

Algorithm

$\{z\}$

$m$ storage cells

$Y$: approximation of $X$

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

- $s_2 < s_5 < s_3 < s_4 < s_6 < s_1$, $X = 253461$, and $m = 3$
Example

- $s_2 < s_5 < s_3 < s_4 < s_6 < s_1$, $X = 253461$, and $m = 3$
Example

- $s_2 < s_5 < s_3 < s_4 < s_6 < s_1$, $X = 253461$, and $m = 3$
Example

- $s_2 < s_5 < s_3 < s_4 < s_6 < s_1$, $X = 253461$, and $m = 3$

$\begin{align*}
  s_2 & < s_1 \\
  s_2 & < s_3 < s_1
\end{align*}$

(storage cells)
Example

- $s_2 < s_5 < s_3 < s_4 < s_6 < s_1$, $X = 253461$, and $m = 3$

$S_6$ $S_5$  

$S_2 < S_1$ $S_2 < S_3 < S_1$ $S_2 < S_3 < S_4$
Example

- $s_2 < s_5 < s_3 < s_4 < s_6 < s_1$, $X=253461$, and $m=3$
Example

- \( s_2 < s_5 < s_3 < s_4 < s_6 < s_1, \quad X = 253461 \), and \( m = 3 \)
Example

- $s_2 < s_5 < s_3 < s_4 < s_6 < s_1$, $X = 253461$, and $m = 3$

- $s_2 < s_1$, $s_2 < s_3 < s_1$, $s_2 < s_3 < s_4$, $s_2 < s_5 < s_4$, $s_2 < s_4 < s_6$

- Output, e.g. $Y = 235146$ ($s_2 < s_3 < s_5 < s_1 < s_4 < s_6$)
Related Work


❖ G. S. Manku, S. Rajagopalan, and B. G. Lindsay. Approximate medians and other quantiles in one pass and with limited storage. ACM SIGMOD 1998


Performance Measures

❖ Kendall tau distortion:
  * Counts # of pairwise disagreements ( = # of transpositions of adjacent elements taking $X$ to $Y$)
  * Example: $d_{\tau}(312, 123)=2$ since $312 \rightarrow 132 \rightarrow 123$

❖ Weighted Kendall distortion

❖ Chebyshev distortion:
  * Maximum error in the rank of any element
  * Example: $d_{c}(35124, 12345)=3$

$X$: permutation defined by the ordering of $s$

Stream $s$

$m$ storage cells

Algorithm

$Y$: approximation of $X$
Universal Bounds: Kendall Distortion

**Theorem:** For any algorithm with storage $m = \mu n$ and average Kendall distortion $D = \delta n$, if $\delta$ is bounded away from zero, then

$$\mu \geq -W_0 \left( \frac{-\delta^\delta}{e(1 + \delta)^{1+\delta}} \right) (1 + o(1))$$
Universal Bounds: Kendall Distortion

**Theorem:** For any algorithm with storage $m = \mu n$ and average Kendall distortion $D = \delta n$, if $\delta$ is bounded away from zero, then

$$\mu \geq -W_0 \left( \frac{-\delta^\delta}{e(1 + \delta)^{1+\delta}} \right) (1 + o(1))$$

- As $\delta$ increases, we asymptotically have $\mu \geq 1/(e^2 \delta)(1+o(1))$
Universal Bounds: Kendall Distortion
Universal Bounds: Kendall Distortion

Proof outline:

❖ Let $C = \{y_1, y_2, y_3, \ldots \}$ denote the set of possible $Y$'s for a given algorithm
Universal Bounds: Kendall Distortion

Proof outline:

❖ Let $C=\{y_1, y_2, y_3, \ldots \}$ denote the set of possible $Y$'s for a given algorithm.

❖ Since algorithm is deterministic, $|C| \leq m!m^{n-m}$. 
Universal Bounds: Kendall Distortion

Proof outline:

❖ Let $\mathcal{C} = \{y_1, y_2, y_3, \ldots\}$ denote the set of possible $Y$’s for a given algorithm

❖ Since algorithm is deterministic, $|\mathcal{C}| \leq m!m^{n-m}$

❖ $\mathcal{C}$ can be viewed as a rate-distortion code [Wang et al 13, Farnoud et al 14]
Universal Bounds: Kendall Distortion

Proof outline:

❖ Let $\mathcal{C}=\{y_1, y_2, y_3, \ldots\}$ denote the set of possible $Y$'s for a given algorithm
❖ Since algorithm is deterministic, $|\mathcal{C}| \leq m! m^{n-m}$
❖ $\mathcal{C}$ can be viewed as a rate-distortion code [Wang et al 13, Farnoud et al 14]
❖ For distortion $D$, $|\mathcal{C}| > \frac{n!}{B(D)(D+1)}$

$B(D)$: size of ball of radius $D$
Universal Bounds: Kendall Distortion

Proof outline:
❖ Let $C=\{y_1, y_2, y_3, \ldots\}$ denote the set of possible $Y$'s for a given algorithm
❖ Since algorithm is deterministic, $|C| \leq m!m^{n-m}$
❖ $C$ can be viewed as a rate-distortion code [Wang et al 13, Farnoud et al 14]
❖ For distortion $D$, $|C| > \frac{n!}{B(D)(D + 1)}$
   - $B(D)$: size of ball of radius $D$
❖ If we have $B(D)$, eliminating $|C|$ gives the result
   - $B(D) \leq \binom{n + D - 1}{D}$
Universal Bounds: Chebyshev Distortion

**Theorem**: For any algorithm with storage $m = \mu n$ and average Chebyshev distortion $D = \delta n$, with $2/n \leq \delta \leq 1/2$, 

$$\mu \geq -\mathcal{W}_0 \left( \frac{-(e/2)^{2\delta}}{2\delta n} \right) (1 + o(1))$$

- For any fixed $\delta$ as $n$ increases, storage requirement becomes a vanishing fraction of $n$
- Constant distortion needs at least constant $\mu$
A simple algorithm:

- Store the first $m-1$ elements of the stream, $s_1, \ldots, s_{m-1}$, as pivots.
- Compare each new element with the pivots.

Example: Suppose $X = 263415$ ($s_2 < s_6 < s_3 < s_4 < s_1 < s_5$) and $m = 3$.
A simple algorithm:

- Store the first $m-1$ elements of the stream, $s_1, \ldots, s_{m-1}$, as pivots.
- Compare each new element with the pivots.

Example: Suppose $X=263415$ ($s_2<s_6<s_3<s_4<s_1<s_5$) and $m=3$: 

\[ \begin{array}{cccc}
  s_6 & s_5 & s_4 & s_3 & s_2 \\
\end{array} \]

\[ \begin{array}{cccc}
  s_1 & s_5 & s_3 & s_2 \\
\end{array} \]

storage cells
A simple algorithm:

- Store the first $m-1$ elements of the stream, $s_1, \ldots, s_{m-1}$, as *pivots*
- Compare each new element with the pivots

Example: Suppose $X=263415$ ($s_2<s_6<s_3<s_4<s_1<s_5$) and $m=3$:

- $s_6 \succ s_5 \succ s_4 \succ s_3$
Algorithm

- A simple algorithm:
  - Store the first $m-1$ elements of the stream, $s_1, \ldots, s_{m-1}$, as pivots
  - Compare each new element with the pivots
- Example: Suppose $X=263415$ ($s_2<s_6<s_3<s_4<s_1<s_5$) and $m=3$:
  
  $s_6 \quad s_5 \quad s_4 \quad \Rightarrow \quad s_1 \quad s_2 \quad s_3$

  $s_2 < s_1 \quad s_2 < s_3 < s_1$
Algorithm

❖ A simple algorithm:
❖ Store the first $m-1$ elements of the stream, $s_1, ..., s_{m-1}$, as pivots
❖ Compare each new element with the pivots
❖ Example: Suppose $X=263415$ ($s_2 < s_6 < s_3 < s_4 < s_1 < s_5$) and $m=3$:

$S_6$ $S_5$  $S_1$ $S_2$ $S_4$

$S_2 < S_1$ $S_2 < S_3 < S_1$ $S_2 < S_4 < S_1$
Algorithm

- A simple algorithm:
  - Store the first \( m-1 \) elements of the stream, \( s_1, \ldots, s_{m-1} \), as pivots
  - Compare each new element with the pivots
- Example: Suppose \( X=263415 \) (\( s_2<s_6<s_3<s_4<s_1<s_5 \)) and \( m=3 \):

\[
\begin{align*}
S_6 & \quad S_1 & S_2 & S_5 \\
S_2 & < S_1 & S_2 & < S_3 < S_1 & S_2 & < S_4 < S_1 & S_2 & < S_1 < S_5
\end{align*}
\]
A simple algorithm:

- Store the first $m-1$ elements of the stream, $s_1, \ldots, s_{m-1}$, as pivots
- Compare each new element with the pivots

Example: Suppose $X=263415$ ($s_2<s_6<s_3<s_4<s_1<s_5$) and $m=3$:

$s_2<s_1 \quad s_2<s_3<s_1 \quad s_2<s_4<s_1 \quad s_2<s_1<s_5 \quad s_2<s_6<s_1$
Algorithm

- A simple algorithm:
  - Store the first $m-1$ elements of the stream, $s_1,...,s_{m-1}$, as *pivots*
  - Compare each new element with the pivots
- Example: Suppose $X=263415$ ($s_2<s_6<s_3<s_4<s_1<s_5$) and $m=3$:

  $s_2<s_1$  $s_2<s_3<s_1$  $s_2<s_4<s_1$  $s_2<s_1<s_5$  $s_2<s_6<s_1$

- Output $Y=234615$, $d_r(263415,234615)=2$, $d_c(263415,234615)=2$
Algorithm: Kendall Distortion

**Theorem:** The algorithm asymptotically requires at most a constant factor as much storage as an optimal algorithm for the same Kendall distortion.

- For small $\delta$,
  - Proposed alg: $\mu \leq 1$
  - Opt. alg: $\mu$ bounded away from 0
- For large $\delta$,
  - Proposed alg: $\mu \leq 1/(2\delta) \ (1 + o(1))$
  - Opt. alg: $\mu \geq 1/(e^2 \delta) \ (1 + o(1))$
Algorithm: Chebyshev Distortion

**Theorem:** If the proposed algorithm has storage \( m = \mu n \) and average Chebyshev distortion \( D = \delta n \), with \( \delta \leq 1/2 \) and \( \delta \) bounded away from 0, then \( \mu \leq W_{-1}(-\delta/e)/\delta n \).

- If \( \delta \) is bounded away from 0, we need at most a constant times as much storage.
- For vanishing distortion, better algorithm and/or bounds are needed.
Algorithm: Chebyshev Distortion

Proof outline:

❖ Given $Y, X$ is unknown only in segments bounded by pivots: If $Y=156, then \ X \in \{156, 2437156, 2374165, 2437165, \ldots\}$

❖ Chebyshev distortion is bounded by the length of the longest segment

❖ Coupling with a randomly broken stick of length $n$ into $m$ parts

❖ Statistics of the length of the longest piece are well known [Holst’80]

❖ $\delta n = E[d_c(X,Y)] \leq E[\text{length of longest piece of stick}] \leq n \ln(me) / m$

❖ $(-m\delta) e^{-m\delta} \leq -\delta/e$
### Weighted Kendall: Why?

**Ranking of Wikipedia pages:**

- Rankings with different methods for important items are all very similar
- This is not reflected by the Kendall tau correlation
- The correlation coefficient is affected by items with low ranks

#### Table 1: Kendall's correlation index

<table>
<thead>
<tr>
<th></th>
<th>Indegree</th>
<th>PageRank</th>
<th>Katz</th>
</tr>
</thead>
<tbody>
<tr>
<td>United States</td>
<td>1</td>
<td>0.75</td>
<td>0.90</td>
</tr>
<tr>
<td>List of sovereign states</td>
<td>0.75</td>
<td>1</td>
<td>0.75</td>
</tr>
<tr>
<td>Animal</td>
<td>0.90</td>
<td>0.75</td>
<td>1</td>
</tr>
</tbody>
</table>

#### Table 2: Top 20 items

<table>
<thead>
<tr>
<th></th>
<th>United States</th>
<th>Animal</th>
<th>List of sovereign states</th>
<th>England</th>
<th>France</th>
<th>Germany</th>
<th>Canada</th>
<th>World War II</th>
<th>India</th>
<th>Australia</th>
<th>London</th>
<th>Japan</th>
<th>Italy</th>
<th>Arthropod</th>
<th>Insect</th>
<th>New York City</th>
<th>English language</th>
<th>Village</th>
<th>Nationa Reg. of Hist. Places</th>
</tr>
</thead>
<tbody>
<tr>
<td>United States</td>
<td>List of sovereign states</td>
<td>Animal</td>
<td>England</td>
<td>France</td>
<td>Germany</td>
<td>Association football</td>
<td>United Kingdom</td>
<td>India</td>
<td>United Kingdom</td>
<td>Canada</td>
<td>World War II</td>
<td>Canada</td>
<td>Arthropod</td>
<td>Insect</td>
<td>World War II</td>
<td>Japan</td>
<td>Australia</td>
<td>London</td>
<td>Japan</td>
</tr>
</tbody>
</table>

---

Rankings of Wiki pages with different methods: correlations and top-20 pages [Vigna’14]
Distortion with Weighted Kendall

- **Weighted Kendall distortion**: [F, Milenkovic 13]
  - Weight $w_i$ for transposing $i$th and $(i+1)st$ elements
  - Can be used to penalize mistakes in higher positions more
  - Example: $w_1 = 2, w_2 = 1, d_w(312, 123) = 3$ since $312 \rightarrow 132 \rightarrow 123$

- Axiomatically derived based on Kemeny’s axioms
- No need for ground truth
- Fast computation: $O(n \log n)$ with small constant for monotonically decreasing weights
What should the ranks of pivots be if errors in higher positions are penalized more? 

Example: Linearly decreasing weight function: $w_i = 1 + c (n-i-1)$ with $c > 0$:

- The pivots are chosen more closely at the top to provide better accuracy for highly ranked items
- Optimum positions for pivots is asymptotically independent from $c$!
Conclusion

- Provided bounds on the performance of algorithms for sorting with limited storage
- Proposed an algorithm and showed that it is asymptotically optimal for
  - Kendall distortion (up to a constant factor), and
  - Chebyshev distortion (up to a constant factor and for $\delta$ bounded away from 0)
- Future work and open problems:
  - What is the best possible algorithm if only the last $m$ are remembered?
  - Tighter bounds for Kendall distortion
  - Better algorithm/bounds for Chebyshev distortion when $\delta$ is small
Thank You!