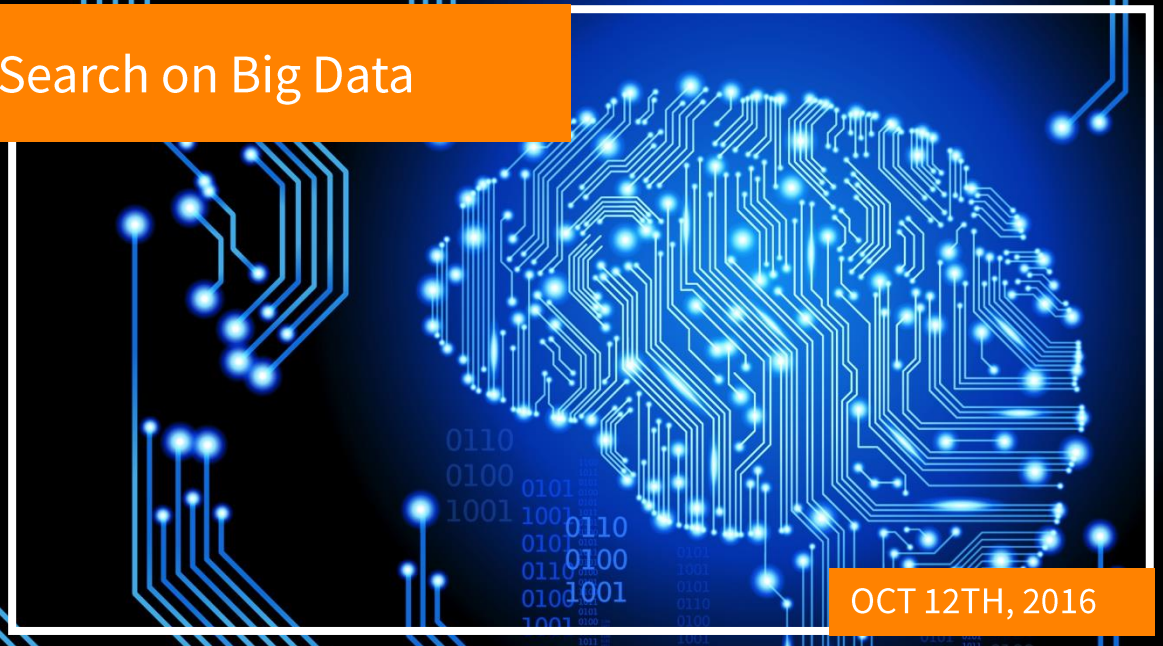
Textual Similarity Search on Big Data



OCT 12TH, 2016

Prof. Johann-Christoph Freytag, Ph.D.,
Fabian Fier

{headtcentre}
humboldt-elsevier
advanced data & text centre

HUMBOLDT-UNIVERSITÄT ZU BERLIN

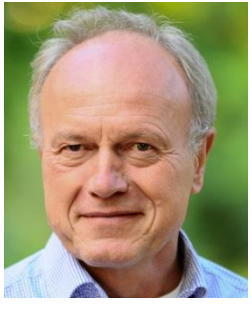
About Us

{headtcentre}

humboldt-elsevier
advanced data & text centre

Research Integrity

Similarity Search



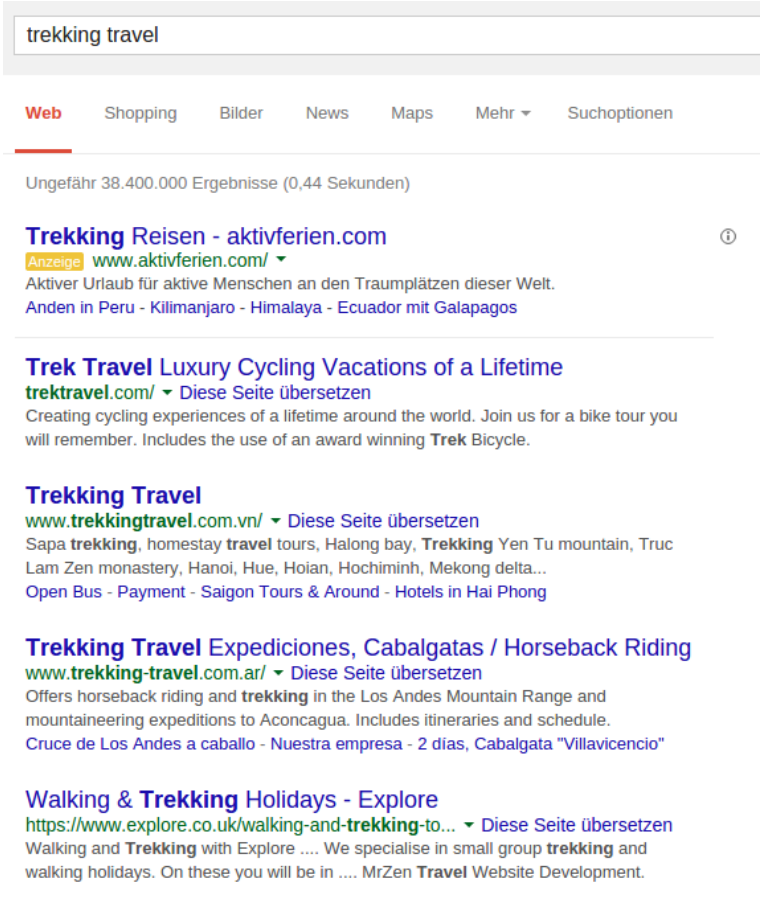
Christoph

Fabian

Motivation 1: Plagiarism Detection

Copying in suspect text (not reference text) with no reference	Grayscale color
Significant copying: multiple identical sentences in a paragraph	6
some copying: ≥5 contiguous words within a sentence of ≥ 5 words	5
near copying: multiple (≥ 3) exact phrases (≥ 3 words) overlap in contiguous sentences in a paragraph	4
similarity 1: several (<3) exact phrases (≥ 3 words) overlap in contiguous sentences in a paragraph or more than 9 words in sequence	3
similarity 2: many (≥5) of the exact same words (excluding function words) in contiguous sentences in the same paragraph	2
similarity 2: topic overlap with facts and standard phrases	1
Standard words or phrases	0
General knowledge / facts	0

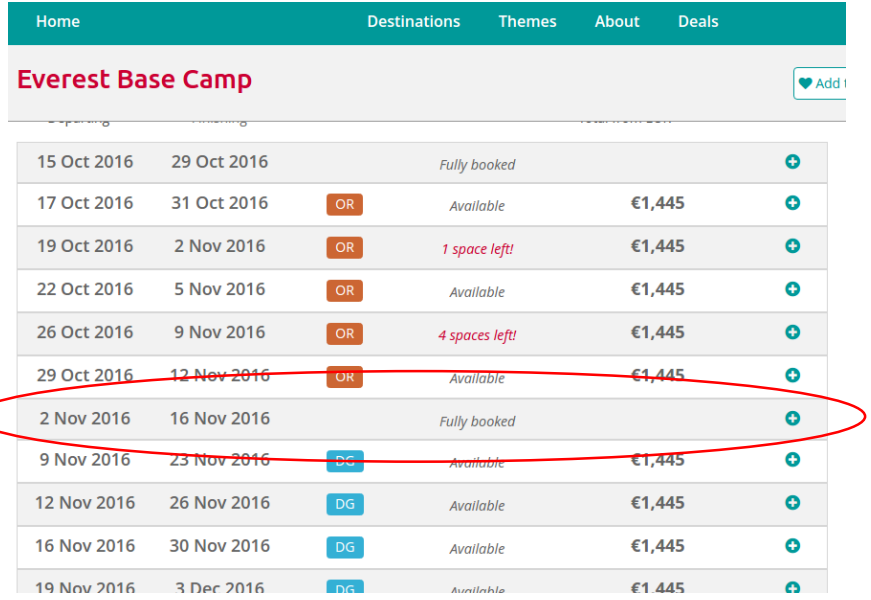
Motivation 2: Enhancing Web Search



Nepal!

{headtcentre}
humboldt-elsevier
advanced data & text centre

Motivation 2: Enhancing Web Search



Web search is **only textual**

no **similarity search**

SELECT * FROM web WHERE
text SIMILAR TO CURRENT website

Problem Statement

Input:

- Set R of documents
- Search document s
- Similarity measure $sim(r, s) \mapsto [0; 1]$
- Similarity threshold θ

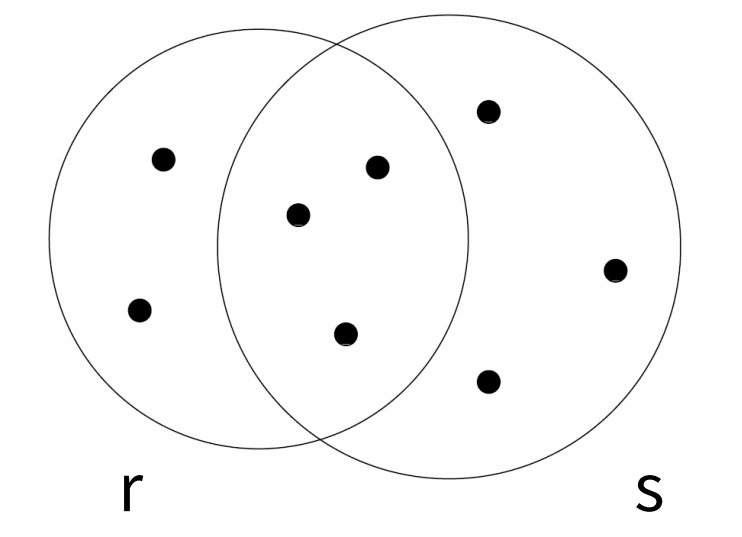
Output:

- $\{r \epsilon R | sim(r, s) \geq \theta\}$

Example: Jaccard Similarity

r = {lorem ipsum consetetur sadipscing elitr}
s = {magna aliquyam erat consetetur sadipscing elitr}

$$sim(r,s) = \frac{|r \cap s|}{|r \cup s|} = \frac{3}{8}$$



Naive Approach

Given: search document s, input dataset R, similarity threshold θ

For each $r \epsilon R$:

    If $\mathrm{sim}(r, s) \geq \theta$:

        Output r

Feasibility?

Our Approach

- Use distributed similarity **join** algorithms (MapReduce)
  - Similar problem: find **all pairs** of similar documents
    - $\{(r,s)\epsilon(R \times R)|sim(r,s) \geq \theta\}$
  - Use intermediate results for indexing

- Find the best join approach(es) and use them for search
  - Metric partitioning
  - Filter-and-verify

Filter-and-Verify Approach

- Generate a set of **candidate pairs**:
  - superset of result set
  - use of filters: orders of magnitude smaller than cross product (usually)
- For each candidate pair, verify if it meets the similarity threshold

Filter-and-Verify: Full-Filtering Approach

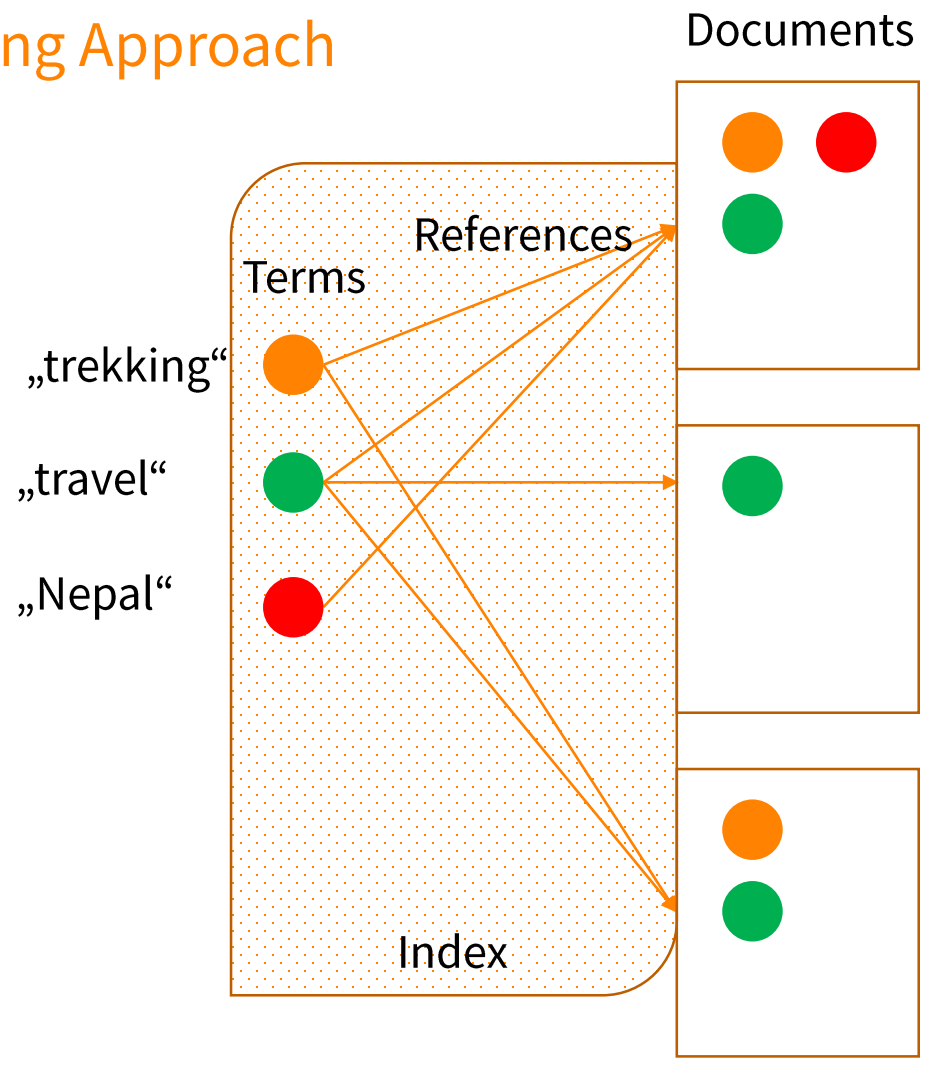
- Build inverted index $\{\langle term, \{document\}\rangle\}$
- For each element $\langle term, \{document\}\rangle$ of the inverted index:

  For each document pair (r, s) :

  If $\mathrm{sim}(r, s) \geq \theta$:

  Output <r,s>

Specific problems?



Filter-and-Verify: Prefix Filtering

Example:

r = {Lorem ipsum dolor sit consetetur sadipscing elitr}

s = {magna aliquyam erat consetetur sadipscing elitr}

$$\text{prefix}_{\text{Jaccard}}(r, 0.5) = 5$$
$$\text{prefix}_{\text{Jaccard}}(s, 0.5) = 4$$

Build inverted index only over the prefix: result stays complete.

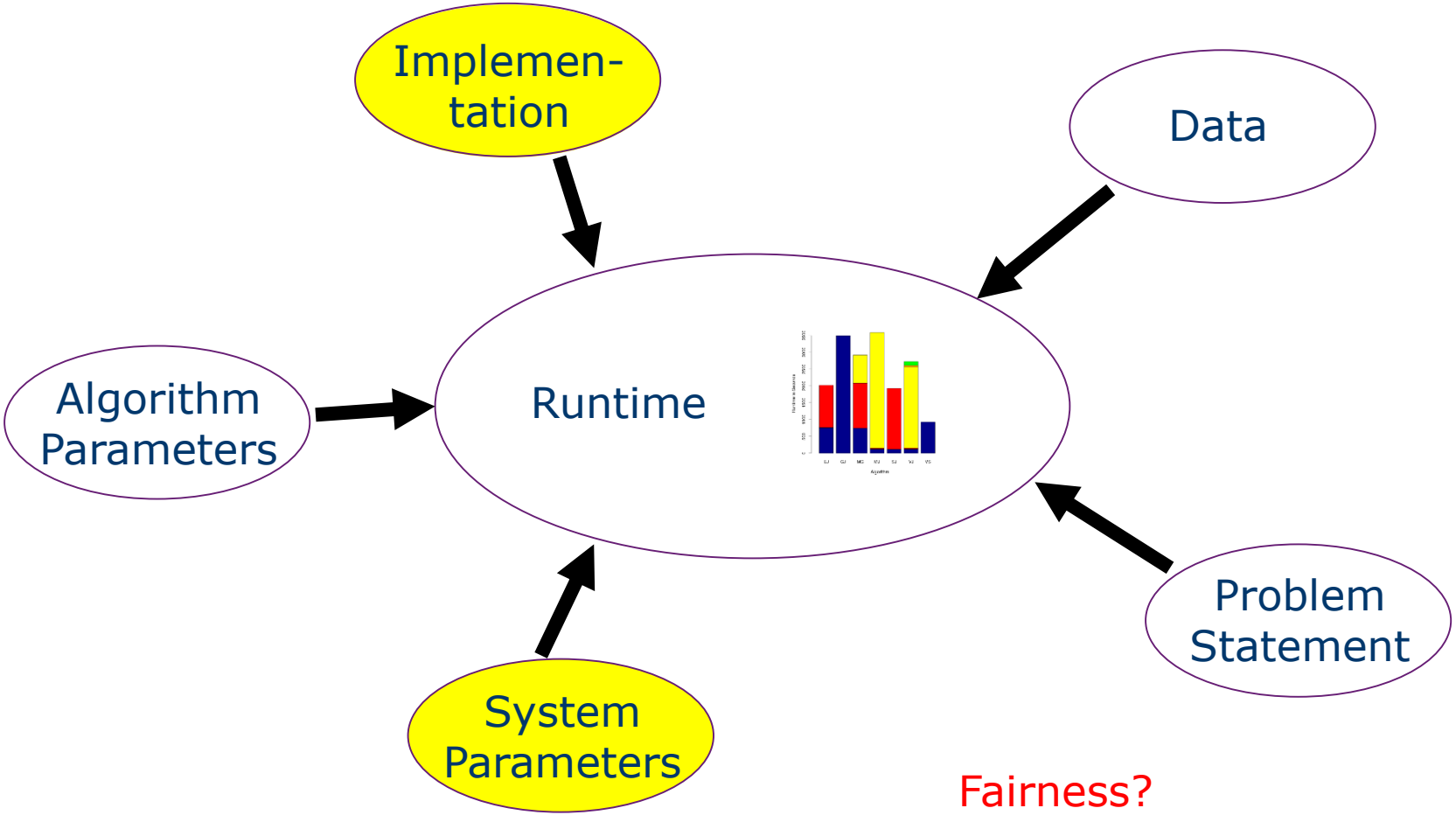
Optimization: use global token order. Sort each set/document in ascending order (already true for this example).

Evaluation

- We experimentally compare the runtime of MapReduce join algorithms

The *V-SMART-Join* algorithms are very efficient and scalable in the number of entities, as well as their cardinalities. They were up to 30 times faster than the state of the art algorithm, *VCL*, when compared on a real dataset of a small size.

Comparability Issue



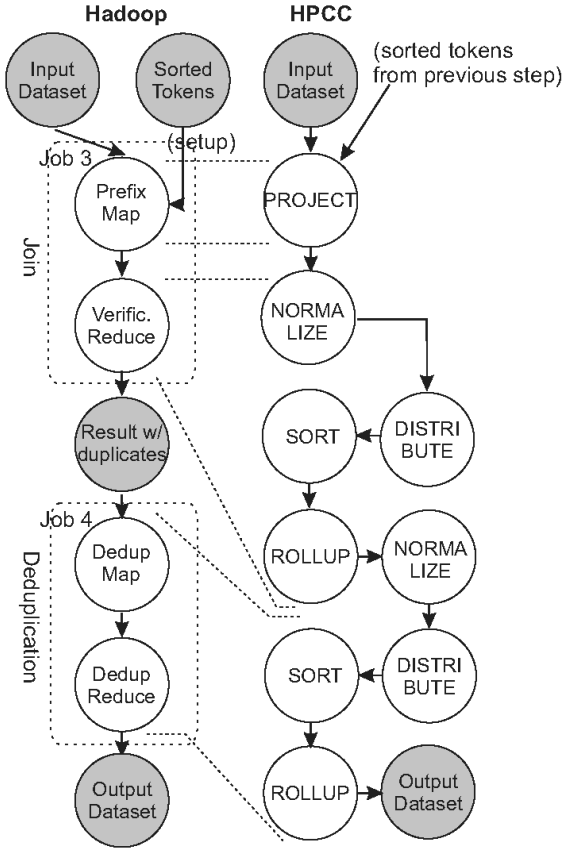
Comparability Issue

- Implementation
  - Combine
  - Prune locally
- System Parameters
  - Data Split
  - Parallelization
  - Memory

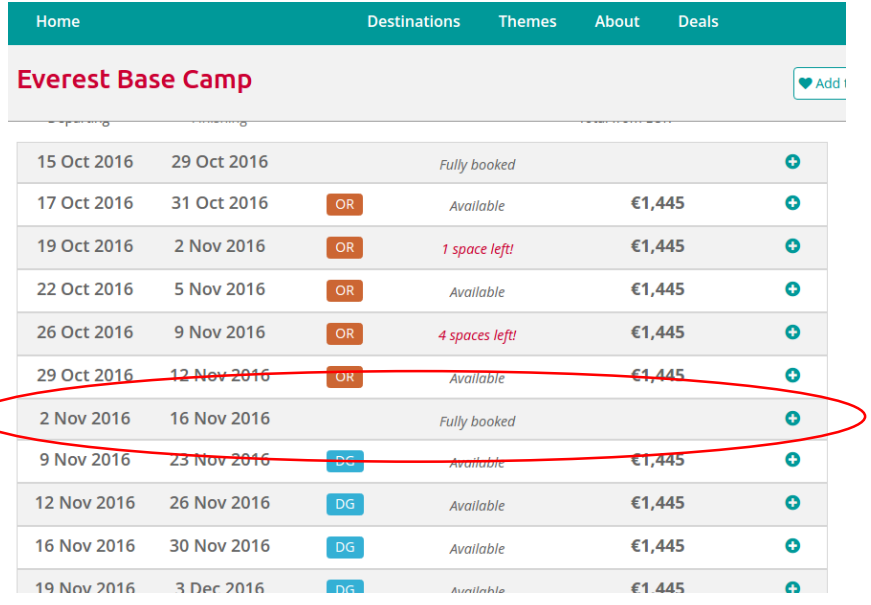
Idea: Use a MapReduce implementation with less tuning knobs than Hadoop: HPCC subsumes MapReduce

Comparability Issue

- Implementation
  - Combine - same
  - Prune locally - same
- System Parameters
  - Data Split – good default
  - Parallelization – good default
  - Memory - ?



Future: Enhancing Web Search



Web search is **only textual**

no **similarity search**

SELECT * FROM web WHERE
text SIMILAR TO CURRENT website
AND start_date > 11/01/2016
AND end_date < 11/30/2016
AND availability > 1
AND location CLOSE TO Nepal

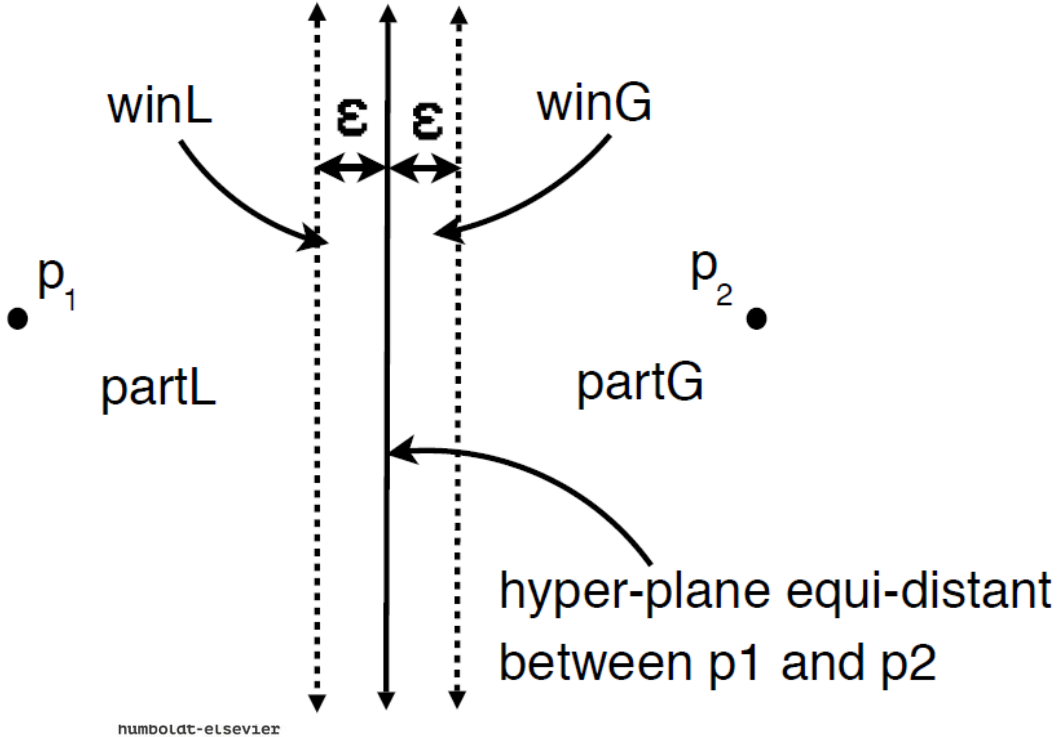
Thank you!

Questions?

Backup

Metric Partitioning Approach

- Idea: use similarity measure to iteratively partition the space until partitions fit into main memory



p_1, p_2: random pivot elements
$\varepsilon = 1 - \theta$

(From Jacox and Samet: Metric Space Similarity Joins)