QDB 2009

7th International Workshop on Quality in Databases at VLDB 2009
August 24th, 2009, Lyon, France

HPI Information Systems Group
Hasso Plattner Institut | Universität Potsdam

A Comparison and Generalization of Blocking and Windowing Algorithms for Duplicate Detection

Uwe Draisbach
uwe.draisbach@fernuni-hagen.de

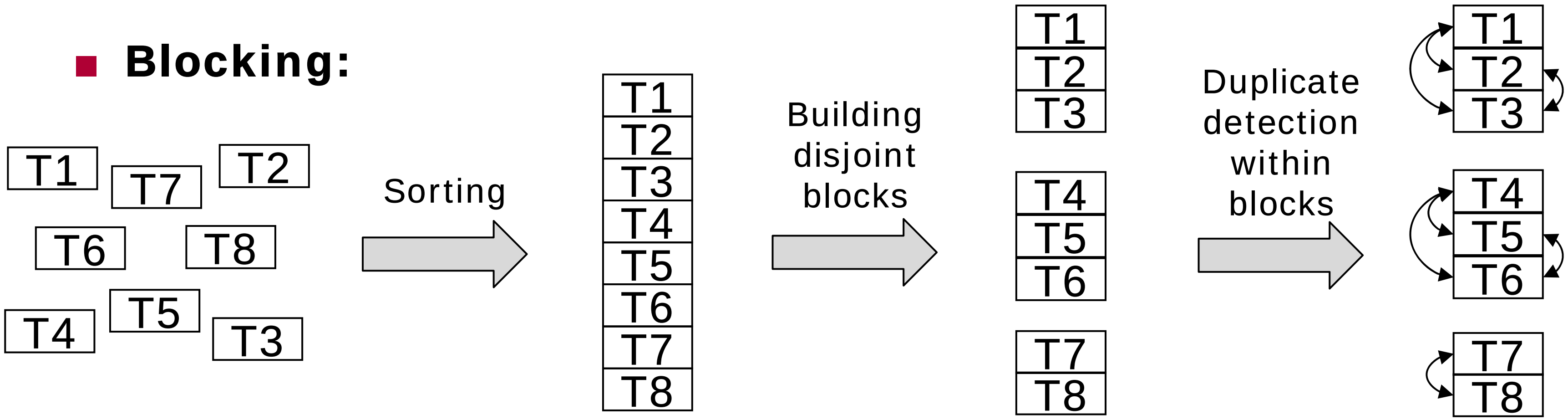
Felix Naumann
naumann@hpi.uni-potsdam.de

Introduction Duplicate Detection

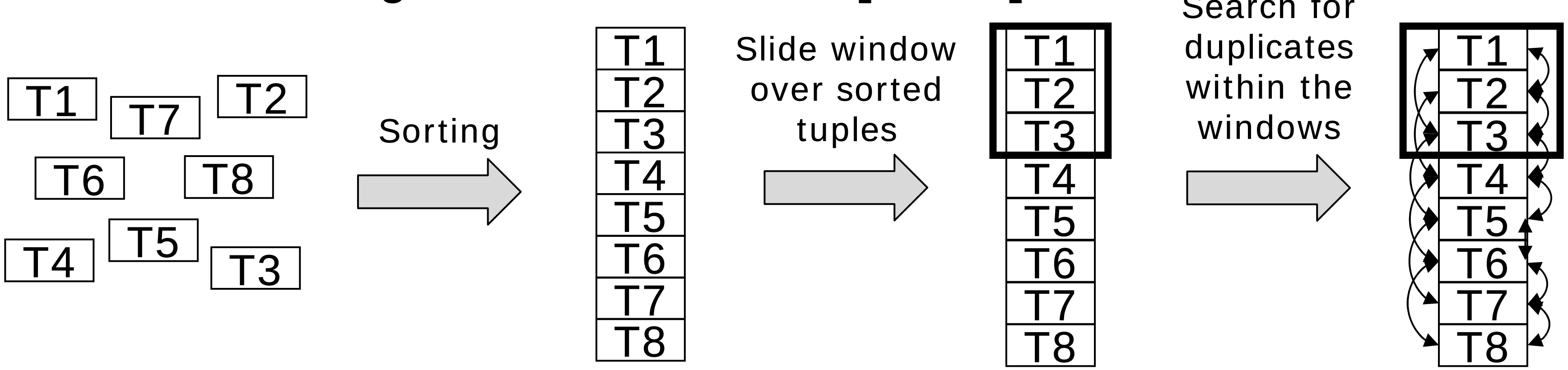
- **Duplicate Detection**
  - Find all pairs of tuples that represent the same real-world object
- Number of comparisons = $\frac{n^2 - n}{2}$
  - 10,000 tuples result in approx. 50 Mio. comparisons
- **Challenge**
  - *Efficiency:* reduce the number of comparisons
  - *Effectiveness*: find a similarity measure that classifies pairs of tuples correctly as duplicates or non-duplicates

Blocking and Windowing Algorithms

- **Blocking:**



- **Sorted Neighborhood Method [HS98]:**



Comparing Blocking and Windowing

SNM

Blocking

Window size: 3
Block size: 5

Sorted tuples

Tuples 1 & 5 are only compared using Blocking

Tuples 16 & 14 are only compared using SNM

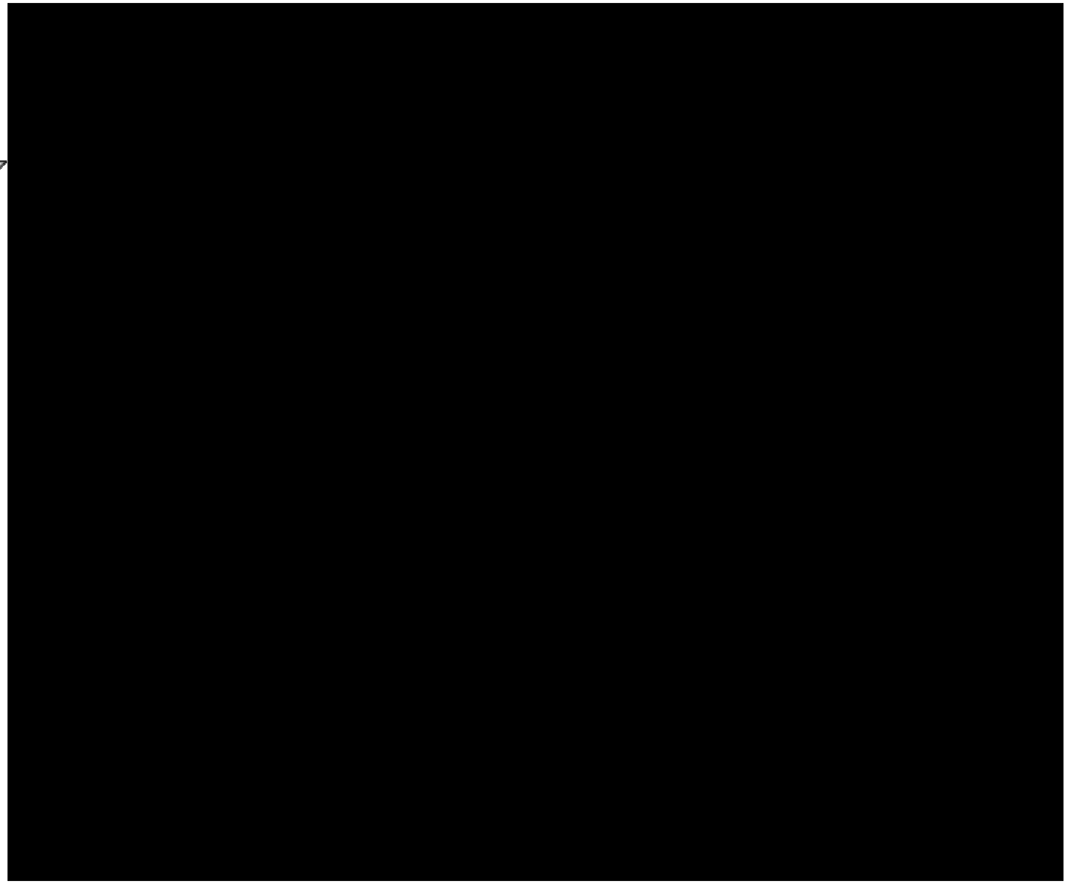
Comparing Blocking and Windowing

SNM

Blocking

Window size: 5
Block size: 5

Sorted tuples



Increasing window size to approximate Blocking

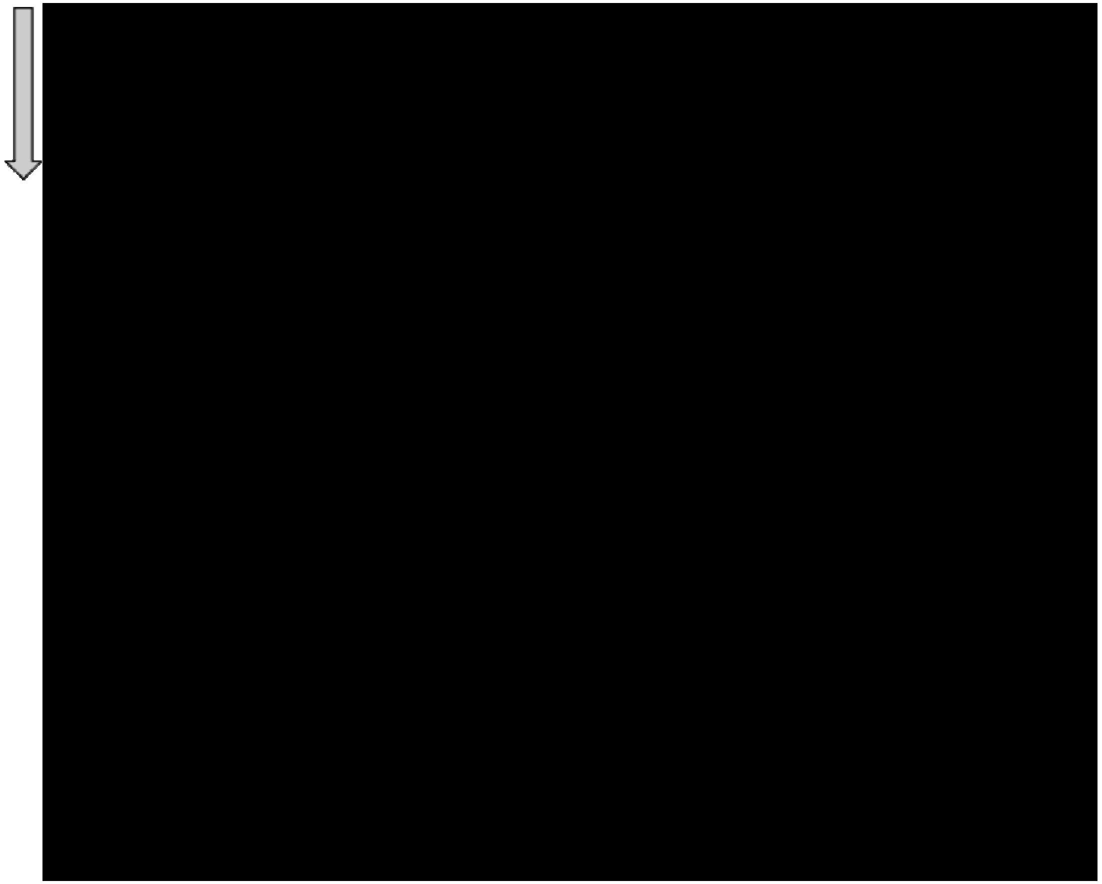
Comparing Blocking and Windowing

SNM

Blocking

Window size: 3
Block size: 5

Sorted tuples



Overlapping blocks to approximate Windowing

Sorted Blocks Method

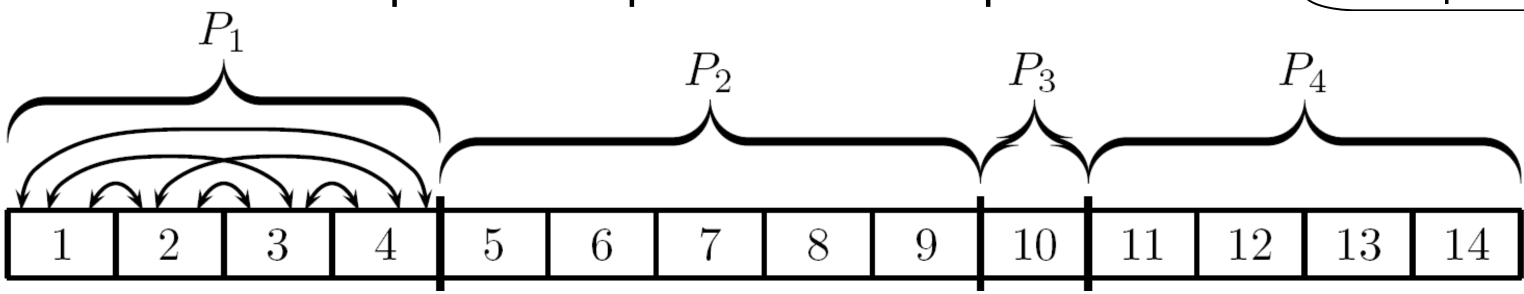
- Approach
  1. Sort tuples
  2. Build disjoint partitions
  3. Perform complete comparison within partitions
  4. Overlap partitions
  5. Slide fixed size window across sorted tuples within overlap

- Overlap
  - Parameter o = number of tuples from one partition that are part of the overlap
  - Overlap size = $2o$
  - Size of window = $o+1$

Sorted Blocks Method

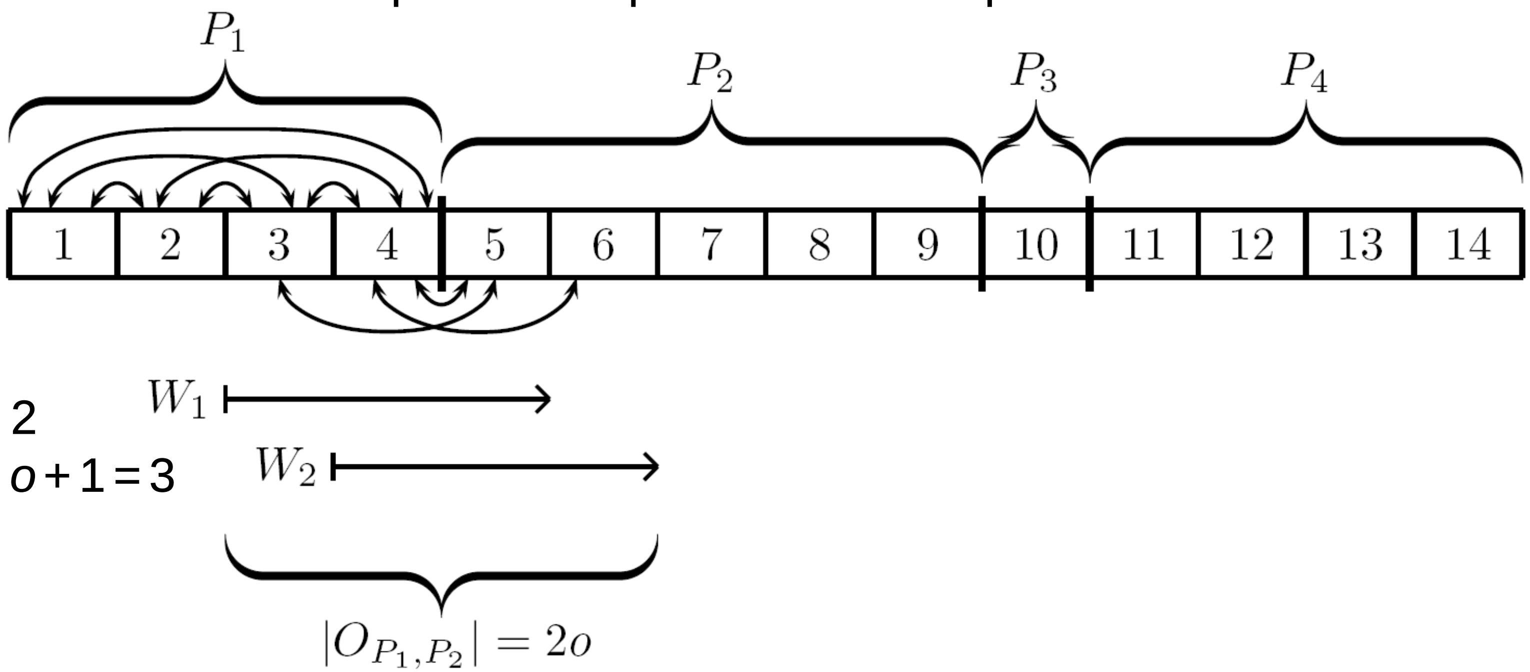
Complete comparison within partitions

Quadratic complexity



Sorted Blocks Method

Complete comparison within partitions



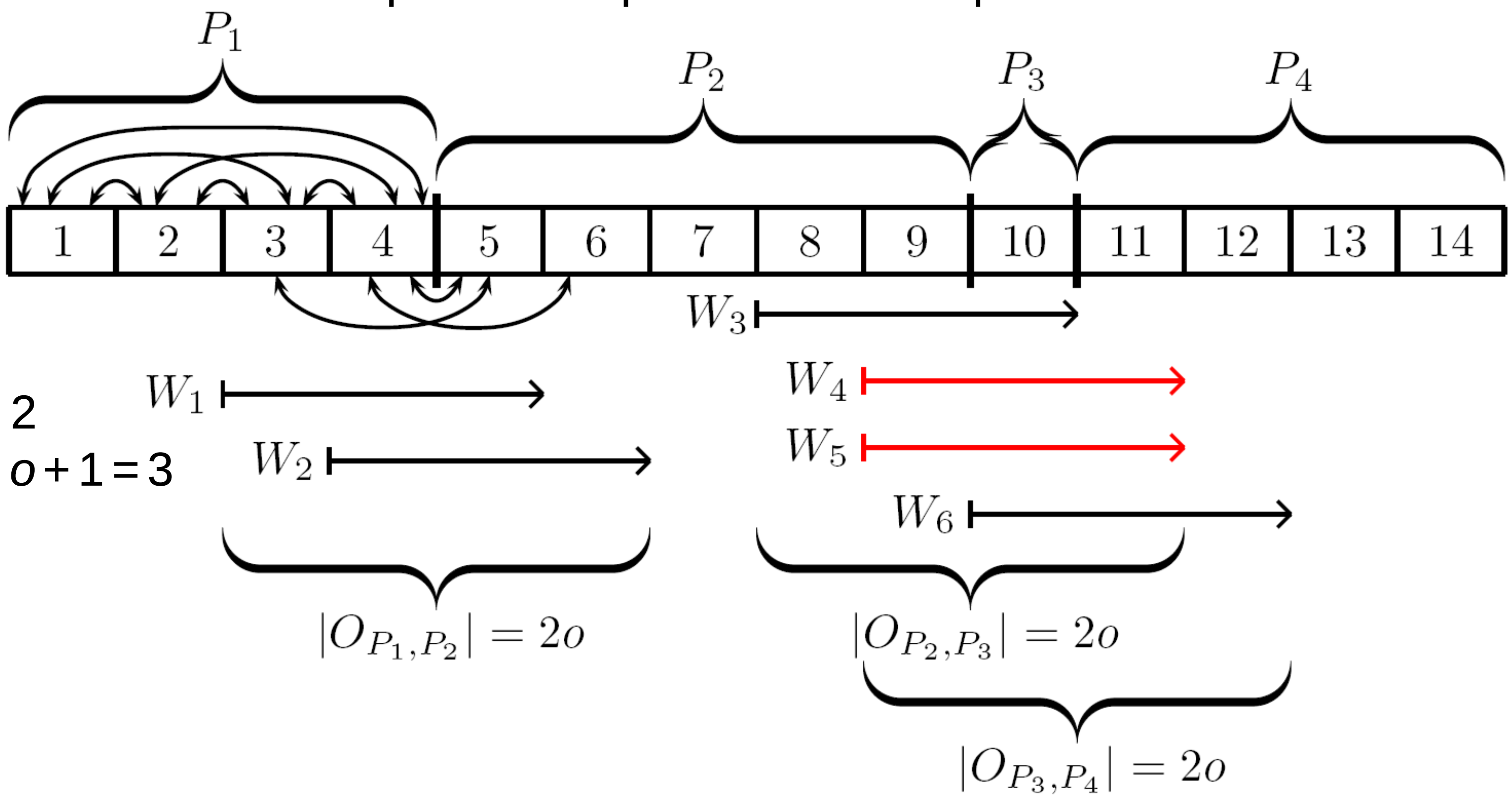
$o = 2$

$w = o+1=3$

Comparisons within overlap — Linear complexity

Sorted Blocks Method

Complete comparison within partitions



$o = 2$

$w = o + 1 = 3$

Comparisons within overlap

Complexity Analysis

	Method			
	Blocking	**Windowing**	**Sorted Blocks** (fixed partition size)	**Full enumeration**
Key generation	$O(n)$	$O(n)$	$O(n)$	---
Sorting	$O(n \log n)$	$O(n \log n)$	$O(n \log n)$	---
Detection	$O(\frac{n^2}{2b})$	$O(wn)$	$O(\frac{nm}{2})$	$O(\frac{n^2}{2})$
Overall	$O(n(\frac{n}{2b} + \log n))$	$O(n(w + \log n))$	$O(n(\frac{m}{2} + \log n))$	$O(\frac{n^2}{2})$

- *n* = number of tuples
- *b* = number of blocks
- *w* = window size
- *m* = partition size

Experiment

- Partitioning methods implemented in Java/MySQL environment
- Data Set
  - 9,763 records with audio CD information from freeDB
  - 298 real duplicates detected manually

[redacted]

- Sorting key
  - Concatenation of the first three letters of attributes *artist1*, *title1* and *track01*

Experiment

- Similarity Measure
  - Average similarity of attributes *artist1*, *title1* and *track01*
  - Not complex, but sufficient for comparing partition methods

$$f(t_1, t_2) = \frac{s(t_1[\text{Artist1}], t_2[\text{Artist1}]) + s(t_1[\text{Title1}], t_2[\text{Title1}]) + s(t_1[\text{Track01}], t_2[\text{Track01}])}{3}$$

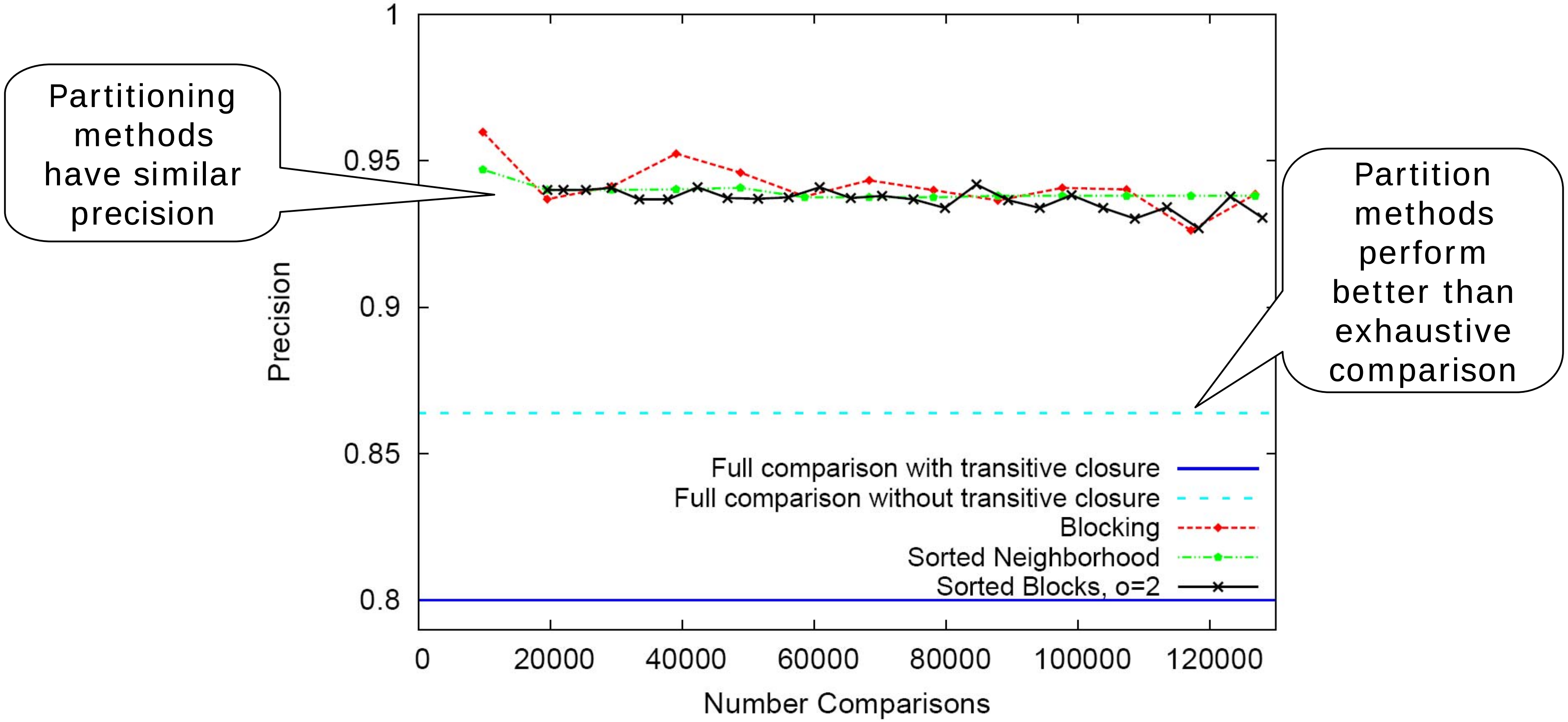
with:

$$s(x,y) = \begin{cases} = 1\text{, if } x = SubstringOf(y) \text{ or } y = SubstringOf(x) \\ = \text{threshold of similarity function, if } IsNull(x) \text{ or } IsNull(y) \\ = 1 - \dfrac{edit\_distance(x,y)}{max\{|x|,|y|\}} \text{ otherwise} \end{cases}$$

- Fixed size blocks/partitions have been used for simplicity
- *o*=2 delivers the best results for this data set

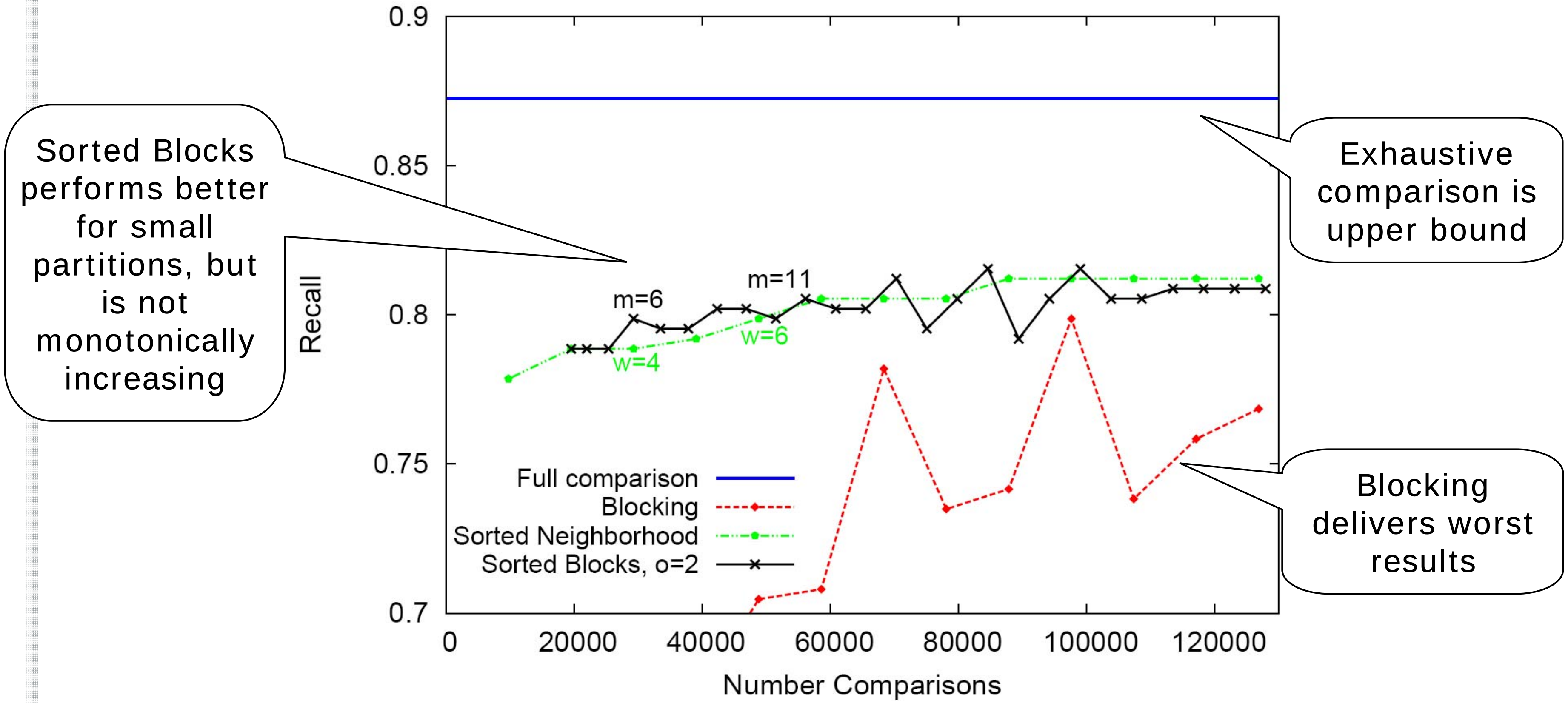
Experiment - Precision

- Precision: proportion of correctly identified duplicates
  - No. correctly identified duplicates / No. all identified duplicates



Experiment - Recall

- Recall: proportion of identified real-world duplicates
  - No. correctly identified duplicates / No. all duplicates



Conclusion

- **Results**
  - Sorted Neighborhood outperforms Blocking
  - Sorted Blocks parameterizes the degree of overlap from none (Blocking) to *w*-1 (Sorted Neighborhood)
  - Sorted Blocks outperforms Sorted Neighborhood slightly

- **Open issues for future research**
  - Confirm results by additional experiments with other data sets
  - Use variable partition sizes for experiments
  - Allow dynamic adaption of overlap
  - Multipass

Experiment – F-Measure

- F-Measure = harmonic mean of precision and recall