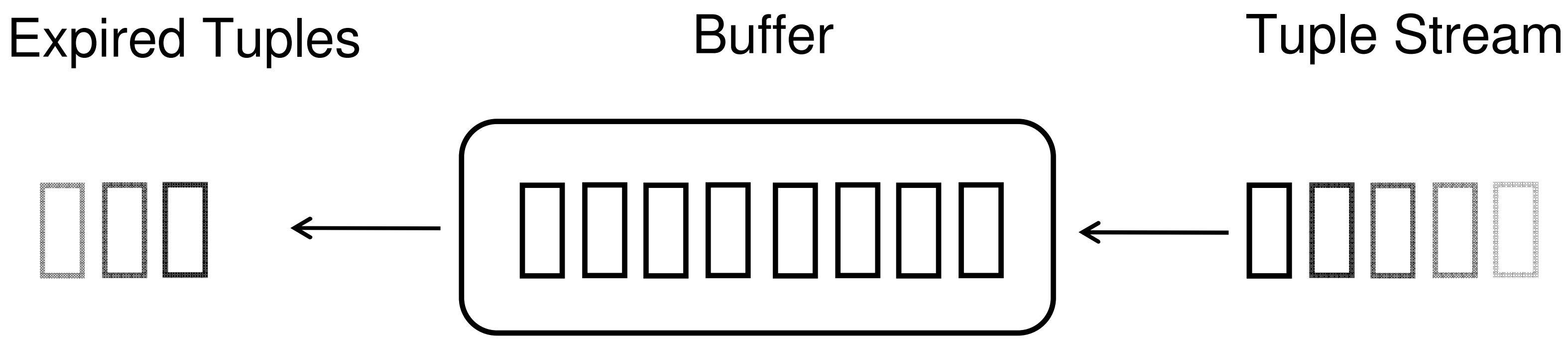
Ad-Hoc Top-k Query Answering for Data Streams

Gautam Das, Univ. of Texas at Arlington
Dimitrios Gunopulos, Univ. of California, Riverside
Nick Koudas, Univ. of Toronto
Nikos Sarkas, Univ. of Toronto

Data Stream



Top-k Queries

- Top-k queries on the contents of the buffer
- Previous work [MBP06]
  - Top-k query maintenance
  - *Static* queries
- Current work
  - Ad-hoc top-k queries
  - *Dynamic* queries

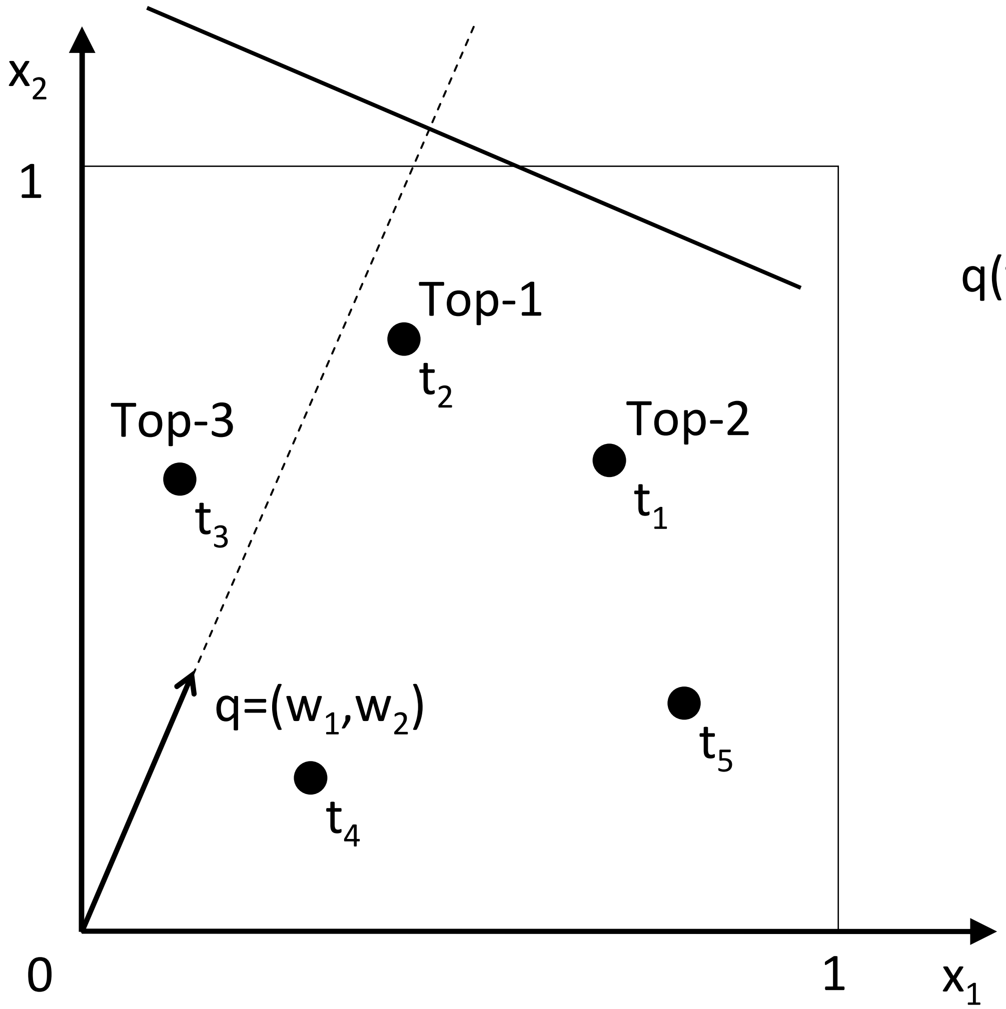
Outline

- Top-k query answering
  - Primal Plane
  - Dual Plane
- Arrangements
  - Representation
  - Operations
- Tuple Pruning
  - Principles
  - Implementation
- Experimental Evaluation

Outline

- Top-k query answering
  - Primal Plane
  - Dual Plane
- Arrangements
  - Representation
  - Operations
- Tuple Pruning
  - Principles
  - Implementation
- Experimental Evaluation

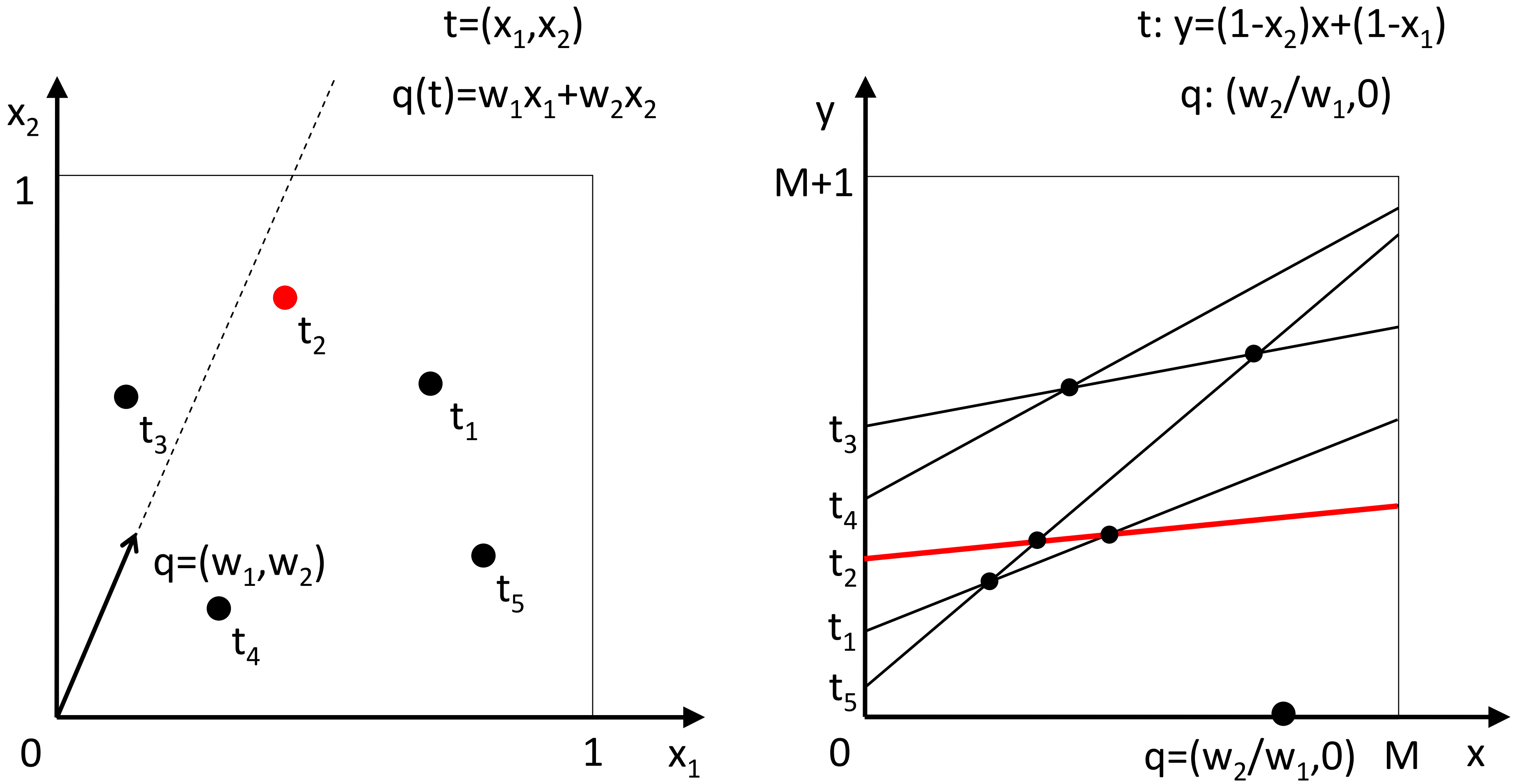
Primal Plane



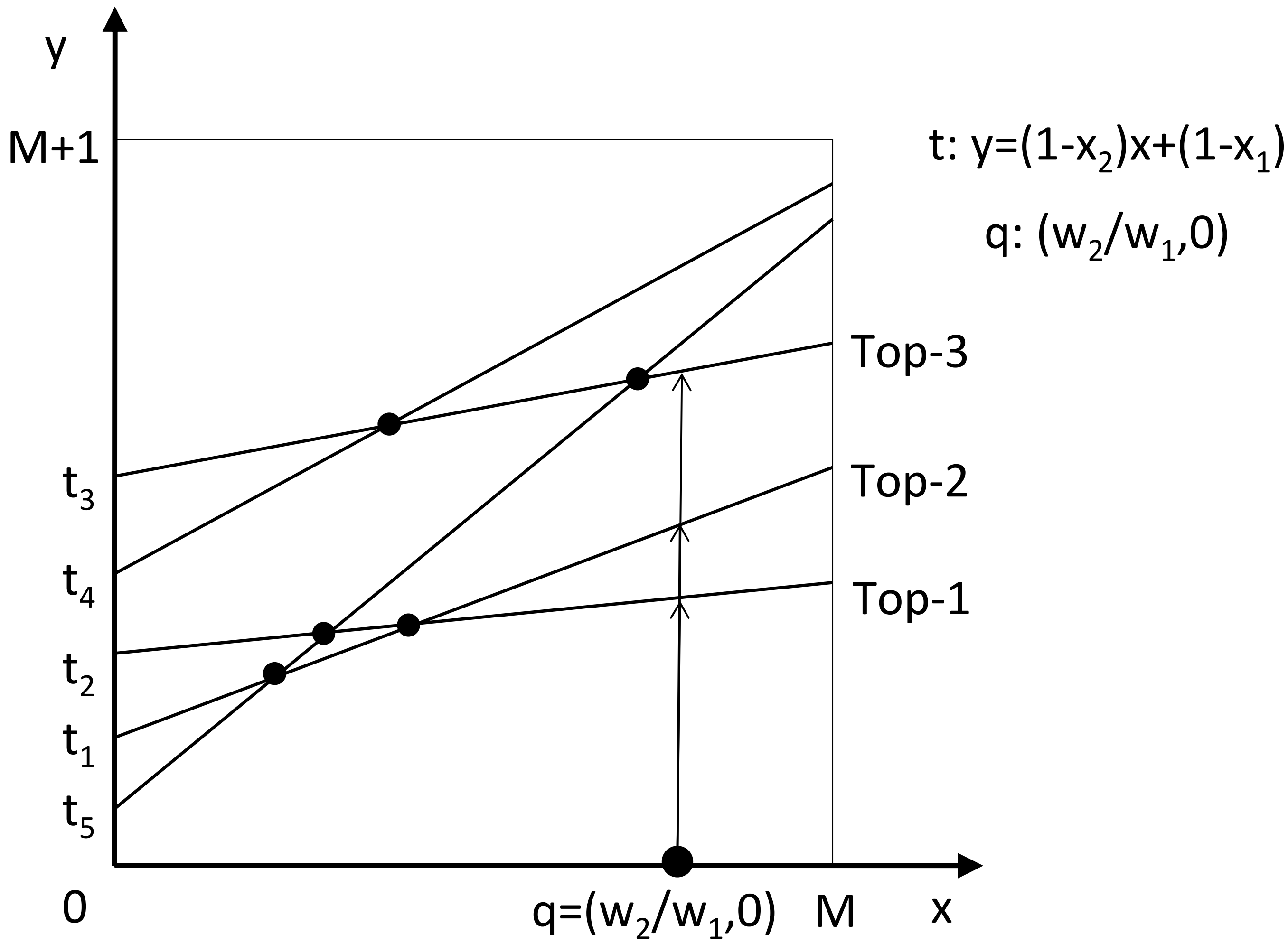
$t=(x_1,x_2)$

$q(t)=w_1x_1+w_2x_2$

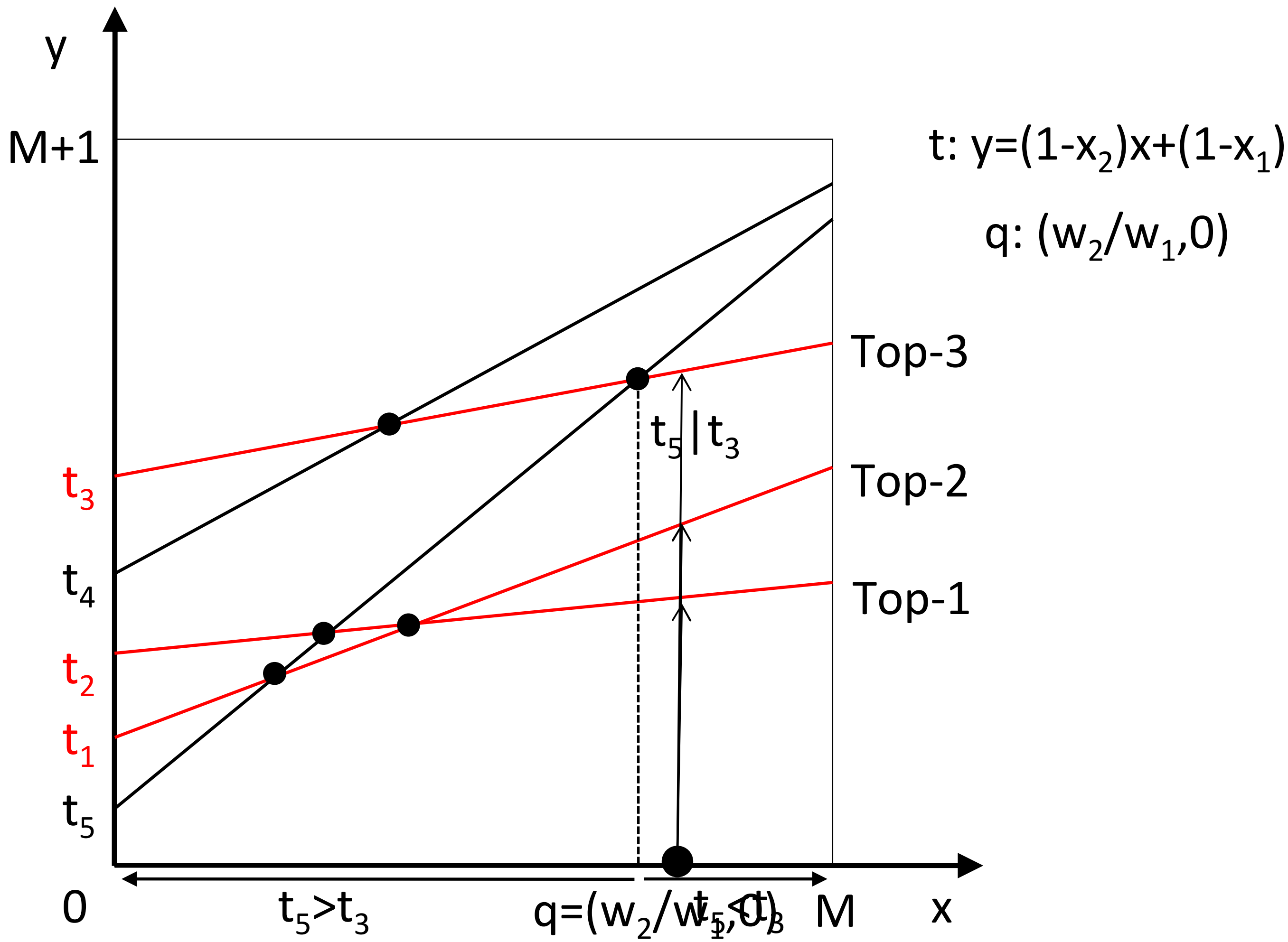
Primal-Dual Transformation



Dual Plane



Dual Plane

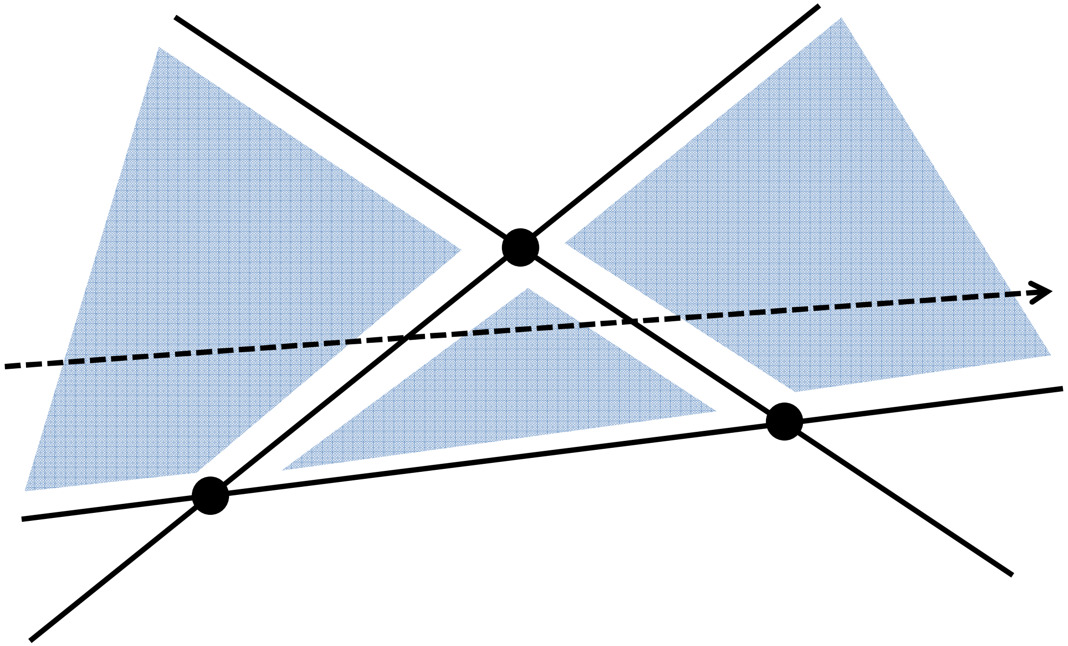


y
M+1
$t: y=(1-x_2)x+(1-x_1)$
$q: (w_2/w_1,0)$
Top-3
Top-2
Top-1
t_3
t_4
t_2
t_1
t_5
$t_5|t_3$
0
$t_5>t_3$
$q=(w_2/w_1,0)$
$t_5<t_3$
M
x

Outline

- Top-k query answering
  - Primal Plane
  - Dual Plane
- Arrangements
  - Representation
  - Operations
- Tuple Pruning
  - Principles
  - Implementation
- Experimental Evaluation

Arrangements: Introduction



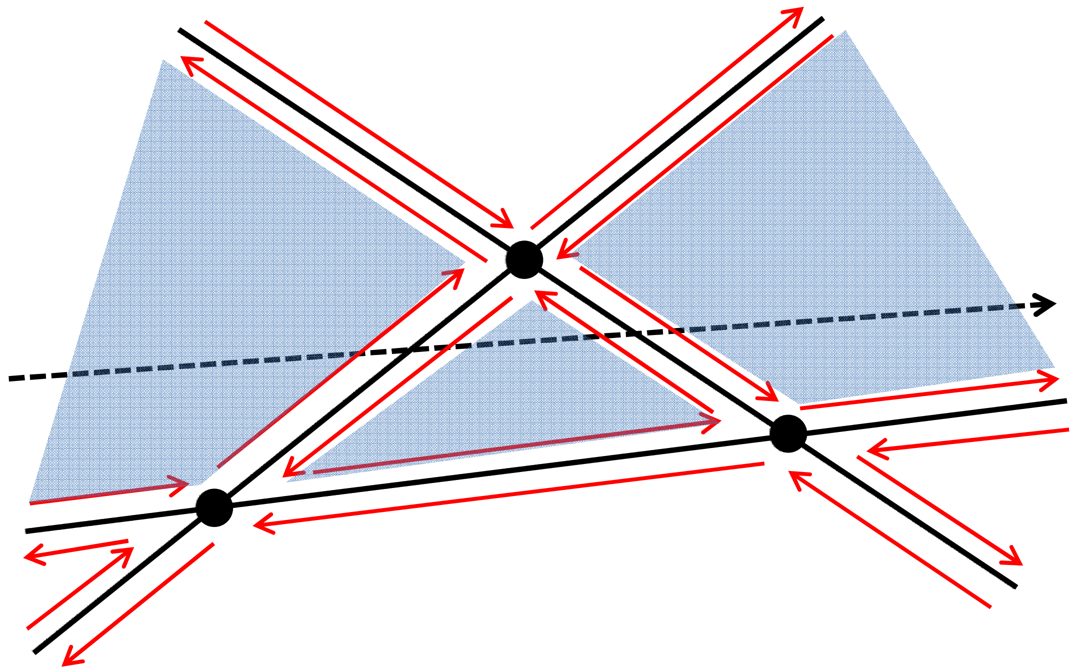
$O(n^2)$ vertices

$O(n^2)$ edges

$O(n^2)$ faces

Zone: $O(n)$ edges

Arrangements: Representation



Vertices

$O(n^2)$ vertices

Twin Half-edges

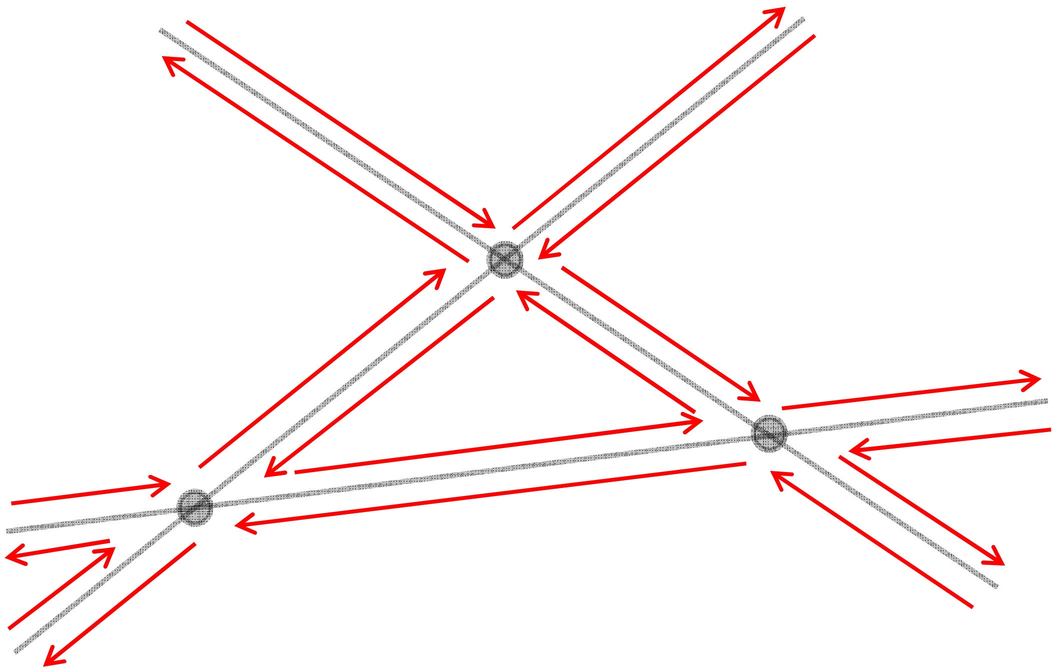
$O(n^2)$ edges

Connected Boundaries

$O(n^2)$ faces

Zone: $O(n)$ edges

Arrangements: Representation

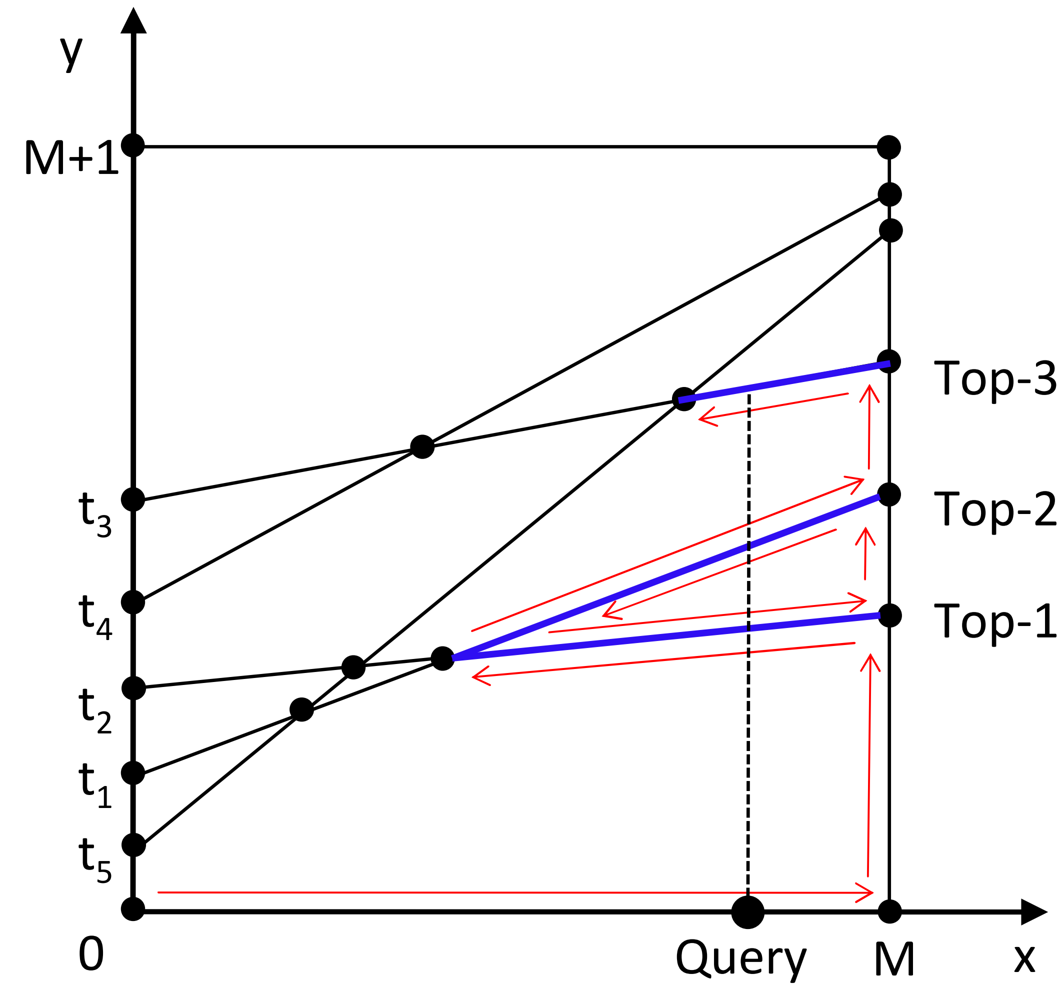


Vertices

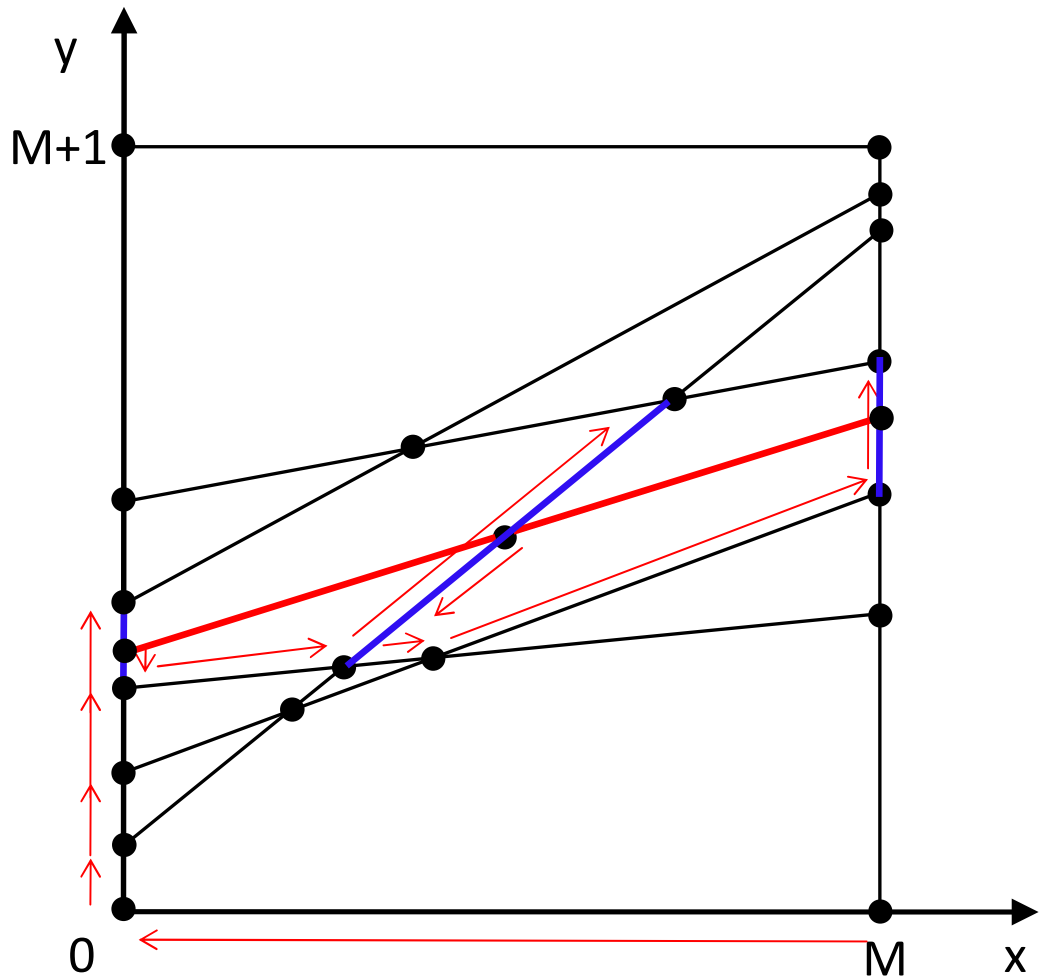
Twin Half-edges

Connected Boundaries

Arrangements: Top-k Query Answering



Arrangements: Tuple Insertion



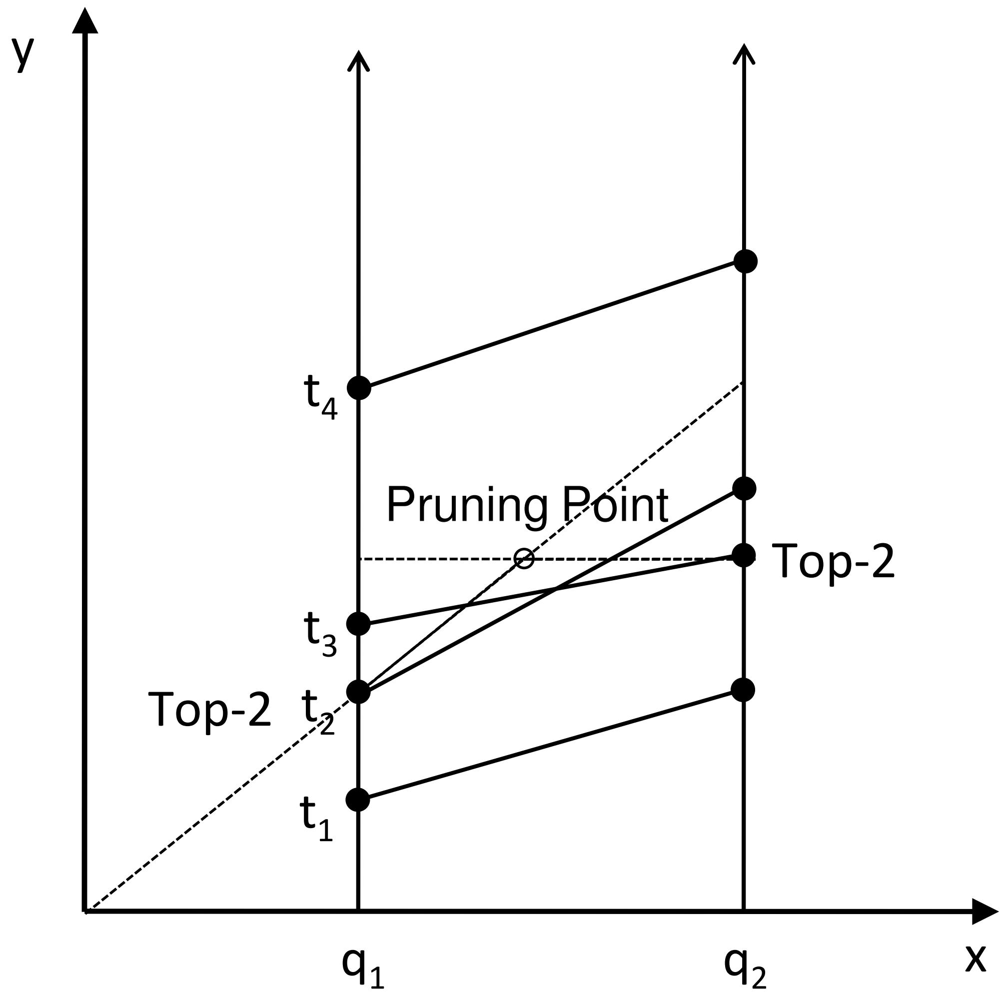
Contributions so far

- Dual representation of top-k problem
- Use of arrangements and development of algorithms
  - O(n) query answering, O(k) in practice
  - O(n) insertion and deletion
  - $O(n^2)$ space overhead
- Benefits
  - *Non-redundant, self-organizing representation of the ranking of all possible top-k queries*
- Still, we can do much better
  - O(k *log*n) operations
  - $O(k^2 \, log^2 n)$ space overhead

Outline

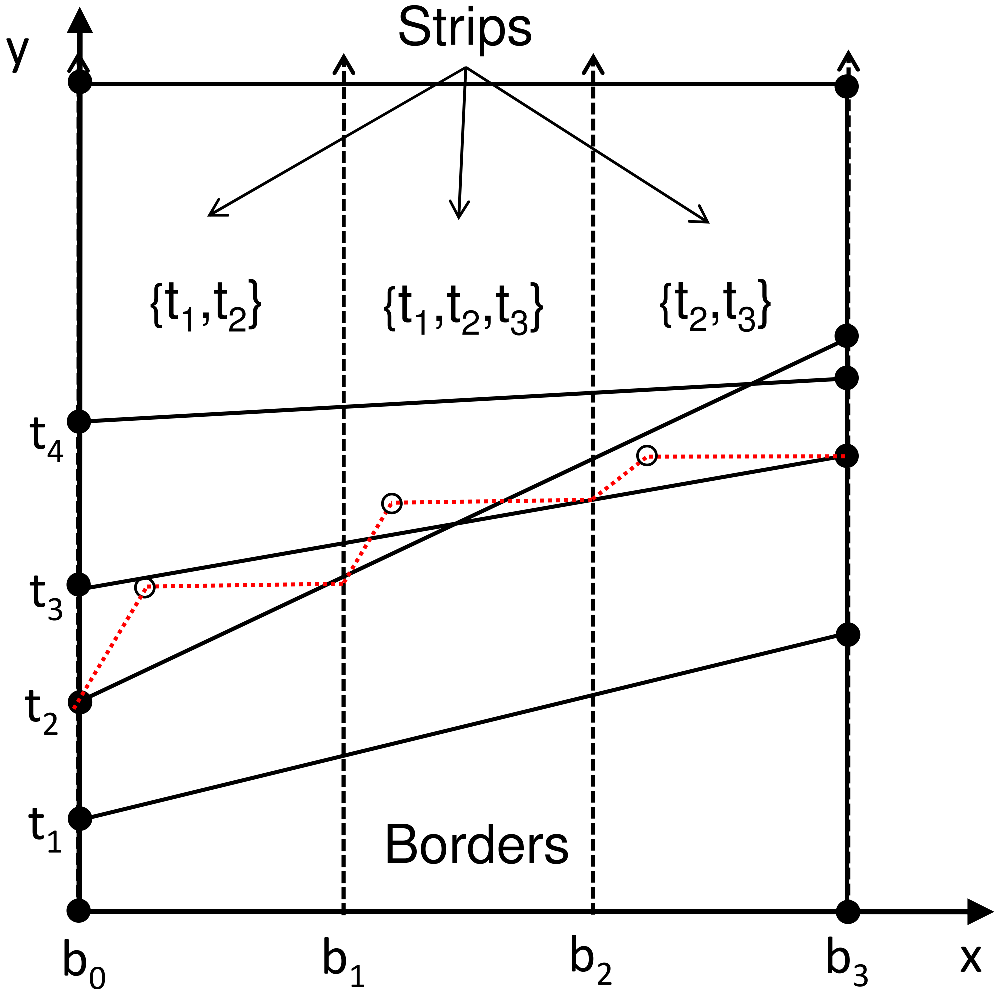
- Top-k query answering
  - Primal Plane
  - Dual Plane
- Arrangements
  - Representation
  - Operations
- Tuple Pruning
  - Principles
  - Implementation
- Experimental Evaluation

Tuple Pruning



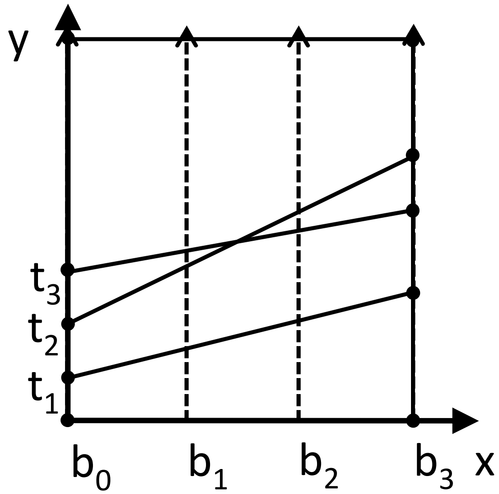
y=ax+b
0≤a,b≤1

Tuple Pruning

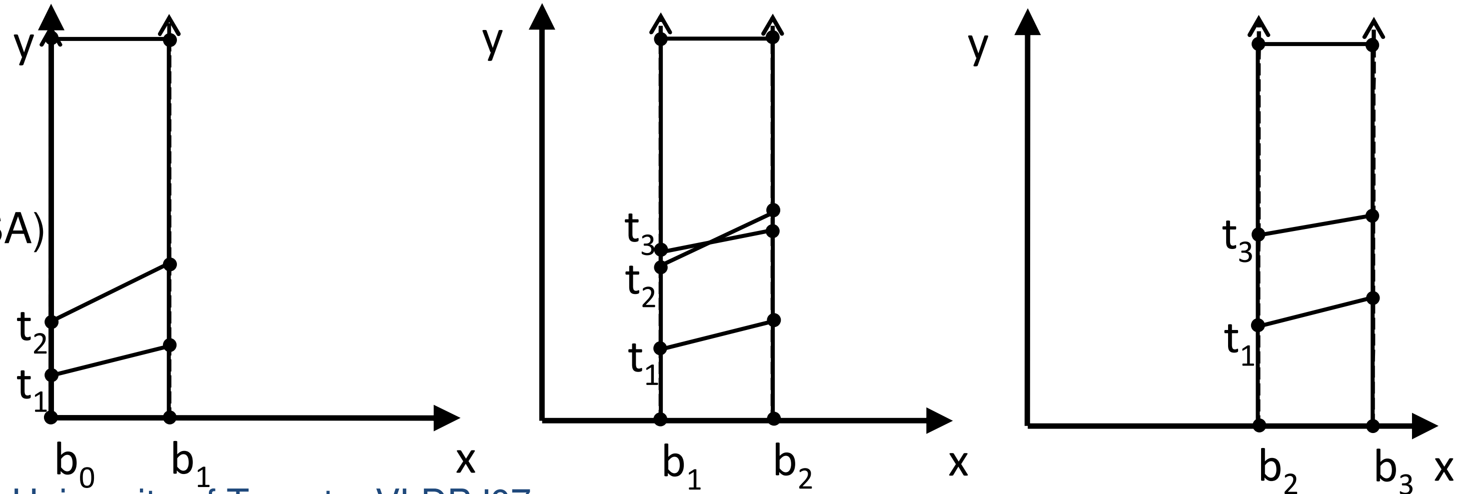


Storing the Pruned Dataset

Full Arrangement (FA)



Strip Arrangements (SA)



Pruning Efficiency

- Size of the filtered dataset is O(k *log*n)
- Thus, O(k *log*n) operations on the arrangement
- Example
  - Top-20 queries
  - 1 million 2d uniformly distributed tuples
  - 16 borders
  - Only 250 tuples need to be stored in the arrangement!

Outline

- Top-k query answering
  - Primal Plane
  - Dual Plane
- Arrangements
  - Representation
  - Operations
- Tuple Pruning
  - Principles
  - Implementation
- Experimental Evaluation

Challenges

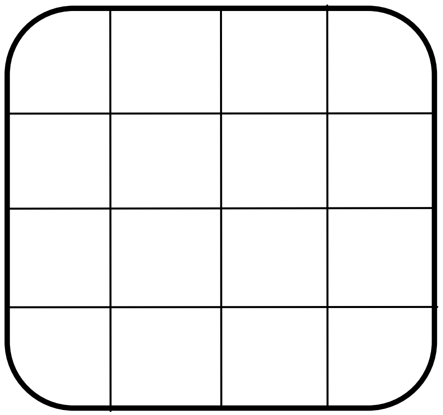
- Maintain relevant tuples in the presence of streaming updates
- Procedure
  - Update the top-k results along the borders
  - Update the pruning points
  - For each strip, update the tuples that fall below the corresponding pruning point
  - Update arrangement

Solutions

- Maintain the top-k result along the borders
  - Top-k query maintenance techniques [MBP06]
- For each strip, update the tuples that fall below the corresponding pruning point
  - Half-space range searching in the primal plane!
- Index the buffer using a grid

Maintaining the Pruned Dataset

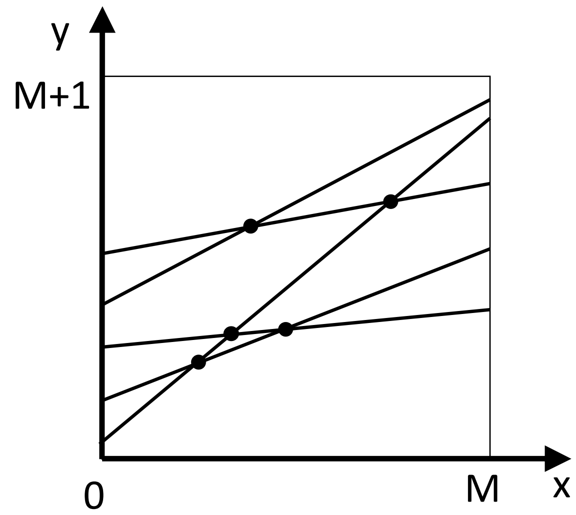
Buffer/Grid
Buffer



Border Maintenance

Half-space
Range Searching

Arrangement



Placing the Borders

- Increasing the number of borders increases the pruning efficiency and overhead
- Objective
  - Equi-depth partitioning
- Heuristic
  - Iteratively split strips until strips have less than a certain number of vertices (*strip complexity*)

Outline

- Top-k query answering
  - Primal Plane
  - Dual Plane
- Arrangements
  - Representation
  - Operations
- Tuple Pruning
  - Principles
  - Implementation
- Experimental Evaluation

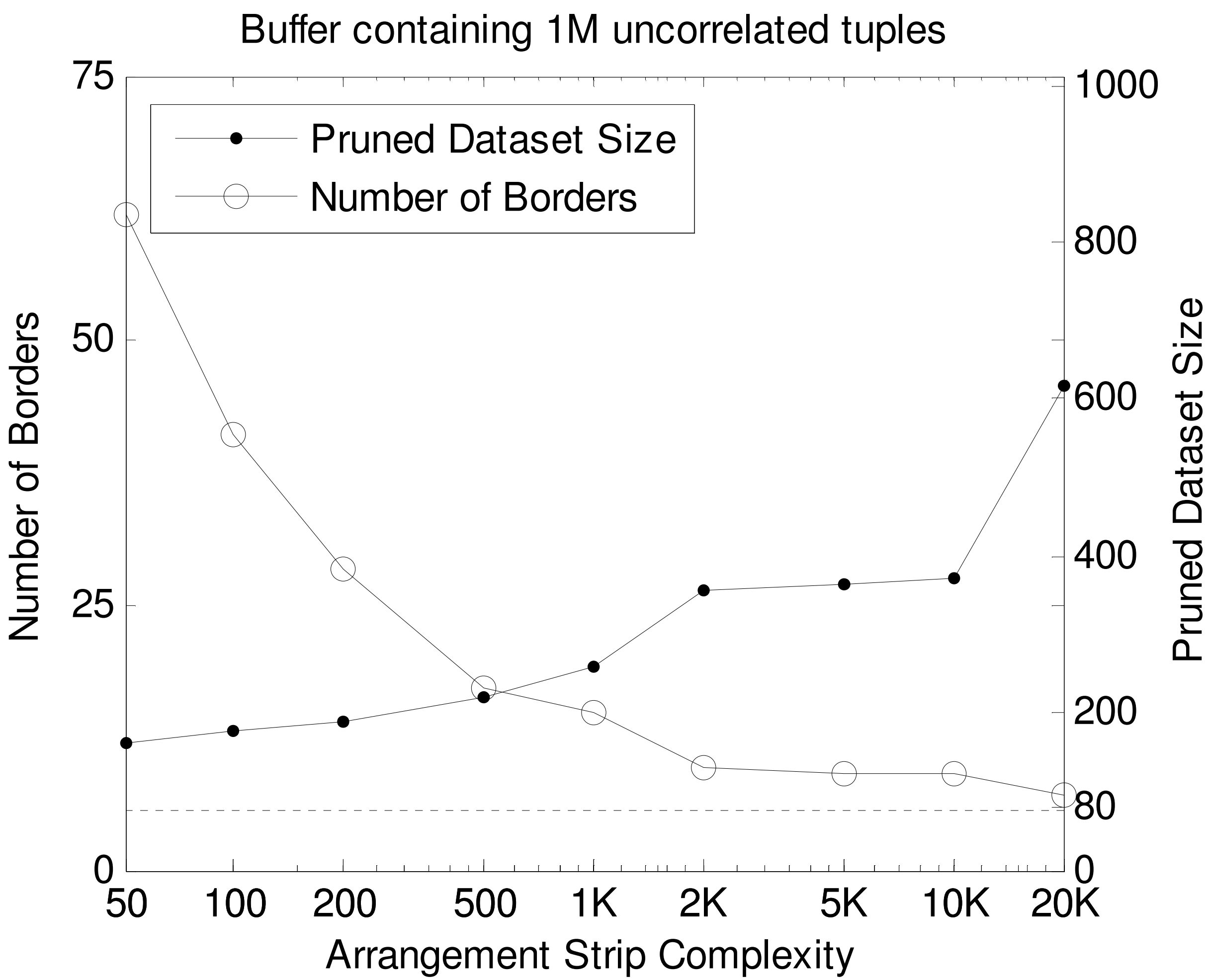
Experimental Setting

- Data sets
  - Synthetic: uniform, correlated, anti-correlated
  - Real: Intel Lab data
- Experiments
  - Pruning Efficiency
  - Memory overhead
  - Variable buffer size, stream rate, query results (k), query frequency, dimensionality

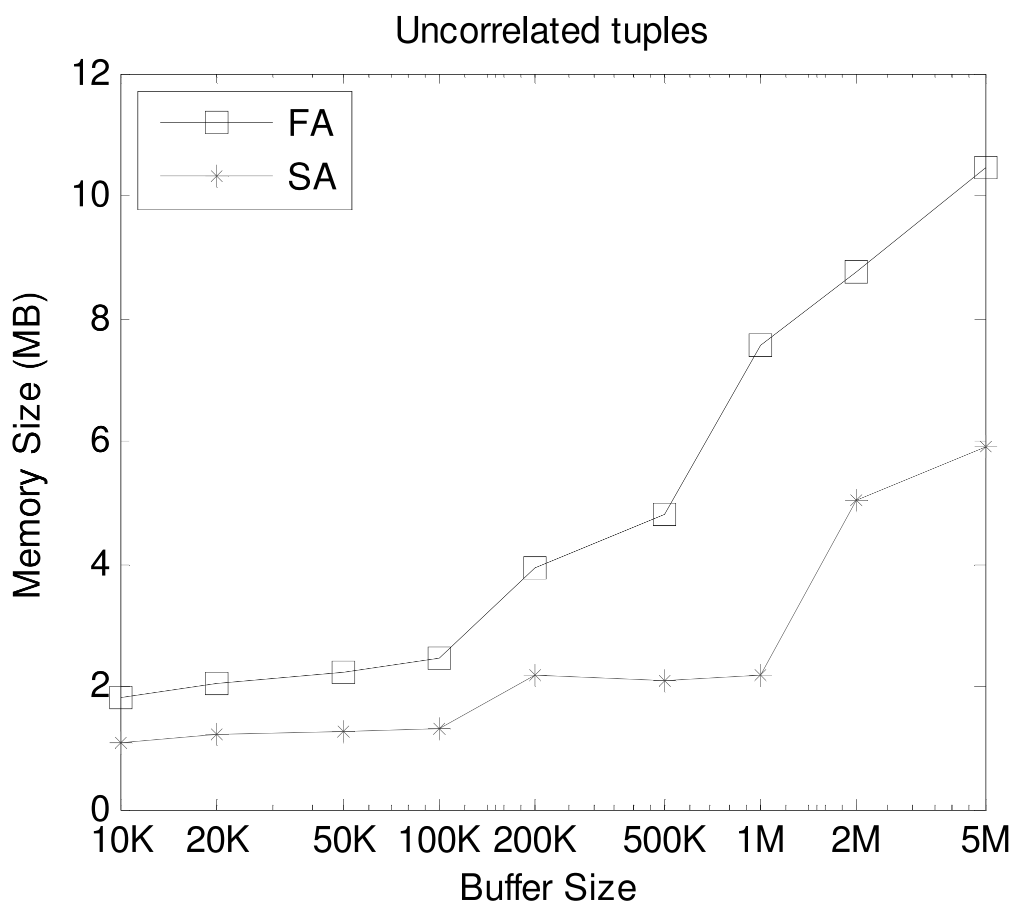
Experimental Setting

- Data sets
  - Synthetic: uniform, correlated, anti-correlated
  - Real: Intel Lab data
- Experiments
  - Pruning Efficiency
  - Memory overhead
  - Variable buffer size, stream rate, query results (k), query frequency

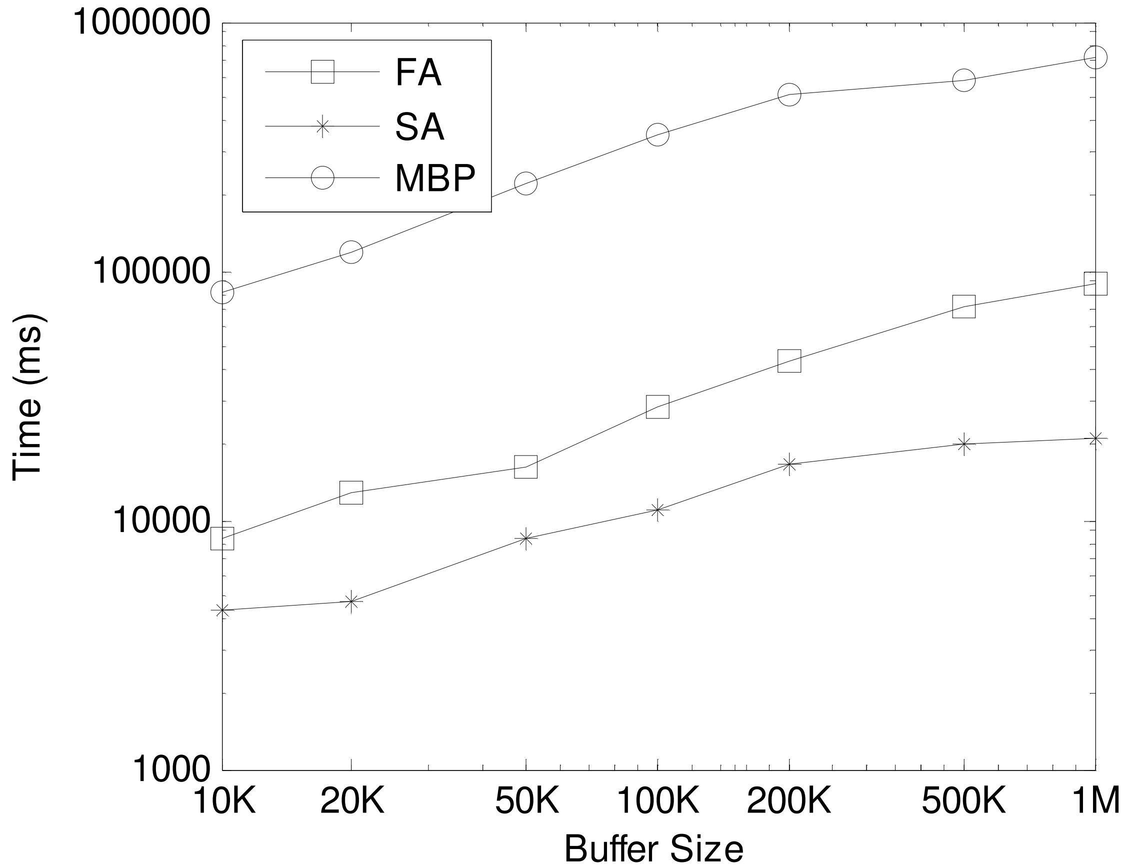
Pruning Efficiency



Memory Overhead



Real Data



1000000
100000
10000
1000
Time (ms)
FA
SA
MBP
10K
20K
50K
100K
200K
500K
1M
Buffer Size

Conclusions

- Dual space representation of the top-k problem
- Use of arrangements
- Tuple pruning technique

Thank you!

References

- [MBP06], K. Mouratidis, S. Bakiras, D. Papadias: Continuous Monitoring of Top-k Queries over Sliding Windows, *SIGMOD* 2006.

Synthetic Data