



Title	Continuous inverse ranking queries in uncertain streams
Author(s)	Bernecker, T; Kriegel, HP; Mamoulis, N; Renz, M; Zuefle, A
Citation	The 23rd International Conference on Scientific and Statistical Database Management (SSDBM 2011), Portland, OR., 20-22 July 2011. In Lecture Notes in Computer Science, 2011, v. 6809, p. 37-54
Issued Date	2011
URL	http://hdl.handle.net/10722/152003
Rights	Creative Commons: Attribution 3.0 Hong Kong License

Continuous Inverse Ranking Queries in Uncertain Streams

Thomas Bernecker^{*}, Hans-Peter Kriegel^{*},
Nikos Mamoulis^{**}, Matthias Renz^{*} and Andreas Zuefle^{*}

^{*})
Ludwig-Maximilians-Universität München (LMU)
Munich, Germany
<http://www.dbs.ifi.lmu.de>
{bernecker, kriegel, renz, zuefle}@dbs.ifi.lmu.de

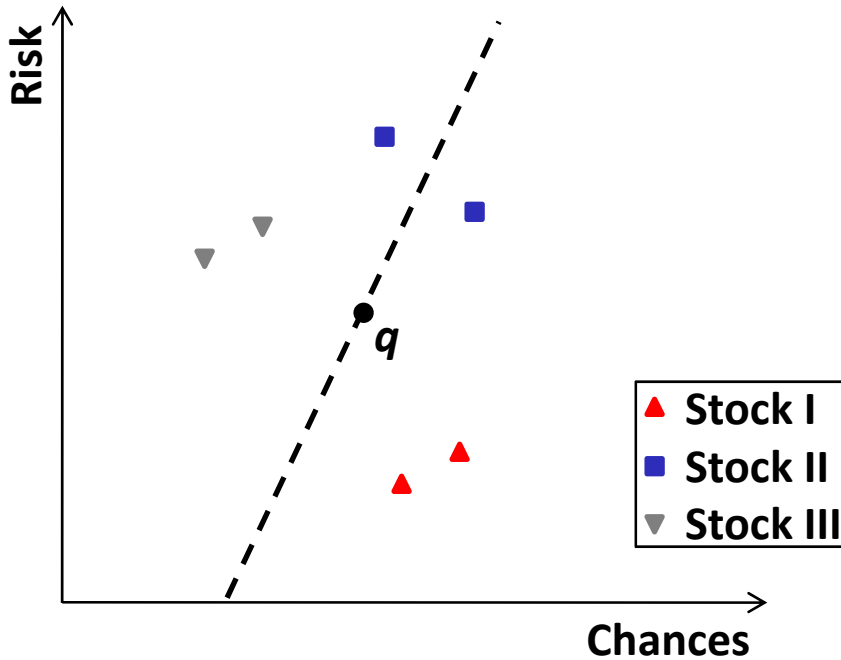
^{**})
University of Hong Kong (HKU)
Hong Kong
<http://www.cs.hku.hk>
nikos@cs.hku.hk



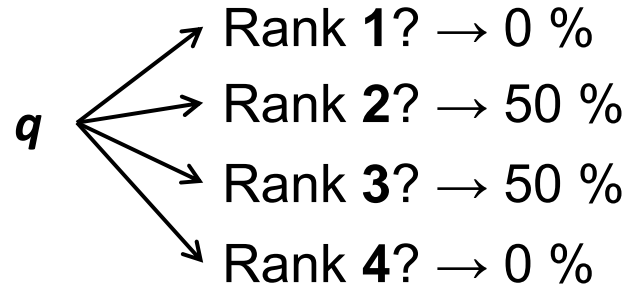
1. Motivation: Probabilistic Inverse Ranking
2. Continuous Inverse Ranking Queries
 - Initial Computation
 - Incremental Processing
3. Experimental Evaluation
4. Summary

Probabilistic Inverse Ranking

- Identification of the significance of objects among peers
 - *Inverse Ranking*: Return the position of the query object q w.r.t. the score function S
 - *Probabilistic Inverse Ranking*: Find all possible positions of q
- Example: Stock rating system



S = Chances - Risk



- Probabilistic Inverse Ranking (PIR) Query
 - Probabilistic database **DB** where $|DB| = n$
 - Uncertain object **o**: m alternative locations (discrete uncertainty) or pdf (continuous uncertainty)
 - Query object **q**
 - Score function $\mathbf{S} : DB \rightarrow \mathbb{R}_0^+$
 - Definition: $\forall i = 1, \dots, k : P(\mathbf{q} \text{ is on rank } i \text{ w.r.t. } \mathbf{S}) = P_q^t(i)$
 - There exist exactly $i - 1$ objects $\mathbf{o} \in DB$ with $\mathbf{S}(\mathbf{o}) > \mathbf{S}(\mathbf{q})$

- Challenge: Application to dynamic data
 - General stream model with location updates retrieved at a time \mathbf{t}
 - $P(\mathbf{q} \text{ is on rank } i \text{ at time } \mathbf{t}) = P_q^t(i)$
 - Initial computation
 - Incremental processing

Initial Computation (1)

- Initial time t : Compute $P_q^t(i) \quad \forall i = 1, \dots, k$
 - Object $o \in DB$: $p_o^t = P(S(o) > S(q))$ at time t
 - j objects have been processed so far (o_j is the latest)
 - Successive processing by the *Poisson Binomial Recurrence (PBR)*:

$$P_{i,j}^t = \begin{cases} 1 & \text{if } i = 0 \wedge j = 0 \\ 0 & \text{if } i < 0 \vee i > j \\ \underbrace{P_{i-1,j-1}^t \cdot p_{o_j}^t}_{\text{callout 1}} + \underbrace{P_{i,j-1}^t \cdot (1 - p_{o_j}^t)}_{\text{callout 2}} & \text{else} \end{cases}$$

i out of j :
 $S(o) > S(q)$

$i-1$ out of $j-1$:
 $S(o) > S(q)$
and
 $S(o_j) > S(q)$

i out of $j-1$:
 $S(o) > S(q)$
and
 $S(o_j) \leq S(q)$

Initial Computation (2)

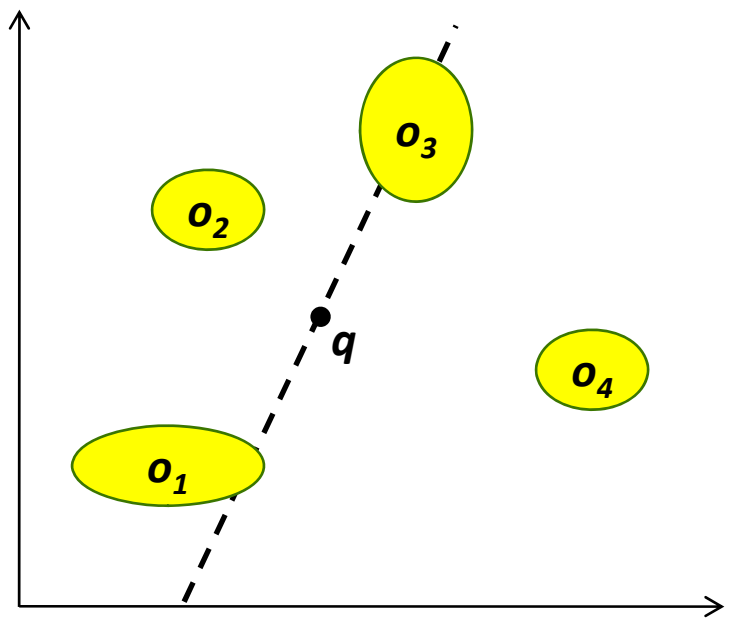
- $j = n (\forall i = 0, \dots, k-1)$: $P_{i,j}^t = P_{i,n}^t = P_q^t(i+1)$
 \Rightarrow PIR result for $q \Rightarrow$ runtime: $O(k \cdot n)$
- Optimizations:
 - $p_o^t = 0 \Rightarrow$ o has no effect on the rank of q
 - $p_o^t = 1 \Rightarrow$ increment counter C^t
- General case ($0 < p_o^t < 1$) \Rightarrow process o by PBR: $\forall i = 0, \dots, k-1$:
 $P(i$ objects processed by PBR have a higher score than $q) = P_{PBR}^t(i)$
- Initial PIR result:

$$P_q^t(i) = \begin{cases} P_{PBR}^t(i-1-C^t) & \text{if } C^t + 1 \leq i \leq C^t + 1 + k \\ 0 & \text{else} \end{cases}$$

Initial Computation (3)

- Example: $n = 4, k = 2$

$$p_{o_1}^t = 0.1 \quad p_{o_2}^t = 0 \quad p_{o_3}^t = 0.6 \quad p_{o_4}^t = 1 \quad C^t = 0$$



Initial Computation (3)

- Example: $n = 4$, $k = 2$, $j = 1$

$$p_{o_1}^t = 0.1 \quad p_{o_2}^t = 0 \quad p_{o_3}^t = 0.6 \quad p_{o_4}^t = 1 \quad C^t = 0$$

- $j = 1$: $P_{0,1}^t = P_{-1,0}^t \cdot p_{o_1}^t + P_{0,0}^t \cdot (1 - p_{o_1}^t) = 0 \cdot 0.1 + 1 \cdot 0.9 = 0.9$

$$P_{1,1}^t = P_{0,0}^t \cdot p_{o_1}^t + P_{1,0}^t \cdot (1 - p_{o_1}^t) = 1 \cdot 0.1 + 0 \cdot 0.9 = 0.1$$

Initial Computation (3)

- Example: $n = 4$, $k = 2$, $j = 2$

$$p_{o_1}^t = 0.1 \quad p_{o_2}^t = 0 \quad p_{o_3}^t = 0.6 \quad p_{o_4}^t = 1 \quad C^t = 0$$

- $j = 1$: $P_{0,1}^t = P_{-1,0}^t \cdot p_{o_1}^t + P_{0,0}^t \cdot (1 - p_{o_1}^t) = 0 \cdot 0.1 + 1 \cdot 0.9 = 0.9$

$$P_{1,1}^t = P_{0,0}^t \cdot p_{o_1}^t + P_{1,0}^t \cdot (1 - p_{o_1}^t) = 1 \cdot 0.1 + 0 \cdot 0.9 = 0.1$$

Initial Computation (3)

- Example: $n = 4$, $k = 2$, $j = 3$

$$p_{o_1}^t = 0.1 \quad p_{o_2}^t = 0 \quad p_{o_3}^t = 0.6 \quad p_{o_4}^t = 1 \quad C^t = 0$$

- $j = 1$: $P_{0,1}^t = P_{-1,0}^t \cdot p_{o_1}^t + P_{0,0}^t \cdot (1 - p_{o_1}^t) = 0 \cdot 0.1 + 1 \cdot 0.9 = 0.9$

$$P_{1,1}^t = P_{0,0}^t \cdot p_{o_1}^t + P_{1,0}^t \cdot (1 - p_{o_1}^t) = 1 \cdot 0.1 + 0 \cdot 0.9 = 0.1$$

- $j = 3$: $P_{0,2}^t = P_{-1,1}^t \cdot p_{o_3}^t + P_{0,1}^t \cdot (1 - p_{o_3}^t) = 0 \cdot 0.6 + 0.9 \cdot 0.4 = 0.36$

$$P_{1,2}^t = P_{0,1}^t \cdot p_{o_3}^t + P_{1,1}^t \cdot (1 - p_{o_3}^t) = 0.9 \cdot 0.6 + 0.1 \cdot 0.4 = 0.58$$

Initial Computation (3)

- Example: $n = 4$, $k = 2$, $j = 4$

$$p_{o_1}^t = 0.1 \quad p_{o_2}^t = 0 \quad p_{o_3}^t = 0.6 \quad p_{o_4}^t = 1 \quad C^t = 1$$

- $j = 1$: $P_{0,1}^t = P_{-1,0}^t \cdot p_{o_1}^t + P_{0,0}^t \cdot (1 - p_{o_1}^t) = 0 \cdot 0.1 + 1 \cdot 0.9 = 0.9$

$$P_{1,1}^t = P_{0,0}^t \cdot p_{o_1}^t + P_{1,0}^t \cdot (1 - p_{o_1}^t) = 1 \cdot 0.1 + 0 \cdot 0.9 = 0.1$$

- $j = 3$: $P_{0,2}^t = P_{-1,1}^t \cdot p_{o_3}^t + P_{0,1}^t \cdot (1 - p_{o_3}^t) = 0 \cdot 0.6 + 0.9 \cdot 0.4 = 0.36$

$$P_{1,2}^t = P_{0,1}^t \cdot p_{o_3}^t + P_{1,1}^t \cdot (1 - p_{o_3}^t) = 0.9 \cdot 0.6 + 0.1 \cdot 0.4 = 0.58$$

Initial Computation (3)

- Example: $n = 4, k = 2$

$$p_{o_1}^t = 0.1 \quad p_{o_2}^t = 0 \quad p_{o_3}^t = 0.6 \quad p_{o_4}^t = 1 \quad C^t = 1$$

- $j = 1$: $P_{0,1}^t = P_{-1,0}^t \cdot p_{o_1}^t + P_{0,0}^t \cdot (1 - p_{o_1}^t) = 0 \cdot 0.1 + 1 \cdot 0.9 = 0.9$

$$P_{1,1}^t = P_{0,0}^t \cdot p_{o_1}^t + P_{1,0}^t \cdot (1 - p_{o_1}^t) = 1 \cdot 0.1 + 0 \cdot 0.9 = 0.1$$

- $j = 3$: $P_{0,2}^t = P_{-1,1}^t \cdot p_{o_3}^t + P_{0,1}^t \cdot (1 - p_{o_3}^t) = 0 \cdot 0.6 + 0.9 \cdot 0.4 = 0.36 = P_{PBR}^t(0)$

$$P_{1,2}^t = P_{0,1}^t \cdot p_{o_3}^t + P_{1,1}^t \cdot (1 - p_{o_3}^t) = 0.9 \cdot 0.6 + 0.1 \cdot 0.4 = 0.58 = P_{PBR}^t(1)$$

- Initial PIR result: $P_q^t(1) = P_{PBR}^t(1 - 1 - 1) = P_{PBR}^t(-1) = 0$

$$P_q^t(2) = P_{PBR}^t(2 - 1 - 1) = P_{PBR}^t(0) = 0.36$$

Incremental Processing (1)

- Location update of one alternative location of object \mathbf{o} :
compute $P_q^t(i) \forall i = 1, \dots, k$
- Naive solution: Apply PBR $\Rightarrow O(n) \forall i = 1, \dots, k$
- Enhanced solution: just consider update of \mathbf{o}
 $\Rightarrow O(1) \forall i = 1, \dots, k$
 - Phase 1
 - Remove effect of old value p_o^t from $P_{PBR}^t(i) \forall i = 0, \dots, k-1$
 - Obtain intermediate result $\hat{P}_{PBR}^{t+1}(i)$
 - Phase 2
 - Incorporate effect of new value p_o^{t+1} in $\hat{P}_{PBR}^{t+1}(i)$
 - Obtain new PIR result $P_q^{t+1}(i)$

- Phase 1: Three cases

- $p_o^t = 0 \Rightarrow \hat{P}_{PBR}^t(i) = P_{PBR}^t(i)$

- $p_o^t = 1 \Rightarrow \hat{P}_{PBR}^t(i) = P_{PBR}^t(i)$ and $C^{t+1} = C^t - 1$

- $0 < p_o^t < 1 \Rightarrow$ remove p_o^t from $P_{PBR}^t(i)$

$$P_{PBR}^t(i) = \hat{P}_{PBR}^t(i-1) \cdot p_o^t + \hat{P}_{PBR}^t(i) \cdot (1 - p_o^t)$$

$$\hat{P}_{PBR}^t(i) = \frac{P_{PBR}^t(i) - \hat{P}_{PBR}^t(i-1) \cdot p_o^t}{1 - p_o^t}$$

$$\hat{P}_{PBR}^t(0) = \frac{P_{PBR}^t(0)}{1 - p_o^t}$$

- Phase 2: Three cases

- $p_o^{t+1} = 0 \Rightarrow P_{PBR}^{t+1}(i) = \hat{P}_{PBR}^t(i)$

- $p_o^{t+1} = 1 \Rightarrow P_{PBR}^{t+1}(i) = \hat{P}_{PBR}^t(i)$ and $C^{t+1} = C^t + 1$

- $0 < p_o^{t+1} < 1 \Rightarrow$ compute $P_{PBR}^{t+1}(i)$ applying PBR

$$P_{PBR}^{t+1}(i) = \hat{P}_{PBR}^t(i-1) \cdot p_o^{t+1} + \hat{P}_{PBR}^t(i) \cdot (1 - p_o^{t+1})$$

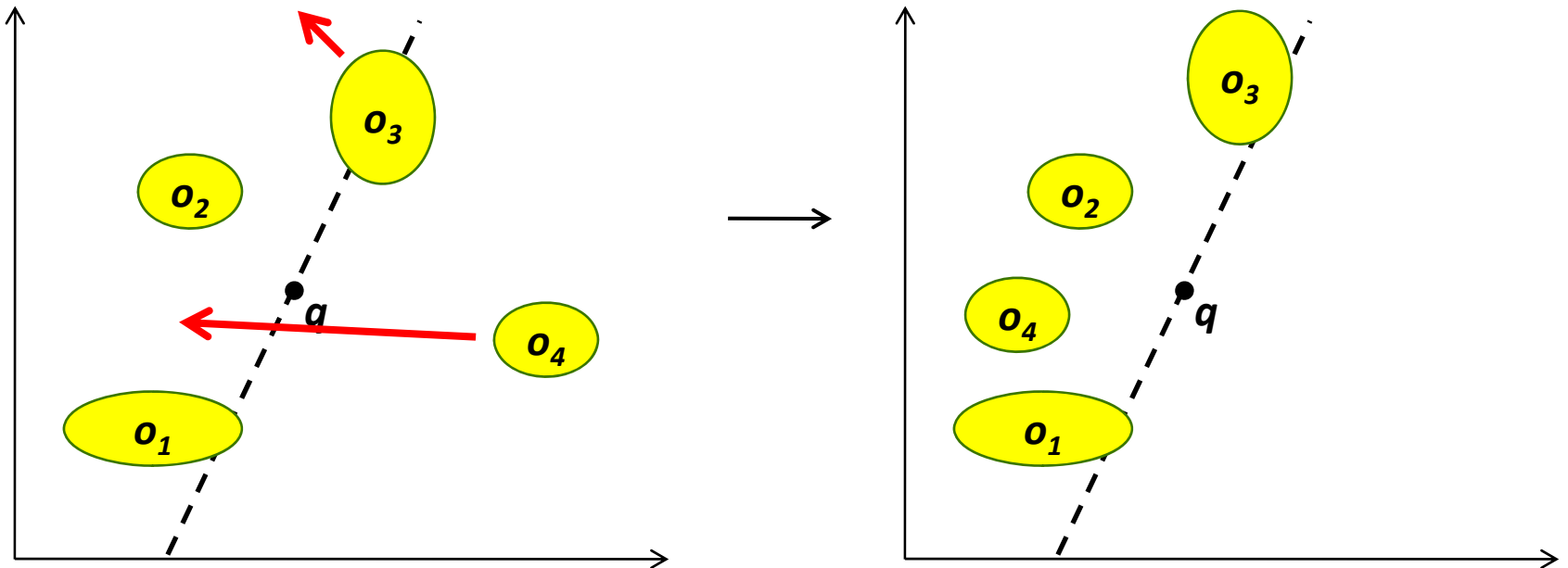
- New PIR result:

$$P_q^{t+1}(i) = \begin{cases} P_{PBR}^{t+1}(i-1-C^{t+1}) & \text{if } C^{t+1} + 1 \leq i \leq C^{t+1} + 1 + k \\ 0 & \text{else} \end{cases}$$

Incremental Processing (4)

- Example: $n = 4, k = 2$

$$p_{o_1}^t = 0.1 \quad p_{o_2}^t = 0 \quad p_{o_3}^t = 0.6 \rightarrow p_{o_3}^{t+1} = 0.2 \quad p_{o_4}^t = 1 \rightarrow p_{o_4}^{t+2} = 0 \quad C^t = 1$$



- Example: $n = 4, k = 2$

$$p_{o_1}^t = 0.1 \quad p_{o_2}^t = 0 \quad p_{o_3}^t = 0.6 \rightarrow p_{o_3}^{t+1} = 0.2 \quad p_{o_4}^t = 1 \rightarrow p_{o_4}^{t+2} = 0 \quad C^t = 1$$

– Phase 1 (Case 3): $\hat{P}_{PBR}^t(0) = \frac{P_{PBR}^t(0)}{1 - p_{o_3}^t} = \frac{0.36}{0.4} = 0.9$

$$\hat{P}_{PBR}^t(1) = \frac{P_{PBR}^t(1) - \hat{P}_{PBR}^t(0) \cdot p_{o_3}^t}{1 - p_{o_3}^t} = \frac{0.58 - 0.9 \cdot 0.6}{0.4} = 0.1$$

- Phase 2 (Case 3):

$$P_{PBR}^{t+1}(0) = \hat{P}_{PBR}^t(-1) \cdot p_{o_3}^{t+1} + \hat{P}_{PBR}^t(0) \cdot (1 - p_{o_3}^{t+1}) = 0 \cdot 0.2 + 0.9 \cdot 0.8 = 0.72$$

$$P_{PBR}^{t+1}(1) = \hat{P}_{PBR}^t(0) \cdot p_{o_3}^{t+1} + \hat{P}_{PBR}^t(1) \cdot (1 - p_{o_3}^{t+1}) = 0.9 \cdot 0.2 + 0.1 \cdot 0.8 = 0.26$$

- PIR result: $P_q^t(1) = 0 \rightarrow P_q^{t+1}(1) = 0 \quad P_q^t(2) = 0.36 \rightarrow P_q^{t+1}(2) = 0.72$

- Example: $n = 4, k = 2$

$$p_{o_1}^t = 0.1 \quad p_{o_2}^t = 0 \quad p_{o_3}^t = 0.6 \rightarrow p_{o_3}^{t+1} = 0.2 \quad p_{o_4}^t = 1 \rightarrow p_{o_4}^{t+2} = 0 \quad C^t = 1$$

– Phase 1 (Case 1): $\hat{P}_{PBR}^{t+1}(0) = P_{PBR}^{t+1}(0) = 0.72$

$$\hat{P}_{PBR}^{t+1}(1) = P_{PBR}^{t+1}(1) = 0.26$$

$$C^t = 0$$

– Phase 2 (Case 2): $P_{PBR}^{t+2}(0) = \hat{P}_{PBR}^{t+1}(0) = 0.72$

$$P_{PBR}^{t+2}(1) = \hat{P}_{PBR}^{t+1}(1) = 0.26$$

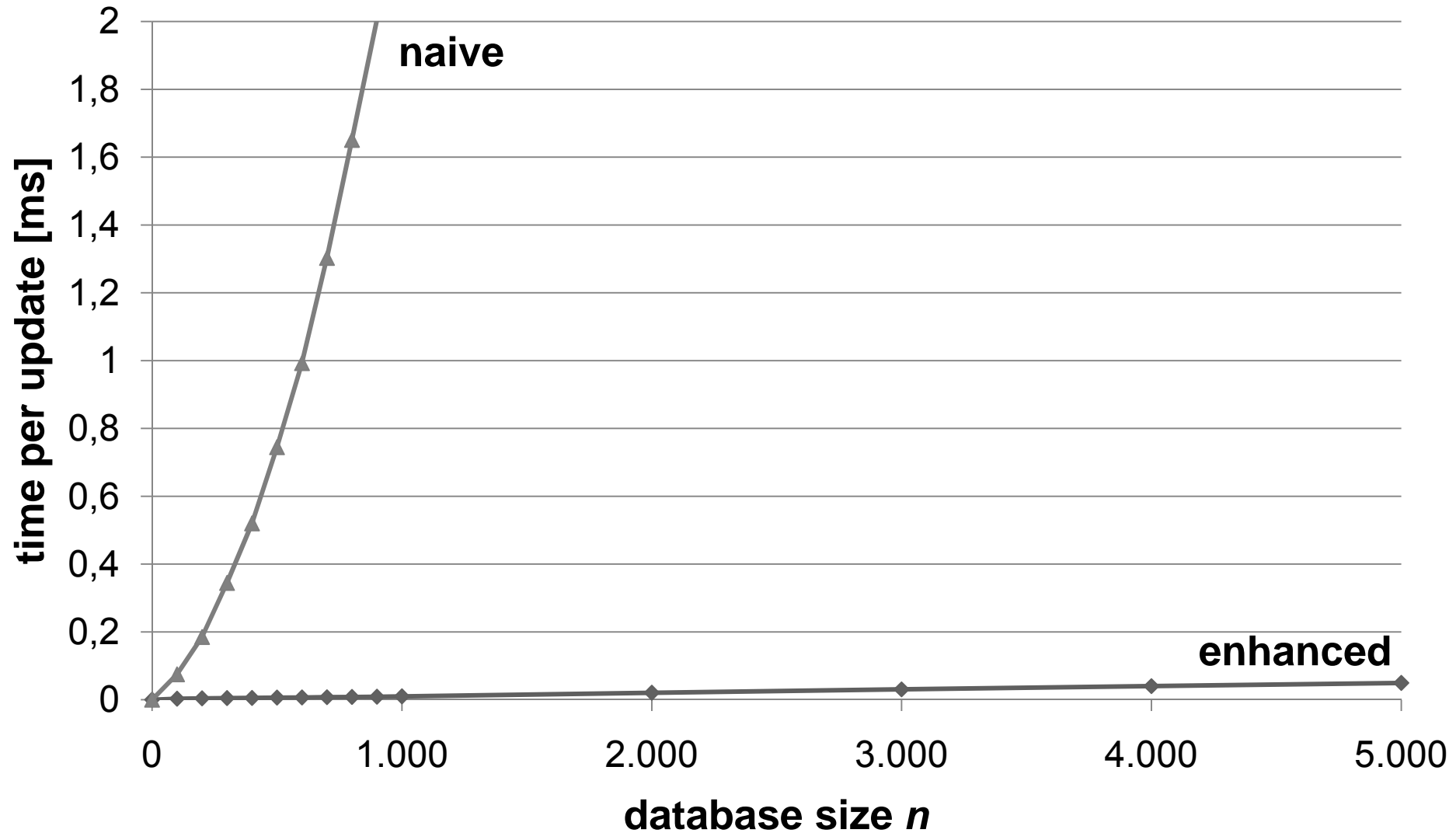
– PIR result:

$$P_q^{t+1}(1) = 0 \rightarrow P_q^{t+2}(1) = P_{PBR}^{t+2}(1-1-0) = P_{PBR}^{t+2}(0) = 0.72$$

$$P_q^{t+1}(2) = 0.72 \rightarrow P_q^{t+2}(2) = P_{PBR}^{t+2}(2-1-0) = P_{PBR}^{t+2}(1) = 0.26$$

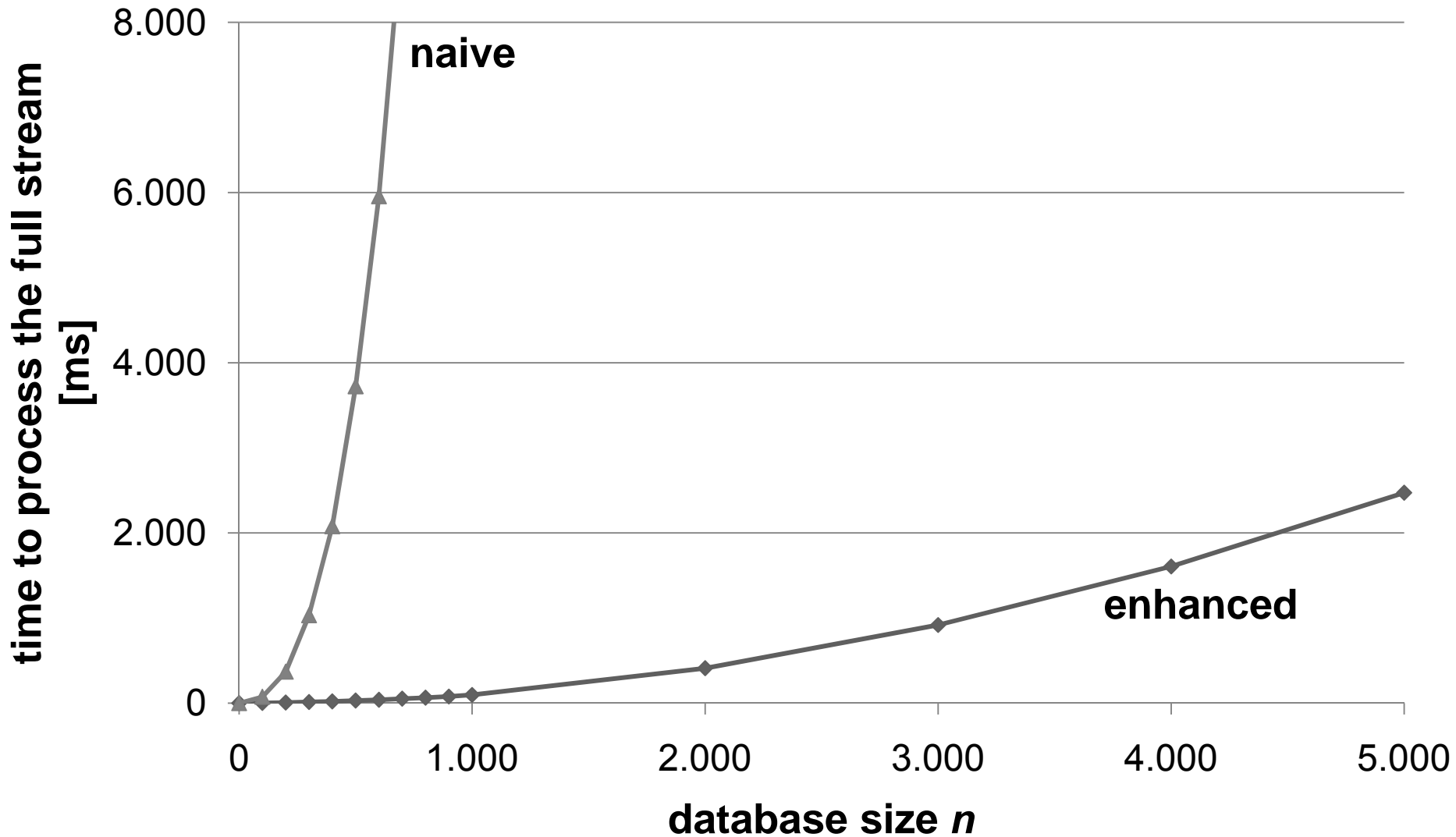
Experiments (1)

- dimensions = 2, $m = 10$, $\sigma = 5$, $k = n$, buffer = 3



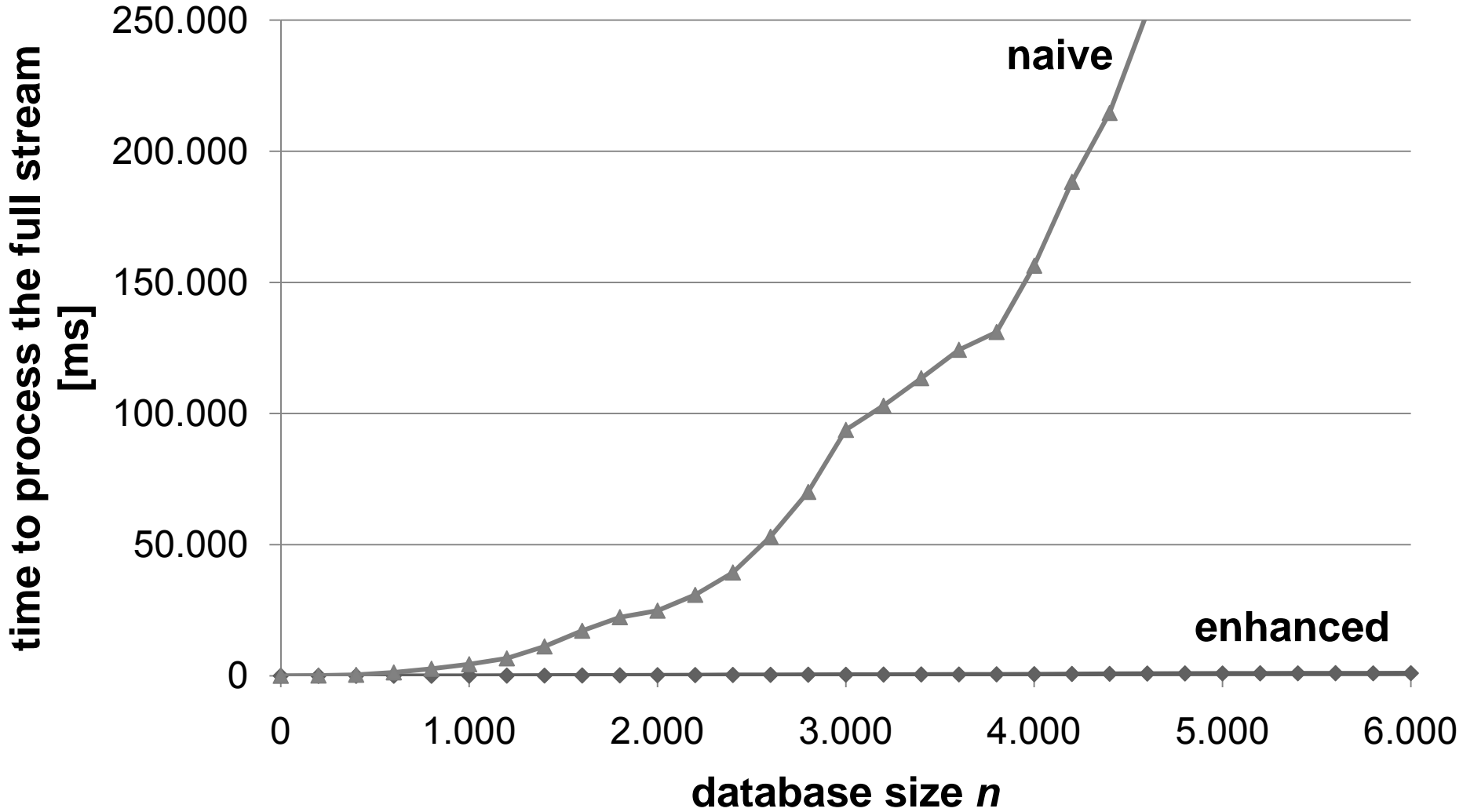
Experiments (2)

- dimensions = 2, $m = 10$, $\sigma = 5$, $k = n$, buffer = 3



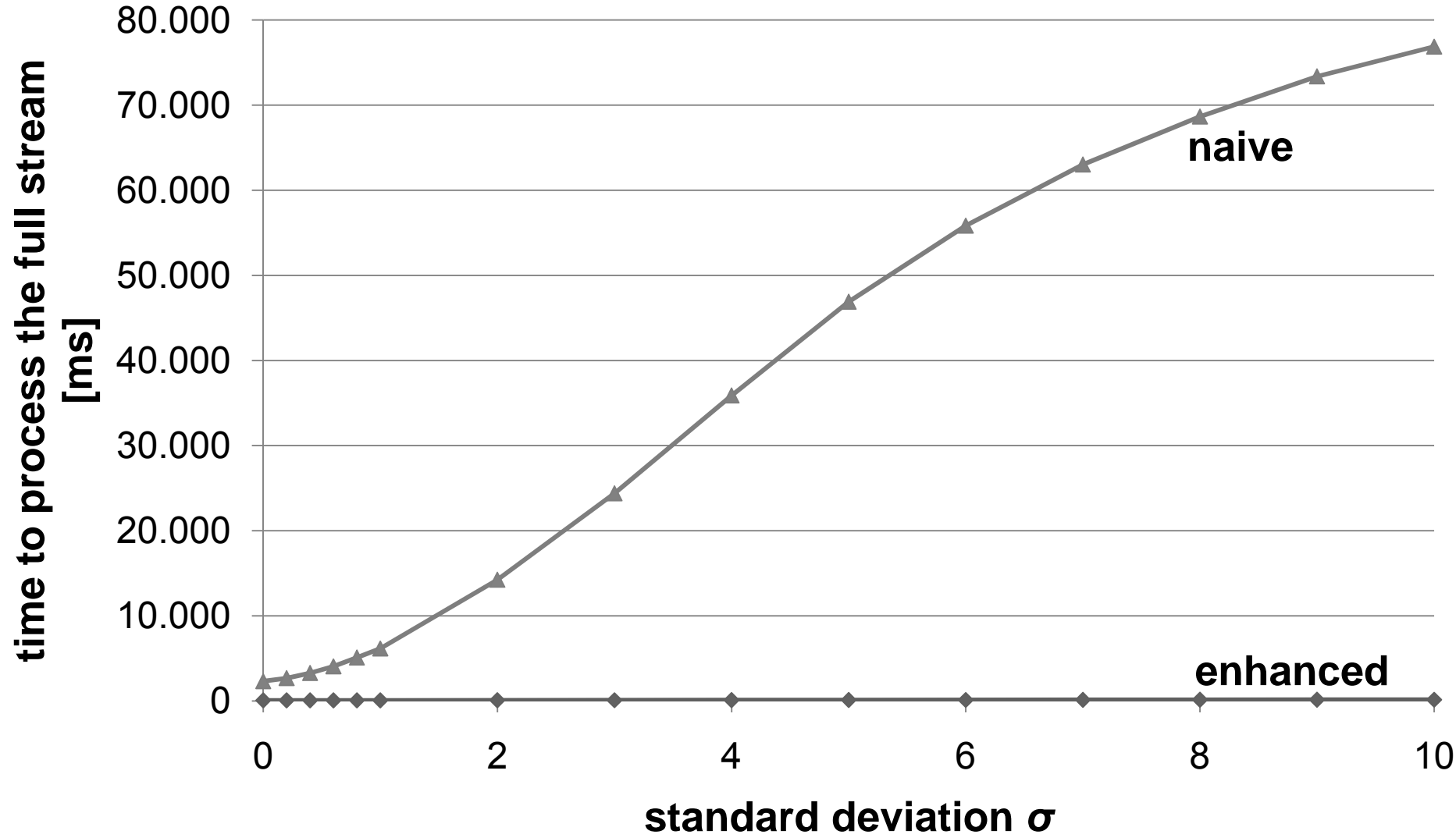
Experiments (3)

- IIP dataset, dimensions = 2, $m = 10$, $k = n$, buffer = 3



Experiments (4)

- $n = 10,000$, dimensions = 2, $m = 10$, $k = n$, buffer = 3



- Efficient solution for PIR queries on continuous data yielding update costs of $O(k)$ instead of $O(k \cdot n)$
- The framework can be adapted to other query types, e.g. the probabilistic threshold inverse ranking query
- Future work: approximate approach using lower and upper bounds for the probabilities and applying the concept of Generating Functions