Counting Triangles under Updates in Worst-Case Optimal Time

Ahmet Kara, Hung Q. Ngo, Milos Nikolic
Dan Olteanu, and Haozhe Zhang

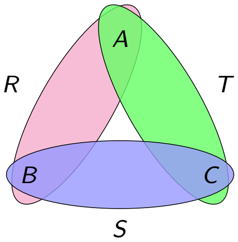
fdbresearch.github.io

ICDT 2019, Lisbon

relationalAI

Problem Setting

The triangle count Q returns the number of tuples in the join of R, S, and T.



Maintain the triangle count Q
under single-tuple updates to R, S, and T!

Data Model

- Relations are functions mapping tuples to multiplicities.

R

A	B	$\#$
a_1	b_1	2
a_2	b_1	3

S

B	C	$\#$
b_1	c_1	2
b_1	c_2	1

T

C	A	$\#$
c_1	a_1	1
c_2	a_1	3
c_2	a_2	3

Data Model

- Relations are functions mapping tuples to multiplicities.

R

A	B	#
a_1	b_1	2
a_2	b_1	3

S

B	C	#
b_1	c_1	2
b_1	c_2	1

T

C	A	#
c_1	a_1	1
c_2	a_1	3
c_2	a_2	3

$R \cdot S \cdot T$

A	B	C	#
a_1	b_1	c_1	$2 \cdot 2 \cdot 1 = 4$

Data Model

- Relations are functions mapping tuples to multiplicities.

R

A	B	$\#$
a_1	b_1	2
a_2	b_1	3

S

B	C	$\#$
b_1	c_1	2
b_1	c_2	1

T

C	A	$\#$
c_1	a_1	1
c_2	a_1	3
c_2	a_2	3

$R \cdot S \cdot T$

A	B	C	$\#$
a_1	b_1	c_1	$2 \cdot 2 \cdot 1 = 4$
a_1	b_1	c_2	$2 \cdot 1 \cdot 3 = 6$
a_2	b_1	c_2	$3 \cdot 1 \cdot 3 = 9$

Data Model

- Relations are functions mapping tuples to multiplicities.
- $Q() = \sum_{a,b,c} R(a,b) \cdot S(b,c) \cdot T(c,a)$

R

A	B	$\#$
a_1	b_1	2
a_2	b_1	3

S

B	C	$\#$
b_1	c_1	2
b_1	c_2	1

T

C	A	$\#$
c_1	a_1	1
c_2	a_1	3
c_2	a_2	3

$R \cdot S \cdot T$

A	B	C	$\#$
a_1	b_1	c_1	$2 \cdot 2 \cdot 1 = 4$
a_1	b_1	c_2	$2 \cdot 1 \cdot 3 = 6$
a_2	b_1	c_2	$3 \cdot 1 \cdot 3 = 9$

$\downarrow$

$Q()$

$\emptyset$	$\#$
$()$	$4 + 6 + 9 = 19$

Data Model

- Relations are functions mapping tuples to multiplicities.
- $Q() = \sum_{a,b,c} R(a,b) \cdot S(b,c) \cdot T(c,a)$
- A single-tuple update is a relation mapping a tuple to a non-zero value (positive for insertions, negative for deletions)

R

A	B	#
a_1	b_1	2
a_2	b_1	3

S

B	C	#
b_1	c_1	2
b_1	c_2	1

T

C	A	#
c_1	a_1	1
c_2	a_1	3
c_2	a_2	3

$R \cdot S \cdot T$

A	B	C	#
a_1	b_1	c_1	$2 \cdot 2 \cdot 1 = 4$
a_1	b_1	c_2	$2 \cdot 1 \cdot 3 = 6$
a_2	b_1	c_2	$3 \cdot 1 \cdot 3 = 9$

$\uparrow$

$\downarrow$

$\delta R = \{(a_2, b_1) \mapsto -2\}$

A	B	#
a_2	b_1	-2

$Q()$

$\emptyset$	#
( )	$4 + 6 + 9 = 19$

Data Model

- Relations are functions mapping tuples to multiplicities.
- $Q() = \sum_{a,b,c} R(a,b) \cdot S(b,c) \cdot T(c,a)$
- A single-tuple update is a relation mapping a tuple to a non-zero value (positive for insertions, negative for deletions)

R

A	B	#
a_1	b_1	2
a_2	b_1	3

S

B	C	#
b_1	c_1	2
b_1	c_2	1

T

C	A	#
c_1	a_1	1
c_2	a_1	3
c_2	a_2	3

$R \cdot S \cdot T$

A	B	C	#
a_1	b_1	c_1	$2 \cdot 2 \cdot 1 = 4$
a_1	b_1	c_2	$2 \cdot 1 \cdot 3 = 6$
a_2	b_1	c_2	$3 \cdot 1 \cdot 3 = 9$

↑

↓

$\delta R = \{(a_2, b_1) \mapsto -2\}$

A	B	#
a_2	b_1	-2

$Q()$

$\emptyset$	#
()	$4 + 6 + 9 = 19$

Data Model

- Relations are functions mapping tuples to multiplicities.
- $Q() = \sum_{a,b,c} R(a,b) \cdot S(b,c) \cdot T(c,a)$
- A single-tuple update is a relation mapping a tuple to a non-zero value (positive for insertions, negative for deletions)

R

A	B	#
a_1	b_1	2
~~a_2~~	~~b_1~~	~~3~~
a_2	b_1	1

S

B	C	#
b_1	c_1	2
b_1	c_2	1

T

C	A	#
c_1	a_1	1
c_2	a_1	3
c_2	a_2	3

$R \cdot S \cdot T$

A	B	C	#
a_1	b_1	c_1	$2 \cdot 2 \cdot 1 = 4$
a_1	b_1	c_2	$2 \cdot 1 \cdot 3 = 6$
a_2	b_1	c_2	$3 \cdot 1 \cdot 3 = 9$

$\uparrow$ $\downarrow$

$\delta R = \{(a_2, b_1) \mapsto -2\}$

A	B	#
a_2	b_1	-2

$Q()$

$\emptyset$	#
( )	$4 + 6 + 9 = 19$

Data Model

- Relations are functions mapping tuples to multiplicities.
- $Q() = \sum_{a,b,c} R(a,b) \cdot S(b,c) \cdot T(c,a)$
- A single-tuple update is a relation mapping a tuple to a non-zero value (positive for insertions, negative for deletions)

R

A	B	#
a_1	b_1	2
~~a_2~~	~~b_1~~	~~3~~
a_2	b_1	1

S

B	C	#
b_1	c_1	2
b_1	c_2	1

T

C	A	#
c_1	a_1	1
c_2	a_1	3
c_2	a_2	3

$R \cdot S \cdot T$

A	B	C	#
a_1	b_1	c_1	$2 \cdot 2 \cdot 1 = 4$
a_1	b_1	c_2	$2 \cdot 1 \cdot 3 = 6$
a_2	b_1	c_2	$3 \cdot 1 \cdot 3 = 9$

$\uparrow$ $\downarrow$

$\delta R = \{(a_2, b_1) \mapsto -2\}$

A	B	#
a_2	b_1	-2

$Q()$

$\emptyset$	#
( )	$4 + 6 + 9 = 19$

Data Model

- Relations are functions mapping tuples to multiplicities.
- $Q() = \sum_{a,b,c} R(a,b) \cdot S(b,c) \cdot T(c,a)$
- A single-tuple update is a relation mapping a tuple to a non-zero value (positive for insertions, negative for deletions)

R

A	B	#
a_1	b_1	2
~~a_2~~	~~b_1~~	~~3~~
a_2	b_1	1

S

B	C	#
b_1	c_1	2
b_1	c_2	1

T

C	A	#
c_1	a_1	1
c_2	a_1	3
c_2	a_2	3

$R \cdot S \cdot T$

A	B	C	#
a_1	b_1	c_1	$2 \cdot 2 \cdot 1 = 4$
a_1	b_1	c_2	$2 \cdot 1 \cdot 3 = 6$
~~a_2~~	~~b_1~~	~~c_2~~	~~$3 \cdot 1 \cdot 3 = 9$~~
a_2	b_1	c_2	$1 \cdot 1 \cdot 3 = 3$

$\uparrow$

$\delta R = \{(a_2, b_1) \mapsto -2\}$

A	B	#
a_2	b_1	-2

$\downarrow$

$Q()$

$\emptyset$	#
( )	$4 + 6 + 9 = 19$

Data Model

- Relations are functions mapping tuples to multiplicities.
- $Q() = \sum_{a,b,c} R(a,b) \cdot S(b,c) \cdot T(c,a)$
- A single-tuple update is a relation mapping a tuple to a non-zero value (positive for insertions, negative for deletions)

R

A	B	$\#$
a_1	b_1	2
~~a_2~~	~~b_1~~	~~3~~
a_2	b_1	1

S

B	C	$\#$
b_1	c_1	2
b_1	c_2	1

T

C	A	$\#$
c_1	a_1	1
c_2	a_1	3
c_2	a_2	3

$R \cdot S \cdot T$

A	B	C	$\#$
a_1	b_1	c_1	$2 \cdot 2 \cdot 1 = 4$
a_1	b_1	c_2	$2 \cdot 1 \cdot 3 = 6$
~~a_2~~	~~b_1~~	~~c_2~~	~~$3 \cdot 1 \cdot 3 = 9$~~
a_2	b_1	c_2	$1 \cdot 1 \cdot 3 = 3$

$\uparrow$

$\downarrow$

$\delta R = \{(a_2, b_1) \mapsto -2\}$

A	B	$\#$
a_2	b_1	-2

$Q()$

$\emptyset$	$\#$
~~$()$~~	~~$4 + 6 + 9 = 19$~~
$()$	$4 + 6 + 3 = 13$

Previous Results

Much Ado about Triangles

The Triangle Query Served as Milestone in Many Fields

- Worst-case optimal join algorithms *[Algorithmica 1997, SIGMOD R. 2013]*
- Parallel query evaluation *[Found. & Trends DB 2018]*
- Randomized approximation in static settings *[FOCS 2015]*
- Randomized approximation in data streams
  [SODA 2002, COCOON 2005, PODS 2006, PODS 2016, Theor. Comput. Sci. 2017]

Intensive Investigation of Answering Queries under Updates

- Theoretical developments *[PODS 2017, ICDT 2018]*
- Systems developments *[F. & T. DB 2012, VLDB J. 2014, SIGMOD 2017, 2018]*
- Lower bounds *[STOC 2015, ICM 2018]*

So far:

No dynamic algorithm maintaining the
exact triangle count in **worst-case optimal** time!

Naïve Maintenance

"Compute from scratch!"

$$Q() = \sum_{a,b,c} R(a, b) \cdot S(b, c) \cdot T(c, a)$$

Naïve Maintenance

"Compute from scratch!"

$$\delta R = \{(\alpha, \beta) \mapsto m\}$$

$$Q() = \sum_{a,b,c} \overbrace{R(a,b) + \delta R(a,b)} \cdot \overbrace{S(b,c)} \cdot \overbrace{T(c,a)}$$

Naïve Maintenance

"Compute from scratch!"

$$\delta R = \{(\alpha, \beta) \mapsto m\}$$

$$Q() = \sum_{a,b,c} R(a,b) + \delta R(a,b) \cdot S(b,c) \cdot T(c,a)$$

Naïve Maintenance

"Compute from scratch!"

$$\delta R = \{(\alpha, \beta) \mapsto m\}$$

$$Q() = \sum_{a,b,c} \underbrace{R(a,b) + \delta R(a,b)} \cdot S(b,c) \cdot T(c,a)$$

- N is the database size
- Update time: $\mathcal{O}(N^{1.5})$ using worst-case optimal join algorithms
  [Algorithmica 1997, SIGMOD R. 2013, ICDT 2014]
- Space: $\mathcal{O}(N)$ to store input relations

Classical Incremental View Maintenance *[Found. & Trends DB 2018]*

"Compute the difference!"

$$Q() = \sum_{a,b,c} R(a,b) \cdot S(b,c) \cdot T(c,a)$$

Classical Incremental View Maintenance *[Found. & Trends DB 2018]*

"Compute the difference!"

$$\delta R = \{(\alpha, \beta) \mapsto m\}$$

$$Q() = \sum_{a,b,c} R(a,b) \cdot S(b,c) \cdot T(c,a)$$

$$\delta Q() = \sum_{a,b,c} \delta R(a,b) \cdot S(b,c) \cdot T(c,a)$$

α β

Classical Incremental View Maintenance *[Found. & Trends DB 2018]*

"Compute the difference!"

$$\delta R = \{(\alpha, \beta) \mapsto m\}$$

$$Q() = \sum_{a,b,c} R(a, b) \cdot S(b, c) \cdot T(c, a)$$

$$\delta Q() = \sum_{a,b,c} \underset{\begin{matrix}\alpha & \beta\end{matrix}}{\delta R(a, b)} \cdot \underset{\begin{matrix} & c \\ \beta & \cdots \\ & c\end{matrix}}{S(b, c)} \cdot \underset{\begin{matrix} c & \\ \cdots & \alpha \\ c & \end{matrix}}{T(c, a)}$$

"Compute the difference!"

$$\delta R = \{(\alpha, \beta) \mapsto m\}$$

$$Q() = \sum_{a,b,c} R(a,b) \cdot S(b,c) \cdot T(c,a)$$

$$\delta Q() = \delta R(\alpha, \beta) \cdot \sum_{c} S(\beta, c) \cdot T(c, \alpha)$$

$$\delta Q() = \begin{array}{cc} \alpha & \beta \end{array} \cdot \sum_{c} \begin{array}{cc} & c \\ \beta & \cdots \\ & c \end{array} \cdot \begin{array}{cc} c & \\ \cdots & \alpha \\ c & \end{array}$$

Classical Incremental View Maintenance *[Found. & Trends DB 2018]*

"Compute the difference!"

$$\delta R = \{(\alpha, \beta) \mapsto m\}$$

$$Q() = \sum_{a,b,c} R(a,b) \cdot S(b,c) \cdot T(c,a)$$

$$\delta Q() = \delta R(\alpha, \beta) \cdot \sum_{c} \underbrace{S(\beta, c)}_{\mathcal{O}(N)} \cdot \underbrace{T(c, \alpha)}_{\mathcal{O}(N)}$$

Classical Incremental View Maintenance *[Found. & Trends DB 2018]*

"Compute the difference!"

$$\delta R = \{(\alpha, \beta) \mapsto m\}$$

$$Q() = \sum_{a,b,c} R(a, b) \cdot S(b, c) \cdot T(c, a)$$

$$\delta Q() = \delta R(\alpha, \beta) \cdot \sum_{c} S(\beta, c) \cdot T(c, \alpha)$$

$$\left.\begin{matrix} \beta & c \\ & \cdots \\ & c \end{matrix}\right\} \mathcal{O}(N) \qquad \mathcal{O}(N) \left\{\begin{matrix} c & \\ \cdots & \alpha \\ c & \end{matrix}\right.$$

$$Q() = Q() + \delta Q()$$

Classical Incremental View Maintenance *[Found. & Trends DB 2018]*

"Compute the difference!"

$$\delta R = \{(\alpha, \beta) \mapsto m\}$$

$$Q() = \sum_{a,b,c} R(a,b) \cdot S(b,c) \cdot T(c,a)$$

$$\delta Q() = \delta R(\alpha,\beta) \cdot \sum_{c} S(\beta, c) \cdot T(c, \alpha)$$

$$\begin{array}{cc|c} \alpha & \beta & \end{array} \quad \cdot \quad \sum_c \quad \underbrace{\begin{array}{cc} & c \\ \beta & \cdots \\ & c \end{array}}_{\mathcal{O}(N)} \cdot \underbrace{\begin{array}{cc} c & \\ \cdots & \alpha \\ c & \end{array}}_{\mathcal{O}(N)}$$

$$Q() = Q() + \delta Q()$$

- Update time: $\mathcal{O}(N)$ to intersect C-values from S and T
- Space: $\mathcal{O}(N)$ to store input relations

Factorized Incremental View Maintenance *[SIGMOD 2018]*

"*Compute the difference by using pre-materialized views!*"

$$Q() = \sum_{a,b,c} R(a,b) \cdot S(b,c) \cdot T(c,a)$$

Factorized Incremental View Maintenance *[SIGMOD 2018]*

"Compute the difference by using pre-materialized views!"

$$\delta R = \{(\alpha, \beta) \mapsto m\}$$

$$Q() = \sum_{a,b,c} R(a,b) \cdot \underbrace{S(b,c) \cdot T(c,a)}_{V_{ST}(b,a) = \sum_{c} S(b,c) \cdot T(c,a)}$$

Factorized Incremental View Maintenance *[SIGMOD 2018]*

"*Compute the difference by using pre-materialized views!*"

$$\delta R = \{(\alpha, \beta) \mapsto m\}$$

$$Q() \quad = \quad \sum_{a,b,c} \underbrace{\qquad}_{} R(a, b) \quad \cdot \quad V_{ST}(b, a)$$

Factorized Incremental View Maintenance *[SIGMOD 2018]*

"Compute the difference by using pre-materialized views!"

$$\delta R = \{(\alpha, \beta) \mapsto m\}$$

$$Q() = \sum_{a,b,c} R(a,b) \cdot V_{ST}(b,a)$$

$$\delta Q() = \sum_{a,b} \delta R(a,b) \cdot V_{ST}(b,a)$$

α	β

"*Compute the difference by using pre-materialized views!*"

$$\delta R = \{(\alpha, \beta) \mapsto m\}$$

$$Q() = \sum_{a,b,c} R(a,b) \cdot V_{ST}(b,a)$$

$$\delta Q() = \delta R(\alpha,\beta) \cdot V_{ST}(\beta,\alpha)$$

$$\boxed{\alpha \quad \beta} \cdot \boxed{\beta \quad \alpha}$$

Factorized Incremental View Maintenance *[SIGMOD 2018]*

"*Compute the difference by using pre-materialized views!*"

$$\delta R = \{(\alpha, \beta) \mapsto m\}$$

$$Q() = \sum_{a,b,c} R(a,b) \cdot V_{ST}(b,a)$$

$$\delta Q() = \delta R(\alpha,\beta) \cdot V_{ST}(\beta,\alpha)$$

$$Q() = Q() + \delta Q()$$

Factorized Incremental View Maintenance *[SIGMOD 2018]*

"*Compute the difference by using pre-materialized views!*"

$$\delta R = \{(\alpha, \beta) \mapsto m\}$$

$$Q() = \sum_{a,b,c} R(a,b) \cdot V_{ST}(b,a)$$

$$\delta Q() = \delta R(\alpha,\beta) \cdot V_{ST}(\beta,\alpha)$$

α β $\cdot$ β α

$$Q() = Q() + \delta Q()$$

- Time for updates to R: $\mathcal{O}(1)$ to look up in V_{ST}

Factorized Incremental View Maintenance

Maintain V_{ST} under updates

$$V_{ST}(b,a) \quad = \quad \sum_{c} \quad S(b,c) \quad \cdot \quad T(c,a)$$

Factorized Incremental View Maintenance

Maintain V_{ST} under updates

$$\delta S = \{(\beta, \gamma) \mapsto m\}$$

$$V_{ST}(b, a) = \sum_{c} S(b, c) \cdot T(c, a)$$

$$\delta V_{ST}(\beta, a) = \sum_{c} \delta S(\beta, c) \cdot T(c, a)$$

β γ

Factorized Incremental View Maintenance

Maintain V_{ST} under updates

$$\delta S = \{(\beta, \gamma) \mapsto m\}$$

$$V_{ST}(b, a) = \sum_c S(b, c) \cdot T(c, a)$$

$$\delta V_{ST}(\beta, a) = \delta S(\beta, \gamma) \cdot T(\gamma, a)$$

$\beta \quad \gamma$

$\gamma \quad a \; \cdots \; a$

Factorized Incremental View Maintenance

Maintain V_{ST} under updates

$$\delta S = \{(\beta, \gamma) \mapsto m\}$$

$$V_{ST}(b, a) \quad = \quad \sum_{c} \underset{S(b, c)}{\boxed{}} \quad \bullet \quad \underset{T(c, a)}{\boxed{}}$$

$$\delta V_{ST}(\beta, a) \quad = \quad \overset{\delta S(\beta, \gamma)}{\begin{array}{cc} \beta & \gamma \end{array}} \quad \bullet \quad \overset{T(\gamma, a)}{\left.\begin{array}{cc} & a \\ \gamma & \cdots \\ & a \end{array}\right\}} \mathcal{O}(N)$$

Factorized Incremental View Maintenance

Maintain V_{ST} under updates

$$\delta S = \{(\beta, \gamma) \mapsto m\}$$

$$V_{ST}(b, a) \quad = \quad \sum_{c} \underset{S(b,c)}{\boxed{}} \quad \bullet \quad \underset{T(c,a)}{\boxed{}}$$

$$\delta V_{ST}(\beta, a) \quad = \quad \underset{\delta S(\beta, \gamma)}{\boxed{\beta \quad \gamma}} \quad \bullet \quad \underset{T(\gamma, a)}{\left.\begin{array}{cc} & a \\ \gamma & \cdots \\ & a \end{array}\right\} \mathcal{O}(N)}$$

$$V_{ST}(\beta, a) = V_{ST}(\beta, a) + \delta V_{ST}(\beta, a)$$

Factorized Incremental View Maintenance

Maintain V_{ST} under updates

$$\delta S = \{(\beta, \gamma) \mapsto m\}$$

$$V_{ST}(b, a) = \sum_{c} S(b, c) \cdot T(c, a)$$

$$\delta V_{ST}(\beta, a) = \delta S(\beta, \gamma) \cdot T(\gamma, a)$$

$\delta S(\beta, \gamma)$: β γ

$T(\gamma, a)$: γ a … a $\Big\}\ \mathcal{O}(N)$

$$V_{ST}(\beta, a) = V_{ST}(\beta, a) + \delta V_{ST}(\beta, a)$$

- Time for updates to S and T: $\mathcal{O}(N)$ to maintain V_{ST}
- Space: $\mathcal{O}(N^2)$ to store input relations and V_{ST}

Our Contribution

Closing the Complexity Gap

Complexity bounds for the maintenance of the triangle count

Known Upper Bound

Update Time: $\mathcal{O}(N)$
Space: $\mathcal{O}(N)$

Known Lower Bound

Amortized update time: not $\mathcal{O}(N^{0.5-\gamma})$ for any $\gamma > 0$
under the Online Matrix-Vector Multiplication Conjecture
(follows from *[PODS 2017]*)

Closing the Complexity Gap

Complexity bounds for the maintenance of the triangle count

Known Upper Bound

Update Time: $\mathcal{O}(N)$
Space: $\mathcal{O}(N)$

Can the triangle count be maintained with sublinear update time?

Known Lower Bound

Amortized update time: not $\mathcal{O}(N^{0.5-\gamma})$ for any $\gamma > 0$ under the Online Matrix-Vector Multiplication Conjecture (follows from *[PODS 2017]*)

Closing the Complexity Gap

Complexity bounds for the maintenance of the triangle count

Known Upper Bound

Update Time: $\mathcal{O}(N)$
Space: $\mathcal{O}(N)$

Can the triangle count be maintained with sublinear update time?

Yes!
We propose: IVM$^{\varepsilon}$
Amortized update time:
$\mathcal{O}(N^{0.5})$
This is worst-case optimal!

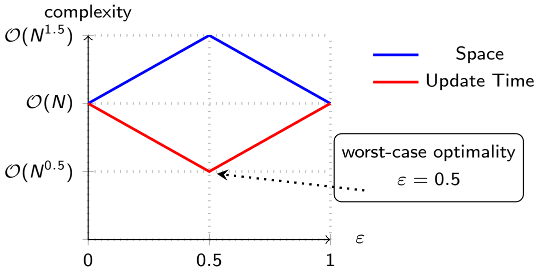
Known Lower Bound

Amortized update time: not $\mathcal{O}(N^{0.5-\gamma})$ for any $\gamma > 0$
under the Online Matrix-Vector Multiplication Conjecture
(follows from *[PODS 2017]*)

IVM$^\varepsilon$ Exhibits a Time-Space Tradeoff

Given $\varepsilon \in [0, 1]$, IVM$^\varepsilon$ maintains the triangle count with

- $\mathcal{O}(N^{\max\{\varepsilon,1-\varepsilon\}})$ amortized update time
- $\mathcal{O}(N^{1+\min\{\varepsilon,1-\varepsilon\}})$ space
- $\mathcal{O}(N^{3/2})$ preprocessing time
- $\mathcal{O}(1)$ answer time.



Known maintenance approaches are recovered by IVM$^\varepsilon$.

Main Ideas in IVM$^{\varepsilon}$

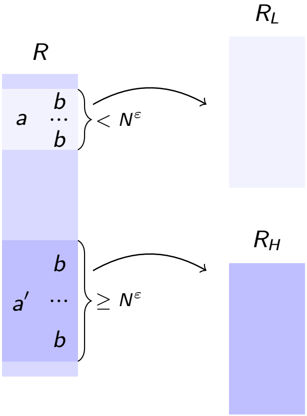
- Compute the difference like in classical IVM!
- Materialize views like in Factorized IVM!
- New ingredient: Use adaptive processing based on data skew!
  $\Longrightarrow$ Treat *heavy* values differently from *light* values!

Quick Look inside IVM$^{\varepsilon}$

Quick Look inside IVM$^{\varepsilon}$

Partition R based on A into

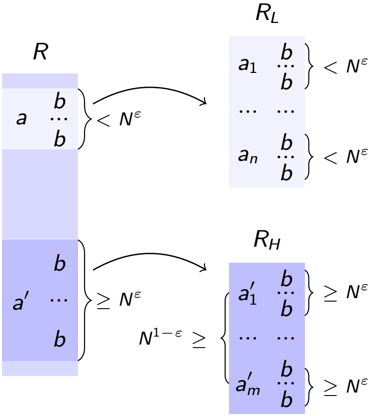
- a light part $R_L = \{t \in R \mid |\sigma_{A=t.A}| < N^{\varepsilon}\}$,
- a heavy part $R_H = R \setminus R_L$!



Quick Look inside IVM$^{\varepsilon}$

Partition R based on A into

- a light part $R_L = \{t \in R \mid |\sigma_{A=t.A}| < N^{\varepsilon}\}$,
- a heavy part $R_H = R \setminus R_L$!



Derived Bounds

from light part:
for all A-values a, $|\sigma_{A=a} R_L| < N^{\varepsilon}$

from heavy part:
$|\pi_A R_H| \leq N^{1-\varepsilon}$, since
for all A-values a, $|\sigma_{A=a} R_H| \geq N^{\varepsilon}$
and $|\pi_A R_H| \cdot N^{\varepsilon} \leq N$

Quick Look inside IVM$^{\varepsilon}$

Likewise, partition

- $S = S_L \cup S_H$ based on B, and
- $T = T_L \cup T_H$ based on C!

Q is the **sum** of skew-aware views

$$\sum_{a,b,c} R_U(a,b) \cdot S_V(b,c) \cdot T_W(c,a) \text{ with } U, V, W \in \{L, H\}.$$

Adaptive Maintenance Strategy

Given an update $\delta R_* = \{(\alpha, \beta) \mapsto m\}$, compute the difference for each of the following skew-aware views using a different strategy:

$$Q_{*LL}() = \sum_{a,b,c} R_*(a, b) \cdot S_L(b, c) \cdot T_L(c, a)$$

$$Q_{*HH}() = \sum_{a,b,c} R_*(a, b) \cdot S_H(b, c) \cdot T_H(c, a)$$

$$Q_{*LH}() = \sum_{a,b,c} R_*(a, b) \cdot S_L(b, c) \cdot T_H(c, a)$$

$$Q_{*HL}() = \sum_{a,b,c} R_*(a, b) \cdot S_H(b, c) \cdot T_L(c, a)$$

Adaptive Maintenance Strategy

Given an update $\delta R_* = \{(\alpha, \beta) \mapsto m\}$, compute the difference for each of the following skew-aware views using a different strategy:

$$\delta Q_{*LL}() = \delta R_*(\alpha, \beta) \cdot \sum_c S_L(\beta, c) \cdot T_L(c, \alpha)$$

$$\delta Q_{*HH}() = \delta R_*(\alpha, \beta) \cdot \sum_c S_H(\beta, c) \cdot T_H(c, \alpha)$$

$$\delta Q_{*LH}() = \delta R_*(\alpha, \beta) \cdot \sum_c S_L(\beta, c) \cdot T_H(c, \alpha)$$

$$\delta Q_{*HL}() = \delta R_*(\alpha, \beta) \cdot \sum_c S_H(\beta, c) \cdot T_L(c, \alpha)$$

Adaptive Maintenance Strategy

$$\delta Q_{*LL}() = \delta R_*(\alpha, \beta) \cdot \sum_c S_L(\beta, c) \cdot T_L(c, \alpha)$$

$\delta R_*(\alpha, \beta)$ $\qquad$ $S_L(\beta, c)$ $\qquad$ $T_L(c, \alpha)$

$$\delta Q_{*LL}() = \begin{array}{cc} \alpha & \beta \end{array} \bullet \sum_c \begin{array}{cc} & c \\ \beta & \cdots \\ & c \end{array} \bullet \begin{array}{cc} c & \\ \cdots & \alpha \\ c & \end{array}$$

Adaptive Maintenance Strategy

$$\delta Q_{*LL}() = \delta R_*(\alpha, \beta) \cdot \sum_c S_L(\beta, c) \cdot T_L(c, \alpha)$$

$\delta R_*(\alpha, \beta)$ $S_L(\beta, c)$ $T_L(c, \alpha)$

$$\delta Q_{*LL}() = \boxed{\alpha \quad \beta} \bullet \sum_c \left.\begin{array}{cc} & c \\ \beta & \cdots \\ & c \end{array}\right\} < N^{\varepsilon} \bullet \begin{array}{cc} c & \\ \cdots & \alpha \\ c & \end{array}$$

Adaptive Maintenance Strategy

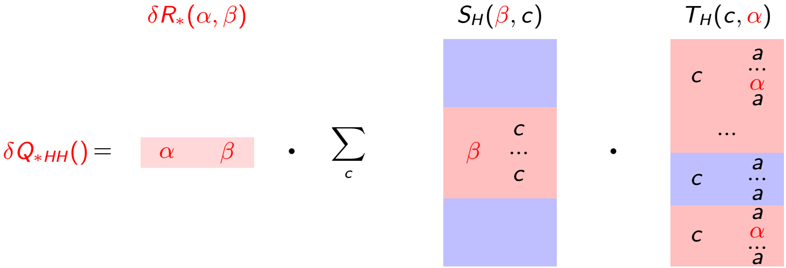
$$\delta Q_{*LL}() = \delta R_*(\alpha, \beta) \cdot \sum_c S_L(\beta, c) \cdot T_L(c, \alpha)$$

$$\delta Q_{*LL}() = \underset{\delta R_*(\alpha, \beta)}{\boxed{\alpha \quad \beta}} \bullet \sum_c \underset{S_L(\beta, c)}{\left.\begin{array}{cc} & c \\ \beta & \cdots \\ & c \end{array}\right\} < N^{\varepsilon}} \bullet \underset{T_L(c, \alpha)}{\begin{array}{cc} c & \\ \cdots & \alpha \\ c & \end{array}}$$

Update time: $\mathcal{O}(N^{\varepsilon})$ to intersect C-values from S_L and T_L

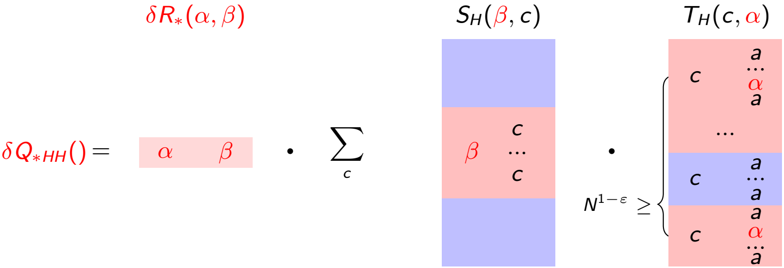
Adaptive Maintenance Strategy

$$\delta Q_{*HH}() = \delta R_*(\alpha, \beta) \cdot \sum_c S_H(\beta, c) \cdot T_H(c, \alpha)$$



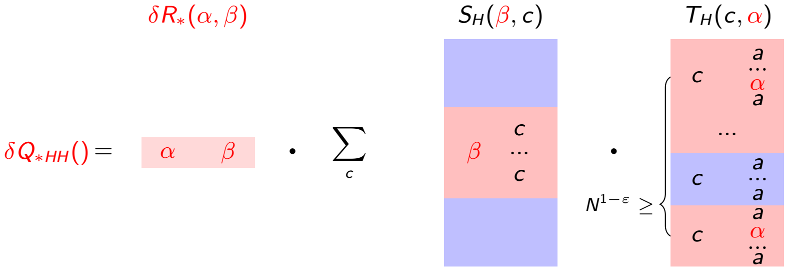
Adaptive Maintenance Strategy

$$\delta Q_{*HH}() = \delta R_*(\alpha, \beta) \cdot \sum_c S_H(\beta, c) \cdot T_H(c, \alpha)$$



Adaptive Maintenance Strategy

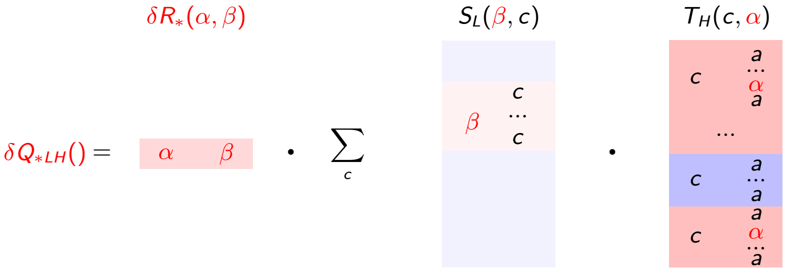
$$\delta Q_{*HH}() = \delta R_*(\alpha, \beta) \cdot \sum_c S_H(\beta, c) \cdot T_H(c, \alpha)$$



Update time: $\mathcal{O}(N^{1-\varepsilon})$ to intersect C-values from S_H and T_H

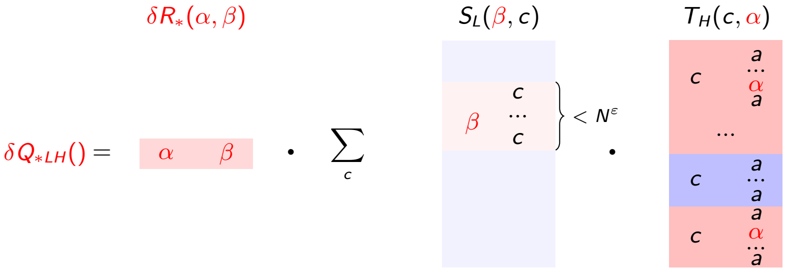
Adaptive Maintenance Strategy

$$\delta Q_{*LH}() = \delta R_*(\alpha, \beta) \cdot \sum_c S_L(\beta, c) \cdot T_H(c, \alpha)$$



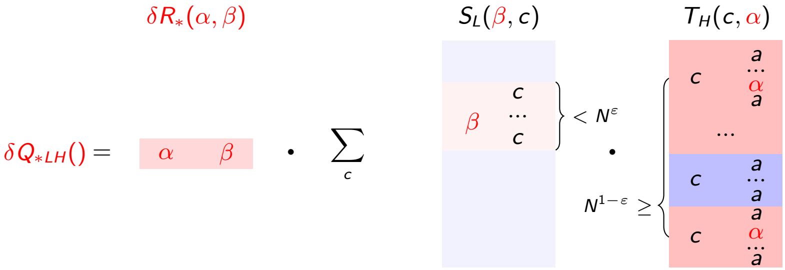
Adaptive Maintenance Strategy

$$\delta Q_{*LH}() = \delta R_*(\alpha,\beta) \cdot \sum_c S_L(\beta,c) \cdot T_H(c,\alpha)$$



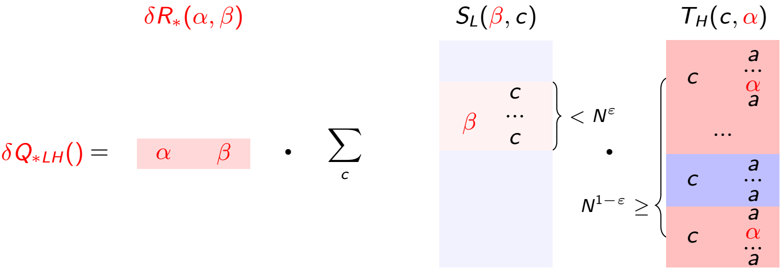
Adaptive Maintenance Strategy

$$\delta Q_{*LH}() = \delta R_*(\alpha, \beta) \cdot \sum_c S_L(\beta, c) \cdot T_H(c, \alpha)$$



Adaptive Maintenance Strategy

$$\delta Q_{*LH}() = \delta R_*(\alpha, \beta) \cdot \sum_c S_L(\beta, c) \cdot T_H(c, \alpha)$$



Update time: $\mathcal{O}(N^{\min\{\varepsilon, 1-\varepsilon\}})$ to intersect C-values from S_L and T_H

Adaptive Maintenance Strategy

$$\delta Q_{*HL}() = \delta R_*(\alpha, \beta) \cdot \sum_c S_H(\beta, c) \cdot T_L(c, \alpha)$$

$\delta R_*(a, b)$ $\quad$ $S_H(b, c)$ $\quad$ $T_L(c, a)$

$$\delta Q_{*HL}() = \sum_{a,b} \boxed{\alpha \quad \beta} \quad \bullet \quad \boxed{} \quad \bullet \quad \boxed{}$$

Adaptive Maintenance Strategy

$$\delta Q_{*HL}() = \delta R_*(\alpha, \beta) \cdot \sum_c S_H(\beta, c) \cdot T_L(c, \alpha)$$

$$\delta Q_{*HL}() = \sum_{a,b} \overset{\delta R_*(a, b)}{\boxed{\alpha \quad \beta}} \cdot \underbrace{\overset{S_H(b, c)}{\square} \cdot \overset{T_L(c, a)}{\square}}_{V_{ST}(b, a) = \sum_c S_H(b, c) \cdot T_L(c, a)}$$

Adaptive Maintenance Strategy

$$\delta Q_{*HL}() = \delta R_*(\alpha, \beta) \cdot \sum_c S_H(\beta, c) \cdot T_L(c, \alpha)$$

$\delta R_*(a, b)$ $V_{ST}(b, a)$

$$\delta Q_{*HL}() = \sum_{a,b} \boxed{\alpha \quad \beta} \bullet$$

Adaptive Maintenance Strategy

$$\delta Q_{*HL}() = \delta R_*(\alpha, \beta) \cdot \sum_c S_H(\beta, c) \cdot T_L(c, \alpha)$$

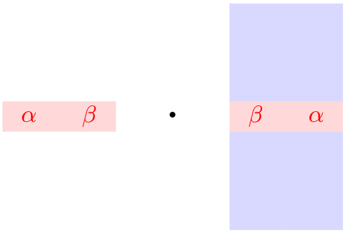
$$\delta R_*(\alpha, \beta) \qquad V_{ST}(\beta, \alpha)$$

$$\delta Q_{*HL}() = \begin{bmatrix} \alpha & \beta \end{bmatrix} \bullet \begin{bmatrix} \vdots & \vdots \\ \beta & \alpha \\ \vdots & \vdots \end{bmatrix}$$

Adaptive Maintenance Strategy

$$\delta Q_{*HL}() = \delta R_*(\alpha, \beta) \cdot \sum_c S_H(\beta, c) \cdot T_L(c, \alpha)$$

$\delta Q_{*HL}() =$ $\delta R_*(\alpha, \beta)$ $\bullet$ $V_{ST}(\beta, \alpha)$



Update time: $\mathcal{O}(1)$ to look up in V_{ST}

Summary of Adaptive Maintenance Strategies

Given an update $\delta R_* = \{(\alpha, \beta) \mapsto m\}$, compute the difference for each skew-aware view using a different strategy:

Skew-aware View	Evaluation from left to right	Time
$\sum_{a,b,c} R_*(a,b) \cdot S_L(b,c) \cdot T_L(c,a)$	$\delta R_*(\alpha,\beta) \cdot \sum_c S_L(\beta,c) \cdot T_L(c,\alpha)$	$\mathcal{O}(N^{\varepsilon})$
$\sum_{a,b,c} R_*(a,b) \cdot S_H(b,c) \cdot T_H(c,a)$	$\delta R_*(\alpha,\beta) \cdot \sum_c T_H(c,\alpha) \cdot S_H(\beta,c)$	$\mathcal{O}(N^{1-\varepsilon})$
$\sum_{a,b,c} R_*(a,b) \cdot S_L(b,c) \cdot T_H(c,a)$	$\delta R_*(\alpha,\beta) \cdot \sum_c S_L(\beta,c) \cdot T_H(c,\alpha)$	$\mathcal{O}(N^{\varepsilon})$
	or	
	$\delta R_*(\alpha,\beta) \cdot \sum_c T_H(c,\alpha) \cdot S_L(\beta,c)$	$\mathcal{O}(N^{1-\varepsilon})$
$\sum_{a,b,c} R_*(a,b) \cdot S_H(b,c) \cdot T_L(c,a)$	$\delta R_*(\alpha,\beta) \cdot V_{ST}(\beta,\alpha)$	$\mathcal{O}(1)$

Overall update time: $\mathcal{O}(N^{\max(\varepsilon, 1-\varepsilon)})$

Materialized Auxiliary Views

$$V_{RS}(a, c) = \sum_b R_H(a, b) \cdot S_L(b, c)$$

$$V_{ST}(b, a) = \sum_c S_H(b, c) \cdot T_L(c, a)$$

$$V_{TR}(a, c) = \sum_a T_H(c, a) \cdot R_L(a, b)$$

Materialized Auxiliary Views

Maintain $V_{ST}(b, a) = \sum_{c} S_H(b, c) \cdot T_L(c, a)$ under an update

$$\delta S_H = \{(\beta, \gamma) \mapsto m\}$$

$$\delta S_H(\beta, c) \qquad\qquad T_L(c, a)$$

$$\delta V_{ST}(\beta, a) = \sum_{c} \boxed{\beta \quad \gamma} \cdot \boxed{}$$

Materialized Auxiliary Views

Maintain $V_{ST}(b,a) = \sum_c S_H(b,c) \cdot T_L(c,a)$ under an update

$$\delta S_H = \{(\beta,\gamma) \mapsto m\}$$

$$\delta V_{ST}(\beta, a) = \underset{\delta S_H(\beta,\gamma)}{\begin{bmatrix} \beta & \gamma \end{bmatrix}} \bullet \underset{}{\overset{T_L(\gamma, a)}{\begin{bmatrix} \gamma & \begin{matrix} a \\ \cdots \\ a \end{matrix} \end{bmatrix}}}$$

Materialized Auxiliary Views

Maintain $V_{ST}(b, a) = \sum_{c} S_H(b, c) \cdot T_L(c, a)$ under an update

$$\delta S_H = \{(\beta, \gamma) \mapsto m\}$$

$$\delta V_{ST}(\beta, a) \quad = \quad \underbrace{\boxed{\beta \quad \gamma}}_{\delta S_H(\beta, \gamma)} \quad \bullet \quad \underbrace{\left.\begin{matrix} & a \\ \gamma & \cdots \\ & a \end{matrix}\right\} < N^{\varepsilon}}_{T_L(\gamma, a)}$$

Materialized Auxiliary Views

Maintain $V_{ST}(b,a) = \sum_{c} S_H(b,c) \cdot T_L(c,a)$ under an update

$$\delta S_H = \{(\beta,\gamma) \mapsto m\}$$

$$\delta V_{ST}(\beta,a) \quad = \quad \begin{array}{c} \delta S_H(\beta,\gamma) \\ \boxed{\beta \quad \gamma} \end{array} \quad \bullet \quad \begin{array}{c} T_L(\gamma,a) \\ \left.\begin{array}{cc} & a \\ \gamma & \cdots \\ & a \end{array}\right\} < N^{\varepsilon} \end{array}$$

Update time: $\mathcal{O}(N^{\varepsilon})$ to iterate over A-values paired with γ from T_L

Materialized Auxiliary Views

Maintain $V_{ST}(b, a) = \sum_c S_H(b, c) \cdot T_L(c, a)$ under an update

$$\delta T_L = \{(\gamma, \alpha) \mapsto m\}$$

$\delta T_L(c, \alpha)$ $S_H(b, c)$

$$\delta V_{ST}(b, \alpha) = \sum_c \boxed{\gamma \quad \alpha} \bullet S_H(b, c)$$

Materialized Auxiliary Views

Maintain $V_{ST}(b, a) = \sum_{c} S_H(b, c) \cdot T_L(c, a)$ under an update

$$\delta T_L = \{(\gamma, \alpha) \mapsto m\}$$

$$\delta V_{ST}(b, \alpha) \quad = \quad \begin{array}{c} \delta T_L(\gamma, \alpha) \\ \begin{array}{|cc|} \hline \gamma & \alpha \\ \hline \end{array} \end{array} \quad \bullet \quad \begin{array}{c} S_H(b, \gamma) \\ \begin{array}{|cc|} \hline & c \\ b & \vdots \\ & \gamma \\ & c \\ \multicolumn{2}{|c|}{\dots} \\ \hline & c \\ b & \vdots \\ & c \\ \hline & c \\ b & \gamma \\ & \vdots \\ & c \\ \hline \end{array} \end{array}$$

Materialized Auxiliary Views

Maintain $V_{ST}(b,a) = \sum_{c} S_H(b,c) \cdot T_L(c,a)$ under an update

$$\delta T_L = \{(\gamma,\alpha) \mapsto m\}$$

$$\begin{array}{cccc} & \delta T_L(\gamma,\alpha) & & S_H(b,\gamma) \\ \delta V_{ST}(b,\alpha) = & \begin{array}{cc} \gamma & \alpha \end{array} & \bullet \quad N^{1-\varepsilon} \geq & \left\{ \begin{array}{cc} b & \begin{array}{c} c \\ \vdots \\ \gamma \\ c \end{array} \\ & \dots \\ b & \begin{array}{c} c \\ \vdots \\ c \end{array} \\ b & \begin{array}{c} c \\ \gamma \\ \vdots \\ c \end{array} \end{array} \right. \end{array}$$

Materialized Auxiliary Views

Maintain $V_{ST}(b,a) = \sum_{c} S_H(b,c) \cdot T_L(c,a)$ under an update

$$\delta T_L = \{(\gamma,\alpha) \mapsto m\}$$

$$\delta V_{ST}(b,\alpha) \quad = \quad \begin{array}{c} \delta T_L(\gamma,\alpha) \\ \boxed{\gamma \quad \alpha} \end{array} \quad \bullet \quad \begin{array}{c} S_H(b,\gamma) \\ N^{1-\varepsilon} \geq \left\{ \begin{array}{cc} & c \\ b & \vdots \\ & \gamma \\ & c \\ & \dots \\ & c \\ b & \vdots \\ & c \\ & c \\ b & \gamma \\ & \vdots \\ & c \end{array} \right. \end{array}$$

Update time: $\mathcal{O}(N^{1-\varepsilon})$ to iterate over B-values paired with γ from S_H

Materialized Auxiliary Views

$$V_{RS}(a,c) = \sum_b R_H(a,b) \cdot S_L(b,c)$$

$$V_{ST}(b,a) = \sum_c S_H(b,c) \cdot T_L(c,a)$$

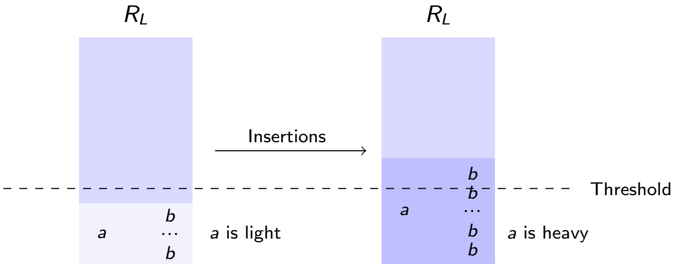
$$V_{TR}(a,c) = \sum_a T_H(c,a) \cdot R_L(a,b)$$

Maintenance Complexity

- Time: $\mathcal{O}(N^{\max\{\varepsilon, 1-\varepsilon\}})$
- Space: $\mathcal{O}(N^{1+\min\{\varepsilon, 1-\varepsilon\}})$

Rebalancing Partitions

Updates can change the frequencies of values in the relation parts!

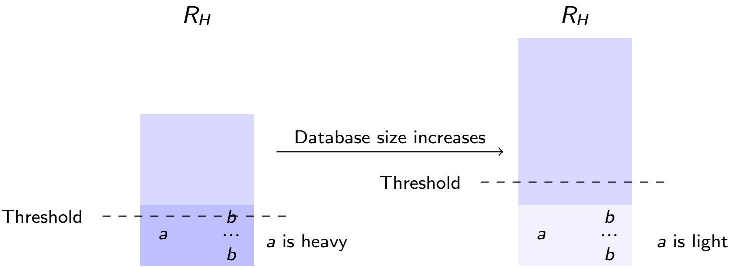


Minor Rebalancing

- Transfer $\mathcal{O}(N^{\varepsilon})$ tuples from one to the other part of the same relation!
- Time complexity: $\mathcal{O}(N^{\varepsilon+\max\{\varepsilon,1-\varepsilon\}})$

Rebalancing Partitions

Updates can change the heavy-light threshold!



Major Rebalancing

- Recompute partitions and views from scratch!
- Time complexity: $\mathcal{O}(N^{1+\min\{\varepsilon,1-\varepsilon\}})$

Amortization of Rebalancing Times

- Both forms of rebalancing require superlinear time.

Amortization of Rebalancing Times

- Both forms of rebalancing require superlinear time.
- The rebalancing times amortize over sequences of updates.
  - Amortized minor rebalancing time: $\mathcal{O}(N^{\max\{\varepsilon, 1-\varepsilon\}})$
  - Amortized major rebalancing time: $\mathcal{O}(N^{\min\{\varepsilon, 1-\varepsilon\}})$

$$\ldots \quad \mathit{update} \quad \overset{\mathcal{O}(N^{\varepsilon+\max\{\varepsilon,1-\varepsilon\}})}{\color{red}{\mathit{minor}}} \quad \underbrace{\mathit{update} \ \ldots \ \mathit{update}}_{\Omega(N^{\varepsilon})} \quad \overset{\mathcal{O}(N^{\varepsilon+\max\{\varepsilon,1-\varepsilon\}})}{\color{red}{\mathit{minor}}} \quad \mathit{update} \ \ldots$$

$$\ldots \quad \mathit{update} \quad \overset{\mathcal{O}(N^{1+\min\{\varepsilon,1-\varepsilon\}})}{\color{red}{\mathit{major}}} \quad \underbrace{\mathit{update} \ \ldots \ \mathit{update}}_{\Omega(N)} \quad \overset{\mathcal{O}(N^{1+\min\{\varepsilon,1-\varepsilon\}})}{\color{red}{\mathit{major}}} \quad \mathit{update} \ \ldots$$

Amortization of Rebalancing Times

- Both forms of rebalancing require superlinear time.
- The rebalancing times amortize over sequences of updates.
  - Amortized minor rebalancing time: $\mathcal{O}(N^{\max\{\varepsilon,1-\varepsilon\}})$
  - Amortized major rebalancing time: $\mathcal{O}(N^{\min\{\varepsilon,1-\varepsilon\}})$
- Overall amortized rebalancing time: $\mathcal{O}(N^{\max\{\varepsilon,1-\varepsilon\}})$

$$\ldots \quad \mathit{update} \quad \overset{\mathcal{O}(N^{\varepsilon+\max\{\varepsilon,1-\varepsilon\}})}{\color{red}{\mathit{minor}}} \quad \underbrace{\mathit{update} \ \ldots \ \mathit{update}}_{\Omega(N^{\varepsilon})} \quad \overset{\mathcal{O}(N^{\varepsilon+\max\{\varepsilon,1-\varepsilon\}})}{\color{red}{\mathit{minor}}} \quad \mathit{update} \quad \ldots$$

$$\ldots \quad \mathit{update} \quad \overset{\mathcal{O}(N^{1+\min\{\varepsilon,1-\varepsilon\}})}{\color{red}{\mathit{major}}} \quad \underbrace{\mathit{update} \ \ldots \ \mathit{update}}_{\Omega(N)} \quad \overset{\mathcal{O}(N^{1+\min\{\varepsilon,1-\varepsilon\}})}{\color{red}{\mathit{major}}} \quad \mathit{update} \quad \ldots$$

Quo Vadis IVM$^{\varepsilon}$?

Quo Vadis IVM$^\varepsilon$?

Generalization of IVM$^\varepsilon$

- IVM$^\varepsilon$ variants admit sublinear update time for counting versions of
  - Loomis-Whitney
  - Queries built from 3 relations
  - 4-cycle
  - 4-path
- IVM$^\varepsilon$ variants admit sublinear update time and constant-delay enumeration for the full triangle query.

Ongoing Work

- Characterization of the class of conjunctive count queries that admit sublinear maintenance time
- Implementation of IVM$^\varepsilon$ on top of DBToaster

References I

[Algorithmica 1997] Noga Alon, Raphael Yuster, and Uri Zwick. *Finding and counting given length cycles.* In Algorithmica 1997

[SODA 2002] Ziv Bar-Yossef, Ravi Kumar, and D Sivakumar. *Reductions in Streaming Algorithms, with an Application to Counting Triangles in Graphs.* In SODA 2002

[COCOON 2005] Hossein Jowhari and Mohammad Ghodsi. *New Streaming Algorithms for Counting Triangles in Graphs.* In COCOON 2005

[PODS 2006] Luciana S Buriol, Gereon Frahling, Stefano Leonardi, Alberto Marchetti-Spaccamela, and Christian Sohler. *Counting Triangles in Data Streams.* In PODS 2006

[Found. & Trends DB 2012] Rada Chirkova and Jun Yang. *Materialized Views.* Found. & Trends DB 2012

[SIGMOD R. 2013] Hung Q Ngo, Christopher Ré, and Atri Rudra. *Skew strikes back: new developments in the theory of join algorithms.* In SIGMOD R. 2013

References II

[ICDT 2014] Todd L. Veldhuizen. *Leapfrog Triejoin: A Simple, Worst-Case Optimal Join Algorithm.* In ICDT 2014

[VLDB J. 2014] Christoph Koch, Yanif Ahmad, Oliver Kennedy, Milos Nikolic, Andres Nötzli, Daniel Lupei, and Amir Shaikhha. *DBToaster: Higher-Order Delta Processing for Dynamic, Frequently Fresh Views.* In VLDB J. 2014

[FOCS 2015] Talya Eden, Amit Levi, Dana Ron, and C. Seshadhri. *Approximately Counting Triangles in Sublinear Time.* In FOCS 2015

[STOC 2015] Monika Henzinger, Sebastian Krinninger, Danupon Nanongkai, and Thatchaphol Saranurak. *Unifying and Strengthening Hardness for Dynamic Problems via the Online Matrix-Vector Multiplication Conjecture.* In STOC 2015

[PODS 2016] Andrew McGregor, Sofya Vorotnikova, and Hoa T Vu. *Better Algorithms for Counting Triangles in Data Streams.* In PODS 2016

[PODS 2017] Christoph Berkholz, Jens Keppeler, and Nicole Schweikardt. *Answering Conjunctive Queries Under Updates.* In PODS 2017

References III

[SIGMOD 2017] Muhammad Idris, Martín Ugarte, and Stijn Vansummeren. *The Dynamic Yannakakis Algorithm: Compact and Efficient Query Processing Under Updates.* In SIGMOD 2017

[Theor. Comput. Sci. 2017] Graham Cormode and Hossein Jowhari. *A Second Look at Counting Triangles in Graph Streams (Corrected).* Theor. Comput. Sci. 2017

[Found. & Trends DB 2018] Paraschos Koutris, Semih Salihoglu, and Dan Suciu. *Algorithmic Aspects of Parallel Data Processing.* In Found. & Trends DB 2018

[ICDT 2018] Christoph Berkholz, Jens Keppeler, and Nicole Schweikardt. *Answering UCQs Under Updates and in the Presence of Integrity Constraints.* In ICDT 2018

[ICM 2018] Virginia Vassilevska Williams. *On Some Fine-Grained Questions in Algorithms and Complexity.* In ICM 2018

[SIGMOD 2018] Nikolic, Milos, and Dan Olteanu. *Incremental view maintenance with triple lock factorization benefits.* In SIGMOD 2018