Counting Triangles under Updates in Worst-Case Optimal Time

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

fdbresearch.github.io

Highlights 2018, Berlin

European Research Council

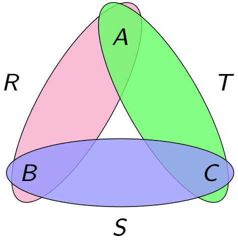
FONDATION WIENER-ANSPACH

UNIVERSITY OF OXFORD

RelationalAI

Problem Setting

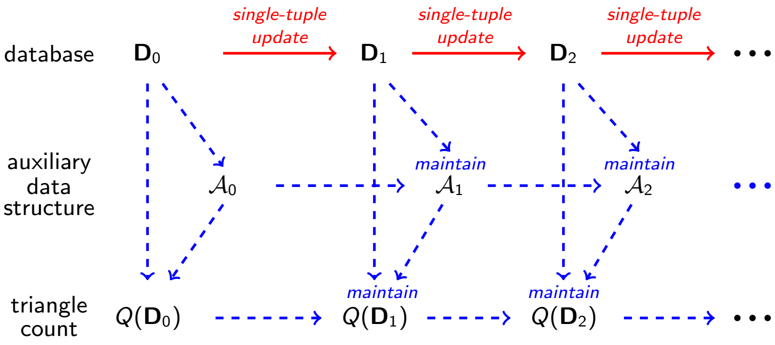
Maintain the triangle count Q
under single-tuple updates to R, S, and T!



Q counts the number of tuples in the join of R, S, and T.

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

The Maintenance Problem



Given a current database $\mathbf{D}$ and a single-tuple update,
what are the time and space complexities for maintaining $Q(\mathbf{D})$?

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!*"

$$\delta R = \{(a', b') \mapsto m\}$$

$$\sum_{a,b,c} \Big[\underbrace{R(a,b) + \delta R(a,b)}_{newR}\Big] \cdot S(b,c) \cdot T(c,a)$$

$$=$$

$$\sum_{a,b,c} \; newR(a,b) \cdot S(b,c) \cdot T(c,a)$$

Maintenance Complexity

- Time: $\mathcal{O}(|\mathbf{D}|^{1.5})$ using worst-case optimal join algorithms
- Space: $\mathcal{O}(|\mathbf{D}|)$ to store input relations

Classical Incremental View Maintenance (IVM)

"*Compute the difference!*"

$$\delta R = \{(a', b') \mapsto m\}$$

$$\sum_{a,b,c} \left[R(a,b) + \delta R(a,b)\right] \cdot S(b,c) \cdot T(c,a)$$
$$=$$
$$\sum_{a,b,c} R(a,b) \cdot S(b,c) \cdot T(c,a)$$
$$+$$
$$\delta R(a',b') \cdot \sum_{c} S(b',c) \cdot T(c,a')$$

Maintenance Complexity

- Time: $\mathcal{O}(|\mathbf{D}|)$ to intersect C-values from S and T
- Space: $\mathcal{O}(|\mathbf{D}|)$ to store input relations

Factorized Incremental View Maintenance (F-IVM)

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

$$\delta R = \{(a', b') \mapsto m\}$$

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

$$\sum_{a,b,c} \left[R(a, b) + \delta R(a, b)\right] \cdot S(b, c) \cdot T(c, a)$$
$$=$$
$$\sum_{a,b,c} R(a, b) \cdot S(b, c) \cdot T(c, a)$$
$$+$$
$$\delta R(a', b') \cdot V_{ST}(b', a')$$

Maintenance Complexity

- Time for updates to R: $\mathcal{O}(1)$ to look up in V_{ST}
- Time for updates to S and T: $\mathcal{O}(|\mathbf{D}|)$ to maintain V_{ST}
- Space: $\mathcal{O}(|\mathbf{D}|^2)$ to store input relations and V_{ST}

Closing the Complexity Gap

Complexity bounds for the maintenance of the triangle count

Known Upper Bound

Maintenance Time: $\mathcal{O}(|\mathbf{D}|)$

Space: $\mathcal{O}(|\mathbf{D}|)$

Known Lower Bound

Amortized maintenance time: not $\mathcal{O}(|\mathbf{D}|^{0.5-\gamma})$ for any $\gamma > 0$

(under reasonable complexity theoretic assumptions)

Closing the Complexity Gap

Complexity bounds for the maintenance of the triangle count

Known Upper Bound

Maintenance Time: $\mathcal{O}(|\mathbf{D}|)$
Space: $\mathcal{O}(|\mathbf{D}|)$

Can the triangle count be maintained in sublinear time?

Known Lower Bound

Amortized maintenance time: not $\mathcal{O}(|\mathbf{D}|^{0.5-\gamma})$ for any $\gamma > 0$
(under reasonable complexity theoretic assumptions)

Closing the Complexity Gap

Complexity bounds for the maintenance of the triangle count

Known Upper Bound

Maintenance Time: $\mathcal{O}(|\mathbf{D}|)$
Space: $\mathcal{O}(|\mathbf{D}|)$

Can the triangle count be maintained in sublinear time?

Yes!

We propose: IVM$^{\varepsilon}$

Amortized maintenance time:

$\mathcal{O}(|\mathbf{D}|^{0.5})$

This is worst-case optimal!

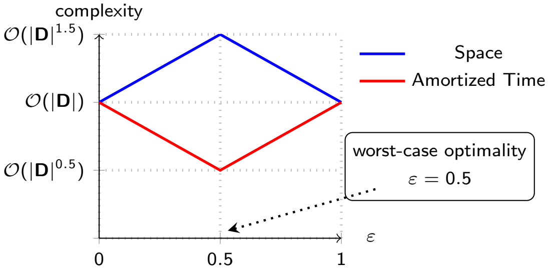
Known Lower Bound

Amortized maintenance time: not $\mathcal{O}(|\mathbf{D}|^{0.5-\gamma})$ for any $\gamma > 0$
(under reasonable complexity theoretic assumptions)

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

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

- $\mathcal{O}(|\mathbf{D}|^{\max\{\varepsilon, 1-\varepsilon\}})$ amortized time and
- $\mathcal{O}(|\mathbf{D}|^{1+\min\{\varepsilon, 1-\varepsilon\}})$ space.



- 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!

Quo Vadis IVM$^\varepsilon$?

Generalization of IVM$^\varepsilon$

- IVM$^\varepsilon$ variants obtain sublinear maintenance time for counting versions of Loomis-Whitney, 4-cycle, and 4-path.

Ongoing Work

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

Quo Vadis IVM$^\varepsilon$?

Generalization of IVM$^\varepsilon$

- IVM$^\varepsilon$ variants obtain sublinear maintenance time for counting versions of Loomis-Whitney, 4-cycle, and 4-path.

Ongoing Work

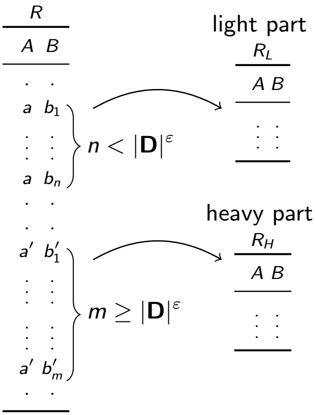
- Characterization of the class of conjunctive count queries that admit sublinear maintenance time
- Implementation of IVM$^\varepsilon$ on top of DBToaster

For details, see arxiv.org/abs/1804.02780

Quick Look inside IVM$^{\varepsilon}$

Partition R into

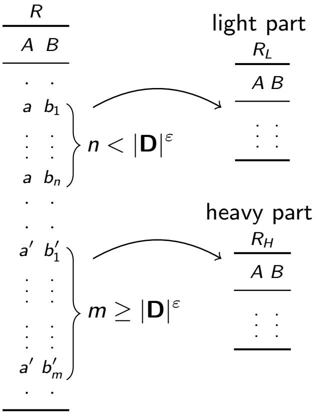
- a light part $R_L = \{t \in R \mid |\sigma_{A=t.A}| < |\mathbf{D}|^{\varepsilon}\}$,
- a heavy part $R_H = R \setminus R_L$!



Quick Look inside IVM$^{\varepsilon}$

Partition R into

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



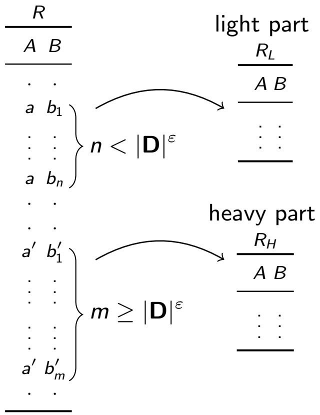
Derived Bounds

- for all A-values a: $|\sigma_{A=a} R_L| < |\mathbf{D}|^{\varepsilon}$
- $|\pi_A R_H| \leq |\mathbf{D}|^{1-\varepsilon}$

Quick Look inside IVM$^{\varepsilon}$

Partition R into

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



Derived Bounds

- for all A-values a:
  $|\sigma_{A=a} R_L| < |\mathbf{D}|^{\varepsilon}$
- $|\pi_A R_H| \leq |\mathbf{D}|^{1-\varepsilon}$

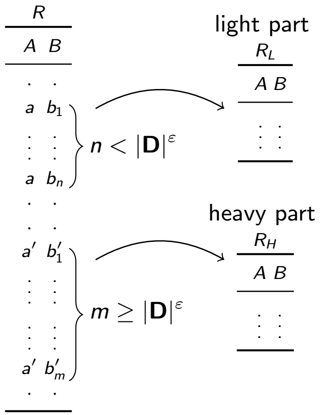
Likewise, partition

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

Quick Look inside IVM$^{\varepsilon}$

Partition R into

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



Derived Bounds

- for all A-values a: $|\sigma_{A=a} R_L| < |\mathbf{D}|^{\varepsilon}$
- $|\pi_A R_H| \leq |\mathbf{D}|^{1-\varepsilon}$

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

$$R_U(a, b) \cdot S_V(b, c) \cdot T_W(c, a)$$

with $U, V, W \in \{L, H\}$.

Adaptive Maintenance Strategy

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

Skew-aware View	Evaluation from left to right	Time
$\sum_{a,b,c} R_*(a,b) \cdot S_L(b,c) \cdot T_L(c,a)$	$\delta R_*(a',b') \cdot \sum_{c} S_L(b',c) \cdot T_L(c,a')$	$\mathcal{O}(\lvert\mathbf{D}\rvert^{\varepsilon})$

Adaptive Maintenance Strategy

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

Skew-aware View	Evaluation from left to right	Time
$\sum_{a,b,c} R_*(a,b) \cdot S_L(b,c) \cdot T_L(c,a)$	$\delta R_*(a', b') \cdot \sum_c S_L(b', c) \cdot T_L(c, a')$	$\mathcal{O}(\lvert\mathbf{D}\rvert^{\varepsilon})$
$\sum_{a,b,c} R_*(a,b) \cdot S_H(b,c) \cdot T_H(c,a)$	$\delta R_*(a', b') \cdot \sum_c T_H(c, a') \cdot S_H(b', c)$	$\mathcal{O}(\lvert\mathbf{D}\rvert^{1-\varepsilon})$

Adaptive Maintenance Strategy

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

Skew-aware View	Evaluation from left to right	Time
$\sum_{a,b,c} R_*(a,b) \cdot S_L(b,c) \cdot T_L(c,a)$	$\delta R_*(a',b') \cdot \sum_c S_L(b',c) \cdot T_L(c,a')$	$\mathcal{O}(\lvert\mathbf{D}\rvert^{\varepsilon})$
$\sum_{a,b,c} R_*(a,b) \cdot S_H(b,c) \cdot T_H(c,a)$	$\delta R_*(a',b') \cdot \sum_c T_H(c,a') \cdot S_H(b',c)$	$\mathcal{O}(\lvert\mathbf{D}\rvert^{1-\varepsilon})$
$\sum_{a,b,c} R_*(a,b) \cdot S_L(b,c) \cdot T_H(c,a)$	$\delta R_*(a',b') \cdot \sum_c S_L(b',c) \cdot T_H(c,a')$	$\mathcal{O}(\lvert\mathbf{D}\rvert^{\varepsilon})$
	or	
	$\delta R_*(a',b') \cdot \sum_c T_H(c,a') \cdot S_L(b',c)$	$\mathcal{O}(\lvert\mathbf{D}\rvert^{1-\varepsilon})$

Adaptive Maintenance Strategy

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

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

Adaptive Maintenance Strategy

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

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

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

Materialized Auxiliary Views

$$V_{RS}(a,c) = \sum_b R_H(a,b) \cdot S_L(b,c)$$
$$V_{ST}(b,a) = \sum_c S_H(b,c) \cdot T_L(c,a)$$
$$V_{TR}(c,b) = \sum_a T_H(c,a) \cdot R_L(a,b)$$

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

Update	Compute the difference for V_{RS}	Time
$\delta R_H = \{(a',b') \mapsto m\}$	$\delta R_H(a',b') \cdot S_L(b',c)$	$\mathcal{O}(\lvert\mathbf{D}\rvert^{\varepsilon})$
$\delta S_L = \{(b',c') \mapsto m\}$	$\delta S_L(b',c') \cdot R_H(a,b')$	$\mathcal{O}(\lvert\mathbf{D}\rvert^{1-\varepsilon})$

Materialized Auxiliary Views

$$V_{RS}(a,c) = \sum_b R_H(a,b) \cdot S_L(b,c)$$
$$V_{ST}(b,a) = \sum_c S_H(b,c) \cdot T_L(c,a)$$
$$V_{TR}(c,b) = \sum_a T_H(c,a) \cdot R_L(a,b)$$

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

Update	Compute the difference for V_{RS}	Time
$\delta R_H = \{(a',b') \mapsto m\}$	$\delta R_H(a',b') \cdot S_L(b',c)$	$\mathcal{O}(\lvert\mathbf{D}\rvert^{\varepsilon})$
$\delta S_L = \{(b',c') \mapsto m\}$	$\delta S_L(b',c') \cdot R_H(a,b')$	$\mathcal{O}(\lvert\mathbf{D}\rvert^{1-\varepsilon})$

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

$$\begin{aligned} |V_{RS}(a,c)| &\leq |R_H| \cdot \max_b\{|S_L(b,c)|\} = \mathcal{O}(|\mathbf{D}|^{1+\varepsilon}) \\ |V_{RS}(a,c)| &\leq |S_L| \cdot \max_b\{|R_H(a,b)|\} = \mathcal{O}(|\mathbf{D}|^{1+(1-\varepsilon)}) \end{aligned}$$

Materialized Auxiliary Views

$$V_{RS}(a,c) = \sum_b R_H(a,b) \cdot S_L(b,c)$$
$$V_{ST}(b,a) = \sum_c S_H(b,c) \cdot T_L(c,a)$$
$$V_{TR}(c,b) = \sum_a T_H(c,a) \cdot R_L(a,b)$$

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

Update	Compute the difference for V_{RS}	Time
$\delta R_H = \{(a',b') \mapsto m\}$	$\delta R_H(a',b') \cdot S_L(b',c)$	$\mathcal{O}(\lvert\mathbf{D}\rvert^{\varepsilon})$
$\delta S_L = \{(b',c') \mapsto m\}$	$\delta S_L(b',c') \cdot R_H(a,b')$	$\mathcal{O}(\lvert\mathbf{D}\rvert^{1-\varepsilon})$

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

$$\begin{array}{rcccr} |V_{RS}(a,c)| & \leq & |R_H| \cdot \max_b\{|S_L(b,c)|\} & = & \mathcal{O}(|\mathbf{D}|^{1+\varepsilon}) \\ |V_{RS}(a,c)| & \leq & |S_L| \cdot \max_b\{|R_H(a,b)|\} & = & \mathcal{O}(|\mathbf{D}|^{1+(1-\varepsilon)}) \end{array}$$

- Overall: Update Time $\mathcal{O}(|\mathbf{D}|^{\max\{\varepsilon,1-\varepsilon\}})$ and Space $\mathcal{O}(|\mathbf{D}|^{1+\min\{\varepsilon,1-\varepsilon\}})$

Rebalancing Partitions

- Updates can change the frequencies of values and the heavy/light threshold!
- This may require rebalancing of partitions:

  $\Longrightarrow$ Minor rebalancing: Transfer tuples from one to the other part of the same relation!

  $\Longrightarrow$ Major rebalancing: Recompute partitions and views from scratch!
- Both forms of rebalancing require superlinear time.
- The rebalancing times amortize over sequences of updates.