Computing Derived Facts: Selective Nearest Neighbor Joins

Francesco Cafagna

Tames Workshop
08.07.2013

Outline:

Computing Derived Facts: Selective Nearest Neighbor Joins

Context

May 12, 2011 the content of nutrients cp (crude protein) and om (organic matter) in a hay sample is analyzed.

s

G	T	K	M
Hay	2011/05/12	CP	137 g
Hay	2011/05/12	OM	900 g

s

G	T	K	M
Hay	2011/05/12	CP	137 g
Hay	2011/05/12	OM	900 g
Hay	**2011/05/12**	**GE**	**191**

- Nutrient cp has value 137
- Nutrient om has value 900.
- **Derived nutrient** $ge = 0.08 * cp + 0.2 * om$ is the gross energy.
- Given $f(k_1, ..., k_n) \rightarrow k'$, derived nutrient k' is calculated using a **formula** on nutrients $k_1, ..., k_n$

Difficulties and Challenges

But the world is never perfect...

- Since lab-analyses are expensive, nutrients cp and om are not always measured and we have sparse data.
- What was the value of derived nutrient $ge = 0.08 * cp + 0.2 * om$ for feed Hay on May 13, 2011?

s

	G	T	K	M
r_1	Hay	2011/05/13	CP	137 g
	Hay	**2011/05/13**	**GE**	**??**

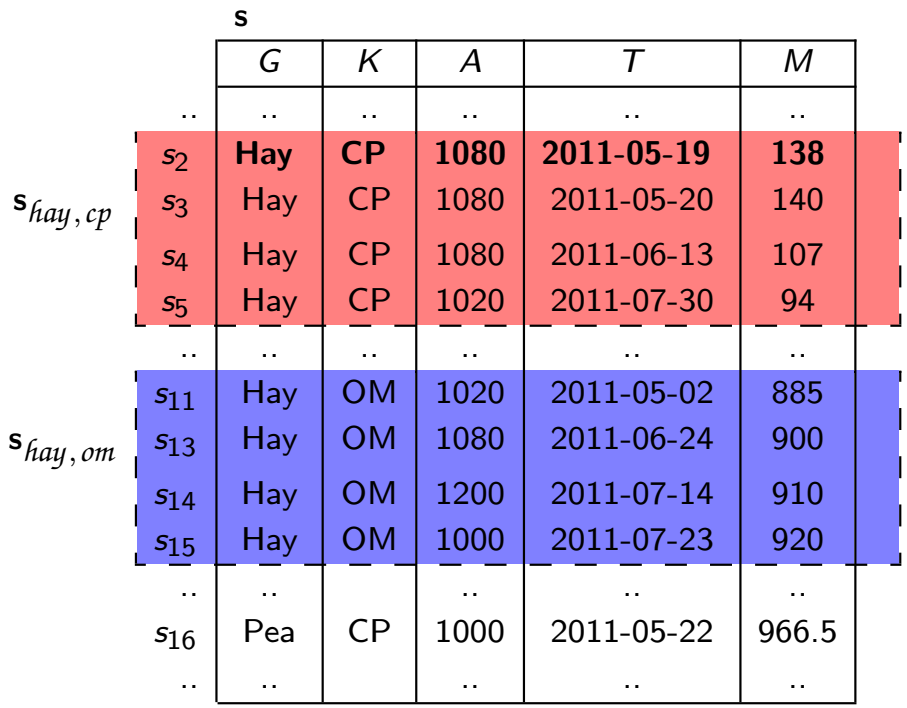
Problem: the value of om is missing!

- For feed Hay we only have its cp value $\Rightarrow$ ge for Hay cannot be computed for May 15, 2013!

What can we do?

- Since we have no om value for May 15 we use the temporally closest.

Application Scenario: The Swiss Feed DW



- Fact table **s** stores the lab analyses on animal feeds.
- Fact table **s** is logically partitioned. Partition $\mathbf{s}_{g,k}$ contains facts having feed g and nutrient k.
- A **fact** is a row of fact table **s** and stores the measure M of nutrient K in feed G on time T. A indicates the altitude where the feed is grown.

Problem Definition

r

	G	T
r_1	Hay	2011-05-21
r_2	Hay	2011-06-21
r_3	Hay	2011-07-21
..	..	..
r_4	Pea	2011-05-21
..	..	..

$\mathbf{r}_{hay}$

s

	G	K	A	T	M	
..	..	..	..	..	..	
s_3	Hay	CP	1080	2011-05-19	138	$\mathbf{s}_{hay,cp}$
s_4	Hay	CP	910	2011-05-20	140	θ
s_1	Hay	CP	1110	2011-06-13	107	
s_5	Hay	CP	1020	2011-07-30	94	
..	..	..	..	..	..	
s_{11}	Hay	OM	1020	2011-05-02	885	$\mathbf{s}_{hay,om}$
s_{13}	Hay	OM	1080	2011-06-24	900	
s_{14}	Hay	OM	1200	2011-07-14	910	
s_{15}	Hay	OM	1000	2011-07-23	920	
..	..	..	..	..	..	
s_{16}	Pea	CP	1000	2011-01-02	1.06	
..	..	..	..	..	..	

Problem definition: For $0.08 * cp + 0.2 * om \rightarrow ge$ and condition $\theta = A > 1000$ *compute the* **derived facts** ge *for relation* **r**

- Only measures for feed grown at an altitude $A > 1000$ may be considered!

$$\bigcup_{\mathbf{r}_g \subseteq \mathbf{r}} \pi_{\mathbf{G}, T, 'ge', 0.08*cp+0.2*om} \left(\mathbf{r}_g \overset{NN(T)}{\text{¿i}} \rho_{CP/M}(\sigma_\theta(\mathbf{s}_{g,cp})) \overset{NN(T)}{\text{¿i}} \rho_{OM/M}(\sigma_\theta(\mathbf{s}_{g,om})) \right)$$

$\mathbf{Q}^{ge}(\mathbf{r}, \sigma_\theta(\mathbf{s}))$

	G	T	K	M
q_1	Hay	2011-03-21	GE	188.04
q_2	Hay	2011-06-21	GE	188.56

The predicate θ

- What is θ?
  - θ is a condition that the facts used for calculating a derived fact must hold.

Example:

1. Only measurements of feeds coming from the mountain must be used
2. Only measurements with high reliability must be used
3. Only measurements of bio-feeds must be used
4. Only measurements of well matured feeds must be used
5. Only measurements of Zurich area must be used

State of the Art: Overview

Three techniques can be used for computing derived facts $\mathbf{Q}^{\mathcal{GE}}(\mathbf{r}, \sigma_\theta(\mathbf{s}))$
For each $r \in \mathbf{r}$, find its nearest neighbor by:

1. **SeqScan** (selection on θ) on $\mathbf{s}$ $=>$ nested loop
2. **Indexed MinMax** on $\mathbf{s}$ + filter on θ, (for each $r \in \mathbf{r}$ find efficiently the timestamp of the nearest neighbor in $\mathbf{s}$)
3. **Sort Merge** after evaluation of θ, (sort and scan $\mathbf{r}$ and $\sigma_\theta(\mathbf{s})$)

SeqScan

$\mathbf{r}_{hay}$

	G	T
→	Hay	15-05
- - →	Hay	15-06
	Hay	15-07

$\sigma_\theta(\mathbf{s}_{hay,cp})$

G	T	CP
Hay	May 01	137
Hay	May 10	140
Hay	May 30	119
Hay	Jun 19	106
Hay	Jul 12	95
Hay	Jul 16	94

MinMax

$\mathbf{r}_{hay}$

	G	T
→	Hay	15-05
- - →	Hay	15-06
	Hay	15-07

$\sigma_\theta(\mathbf{s}_{hay,cp})$

G	T	CP
Hay	May 01	137
Hay	May 10	140
Hay	May 30	119
Hay	Jun 19	106
Hay	Jul 12	95
Hay	Jul 16	94

SortMerge

$\mathbf{r}_{hay}$

	G	T
→	Hay	15-05
- - →	Hay	15-06
	Hay	15-07

$\sigma_\theta(\mathbf{s}_{hay,cp})$

G	T	CP
Hay	May 01	137
Hay	May 10	140
Hay	May 30	119
Hay	Jun 19	106
Hay	Jul 12	95
Hay	Jul 16	94

State of the Art: Overview

Three techniques can be used for computing derived facts $\mathbf{Q}^{\mathcal{ge}}(\mathbf{r}, \sigma_\theta(\mathbf{s}))$
For each $r \in \mathbf{r}$, find its nearest neighbor by:

1. **SeqScan** (selection on θ) on $\mathbf{s}$ $=>$ nested loop
2. **Indexed MinMax** on $\mathbf{s}$ + filter on θ, (for each $r \in \mathbf{r}$ find efficiently the timestamp of the nearest neighbor in $\mathbf{s}$)
3. **SortMerge** after evaluation of θ, (sort and scan $\mathbf{r}$ and $\sigma_\theta(\mathbf{s})$)

What is our problem?

- We want to evaluate how techniques for computing Nearest Neighbor Joins perform in deriving facts

Why is this a problem?

1. **MinMax** becomes inefficient in computing derived facts when a selection σ_θ on the fact table $\mathbf{s}$ is involved due to index false hits
2. **SortMerge** becomes inefficient when the number of outer partitions $\mathbf{r}_g \subseteq \mathbf{r}$ grows due to multiple sorting

Indexed MinMax: Literature

- MinMax has been proposed in *"K Nearest Neighbor Queries and KNN-Joins in Large Relational Databases (Almost) for Free"* by Bin Yao, Feifei Li, et al. and presented at ICDE 2010
- This work show how to reduce multidimensional nearest neighbor joins to one dimensional nearest neighbor joins using z-values $\Rightarrow$ merge multiple dimensions into one.
- An SQL implementation of the nearest neighbor join is presented
- An efficient QEP for the SQL query is presented on the assumption that an index on the distance attribute exists.

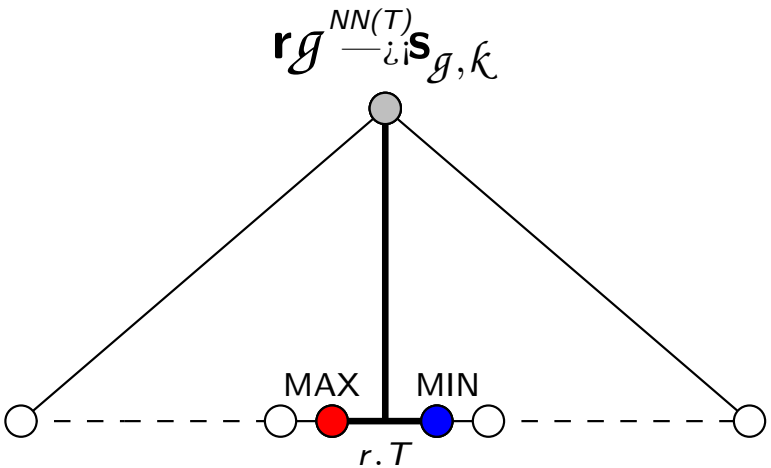
State of the Art: Indexed MinMax

MinMax can be efficiently implemented using SQL by two MIN / MAX subqueries. [3]

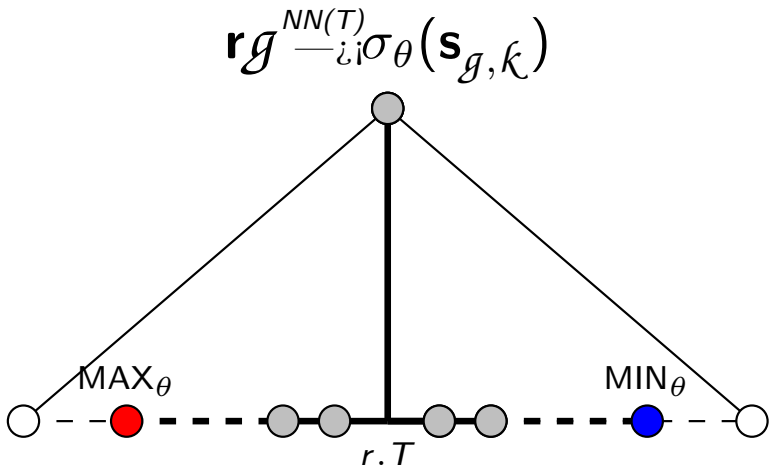
SELECT MAX(T) FROM $\mathbf{s}_{\mathcal{g},\mathcal{k}}$ WHERE $\mathbf{s}_{\mathcal{g},\mathcal{k}}.T < r.T$ **AND** θ

SELECT MIN(T) FROM $\mathbf{s}_{\mathcal{g},\mathcal{k}}$ WHERE $\mathbf{s}_{\mathcal{g},\mathcal{k}}.T >= r.T$ **AND** θ

Q: What is a good θ?



Example:
$\mathbf{Q}^{\mathcal{ge}}(\mathbf{r}, \sigma_{\top}(\mathbf{s}))$= 135ms



Example:
$\mathbf{Q}^{\mathcal{ge}}(\mathbf{r}, \sigma_{height>1000}(\mathbf{s}))$: 250$ms$

When σ_{θ} becomes very selective, performances turn out to be bad!

State of the Art: Indexed MinMax

MinMax can be efficiently implemented using SQL by two MIN / MAX subqueries. [3]

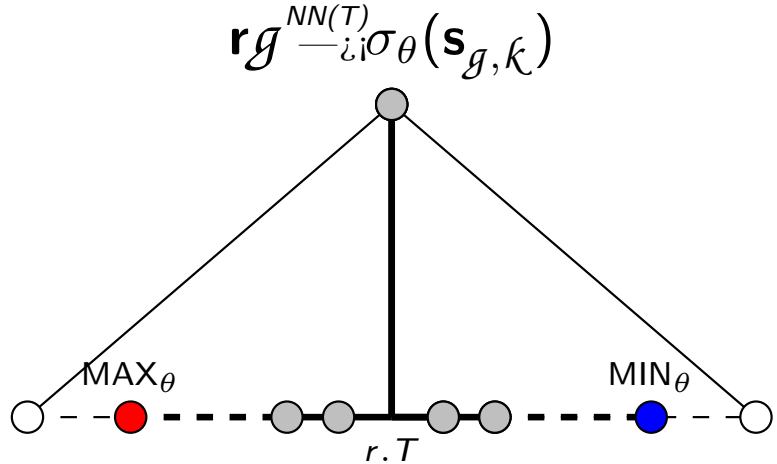
SELECT MAX(T) FROM $\mathbf{s}_{\mathcal{G},\mathcal{K}}$ WHERE $\mathbf{s}_{\mathcal{G},\mathcal{K}}.T < r.T$ **AND** θ

SELECT MIN(T) FROM $\mathbf{s}_{\mathcal{G},\mathcal{K}}$ WHERE $\mathbf{s}_{\mathcal{G},\mathcal{K}}.T >= r.T$ **AND** θ

Q: What is a good θ?

selectivity(θ)	runtime
100 %	x
10 %	1.11 x
5 %	1.22 x
2 %	1.55 x
1 %	2.09 x
0.5 %	3.14 x
0.2 %	6.24 x
0.1 %	11.5 x

Example:
$\mathbf{Q}^{\mathcal{GE}}(\mathbf{r}, \sigma_{\top}(\mathbf{s}))$= 135ms



Example:
$\mathbf{Q}^{\mathcal{GE}}(\mathbf{r}, \sigma_{height>1000}(\mathbf{s}))$: 250$ms$

When σ_{θ} becomes very selective, performances turn out to be bad!

SortMerge: Literature

- SortMerge has been proposed in *"The Similarity Join Database Operator"* by Y. Silva, W. G. Aref, et al. and presented at ICDE 2010
- This work propose 4 similarity join operators, one of which is the nearest neighbor join
- An extension of SortMerge is presented for integrating nearest neighbor join in DBMSs
- Equivalence Rules are presented for optimizing nearest neighbor join queries

SortMerge: the Algorithm

- After the evaluation of θ, partitions $\mathbf{r}_g$ and $\mathbf{s}_{g,k}$ are sorted by the distance attribute. $\Rightarrow$ only *cp* and *om* measures with $A > 1000$ will be sorted!
- Scan $\mathbf{r}_g$: the nearest neighbor of a given row $r \in \mathbf{r}_g$ is the current row of $\mathbf{s}_{g,k}$ or one of the next.
- There is no need to refetch tuples in **s** which are not n.n. of any *r*
- A seq scan after the sorting is enough to compute a join.

$\mathbf{r}_{hay}$

G	T
Hay	May 15
Hay	Jun 15
Hay	Jul 15

$\sigma_\theta(\mathbf{s}_{hay,cp})$

G	T	CP	
Hay	May 01	137	?
Hay	May 10	140	?
Hay	May 30	119	?
Hay	Jun 19	106	?
Hay	Jul 12	95	?
Hay	Jul 16	94	

$\mathbf{r}_g \overset{NN(T)}{\text{¿¡}} \sigma_\theta(\mathbf{s}_{hay,k})$

G	T	CP
Hay	May 15	140
Hay	Jun 15	106
Hay	Jul 15	94

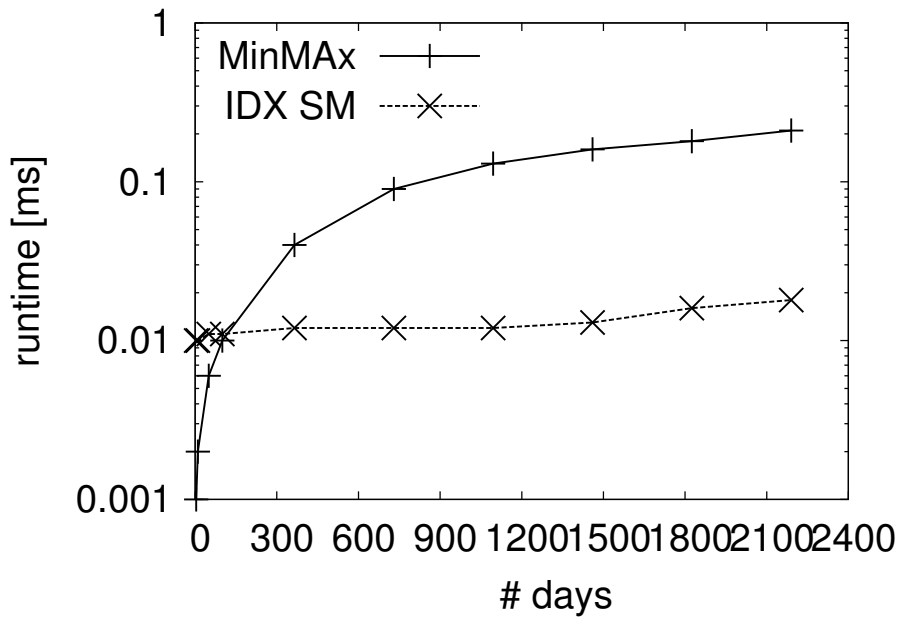
- Total complexity: $O(|\mathbf{r}_g|\ log\ |\mathbf{r}_g| + |\mathbf{s}_{g,k}|\ log\ |\mathbf{s}_{g,k}|)$

An Experimental Evaluation

- Which approach suits better for computing derived facts $\mathbf{Q}^{\mathcal{ge}}(\mathbf{r}, \sigma_\theta(\mathbf{s}))$?
- There is no clear winner.
- It depends on:
  1. the size of $\mathbf{r}$
     i.e. the number of tuples for which we want to compute $\mathcal{ge}$
  2. the number of partitions $\mathbf{r}_\mathcal{g} \subseteq \mathbf{r}$
     i.e. the number of different feeds for which we want to compute $\mathcal{ge}$
  3. the selectivity of predicate θ
     i.e.how many $\mathcal{cp}$ *and* $\mathcal{om}$ *measurements satisfy* θ

An Experimental Evaluation
1) The size of r

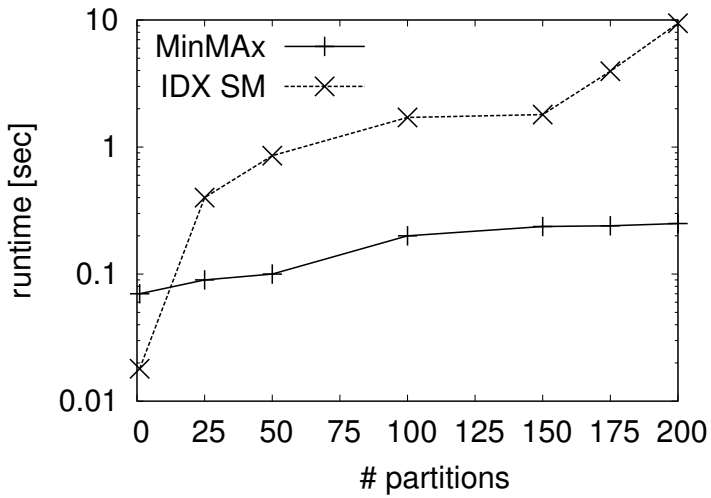
- Which approach suits better for computing derived facts $\mathbf{Q}^{\mathcal{ge}}(\mathbf{r}, \sigma_{\top}(\mathbf{s}))$?
- Let's ignore θ.
- We consider a monopartitioned **r** (only one feed) and increase the number of tuples in **r** (timestamps for which the derived fact must be calculated).



- MinMax suffers from large outer relations!

An Experimental Evaluation
2) The Number of Outer Partitions

- Which approach suits better for computing derived facts $\mathbf{Q}^{\mathcal{GE}}(\mathbf{r}, \sigma_{\top}(\mathbf{s}))$?
- Let's ignore θ.
- We fix the size of **r** and increase the number of outer partitions (feeds).

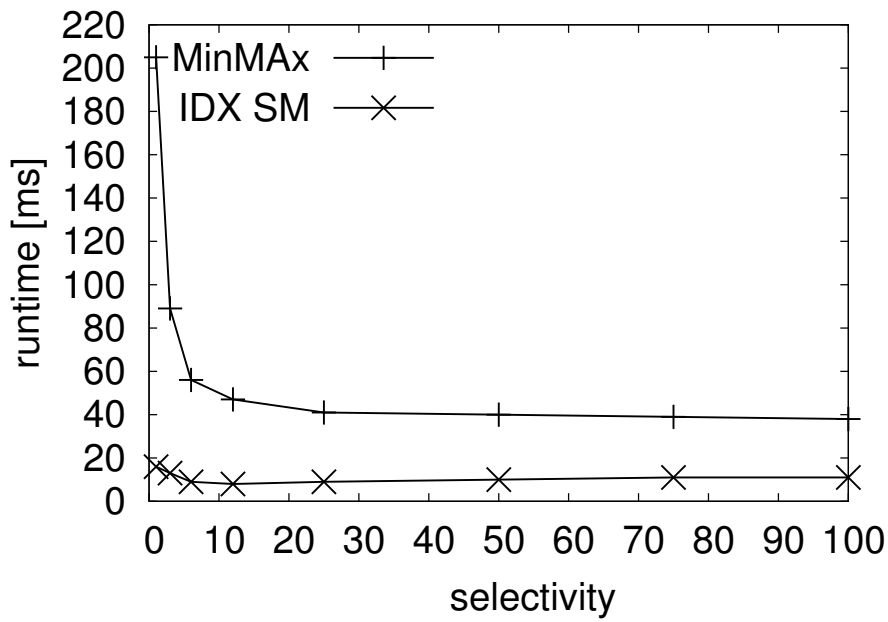


- MinMax is unaffected by the number of outer partitions.
- SortMerge suffers from highly partitioned outer relations!

  - *SortMerge without partitions would not depend on the distribution of the tuples of* **r** *but is slow*

An Experimental Evaluation
3) The Selectivity of Predicate θ

- Which approach suits better for computing derived facts $\mathbf{Q}^{\mathcal{GE}}(\mathbf{r}, \sigma_\theta(\mathbf{s}))$?
- We consider a monopartitioned **r** (only one feed)
- We fix the size of **r** to 365 (1 year)
- We vary the selectivity of predicate θ



- MinMax suffers from highly selective predicates!

Summary

- Which approach suits better for computing derived facts $\mathbf{Q}^{\mathcal{GE}}(\mathbf{r}, \sigma_\theta(\mathbf{s}))$?
- MinMax depends on the size of $\mathbf{r}$ D
- MinMax is unaffected by the number of outer partitions $\mathbf{r}_\mathcal{G} \subseteq \mathbf{r}$ U
- MinMax depends on the selectivity of θ D

- SortMerge does not depends on the size of $\mathbf{r}$ U
- SortMerge depends the number of outer partitions D
- Sort Merge does not depend on the selectivity of θ U

Computing Derived Facts: Selective Nearest Neighbor Joins

Francesco Cafagna

Tames Workshop
08.07.2013