Differentiating Relational Queries

PhD Workshop at VLDB21

Paul Peseux

Tutors: T.Paquet, M.Berar, V.Nicollet

x

August 16, 2021

Outline

Outline

Context

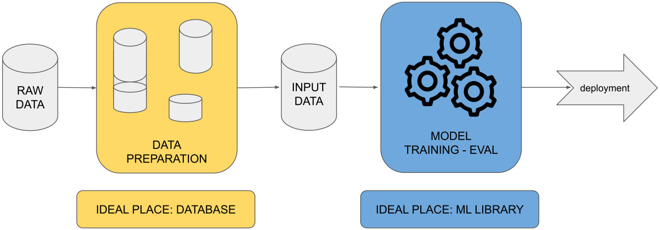


Figure: Classic Machine Learning Pipeline.

Context

- costly data transfer (Schüle 2019)
-

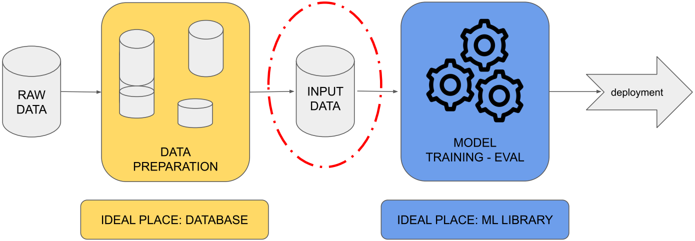


Figure: Classic Machine Learning Pipeline.

Context

- costly data transfer (Schüle 2019)
- ML libraries built for computer vision, NLP . . .
  $\longrightarrow$ **inadapted to relational data**

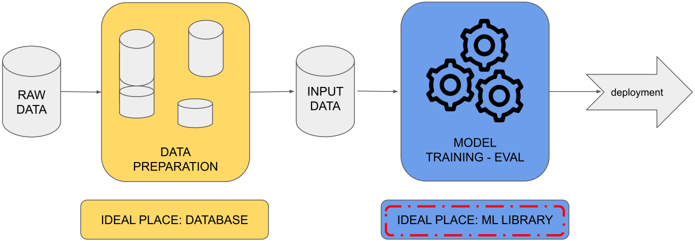


Figure: Classic Machine Learning Pipeline.

Context

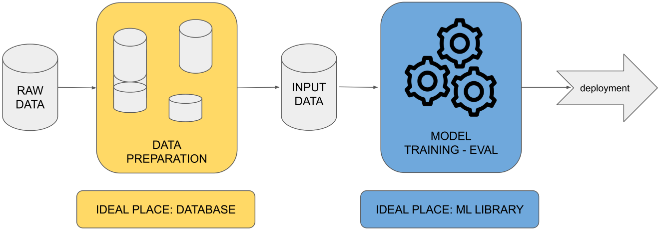


Figure: Classic Machine Learning Pipeline.

Context

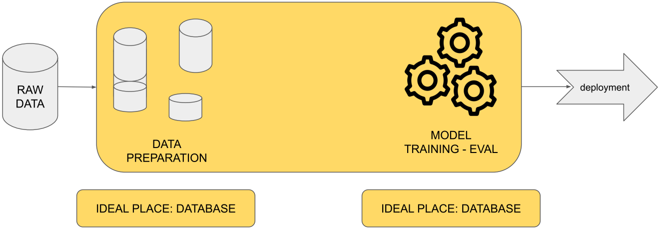


Figure: Proposed Pipeline.

Context

Many Machine Learning methods are based on **gradient** methods.

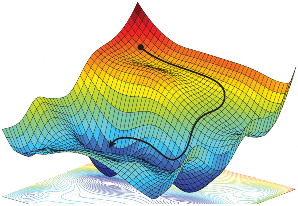


Figure: Gradient Descent, source (Hutson)

Context

Many Machine Learning methods are based on **gradient** methods.

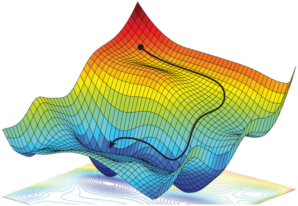


Figure: Gradient Descent, source (Hutson)

⟶ To optimize models, **relational queries differentiation** is missing (Schüle 2019)

Context

Differentiating Relational Queries $\Leftrightarrow$ Derivative of the Relational Queries

⚠

This is **not** differential dataflow (Mcsherry 2021)

Context

```
SELECT X FROM Observations
        should give
SELECT 1 FROM Observations
```

```
SELECT X * X FROM Observations
        should give
SELECT 2 * X FROM Observations
```

Figure: What we are looking for

Outline

Formalization

For the rest of the presentation, optimisation means minimisation and is allowed through gradient descent.

$$x^\star = \underset{x}{arg\,min} \quad f(x)$$

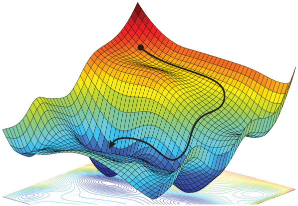


Figure: Gradient Descent, source (Hutson)

f is called *loss*

Formalization

We want to minimize (*and thus compute the gradient of*):

```
SELECT sum(loss) FROM Observations
```

For that we need:

- a framework
- constraints on the query

Formalization

Minimization is only possible on scalar.

$$Loss = \sum_{i \in Obs} loss_i = \sum_{i \in Obs} f(data_i)$$

Formalization

Minimization is only possible on scalar.

$$Loss = \sum_{i \in Obs} loss_i = \sum_{i \in Obs} f(data_i)$$

Constraint 1

Loss is computed line by line.

Example

Let's make it concrete with the Chicago taxi trip dataset.

Trips

taxiId	Company	distance	Tips
d904764719c56cfb36906cb74c...	Choice Taxi Association	0.7	2.0
11f73b08790612efe341cf8cf69...	Choice Taxi Association	1.2	2.7
e61ce97d61bec30e506e2ff56ea...	Chicago Medallion Management	15.1	5.0
74605d6aa0c8ba08190a5824f7...	Blue Ribbon Taxi Association Inc.	0.2	3.7
abb33898b8d70d22237631f6bd...	Chicago Medallion Management	3.4	3.1
e6a8d59b08b735bf949c799157...	Taxi Affiliation Services	0.2	2.6
d24314a66ebc6319a50cc335d6...	Taxi Affiliation Services	0.1	2.3
9d20d7617e35a8d763ca0bbe3b...	Northwest Management LLC	4.5	3.0
1226e3d8b86299171525b37f43...	Choice Taxi Association	0.3	1.1
7985168ea616aa1c4437d4ebe4...	KOAM Taxi Association	13.2	5.7
365689b9f3107b807470fe16b7...	Taxi Affiliation Services	0.2	1.4
8c4cce532e3fa081753ea28c1b...	Taxi Affiliation Services	1.3	20.0
627de0f7c9251f9731fe27af6bb...	Taxi Affiliation Services	2.7	2.0
705cc88d7a216145f6c762aa70...	Dispatch Taxi Affiliation	3.7	2.3
01dfe8a384fbd91738442964e7...	Dispatch Taxi Affiliation	1.2	4.0
493b6af5931ea2c7c6a82d9d6e...	Taxi Affiliation Services	0.9	7.0
e5a4715f2ec431f404f71c7e4d0...	Choice Taxi Association	2.7	1.0
9aabfe03b5b0f6d742bc86499ea...	Choice Taxi Association	13.4	13.0
e61ce97d61bec30e506e2ff56ea...	Chicago Medallion Management	2.2	2.0

Figure: Chicago trips dataset, source (Chicago)

Example

Objective: *explain the trip's tip with distance and company "quality".*

With Linear Regression as the machine learning model.

Model

$$\boxed{Tip_{estimated} = a_{company} \times distance + b}$$

One slope per company; Intercept is shared among all the taxis.

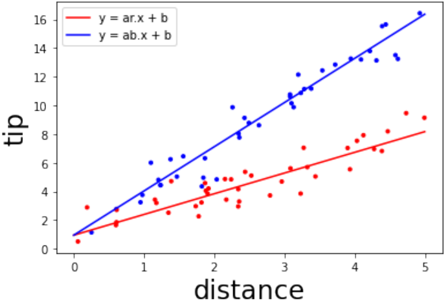


Figure: Model

Approach

Comparing the matrix approach (ML Libraries) and relational one

Approach

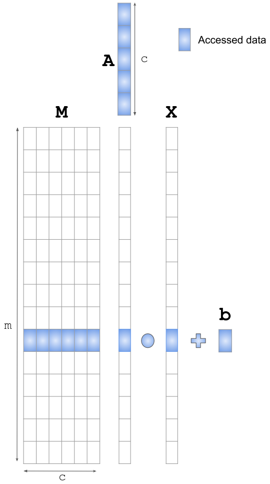


Figure: Matrix approach

$$(M.A) \circ X + b$$

$\circ$ is the point-wise product

Approach

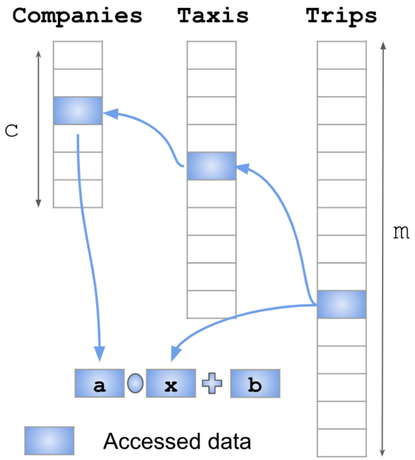


Figure: Relational approach

Approach

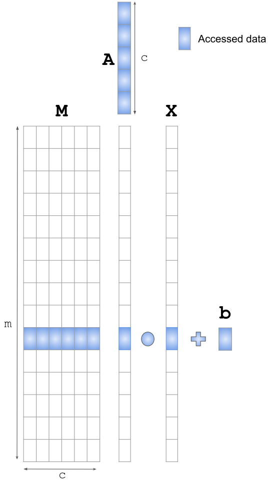


Figure: Matrix approach

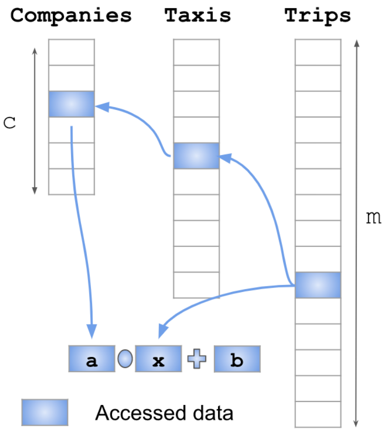


Figure: Relational approach

Linear Regression on the Chicago dataset

Model

$$\boxed{Tip_{estimated} = a_{company} \times distance + b}$$

In SQL it gives

```sql
WITH TaxisWithSlope AS (
  SELECT *
  FROM Taxis
  INNER JOIN Companies
      ON Taxis.company = Companies.company)

SELECT
  tripId,
  POWER(Estimated - tip, 2) AS Loss
  FROM (
    SELECT
      Trips.*,
      TaxisWithSlope.slope * Trips.distance + @intercept AS Estimated
    FROM Trips
    INNER JOIN TaxisWithSlope
        ON Trips.taxiId = TaxisWithSlope.taxiId  )
  AS Observations;
```

SQL query of our model.

Formalization

$$\triangle \text{Trips} = \text{Observations}$$

$$Loss = \sum_{t \in Trips} loss_t = \sum_{t \in Trips} f(data_t) = \sum_{t \in Trips} (a_{comp_t} \times dist_t + b - tip_t)^2$$

with

$$f(a, x, b, y) = (ax + b - y)^2$$

Formalization

$$\triangle \text{Trips} = \text{Observations}$$

$$Loss = \sum_{t \in Trips} loss_t = \sum_{t \in Trips} f(data_t) = \sum_{t \in Trips} (a_{comp_t} \times dist_t + b - tip_t)^2$$

with

$$f(a, x, b, y) = (ax + b - y)^2$$

Then it is feasible to compute gradients!

$$\frac{\partial f}{\partial a} \quad ; \quad \frac{\partial f}{\partial x} \quad ; \quad \frac{\partial f}{\partial b} \quad ; \quad \frac{\partial f}{\partial y}$$

Constraint 2

f has to be differentiable.

Formalization

$$f(a,x,b,y) = (ax + b - y)^2$$

Figure: Inputs origin.

Formalization

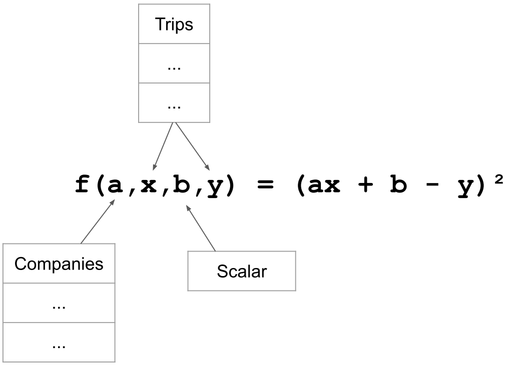


Figure: Inputs origin.

Math

Relational

```sql
WITH TaxisWithSlope AS (
  SELECT *
  FROM Taxis
  INNER JOIN Companies
      ON Taxis.company = Companies.company)

SELECT
  tripId,
  POWER(Estimated - tip, 2) AS Loss
  FROM (
    SELECT
      Trips.*,
      TaxisWithSlope.slope * Trips.distance + @intercept AS Estimated
    FROM Trips
    INNER JOIN TaxisWithSlope
        ON Trips.taxiId = TaxisWithSlope.taxiId )
  AS Observations;
```

SQL query of our model.

Approach

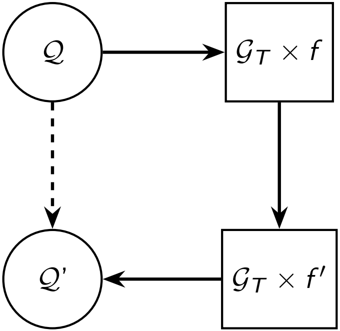


Figure: Path to Differentiating Relational Queries.

$\mathcal{Q}$: query
$\mathcal{G}_T$: tables graph
f : loss function

Outline

Approach

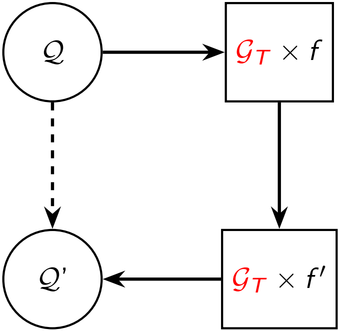


Figure: Path to Differentiating Relational Queries.

$\mathcal{Q}$: query
$\mathcal{G}_T$: tables graph
f : loss function

Definition 1 (Broadcast)

Let's note "$T_A \longrightarrow T_B$" when the primary key of T_A is a foreign key in T_B. It is said that T_A broadcasts into T_B.

Tables Relations

Definition 1 (Broadcast)

Let's note "$T_A \longrightarrow T_B$" when the primary key of T_A is a foreign key in T_B. It is said that T_A broadcasts into T_B.

Tables used in the query with the relationship $\longrightarrow$ forms a **graph** $\mathcal{G}_T$.

Tables Relations

Definition 1 (Broadcast)

Let's note "$T_A \longrightarrow T_B$" when the primary key of T_A is a foreign key in T_B. It is said that T_A broadcasts into T_B.

Tables used in the query with the relationship $\longrightarrow$ forms a **graph** $\mathcal{G}_T$.

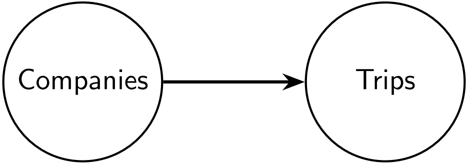


Figure: Graph from our linear regression model.

Tables Relations

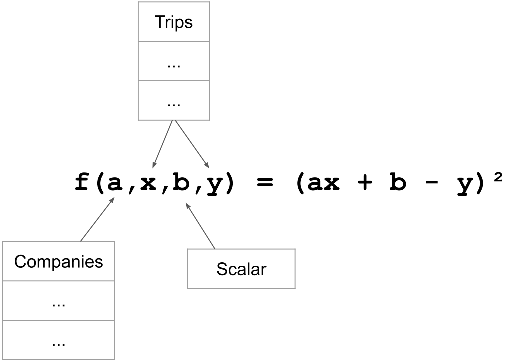


Figure: Inputs origin.

Tables Relations

Let be

- T a table used in the query
- $T.A$ be a column of T
- a the input of f representing $T.A$

If T (transitively) broadcasts into *Observations* then a the input of f representing $T.A$ is a **scalar**.

Tables Relations

Let be

- T a table used in the query
- $T.A$ be a column of T
- a the input of f representing $T.A$

If T (transitively) broadcasts into *Observations* then a the input of f representing $T.A$ is a **scalar**.

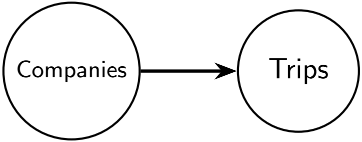


Figure: Graph from our linear regression model.

$$\boxed{Tip_{estimated} = a_{company} \times distance + b}$$

Approach

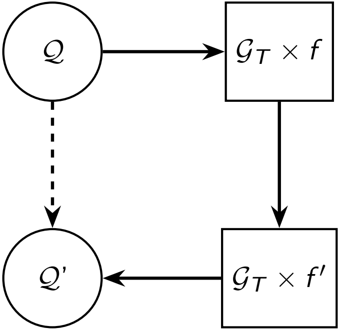


Figure: Path to Differentiating Relational Queries.

$\mathcal{Q}$: query
$\mathcal{G}_T$: tables graph
f : loss function

Outline

Approach

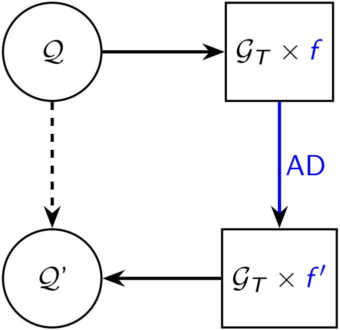


Figure: Path to Differentiating Relational Queries.

$\mathcal{Q}$: query
$\mathcal{G}_T$: tables graph
f : loss function
AD : Automatic Differentiation

Automatic Differentiation

P a program that apply the mathematical function f to its inputs.

Automatic Differentiation constructs program the program P' that apply f' to its inputs.

Automatic Differentiation

P a program that apply the mathematical function f to its inputs.

Automatic Differentiation constructs program the program P' that apply f' to its inputs.

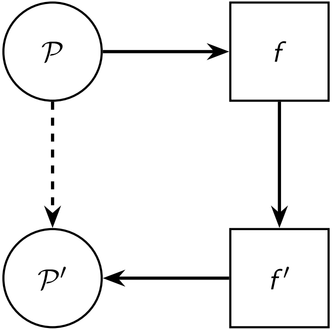


Figure: Automatic Differentiation.

Automatic Differentiation

- Fortran, C: Tapenade
- Python: Tangent, Myia
- Julia: Zygote
- F#: DiffSharp
- ...

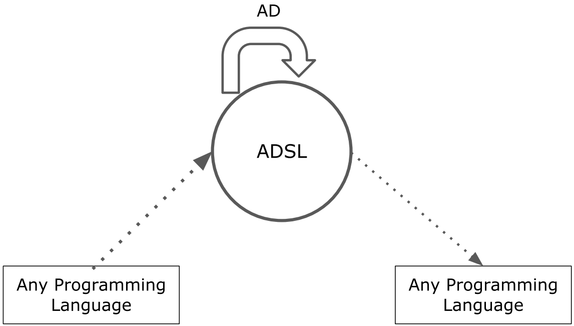
Automatic Differentiation

- Fortran, C: Tapenade
- Python: Tangent, Myia
- Julia: Zygote
- F#: DiffSharp
- ...

- not differentiating a specific programming language.
- define a narrowed programming language: **ADSL**. Similar to (Abadi 2019) (Hu 2020) (Mak 2020).

ADSL is **closed by differentiation**

Automatic Differentiation compilation



We can use this pipeline to differentiate a function written in any programming language *You just need to pay the price of compilation*.

Outline

Implementation

This work has been implemented at Lokad:

- on the DSL *Envision*
- live in production

Optimization through gradient descent is used daily and triggers orders on millions of SKUs.

Outline

Conclusion

In this work we've presented a framework on automatic differentiation on relational queries.

$$
\begin{tikzcd}
\mathcal{Q} \arrow[r] \arrow[d, dashed] & \mathcal{G}_T \times f \arrow[d] \\
\mathcal{Q}' & \mathcal{G}_T \times f' \arrow[l]
\end{tikzcd}
$$

Figure: Path to Differentiating Relational Queries.

Conclusion

This will unlock ML model construction and optimisation in databases.

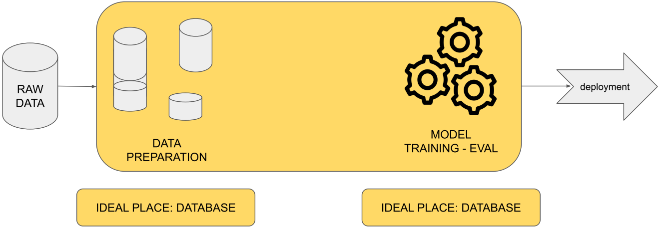


Figure: Proposed Pipeline.

Conclusion

Thanks for listening!

References

[Abadi 2019] Martín Abadi et Gordon Plotkin. *A simple differentiable programming language*. Proceedings of the ACM on Programming Languages, vol. 4, pages 1–28, 12 2019.

[Chicago] City Of Chicago. *image*. `https://data.cityofchicago.org/Transportation/Taxi-Trips/wrvz-psew`. Accessed: 2021-07-13.

[Hu 2020] Y. Hu, L. Anderson, Tzu-Mao Li, Q. Sun, N. Carr, Jonathan Ragan-Kelley et F. Durand. *DiffTaichi: Differentiable Programming for Physical Simulation*. ArXiv, vol. abs/1910.00935, 2020.

[Hutson] Matthew Hutson. *image*. `https://www.sciencemag.org/news/2018/05/ai-researchers-allege-machine-learning-alchemy`. Accessed: 2021-07-15.

[Mak 2020] Carol Mak et C. Ong. *A Differential-form Pullback Programming Language for Higher-order Reverse-mode Automatic Differentiation*. ArXiv, vol. abs/2002.08241, 2020.

[Mcsherry 2021] Frank Mcsherry, Derek Murray, Rebecca Isaacs et Michael Isard. *Differential dataflow*. 08 2021.

[Schüle 2019] Maximilian E. Schüle, Frédéric Simonis, Thomas Heyenbrock, A. Kemper, Stephan Günnemann et T. Neumann. *In-Database Machine Learning: Gradient Descent and Tensor Algebra for Main Memory Database Systems*. In BTW, 2019.

Appendix

Demo

Link to an example

ADSL

$$
\begin{array}{lcll}
\langle S \rangle & ::= & . & \\
 & \mid & \langle v \leftarrow e \rangle & \text{Variable assignment} \\
 & \mid & \langle \mathit{Cond}\ (v \quad \Psi \quad P_T \quad P_E \quad \Phi) \rangle & \text{Conditional} \\
 & \mid & \langle \mathit{For}\ (\chi \quad P \quad \Xi) \rangle & \text{Loop} \\
 & \mid & \langle \mathit{Return}\ v \rangle & \text{Output of a program} \\
 & & & \\
\langle e \rangle & ::= & . & \\
 & \mid & \langle v \rangle & \text{Variable} \\
 & \mid & \langle f \rangle & \text{Scalar} \\
 & \mid & \langle b \rangle & \text{Boolean} \\
 & \mid & \langle v + w \rangle & \text{Variable Addition} \\
 & \mid & \langle \mathit{Call1}\ op\ v \rangle & \text{Function Call} \\
 & \mid & \langle \mathit{Call2}\ op\ v\ w \rangle & \text{Function Call (2 parameters)} \\
 & \mid & \langle \mathit{Param}\ i \rangle & \text{Parameter access} \\
 & \mid & \langle \mathit{Const}\ i \rangle & \text{Constant access} \\
 & \mid & \langle v \quad \triangleleft \quad \beta \quad \rangle & \text{Broadcast Projector} \\
 & \mid & \langle v \quad \triangleright \quad \alpha \quad \rangle & \text{Aggregation Projector} \\
 & \mid & \langle \mathit{Pred} \rangle & \text{Predicate}
\end{array}
$$

ADSL

$$
\begin{array}{rcl}
\langle\, \mathit{Pred} \,\rangle & ::= & . \\
& | & \langle\, \mathit{And}\ v\ w \,\rangle \\
& | & \langle\, \mathit{Or}\ v\ w \,\rangle \\
& | & \langle\, \mathit{Not}\ v \,\rangle \\
& | & \langle\, v < w \,\rangle \\
& | & \langle\, v \leq w \rangle
\end{array}
$$

PolyStar

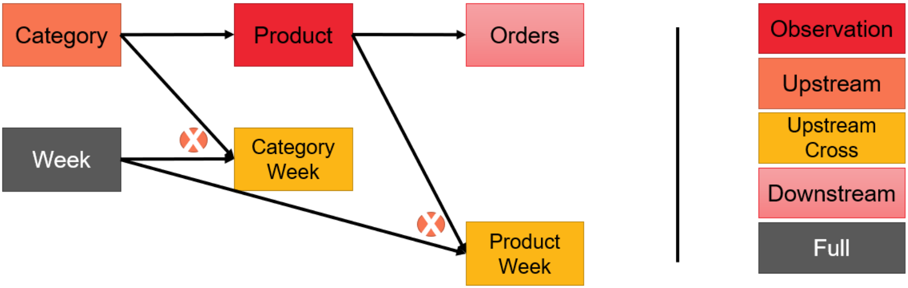


Figure: PolyStar

Realtional versus Math

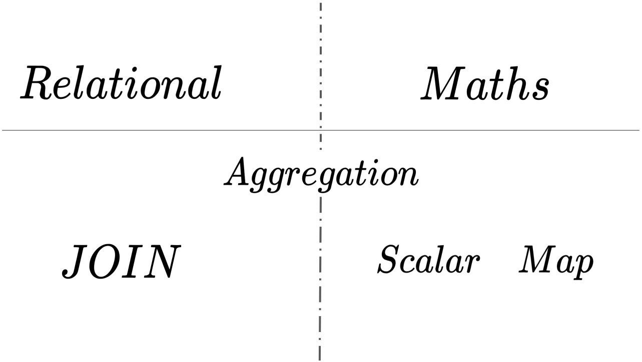


Figure: Realtional - Math decomposition