Differentiating Relational Queries

PhD Workshop at VLDB21

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

x

August 16, 2021

Outline

Outline

Context

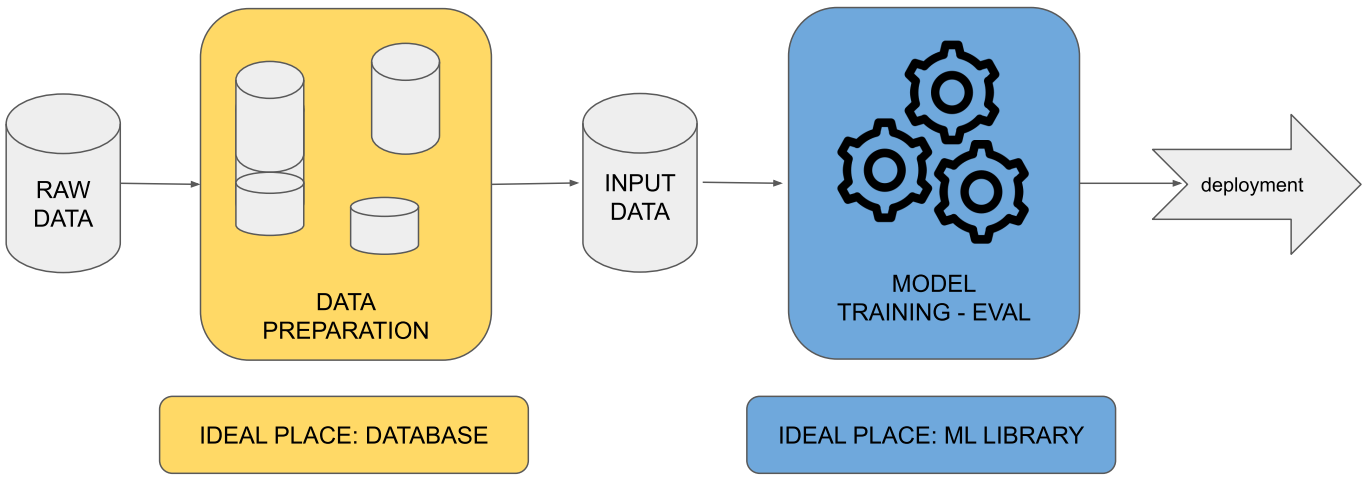


Figure: Classic Machine Learning Pipeline.

Context

- costly data transfer (Schüle 2019)
-

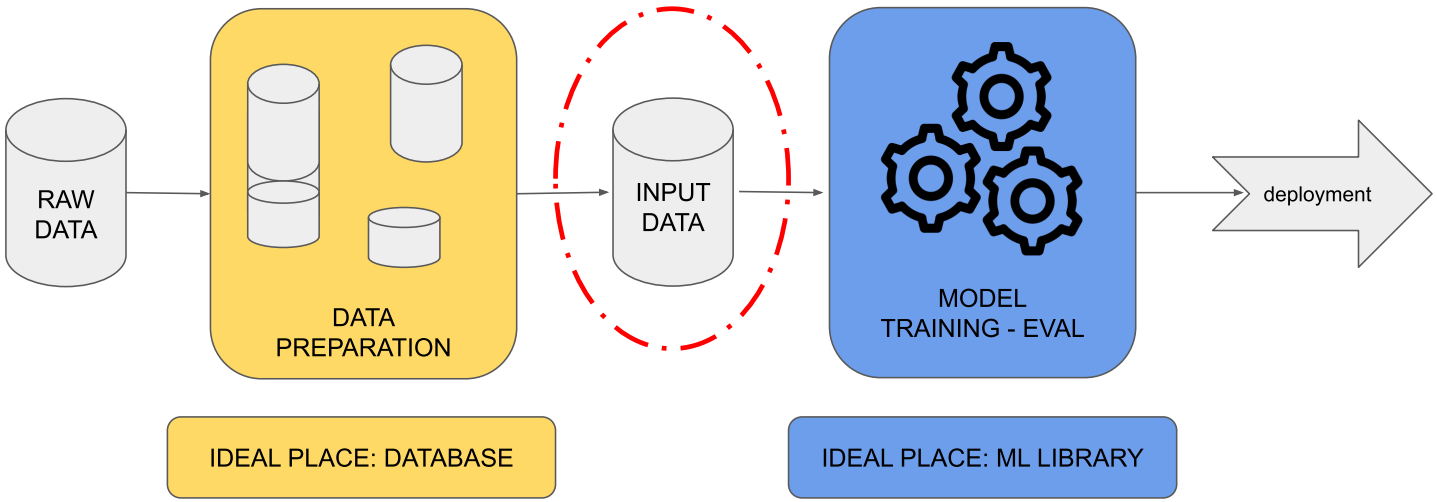


Figure: Classic Machine Learning Pipeline.

Context

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

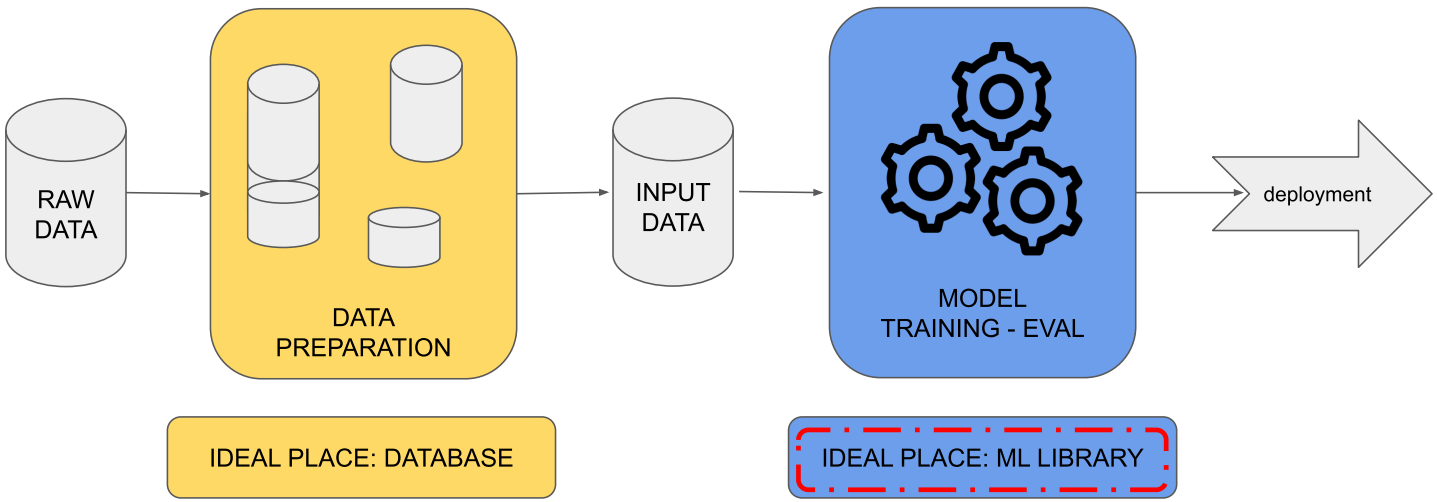


Figure: Classic Machine Learning Pipeline.

Context

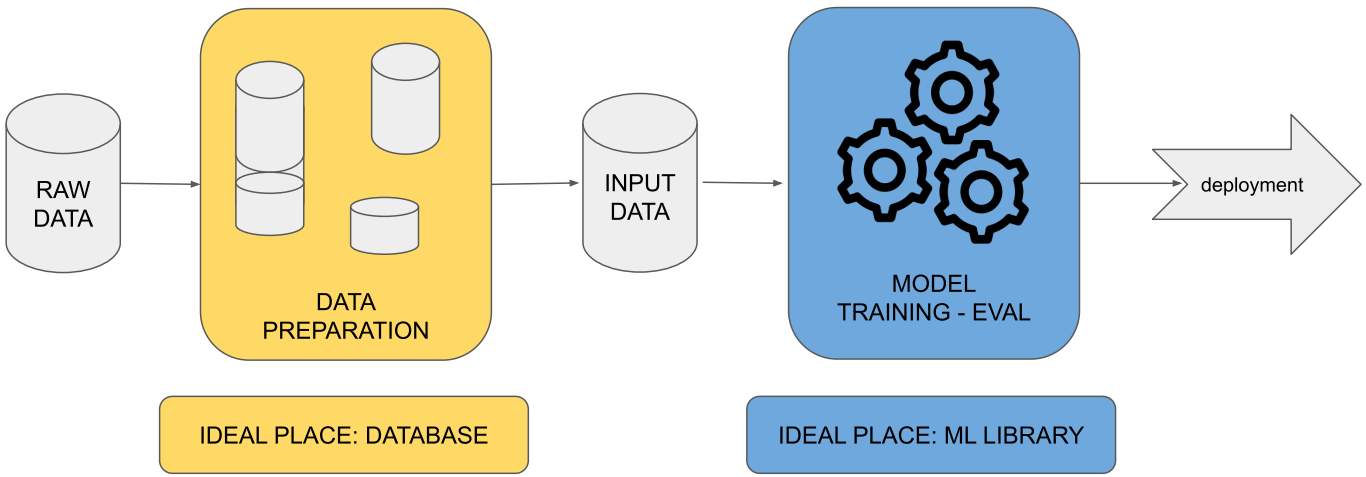


Figure: Classic Machine Learning Pipeline.

Context

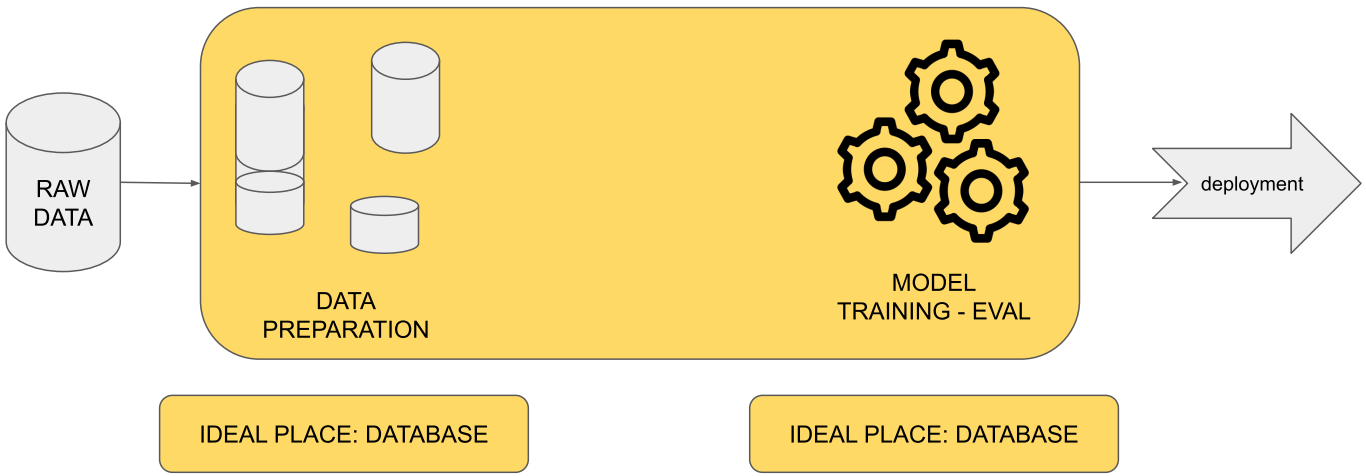


Figure: Proposed Pipeline.

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 di erentiation is missing (Schule 2019)

Context

Di erentiating Relational Queries, Derivative of the Relational Queries

"

This is not di erential data ow (Mcsherry 2021)

Context

Figure: What we are looking for

Outline

Formalization

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

$$x^{?} = \arg\min_{x} \quad f(x)$$

Figure: Gradient Descent, source (Hutson)

f is calledloss

Formalization

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

For that we need:

- a framework
- constraints on the query

Formalization

Minimization is only possible on scalar.

$$\text{Loss} = \sum_{i \in \text{Obs}} \text{loss}_i = \sum_{i \in \text{Obs}} f(\text{data}_i)$$

Formalization

Minimization is only possible on scalar.

$$\text{Loss} = \sum_{i \in \text{Obs}} \text{loss}_i = \sum_{i \in \text{Obs}} f(\text{data}_i)$$

Constraint 1

Loss is computed line by line

Example

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

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.

Linear Regression on the Chicago dataset

Model

$$\text{Tip}_{\text{estimated}} = a_{\text{company}} \cdot \text{distance} + b$$

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

Figure: Model

Approach

Comparing the matrix approach (ML Libraries) and relational one

Approach

$$(M:A) \quad X + b$$

is the point-wise product

Figure: Matrix approach

Approach

Figure: Relational approach

Approach

Figure: Relational approach

Figure: Matrix approach

Linear Regression on the Chicago dataset

Model

$$\text{Tip}_{\text{estimated}} = a_{\text{company}} \cdot \text{distance} + b$$

In SQL it gives

SQL query of our model.

Formalization

$$" \quad \text{Trips} = \text{Observations}$$

$$\text{Loss} = \sum_{t \in \text{Trips}} \text{loss}_t = \sum_{t \in \text{Trips}} f(\text{data}_t) = \sum_{t \in \text{Trips}} (a_{\text{comp}_t} \quad \text{dist}_t + b \quad \text{tip}_t)^2$$

with

$$f(a; x; b; y) = (ax + b \quad y)^2$$

Formalization

$$" \quad \text{Trips} = \text{Observations}$$

$$\text{Loss} = \sum_{t \in \text{Trips}} \text{loss}_t = \sum_{t \in \text{Trips}} f(\text{data}_t) = \sum_{t \in \text{Trips}} (a_{\text{comp}_t} \quad \text{dist}_t + b \quad \text{tip}_t)^2$$

with

$$f(a; x; b; y) = (ax + b \quad y)^2$$

Then it is feasible to compute gradients!

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

Constraint 2

f has to be di erentiable.

Formalization

Figure: Inputs origin.

Formalization

Figure: Inputs origin.

SQL query of our model.

Approach

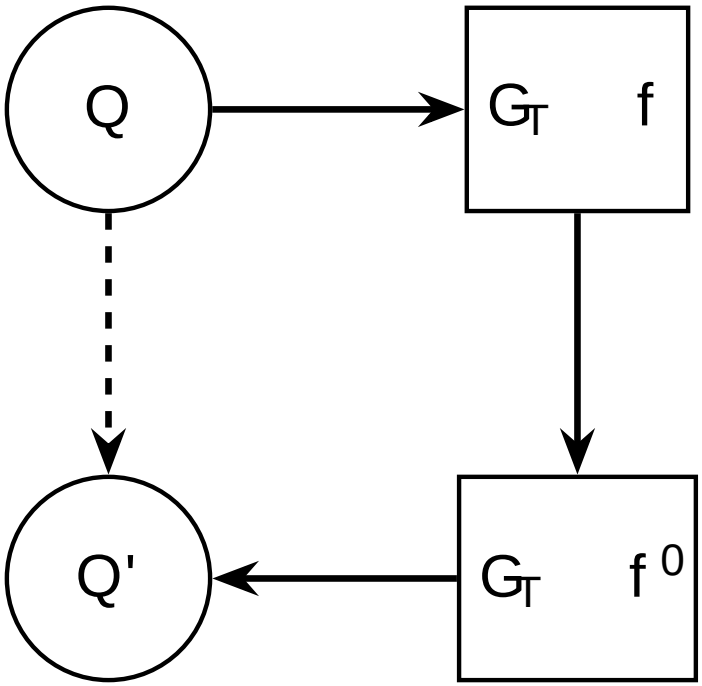


Figure: Path to Di erentiating Relational Queries.

Q : query
G_T : tables graph
f : loss function

Outline

Approach

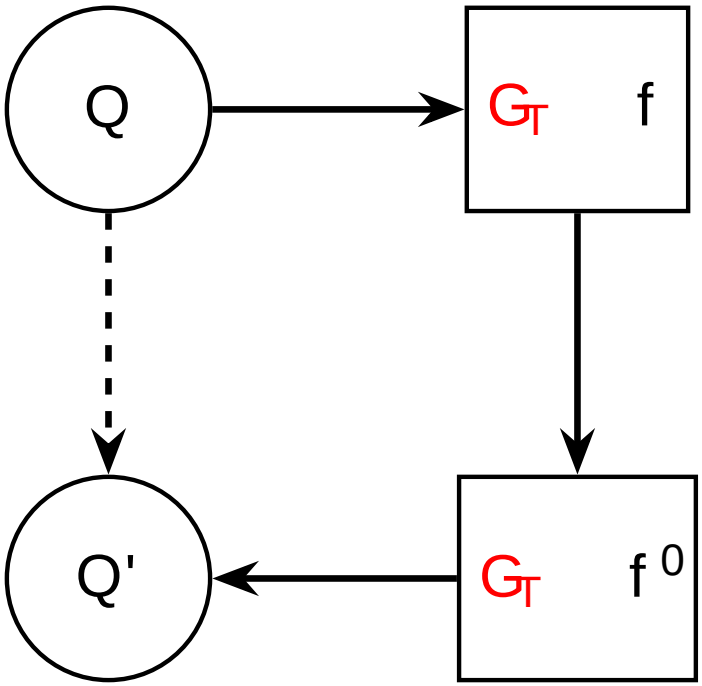


Figure: Path to Di erentiating Relational Queries.

Q : query
G_T : tables graph
f : loss function

Tables Relations

De nition 1 (Broadcast)

Let's note "T_A ! 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

De nition 1 (Broadcast)

Let's note "T_A ! 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 relationshlp forms a graph G_T.

Tables Relations

De nition 1 (Broadcast)

Let's note "T_A ! 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 relationshlp forms a graph G_T.

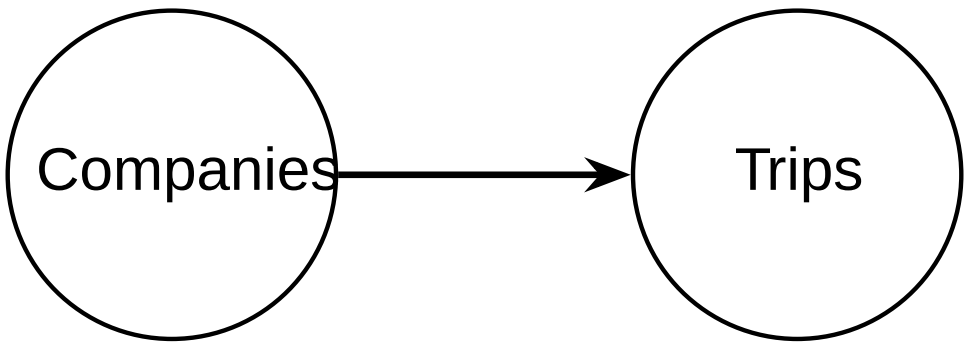


Figure: Graph from our linear regression model.

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

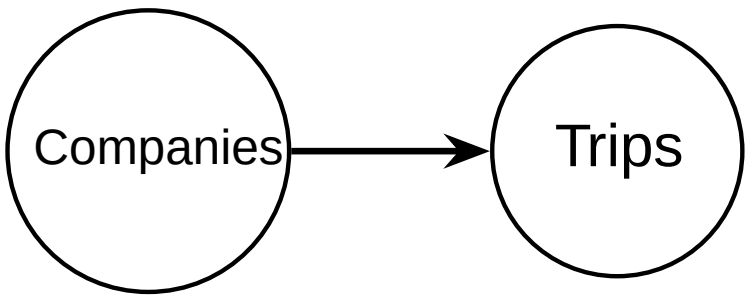
Let be

T a table used in the query

T :A be a column ofT

a the input of f representingT :A

If T (transitively) broadcasts into Observations then a the input of f representingT :A is a scalar.



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

Figure: Graph from our linear regression model.

Approach

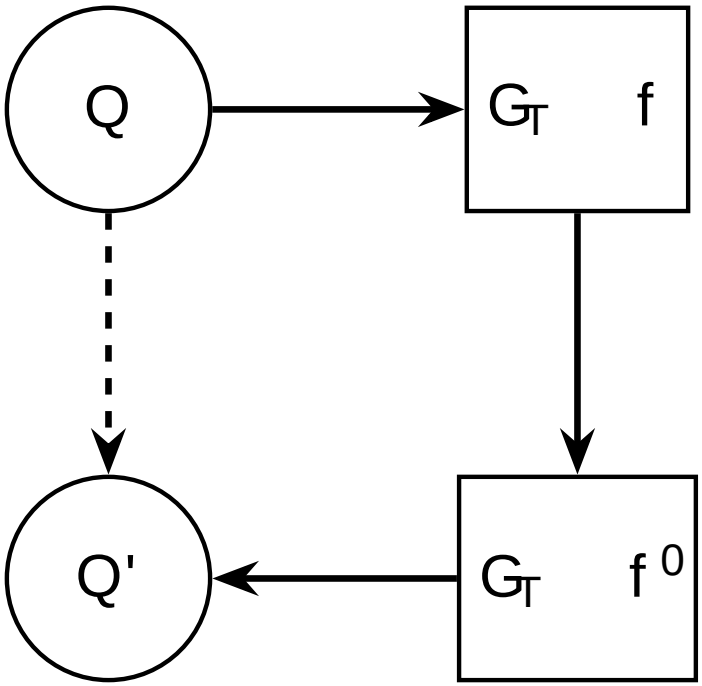


Figure: Path to Di erentiating Relational Queries.

Q : query
G_T : tables graph
f : loss function

Outline

Approach

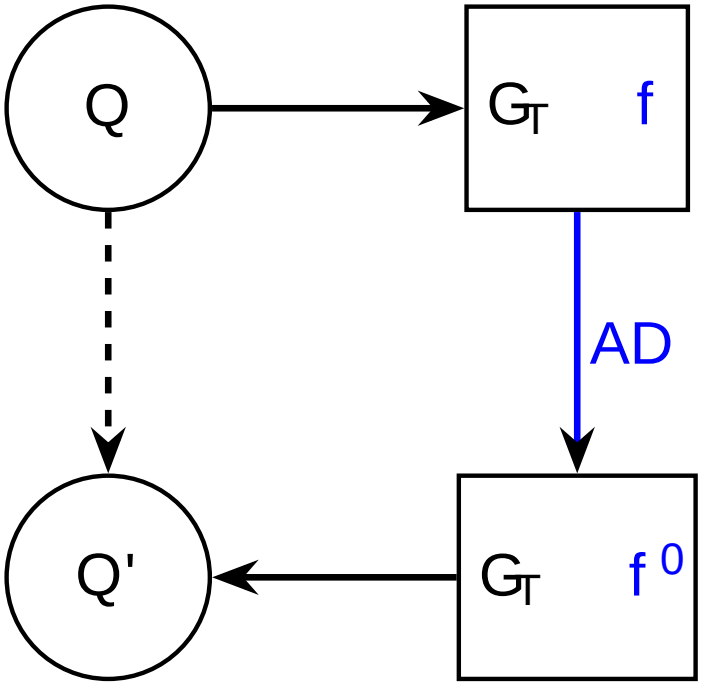


Figure: Path to Di erentiating Relational Queries.

Q : query
G_T : tables graph
f : loss function
AD : Automatic Di erentiation

Automatic Di erentiation

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

Automatic Di erentiation constructs program the program P^0 that apply f^0 to its inputs.

Automatic Di erentiation

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

Automatic Di erentiation constructs program the program P^0 that apply f^0 to its inputs.

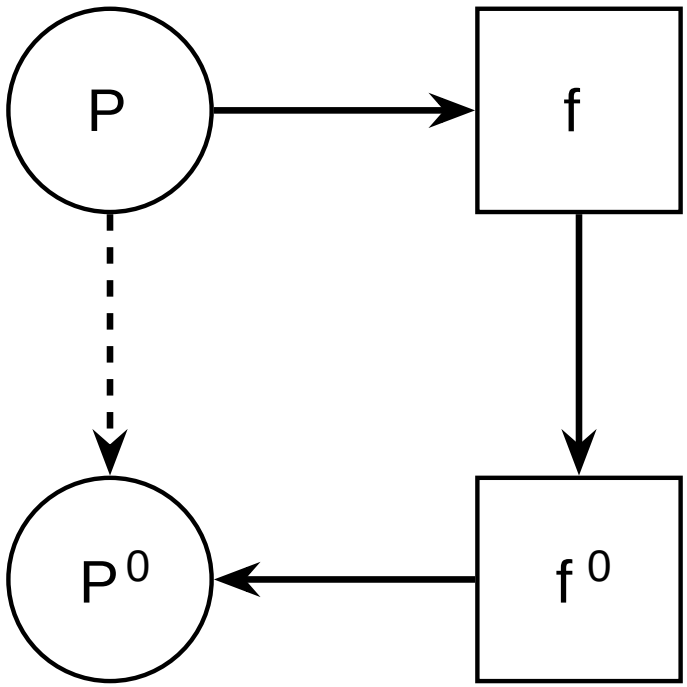


Figure: Automatic Di erentiation.

Automatic Di erentiation

Fortran, C: Tapenade

Python: Tangent, Myia

Julia: Zygote

F#: Di Sharp

...

Automatic Di erentiation

Fortran, C: Tapenade

Python: Tangent, Myia

Julia: Zygote

F#: Di Sharp

...

not di erentiating a speci c programming language.

de ne a narrowed programming language: ADSL. Similar to (Abadi 2019) (Hu 2020) (Mak 2020).

ADSL is closed by di erentiation

Automatic Di erentiation compilation

We can use this pipeline to di erentiate 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 DSLEnvision
- live in production

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

Outline

Conclusion

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

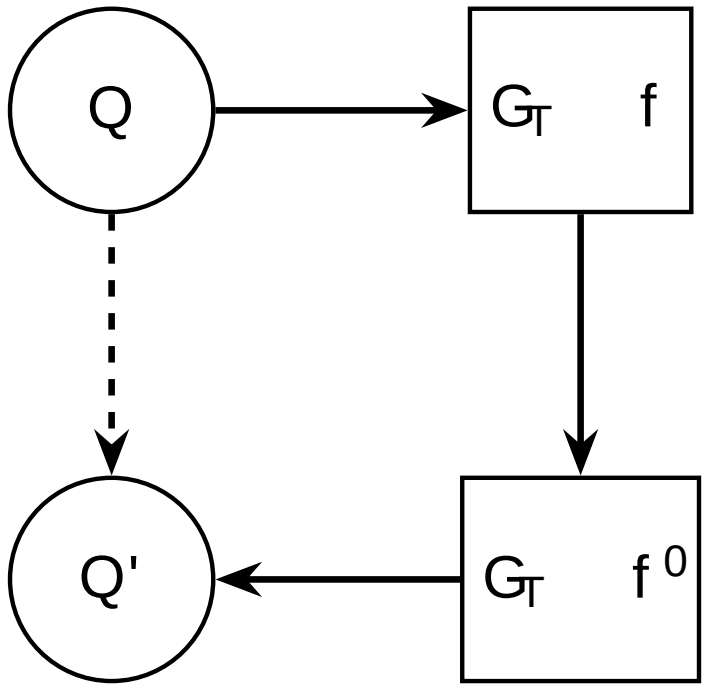


Figure: Path to Di erentiating Relational Queries.

Conclusion

This will unlock ML model construction and optimisation in databases.

Figure: Proposed Pipeline.

Conclusion

Thanks for listening!

References

[Abadi 2019] Martn Abadi et Gordon Plotkin. A simple di erentiable 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. Di Taichi: Di erentiable 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 Di erential-form Pullback Programming Language for Higher-order Reverse-mode Automatic Di erentiation . ArXiv, vol. abs/2002.08241, 2020.

[Mcsherry 2021] Frank Mcsherry, Derek Murray, Rebecca Isaacs et Michael IsardDi erential data ow . 08 2021.

[Schule 2019] Maximilian E. Schule, Frederic Simonis, Thomas Heyenbrock, A. Kemper, Stephan Gunnemann 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

ADSL

$$
\begin{aligned}
\langle\, Pred \,\rangle \;::=\; & . \\
\mid\; & \langle\, And\ v\ w \,\rangle \\
\mid\; & \langle\, Or\ v\ w \,\rangle \\
\mid\; & \langle\, Not\ v \,\rangle \\
\mid\; & \langle\, v < w \,\rangle \\
\mid\; & \langle\, v \leq w \rangle
\end{aligned}
$$

PolyStar

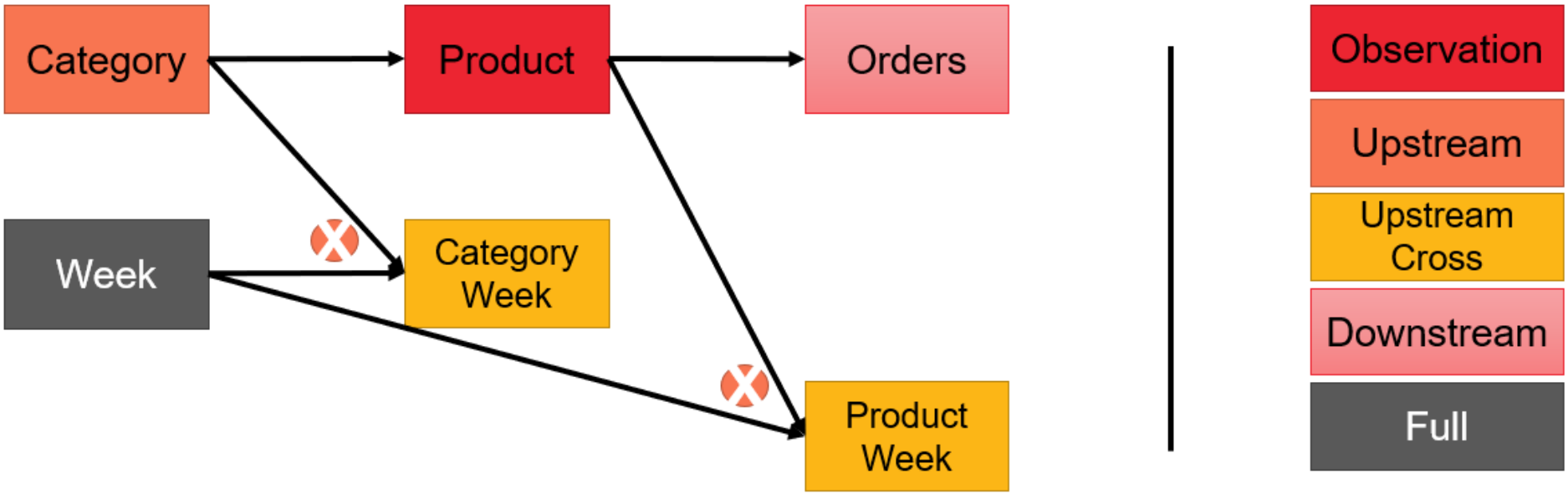


Figure: PolyStar

Realtional versus Math

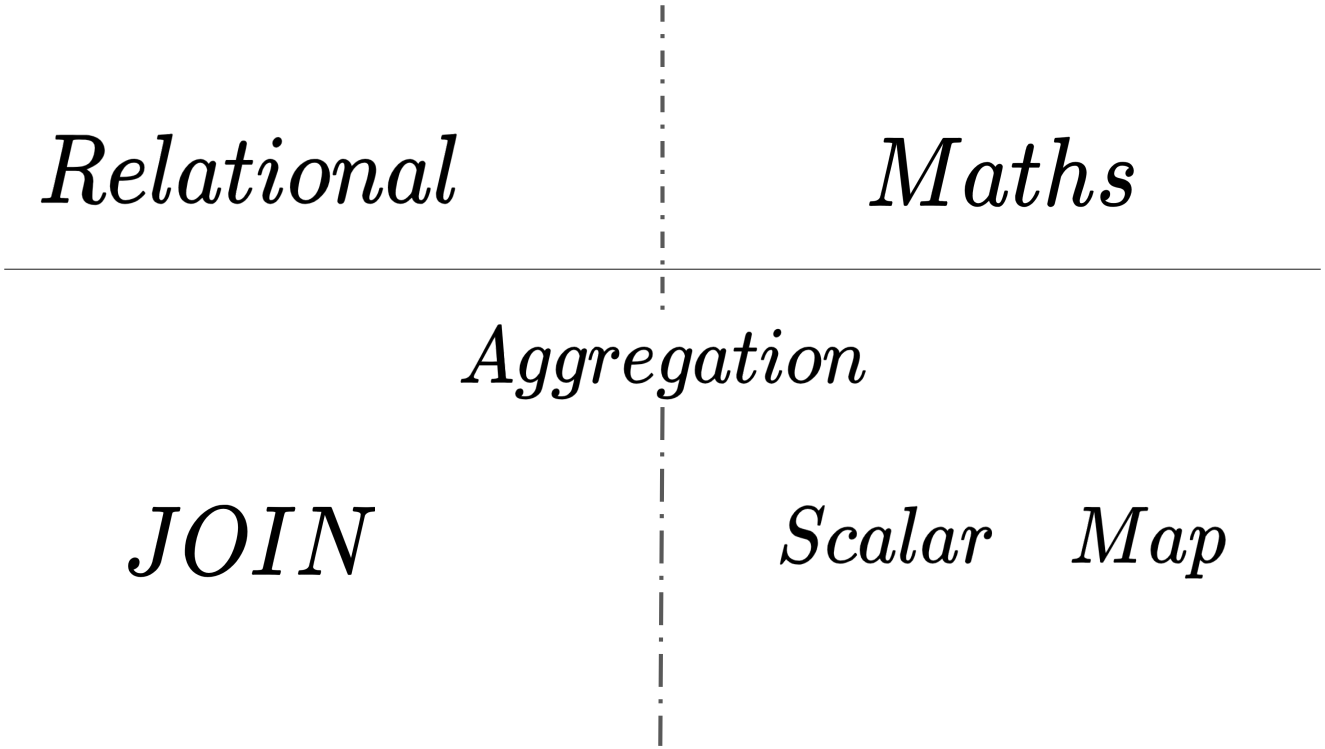


Figure: Realtional - Math decomposition