Relational Algorithms

Mahmoud Abo Khamis

Relational AI

Fine-Grained Complexity, Logic, and Query Evaluation

Simons Institute

Outline

- What are Relational Algorithms?
- Technical Tools
  - Sum-Product Queries
- Example: Relational Linear Regression
- Generalizing the Example?
- Open Problems

What are Relational Algorithms?

Relational Algorithms

Relational Algorithms

- The input is relational data

Relational Algorithms

- The input is relational data
- The algorithms *utilizes* relational structure for
  - *better performance*
  - *better results*

Relational Algorithms

- The input is relational data
- The algorithms *utilizes* relational structure for
  - *better performance*
  - *better results*
- The output can be represented relationally

Example

$$Q(x_1, x_2, x_3, y) \quad = \quad R(x_1, x_2) \quad \wedge \quad S(x_2, x_3) \quad \wedge \quad T(x_3, y)$$

Example

$$Q(x_1, x_2, x_3, y) \quad = \quad R(x_1, x_2) \quad \wedge \quad S(x_2, x_3) \quad \wedge \quad T(x_3, y)$$

x_1	x_2
a	1
a	3
b	1

x_2	x_3
1	e
4	d

x_3	y
d	1.5
e	2.0
e	2.5

Example

$$Q(x_1, x_2, x_3, y) \quad = \quad R(x_1, x_2) \quad \wedge \quad S(x_2, x_3) \quad \wedge \quad T(x_3, y)$$

x_1	x_2	x_3	y
a	1	e	2.0
a	1	e	2.5
b	1	e	2.0
b	1	e	2.5

x_1	x_2
a	1
a	3
b	1

x_2	x_3
1	e
4	d

x_3	y
d	1.5
e	2.0
e	2.5

Example

$$Q(x_1, x_2, x_3, y) \quad = \quad R(x_1, x_2) \quad \wedge \quad S(x_2, x_3) \quad \wedge \quad T(x_3, y)$$

x_1	x_2	x_3	y
a	1	e	2.0
a	1	e	2.5
b	1	e	2.0
b	1	e	2.5

x_1	x_2
a	1
a	3
b	1

x_2	x_3
1	e
4	d

x_3	y
d	1.5
e	2.0
e	2.5

Input

Example

$$Q(x_1, x_2, x_3, y) \quad = \quad R(x_1, x_2) \quad \wedge \quad S(x_2, x_3) \quad \wedge \quad T(x_3, y)$$

x_1	x_2	x_3	y
a	1	e	2.0
a	1	e	2.5
b	1	e	2.0
b	1	e	2.5

x_1	x_2
a	1
a	3
b	1

x_2	x_3
1	e
4	d

x_3	y
d	1.5
e	2.0
e	2.5

Input

Goal: Compute *something* that depends on Q

Low-level Computation: Aggregation

$$Q(x_1, x_2, x_3, y) \quad = \quad R(x_1, x_2) \quad \wedge \quad S(x_2, x_3) \quad \wedge \quad T(x_3, y)$$

$$s = \sum_{(x_1, x_2, x_3, y) \in Q} y^2$$

Low-level Computation: Aggregation

$$Q(x_1, x_2, x_3, y) \quad = \quad R(x_1, x_2) \quad \wedge \quad S(x_2, x_3) \quad \wedge \quad T(x_3, y)$$

$$s = \sum_{(x_1, x_2, x_3, y) \in Q} y^2$$

- Non-relational Algorithm
  - Compute $Q(x_1, x_2, x_3, y)$ then sum y^2
  - Time: $\tilde{O}(|R| \cdot |T|)$

Low-level Computation: Aggregation

$$Q(x_1, x_2, x_3, y) \quad = \quad R(x_1, x_2) \quad \wedge \quad S(x_2, x_3) \quad \wedge \quad T(x_3, y)$$

$$s = \sum_{(x_1,x_2,x_3,y)\in Q} y^2$$

- Non-relational Algorithm
  - Compute $Q(x_1, x_2, x_3, y)$ then sum y^2
  - Time: $\tilde{O}(|R| \cdot |T|)$
- Relational Algorithm
  - *Push* the sum *inside* the query
    - $$s = \sum_{(x_1,x_2)\in R} \left(\sum_{x_3:(x_2,x_3)\in S} \left(\sum_{y:(x_3,y)\in T} y^2 \right) \right)$$
  - Time: $\tilde{O}(|R| + |S| + |T|)$

High-level Computation: ML Training

$$Q(x_1, x_2, x_3, y) \quad = \quad R(x_1, x_2) \quad \wedge \quad S(x_2, x_3) \quad \wedge \quad T(x_3, y)$$

Train some ML model on $Q(x_1, x_2, x_3, y)$

to predict y from (x_1, x_2, x_3)

High-level Computation: ML Training

$$Q(x_1, x_2, x_3, y) \quad = \quad R(x_1, x_2) \quad \wedge \quad S(x_2, x_3) \quad \wedge \quad T(x_3, y)$$

Train some ML model on $Q(x_1, x_2, x_3, y)$

to predict y from (x_1, x_2, x_3)

- Non-relational Algorithm
  - Materialize Q (a big matrix)
  - Run ML

High-level Computation: ML Training

$$Q(x_1, x_2, x_3, y) \quad = \quad R(x_1, x_2) \quad \wedge \quad S(x_2, x_3) \quad \wedge \quad T(x_3, y)$$

Train some ML model on $Q(x_1, x_2, x_3, y)$

to predict y from (x_1, x_2, x_3)

- Non-relational Algorithm
  - Materialize Q (a big matrix)
  - Run ML
- Relational Algorithm
  - Operate directly on (small) R, S, T
  - *Push* ML *inside* the query?

Technical Tools

Sum-Product Queries

Sum-Product Queries

- A technical tool to *describe* and *develop* (some) relational algorithms

Sum-Product Queries

- A technical tool to *describe* and *develop* (some) relational algorithms
  - Relational Algorithm $\Rightarrow$ Solving SPQs

Sum-Product Queries

- A technical tool to *describe* and *develop* (some) relational algorithms
  - Relational Algorithm $\Rightarrow$ Solving SPQs
- A universal language to express many problems
  - Databases
  - Linear Algebra
  - CSP
  - Logic
  - PGMs
  - ...

Sum-Product Queries

- Input

Sum-Product Queries

- Input
  - A commutative semiring $(\boldsymbol{D}, \oplus, \otimes, \mathbf{0}, \mathbf{1})$
    - $\mathbf{0} \oplus a = a$
    - $\mathbf{1} \otimes a = a$
    - $\mathbf{0} \otimes a = \mathbf{0}$
    - $a \otimes (b \oplus c) = (a \otimes b) \oplus (a \otimes c)$

Sum-Product Queries

- Input
  - A commutative semiring $(\boldsymbol{D}, \oplus, \otimes, \mathbf{0}, \mathbf{1})$
    - $\mathbf{0} \oplus a = a$
    - $\mathbf{1} \otimes a = a$
    - $\mathbf{0} \otimes a = \mathbf{0}$
    - $a \otimes (b \oplus c) = (a \otimes b) \oplus (a \otimes c)$
  - A variable set X

Sum-Product Queries

- Input
  - A commutative semiring $(\boldsymbol{D}, \oplus, \otimes, \mathbf{0}, \mathbf{1})$
    - $\mathbf{0} \oplus a = a$
    - $\mathbf{1} \otimes a = a$
    - $\mathbf{0} \otimes a = \mathbf{0}$
    - $a \otimes (b \oplus c) = (a \otimes b) \oplus (a \otimes c)$
  - A variable set X
  - Subsets $X_1, \ldots, X_m \subseteq X$

Sum-Product Queries

- Input
  - A commutative semiring $(\boldsymbol{D}, \oplus, \otimes, \mathbf{0}, \mathbf{1})$
    - $\mathbf{0} \oplus a = a$
    - $\mathbf{1} \otimes a = a$
    - $\mathbf{0} \otimes a = \mathbf{0}$
    - $a \otimes (b \oplus c) = (a \otimes b) \oplus (a \otimes c)$
  - A variable set X
  - Subsets $X_1, \ldots, X_m \subseteq X$
  - Functions $\psi_1(X_1), \ldots, \psi_m(X_m)$ to $\boldsymbol{D}$

Sum-Product Queries

- Input
  - A commutative semiring $(\boldsymbol{D}, \oplus, \otimes, \mathbf{0}, \mathbf{1})$
    - $\mathbf{0} \oplus a = a$
    - $\mathbf{1} \otimes a = a$
    - $\mathbf{0} \otimes a = \mathbf{0}$
    - $a \otimes (b \oplus c) = (a \otimes b) \oplus (a \otimes c)$
  - A variable set X
  - Subsets $X_1, \ldots, X_m \subseteq X$
  - Functions $\psi_1(X_1), \ldots, \psi_m(X_m)$ to $\boldsymbol{D}$
  - A subset $Y \subseteq X$

Sum-Product Queries

- Input
  - A commutative semiring $(\boldsymbol{D}, \oplus, \otimes, \mathbf{0}, \mathbf{1})$
    - $\mathbf{0} \oplus a = a$
    - $\mathbf{1} \otimes a = a$
    - $\mathbf{0} \otimes a = \mathbf{0}$
    - $a \otimes (b \oplus c) = (a \otimes b) \oplus (a \otimes c)$
  - A variable set X
  - Subsets $X_1, \ldots, X_m \subseteq X$
  - Functions $\psi_1(X_1), \ldots, \psi_m(X_m)$ to $\boldsymbol{D}$
  - A subset $Y \subseteq X$
- Goal: Compute
  - $$\varphi(Y) := \bigoplus_{X-Y} \psi_1(X_1) \otimes \ldots \otimes \psi_m(X_m)$$

Examples

$$\varphi(a,d) \quad = \quad \bigoplus_{b,c} \psi_1(a,b) \otimes \psi_2(b,c) \otimes \psi_3(c,d)$$

Examples

$$\varphi(a, d) \quad = \quad \bigoplus_{b,c} \psi_1(a, b) \otimes \psi_2(b, c) \otimes \psi_3(c, d)$$

- Is there a 3-path from a to d?
  - $Q_1(a, d) \quad = \quad \bigvee_{b,c} E(a, b) \wedge E(b, c) \wedge E(c, d)$

Examples

$$\varphi(a,d) \quad = \quad \bigoplus_{b,c} \psi_1(a,b) \otimes \psi_2(b,c) \otimes \psi_3(c,d)$$

- Is there a 3-path from a to d?
  - $Q_1(a,d) \quad = \quad \bigvee_{b,c} E(a,b) \wedge E(b,c) \wedge E(c,d)$
- How many 3-paths?
  - $Q_2(a,d) \quad = \quad \sum_{b,c} 1_E(a,b) \times 1_E(b,c) \times 1_E(c,d)$

Examples

$$\varphi(a,d) \quad = \quad \bigoplus_{b,c} \psi_1(a,b) \otimes \psi_2(b,c) \otimes \psi_3(c,d)$$

- Is there a 3-path from a to d?
  - $Q_1(a,d) \quad = \quad \bigvee_{b,c} E(a,b) \wedge E(b,c) \wedge E(c,d)$
- How many 3-paths?
  - $Q_2(a,d) \quad = \quad \sum_{b,c} 1_E(a,b) \times 1_E(b,c) \times 1_E(c,d)$
- Shortest 3-path?
  - $Q_3(a,d) \quad = \quad \min_{b,c} W(a,b) + W(b,c) + W(c,d)$

Examples

$$\varphi(a,d) \quad = \quad \bigoplus_{b,c} \psi_1(a,b) \otimes \psi_2(b,c) \otimes \psi_3(c,d)$$

- Is there a 3-path from a to d?
  - $Q_1(a,d) \quad = \quad \bigvee_{b,c} E(a,b) \wedge E(b,c) \wedge E(c,d)$
- How many 3-paths?
  - $Q_2(a,d) \quad = \quad \sum_{b,c} 1_E(a,b) \times 1_E(b,c) \times 1_E(c,d)$
- Shortest 3-path?
  - $Q_3(a,d) \quad = \quad \min_{b,c} W(a,b) + W(b,c) + W(c,d)$
- Matrix multiplication
  - $M(a,d) \quad = \quad \sum_{b,c} M_1(a,b) \times M_2(b,c) \times M_3(c,d)$

Examples

$$Q(x_1, x_2, x_3, x_4) = R(x_1, x_2) \wedge S(x_2, x_3) \wedge T(x_3, x_4)$$

Examples

$$Q(x_1, x_2, x_3, x_4) = R(x_1, x_2) \wedge S(x_2, x_3) \wedge T(x_3, x_4)$$

Compute minimum ℓ_1-norm among $(x_1, x_2, x_3, x_4) \in Q$

Examples

$$Q(x_1, x_2, x_3, x_4) = R(x_1, x_2) \wedge S(x_2, x_3) \wedge T(x_3, x_4)$$

Compute minimum ℓ_1-norm among $(x_1, x_2, x_3, x_4) \in Q$

$$\varphi = \min_{x_1, x_2, x_3, x_4} \psi_R(x_1, x_2) + \psi_S(x_2, x_3) + \psi_T(x_3, x_4)$$

Examples

$$Q(x_1, x_2, x_3, x_4) = R(x_1, x_2) \wedge S(x_2, x_3) \wedge T(x_3, x_4)$$

Compute minimum ℓ_1-norm among $(x_1, x_2, x_3, x_4) \in Q$

$$\varphi = \min_{x_1, x_2, x_3, x_4} \psi_R(x_1, x_2) + \psi_S(x_2, x_3) + \psi_T(x_3, x_4)$$

- $\psi_R(x_1, x_2) := \begin{cases} |x_1| + |x_2| & \text{if } (x_1, x_2) \in R \\ \infty & \text{otherwise} \end{cases}$

Examples

$$Q(x_1, x_2, x_3, x_4) = R(x_1, x_2) \wedge S(x_2, x_3) \wedge T(x_3, x_4)$$

Compute minimum ℓ_1-norm among $(x_1, x_2, x_3, x_4) \in Q$

$$\varphi = \min_{x_1, x_2, x_3, x_4} \psi_R(x_1, x_2) + \psi_S(x_2, x_3) + \psi_T(x_3, x_4)$$

- $\psi_R(x_1, x_2) := \begin{cases} |x_1| + |x_2| & \text{if } (x_1, x_2) \in R \\ \infty & \text{otherwise} \end{cases}$
- $\psi_S(x_2, x_3) := \begin{cases} |x_3| & \text{if } (x_2, x_3) \in S \\ \infty & \text{otherwise} \end{cases}$

Examples

$$Q(x_1, x_2, x_3, x_4) = R(x_1, x_2) \wedge S(x_2, x_3) \wedge T(x_3, x_4)$$

Compute minimum ℓ_1-norm among $(x_1, x_2, x_3, x_4) \in Q$

$$\varphi = \min_{x_1, x_2, x_3, x_4} \psi_R(x_1, x_2) + \psi_S(x_2, x_3) + \psi_T(x_3, x_4)$$

- $\psi_R(x_1, x_2) := \begin{cases} |x_1| + |x_2| & \text{if } (x_1, x_2) \in R \\ \infty & \text{otherwise} \end{cases}$
- $\psi_S(x_2, x_3) := \begin{cases} |x_3| & \text{if } (x_2, x_3) \in S \\ \infty & \text{otherwise} \end{cases}$
- $\psi_T(x_3, x_4) := \begin{cases} |x_4| & \text{if } (x_3, x_4) \in T \\ \infty & \text{otherwise} \end{cases}$

Solving SPQs using Join Algorithms

$$\varphi(x) = \bigoplus_{y,z} \psi_1(x, y) \otimes \psi_2(y, z) \otimes \psi_3(x, z)$$

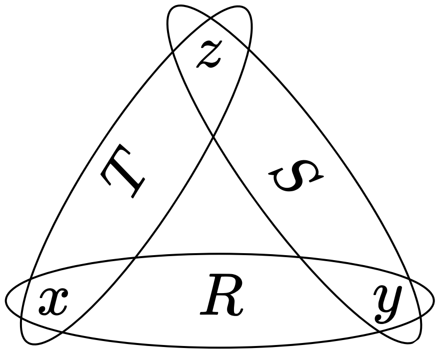
Solving SPQs using Join Algorithms

$$\varphi(x) = \bigoplus_{y,z} \psi_1(x,y) \otimes \psi_2(y,z) \otimes \psi_3(x,z)$$

$$\Downarrow$$

$$Q(x,y,z) := R(x,y) \wedge S(y,z) \wedge T(x,z)$$

$$R := \{(x,y) \mid \psi_1(x,y) \neq \mathbf{0}\}$$



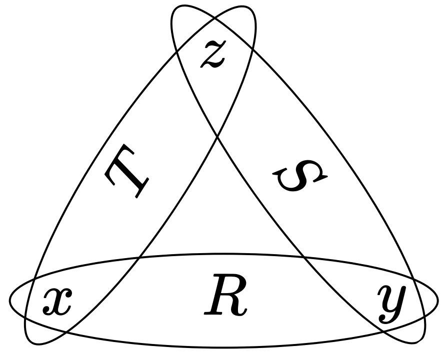
Solving SPQs using Join Algorithms

$$\varphi(x) = \bigoplus_{y,z} \psi_1(x,y) \otimes \psi_2(y,z) \otimes \psi_3(x,z)$$

$$\Downarrow$$

$$Q(x,y,z) := R(x,y) \wedge S(y,z) \wedge T(x,z)$$

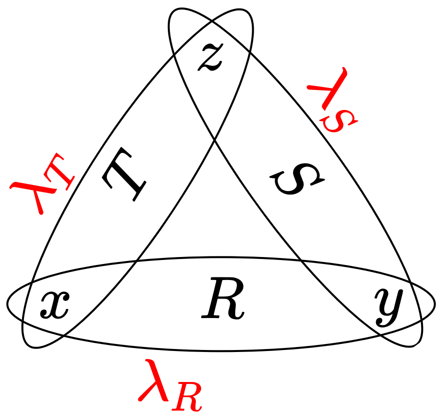
$$R := \{(x,y) \mid \psi_1(x,y) \neq \mathbf{0}\}$$



- Use *Worst-case Optimal Joins (WCOJ)* or beyond
  - See [Logic&Algo-in-DB&AI BootCamp]

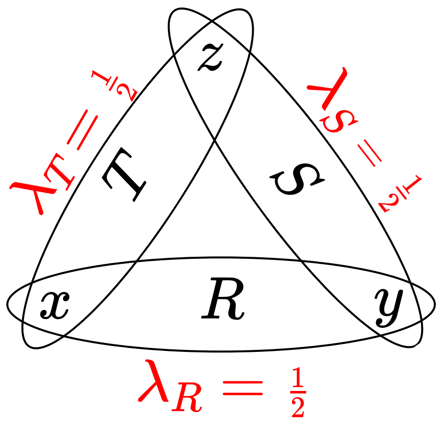
WCOJ Runtime

- $(\lambda_R, \lambda_S, \lambda_T)$ is a *fractional edge cover*



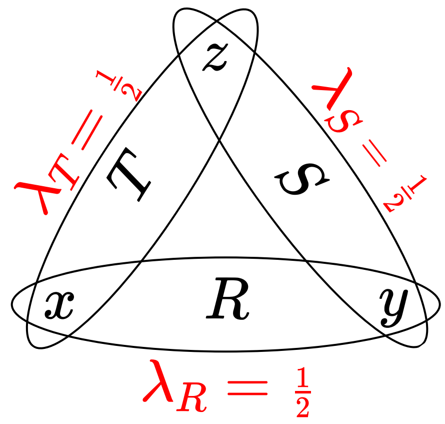
WCOJ Runtime

- $(\lambda_R, \lambda_S, \lambda_T)$ is a *fractional edge cover*



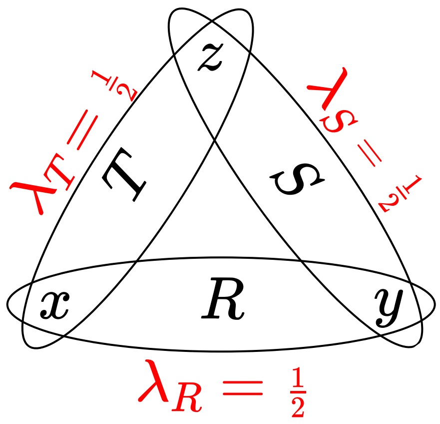
WCOJ Runtime

- $(\lambda_R, \lambda_S, \lambda_T)$ is a *fractional edge cover*
- ρ^* is the minimum $\lambda_R + \lambda_S + \lambda_T$



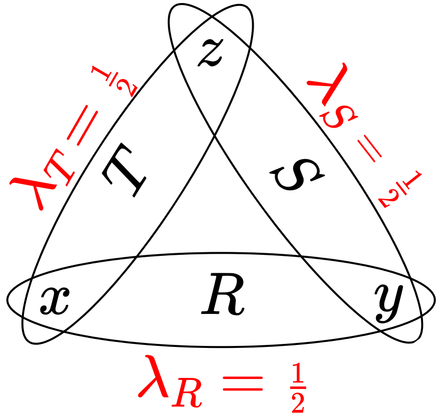
WCOJ Runtime

- $(\lambda_R, \lambda_S, \lambda_T)$ is a *fractional edge cover*
- ρ^* is the minimum $\lambda_R + \lambda_S + \lambda_T$
- $\Rightarrow$
  - $|Q| \leq N^{\rho^*}$ $\qquad$ (AGM-bound)
  - Time = $\tilde{O}(N^{\rho^*})$



WCOJ Runtime

- $(\lambda_R, \lambda_S, \lambda_T)$ is a *fractional edge cover*
- ρ^* is the minimum $\lambda_R + \lambda_S + \lambda_T$
- $\Rightarrow$
  - $|Q| \leq N^{\rho^*}$ (AGM-bound)
  - Time = $\tilde{O}(N^{\rho^*})$



Examples:

- k-clique: $\rho^* = k/2$
- k-path: $\rho^* = \lceil k/2 \rceil$

Solving SPQs using Variable Elimination

$$\varphi = \bigoplus_{x,y,z,t} \psi_1(x, y) \otimes \psi_2(y, z) \otimes \psi_3(z, t)$$

Solving SPQs using Variable Elimination

$$\varphi = \bigoplus_{x,y,z,t} \psi_1(x,y) \otimes \psi_2(y,z) \otimes \psi_3(z,t)$$

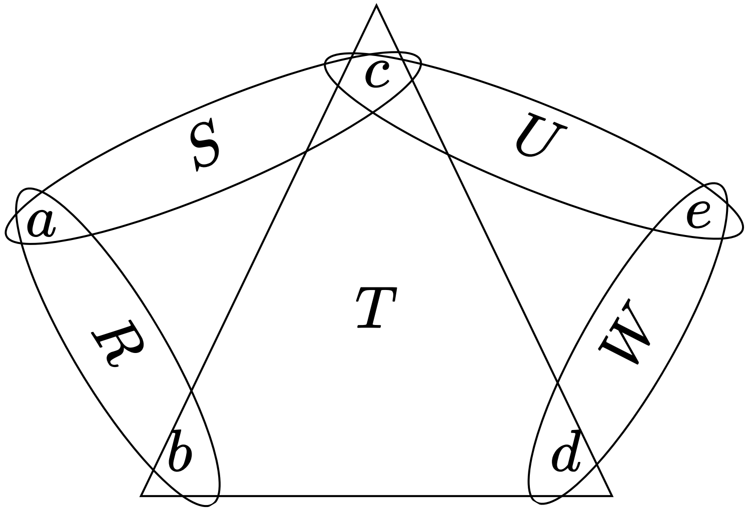
- Eliminate t
  - $\varphi = \bigoplus_{x,y,z} \psi_1(x,y) \otimes \psi_2(y,z) \otimes \underbrace{\bigoplus_{t} \psi_3(z,t)}_{\psi_t(z)}$
  - $\varphi = \bigoplus_{x,y,z} \psi_1(x,y) \otimes \psi_2(y,z) \otimes \psi_t(z)$

Solving SPQs using Variable Elimination

$$\varphi = \bigoplus_{x,y,z,t} \psi_1(x,y) \otimes \psi_2(y,z) \otimes \psi_3(z,t)$$

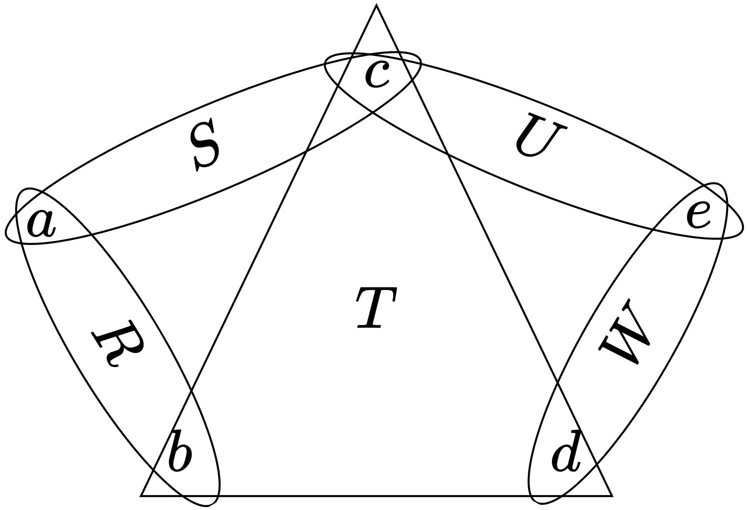
- Eliminate t
  - $\varphi = \bigoplus_{x,y,z} \psi_1(x,y) \otimes \psi_2(y,z) \otimes \underbrace{\bigoplus_t \psi_3(z,t)}_{\psi_t(z)}$
  - $\varphi = \bigoplus_{x,y,z} \psi_1(x,y) \otimes \psi_2(y,z) \otimes \psi_t(z)$
- Eliminate z
  - $\varphi = \bigoplus_{x,y} \psi_1(x,y) \otimes \underbrace{\bigoplus_z \psi_2(y,z) \otimes \psi_t(z)}_{\psi_z(y)}$
  - $\varphi = \bigoplus_{x,y} \psi_1(x,y) \otimes \psi_z(y)$

Solving SPQs using Variable Elimination



$$\varphi(a) = \bigoplus_{b,c,d,e} R(a,b) \otimes S(a,c) \otimes T(b,c,d) \otimes U(c,e) \otimes W(d,e)$$

Solving SPQs using Variable Elimination

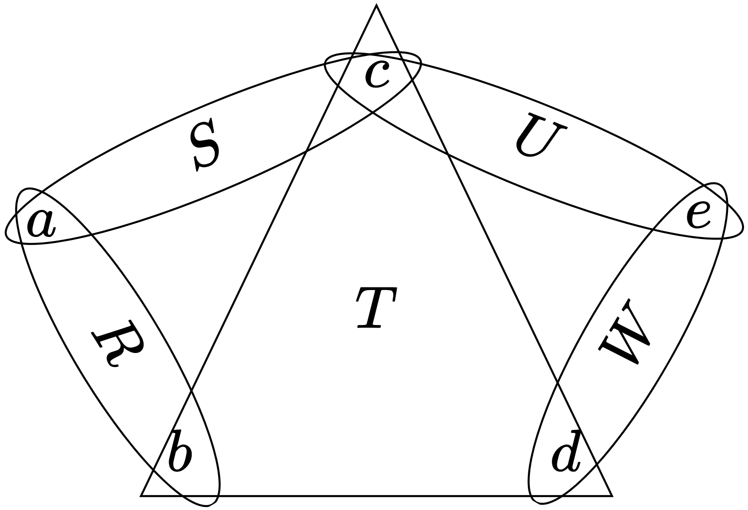


$$\varphi(a) = \bigoplus_{b,c,d,e} R(a,b) \otimes S(a,c) \otimes T(b,c,d) \otimes U(c,e) \otimes W(d,e)$$

- Eliminate d

$$= \bigoplus_{b,c,e} R(a,b) \otimes S(a,c) \otimes U(c,e) \otimes \underbrace{\bigoplus_{d} T(b,c,d) \otimes W(d,e)}_{\substack{\psi_d(b,c,e) \\ \tilde{O}(N^2)}}$$

Solving SPQs using Variable Elimination



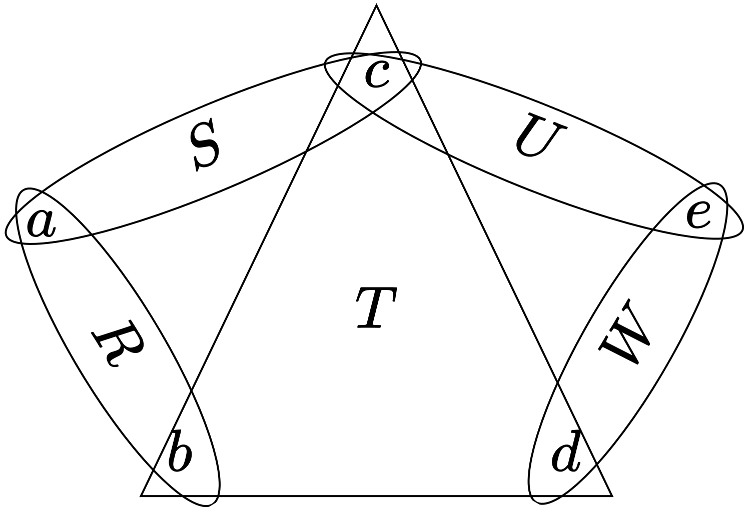
$$\varphi(a) = \bigoplus_{b,c,d,e} R(a,b) \otimes S(a,c) \otimes T(b,c,d) \otimes U(c,e) \otimes W(d,e)$$

- Eliminate e

$$= \bigoplus_{b,c,d} R(a,b) \otimes S(a,c) \otimes T(b,c,d) \otimes \underbrace{\bigoplus_{e} U(c,e) \otimes W(d,e)}_{\psi_e(c,d)}$$

$\tilde{O}(N^2)$

Solving SPQs using Variable Elimination



$$\varphi(a) = \bigoplus_{b,c,d,e} R(a,b) \otimes S(a,c) \otimes T(b,c,d) \otimes U(c,e) \otimes W(d,e)$$

- Eliminate e smarter

$$= \bigoplus_{b,c,d} R(a,b) \otimes S(a,c) \otimes T(b,c,d) \otimes \underbrace{\bigoplus_{e} U(c,e) \otimes W(d,e) \otimes T'(c,d)}_{\psi'_e(c,d)}$$

$$T'(c,d) := 1_{\exists b \,:\, T(b,c,d) \neq \mathbf{0}}$$

$\tilde{O}(N^{1.5})$

SPQ Width

$$\mathsf{spqw} := \min_{\substack{\sigma:\text{var} \\ \text{elimination} \\ \text{order}}} \max_{v \in \sigma} \rho^*(\underbrace{\psi_v^\sigma}_{\substack{\text{SPQ to} \\ \text{eliminate } v}})$$

SPQ Width

$$\text{spqw} := \min_{\substack{\sigma:\text{var}\\ \text{elimination}\\ \text{order}}} \max_{v\in\sigma} \rho^*(\underbrace{\psi_v^\sigma}_{\substack{\text{SPQ to}\\ \text{eliminate } v}})$$

- Special cases
  - $\text{spqw} = 1$ for *acyclic* queries w/o free vars
  - $\text{spqw} = \text{fhtw}$ for queries w/o free vars
    - fhtw = fractional hypertree width
    - See Logic&Algo-in-DB&AI BootCamp

SPQ Width

Examples

- $Q_1 = \sum_{x,y,z,w} R(x,y) \times S(y,z) \times T(z,w)$

- $Q_2(y,z) = \sum_{x,w} R(x,y) \times S(y,z) \times T(z,w)$

- $Q_3(x,z) = \sum_{y} R(x,y) \times S(y,z) \qquad \text{(Sparse MM)}$

- $Q_4(x,w) = \sum_{y,z} R(x,y) \times S(y,z) \times T(z,w)$

SPQ Width

Examples

- $Q_1 = \sum_{x,y,z,w} R(x,y) \times S(y,z) \times T(z,w)$
  - spqw $= 1$
- $Q_2(y,z) = \sum_{x,w} R(x,y) \times S(y,z) \times T(z,w)$
- $Q_3(x,z) = \sum_{y} R(x,y) \times S(y,z)$ (Sparse MM)
- $Q_4(x,w) = \sum_{y,z} R(x,y) \times S(y,z) \times T(z,w)$

SPQ Width

Examples

- $Q_1 = \sum_{x,y,z,w} R(x, y) \times S(y, z) \times T(z, w)$
  - spqw $= 1$
- $Q_2(y, z) = \sum_{x,w} R(x, y) \times S(y, z) \times T(z, w)$
  - spqw $= 1$
- $Q_3(x, z) = \sum_{y} R(x, y) \times S(y, z)$ (Sparse MM)
- $Q_4(x, w) = \sum_{y,z} R(x, y) \times S(y, z) \times T(z, w)$

SPQ Width

Examples

- $Q_1 = \sum_{x,y,z,w} R(x,y) \times S(y,z) \times T(z,w)$
  - spqw $= 1$
- $Q_2(y,z) = \sum_{x,w} R(x,y) \times S(y,z) \times T(z,w)$
  - spqw $= 1$
- $Q_3(x,z) = \sum_{y} R(x,y) \times S(y,z)$ (Sparse MM)
  - spqw $= 2$
- $Q_4(x,w) = \sum_{y,z} R(x,y) \times S(y,z) \times T(z,w)$

SPQ Width

Examples

- $Q_1 = \sum_{x,y,z,w} R(x,y) \times S(y,z) \times T(z,w)$
  - spqw $= 1$
- $Q_2(y,z) = \sum_{x,w} R(x,y) \times S(y,z) \times T(z,w)$
  - spqw $= 1$
- $Q_3(x,z) = \sum_{y} R(x,y) \times S(y,z)$ (Sparse MM)
  - spqw $= 2$
- $Q_4(x,w) = \sum_{y,z} R(x,y) \times S(y,z) \times T(z,w)$
  - spqw $= 2$

Example

Relational Linear Regression

Some References

- *"Learning Models over Relational Data using Sparse Tensors and Functional Dependencies"*, **TODS'20**
  - Abo Khamis, Ngo, Nguyen, Olteanu, Schleich
- *"A Layered Aggregate Engine for Analytics Workloads"*, **SIGMOD'19**
  - Schleich, Olteanu, Abo Khamis, Ngo, Nguyen
- *"Learning Generalized Linear Models Over Normalized Data"*, **SIGMOD'15**
  - Kumar, Naughton, Patel

FactorizedDB (Zurich)

RelationalAI Research

Relational Linear Regression

Relational Linear Regression

- Input: Feature Extraction Query (FEQ)

$$\begin{aligned}&\mathrm{Q}(\mathrm{item}, \mathrm{price}, \mathrm{city}, \mathrm{country} \to \mathrm{sales}) = \\ &\qquad \mathrm{S}(\mathrm{item}, \mathrm{city} \to \mathrm{sales}) \wedge \mathrm{P}(\mathrm{item}, \mathrm{price}) \wedge \mathrm{C}(\mathrm{city} \to \mathrm{country})\end{aligned}$$

Relational Linear Regression

- Input: Feature Extraction Query (FEQ)

$$\mathrm{Q}(\mathrm{item}, \mathrm{price}, \mathrm{city}, \mathrm{country} \to \mathrm{sales}) =$$
$$\mathrm{S}(\mathrm{item}, \mathrm{city} \to \mathrm{sales}) \wedge \mathrm{P}(\mathrm{item}, \mathrm{price}) \wedge \mathrm{C}(\mathrm{city} \to \mathrm{country})$$

item	city	sales
coat	Oxford	1000
...	...	...

item	price
coat	50$
...	...

city	country
Oxford	UK
...	...

Relational Linear Regression

- Input: Feature Extraction Query (FEQ)

$$\mathrm{Q}(\mathrm{item}, \mathrm{price}, \mathrm{city}, \mathrm{country} \to \mathrm{sales}) =$$
$$\mathrm{S}(\mathrm{item}, \mathrm{city} \to \mathrm{sales}) \wedge \mathrm{P}(\mathrm{item}, \mathrm{price}) \wedge \mathrm{C}(\mathrm{city} \to \mathrm{country})$$

item	city	sales
coat	Oxford	1000
...	...	...

item	price
coat	50$
...	...

city	country
Oxford	UK
...	...

- Goal
  - Learn to predict sales from $(\mathrm{item}, \mathrm{price}, \mathrm{city}, \mathrm{country})$
    - using Linear Regression with Square Loss

Traditional ML (Structure-agnostic)

Traditional ML (Structure-agnostic)

- Compute $\text{Q}(\text{item}, \text{price}, \text{city}, \text{country}, \text{sales})$

item	price	city	country	sales
coat	50$	Oxford	UK	1000
...	...	...	...	...

Traditional ML (Structure-agnostic)

- Compute $\mathrm{Q}(\mathrm{item}, \mathrm{price}, \mathrm{city}, \mathrm{country}, \mathrm{sales})$

item	price	city	country	sales
coat	50$	Oxford	UK	1000
...	...	...	...	...

- One-hot encode categorical features $(\mathrm{item}, \mathrm{city}, \mathrm{country})$ to obtain D

item coat	item shoes	price	city London	city Oxford	city Saigon	country UK	country Vietnam	sales
1	0	50$	0	1	0	1	0	1000
...	...	...	...	...	...	...	...	...

Traditional ML (Structure-agnostic)

- Compute $Q(\text{item}, \text{price}, \text{city}, \text{country}, \text{sales})$

item	price	city	country	sales
coat	50$	Oxford	UK	1000
...	...	...	...	...

- One-hot encode categorical features $(\text{item}, \text{city}, \text{country})$ to obtain D

item coat	item shoes	price	city London	city Oxford	city Saigon	country UK	country Vietnam	sales
1	0	50$	0	1	0	1	0	1000
...	...	...	...	...	...	...	...	...

- Feed D into an ML tool to search for $\boldsymbol{\theta}$
  - $\boldsymbol{\theta} = (\theta_{\text{coat}}, \theta_{\text{shoes}}, \theta_{\text{price}}, \theta_{\text{London}}, \theta_{\text{Oxford}}, \theta_{\text{Saignon}}, \theta_{\text{UK}}, \theta_{\text{Vietnam}})$

Traditional ML (Structure-agnostic)

Traditional ML (Structure-agnostic)

- $\boldsymbol{\theta}$ minimizes
  - $$J(\boldsymbol{\theta}) := \underbrace{\frac{1}{2|D|} \sum_{(\boldsymbol{x},y)\in D} (\langle \boldsymbol{\theta}, \boldsymbol{x} \rangle - y)^2}_{\text{least-squares loss}} + \underbrace{\frac{\lambda}{2} \|\boldsymbol{\theta}\|_2^2}_{\ell_2\text{-regularizer}}$$

Traditional ML (Structure-agnostic)

- $\boldsymbol{\theta}$ minimizes
  - $$J(\boldsymbol{\theta}) := \underbrace{\frac{1}{2|D|} \sum_{(\boldsymbol{x},y)\in D} (\langle \boldsymbol{\theta}, \boldsymbol{x} \rangle - y)^2}_{\text{least-squares loss}} + \underbrace{\frac{\lambda}{2} \|\boldsymbol{\theta}\|_2^2}_{\ell_2\text{-regularizer}}$$
- Optimal $\boldsymbol{\theta}$ is found using Batch Gradient Descent (BGD)
  - $$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \alpha \cdot \nabla J(\boldsymbol{\theta})$$

Drawbacks

$$\mathrm{Q(item, price, city, country \to sales)} =$$

$$\mathrm{S(item, city \to sales) \wedge P(item, price) \wedge C(city \to country)}$$

Drawbacks

$$Q(\text{item}, \text{price}, \text{city}, \text{country} \rightarrow \text{sales}) = \\ S(\text{item}, \text{city} \rightarrow \text{sales}) \wedge P(\text{item}, \text{price}) \wedge C(\text{city} \rightarrow \text{country})$$

- Materializing a doubly large table
  - $\circ \quad \underbrace{|S| + |P| + |C|}_{\text{input tables}} \ll \underbrace{|Q|}_{\text{their join}} \ll \underbrace{|D|}_{\text{One-hot encoding}}$

Drawbacks

$$\text{Q}(\text{item}, \text{price}, \text{city}, \text{country} \to \text{sales}) =$$
$$\text{S}(\text{item}, \text{city} \to \text{sales}) \wedge \text{P}(\text{item}, \text{price}) \wedge \text{C}(\text{city} \to \text{country})$$

- Materializing a doubly large table
  - ○ $\underbrace{|S| + |P| + |C|}_{\text{input tables}} \ll \underbrace{|Q|}_{\text{their join}} \ll \underbrace{|D|}_{\text{One-hot encoding}}$
- Destroying data structure (e.g. JDs and FDs)
  - ML has to rediscover structure

Drawbacks

$$Q(\text{item}, \text{price}, \text{city}, \text{country} \to \text{sales}) =$$
$$S(\text{item}, \text{city} \to \text{sales}) \wedge P(\text{item}, \text{price}) \wedge C(\text{city} \to \text{country})$$

- Materializing a doubly large table
  - ○ $\underbrace{|S| + |P| + |C|}_{\text{input tables}} \ll \underbrace{|Q|}_{\text{their join}} \ll \underbrace{|D|}_{\text{One-hot encoding}}$
- Destroying data structure (e.g. JDs and FDs)
  - ML has to rediscover structure
- Expensive data move DB $\Leftrightarrow$ ML

Drawbacks

$$\text{Q(item, price, city, country} \rightarrow \text{sales)} =$$
$$\text{S(item, city} \rightarrow \text{sales)} \wedge \text{P(item, price)} \wedge \text{C(city} \rightarrow \text{country)}$$

- Materializing a doubly large table
  - ◦ $\underbrace{|S| + |P| + |C|}_{\text{input tables}} \ll \underbrace{|Q|}_{\text{their join}} \ll \underbrace{|D|}_{\text{One-hot encoding}}$
- Destroying data structure (e.g. JDs and FDs)
  - ML has to rediscover structure
- Expensive data move DB $\Leftrightarrow$ ML
- High maintenance cost (updating $\boldsymbol{\theta}$ after updates to input tables)
  - Recompute from scratch

Relational ML (Structure-aware)

- One Approach

Relational ML (Structure-aware)

- One Approach
  - Step 1: Factor out θ-independent computations
    - i.e. only dependent on the data

Relational ML (Structure-aware)

- One Approach
  - Step 1: Factor out θ-independent computations
    - i.e. only dependent on the data
  - Step 2: Encode them as SPQs
    - factor them over input tables

Relational ML (Structure-aware)

- One Approach
  - Step 1: Factor out θ-independent computations
    - i.e. only dependent on the data
  - Step 2: Encode them as SPQs
    - factor them over input tables
  - Step 3: Share computations across SPQs

Step 1: Factor out θ-independent computations

Step 1: Factor out θ-independent computations

$$J(\boldsymbol{\theta}) = \frac{1}{2|D|} \sum_{(\boldsymbol{x},y)\in D} \left(\langle \boldsymbol{\theta}, \boldsymbol{x} \rangle - y\right)^2 + \frac{\lambda}{2} \|\boldsymbol{\theta}\|_2^2$$

$$= \frac{1}{2|D|} \left(\boldsymbol{\theta}^T \underbrace{\left(\sum_{(\boldsymbol{x},y)\in D} \boldsymbol{x}\boldsymbol{x}^T \right)}_{\boldsymbol{\Sigma}} \boldsymbol{\theta} - 2 \left\langle \boldsymbol{\theta}, \underbrace{\sum_{(\boldsymbol{x},y)\in D} y \cdot \boldsymbol{x}}_{\boldsymbol{c}} \right\rangle + \underbrace{\sum_{(\boldsymbol{x},y)\in D} y^2}_{\boldsymbol{s}} \right) + \frac{\lambda}{2} \|\boldsymbol{\theta}\|_2^2$$

$$= \frac{1}{2|D|} \left(\boldsymbol{\theta}^T \boldsymbol{\Sigma} \boldsymbol{\theta} - 2 \langle \boldsymbol{\theta}, \boldsymbol{c} \rangle + \boldsymbol{s} \right) + \frac{\lambda}{2} \|\boldsymbol{\theta}\|_2^2$$

Step 1: Factor out θ-independent computations

$$J(\boldsymbol{\theta}) = \frac{1}{2|D|} \sum_{(\boldsymbol{x},y)\in D} \left(\langle \boldsymbol{\theta}, \boldsymbol{x} \rangle - y\right)^2 + \frac{\lambda}{2} \|\boldsymbol{\theta}\|_2^2$$

$$= \frac{1}{2|D|} \left(\boldsymbol{\theta}^T \underbrace{\left(\sum_{(\boldsymbol{x},y)\in D} \boldsymbol{x}\boldsymbol{x}^T \right)}_{\boldsymbol{\Sigma}} \boldsymbol{\theta} - 2 \left\langle \boldsymbol{\theta}, \underbrace{\sum_{(\boldsymbol{x},y)\in D} y \cdot \boldsymbol{x}}_{\boldsymbol{c}} \right\rangle + \underbrace{\sum_{(\boldsymbol{x},y)\in D} y^2}_{\boldsymbol{s}} \right) + \frac{\lambda}{2} \|\boldsymbol{\theta}\|_2^2$$

$$= \frac{1}{2|D|} \left(\boldsymbol{\theta}^T \boldsymbol{\Sigma}\boldsymbol{\theta} - 2\langle \boldsymbol{\theta}, \boldsymbol{c} \rangle + \boldsymbol{s} \right) + \frac{\lambda}{2} \|\boldsymbol{\theta}\|_2^2$$

$$\nabla J(\boldsymbol{\theta}) = \frac{1}{|D|} \left(\boldsymbol{\Sigma}\boldsymbol{\theta} - \boldsymbol{c} \right) + \lambda \boldsymbol{\theta}$$

Step 2: Encode $\boldsymbol{\Sigma}, \boldsymbol{c}, s$ as SPQs

Step 2: Encode $\boldsymbol{\Sigma}, \boldsymbol{c}, s$ as SPQs

item coat	item shoes	price	city London	city Oxford	city Saigon	country UK	country Vietnam	sales
1	0	50$	0	1	0	1	0	1000
...	...	...	...	...	...	...	...	...

Step 2: Encode Σ, c, s as SPQs

item coat	item shoes	price	city London	city Oxford	city Saigon	country UK	country Vietnam	sales
1	0	50$	0	1	0	1	0	1000
...	...	...	...	...	...	...	...	...

	coat shoes	price	London Oxford Saigon	UK Vietnam	sales
coat shoes					
price					
London Oxford Saigon					
UK Vietnam					
sales					

Step 2: Encode Σ, c, s as SPQs

item coat	item shoes	price	city London	city Oxford	city Saigon	country UK	country Vietnam	sales
1	0	50$	0	1	0	1	0	1000
...	...	...	...	...	...	...	...	...

	coat shoes	price	London Oxford Saigon	UK Vietnam	sales
coat shoes					
price			Σ		
London Oxford Saigon					
UK Vietnam					
sales					

Step 2: Encode $\boldsymbol{\Sigma}, \boldsymbol{c}, s$ as SPQs

item coat	item shoes	price	city London	city Oxford	city Saigon	country UK	country Vietnam	sales
1	0	50$	0	1	0	1	0	1000
...	...	...	...	...	...	...	...	...

	coat	shoes	price	London	Oxford	Saigon	UK	Vietnam	sales
coat									
shoes									
price									
London					$\boldsymbol{\Sigma}$				$\boldsymbol{c}$
Oxford									
Saigon									
UK									
Vietnam									
sales					$\boldsymbol{c}^T$				

Step 2: Encode $\boldsymbol{\Sigma}, \boldsymbol{c}, s$ as SPQs

item coat	item shoes	price	city London	city Oxford	city Saigon	country UK	country Vietnam	sales
1	0	50\$	0	1	0	1	0	1000
...	...	...	...	...	...	...	...	...

	coat	shoes	price	London	Oxford	Saigon	UK	Vietnam	sales
coat shoes									
price									
London Oxford Saigon					$\boldsymbol{\Sigma}$				$\boldsymbol{c}$
UK Vietnam									
sales					$\boldsymbol{c}^T$				s

Step 2: Encode Σ, c, s as SPQs

item coat	item shoes	price	city London	city Oxford	city Saigon	country UK	country Vietnam	sales
1	0	50$	0	1	0	1	0	1000
…	…	…	…	…	…	…	…	…

	coat shoes	price	London Oxford Saigon	UK Vietnam	sales
coat shoes	$\sigma_{\text{item,item}}$	$\sigma_{\text{item,price}}$	$\sigma_{\text{item,city}}$	$\sigma_{\text{item,country}}$	
price	$\sigma_{\text{item,price}}$	$\sigma_{\text{price,price}}$	$\sigma_{\text{price,city}}$	$\sigma_{\text{price,country}}$	
London Oxford Saigon	$\sigma_{\text{item,city}}$	$\sigma_{\text{price,city}}$	$\sigma_{\text{city,city}}$	$\sigma_{\text{city,country}}$	
UK Vietnam	$\sigma_{\text{item,country}}$	$\sigma_{\text{price,country}}$	$\sigma_{\text{city,country}}$	$\sigma_{\text{country,country}}$	
sales					

Step 2: Encode $\boldsymbol{\Sigma}, \boldsymbol{c}, s$ as SPQs

$$\mathrm{Q}(\mathrm{item}, \mathrm{price}, \mathrm{city}, \mathrm{country} \rightarrow \mathrm{sales}) =$$
$$\mathrm{S}(\mathrm{item}, \mathrm{city} \rightarrow \mathrm{sales}) \wedge \mathrm{P}(\mathrm{item}, \mathrm{price}) \wedge \mathrm{C}(\mathrm{city} \rightarrow \mathrm{country})$$

Step 2: Encode $\boldsymbol{\Sigma}, \boldsymbol{c}, \boldsymbol{s}$ as SPQs

$$\mathrm{Q}(\mathrm{item}, \mathrm{price}, \mathrm{city}, \mathrm{country} \rightarrow \mathrm{sales}) =$$
$$\mathrm{S}(\mathrm{item}, \mathrm{city} \rightarrow \mathrm{sales}) \wedge \mathrm{P}(\mathrm{item}, \mathrm{price}) \wedge \mathrm{C}(\mathrm{city} \rightarrow \mathrm{country})$$

$$\sigma_{\mathrm{item},\mathrm{item}}(\mathrm{item}) = \sum_{\mathrm{price},\mathrm{city},\mathrm{country},\mathrm{sales}} 1_{\mathrm{S}}(\mathrm{item}, \mathrm{city}, \mathrm{sales}) \cdot 1_{\mathrm{P}}(\mathrm{item}, \mathrm{price}) \cdot 1_{\mathrm{C}}(\mathrm{city}, \mathrm{country})$$

Step 2: Encode $\boldsymbol{\Sigma}, \boldsymbol{c}, \boldsymbol{s}$ as SPQs

$$\text{Q}(\text{item}, \text{price}, \text{city}, \text{country} \to \text{sales}) = \text{S}(\text{item}, \text{city} \to \text{sales}) \wedge \text{P}(\text{item}, \text{price}) \wedge \text{C}(\text{city} \to \text{country})$$

$$\sigma_{\text{item},\text{item}}(\text{item}) = \sum_{\text{price},\text{city},\text{country},\text{sales}} 1_{\text{S}}(\text{item}, \text{city}, \text{sales}) \cdot 1_{\text{P}}(\text{item}, \text{price}) \cdot 1_{\text{C}}(\text{city}, \text{country})$$

$$\sigma_{\text{item},\text{city}}(\text{item}, \text{city}) = \sum_{\text{price},\text{country},\text{sales}} 1_{\text{S}}(\text{item}, \text{city}, \text{sales}) \cdot 1_{\text{P}}(\text{item}, \text{price}) \cdot 1_{\text{C}}(\text{city}, \text{country})$$

Step 2: Encode $\boldsymbol{\Sigma}, \boldsymbol{c}, \boldsymbol{s}$ as SPQs

$$\mathrm{Q}(\mathrm{item}, \mathrm{price}, \mathrm{city}, \mathrm{country} \rightarrow \mathrm{sales}) =$$
$$\mathrm{S}(\mathrm{item}, \mathrm{city} \rightarrow \mathrm{sales}) \wedge \mathrm{P}(\mathrm{item}, \mathrm{price}) \wedge \mathrm{C}(\mathrm{city} \rightarrow \mathrm{country})$$

$$\sigma_{\mathrm{item},\mathrm{item}}(\mathrm{item}) = \sum_{\mathrm{price},\mathrm{city},\mathrm{country},\mathrm{sales}} 1_{\mathrm{S}}(\mathrm{item}, \mathrm{city}, \mathrm{sales}) \cdot 1_{\mathrm{P}}(\mathrm{item}, \mathrm{price}) \cdot 1_{\mathrm{C}}(\mathrm{city}, \mathrm{country})$$

$$\sigma_{\mathrm{item},\mathrm{city}}(\mathrm{item}, \mathrm{city}) = \sum_{\mathrm{price},\mathrm{country},\mathrm{sales}} 1_{\mathrm{S}}(\mathrm{item}, \mathrm{city}, \mathrm{sales}) \cdot 1_{\mathrm{P}}(\mathrm{item}, \mathrm{price}) \cdot 1_{\mathrm{C}}(\mathrm{city}, \mathrm{country})$$

$$\sigma_{\mathrm{item},\mathrm{price}}(\mathrm{item}) = \sum_{\mathrm{price},\mathrm{city},\mathrm{country},\mathrm{sales}} 1_{\mathrm{S}}(\mathrm{item}, \mathrm{city}, \mathrm{sales}) \cdot 1_{\mathrm{P}}(\mathrm{item}, \mathrm{price}) \cdot 1_{\mathrm{C}}(\mathrm{city}, \mathrm{country}) \cdot \mathrm{price}$$

Step 2: Encode $\boldsymbol{\Sigma}, \boldsymbol{c}, \boldsymbol{s}$ as SPQs

$$\mathrm{Q}(\text{item}, \text{price}, \text{city}, \text{country} \to \text{sales}) =$$
$$\mathrm{S}(\text{item}, \text{city} \to \text{sales}) \wedge \mathrm{P}(\text{item}, \text{price}) \wedge \mathrm{C}(\text{city} \to \text{country})$$

$$\sigma_{\text{item,item}}(\text{item}) = \sum_{\text{price,city,country,sales}} 1_{\mathrm{S}}(\text{item}, \text{city}, \text{sales}) \cdot 1_{\mathrm{P}}(\text{item}, \text{price}) \cdot 1_{\mathrm{C}}(\text{city}, \text{country})$$

$$\sigma_{\text{item,city}}(\text{item}, \text{city}) = \sum_{\text{price,country,sales}} 1_{\mathrm{S}}(\text{item}, \text{city}, \text{sales}) \cdot 1_{\mathrm{P}}(\text{item}, \text{price}) \cdot 1_{\mathrm{C}}(\text{city}, \text{country})$$

$$\sigma_{\text{item,price}}(\text{item}) = \sum_{\text{price,city,country,sales}} 1_{\mathrm{S}}(\text{item}, \text{city}, \text{sales}) \cdot 1_{\mathrm{P}}(\text{item}, \text{price}) \cdot 1_{\mathrm{C}}(\text{city}, \text{country}) \cdot \text{price}$$

$$\sigma_{\text{price,price}} = \sum_{\text{item,price,city,country,sales}} 1_{\mathrm{S}}(\text{item}, \text{city}, \text{sales}) \cdot 1_{\mathrm{P}}(\text{item}, \text{price}) \cdot 1_{\mathrm{C}}(\text{city}, \text{country}) \cdot \text{price}^2$$

Step 3: Share Computations across SPQs

$$\mathrm{Q}(\mathrm{item}, \mathrm{price}, \mathrm{city}, \mathrm{country} \rightarrow \mathrm{sales}) =$$

$$\mathrm{S}(\mathrm{item}, \mathrm{city} \rightarrow \mathrm{sales}) \wedge \mathrm{P}(\mathrm{item}, \mathrm{price}) \wedge \mathrm{C}(\mathrm{city} \rightarrow \mathrm{country})$$

Step 3: Share Computations across SPQs

$$Q(\text{item}, \text{price}, \text{city}, \text{country} \rightarrow \text{sales}) =$$
$$S(\text{item}, \text{city} \rightarrow \text{sales}) \wedge P(\text{item}, \text{price}) \wedge C(\text{city} \rightarrow \text{country})$$

$$\sigma_{\text{item},\text{item}}(\text{item}) = \sum_{\text{price},\text{city},\text{country},\text{sales}} 1_{\text{S}}(\text{item}, \text{city}, \text{sales}) \cdot 1_{\text{P}}(\text{item}, \text{price}) \cdot 1_{\text{C}}(\text{city}, \text{country})$$

Step 3: Share Computations across SPQs

$$\mathrm{Q}(\mathrm{item}, \mathrm{price}, \mathrm{city}, \mathrm{country} \rightarrow \mathrm{sales}) =$$
$$\mathrm{S}(\mathrm{item}, \mathrm{city} \rightarrow \mathrm{sales}) \wedge \mathrm{P}(\mathrm{item}, \mathrm{price}) \wedge \mathrm{C}(\mathrm{city} \rightarrow \mathrm{country})$$

$$\sigma_{\mathrm{item},\mathrm{item}}(\mathrm{item}) = \sum_{\mathrm{price},\mathrm{city},\mathrm{country},\mathrm{sales}} 1_{\mathrm{S}}(\mathrm{item}, \mathrm{city}, \mathrm{sales}) \cdot 1_{\mathrm{P}}(\mathrm{item}, \mathrm{price}) \cdot 1_{\mathrm{C}}(\mathrm{city}, \mathrm{country})$$

$\Downarrow$ (Var Elimination)

Step 3: Share Computations across SPQs

$$\mathrm{Q}(\mathrm{item}, \mathrm{price}, \mathrm{city}, \mathrm{country} \rightarrow \mathrm{sales}) = \mathrm{S}(\mathrm{item}, \mathrm{city} \rightarrow \mathrm{sales}) \land \mathrm{P}(\mathrm{item}, \mathrm{price}) \land \mathrm{C}(\mathrm{city} \rightarrow \mathrm{country})$$

$$\sigma_{\mathrm{item},\mathrm{item}}(\mathrm{item}) = \sum_{\mathrm{price},\mathrm{city},\mathrm{country},\mathrm{sales}} 1_{\mathrm{S}}(\mathrm{item}, \mathrm{city}, \mathrm{sales}) \cdot 1_{\mathrm{P}}(\mathrm{item}, \mathrm{price}) \cdot 1_{\mathrm{C}}(\mathrm{city}, \mathrm{country})$$

$\Downarrow$ (Var Elimination)

$$\sigma_{\mathrm{item},\mathrm{item}}(\mathrm{item}) = \sum_{\mathrm{city},\mathrm{sales}} 1_{\mathrm{S}}(\mathrm{item}, \mathrm{city}, \mathrm{sales}) \cdot \left(\sum_{\mathrm{country}} 1_{\mathrm{C}}(\mathrm{city}, \mathrm{country}) \right) \cdot \left(\sum_{\mathrm{price}} 1_{\mathrm{P}}(\mathrm{item}, \mathrm{price}) \right)$$

Step 3: Share Computations across SPQs

$$\text{Q}(\text{item}, \text{price}, \text{city}, \text{country} \rightarrow \text{sales}) =$$
$$\text{S}(\text{item}, \text{city} \rightarrow \text{sales}) \wedge \text{P}(\text{item}, \text{price}) \wedge \text{C}(\text{city} \rightarrow \text{country})$$

$$\sigma_{\text{item},\text{item}}(\text{item}) = \sum_{\text{price},\text{city},\text{country},\text{sales}} 1_{\text{S}}(\text{item}, \text{city}, \text{sales}) \cdot 1_{\text{P}}(\text{item}, \text{price}) \cdot 1_{\text{C}}(\text{city}, \text{country})$$

$\Downarrow$ (Var Elimination)

$$\sigma_{\text{item},\text{item}}(\text{item}) = \sum_{\text{city},\text{sales}} 1_{\text{S}}(\text{item}, \text{city}, \text{sales}) \cdot \left(\sum_{\text{country}} 1_{\text{C}}(\text{city}, \text{country}) \right) \cdot \left(\sum_{\text{price}} 1_{\text{P}}(\text{item}, \text{price}) \right)$$

Step 3: Share Computations across SPQs

$$\text{Q}(\text{item}, \text{price}, \text{city}, \text{country} \rightarrow \text{sales}) =$$
$$\text{S}(\text{item}, \text{city} \rightarrow \text{sales}) \wedge \text{P}(\text{item}, \text{price}) \wedge \text{C}(\text{city} \rightarrow \text{country})$$

$$\sigma_{\text{item},\text{item}}(\text{item}) = \sum_{\text{price},\text{city},\text{country},\text{sales}} 1_{\text{S}}(\text{item}, \text{city}, \text{sales}) \cdot 1_{\text{P}}(\text{item}, \text{price}) \cdot 1_{\text{C}}(\text{city}, \text{country})$$

$\Downarrow$ (Var Elimination)

$$\sigma_{\text{item},\text{item}}(\text{item}) = \sum_{\text{city},\text{sales}} 1_{\text{S}}(\text{item}, \text{city}, \text{sales}) \cdot \left(\sum_{\text{country}} 1_{\text{C}}(\text{city}, \text{country})\right) \cdot \left(\sum_{\text{price}} 1_{\text{P}}(\text{item}, \text{price})\right)$$

$$\sigma_{\text{item},\text{city}}(\text{item}, \text{city}) = \sum_{\text{price},\text{country},\text{sales}} 1_{\text{S}}(\text{item}, \text{city}, \text{sales}) \cdot 1_{\text{P}}(\text{item}, \text{price}) \cdot 1_{\text{C}}(\text{city}, \text{country})$$

Step 3: Share Computations across SPQs

$$Q(\text{item}, \text{price}, \text{city}, \text{country} \rightarrow \text{sales}) =$$
$$S(\text{item}, \text{city} \rightarrow \text{sales}) \wedge P(\text{item}, \text{price}) \wedge C(\text{city} \rightarrow \text{country})$$

$$\sigma_{\text{item,item}}(\text{item}) = \sum_{\text{price,city,country,sales}} 1_S(\text{item}, \text{city}, \text{sales}) \cdot 1_P(\text{item}, \text{price}) \cdot 1_C(\text{city}, \text{country})$$

$\Downarrow$ (Var Elimination)

$$\sigma_{\text{item,item}}(\text{item}) = \sum_{\text{city,sales}} 1_S(\text{item}, \text{city}, \text{sales}) \cdot \left(\sum_{\text{country}} 1_C(\text{city}, \text{country})\right) \cdot \left(\sum_{\text{price}} 1_P(\text{item}, \text{price})\right)$$

$$\sigma_{\text{item,city}}(\text{item}, \text{city}) = \sum_{\text{price,country,sales}} 1_S(\text{item}, \text{city}, \text{sales}) \cdot 1_P(\text{item}, \text{price}) \cdot 1_C(\text{city}, \text{country})$$

$\Downarrow$ (Var Elimination)

Step 3: Share Computations across SPQs

$$Q(\text{item}, \text{price}, \text{city}, \text{country} \rightarrow \text{sales}) =$$
$$S(\text{item}, \text{city} \rightarrow \text{sales}) \wedge P(\text{item}, \text{price}) \wedge C(\text{city} \rightarrow \text{country})$$

$$\sigma_{\text{item,item}}(\text{item}) = \sum_{\text{price,city,country,sales}} 1_S(\text{item}, \text{city}, \text{sales}) \cdot 1_P(\text{item}, \text{price}) \cdot 1_C(\text{city}, \text{country})$$

$\Downarrow$ (Var Elimination)

$$\sigma_{\text{item,item}}(\text{item}) = \sum_{\text{city,sales}} 1_S(\text{item}, \text{city}, \text{sales}) \cdot \left(\sum_{\text{country}} 1_C(\text{city}, \text{country}) \right) \cdot \left(\sum_{\text{price}} 1_P(\text{item}, \text{price}) \right)$$

$$\sigma_{\text{item,city}}(\text{item}, \text{city}) = \sum_{\text{price,country,sales}} 1_S(\text{item}, \text{city}, \text{sales}) \cdot 1_P(\text{item}, \text{price}) \cdot 1_C(\text{city}, \text{country})$$

$\Downarrow$ (Var Elimination)

$$\sigma_{\text{item,city}}(\text{item}, \text{city}) = \sum_{\text{sales}} 1_S(\text{item}, \text{city}, \text{sales}) \cdot \left(\sum_{\text{country}} 1_C(\text{city}, \text{country}) \right) \cdot \left(\sum_{\text{price}} 1_P(\text{item}, \text{price}) \right)$$

Relational ML Benefits

Relational ML Benefits

- DB relation $\Rightarrow$ sparse tensor representation
  - e.g. $\sigma_{\mathrm{item,city}}$ only stores non-zeros

Relational ML Benefits

- DB relation $\Rightarrow$ sparse tensor representation
  - e.g. $\sigma_{\mathrm{item,city}}$ only stores non-zeros
- Utilizing DB query optimization
  - SPQ solver (var elimination)
  - Utilizing ICs
    - e.g. $\mathrm{city} \rightarrow \mathrm{country}$ simplifies $\sum_{\mathrm{country}} 1_{\mathrm{C}}(\mathrm{city}, \mathrm{country})$

Relational ML Benefits

- DB relation $\Rightarrow$ sparse tensor representation
  - e.g. $\sigma_{\text{item,city}}$ only stores non-zeros
- Utilizing DB query optimization
  - SPQ solver (var elimination)
  - Utilizing ICs
    - e.g. $\text{city} \rightarrow \text{country}$ simplifies $\sum_{\text{country}} 1_{\text{C}}(\text{city}, \text{country})$
- Utilizing DB query evaluation
  - WCOJ

Relational ML Benefits

- DB relation $\Rightarrow$ sparse tensor representation
  - e.g. $\sigma_{\text{item,city}}$ only stores non-zeros
- Utilizing DB query optimization
  - SPQ solver (var elimination)
  - Utilizing ICs
    - e.g. $\text{city} \to \text{country}$ simplifies $\sum_{\text{country}} 1_{\text{C}}(\text{city}, \text{country})$
- Utilizing DB query evaluation
  - WCOJ
- DB parallelization $\Rightarrow$ Distributed ML for free

Relational ML Benefits

- DB relation $\Rightarrow$ sparse tensor representation
  - e.g. $\sigma_{\text{item,city}}$ only stores non-zeros
- Utilizing DB query optimization
  - SPQ solver (var elimination)
  - Utilizing ICs
    - e.g. $\text{city} \rightarrow \text{country}$ simplifies $\sum_{\text{country}} 1_{\text{C}}(\text{city}, \text{country})$
- Utilizing DB query evaluation
  - WCOJ
- DB parallelization $\Rightarrow$ Distributed ML for free
- DB IVM $\Rightarrow$ *Incremental Learning* for free
  - Maintained SPQs $\Rightarrow$ Maintained $\boldsymbol{\Sigma}$

FD-Aware Linear Regression

FD-Aware Linear Regression

item coat	item shoes	price	city London	city Oxford	city Saigon	country UK	country Vietnam	sales
1	0	50$	0	1	0	1	0	1000
...	...	...	...	...	...	...	...	...

FD-Aware Linear Regression

item coat	item shoes	price	city London	city Oxford	city Saigon	country UK	country Vietnam	sales
1	0	50$	0	1	0	1	0	1000
...	...	...	...	...	...	...	...	...

- Consider the FD $\text{city} \rightarrow \text{country}$

FD-Aware Linear Regression

item coat	item shoes	price	city London	city Oxford	city Saigon	country UK	country Vietnam	sales
1	0	50$	0	1	0	1	0	1000
...	...	...	...	...	...	...	...	...

- Consider the FD $\text{city} \rightarrow \text{country}$
- country does not increase *prediction power*
  - Suppose $\boldsymbol{R}\boldsymbol{x}_{\text{city}} = \boldsymbol{x}_{\text{country}}$

FD-Aware Linear Regression

item coat	item shoes	price	city London	city Oxford	city Saigon	country UK	country Vietnam	sales
1	0	50$	0	1	0	1	0	1000
...	...	...	...	...	...	...	...	...

- Consider the FD $\text{city} \rightarrow \text{country}$
- country does not increase *prediction power*
  - Suppose $\boldsymbol{R}\boldsymbol{x}_{\text{city}} = \boldsymbol{x}_{\text{country}}$

$$\langle \boldsymbol{\theta}, \boldsymbol{x} \rangle = \sum_{\text{j} \notin \{\text{city},\text{country}\}} \langle \boldsymbol{\theta}_{\text{j}}, \boldsymbol{x}_{\text{j}} \rangle + \langle \boldsymbol{\theta}_{\text{city}}, \boldsymbol{x}_{\text{city}} \rangle + \langle \boldsymbol{\theta}_{\text{country}}, \boldsymbol{x}_{\text{country}} \rangle$$

FD-Aware Linear Regression

item coat	item shoes	price	city London	city Oxford	city Saigon	country UK	country Vietnam	sales
1	0	50$	0	1	0	1	0	1000
...	...	...	...	...	...	...	...	...

- Consider the FD $\text{city} \rightarrow \text{country}$
- country does not increase *prediction power*
  - Suppose $\boldsymbol{R}\boldsymbol{x}_{\text{city}} = \boldsymbol{x}_{\text{country}}$

$$\langle \boldsymbol{\theta}, \boldsymbol{x} \rangle = \sum_{\text{j} \notin \{\text{city},\text{country}\}} \langle \boldsymbol{\theta}_{\text{j}}, \boldsymbol{x}_{\text{j}} \rangle + \langle \boldsymbol{\theta}_{\text{city}}, \boldsymbol{x}_{\text{city}} \rangle + \langle \boldsymbol{\theta}_{\text{country}}, \boldsymbol{x}_{\text{country}} \rangle$$

$$= \sum_{\text{j} \notin \{\text{city},\text{country}\}} \langle \boldsymbol{\theta}_{\text{j}}, \boldsymbol{x}_{\text{j}} \rangle + \langle \boldsymbol{\theta}_{\text{city}}, \boldsymbol{x}_{\text{city}} \rangle + \langle \boldsymbol{\theta}_{\text{country}}, \boldsymbol{R}\boldsymbol{x}_{\text{city}} \rangle$$

FD-Aware Linear Regression

item coat	item shoes	price	city London	city Oxford	city Saigon	country UK	country Vietnam	sales
1	0	50$	0	1	0	1	0	1000
...	...	...	...	...	...	...	...	...

- Consider the FD $\text{city} \to \text{country}$
- country does not increase *prediction power*
  - Suppose $\boldsymbol{R}\boldsymbol{x}_{\text{city}} = \boldsymbol{x}_{\text{country}}$

$$\langle \boldsymbol{\theta}, \boldsymbol{x} \rangle = \sum_{\text{j} \notin \{\text{city,country}\}} \langle \boldsymbol{\theta}_{\text{j}}, \boldsymbol{x}_{\text{j}} \rangle + \langle \boldsymbol{\theta}_{\text{city}}, \boldsymbol{x}_{\text{city}} \rangle + \langle \boldsymbol{\theta}_{\text{country}}, \boldsymbol{x}_{\text{country}} \rangle$$

$$= \sum_{\text{j} \notin \{\text{city,country}\}} \langle \boldsymbol{\theta}_{\text{j}}, \boldsymbol{x}_{\text{j}} \rangle + \langle \boldsymbol{\theta}_{\text{city}}, \boldsymbol{x}_{\text{city}} \rangle + \langle \boldsymbol{\theta}_{\text{country}}, \boldsymbol{R}\boldsymbol{x}_{\text{city}} \rangle$$

$$= \sum_{\text{j} \notin \{\text{city,country}\}} \langle \underbrace{\boldsymbol{\theta}_{\text{j}}}_{\gamma_{\text{j}}}, \boldsymbol{x}_{\text{j}} \rangle + \langle \underbrace{\boldsymbol{\theta}_{\text{city}} + \boldsymbol{R}^{\text{T}} \boldsymbol{\theta}_{\text{country}}}_{\gamma_{\text{city}}}, \boldsymbol{x}_{\text{city}} \rangle$$

FD-Aware Linear Regression

item coat	item shoes	price	city London	city Oxford	city Saigon	country UK	country Vietnam	sales
1	0	50$	0	1	0	1	0	1000
...	...	...	...	...	...	...	...	...

- However country can help create *a smaller model*

FD-Aware Linear Regression

item coat	item shoes	price	city London	city Oxford	city Saigon	country UK	country Vietnam	sales
1	0	50$	0	1	0	1	0	1000
…	…	…	…	…	…	…	…	…

- However country can help create *a smaller model*
  - e.g. compress $(\theta_{\text{London}}, \theta_{\text{Oxford}}, \theta_{\text{UK}}) = (1, 1, 0)$ into $(0, 0, 1)$

FD-Aware Linear Regression

item coat	item shoes	price	city London	city Oxford	city Saigon	country UK	country Vietnam	sales
1	0	50$	0	1	0	1	0	1000
...	...	...	...	...	...	...	...	...

- However country can help create *a smaller model*
  - e.g. compress $(\theta_{\text{London}}, \theta_{\text{Oxford}}, \theta_{\text{UK}}) = (1, 1, 0)$ into $(0, 0, 1)$

$$\mathrm{J}(\boldsymbol{\theta}) = \frac{1}{2|\mathrm{D}|} \sum_{(\boldsymbol{x},\mathrm{y}) \in \mathrm{D}} \left(\langle \boldsymbol{\theta}, \boldsymbol{x} \rangle - \mathrm{y} \right)^2 + \frac{\lambda}{2} \|\boldsymbol{\theta}\|_2^2$$

FD-Aware Linear Regression

item coat	item shoes	price	city London	city Oxford	city Saigon	country UK	country Vietnam	sales
1	0	50$	0	1	0	1	0	1000
...	...	...	...	...	...	...	...	...

- However $\mathrm{country}$ can help create *a smaller model*
  - e.g. compress $(\theta_{\mathrm{London}}, \theta_{\mathrm{Oxford}}, \theta_{\mathrm{UK}}) = (1, 1, 0)$ into $(0, 0, 1)$

$$J(\boldsymbol{\theta}) = \frac{1}{2|D|} \sum_{(\boldsymbol{x},y)\in D} \left(\langle \boldsymbol{\theta}, \boldsymbol{x} \rangle - y \right)^2 + \frac{\lambda}{2} \|\boldsymbol{\theta}\|_2^2$$

$$J(\gamma, \boldsymbol{\theta}_{\mathrm{country}}) = \frac{1}{2|D|} \sum_{(\boldsymbol{x},y)\in D} \left(\langle \gamma, \bar{\boldsymbol{x}} \rangle - y \right)^2$$

$$+ \frac{\lambda}{2} \left(\sum_{j\neq\mathrm{city}} \|\underbrace{\gamma_j}_{\boldsymbol{\theta}_j}\|_2^2 + \|\underbrace{\gamma_{\mathrm{city}} - \boldsymbol{R}^{\mathrm{T}} \boldsymbol{\theta}_{\mathrm{country}}}_{\boldsymbol{\theta}_{\mathrm{city}}}\|_2^2 + \|\boldsymbol{\theta}_{\mathrm{country}}\|_2^2 \right)$$

FD-Aware Linear Regression

$$+\frac{\lambda}{2}\left(\sum_{\mathrm{j}\neq\mathrm{city}}\|\underbrace{\boldsymbol{\gamma}_{\mathrm{j}}}_{\boldsymbol{\theta}_{\mathrm{j}}}\|_2^2+\|\underbrace{\boldsymbol{\gamma}_{\mathrm{city}}-\boldsymbol{R}^{\mathrm{T}}\boldsymbol{\theta}_{\mathrm{country}}}_{\boldsymbol{\theta}_{\mathrm{city}}}\|_2^2+\|\boldsymbol{\theta}_{\mathrm{country}}\|_2^2\right)$$

FD-Aware Linear Regression

$$+\frac{\lambda}{2}\left(\sum_{j\neq city}\|\underbrace{\gamma_j}_{\boldsymbol{\theta}_j}\|_2^2+\|\underbrace{\gamma_{city}-\boldsymbol{R}^{T}\boldsymbol{\theta}_{country}}_{\boldsymbol{\theta}_{city}}\|_2^2+\|\boldsymbol{\theta}_{country}\|_2^2\right)$$

For a fixed γ_j, optimal $\boldsymbol{\theta}_{country}$:

FD-Aware Linear Regression

$$+\frac{\lambda}{2}\left(\sum_{\mathrm{j}\neq\mathrm{city}} \|\underbrace{\gamma_{\mathrm{j}}}_{\boldsymbol{\theta}_{\mathrm{j}}}\|_2^2 + \|\underbrace{\gamma_{\mathrm{city}} - \boldsymbol{R}^{\mathrm{T}}\boldsymbol{\theta}_{\mathrm{country}}}_{\boldsymbol{\theta}_{\mathrm{city}}}\|_2^2 + \|\boldsymbol{\theta}_{\mathrm{country}}\|_2^2\right)$$

For a fixed γ_{j}, optimal $\boldsymbol{\theta}_{\mathrm{country}}$:

$$\boldsymbol{\theta}^*_{\mathrm{country}} = \boldsymbol{R}(\boldsymbol{I}_{\mathrm{city}} + \boldsymbol{R}^{\mathrm{T}}\boldsymbol{R})^{-1}\gamma_{\mathrm{city}}$$

FD-Aware Linear Regression

$$+\frac{\lambda}{2}\left(\sum_{j\neq city}\|\underbrace{\gamma_j}_{\boldsymbol{\theta}_j}\|_2^2+\|\underbrace{\gamma_{city}-\boldsymbol{R}^T\boldsymbol{\theta}_{country}}_{\boldsymbol{\theta}_{city}}\|_2^2+\|\boldsymbol{\theta}_{country}\|_2^2\right)$$

For a fixed γ_j, optimal $\boldsymbol{\theta}_{country}$:

$$\boldsymbol{\theta}^*_{country}=\boldsymbol{R}(\boldsymbol{I}_{city}+\boldsymbol{R}^T\boldsymbol{R})^{-1}\gamma_{city}$$

$$J(\gamma)=\frac{1}{2|D|}\sum_{(\boldsymbol{x},y)\in D}\left(\langle\gamma,\bar{\boldsymbol{x}}\rangle-y\right)^2$$

$$+\frac{\lambda}{2}\left(\sum_{j\neq city}\|\gamma_j\|_2^2+\langle(\boldsymbol{I}_{city}+\boldsymbol{R}^T\boldsymbol{R})^{-1}\gamma_{city},\gamma_{city}\rangle\right)$$

Generalizing The Linear Regression Example

(Some) Supported Models

(Some) Supported Models

- Out-of-the-box
  - Polynomial Regression
  - Factorization Machines
  - PCA

(Some) Supported Models

- Out-of-the-box
  - Polynomial Regression
  - Factorization Machines
  - PCA
- With more work (SPQs + *Linear Inequalities*) [Abo Khamis et al, TODS'20]
  - Piece-wise linear loss functions
  - SVMs
  - k-means
  - CART
  - ***and more...***

(Some) Supported Models

- Out-of-the-box
  - Polynomial Regression
  - Factorization Machines
  - PCA
- With more work (SPQs + *Linear Inequalities*) [Abo Khamis et al, TODS'20]
  - Piece-wise linear loss functions
  - SVMs
  - k-means
  - CART
  - ***and more...***
- Unsupported
  - Deep Neural Networks

Open Problems

- Relational Algorithms for Deep Neural Networks?
- Based on *Relational Data Structure*, can we..

Open Problems

- Relational Algorithms for Deep Neural Networks?
- Based on *Relational Data Structure*, can we..
  - design a *Model Structure* ?
    - e.g.
      - Graphs $\Rightarrow$ Graph Neural Networks
      - Factorized data $\Rightarrow$ *"Factorized NNs"*?

Open Problems

- Relational Algorithms for Deep Neural Networks?
- Based on *Relational Data Structure*, can we..
  - design a *Model Structure* ?
    - e.g.
      - Graphs $\Rightarrow$ Graph Neural Networks
      - Factorized data $\Rightarrow$ *"Factorized NNs"*?
  - impose constraints on *Model Parameters*?
    - e.g. FDs/JDs in the training data $\Rightarrow$ constraints on NN weights?

Thank You!

Any Questions/Comments?