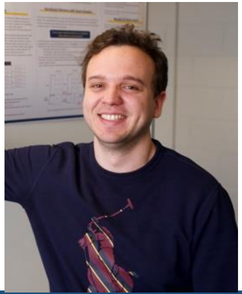
StarfishDB: a Query Execution Engine for Relational Probabilistic Programming

Ouael Ben Amara*
benamara@umich.edu



Sami Hadouaj*
shadouaj@umich.edu

Niccolò Meneghetti
niccolom@umich.edu

University of Michigan-Dearborn

What is Probabilistic Programming?

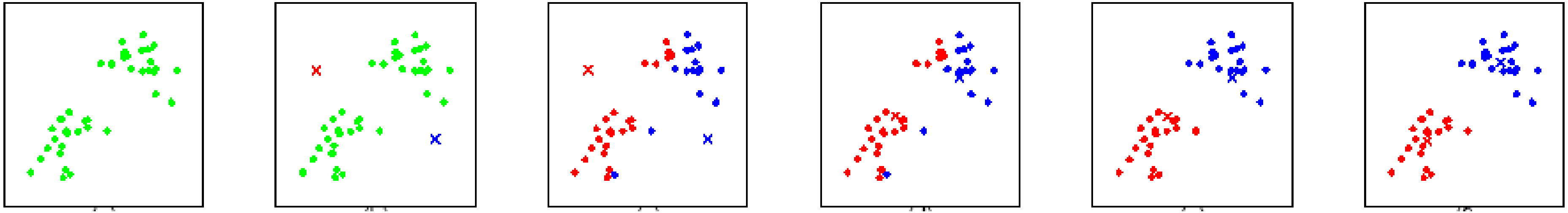
Generative Story:

$$z \sim \text{Categorical}(\phi)$$

$$\mathbf{x} \sim \text{Gaussian}(\boldsymbol{\mu}_z, \boldsymbol{\Sigma}_z)$$

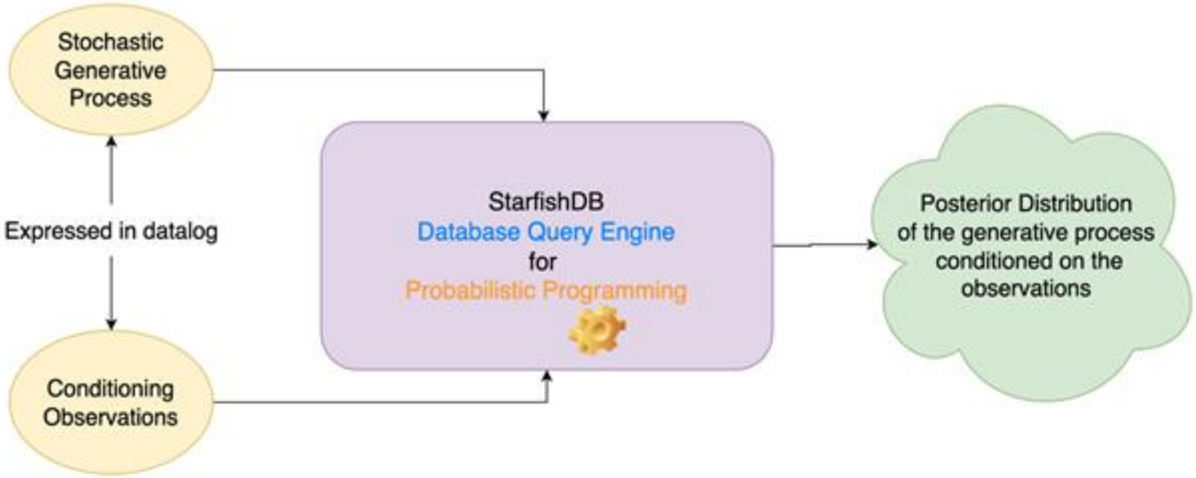
Data: $\mathcal{D} = \{\mathbf{x}^{(i)}\}_{i=1}^{N}$ where $\mathbf{x}^{(i)} \in \mathbb{R}^M$

Goal: compute the posterior density of the generative process w.r.t. the data.



Main Contributions

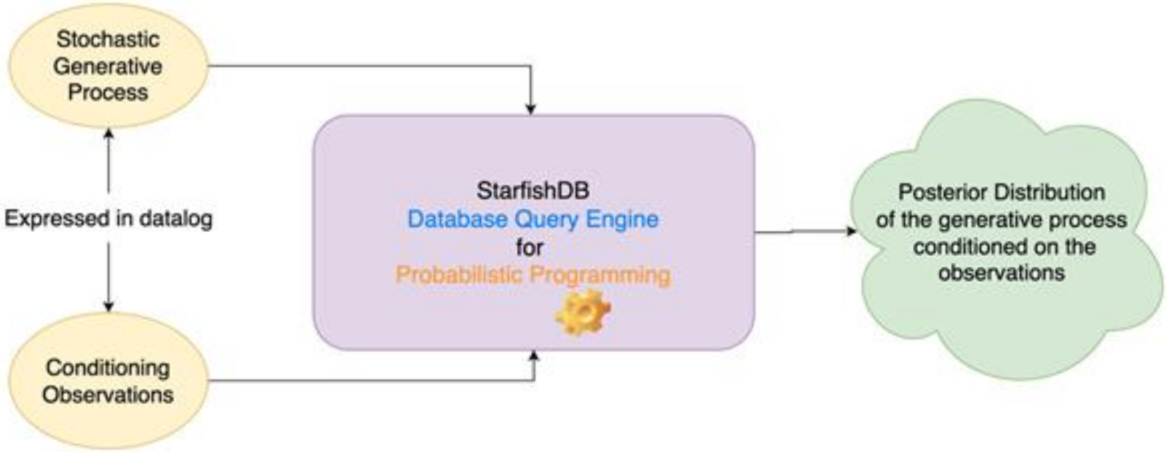
- We introduce our own probabilistic programming language that is database-centric



[1] Vince BáRány, Balder Ten Cate, Benny Kimelfeld, Dan Olteanu, and Zografoula Vagena. 2017. **Declarative Probabilistic Programming with Datalog**. ACM Trans. Database Syst. 42, 4, Article 22 (December 2017), 35 pages. https://doi.org/10.1145/3132700

Main Contributions

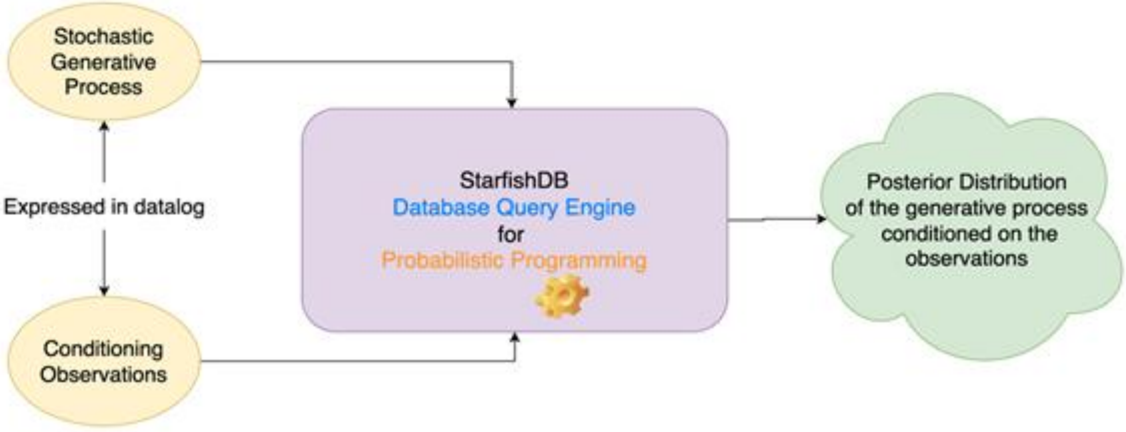
- We introduce our own probabilistic programming language that is database-centric
- We use probabilistic programming Datalog [1] to our framework



[1] Vince BáRány, Balder Ten Cate, Benny Kimelfeld, Dan Olteanu, and Zografoula Vagena. 2017. **Declarative Probabilistic Programming with Datalog**. ACM Trans. Database Syst. 42, 4, Article 22 (December 2017), 35 pages. https://doi.org/10.1145/3132700

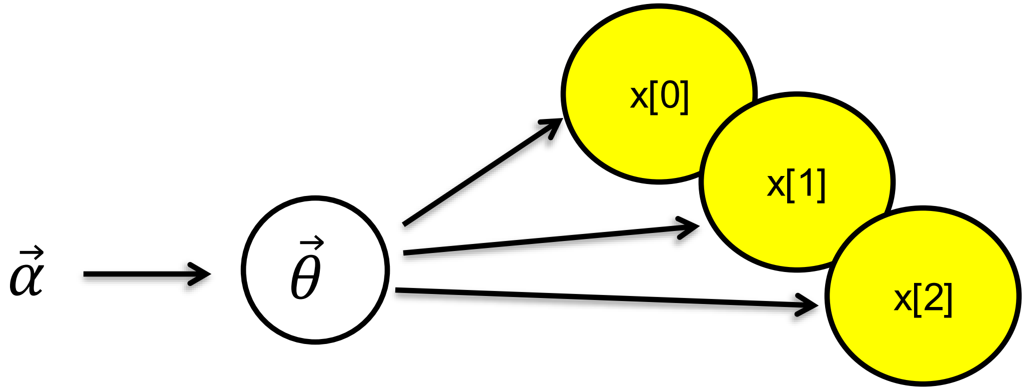
Main Contributions

- We introduce our own probabilistic programming language that is database-centric
- We use probabilistic programming Datalog [1] to our framework
- We leverage Just in time compilation to speed up inference



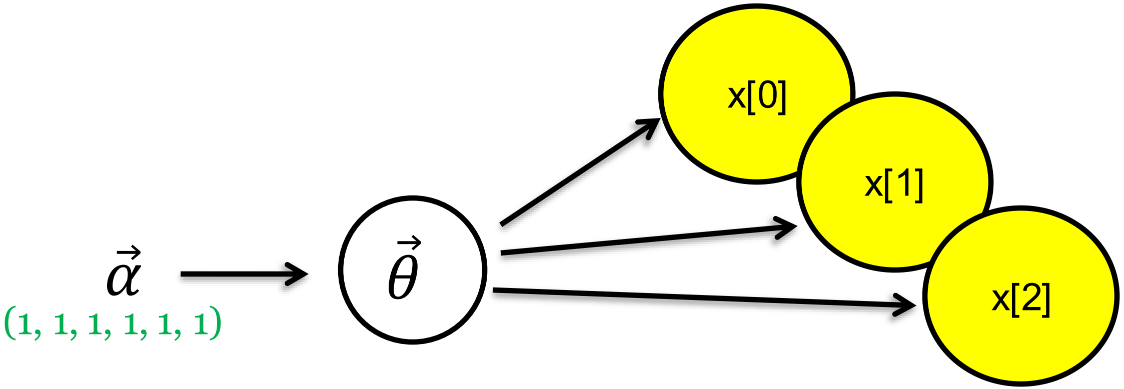
[1] Vince BáRány, Balder Ten Cate, Benny Kimelfeld, Dan Olteanu, and Zografoula Vagena. 2017. **Declarative Probabilistic Programming with Datalog**. ACM Trans. Database Syst. 42, 4, Article 22 (December 2017), 35 pages. https://doi.org/10.1145/3132700

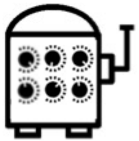
Pólya Die



`Pólya die` → a *Categorical* distribution (parametrized by $\vec{\theta}$)
with a *Dirichlet* prior (parametrized by $\vec{\alpha}$)

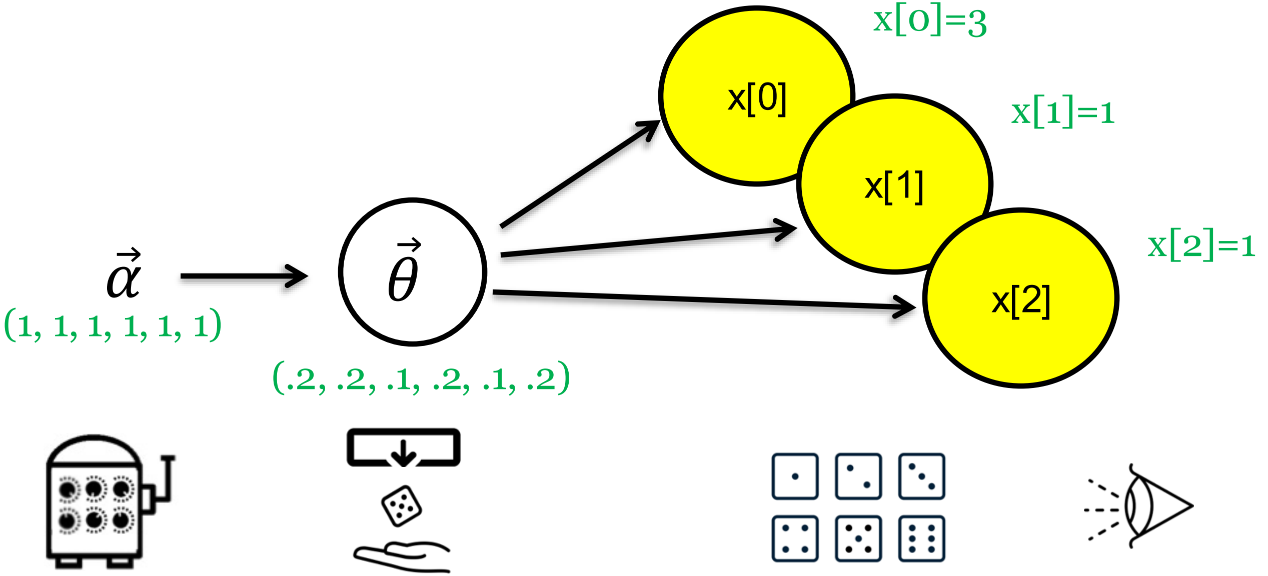
Pólya Die





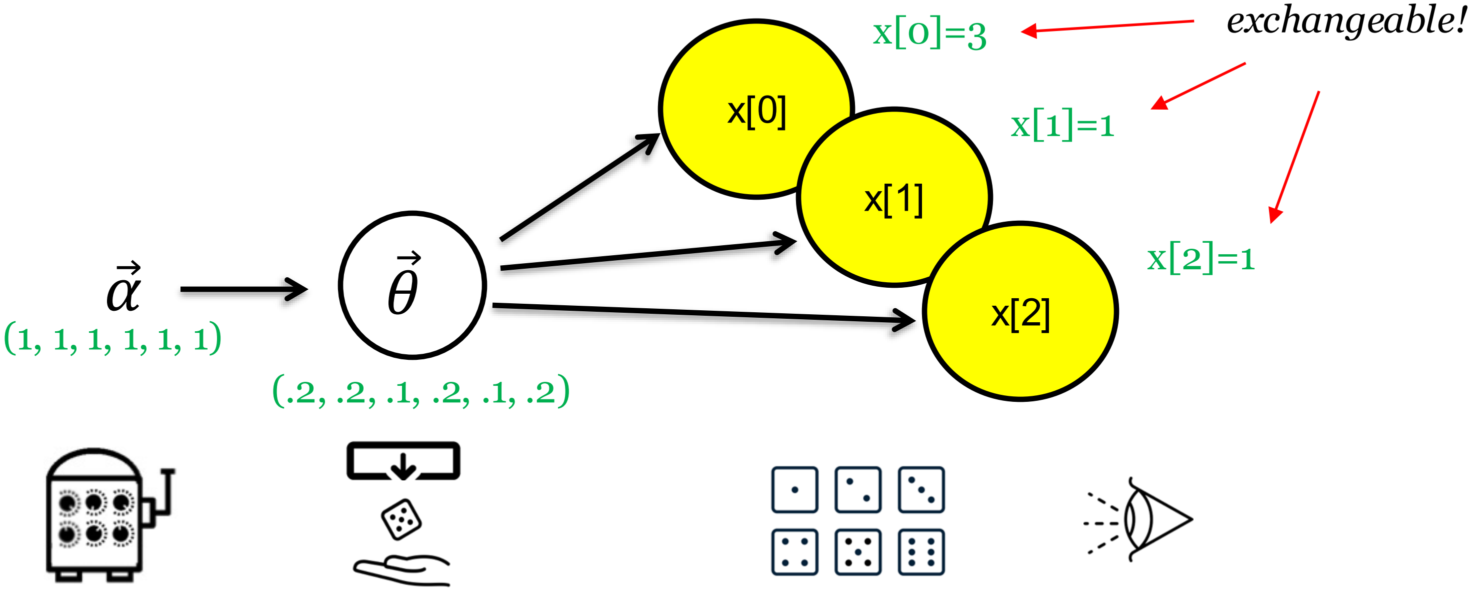
`Pólya die` → a *Categorical* distribution (parametrized by $\vec{\theta}$)
with a *Dirichlet* prior (parametrized by $\vec{\alpha}$)

Pólya Die



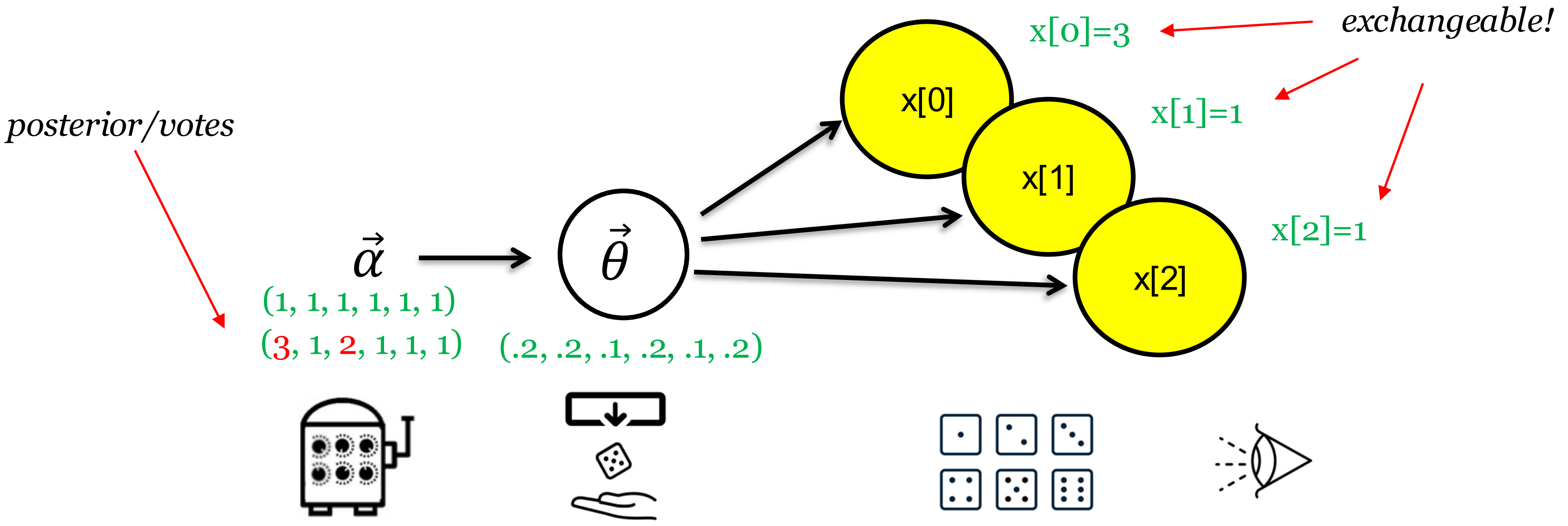
`Pólya die` → a *Categorical* distribution (parametrized by $\vec{\theta}$) with a *Dirichlet* prior (parametrized by $\vec{\alpha}$)

Pólya Die



`Pólya die` → a *Categorical* distribution (parametrized by $\vec{\theta}$)
with a *Dirichlet* prior (parametrized by $\vec{\alpha}$)

Pólya Die



`Pólya die` → a *Categorical* distribution (parametrized by $\vec{\theta}$)
with a *Dirichlet* prior (parametrized by $\vec{\alpha}$)

Toy Example: 2 dice, 2 constraints

(Ada, Lead)

(Ada, Dev)

(Ada, QA)

Pólya Die X_1 ⟹ Defines a role for Ada

(Bob, Lead)

(Bob, Dev)

(Bob, QA)

Pólya Die X_2 ⟹ Defines a role for Bob

Toy Example

Example of a probabilistic relation:

EMP	ROLE		
Ada (x_1)	Lead (v_1)	α_1	1 ($\alpha_{1,1}$)
	Dev (v_2)		1 ($\alpha_{1,2}$)
	QA (v_3)		1 ($\alpha_{1,3}$)
Bob (x_2)	Lead (v_1)	α_2	1 ($\alpha_{2,1}$)
	Dev (v_2)		3 ($\alpha_{2,2}$)
	QA (v_3)		4 ($\alpha_{2,3}$)

There 9 possible worlds for Ada and Bob positions

EMP	ROLE
Ada	Lead
Bob	Lead

EMP	ROLE
Ada	Dev
Bob	Lead

EMP	ROLE
Ada	QA
Bob	Lead

EMP	ROLE
Ada	Lead
Bob	Dev

EMP	ROLE
Ada	Dev
Bob	Dev

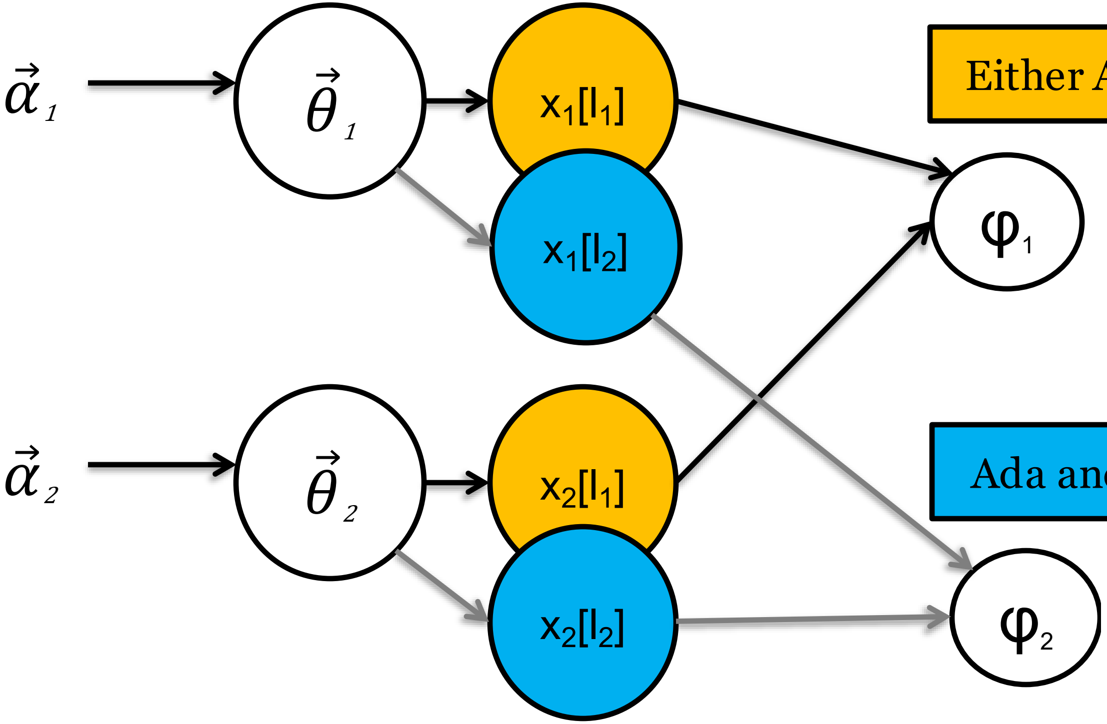
EMP	ROLE
Ada	QA
Bob	Dev

EMP	ROLE
Ada	Lead
Bob	QA

EMP	ROLE
Ada	Dev
Bob	QA

EMP	ROLE
Ada	QA
Bob	QA

Toy Example: 2 dice, 2 constraints



Either Ada or Bob has the role of "Lead"

$$\phi_1 = (X_1[l_1] = v_1) \vee (X_2[l_1] = v_1)$$

Ada and Bob have the same role

$$\phi_2 = \bigvee_{i=1}^{3} [(X_1[l_2] = v_i) \wedge (X_2[l_2] = v_i)]$$

Toy Example

EMP	ROLE
Ada	Lead
Bob	Lead

EMP	ROLE
Ada	Dev
Bob	Lead

EMP	ROLE
Ada	QA
Bob	Lead

EMP	ROLE
Ada	Lead
Bob	Dev

EMP	ROLE
Ada	Dev
Bob	Dev

EMP	ROLE
Ada	QA
Bob	Dev

EMP	ROLE
Ada	Lead
Bob	QA

EMP	ROLE
Ada	Dev
Bob	QA

EMP	ROLE
Ada	QA
Bob	QA

c_1: Either Ada or Bob has the of role “Lead”

$$\phi_1 = (X_1[l_1] = v_1) \vee (X_2[l_1] = v_1)$$

$\tau_{1,1}$: Ada is Lead, Bob is Lead

$\tau_{1,2}$: Ada is Dev, Bob is Lead

$\tau_{1,3}$: Ada is QA, Bob is Lead

$\tau_{1,4}$: Ada is Lead, Bob is Dev

$\tau_{1,5}$: Ada is Lead, Bob is QA

$$\textsc{Sat}\,(\Phi, \mathbb{X}) = \{\tau_{1,1}, \tau_{1,2}, \tau_{1,3}, \tau_{1,4}, \tau_{1,5}\} \times \{\tau_{2,1}, \tau_{2,2}, \tau_{2,3}\}$$

Toy Example

EMP	ROLE
Ada	Lead
Bob	Lead

EMP	ROLE
Ada	Dev
Bob	Lead

EMP	ROLE
Ada	QA
Bob	Lead

EMP	ROLE
Ada	Lead
Bob	Dev

EMP	ROLE
Ada	Dev
Bob	Dev

EMP	ROLE
Ada	QA
Bob	Dev

EMP	ROLE
Ada	Lead
Bob	QA

EMP	ROLE
Ada	Dev
Bob	QA

EMP	ROLE
Ada	QA
Bob	QA

c_2: Ada and Bob have the same role

$$\phi_2 = \bigvee_{i=1}^{3} [(X_1[l_2] = v_i) \wedge (X_2[l_2] = v_i)]$$

$\tau_{2,1}$: Ada is Lead, Bob is Lead
$\tau_{2,2}$: Ada is Dev, Bob is Dev
$\tau_{2,3}$: Ada is QA, Bob is QA

$$\textsc{Sat}(\Phi, \mathbb{X}) = \{\tau_{1,1}, \tau_{1,2}, \tau_{1,3}, \tau_{1,4}, \tau_{1,5}\} \times \{\tau_{2,1}, \tau_{2,2}, \tau_{2,3}\}$$

Toy Example

Example of a probabilistic relation:

EMP	ROLE		
Ada (x_1)	Lead (v_1)	α_1	1 ($\alpha_{1,1}$)
	Dev (v_2)		1 ($\alpha_{1,2}$)
	QA (v_3)		1 ($\alpha_{1,3}$)
Bob (x_2)	Lead (v_1)	α_2	1 ($\alpha_{2,1}$)
	Dev (v_2)		3 ($\alpha_{2,2}$)
	QA (v_3)		4 ($\alpha_{2,3}$)

Generative process

+

Probabilistic program

Constraints

c_1: Either Ada or Bob has the role of "Lead"

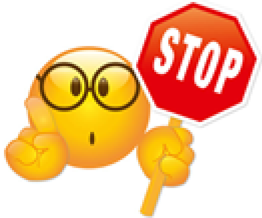
$$\phi_1 = (X_1[l_1] = v_1) \vee (X_2[l_1] = v_1)$$

c_2: Ada and Bob have the same role

$$\phi_2 = \bigvee_{i=1}^{3} [(X_1[l_2] = v_i) \wedge (X_2[l_2] = v_i)]$$

Main Goal of a Probabilistic Program

Compute the posterior distribution of generative process w.r.t constraints :



product of Dirichlet densities

$$p(\Theta \mid \Phi, \mathbb{A}) = \sum_{\tau \in \mathrm{SAT}(\Phi, \mathbb{X})} p(\Theta \mid \tau, \mathbb{A}) \cdot P(\tau \mid \Phi, \mathbb{A})$$

posterior

exponentially large ☹

marginal probability of a possible world

Toy example: Gibbs Sampling

i	$S[\phi_1]$	$S[\phi_2]$
(init)	$(X_1\ [l_1] = v_1)\ \wedge (X_2\ [l_1] = v3)$ Ada is lead in obs l_1 ∧ Bob is QA in obs l_1	$(X_1\ [l_2] = v_3)\ \wedge (X_2\ [l_2] = v3)$ Ada is QA in obs l_2 ∧ Bob is QA in obs l_2

c_1: Either Ada or Bob has the role of "Lead"

$$\phi_1 = (X_1[l_1] = v_1) \vee (X_2[l_1] = v_1)$$

c_2: Ada and Bob have the same role

$$\phi_2 = \bigvee_{i=1}^{3} [(X_1[l_2] = v_i) \wedge (X_2[l_2] = v_i)]$$

Toy example : Gibbs Sampling

i	$S[\phi_1]$	$S[\phi_2]$
(init)	$(X_1\,[l_1] = v_1) \wedge (X_2\,[l_1] = v3)$ Ada is lead in obs l_1 $\wedge$ Bob is QA in obs l_1	$(X_1\,[l_2] = v_3) \wedge (X_2\,[l_2] = v3)$ Ada is QA in obs l_2 $\wedge$ Bob is QA in obs l_2
0	$(X_1\,[l_1] = v_2) \wedge (X_2\,[l_1] = v_1)$ Ada is dev in obs l_1 $\wedge$ Bob is lead in obs l_1	$(X_1\,[l_2] = v_1) \wedge (X_2\,[l_2] = v_1)$ Ada is lead in obs l_2 $\wedge$ Bob is lead in obs l_2

c_1: Either Ada or Bob has the role of "Lead"

$$\phi_1 = (X_1[l_1] = v_1) \vee (X_2[l_1] = v_1)$$

c_2: Ada and Bob have the same role

$$\phi_2 = \bigvee_{i=1}^{3} [(X_1[l_2] = v_i) \wedge (X_2[l_2] = v_i)]$$

Toy example : Gibbs Sampling

i	$S[\phi_1]$	$S[\phi_2]$
(init)	$(X_1\ [l_1] = v_1)\ \wedge (X_2\ [l_1] = v3)$ Ada is lead in obs l_1 ∧ Bob is QA in obs l_1	$(X_1\ [l_2] = v_3)\ \wedge (X_2\ [l_2] = v3)$ Ada is QA in obs l_2 ∧ Bob is QA in obs l_2
0	$(X_1\ [l_1] = v_2)\ \wedge (X_2\ [l_1] = v1)$ Ada is dev in obs l_1 ∧ Bob is lead in obs l_1	$(X_1\ [l_2] = v_1)\ \wedge (X_2\ [l_2] = v_1)$ Ada is lead in obs l_2 ∧ Bob is lead in obs l_2
1	$(X_1\ [l_1] = v_3)\ \wedge (X_2\ [l_1] = v1)$ Ada is QA in obs l_1 ∧ Bob is lead in obs l_1	$(X_1\ [l_2] = v_3)\ \wedge (X_2\ [l_2] = v3)$ Ada is QA in obs l_2 ∧ Bob is QA in obs l_2

c_1: Either Ada or Bob has the role of "Lead"

$$\phi_1 = (X_1[l_1] = v_1) \vee (X_2[l_1] = v_1)$$

c_2: Ada and Bob have the same role

$$\phi_2 = \bigvee_{i=1}^{3} [(X_1[l_2] = v_i) \wedge (X_2[l_2] = v_i)]$$

i	$S[\phi_1]$	$S[\phi_2]$
(init)	$(X_1\,[l_1] = v_1) \wedge (X_2\,[l_1] = v3)$ Ada is lead in obs l_1 ∧ Bob is QA in obs l_1	$(X_1\,[l_2] = v_3) \wedge (X_2\,[l_2] = v3)$ Ada is QA in obs l_2 ∧ Bob is QA in obs l_2
0	$(X_1\,[l_1] = v_2) \wedge (X_2\,[l_1] = v1)$ Ada is dev in obs l_1 ∧ Bob is lead in obs l_1	$(X_1\,[l_2] = v_1) \wedge (X_2\,[l_2] = v_1)$ Ada is lead in obs l_2 ∧ Bob is lead in obs l_2
1	$(X_1\,[l_1] = v_3) \wedge (X_2\,[l_1] = v1)$ Ada is QA in obs l_1 ∧ Bob is lead in obs l_1	$(X_1\,[l_2] = v_3) \wedge (X_2\,[l_2] = v_3)$ Ada is QA in obs l_2 ∧ Bob is QA in obs l_2
2	$(X_1\,[l_1] = v_1) \wedge (X_2\,[l_1] = v_1)$ Ada is lead in obs l_1 ∧ Bob is lead in obs l_1	$(X_1\,[l_2] = v_2) \wedge (X_2\,[l_2] = v_2)$ Ada is dev in obs l_2 ∧ Bob is dev in obs l_2
…	…	…

c_1: Either Ada or Bob has the role of "Lead"

$$\phi_1 = (X_1[l_1] = v_1) \vee (X_2[l_1] = v_1)$$

c_2: Ada and Bob have the same role

$$\phi_2 = \bigvee_{i=1}^{3} [(X_1[l_2] = v_i) \wedge (X_2[l_2] = v_i)]$$

Probabilistic Programming Datalog[1] (PPDL)

$$\mathrm{weather}(\underline{C, T}, w \in \{\mathrm{sun}, \mathrm{rain}\} \sim Cat[\![P]\!]) \leftarrow \mathrm{city}(C, P), \mathrm{ts}(T).$$

```
city('Fargo', [.1, .9]), ts('noon')
```

```
weather('Fargo', 'noon', sun) with prob 0.1

weather('Fargo', 'noon', rain) with prob 0.9
```

[1] Vince BáRány, Balder Ten Cate, Benny Kimelfeld, Dan Olteanu, and Zografoula Vagena. 2017. **Declarative Probabilistic Programming with Datalog**. ACM Trans. Database Syst. 42, 4, Article 22 (December 2017), 35 pages. https://doi.org/10.1145/3132700

Probabilistic Programming Datalog[1] (PPDL)

$$\mathrm{weather}(\underline{C, T}, w \in \{\mathrm{sun}, \mathrm{rain}\} \sim Cat[\![P]\!]) \leftarrow \mathrm{city}(C, P), \mathrm{ts}(T).$$

```
city('Fargo', [.1, .9]), ts('noon')
```

```
weather('Fargo', 'noon', sun) with prob 0.1

weather('Fargo', 'noon', rain) with prob 0.9
```

[1] Vince BáRány, Balder Ten Cate, Benny Kimelfeld, Dan Olteanu, and Zografoula Vagena. 2017. **Declarative Probabilistic Programming with Datalog**. ACM Trans. Database Syst. 42, 4, Article 22 (December 2017), 35 pages. https://doi.org/10.1145/3132700

Probabilistic Programming Datalog[1] (PPDL)

$$\text{weather}(\underline{C, T}, w \in \{\text{sun}, \text{rain}\} \sim Cat[\![P]\!]) \leftarrow \text{city}(C, P), \text{ts}(T).$$

```
city('Fargo', [.1, .9]), ts('noon')
```

```
weather('Fargo', 'noon', sun) with prob 0.1

weather('Fargo', 'noon', rain) with prob 0.9
```

[1] Vince BáRány, Balder Ten Cate, Benny Kimelfeld, Dan Olteanu, and Zografoula Vagena. 2017. **Declarative Probabilistic Programming with Datalog**. ACM Trans. Database Syst. 42, 4, Article 22 (December 2017), 35 pages. https://doi.org/10.1145/3132700

Probabilistic Programming Datalog[1] (PPDL)

$$\text{weather}(\underline{C, T}, w \in \{\text{sun}, \text{rain}\} \sim Cat[\![P]\!]) \leftarrow \text{city}(C, P), \text{ts}(T).$$

```
city('Fargo', [.1, .9]), ts('noon')
```

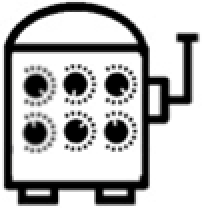
```
weather('Fargo', 'noon', sun) with prob 0.1
```

```
weather('Fargo', 'noon', rain) with prob 0.9
```

[1] Vince BáRány, Balder Ten Cate, Benny Kimelfeld, Dan Olteanu, and Zografoula Vagena. 2017. **Declarative Probabilistic Programming with Datalog**. ACM Trans. Database Syst. 42, 4, Article 22 (December 2017), 35 pages. https://doi.org/10.1145/3132700

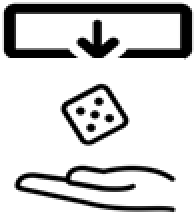
Probabilistic Programming Datalog

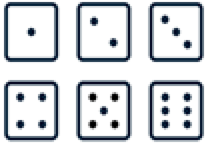
(1, 1, 1, 1, 1, 1)

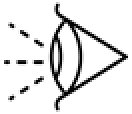


$$\mathrm{lp}(\underline{\mathit{VarId}}, D, p \in \mathcal{S}_{|D|} \sim \mathcal{D}ir[\![H]\!]) \leftarrow \mathrm{dt}(\mathit{VarId}, D, H)$$

Probabilistic Programming Datalog

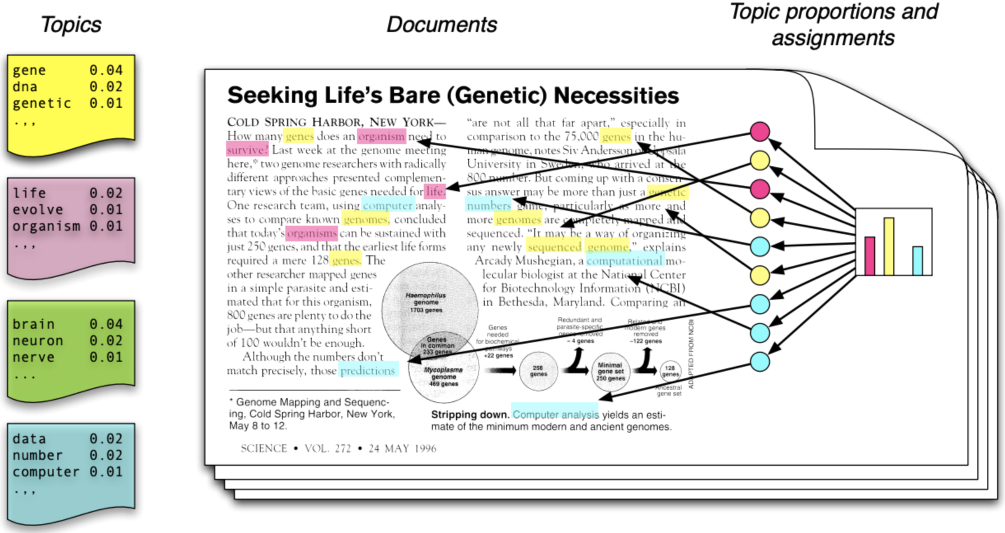






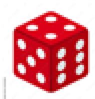
$$\boxed{\mathrm{obs}(\underline{\mathit{VarId}, \mathit{ObsId}}, v \in D \sim \mathit{Cat}[\![P]\!])} \leftarrow \mathrm{lp}(\mathit{VarId}, D, P),\ \boxed{\mathrm{sample}(\mathit{VarId}, \mathit{ObsId})}$$

Latent Dirichlet Allocation[3] with Pólya Dice



N documents → **N** red dice

K topics → **K** blue dice



Red die: generate numbers between 1 and K

Blue die: generate words from a fixed vocabulary

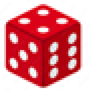
- Each **topic** is a distribution over words
- Each **document** is a mixture of corpus-wide topics
- Each **word** is drawn from one of those topics

Source: Blei, ICML 2012 Tutorial

[3] Blei, David M., Andrew Y. Ng, and Michael I. Jordan. "Latent Dirichlet allocation

Latent Dirichlet Allocation with Pólya Dice

$$\mathrm{dt}([\mathrm{red}, D], \mathrm{ts}, [1, 1, .., 1]) \leftarrow \mathrm{d}(D, P, W)$$



Instantiate a red die for each document

$$\mathrm{dt}([\mathrm{blue}, T], \mathrm{ws}, [1, 1, .., 1]) \leftarrow \mathrm{t}(T).$$

Instantiate a blue die for each topic

Latent Dirichlet Allocation with Pólya Dice

$$\mathrm{dt}([\mathrm{red}, D], \mathrm{ts}, [1, 1, .., 1]) \leftarrow \mathrm{d}(D, P, W).$$

Instantiate a red die for each document

$$\mathrm{dt}([\mathrm{blue}, T], \mathrm{ws}, [1, 1, .., 1]) \leftarrow \mathrm{t}(T).$$

Instantiate a blue die for each topic

$$\mathrm{sample}([\mathrm{red}, D], P) \leftarrow \mathrm{d}(D, P, W).$$

Roll the red die for every document D and position P

Latent Dirichlet Allocation with Pólya Dice

$$\mathrm{dt}([\mathrm{red}, D], \mathrm{ts}, [1, 1, .., 1]) \leftarrow \mathrm{d}(D, P, W).$$ Instantiate a red die for each document

$$\mathrm{dt}([\mathrm{blue}, T], \mathrm{ws}, [1, 1, .., 1]) \leftarrow \mathrm{t}(T).$$ Instantiate a blue die for each document

$$\mathrm{sample}([\mathrm{red}, D], P) \leftarrow \mathrm{d}(D, P, W).$$ Roll the red die for a given document D and position P

$$\mathrm{sample}([\mathrm{blue}, T], [D, P]) \leftarrow \mathrm{d}(D, P, W), \mathrm{obs}([\mathrm{red}, D], P, T).$$

For every document D and position P, **roll the blue die** that corresponds to the topic sampled by rolling the red die

Latent Dirichlet Allocation with Pólya Dice

$$\mathrm{dt}([\mathrm{red}, D], \mathrm{ts}, [1, 1, .., 1]) \leftarrow \mathrm{d}(D, P, W).$$

$$\mathrm{dt}([\mathrm{blue}, T], \mathrm{ws}, [1, 1, .., 1]) \leftarrow \mathrm{t}(T).$$

Generative Process

$$\left\{\begin{array}{l} \mathrm{sample}([\mathrm{red}, D], P) \leftarrow \mathrm{d}(D, P, W). \\ \mathrm{sample}([\mathrm{blue}, T], [D, P]) \leftarrow \mathrm{d}(D, P, W), \mathrm{obs}([\mathrm{red}, D], P, T). \end{array}\right.$$

$$\boxed{\mathrm{qa}^*(D, P, W) \leftarrow \mathrm{d}(D, P, W), \mathrm{obs}([\mathrm{blue}, T], [D, P], W).}$$

Enforce the **condition** that the generated words must match the initial words that we observed in the corpus.

The Grounding Engine

(probabilistic program)

$$\phi_{d,p,w} = \bigvee_{i=1}^{K} \left[(a_d[e_{d,p}] = t_i) \wedge (b_i[(a_d[e_{d,p}] = t_i)] = v_w) \right]$$

Grounding

APACHE ARROW Acero

$$\phi_{1,1,1} = [(a_1[e_{1,1}] = t_1) \wedge (b_1[a_1[e_{1,1}] = t_1] = v_1)] \vee [(a_1[e_{1,1}] = t_2) \wedge (b_2[a_1[e_{1,1}] = t_2] = v_1)] \vee \ldots$$
$$\vee[(a_1[e_{1,1}] = t_K) \wedge (b_K[a_1[e_{1,1}] = t_K] = v_1)]$$

$$\phi_{1,2,4} = [(a_1[e_{1,2}] = t_1) \wedge (b_1[a_1[e_{1,2}] = t_1] = v_4)] \vee [(a_1[e_{1,2}] = t_2) \wedge (b_2[a_1[e_{1,2}] = t_2] = v_4)] \vee \ldots$$
$$\vee[(a_1[e_{1,2}] = t_K) \wedge (b_k[a_1[e_{1,2}] = t_K] = v_4)]$$

$$\phi_{1,3,6} = [(a_1[e_{1,3}] = t_1) \wedge (b_1[a_1[e_{1,3}] = t_1] = v_6)] \vee [(a_1[e_{1,3}] = t_2) \wedge (b_2[a_1[e_{1,3}] = t_2] = v_6)] \vee \ldots$$
$$\vee[(a_1[e_{1,3}] = t_K) \wedge (b_k[a_1[e_{1,3}] = t_K] = v_6)]$$

…

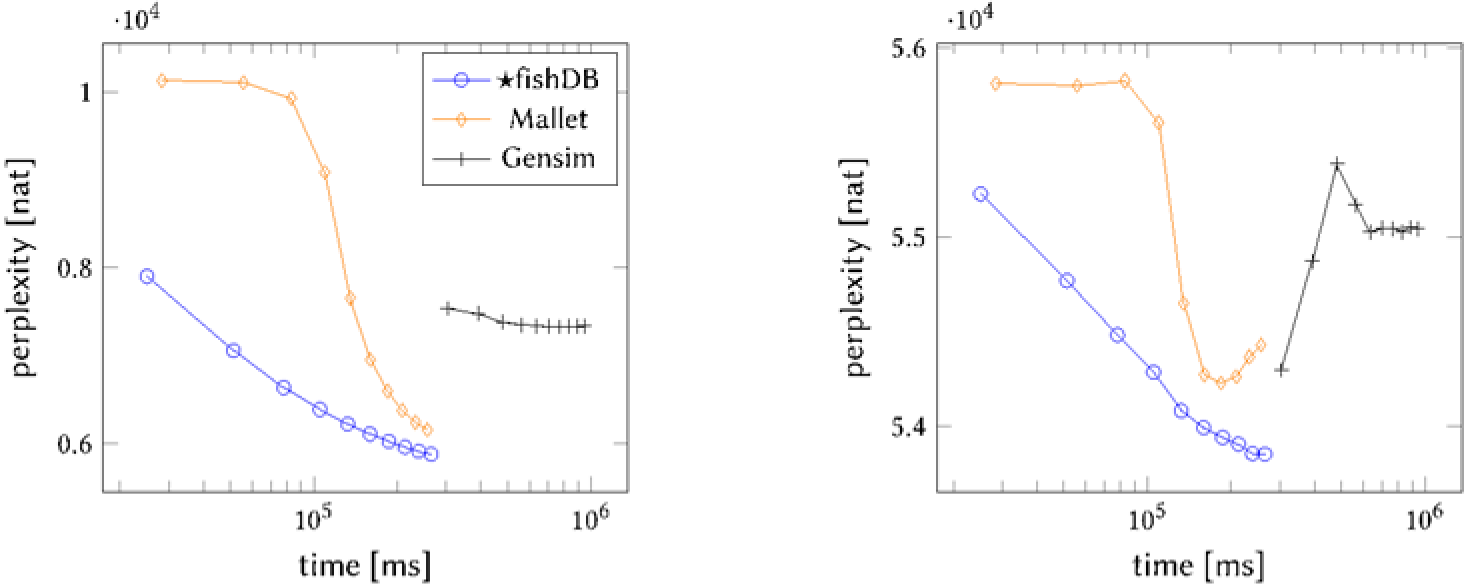
…

… and many many others

LLVM/ClangJIT[2]

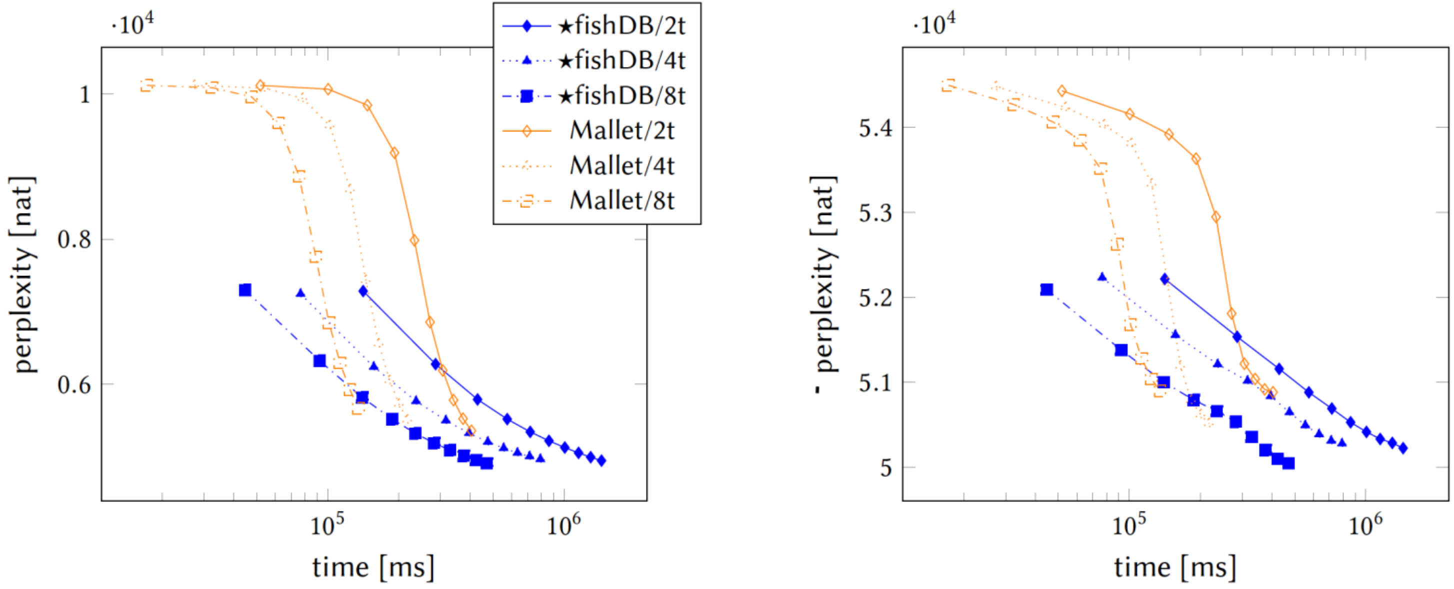
[2] Finkel et al, Clangjit: Enhancing C++ with just-in-time compilation

Experimental Evaluation: LDA



(a) Train-set NYTIMES (b) Test-set NYTIMES

Experimental Evaluation: LDA



(a) Train-Set NYTIMES

(b) Test-Set NYTIMES

LDA Multithreaded 100 topics benchmarking on NYTIMES using 2, 4 and 8 threads

Future work

a) Query-Driven Variational Inference

Blei, David M., Alp Kucukelbir, and Jon D. McAuliffe. "Variational inference: A review for statisticians." *Journal of the American statistical Association* 112.518 (2017): 859-877.

b) Fairness through rel. constraints

Salimi, Babak, Luke Rodriguez, Bill Howe, and Dan Suciu. "Interventional fairness: Causal database repair for algorithmic fairness" In *Proceedings of the 2019 International Conference on Management of Data*, pp. 793-810. 2019.

Alireza Pirhadi, Mohammad Hossein Moslemi, Alexander Cloninger, Mostafa Milani, and Babak Salimi. 2024. OTClean: Data Cleaning for Conditional Independence Violations using Optimal Transport. Proc. ACM Manag. Data 2, 3, Article 160 (June 2024)

c) Non-parametric Bayesian models

Grohe, Martin, and Peter Lindner. "Infinite probabilistic databases." *Logical Methods in Computer Science* 18 (2022).

Thank You ^^

Any Questions?

References

[1] Bárány, Vince, Balder Ten Cate, Benny Kimelfeld, Dan Olteanu, and Zografoula Vagena. "Declarative probabilistic programming with datalog." *ACM Transactions on Database Systems (TODS)* 42, no. 4 (2017): 1-35

[2] Finkel, Hal, David Poliakoff, Jean-Sylvain Camier, and David F. Richards. "Clangjit: Enhancing C++ with just-in-time compilation." In *2019 IEEE/ACM International Workshop on Performance, Portability and Productivity in HPC (P3HPC)*, pp. 82-95. IEEE, 2019.

[3] Blei, David M., Andrew Y. Ng, and Michael I. Jordan. "Latent Dirichlet allocation." Journal of machine Learning research 3, no. Jan (2003): 993-1022.