Scaling Bayesian inference by constructing approximating exponential families

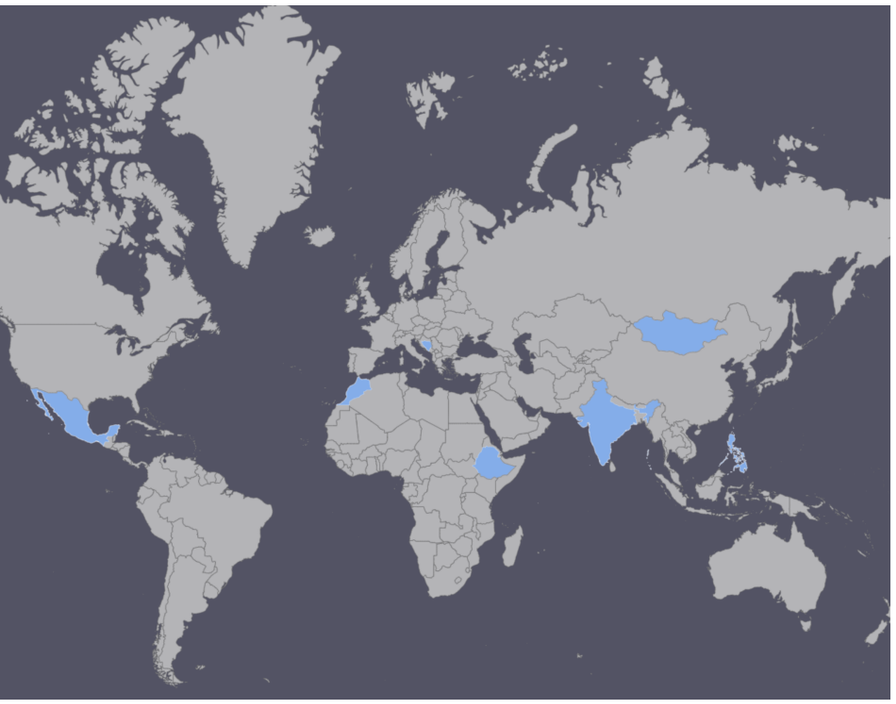
Jonathan H. Huggins
Postdoctoral Research Fellow
Department of Biostatistics, Harvard

With: Ryan Adams, Tamara Broderick, James Zou

Bayesian Inference

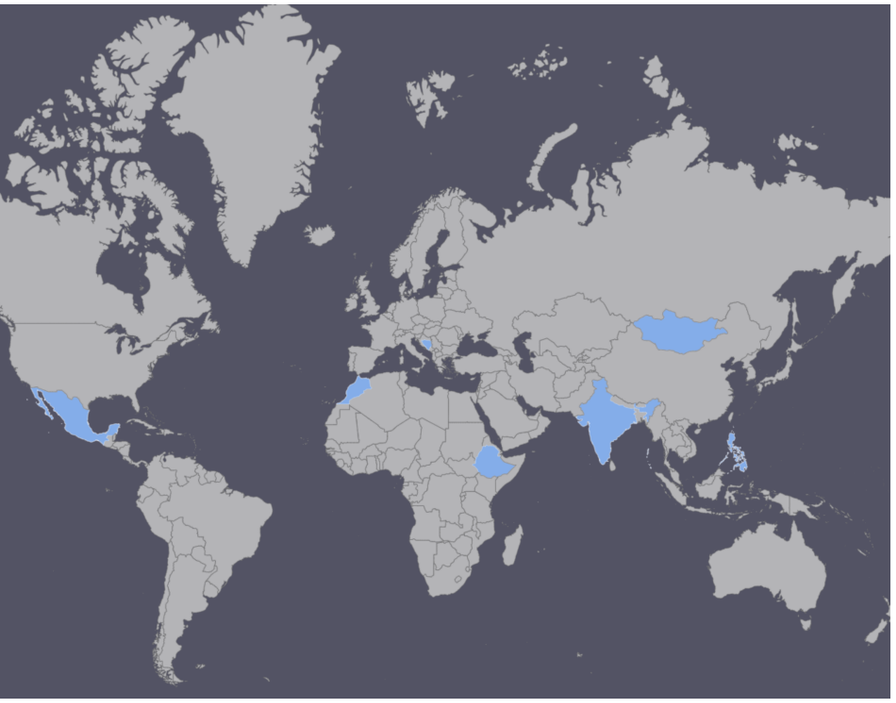
Bayesian Inference

Microcredit

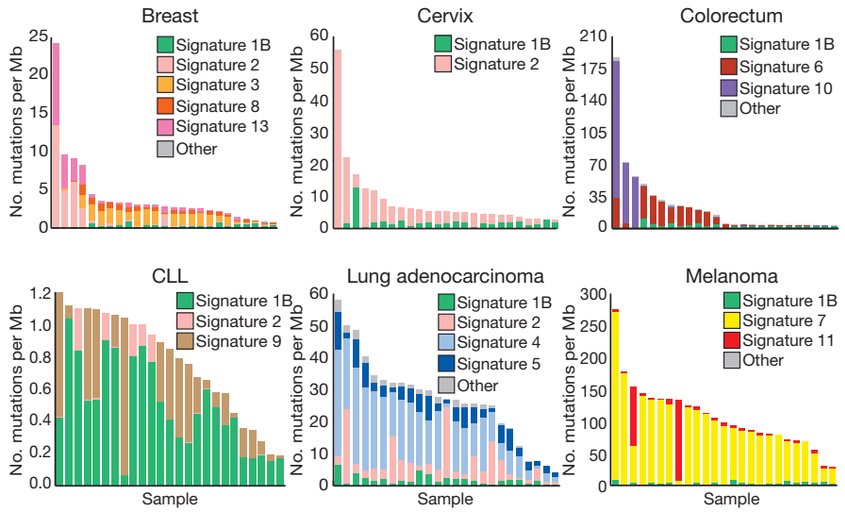


Bayesian Inference

Microcredit

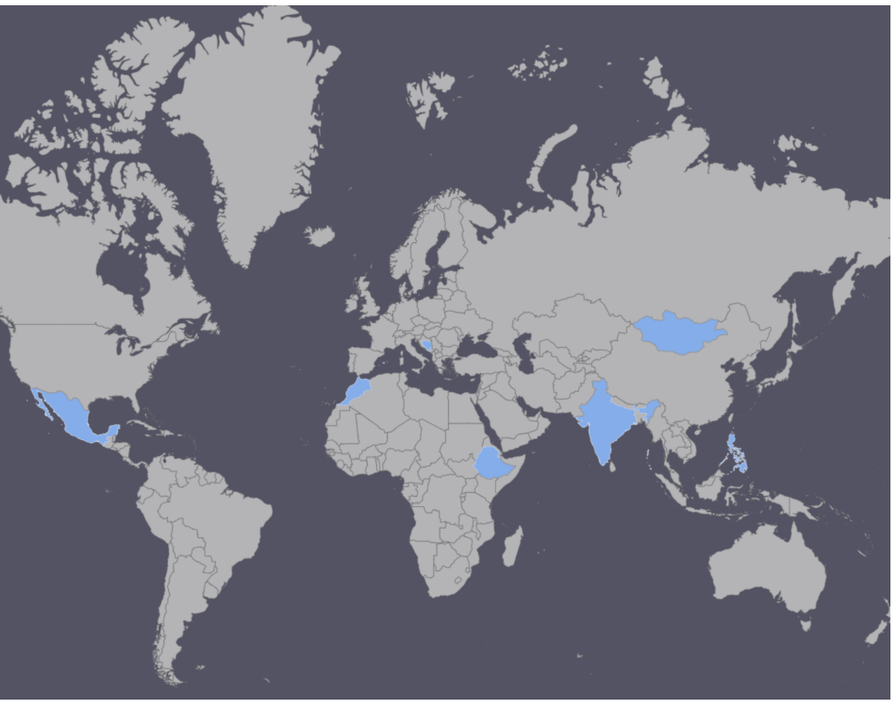


Cancer genomics

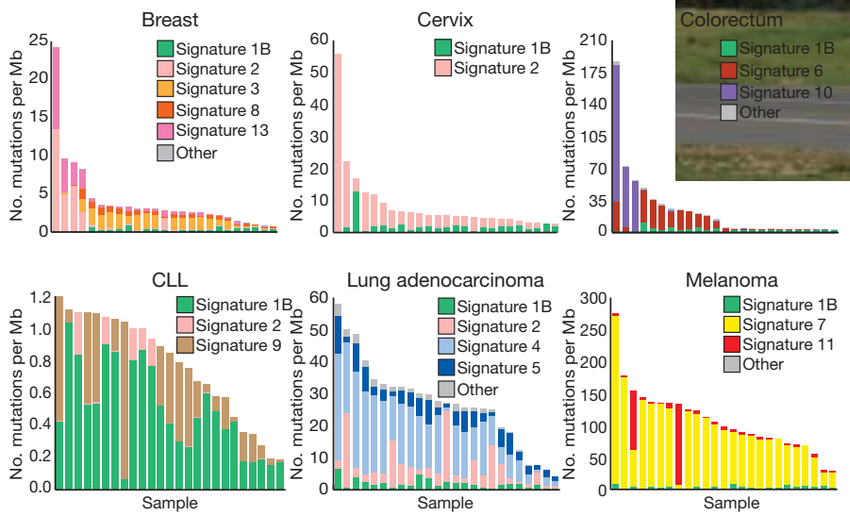


Bayesian Inference

Microcredit



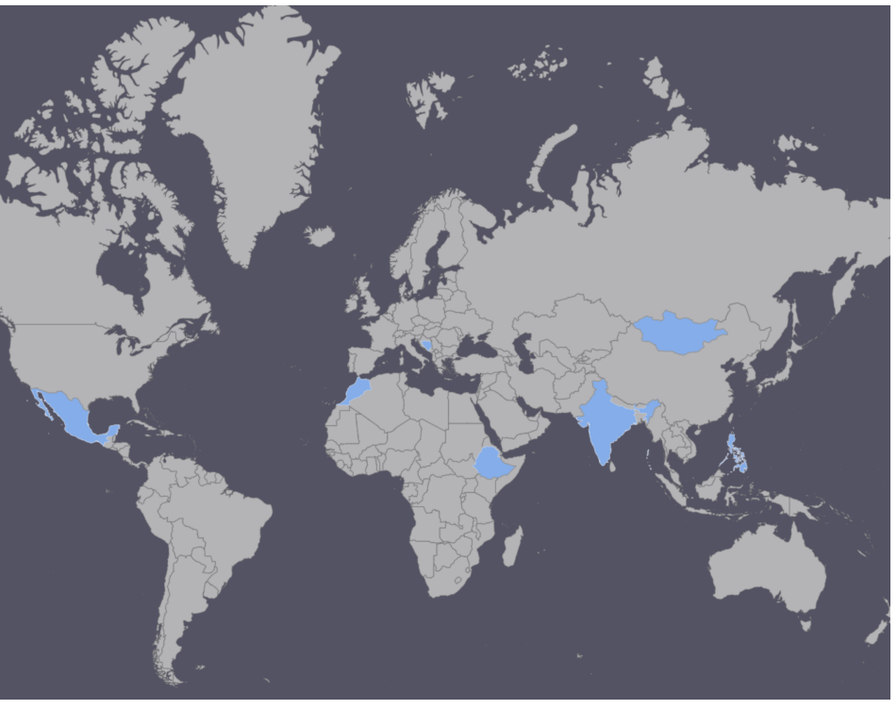
Cancer genomics



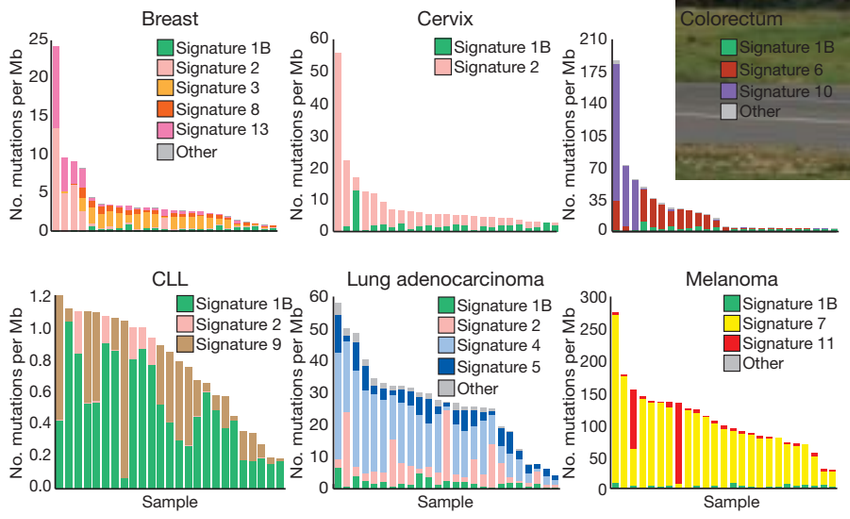
Fuel consumption

Bayesian Inference

Microcredit



Cancer genomics

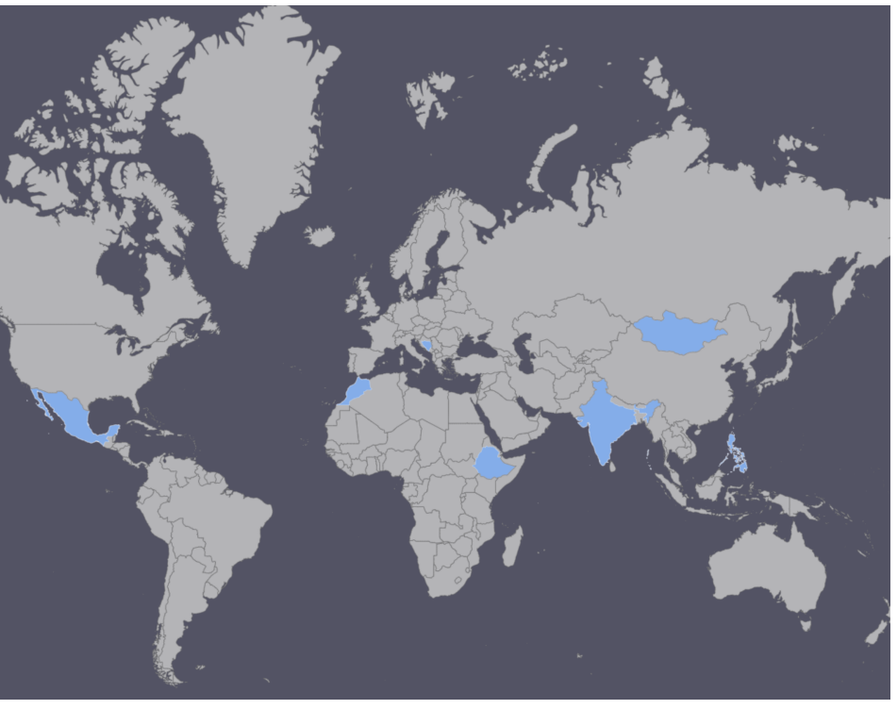


Fuel consumption

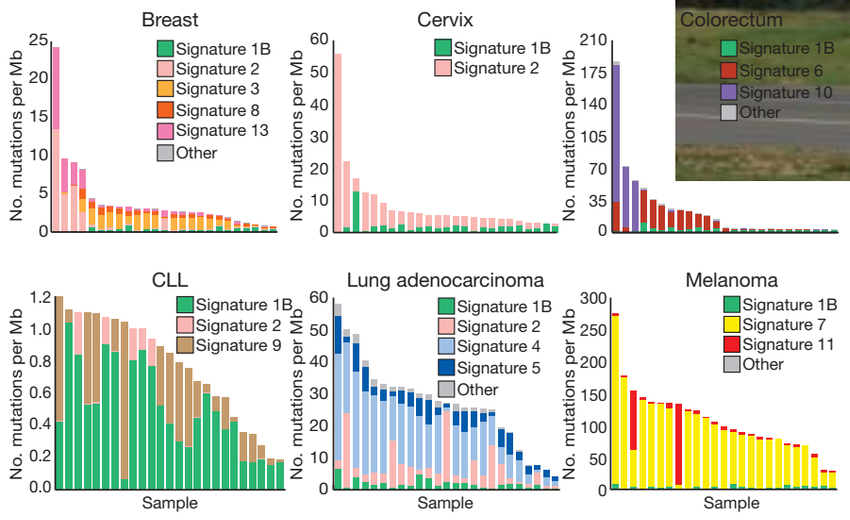
- Challenge: existing methods slow (and/or tedious, unreliable)

Bayesian Inference

Microcredit



Cancer genomics

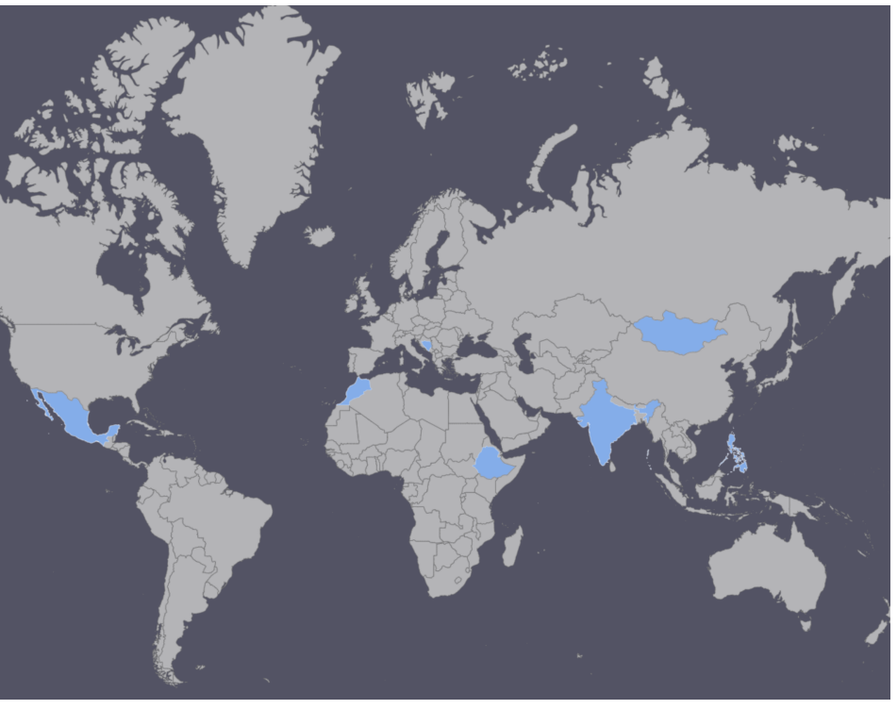


Fuel consumption

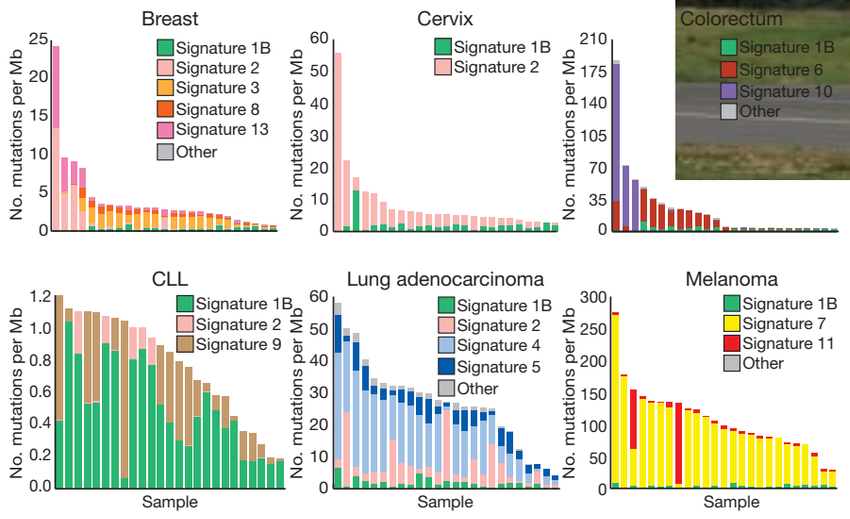
- Challenge: existing methods slow (and/or tedious, unreliable)
- Our proposal: use **efficient summaries** of data

Bayesian Inference

Microcredit



Cancer genomics



Fuel consumption

- Challenge: existing methods slow (and/or tedious, unreliable)
- Our proposal: use **efficient summaries** of data
- Approximate sufficient statistics for **simple, scalable** Bayesian inference with **error bounds for finite data**

Roadmap

- Approximate Bayes review
- Likelihood approximation and dataset compression
- Approximate sufficient statistics
- Accuracy guarantees

Roadmap

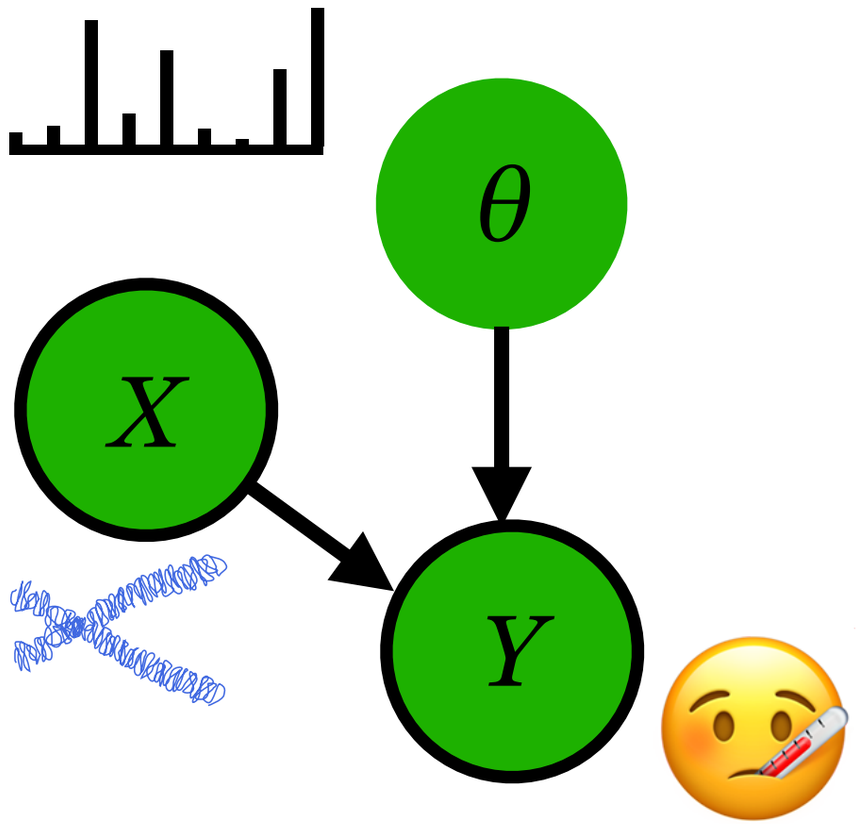
- Approximate Bayes review
- Likelihood approximation and dataset compression
- Approximate sufficient statistics
- Accuracy guarantees

Bayesian inference is challenging…

$$\pi(\theta|Y) = \frac{p(Y|\theta)\pi_0(\theta)}{Z}$$

Bayesian inference is challenging…

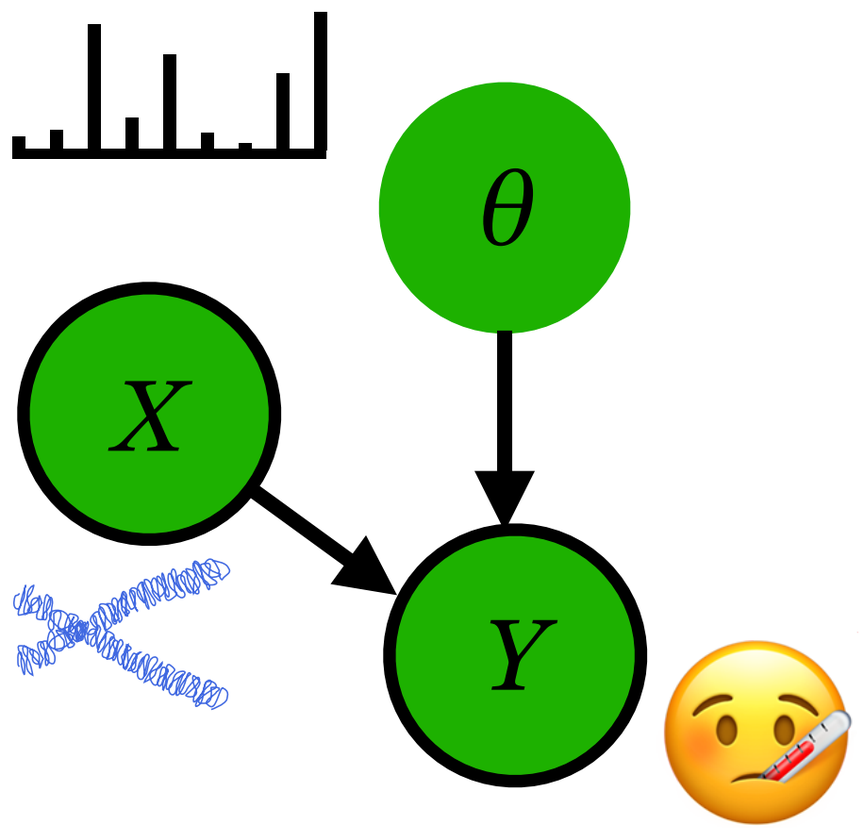
$$\pi(\theta|Y) = \frac{p(Y|\theta)\pi_0(\theta)}{Z}$$



Bayesian inference is challenging…

$$\pi(\theta|Y) = \frac{p(Y|\theta)\pi_0(\theta)}{Z}$$

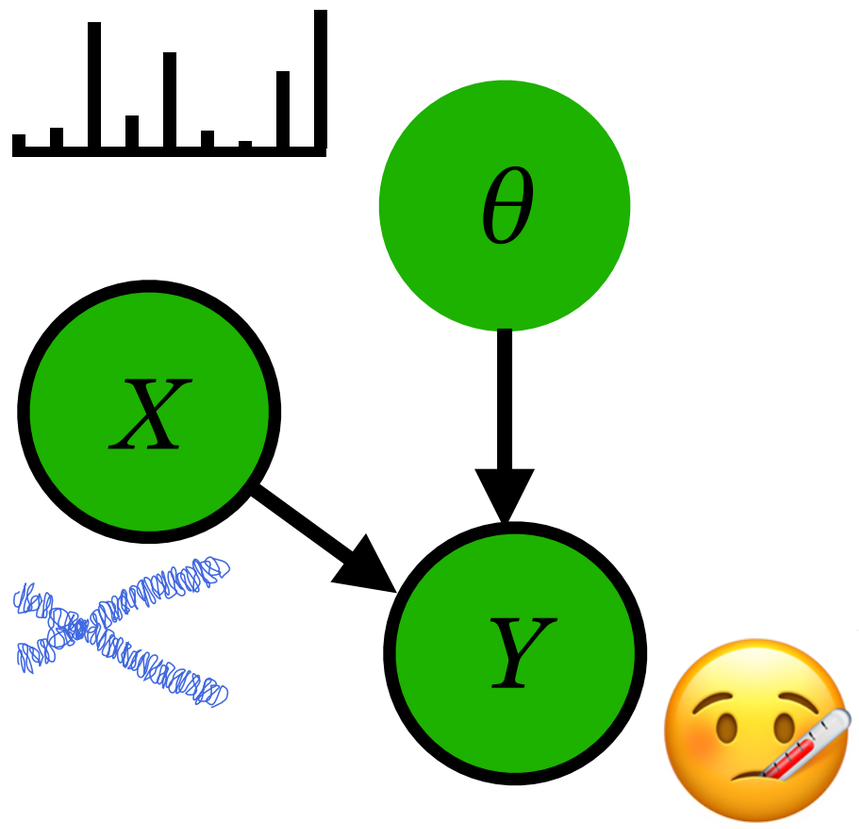
- calculate…



Bayesian inference is challenging…

$$\pi(\theta|Y) = \frac{p(Y|\theta)\pi_0(\theta)}{Z}$$

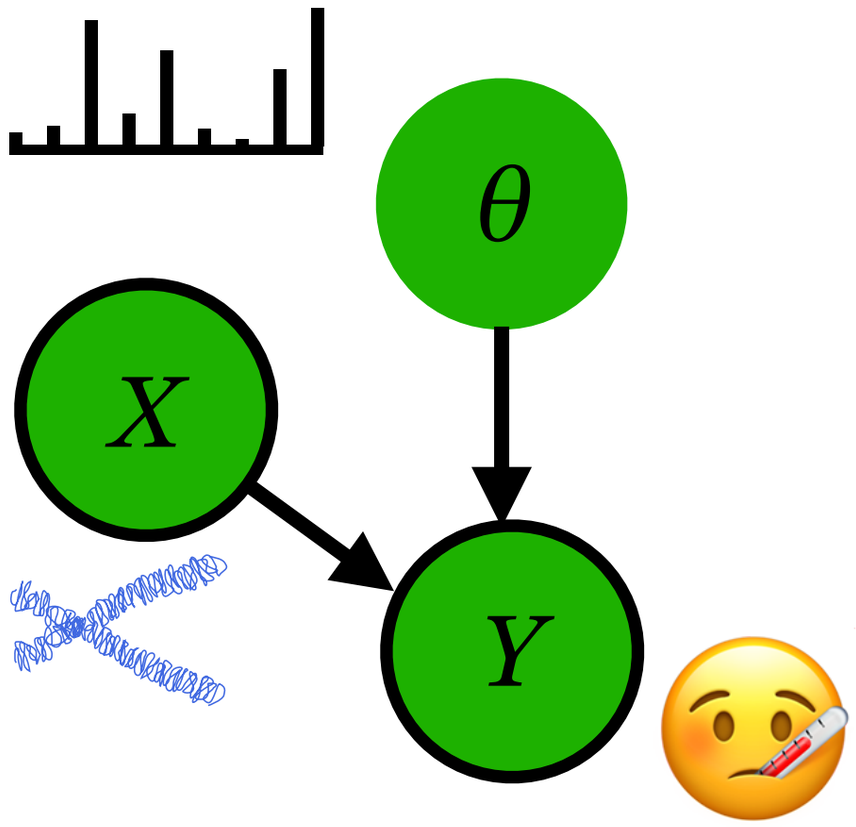
- calculate…
  - mean($\theta_{20} \mid Y$)



Bayesian inference is challenging…

$$\pi(\theta|Y) = \frac{p(Y|\theta)\pi_0(\theta)}{Z}$$

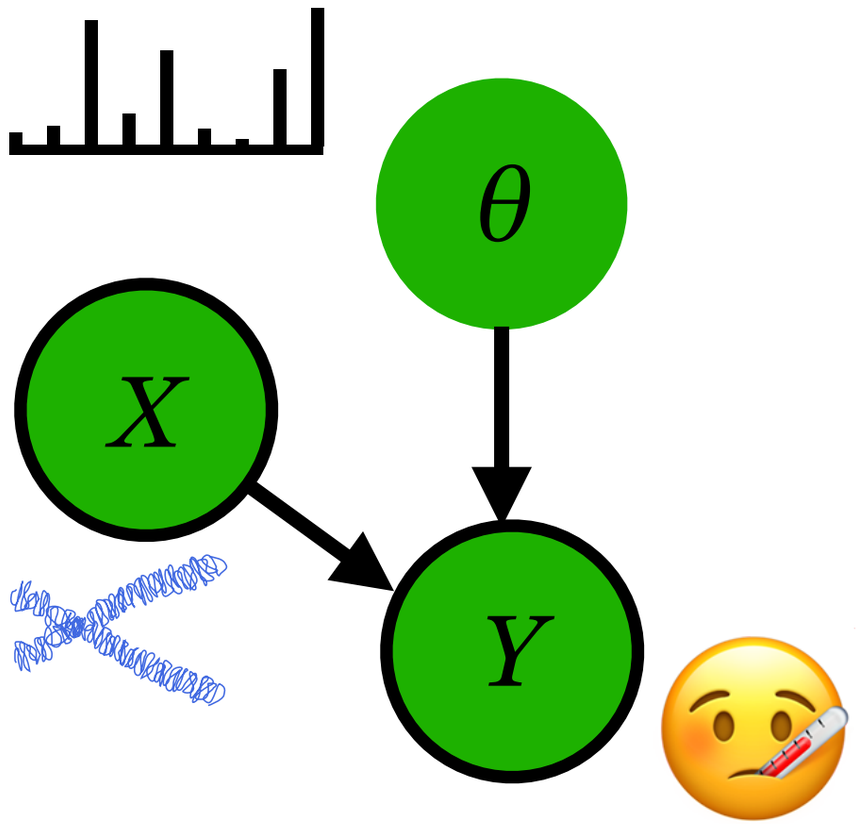
- calculate…
  - mean($\theta_{20} \mid Y$)
  - var($\theta_{20} \mid Y$)



Bayesian inference is challenging…

$$\pi(\theta|Y) = \frac{p(Y|\theta)\pi_0(\theta)}{Z}$$

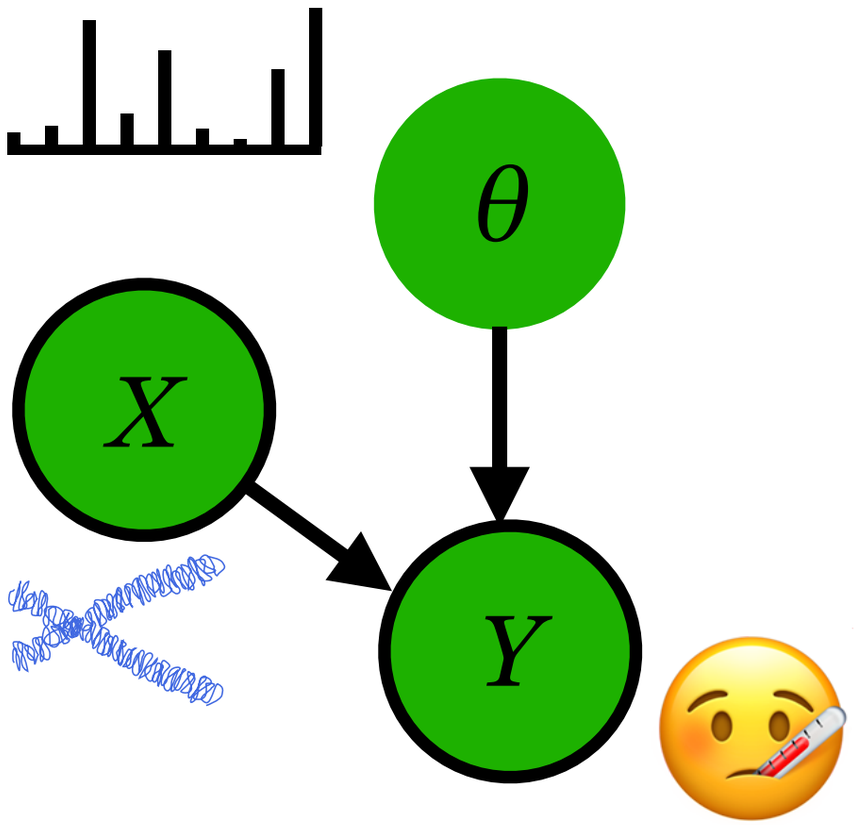
- calculate…
  - mean($\theta_{20} \mid Y$)
  - var($\theta_{20} \mid Y$)
  - Pr[$\theta_{20} < .1 \mid Y$]



Bayesian inference is challenging…

$$\pi(\theta|Y) = \frac{p(Y|\theta)\pi_0(\theta)}{Z}$$

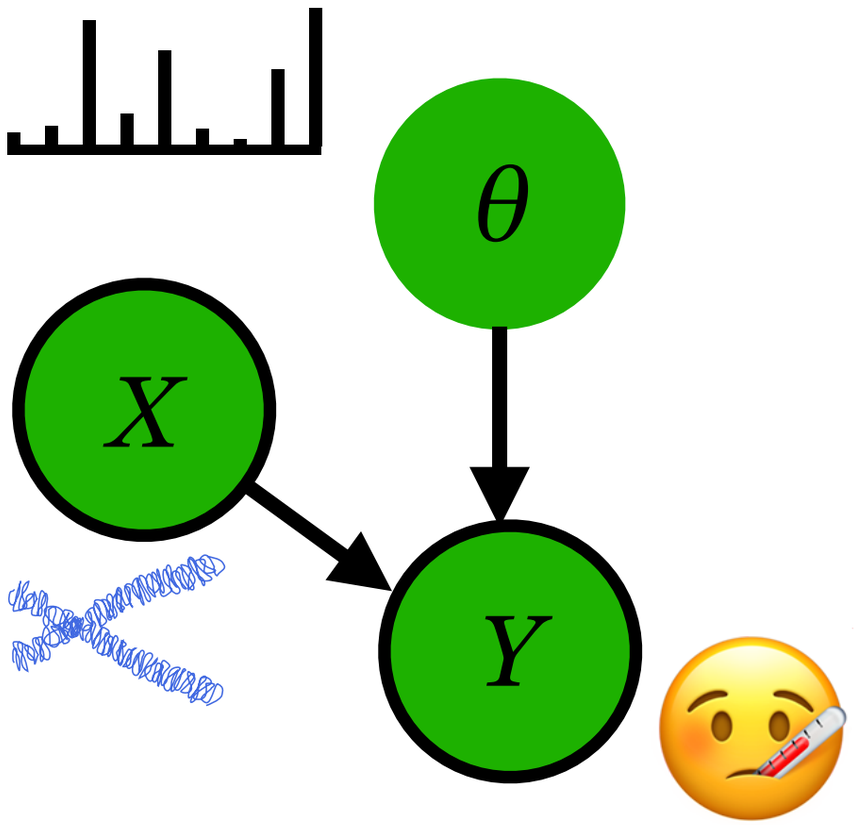
- calculate…
  - mean($\theta_{20} \mid Y$)
  - var($\theta_{20} \mid Y$)
  - Pr[$\theta_{20} < .1 \mid Y$]
- **Goal:** compute expectations wrt π



Bayesian inference is challenging…

$$\pi(\theta|Y) = \frac{p(Y|\theta)\pi_0(\theta)}{Z}$$

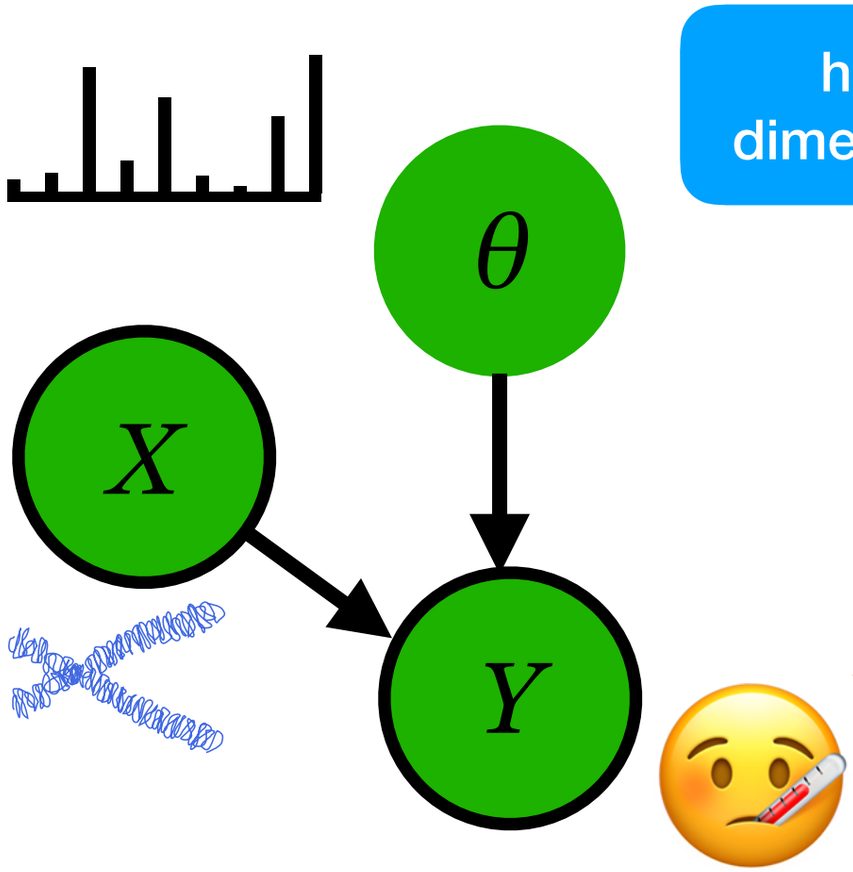
- calculate…
  - mean($\theta_{20} \mid Y$)
  - var($\theta_{20} \mid Y$)
  - Pr[$\theta_{20} < .1 \mid Y$]
- **Goal:** compute expectations wrt π
- **A hard problem!**



Bayesian inference is challenging…

$$\pi(\theta|Y) = \frac{p(Y|\theta)\pi_0(\theta)}{Z}$$

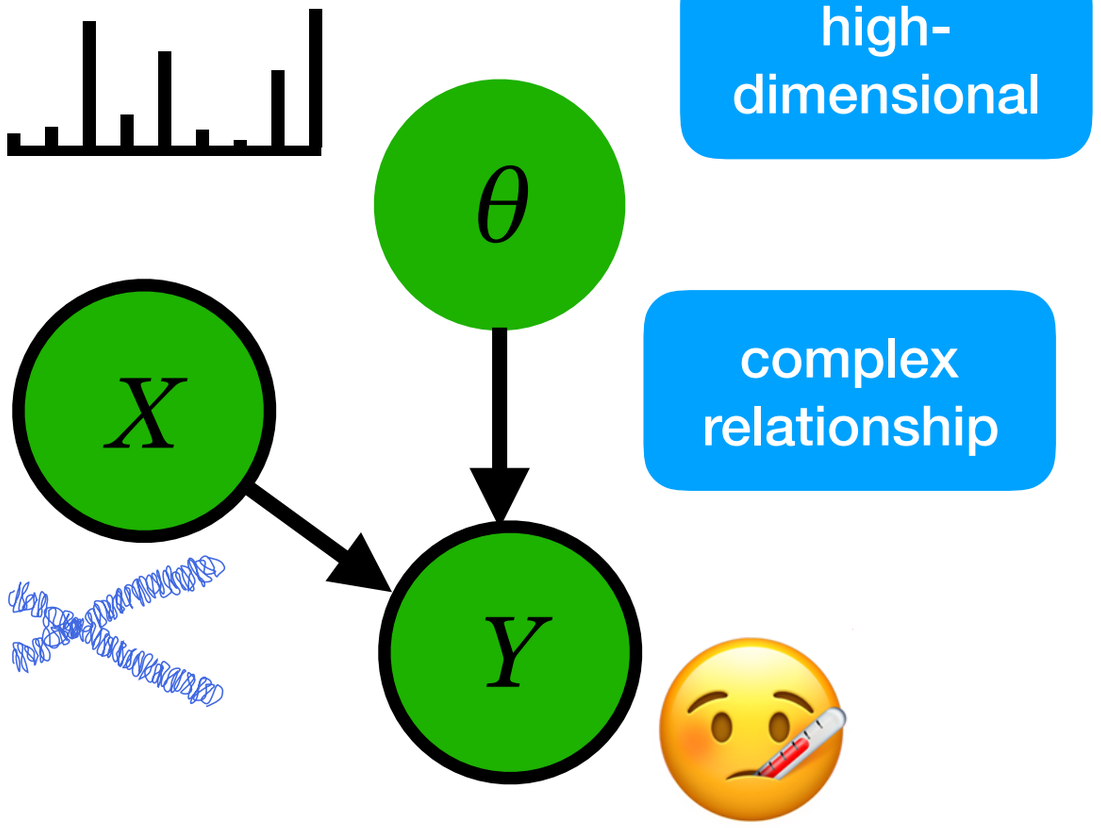
- calculate…
  - mean($\theta_{20} \mid Y$)
  - var($\theta_{20} \mid Y$)
  - Pr[$\theta_{20} < .1 \mid Y$]
- **Goal:** compute expectations wrt π
- **A hard problem!**



Bayesian inference is challenging…

$$\pi(\theta|Y) = \frac{p(Y|\theta)\pi_0(\theta)}{Z}$$

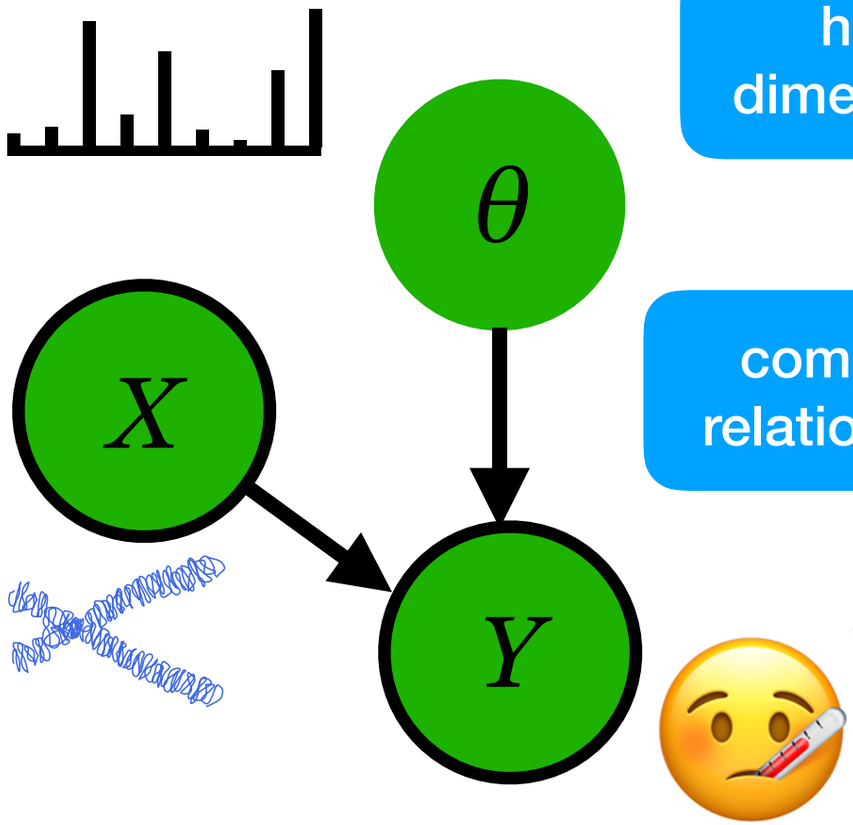
- calculate…
  - mean($\theta_{20} \mid Y$)
  - var($\theta_{20} \mid Y$)
  - Pr[$\theta_{20} < .1 \mid Y$]
- **Goal:** compute expectations wrt π
- **A hard problem!**



Bayesian inference is challenging…

$$\pi(\theta|Y) = \frac{p(Y|\theta)\pi_0(\theta)}{Z}$$

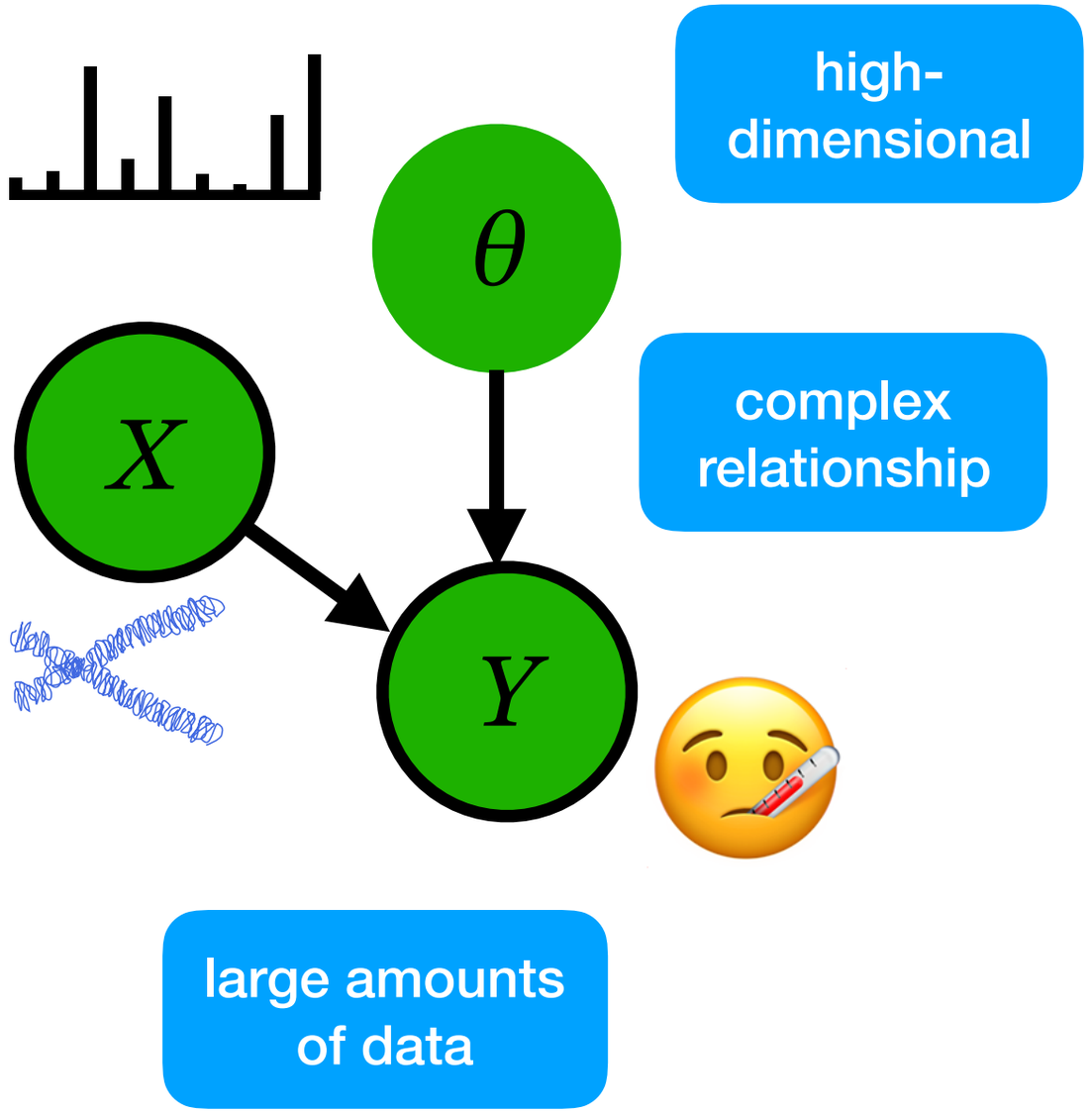
- calculate…
  - mean($\theta_{20} \mid Y$)
  - var($\theta_{20} \mid Y$)
  - Pr[$\theta_{20} < .1 \mid Y$]
- **Goal:** compute expectations wrt π
- **A hard problem!**



Bayesian inference is challenging…

$$\pi(\theta|Y) = \frac{p(Y|\theta)\pi_0(\theta)}{Z}$$

- calculate…
  - mean($\theta_{20} \mid Y$)
  - var($\theta_{20} \mid Y$)
  - Pr[$\theta_{20} < .1 \mid Y$]
- **Goal:** compute expectations wrt π
- **A hard problem!**
- **Solution:** approximate π



What do we want from an approximate inference algorithm?

What do we want from an approximate inference algorithm?

1. Scalability

What do we want from an approximate inference algorithm?

1. Scalability

    - large datasets

What do we want from an approximate inference algorithm?

1. Scalability

    - large datasets
    - streaming and distributed data

What do we want from an approximate inference algorithm?

1. Scalability

    - large datasets
    - streaming and distributed data
    - moderate-sized data with complex models

What do we want from an approximate inference algorithm?

1. Scalability

    - large datasets
    - streaming and distributed data
    - moderate-sized data with complex models

2. Arbitrary accuracy: $d(\pi, \pi_{\text{approx}}) \to 0$

What do we want from an approximate inference algorithm?

1. Scalability

    - large datasets
    - streaming and distributed data
    - moderate-sized data with complex models

2. Arbitrary accuracy: $d(\pi, \pi_{\text{approx}}) \to 0$

3. Validation of approximation quality: $d(\pi, \pi_{\text{approx}}) < c$

Modern Bayesian inference

1. Scalability

2. Arbitrary accuracy

3. Validation of approximation quality

Modern Bayesian inference

1. Scalability

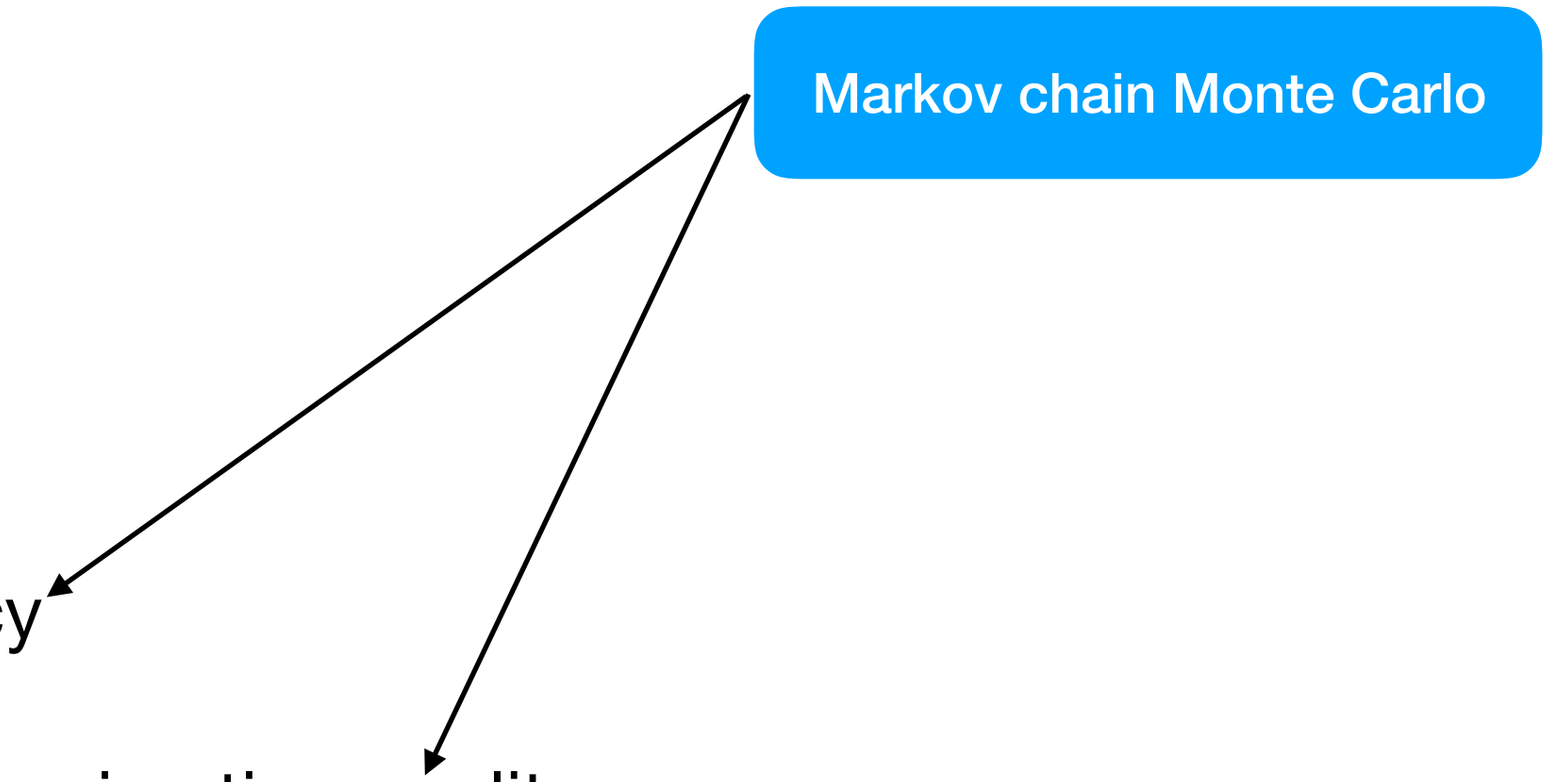
Markov chain Monte Carlo

2. Arbitrary accuracy

3. Validation of approximation quality

Modern Bayesian inference

1. Scalability
2. Arbitrary accuracy
3. Validation of approximation quality



Modern Bayesian inference

1. Scalability

Markov chain Monte Carlo

subsampling MCMC

2. Arbitrary accuracy

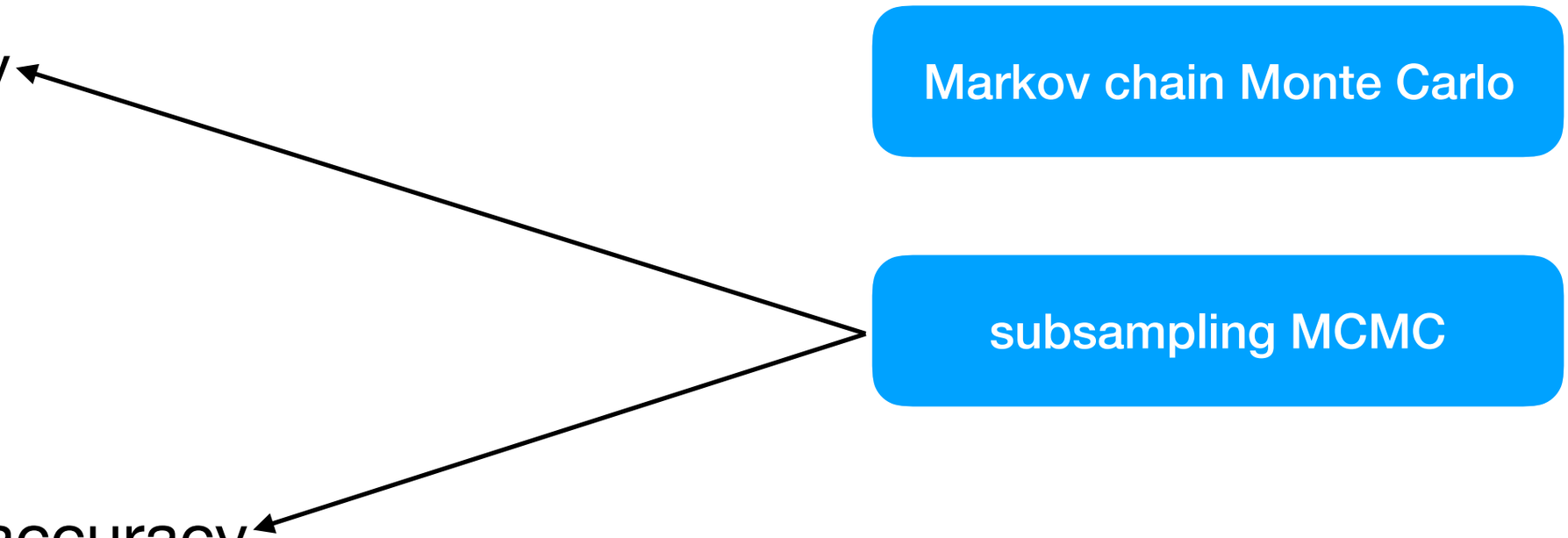
3. Validation of approximation quality

Modern Bayesian inference

1. Scalability

2. Arbitrary accuracy

3. Validation of approximation quality



Modern Bayesian inference

1. Scalability

Markov chain Monte Carlo

subsampling MCMC

2. Arbitrary accuracy

variational Bayes

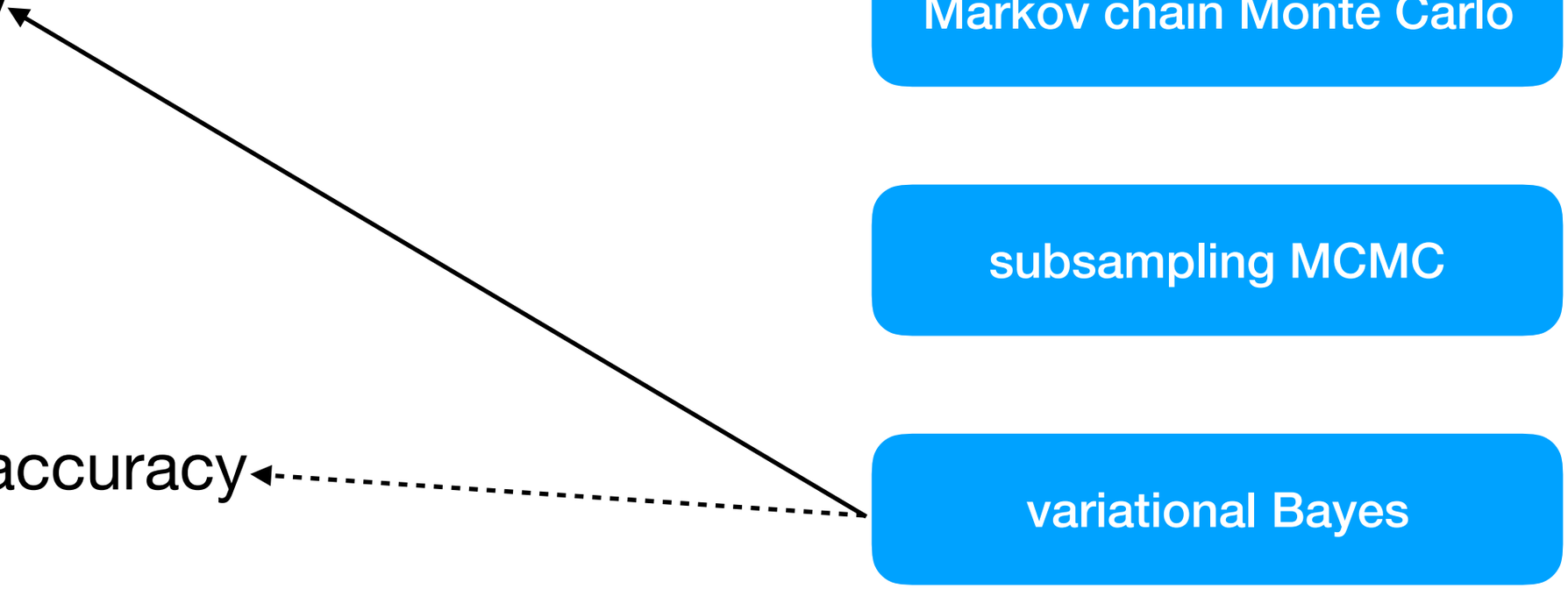
3. Validation of approximation quality

Modern Bayesian inference

1. Scalability

2. Arbitrary accuracy

3. Validation of approximation quality



Modern Bayesian inference

1. Scalability

Markov chain Monte Carlo

subsampling MCMC

2. Arbitrary accuracy

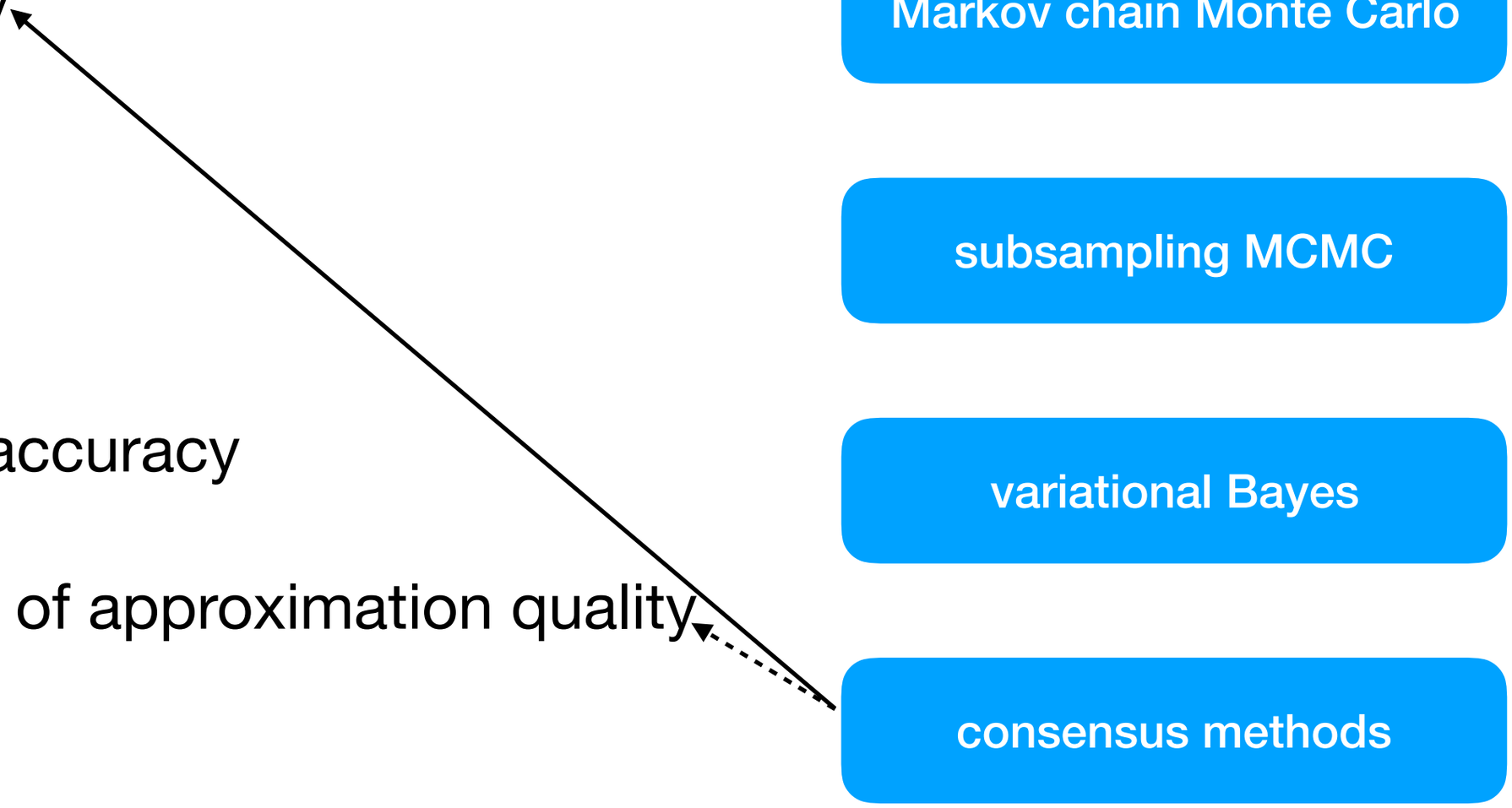
variational Bayes

3. Validation of approximation quality

consensus methods

Modern Bayesian inference

1. Scalability
2. Arbitrary accuracy
3. Validation of approximation quality



Modern Bayesian inference

1. Scalability

Markov chain Monte Carlo

subsampling MCMC

2. Arbitrary accuracy

variational Bayes

3. Validation of approximation quality

consensus methods

this talk: we get all three!

Roadmap

- Approximate Bayes review
- Likelihood approximation and dataset compression
- Approximate sufficient statistics
- Accuracy guarantees

Key idea: likelihood approximation

Key idea: likelihood approximation

Monte Carlo

Key idea: likelihood approximation

Monte Carlo $\mathbb{E}[f(\theta) \mid Y] \approx T^{-1} \sum_{t=1}^{T} f(\theta_t)$

Key idea: likelihood approximation

Monte Carlo $\mathbb{E}[f(\theta) \mid Y] \approx T^{-1} \sum_{t=1}^{T} f(\theta_t) \qquad \theta_t \overset{\text{i.i.d.}}{\sim} \pi(\theta \mid Y)$

Key idea: likelihood approximation

Monte Carlo $\mathbb{E}[f(\theta) \mid Y] \approx T^{-1} \sum_{t=1}^{T} f(\theta_t) \qquad \theta_t \overset{\text{i.i.d.}}{\sim} \pi(\theta \mid Y)$

Markov chain Monte Carlo (MCMC)

Key idea: likelihood approximation

Monte Carlo $\mathbb{E}[f(\theta) \mid Y] \approx T^{-1} \sum_{t=1}^{T} f(\theta_t) \quad \theta_t \overset{\text{i.i.d.}}{\sim} \pi(\theta \mid Y)$

Markov chain Monte Carlo (MCMC) $\theta_t \; not$ independent

Key idea: likelihood approximation

Monte Carlo $\mathbb{E}[f(\theta) \mid Y] \approx T^{-1} \sum_{t=1}^{T} f(\theta_t) \quad \theta_t \overset{\text{i.i.d.}}{\sim} \pi(\theta \mid Y)$

Markov chain Monte Carlo (MCMC) $\theta_t \; not \; \text{independent}$

θ_0

Key idea: likelihood approximation

Monte Carlo $\mathbb{E}[f(\theta) \mid Y] \approx T^{-1} \sum_{t=1}^{T} f(\theta_t) \quad \theta_t \overset{\text{i.i.d.}}{\sim} \pi(\theta \mid Y)$

Markov chain Monte Carlo (MCMC) $\theta_t \; not$ independent

Key idea: likelihood approximation

Monte Carlo $\mathbb{E}[f(\theta) \mid Y] \approx T^{-1} \sum_{t=1}^{T} f(\theta_t) \quad \theta_t \overset{\text{i.i.d.}}{\sim} \pi(\theta \mid Y)$

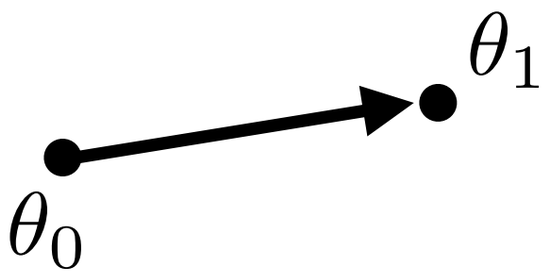
Markov chain Monte Carlo (MCMC) $\theta_t \; not$ independent

evaluate $\log p(Y \mid \theta)\pi_0(\theta)$ at θ_0 and θ_1'

Key idea: likelihood approximation

Monte Carlo $\mathbb{E}[f(\theta) \mid Y] \approx T^{-1} \sum_{t=1}^{T} f(\theta_t) \quad \theta_t \overset{\text{i.i.d.}}{\sim} \pi(\theta \mid Y)$

Markov chain Monte Carlo (MCMC) $\theta_t \; not$ independent

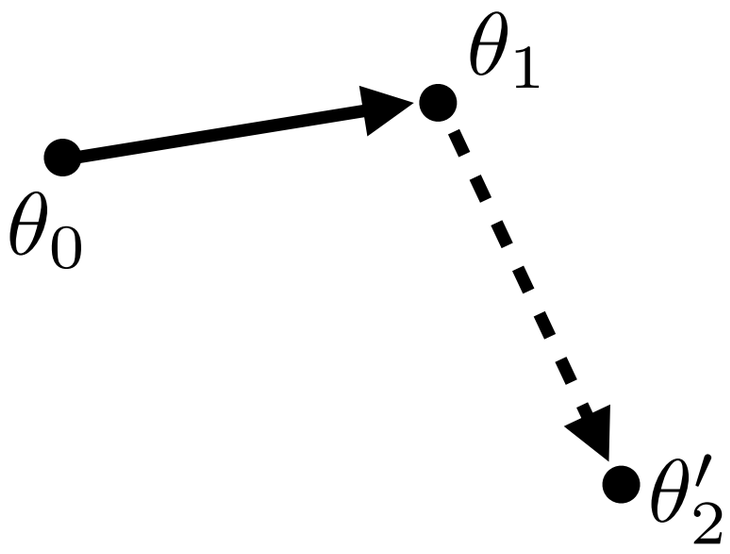


evaluate $\log p(Y \mid \theta)\pi_0(\theta)$ at θ_0 and θ_1'

Key idea: likelihood approximation

Monte Carlo $\mathbb{E}[f(\theta) \mid Y] \approx T^{-1} \sum_{t=1}^{T} f(\theta_t) \quad \theta_t \overset{\text{i.i.d.}}{\sim} \pi(\theta \mid Y)$

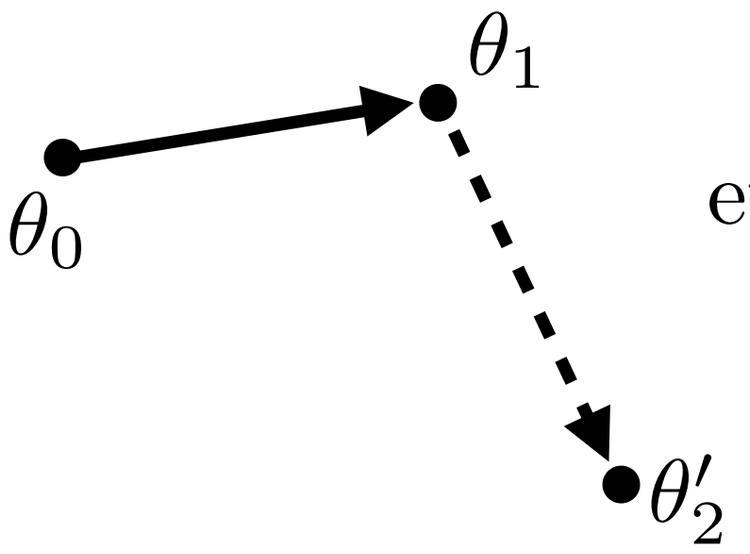
Markov chain Monte Carlo (MCMC) $\theta_t \; \textit{not} \; \text{independent}$



Key idea: likelihood approximation

Monte Carlo $\mathbb{E}[f(\theta) \mid Y] \approx T^{-1} \sum_{t=1}^{T} f(\theta_t) \quad \theta_t \overset{\text{i.i.d.}}{\sim} \pi(\theta \mid Y)$

Markov chain Monte Carlo (MCMC) $\theta_t \; not$ independent

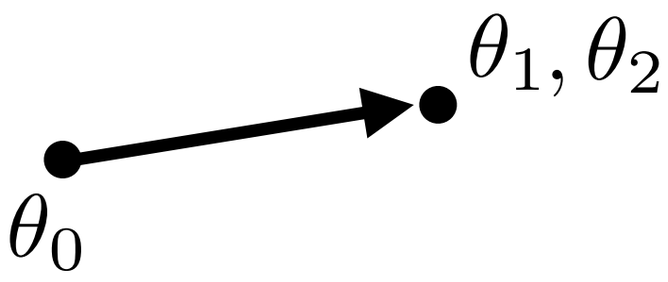


evaluate $\log p(Y \mid \theta)\pi_0(\theta)$ at θ_1 and θ_2'

Key idea: likelihood approximation

Monte Carlo $\mathbb{E}[f(\theta) \mid Y] \approx T^{-1} \sum_{t=1}^{T} f(\theta_t) \quad \theta_t \overset{\text{i.i.d.}}{\sim} \pi(\theta \mid Y)$

Markov chain Monte Carlo (MCMC) $\theta_t \; not$ independent

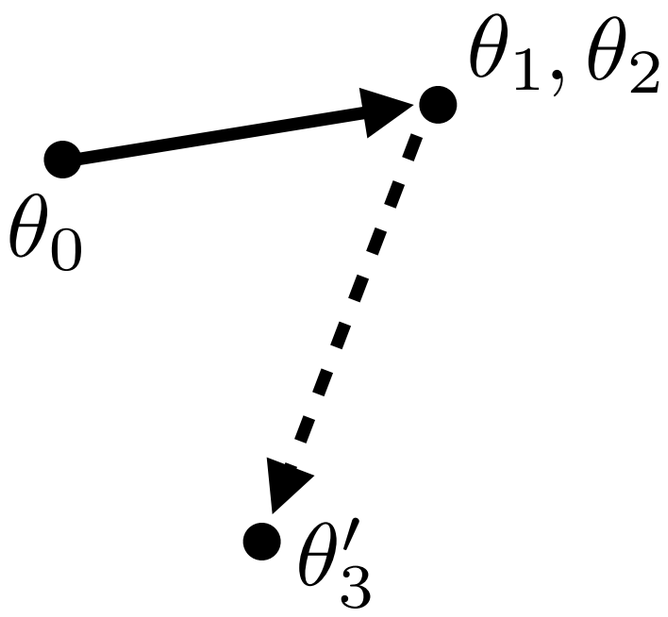


evaluate $\log p(Y \mid \theta)\pi_0(\theta)$ at θ_1 and θ_2'

Key idea: likelihood approximation

Monte Carlo $\mathbb{E}[f(\theta) \mid Y] \approx T^{-1} \sum_{t=1}^{T} f(\theta_t) \quad \theta_t \overset{\text{i.i.d.}}{\sim} \pi(\theta \mid Y)$

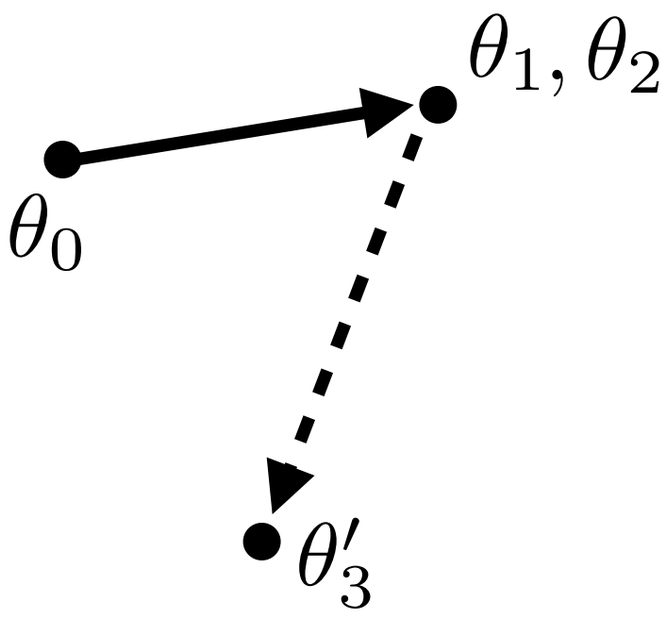
Markov chain Monte Carlo (MCMC) $\theta_t \; not$ independent



Key idea: likelihood approximation

Monte Carlo $\mathbb{E}[f(\theta) \mid Y] \approx T^{-1} \sum_{t=1}^{T} f(\theta_t) \quad \theta_t \overset{\text{i.i.d.}}{\sim} \pi(\theta \mid Y)$

Markov chain Monte Carlo (MCMC) $\theta_t \; not$ independent

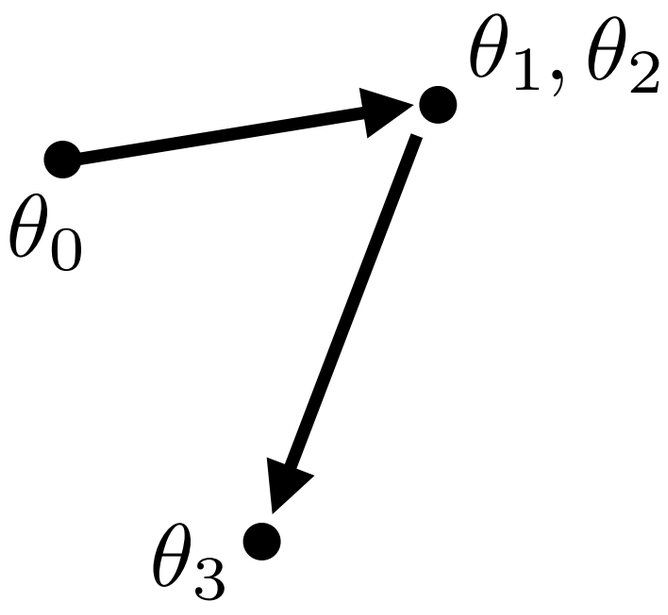


evaluate $\log p(Y \mid \theta)\pi_0(\theta)$ at θ_2 and θ_3'

Key idea: likelihood approximation

Monte Carlo $\mathbb{E}[f(\theta) \mid Y] \approx T^{-1} \sum_{t=1}^{T} f(\theta_t) \quad \theta_t \overset{\text{i.i.d.}}{\sim} \pi(\theta \mid Y)$

Markov chain Monte Carlo (MCMC) $\theta_t \; not$ independent

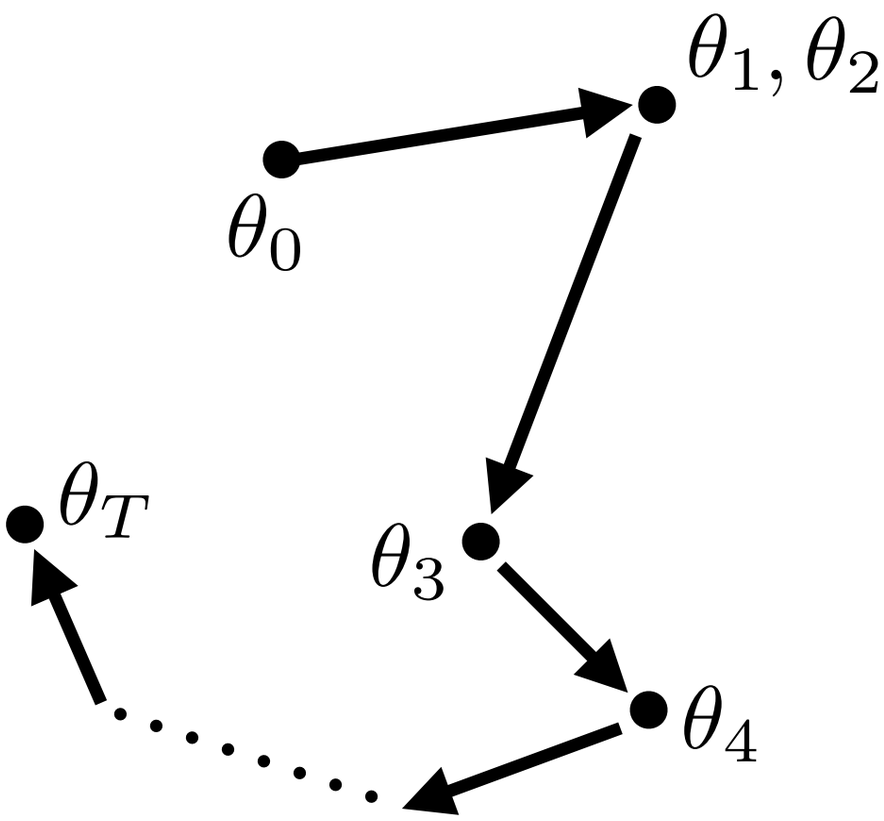


evaluate $\log p(Y \mid \theta)\pi_0(\theta)$ at θ_2 and θ_3'

Key idea: likelihood approximation

Monte Carlo $\mathbb{E}[f(\theta) \mid Y] \approx T^{-1} \sum_{t=1}^{T} f(\theta_t) \quad \theta_t \overset{\text{i.i.d.}}{\sim} \pi(\theta \mid Y)$

Markov chain Monte Carlo (MCMC) $\theta_t \; not$ independent

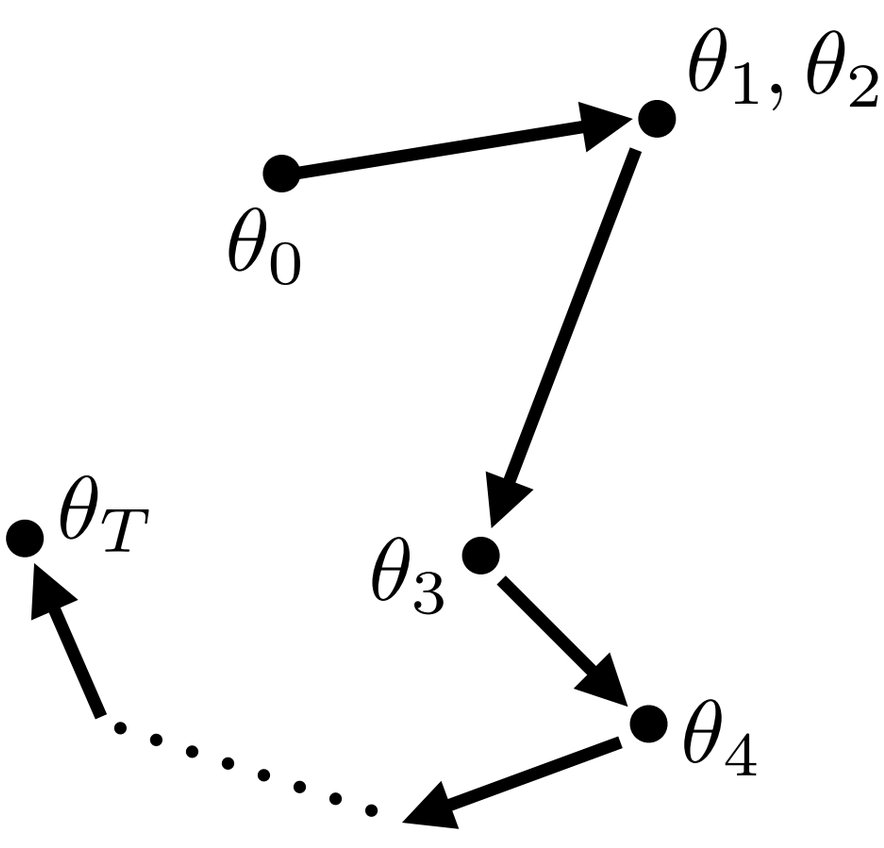


Key idea: likelihood approximation

Monte Carlo $\mathbb{E}[f(\theta) \mid Y] \approx T^{-1} \sum_{t=1}^{T} f(\theta_t) \quad \theta_t \overset{\text{i.i.d.}}{\sim} \pi(\theta \mid Y)$

Markov chain Monte Carlo (MCMC) $\theta_t \; not \; \text{independent}$

$$Y = \{y_1, y_2, \ldots, y_N\}$$

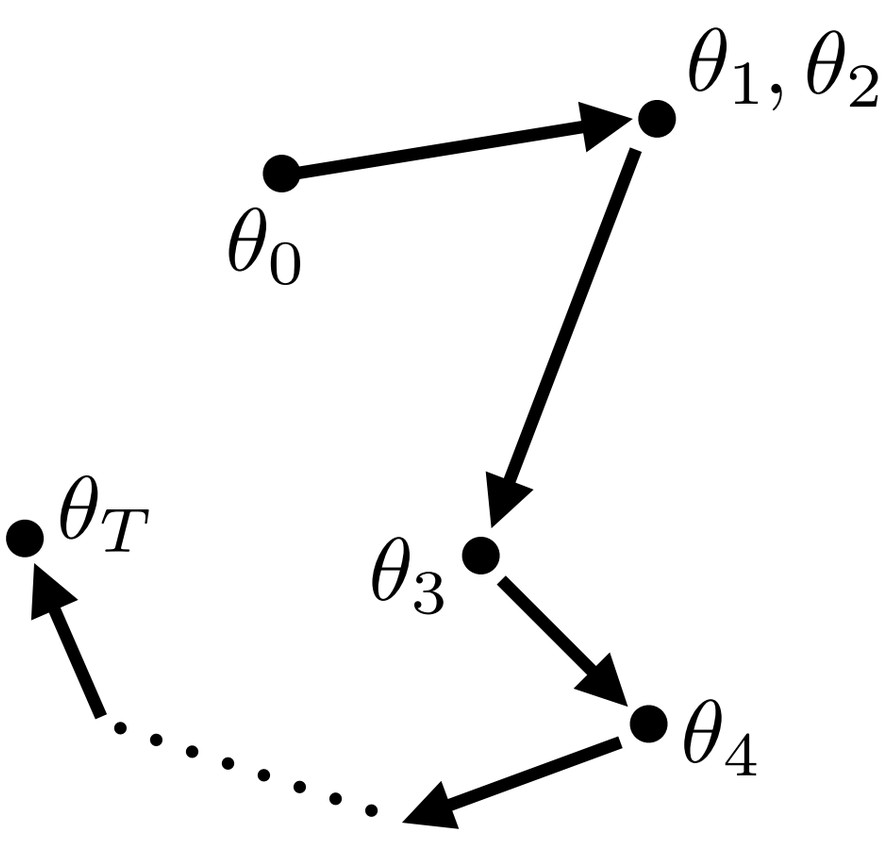


Key idea: likelihood approximation

Monte Carlo $\mathbb{E}[f(\theta) \mid Y] \approx T^{-1} \sum_{t=1}^{T} f(\theta_t) \quad \theta_t \overset{\text{i.i.d.}}{\sim} \pi(\theta \mid Y)$

Markov chain Monte Carlo (MCMC) $\theta_t \; not$ independent

$$Y = \{y_1, y_2, \ldots, y_N\}$$



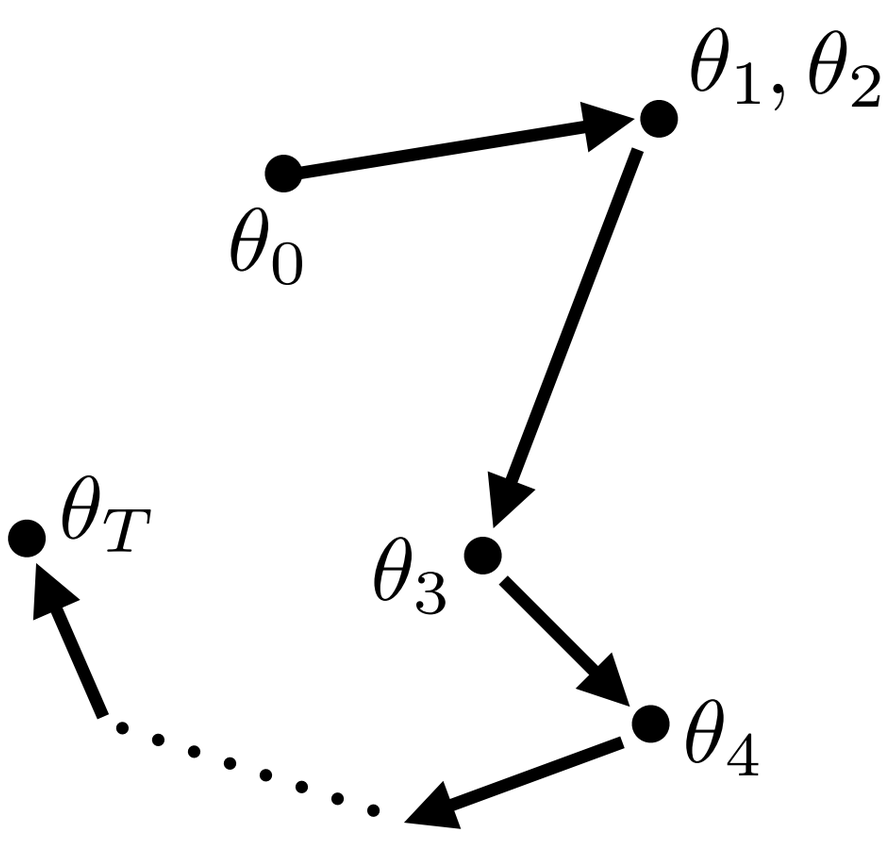
Problem: $\log p(Y|\theta)$ is expensive to evaluate if N is large

Key idea: likelihood approximation

Monte Carlo $\mathbb{E}[f(\theta) \mid Y] \approx T^{-1} \sum_{t=1}^{T} f(\theta_t) \quad \theta_t \overset{\text{i.i.d.}}{\sim} \pi(\theta \mid Y)$

Markov chain Monte Carlo (MCMC) $\theta_t \; not \; \text{independent}$

$$Y = \{y_1, y_2, \ldots, y_N\}$$



Problem: $\log p(Y|\theta)$ is expensive to evaluate if N is large

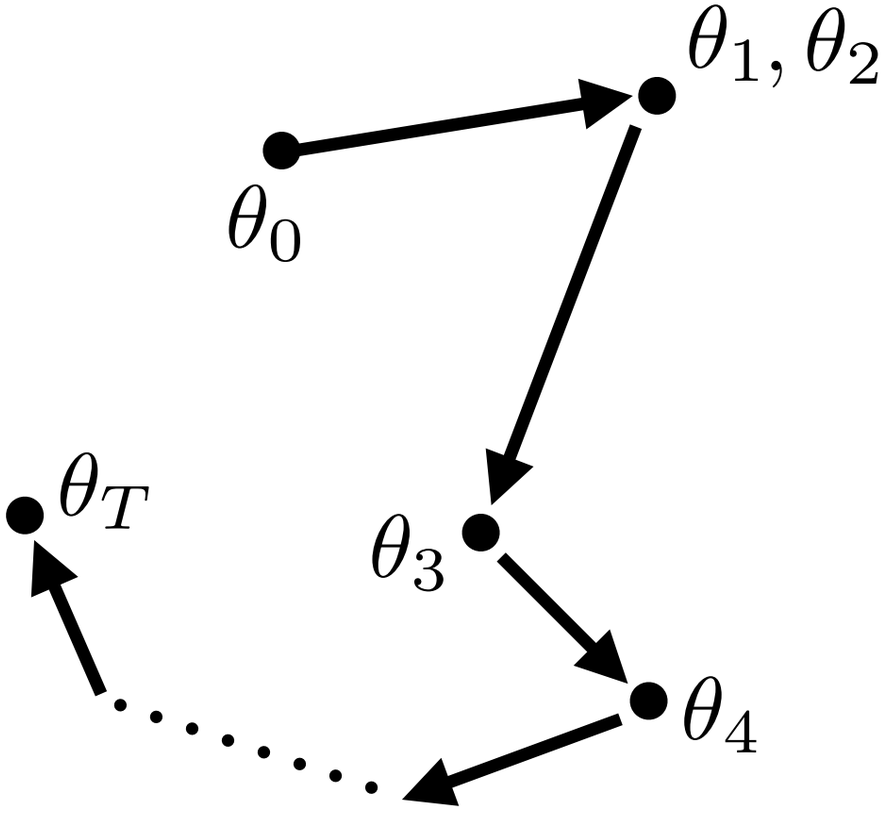
$\Omega(N \times T)$ time

Key idea: likelihood approximation

Monte Carlo $\mathbb{E}[f(\theta) \mid Y] \approx T^{-1} \sum_{t=1}^{T} f(\theta_t) \quad \theta_t \overset{\text{i.i.d.}}{\sim} \pi(\theta \mid Y)$

Markov chain Monte Carlo (MCMC) $\theta_t \; not$ independent

$Y = \{y_1, y_2, \ldots, y_N\}$



Problem: $\log p(Y|\theta)$ is expensive to evaluate if N is large

$\Omega(N \times T)$ time

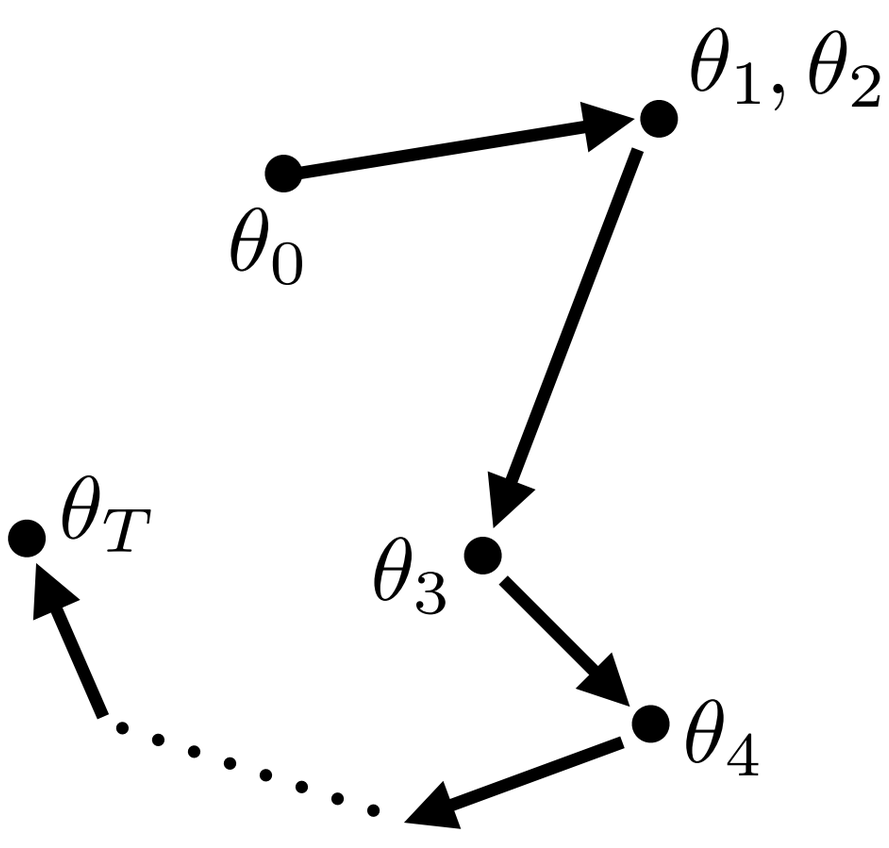
Solution:

Key idea: likelihood approximation

Monte Carlo $\mathbb{E}[f(\theta) \mid Y] \approx T^{-1} \sum_{t=1}^{T} f(\theta_t) \quad \theta_t \overset{\text{i.i.d.}}{\sim} \pi(\theta \mid Y)$

Markov chain Monte Carlo (MCMC) $\theta_t \; not$ independent

$$Y = \{y_1, y_2, \ldots, y_N\}$$



Problem: $\log p(Y|\theta)$ is expensive to evaluate if N is large

$\Omega(N \times T)$ time

Solution:

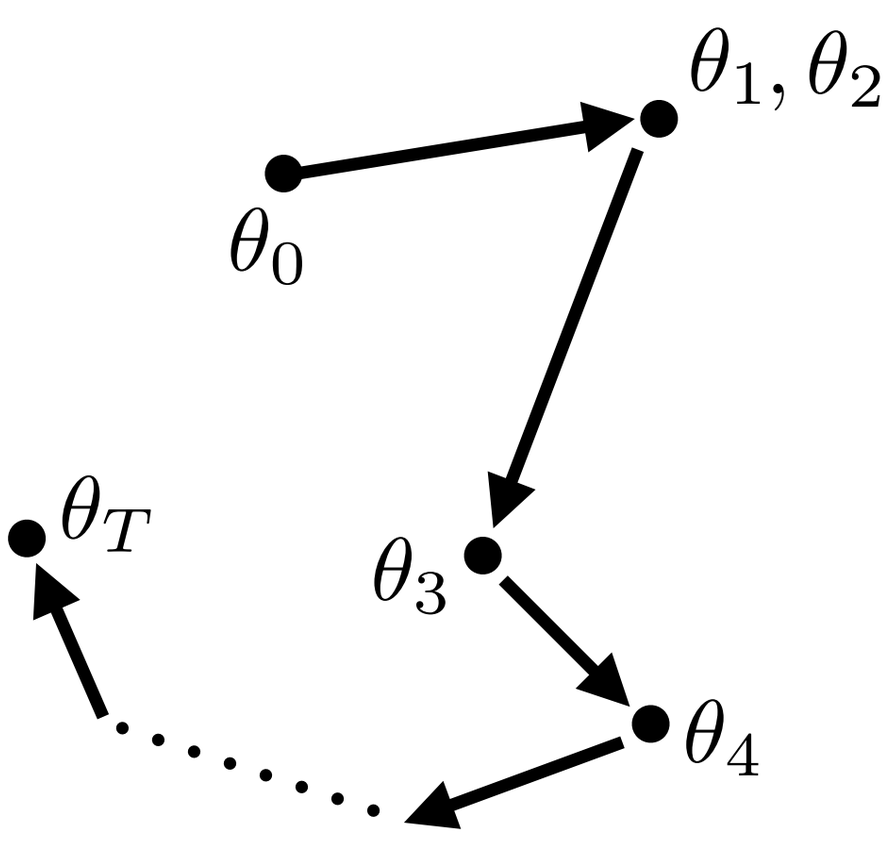
1. replace $\log p(Y|\theta)$ with a fast-to-compute proxy $\ell(\theta, g(Y))$

Key idea: likelihood approximation

Monte Carlo $\mathbb{E}[f(\theta) \mid Y] \approx T^{-1} \sum_{t=1}^{T} f(\theta_t) \quad \theta_t \overset{\text{i.i.d.}}{\sim} \pi(\theta \mid Y)$

Markov chain Monte Carlo (MCMC) $\theta_t \; not$ independent

$Y = \{y_1, y_2, \ldots, y_N\}$



Problem: $\log p(Y|\theta)$ is expensive to evaluate if N is large

$\Omega(N \times T)$ time

Solution:

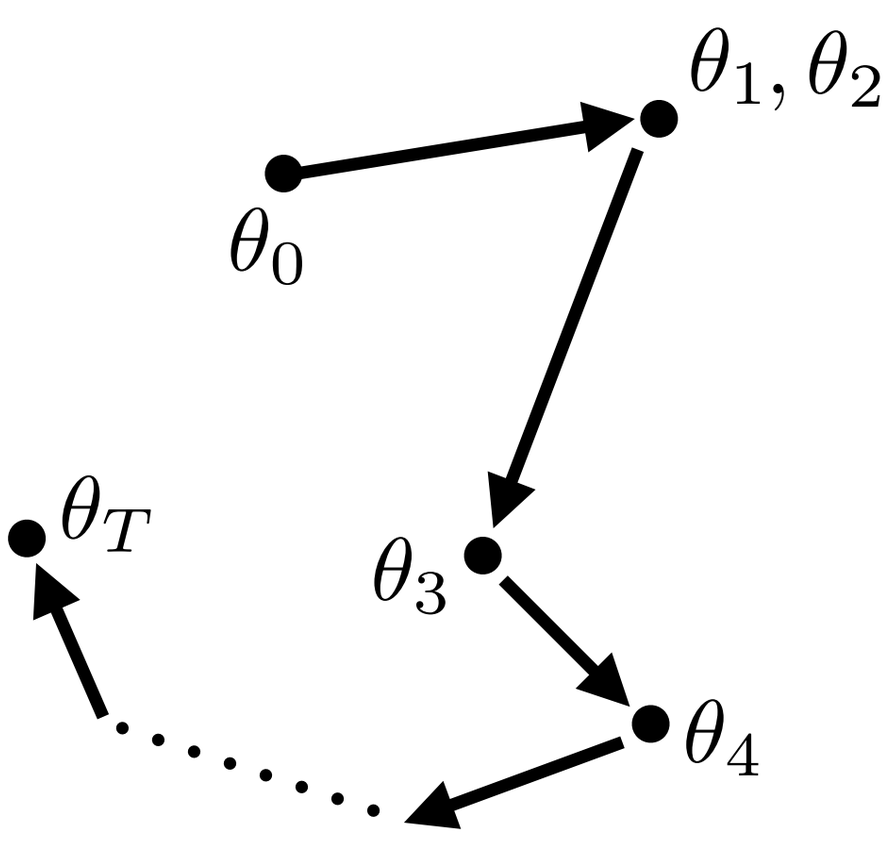
1. replace $\log p(Y|\theta)$ with a fast-to-compute proxy $\ell(\theta, g(Y))$ ← short summary of Y

Key idea: likelihood approximation

Monte Carlo $\mathbb{E}[f(\theta) \mid Y] \approx T^{-1} \sum_{t=1}^{T} f(\theta_t) \quad \theta_t \overset{\text{i.i.d.}}{\sim} \pi(\theta \mid Y)$

Markov chain Monte Carlo (MCMC) $\theta_t \; not$ independent

$Y = \{y_1, y_2, \ldots, y_N\}$



Problem: $\log p(Y|\theta)$ is expensive to evaluate if N is large

$\Omega(N \times T)$ time

Solution:

1. replace $\log p(Y|\theta)$ with a fast-to-compute proxy $\ell(\theta, g(Y))$
2. choose $\ell(\theta, g(Y))$ so approximate posterior is accurate

Ideal case: exponential family

$$Y = \{y_1, y_2, \ldots, y_N\}$$

Ideal case: exponential family

$$Y = \{y_1, y_2, \ldots, y_N\}$$

$$\log p(y_n|\theta) = \eta(\theta) \cdot \tau(y_n)$$

Ideal case: exponential family

$$Y = \{y_1, y_2, \ldots, y_N\}$$

$$\log p(y_n|\theta) = \eta(\theta) \cdot \tau(y_n)$$

reparameterization

Ideal case: exponential family

$$Y = \{y_1, y_2, \ldots, y_N\}$$

$$\log p(y_n|\theta) = \eta(\theta) \cdot \tau(y_n)$$

reparameterization **sufficient statistics**

Ideal case: exponential family

$$Y = \{y_1, y_2, \ldots, y_N\}$$

$$\sum_{n=1}^{N} \log p(y_n|\theta) = \eta(\theta) \cdot \sum_{n=1}^{N} \tau(y_n)$$

Ideal case: exponential family

$$Y = \{y_1, y_2, \ldots, y_N\}$$

$$\log p(Y|\theta) = \sum_{n=1}^{N} \log p(y_n|\theta) = \eta(\theta) \cdot \sum_{n=1}^{N} \tau(y_n)$$

Ideal case: exponential family

$$Y = \{y_1, y_2, \ldots, y_N\}$$

$$\log p(Y|\theta) = \sum_{n=1}^{N} \log p(y_n|\theta) = \eta(\theta) \cdot \underbrace{\sum_{n=1}^{N} \tau(y_n)}_{g(Y)}$$

Ideal case: exponential family

$$Y = \{y_1, y_2, \ldots, y_N\}$$

$$\log p(Y|\theta) = \sum_{n=1}^{N} \log p(y_n|\theta) = \eta(\theta) \cdot \underbrace{\sum_{n=1}^{N} \tau(y_n)}_{g(Y)}$$

Run MCMC for T iterations:

Ideal case: exponential family

$$Y = \{y_1, y_2, \ldots, y_N\}$$

$$\log p(Y|\theta) = \sum_{n=1}^{N} \log p(y_n|\theta) = \eta(\theta) \cdot \underbrace{\sum_{n=1}^{N} \tau(y_n)}_{g(Y)}$$

Run MCMC for T iterations:

- Naively: $\Omega(N \times T)$ time

Ideal case: exponential family

$$Y = \{y_1, y_2, \ldots, y_N\}$$

$$\log p(Y|\theta) = \sum_{n=1}^{N} \log p(y_n|\theta) = \eta(\theta) \cdot \underbrace{\sum_{n=1}^{N} \tau(y_n)}_{g(Y)}$$

Run MCMC for T iterations:

$\log p(Y|\theta)$ takes $\Omega(N)$ time to evaluate

- Naively: $\Omega(N \times T)$ time

Ideal case: exponential family

$$Y = \{y_1, y_2, \ldots, y_N\}$$

$$\log p(Y|\theta) = \sum_{n=1}^{N} \log p(y_n|\theta) = \eta(\theta) \cdot \underbrace{\sum_{n=1}^{N} \tau(y_n)}_{g(Y)}$$

Run MCMC for T iterations:

- Naively: $\Omega(N \times T)$ time

$\log p(Y|\theta)$ takes $\Omega(N)$ time to evaluate

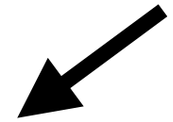
Ideal case: exponential family

$$Y = \{y_1, y_2, \ldots, y_N\}$$

$$\log p(Y|\theta) = \sum_{n=1}^{N} \log p(y_n|\theta) = \eta(\theta) \cdot \underbrace{\sum_{n=1}^{N} \tau(y_n)}_{g(Y)}$$

Run MCMC for T iterations:

$\log p(Y|\theta)$ takes $\Omega(N)$ time to evaluate



- Naively: $\Omega(N \times T)$ time 
- Using EF structure: $O(N + T)$ time

Ideal case: exponential family

$$Y = \{y_1, y_2, \ldots, y_N\}$$

$$\log p(Y|\theta) = \sum_{n=1}^{N} \log p(y_n|\theta) = \eta(\theta) \cdot \underbrace{\sum_{n=1}^{N} \tau(y_n)}_{g(Y)}$$

Run MCMC for T iterations:

$\log p(Y|\theta)$ takes $\Omega(N)$ time to evaluate

- Naively: $\Omega(N \times T)$ time 
- Using EF structure: $O(N + T)$ time

compute $g(Y)$

Ideal case: exponential family

$$Y = \{y_1, y_2, \ldots, y_N\}$$

$$\log p(Y|\theta) = \sum_{n=1}^{N} \log p(y_n|\theta) = \eta(\theta) \cdot \underbrace{\sum_{n=1}^{N} \tau(y_n)}_{g(Y)}$$

Run MCMC for T iterations:

- Naively: $\Omega(N \times T)$ time

- Using EF structure: $O(N + T)$ time

$\log p(Y|\theta)$ takes $\Omega(N)$ time to evaluate

compute $g(Y)$

$\log p(Y|\theta)$ takes $O(1)$ time to evaluate

Ideal case: exponential family

$$Y = \{y_1, y_2, \ldots, y_N\}$$

$$\log p(Y|\theta) = \sum_{n=1}^{N} \log p(y_n|\theta) = \eta(\theta) \cdot \underbrace{\sum_{n=1}^{N} \tau(y_n)}_{g(Y)}$$

Run MCMC for T iterations:

- Naively: $\Omega(N \times T)$ time

- Using EF structure: $O(N + T)$ time

$\log p(Y|\theta)$ takes $\Omega(N)$ time to evaluate

compute $g(Y)$

$\log p(Y|\theta)$ takes $O(1)$ time to evaluate

Streaming and distributed exponential family inference

$$Y = \{y_1, y_2, \ldots, y_N\}$$

$$\log p(Y|\theta) = \sum_{n=1}^{N} \log p(y_n|\theta) = \eta(\theta) \cdot \sum_{n=1}^{N} \tau(y_n)$$

Streaming and distributed exponential family inference

$$Y = \{y_1, y_2, \ldots, y_N\}$$

$$\log p(Y|\theta) = \sum_{n=1}^{N} \log p(y_n|\theta) = \eta(\theta) \cdot \sum_{n=1}^{N} \tau(y_n)$$

Streaming

Streaming and distributed exponential family inference

$$Y = \{y_1, y_2, \ldots, y_N\}$$

$$\log p(Y|\theta) = \sum_{n=1}^{N} \log p(y_n|\theta) = \eta(\theta) \cdot \sum_{n=1}^{N} \tau(y_n)$$

Streaming

$\tau = 0$

Streaming and distributed exponential family inference

$$Y = \{y_1, y_2, \ldots, y_N\}$$

$$\log p(Y|\theta) = \sum_{n=1}^{N} \log p(y_n|\theta) = \eta(\theta) \cdot \sum_{n=1}^{N} \tau(y_n)$$

STREAMING

$\tau = 0$

for $n = 1, \ldots, N$

Streaming and distributed exponential family inference

$$Y = \{y_1, y_2, \ldots, y_N\}$$

$$\log p(Y|\theta) = \sum_{n=1}^{N} \log p(y_n|\theta) = \eta(\theta) \cdot \sum_{n=1}^{N} \tau(y_n)$$

Streaming

$\tau = 0$

for $n = 1, \ldots, N$

$\quad \tau = \tau + \tau(y_n)$

Streaming and distributed exponential family inference

$$Y = \{y_1, y_2, \ldots, y_N\}$$

$$\log p(Y|\theta) = \sum_{n=1}^{N} \log p(y_n|\theta) = \eta(\theta) \cdot \sum_{n=1}^{N} \tau(y_n)$$

Streaming

$\tau = 0$

for $n = 1, \ldots, N$

$\quad \tau = \tau + \tau(y_n)$

construct $\log p(Y|\theta)$ using τ

Streaming and distributed exponential family inference

$$Y = \{y_1, y_2, \ldots, y_N\}$$

$$\log p(Y|\theta) = \sum_{n=1}^{N} \log p(y_n|\theta) = \eta(\theta) \cdot \sum_{n=1}^{N} \tau(y_n)$$

STREAMING

$\tau = 0$

for $n = 1, \ldots, N$

    $\tau = \tau + \tau(y_n)$

construct $\log p(Y|\theta)$ using τ

run MCMC

Streaming and distributed exponential family inference

$$Y = \{y_1, y_2, \ldots, y_N\}$$

$$\log p(Y|\theta) = \sum_{n=1}^{N} \log p(y_n|\theta) = \eta(\theta) \cdot \sum_{n=1}^{N} \tau(y_n)$$

STREAMING

$\tau = 0$

for $n = 1, \ldots, N$

$\quad \tau = \tau + \tau(y_n)$

construct $\log p(Y|\theta)$ using τ

run MCMC

DISTRIBUTED

Streaming and distributed exponential family inference

$$Y = \{y_1, y_2, \ldots, y_N\}$$

$$\log p(Y|\theta) = \sum_{n=1}^{N} \log p(y_n|\theta) = \eta(\theta) \cdot \sum_{n=1}^{N} \tau(y_n)$$

STREAMING

$\tau = 0$

for $n = 1, \ldots, N$

$\tau = \tau + \tau(y_n)$

construct $\log p(Y|\theta)$ using τ

run MCMC

DISTRIBUTED

for $b = 1, \ldots, B$ **in parallel**

Streaming and distributed exponential family inference

$$Y = \{y_1, y_2, \ldots, y_N\}$$

$$\log p(Y|\theta) = \sum_{n=1}^{N} \log p(y_n|\theta) = \eta(\theta) \cdot \sum_{n=1}^{N} \tau(y_n)$$

STREAMING

$\tau = 0$

for $n = 1, \ldots, N$

$\tau = \tau + \tau(y_n)$

construct $\log p(Y|\theta)$ using τ

run MCMC

DISTRIBUTED

for $b = 1, \ldots, B$ **in parallel**

1. worker b reads data subset $Y_b = \{ y_{B(b-1)/n}, \ldots, y_{Bb/n-1} \}$

Streaming and distributed exponential family inference

$$Y = \{y_1, y_2, \ldots, y_N\}$$

$$\log p(Y|\theta) = \sum_{n=1}^{N} \log p(y_n|\theta) = \eta(\theta) \cdot \sum_{n=1}^{N} \tau(y_n)$$

STREAMING

$\tau = 0$

for $n = 1, \ldots, N$

$\tau = \tau + \tau(y_n)$

construct $\log p(Y|\theta)$ using τ

run MCMC

DISTRIBUTED

for $b = 1, \ldots, B$ **in parallel**

1. worker b reads data subset $Y_b = \{ y_{B(b-1)/n}, \ldots, y_{Bb/n-1} \}$
2. worker b computes $\tau_b = \tau(Y_b)$

Streaming and distributed exponential family inference

$$Y = \{y_1, y_2, \ldots, y_N\}$$

$$\log p(Y|\theta) = \sum_{n=1}^{N} \log p(y_n|\theta) = \eta(\theta) \cdot \sum_{n=1}^{N} \tau(y_n)$$

STREAMING

$\tau = 0$

for $n = 1, \ldots, N$

$\tau = \tau + \tau(y_n)$

construct $\log p(Y|\theta)$ using τ

run MCMC

DISTRIBUTED

for $b = 1, \ldots, B$ **in parallel**

1. worker b reads data subset $Y_b = \{ y_{B(b-1)/n}, \ldots, y_{Bb/n-1} \}$
2. worker b computes $\tau_b = \tau(Y_b)$

$\tau = \tau_1 + \cdots + \tau_b$

Streaming and distributed exponential family inference

$$Y = \{y_1, y_2, \ldots, y_N\}$$

$$\log p(Y|\theta) = \sum_{n=1}^{N} \log p(y_n|\theta) = \eta(\theta) \cdot \sum_{n=1}^{N} \tau(y_n)$$

STREAMING

$\tau = 0$

for $n = 1, \ldots, N$

$\tau = \tau + \tau(y_n)$

construct $\log p(Y|\theta)$ using τ

run MCMC

DISTRIBUTED

for $b = 1, \ldots, B$ **in parallel**

1. worker b reads data subset $Y_b = \{ y_{B(b-1)/n}, \ldots, y_{Bb/n-1} \}$
2. worker b computes $\tau_b = \tau(Y_b)$

$\tau = \tau_1 + \cdots + \tau_b$

construct $\log p(Y|\theta)$ using τ

Streaming and distributed exponential family inference

$$Y = \{y_1, y_2, \ldots, y_N\}$$

$$\log p(Y|\theta) = \sum_{n=1}^{N} \log p(y_n|\theta) = \eta(\theta) \cdot \sum_{n=1}^{N} \tau(y_n)$$

STREAMING

$\tau = 0$

for $n = 1, \ldots, N$

$\tau = \tau + \tau(y_n)$

construct $\log p(Y|\theta)$ using τ

run MCMC

DISTRIBUTED

for $b = 1, \ldots, B$ **in parallel**

1. worker b reads data subset $Y_b = \{ y_{B(b-1)/n}, \ldots, y_{Bb/n-1} \}$
2. worker b computes $\tau_b = \tau(Y_b)$

$\tau = \tau_1 + \cdots + \tau_b$

construct $\log p(Y|\theta)$ using τ

run MCMC

Streaming and distributed exponential family inference

Roadmap

- Approximate Bayes review
- Likelihood approximation and dataset compression
- Approximate sufficient statistics
- Accuracy guarantees

Polynomials are good for approximation

$$-5.8x^6 + 14.9x^4 - 10.1x^2 + 1.1$$

Polynomials are good for approximation

$$-5.8x^6 + 14.9x^4 - 10.1x^2 + 1.1$$

1. Computationally convenient

Polynomials are good for approximation

$$-5.8x^6 + 14.9x^4 - 10.1x^2 + 1.1$$

1. Computationally convenient

2. Can approximate any smooth function

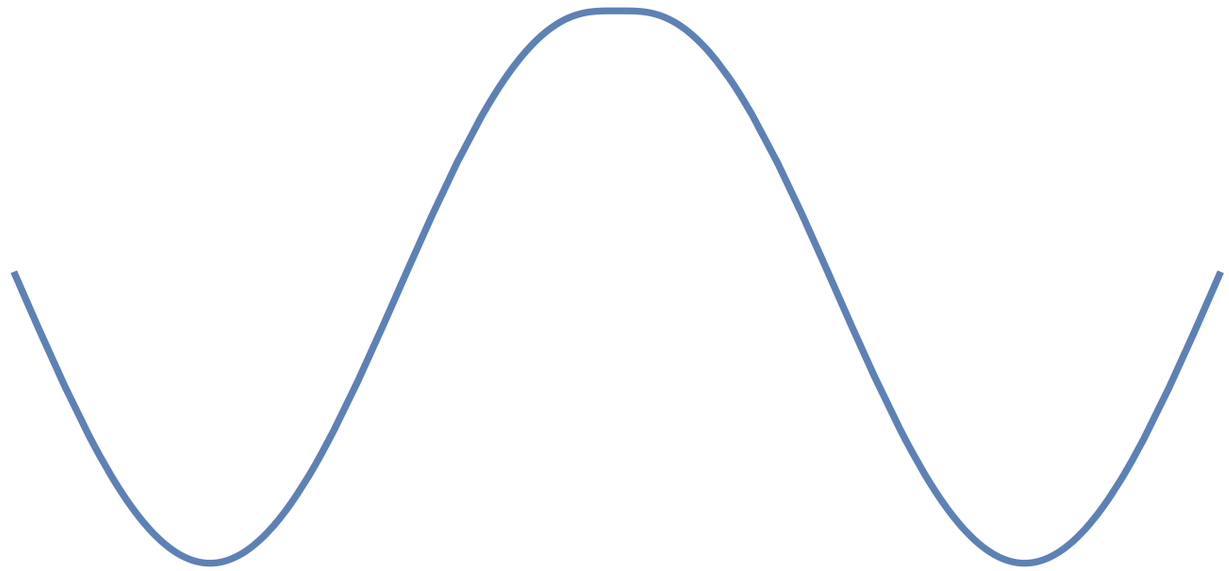
Polynomials are good for approximation

$$-5.8x^6 + 14.9x^4 - 10.1x^2 + 1.1$$

1. Computationally convenient
2. Can approximate any smooth function
3. Approximation properties are well-understood

Polynomials are good for approximation

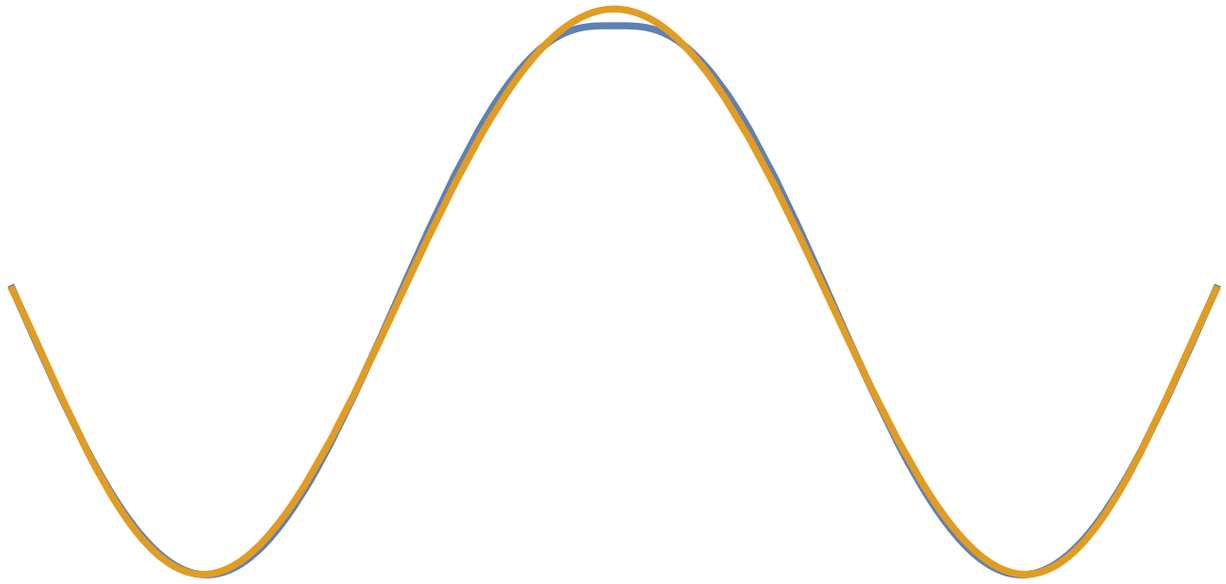
$$-5.8x^6 + 14.9x^4 - 10.1x^2 + 1.1$$



1. Computationally convenient
2. Can approximate any smooth function
3. Approximation properties are well-understood

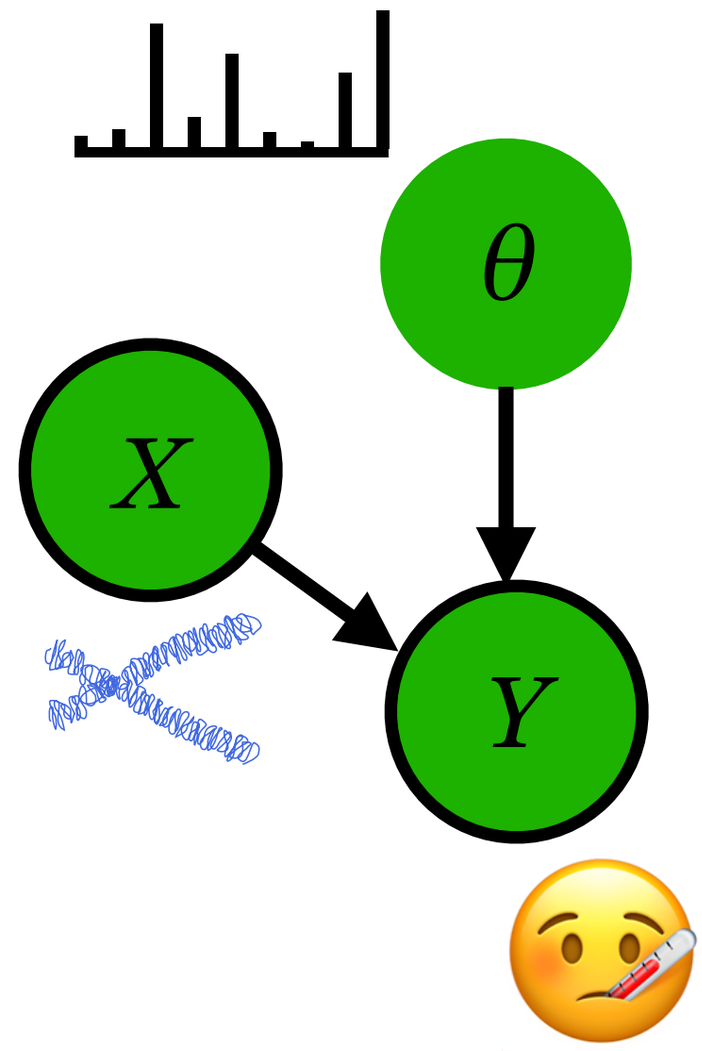
Polynomials are good for approximation

$$-5.8x^6 + 14.9x^4 - 10.1x^2 + 1.1$$



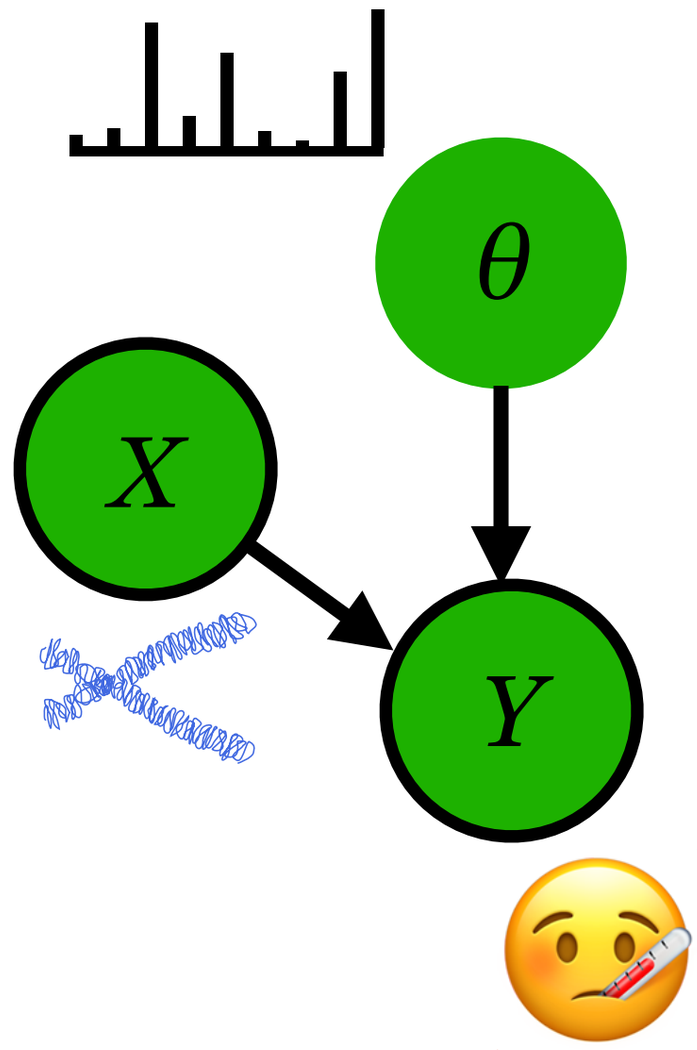
1. Computationally convenient
2. Can approximate any smooth function
3. Approximation properties are well-understood

Polynomial approximate sufficient statistics (PASS)



Polynomial approximate sufficient statistics (PASS)

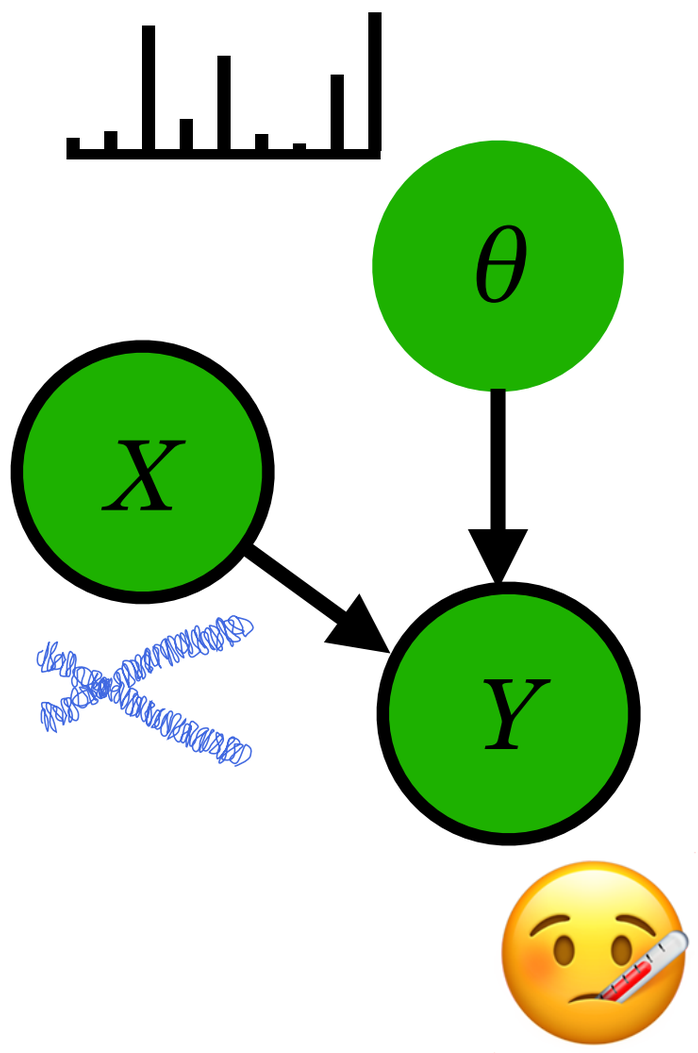
$$Y = \{y_1, y_2, \ldots, y_N\}$$



Polynomial approximate sufficient statistics (PASS)

$$Y = \{y_1, y_2, \ldots, y_N\}$$

$$X = \{x_1, x_2, \ldots, x_N\}$$

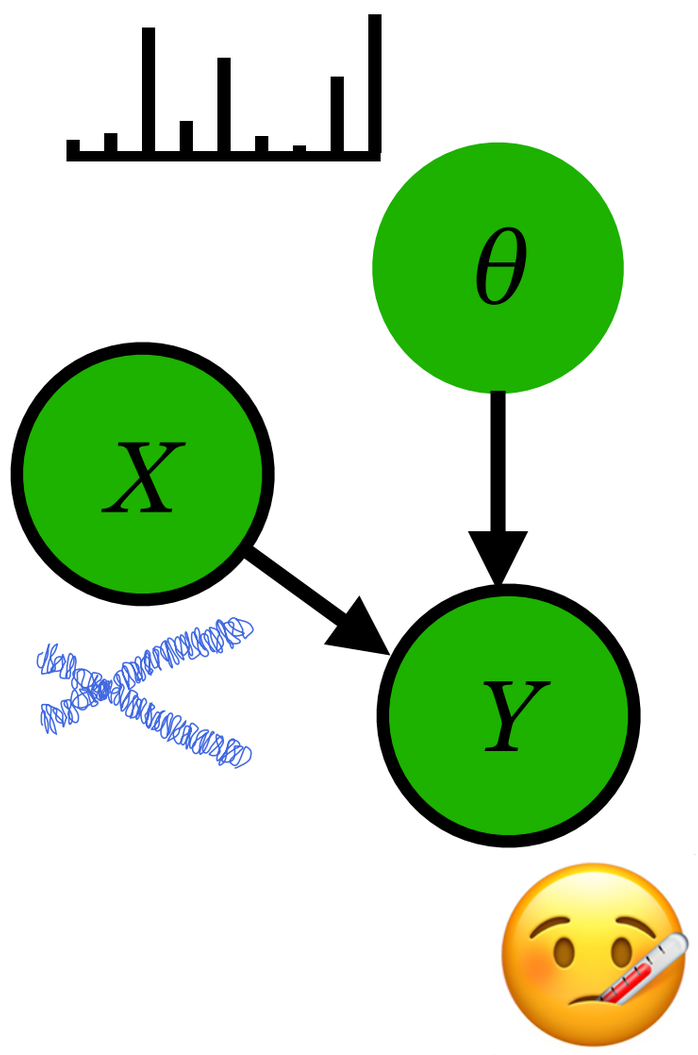


Polynomial approximate sufficient statistics (PASS)

$$Y = \{y_1, y_2, \ldots, y_N\}$$

$$X = \{x_1, x_2, \ldots, x_N\}$$

$$x_n = (x_{n1}, x_{n2}, \ldots, x_{nd})$$



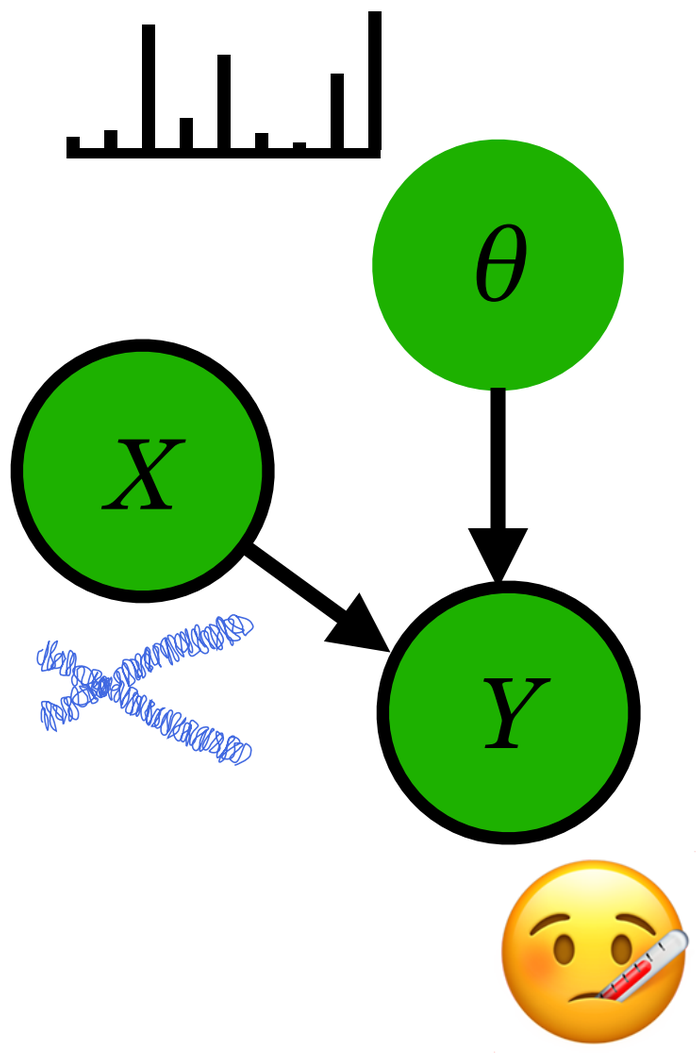
Polynomial approximate sufficient statistics (PASS)

$$Y = \{y_1, y_2, \ldots, y_N\}$$

$$X = \{x_1, x_2, \ldots, x_N\}$$

$$x_n = (x_{n1}, x_{n2}, \ldots, x_{nd})$$

$$\log p(y_n | x_n, \theta) \approx \eta(\theta) \cdot \tau(y_n, x_n)$$



Polynomial approximate sufficient statistics (PASS)

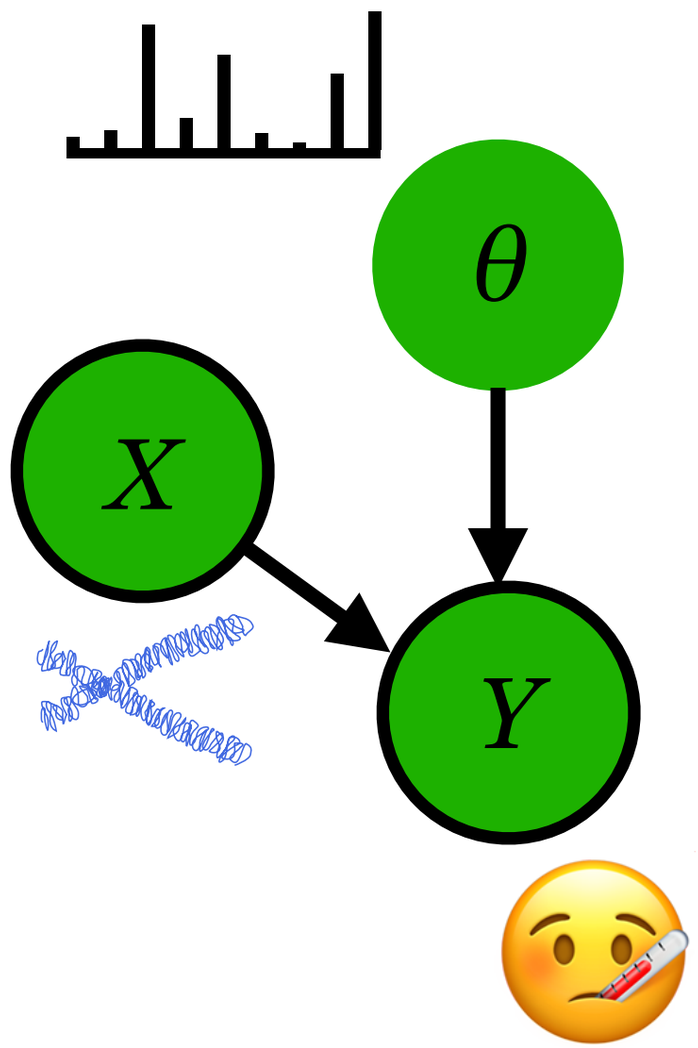
$$Y = \{y_1, y_2, \ldots, y_N\}$$

$$X = \{x_1, x_2, \ldots, x_N\}$$

$$x_n = (x_{n1}, x_{n2}, \ldots, x_{nd})$$

$$\log p(y_n | x_n, \theta) \approx \eta(\theta) \cdot \tau(y_n, x_n)$$

$$\tau(y_n, x_n) = (y_n, x_{n1}, x_{n2}, \ldots, x_{nd},$$



Polynomial approximate sufficient statistics (PASS)

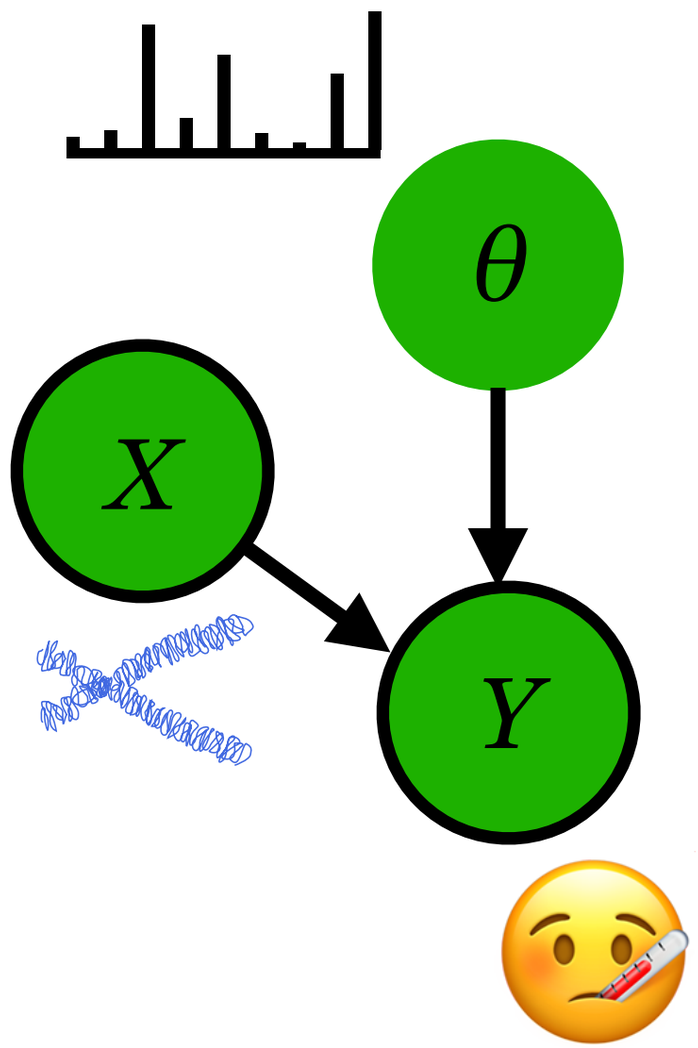
$$Y = \{y_1, y_2, \ldots, y_N\}$$

$$X = \{x_1, x_2, \ldots, x_N\}$$

$$x_n = (x_{n1}, x_{n2}, \ldots, x_{nd})$$

$$\log p(y_n | x_n, \theta) \approx \eta(\theta) \cdot \tau(y_n, x_n)$$

$$\tau(y_n, x_n) = (y_n, x_{n1}, x_{n2}, \ldots, x_{nd},$$
$$y_n^2, x_{n1}^2, x_{n2}^2, \ldots, x_{nd}^2,$$



Polynomial approximate sufficient statistics (PASS)

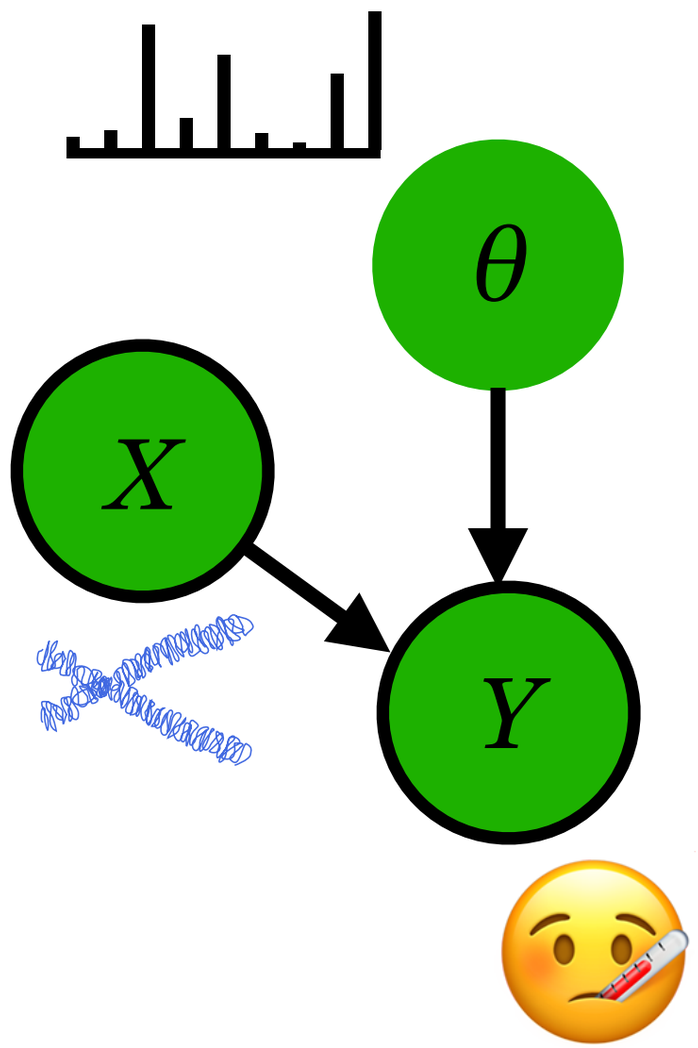
$$Y = \{y_1, y_2, \ldots, y_N\}$$

$$X = \{x_1, x_2, \ldots, x_N\}$$

$$x_n = (x_{n1}, x_{n2}, \ldots, x_{nd})$$

$$\log p(y_n | x_n, \theta) \approx \eta(\theta) \cdot \tau(y_n, x_n)$$

$$\tau(y_n, x_n) = (y_n, x_{n1}, x_{n2}, \ldots, x_{nd},$$
$$y_n^2, x_{n1}^2, x_{n2}^2, \ldots, x_{nd}^2,$$
$$y_n x_{n1}, \ldots, y_n x_{nd},$$



Polynomial approximate sufficient statistics (PASS)

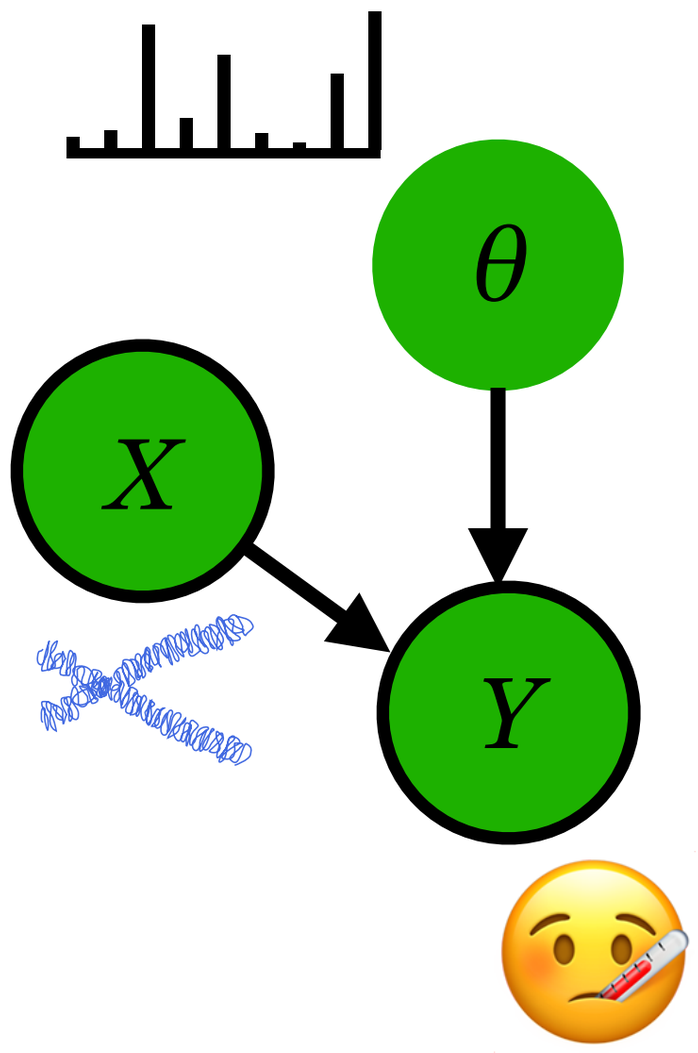
$$Y = \{y_1, y_2, \ldots, y_N\}$$

$$X = \{x_1, x_2, \ldots, x_N\}$$

$$x_n = (x_{n1}, x_{n2}, \ldots, x_{nd})$$

$$\log p(y_n | x_n, \theta) \approx \eta(\theta) \cdot \tau(y_n, x_n)$$

$$\tau(y_n, x_n) = (y_n, x_{n1}, x_{n2}, \ldots, x_{nd},$$
$$y_n^2, x_{n1}^2, x_{n2}^2, \ldots, x_{nd}^2,$$
$$y_n x_{n1}, \ldots, y_n x_{nd},$$
$$x_{n1} x_{n2}, x_{n1} x_{n3}, \ldots,$$



Polynomial approximate sufficient statistics (PASS)

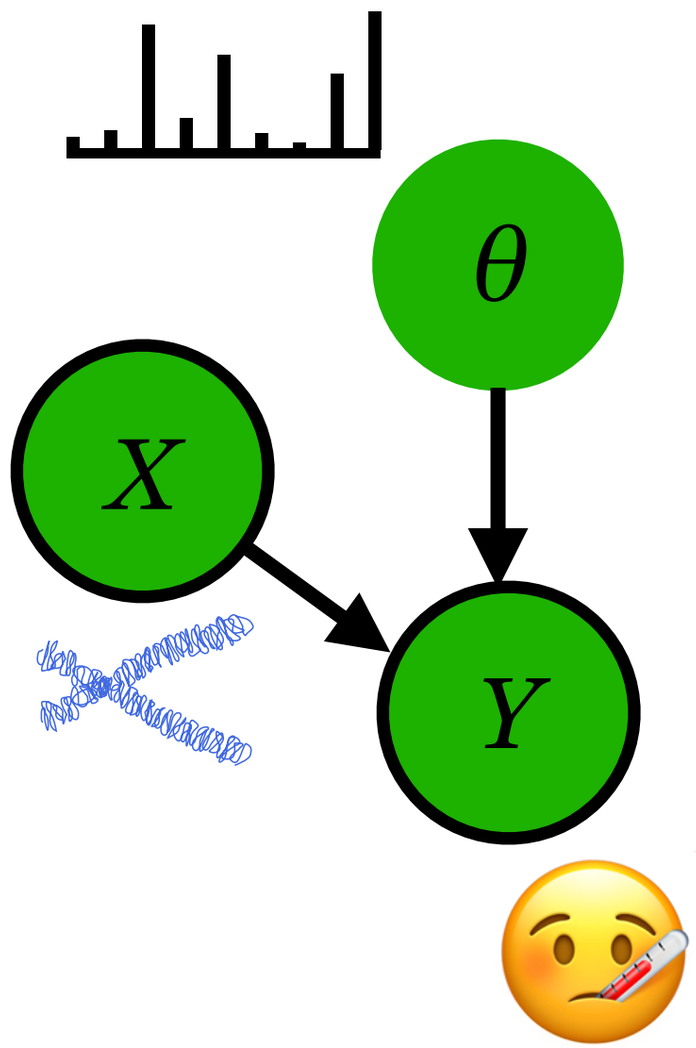
$$Y = \{y_1, y_2, \ldots, y_N\}$$

$$X = \{x_1, x_2, \ldots, x_N\}$$

$$x_n = (x_{n1}, x_{n2}, \ldots, x_{nd})$$

$$\log p(y_n | x_n, \theta) \approx \eta(\theta) \cdot \tau(y_n, x_n)$$

$$\tau(y_n, x_n) = (y_n, x_{n1}, x_{n2}, \ldots, x_{nd},$$
$$y_n^2, x_{n1}^2, x_{n2}^2, \ldots, x_{nd}^2,$$
$$y_n x_{n1}, \ldots, y_n x_{nd},$$
$$x_{n1} x_{n2}, x_{n1} x_{n3}, \ldots,$$
$$\ldots,$$



Polynomial approximate sufficient statistics (PASS)

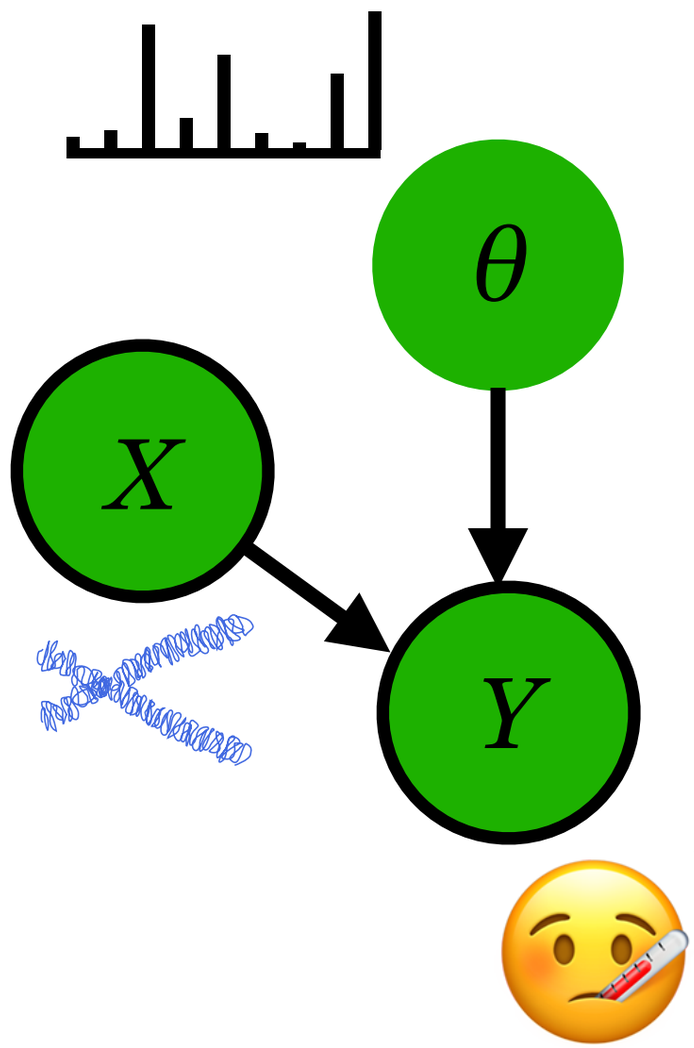
$$Y = \{y_1, y_2, \ldots, y_N\}$$

$$X = \{x_1, x_2, \ldots, x_N\}$$

$$x_n = (x_{n1}, x_{n2}, \ldots, x_{nd})$$

$$\log p(y_n | x_n, \theta) \approx \eta(\theta) \cdot \tau(y_n, x_n)$$

$$\begin{aligned}\tau(y_n, x_n) = (&y_n, x_{n1}, x_{n2}, \ldots, x_{nd},\\ &y_n^2, x_{n1}^2, x_{n2}^2, \ldots, x_{nd}^2,\\ &y_n x_{n1}, \ldots, y_n x_{nd},\\ &x_{n1} x_{n2}, x_{n1} x_{n3}, \ldots,\\ &\ldots,\\ &y_n^M, x_{n1}^M, x_{n2}^M, \ldots, x_{nd}^M)\end{aligned}$$



Polynomial approximate sufficient statistics (PASS)

$$Y = \{y_1, y_2, \ldots, y_N\}$$

$$X = \{$$

$$x_n$$

$$\log p(y_n|x$$

1. Scalability

$$\tau(y_n, x_n) = (y_n, x_{n1}, x_{n2}, \ldots, x_{nd},$$

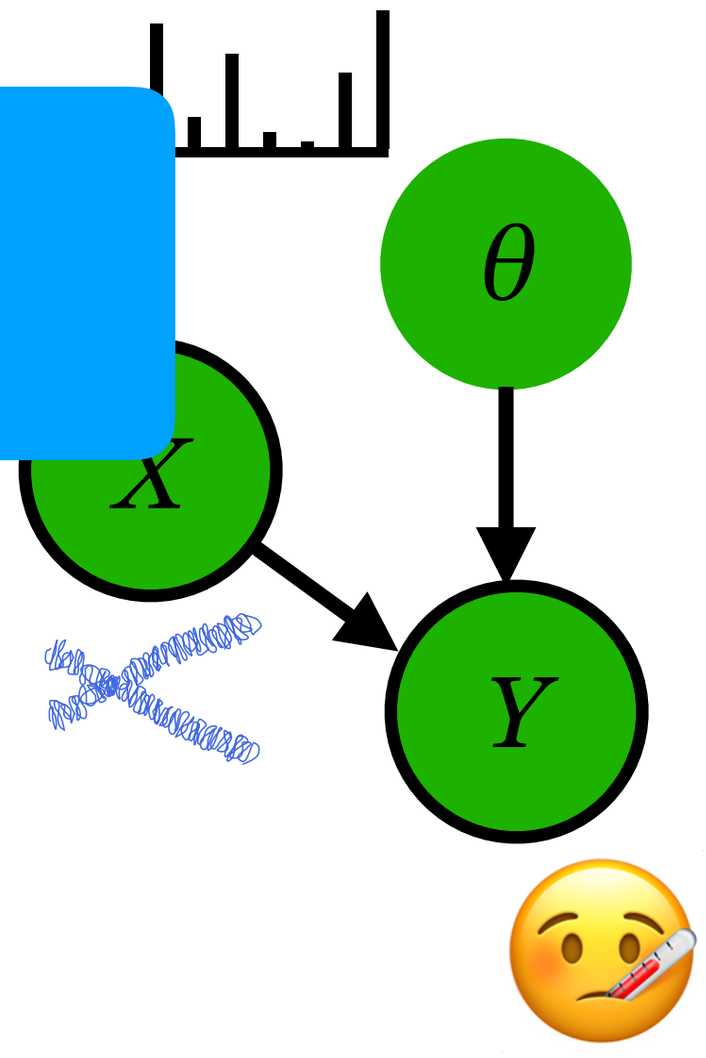
$$y_n^2, x_{n1}^2, x_{n2}^2, \ldots, x_{nd}^2,$$

$$y_n x_{n1}, \ldots, y_n x_{nd},$$

$$x_{n1}x_{n2}, x_{n1}x_{n3}, \ldots,$$

$$\ldots,$$

$$y_n^M, x_{n1}^M, x_{n2}^M, \ldots, x_{nd}^M)$$



Polynomial approximate sufficient statistics (PASS)

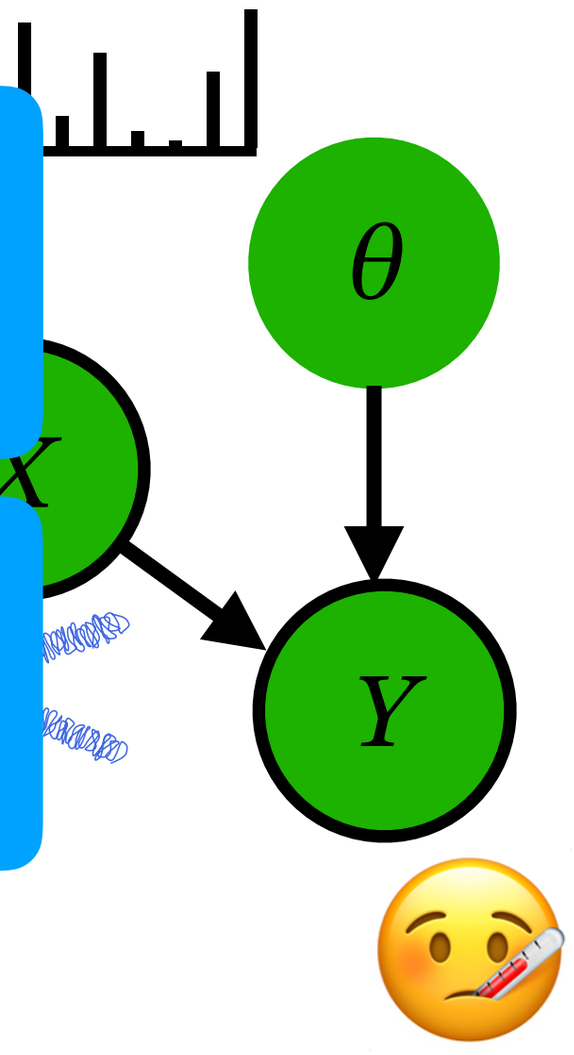
$$Y = \{y_1, y_2, \ldots, y_N\}$$

$$X = \{$$

$$x_n$$

$$\log p(y_n|x$$

$$\tau(y_n, x_n) =$$



1. Scalability

2. Arbitrary accuracy

$$x_{n1}x_{n2}, x_{n1}x_{n3}, \ldots,$$

$$\ldots,$$

$$y_n^M, x_{n1}^M, x_{n2}^M, \ldots, x_{nd}^M)$$

PASS for generalized linear models (PASS-GLM)

PASS for generalized linear models (PASS-GLM)

generalized linear models: $\log p(y_n \mid x_n, \theta) = \phi(y_n, \theta \cdot x_n)$

PASS for generalized linear models (PASS-GLM)

generalized linear models: $\log p(y_n \mid x_n, \theta) = \phi(y_n, \theta \cdot x_n)$

- Poisson regression (count data):

PASS for generalized linear models (PASS-GLM)

generalized linear models: $\log p(y_n \mid x_n, \theta) = \phi(y_n, \theta \cdot x_n)$

- Poisson regression (count data):
  - ➡ # of foreclosures by region

PASS for generalized linear models (PASS-GLM)

generalized linear models: $\log p(y_n \mid x_n, \theta) = \phi(y_n, \theta \cdot x_n)$

- Poisson regression (count data):
  - ➡ # of foreclosures by region
- Logistic regression (binary data)

PASS for generalized linear models (PASS-GLM)

generalized linear models: $\log p(y_n \mid x_n, \theta) = \phi(y_n, \theta \cdot x_n)$

- Poisson regression (count data):
  - ➡ # of foreclosures by region
- Logistic regression (binary data)
  - ➡ patient has cancer?

PASS for generalized linear models (PASS-GLM)

generalized linear models: $\log p(y_n \mid x_n, \theta) = \phi(y_n, \theta \cdot x_n)$

- Poisson regression (count data):
  - ➡ # of foreclosures by region
- Logistic regression (binary data)
  - ➡ patient has cancer?
- Robust regression (continuous data)

PASS for generalized linear models (PASS-GLM)

generalized linear models: $\log p(y_n \mid x_n, \theta) = \phi(y_n, \theta \cdot x_n)$

- Poisson regression (count data):
  - ➡ # of foreclosures by region
- Logistic regression (binary data)
  - ➡ patient has cancer?
- Robust regression (continuous data)
  - ➡ birth rate by region

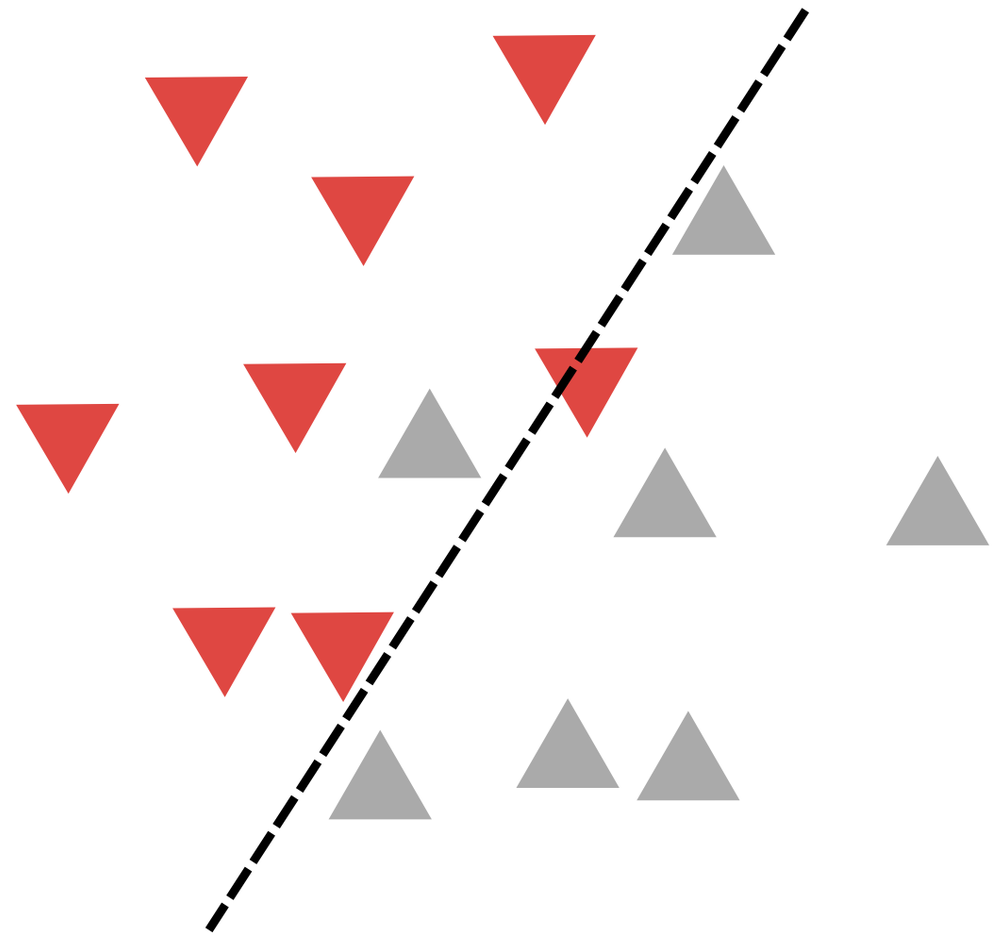
PASS for generalized linear models (PASS-GLM)

generalized linear models: $\log p(y_n \mid x_n, \theta) = \phi(y_n, \theta \cdot x_n)$

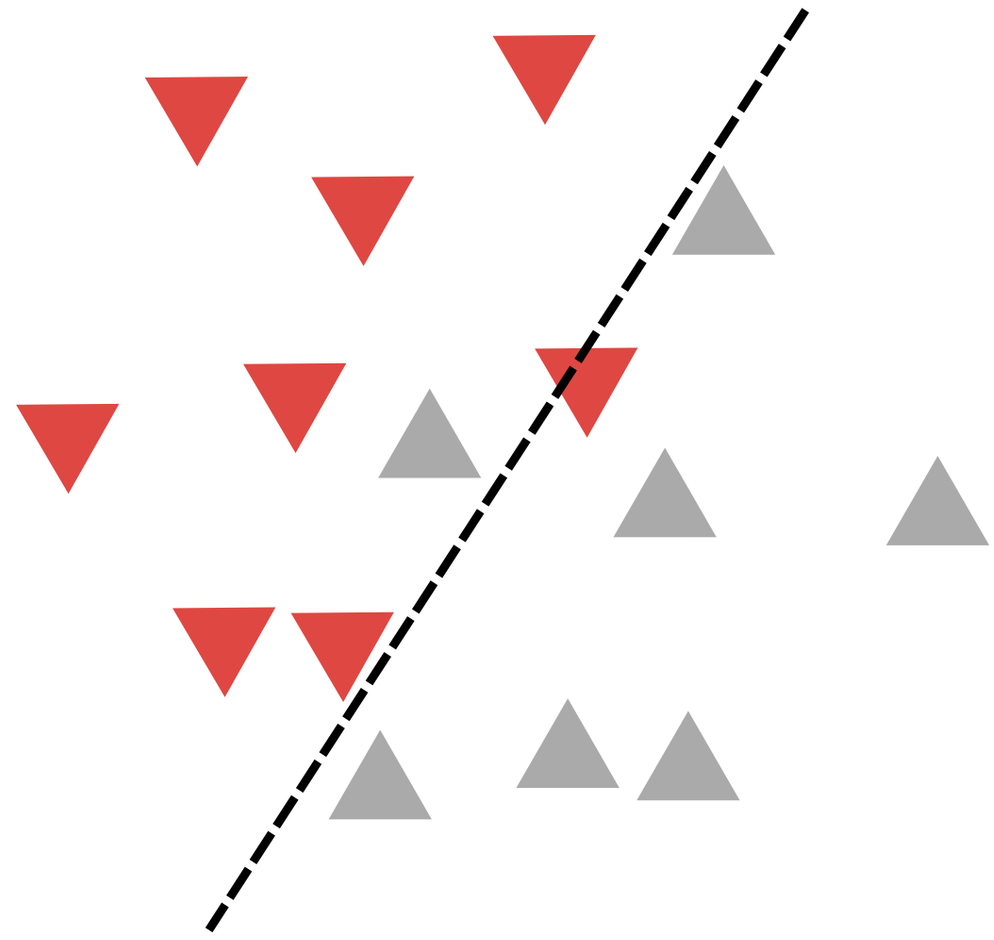
- Poisson regression (count data):
  - ➡ # of foreclosures by region
- Logistic regression (binary data)
  - ➡ patient has cancer?
- Robust regression (continuous data)
  - ➡ birth rate by region

$$\tau(y_n, x_n) = \left(a(k, M) y_n^{k_0} \prod_{i=1}^{d} x_{ni}^{k_i}\right)_{\substack{k \in \mathbb{N}^{d+1} \\ \sum_i k \leq M}} \qquad \eta(\theta) = \left(\prod_{i=1}^{d} \theta_i^{k_i}\right)_{\substack{k \in \mathbb{N}^{d+1} \\ \sum_i k \leq M}}$$

Case study: logistic regression

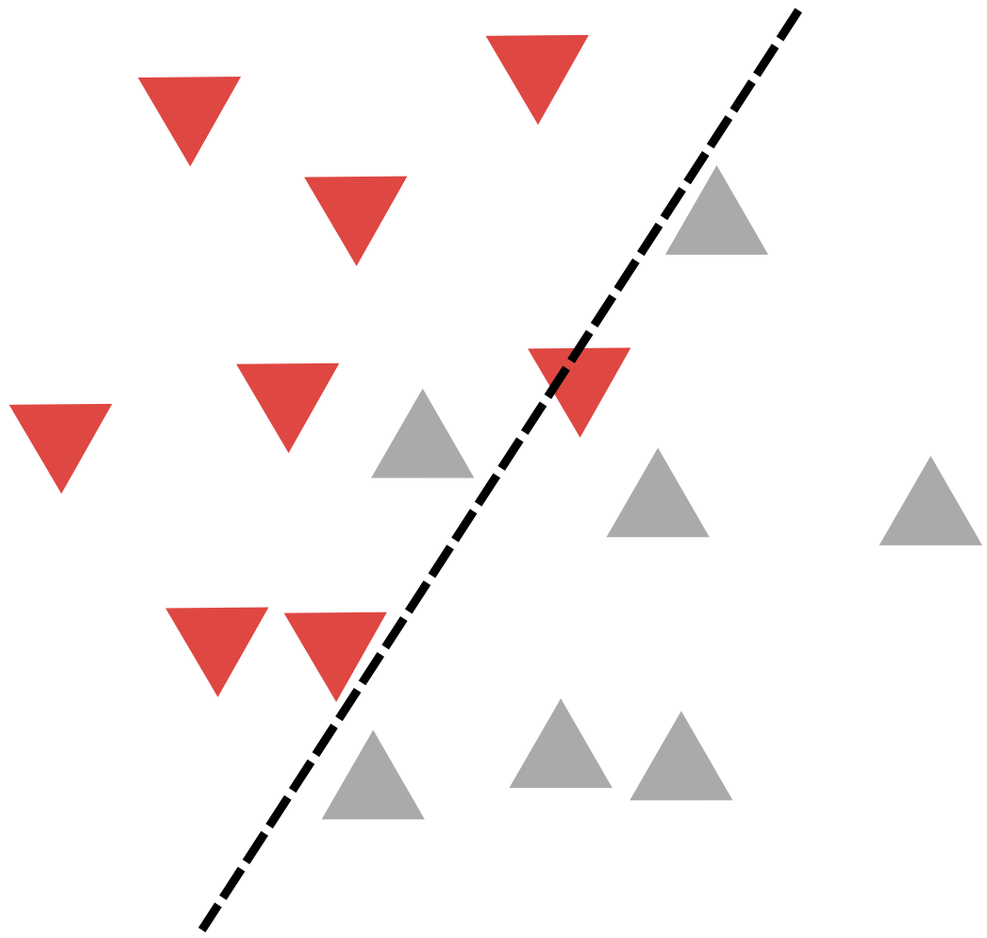


Case study: logistic regression



data $(x_n, y_n) \in \mathbb{R}^d \times \{\pm 1\}$

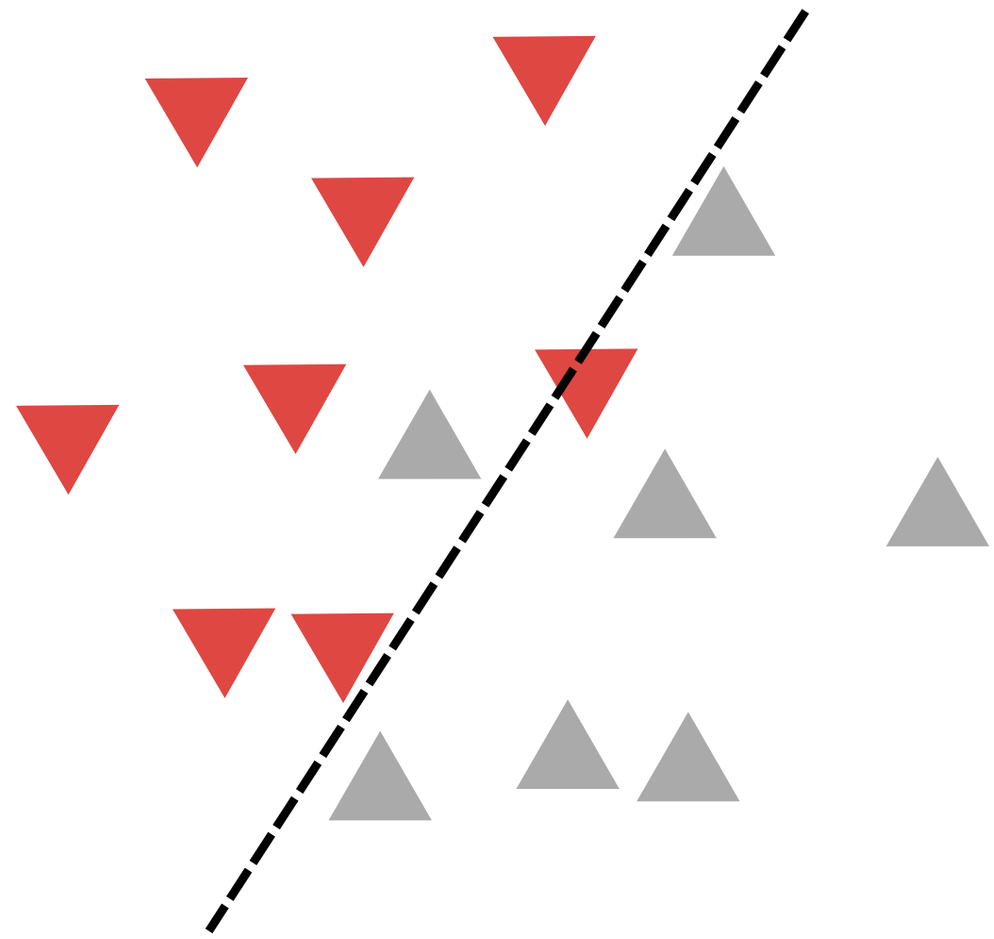
Case study: logistic regression



data $(x_n, y_n) \in \mathbb{R}^d \times \{\pm 1\}$

parameter $\theta \in \mathbb{R}^d$

Case study: logistic regression

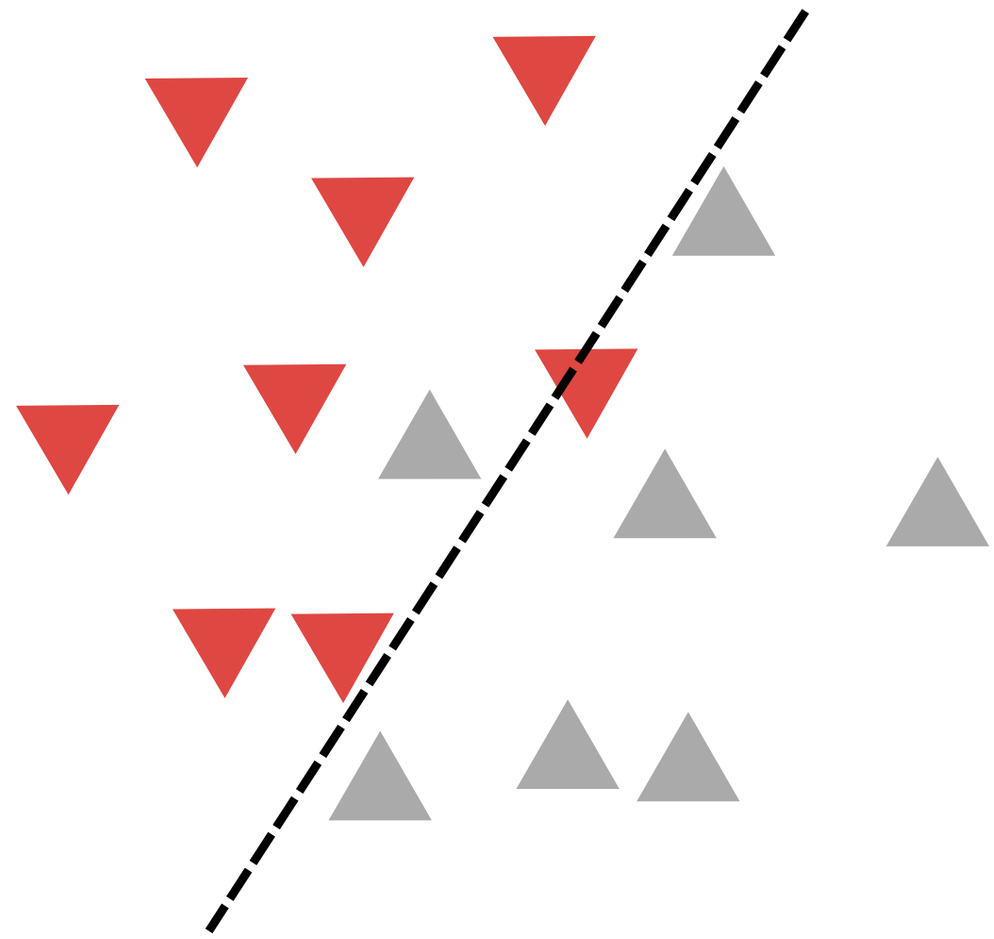


data $(x_n, y_n) \in \mathbb{R}^d \times \{\pm 1\}$

parameter $\theta \in \mathbb{R}^d$

log-likelihood $\log p(y_n \,|\, x_n, \theta) = -\log(1 + e^{-y_n x_n \cdot \theta})$

Case study: logistic regression

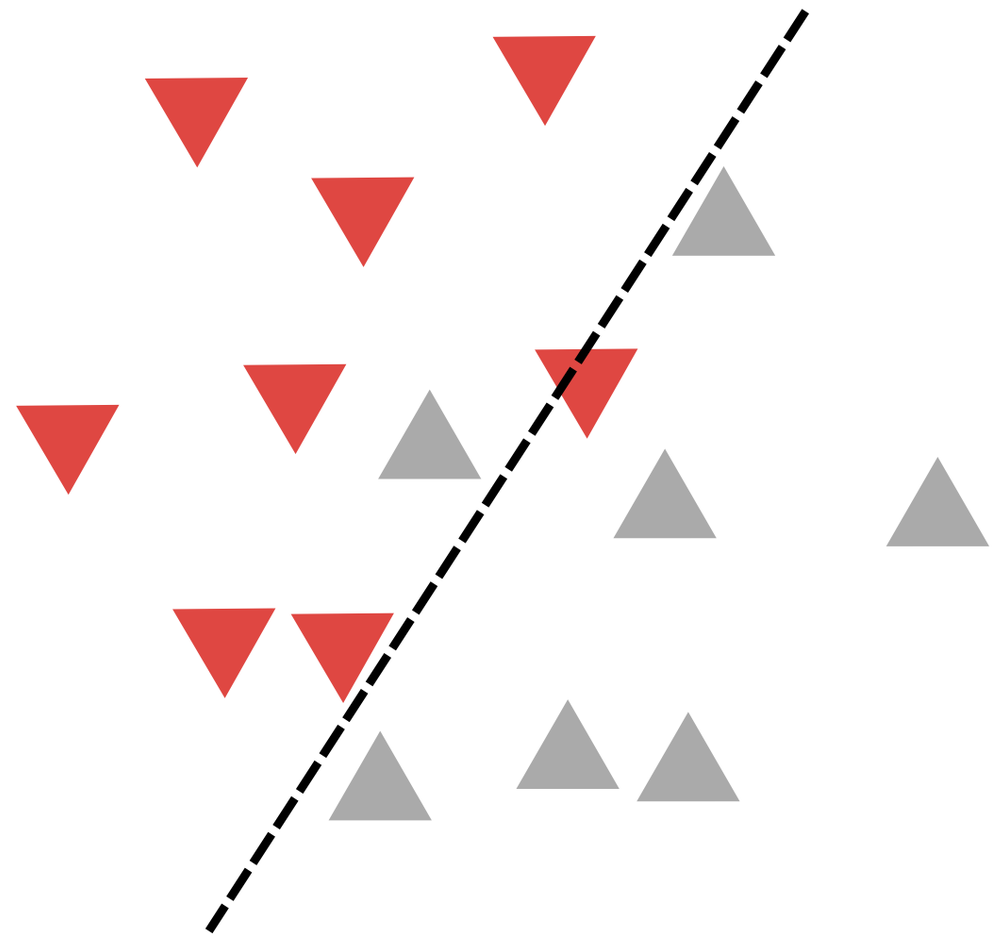


data $(x_n, y_n) \in \mathbb{R}^d \times \{\pm 1\}$

parameter $\theta \in \mathbb{R}^d$

$$\text{log-likelihood} \quad \log p(y_n \,|\, x_n, \theta) = -\log(1 + e^{-y_n x_n \cdot \theta})$$
$$= \phi(y_n x_n \cdot \theta)$$

Case study: logistic regression

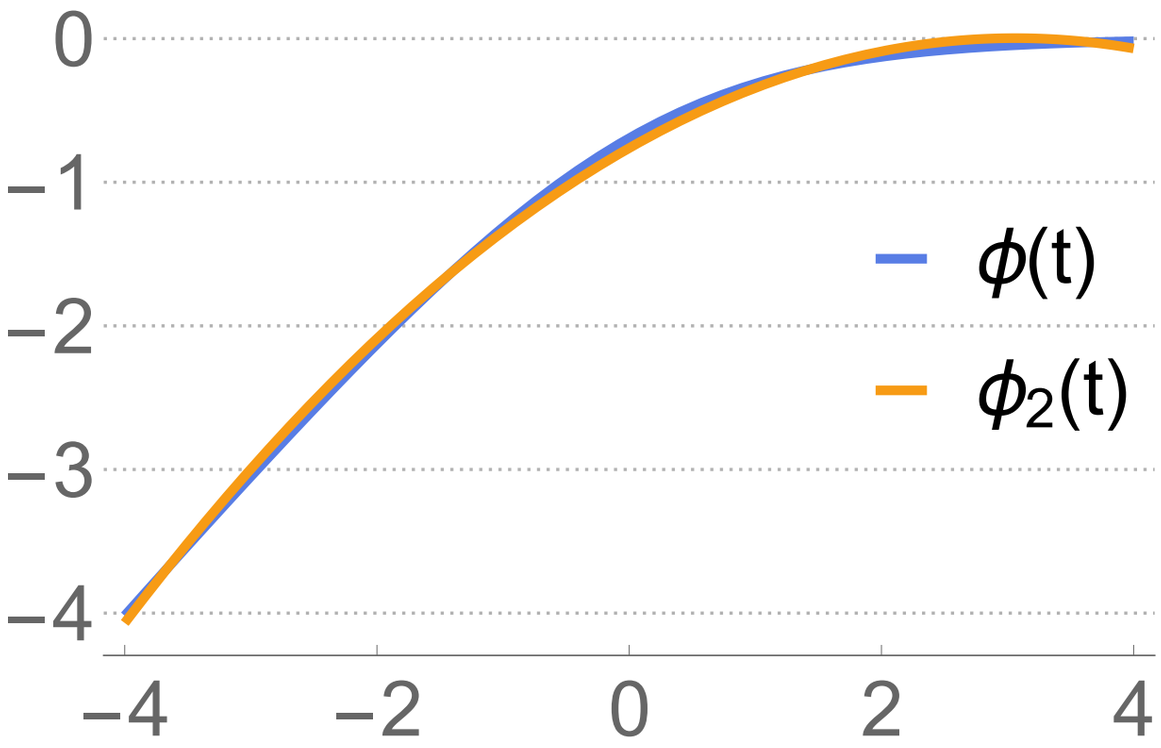


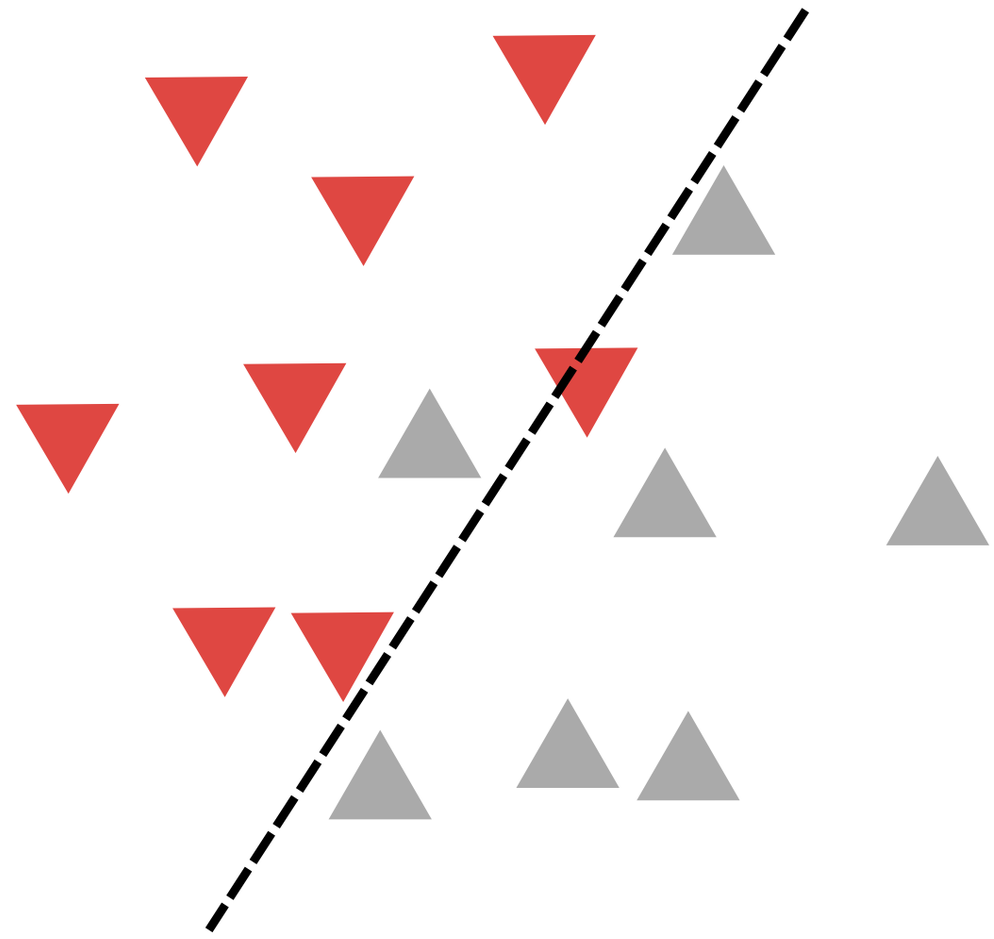
data $(x_n, y_n) \in \mathbb{R}^d \times \{\pm 1\}$

parameter $\theta \in \mathbb{R}^d$

$$\text{log-likelihood} \;\; \log p(y_n \,|\, x_n, \theta) = -\log(1 + e^{-y_n x_n \cdot \theta})$$
$$= \phi(y_n x_n \cdot \theta) \approx \phi_2(y_n x_n \cdot \theta)$$

Case study: logistic regression





data $(x_n, y_n) \in \mathbb{R}^d \times \{\pm 1\}$

parameter $\theta \in \mathbb{R}^d$

log-likelihood $\log p(y_n \mid x_n, \theta) = -\log(1 + e^{-y_n x_n \cdot \theta})$

$= \phi(y_n x_n \cdot \theta) \approx \phi_2(y_n x_n \cdot \theta)$

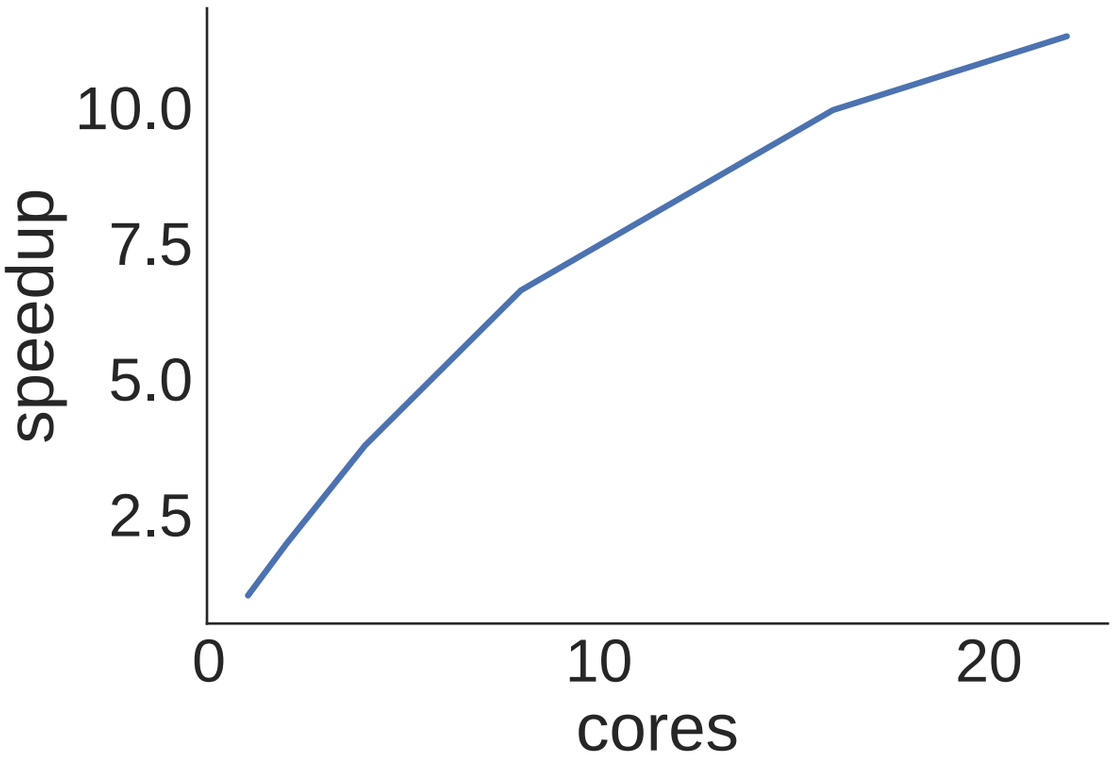
PASS-GLM effective in distributed and streaming settings

PASS-GLM effective in distributed and streaming settings

Criteo advertising dataset

PASS-GLM effective in distributed and streaming settings

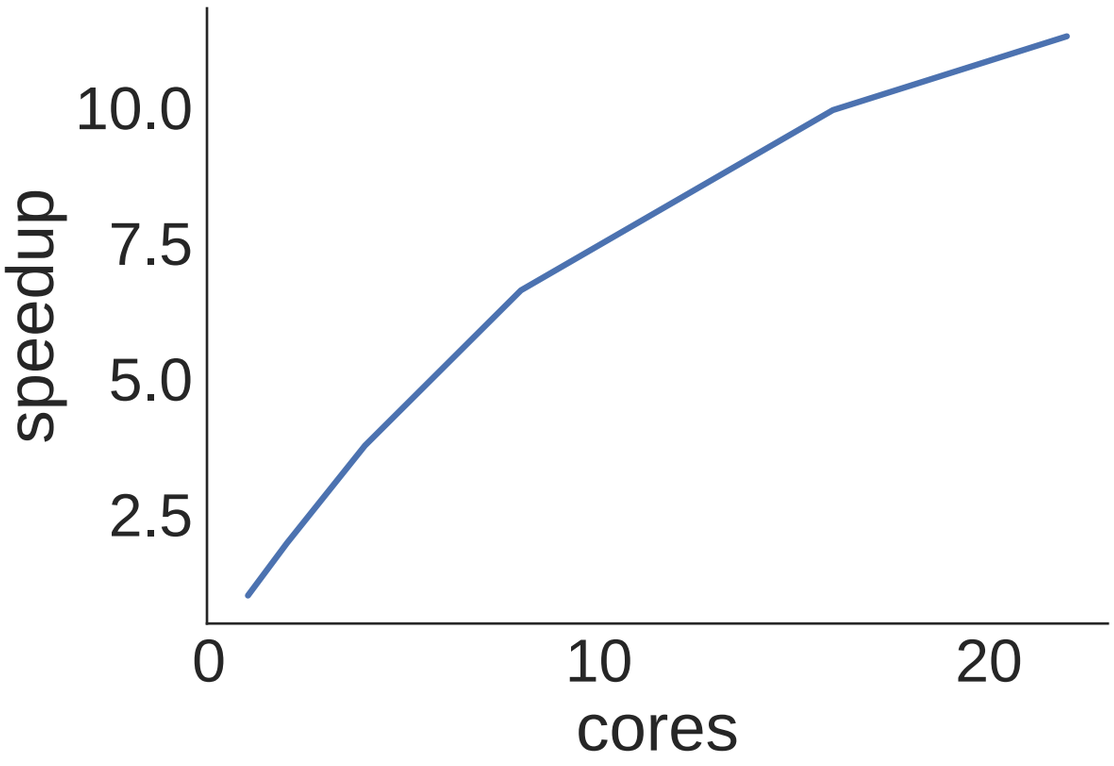
Criteo advertising dataset



- Distributed

PASS-GLM effective in distributed and streaming settings

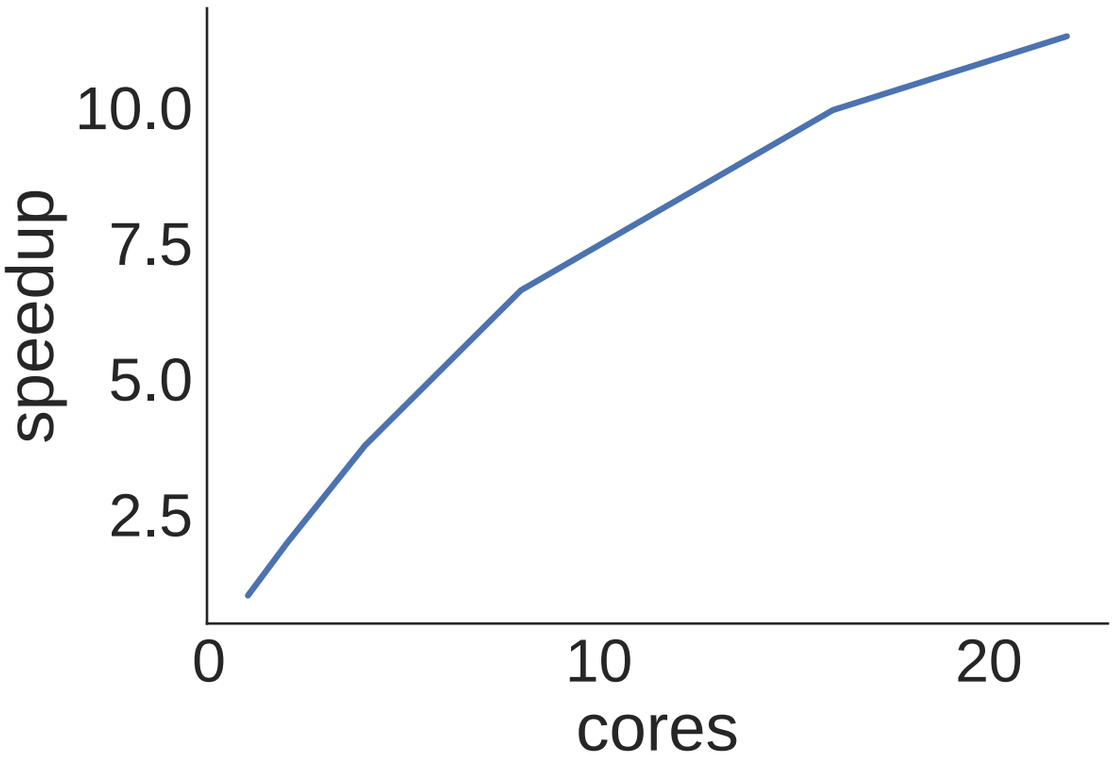
Criteo advertising dataset



- Distributed
  - 6M observations with 1K covariates

PASS-GLM effective in distributed and streaming settings

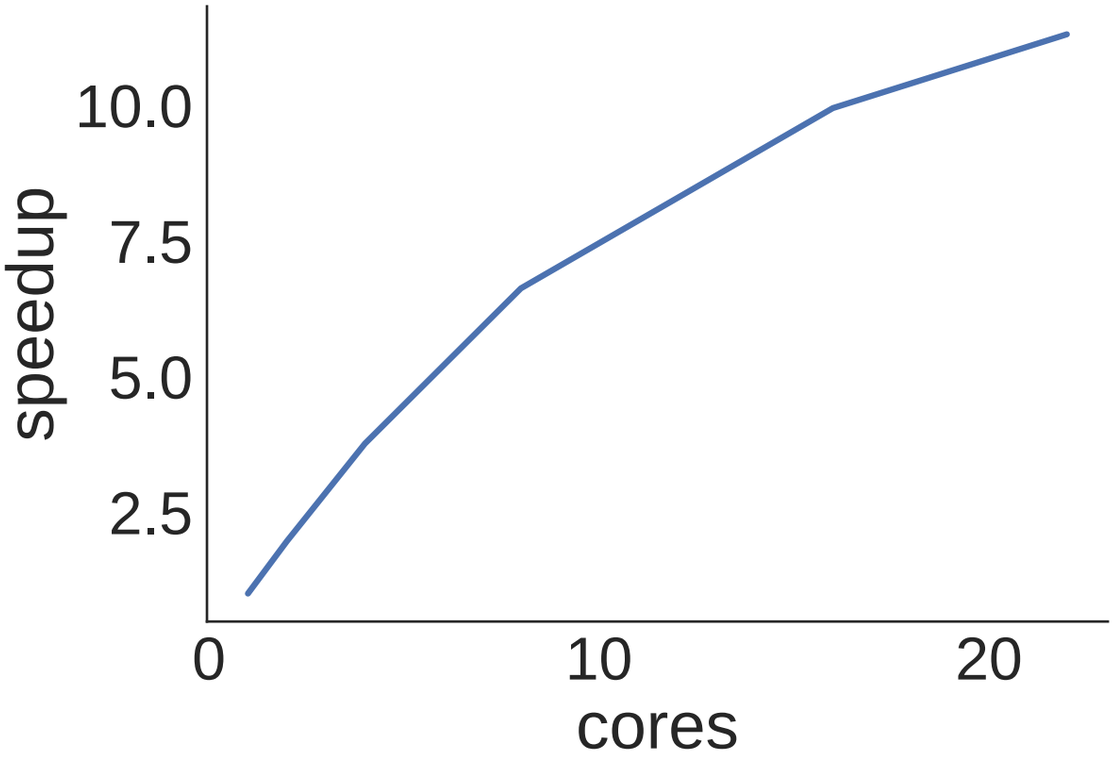
Criteo advertising dataset

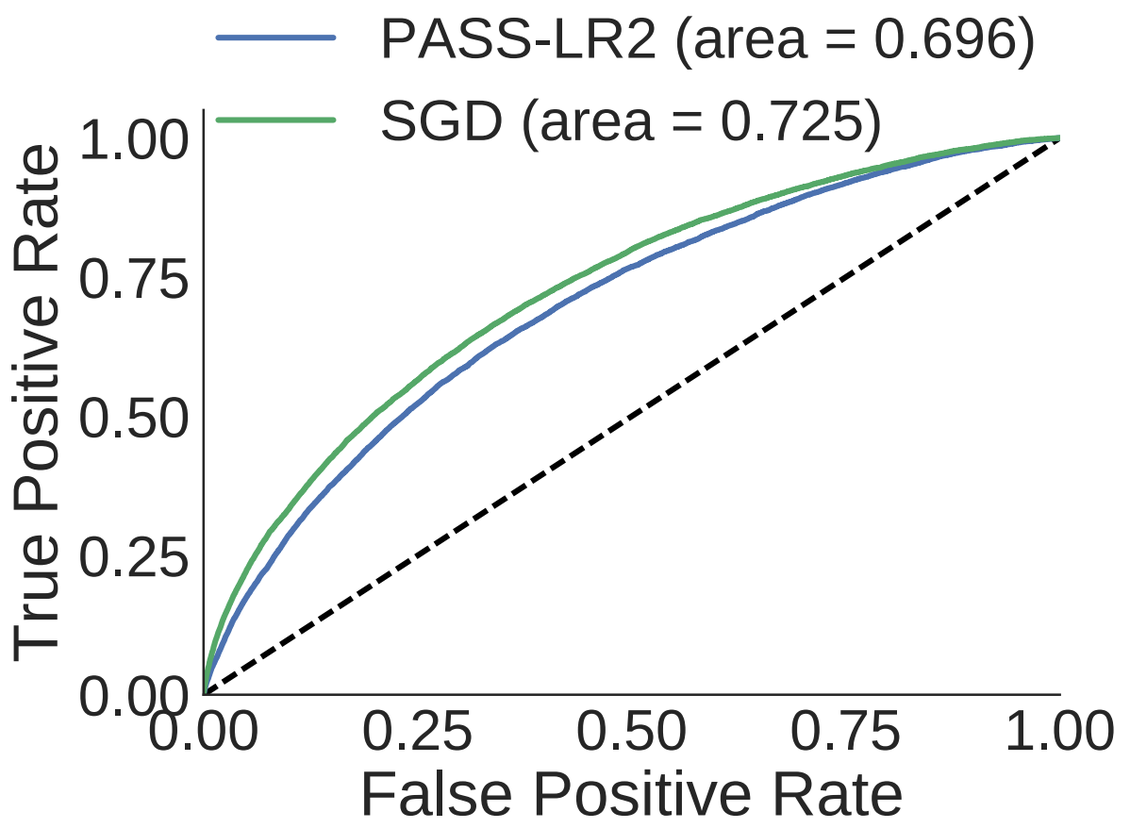


- Distributed
  - 6M observations with 1K covariates
  - **16 seconds** using 22 cores

PASS-GLM effective in distributed and streaming settings

Criteo advertising dataset

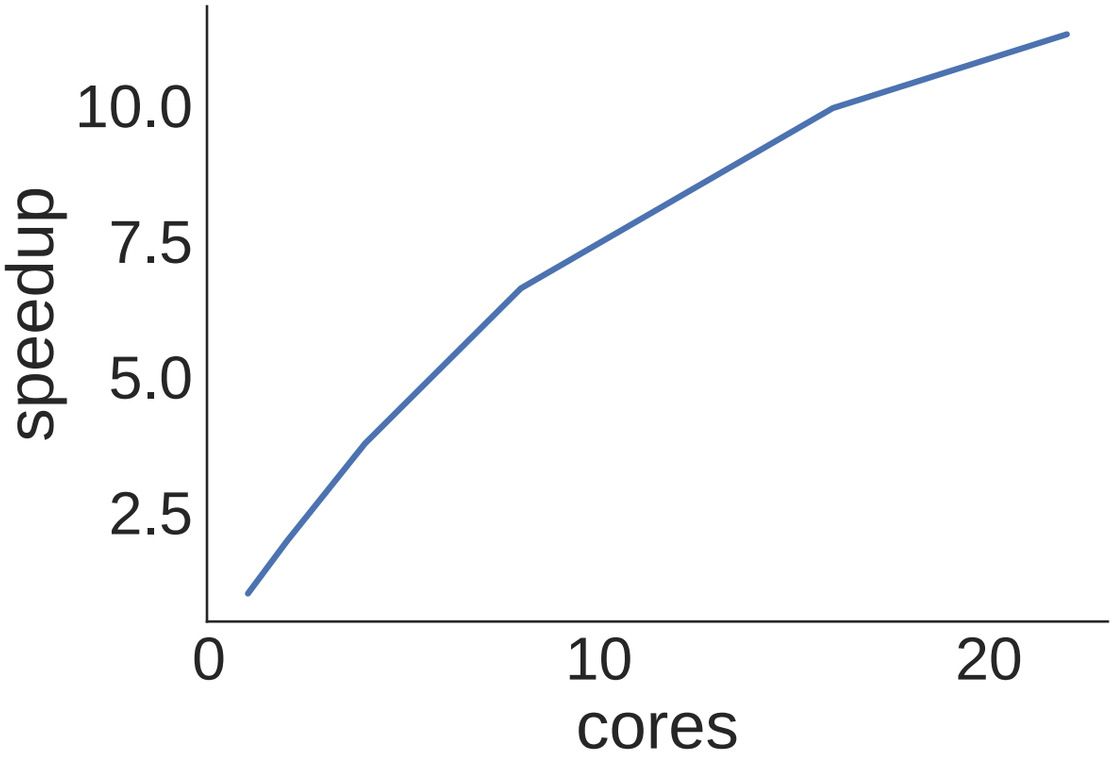


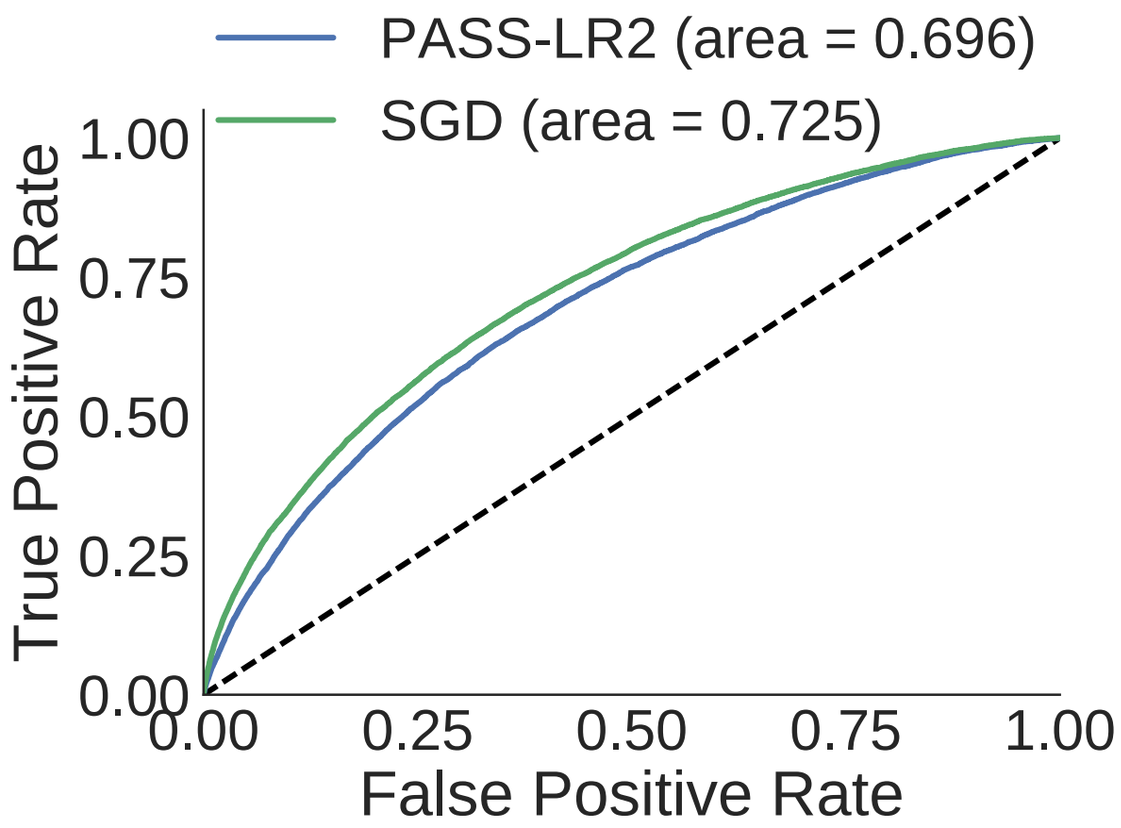


- Distributed
  - 6M observations with 1K covariates
  - **16 seconds** using 22 cores
- Streaming

PASS-GLM effective in distributed and streaming settings

Criteo advertising dataset

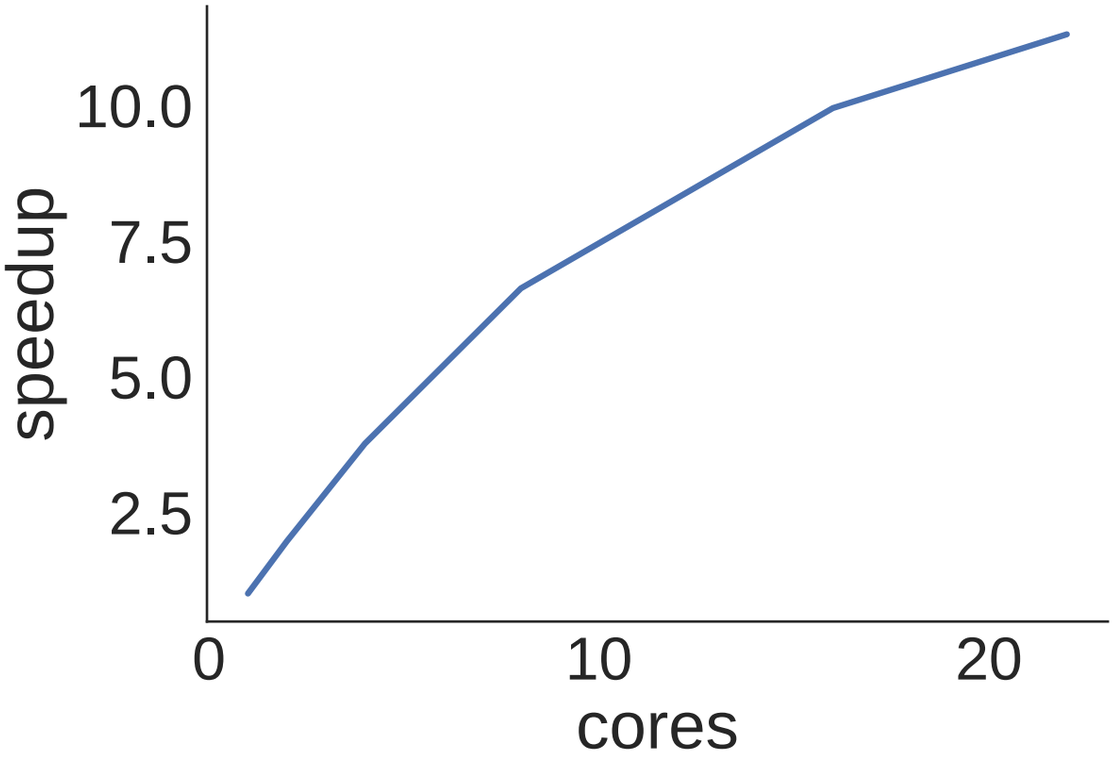


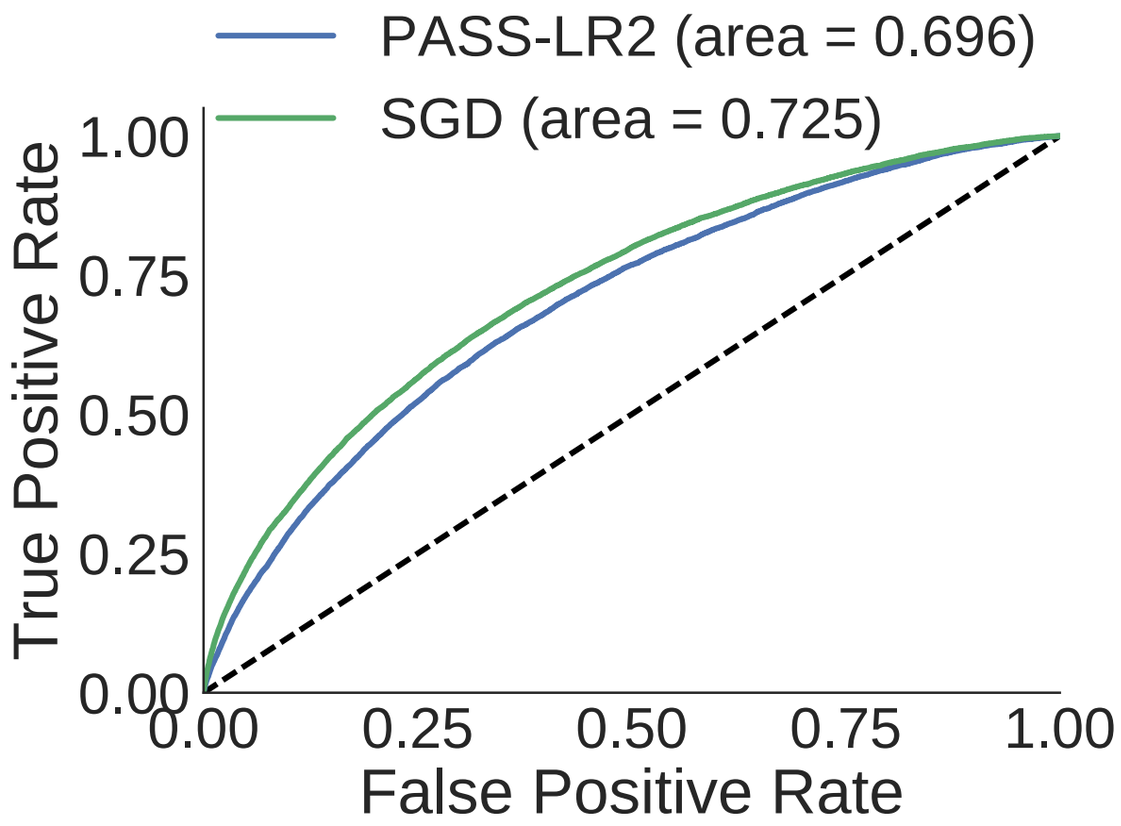


- Distributed
  - 6M observations with 1K covariates
  - **16 seconds** using 22 cores
- Streaming
  - 40M observations with 20K covariates

PASS-GLM effective in distributed and streaming settings

Criteo advertising dataset





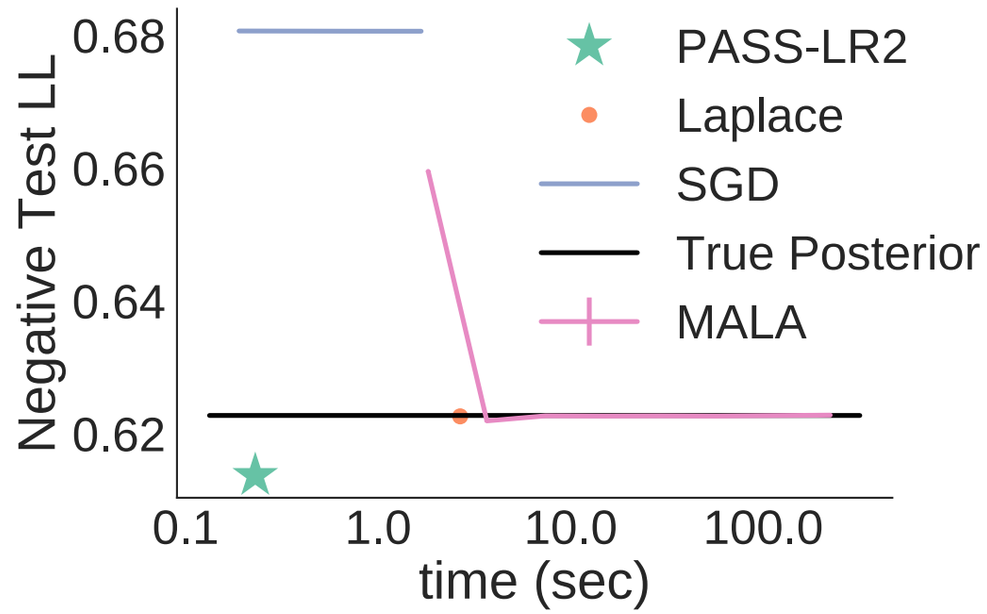
- Distributed
  - 6M observations with 1K covariates
  - **16 seconds** using 22 cores
- Streaming
  - 40M observations with 20K covariates
  - Competitive with SGD

Competitive approximation performance

- Webspam dataset
- N = 350,000
- d = 127

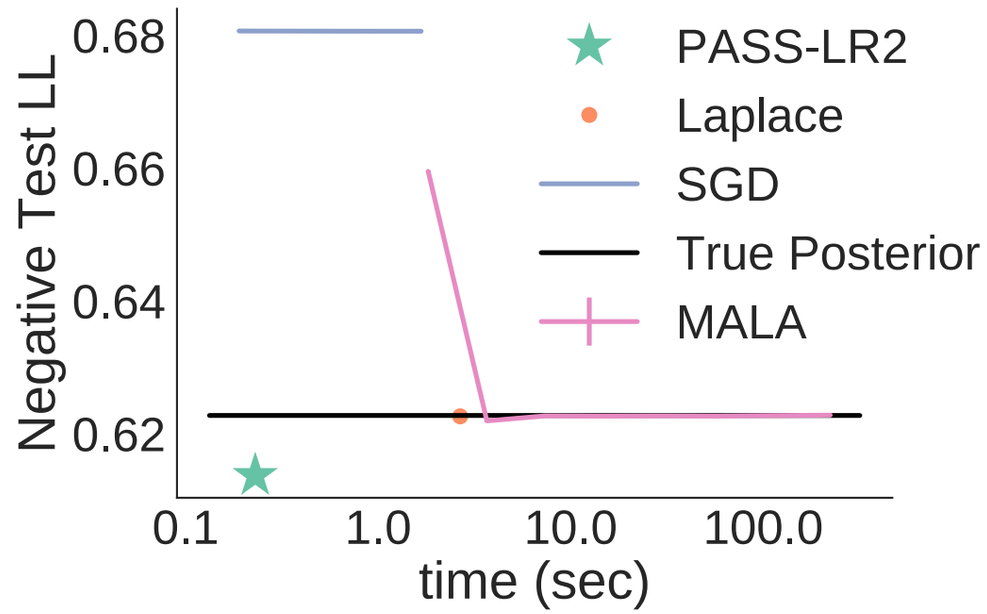
Competitive approximation performance

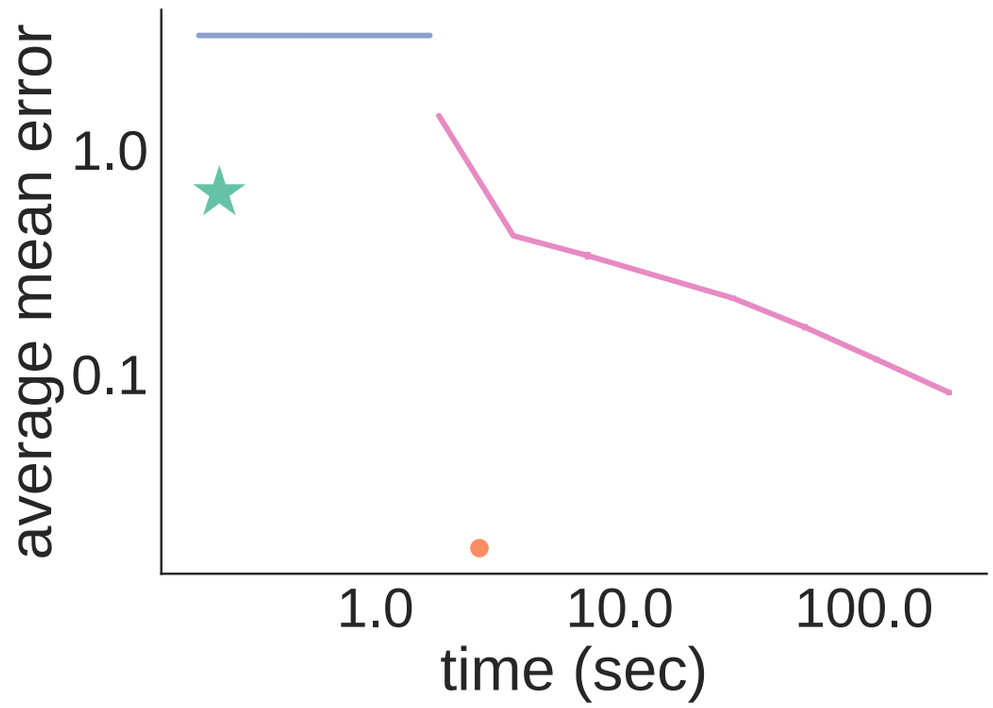
- Webspam dataset
- $N = 350{,}000$
- $d = 127$



Competitive approximation performance

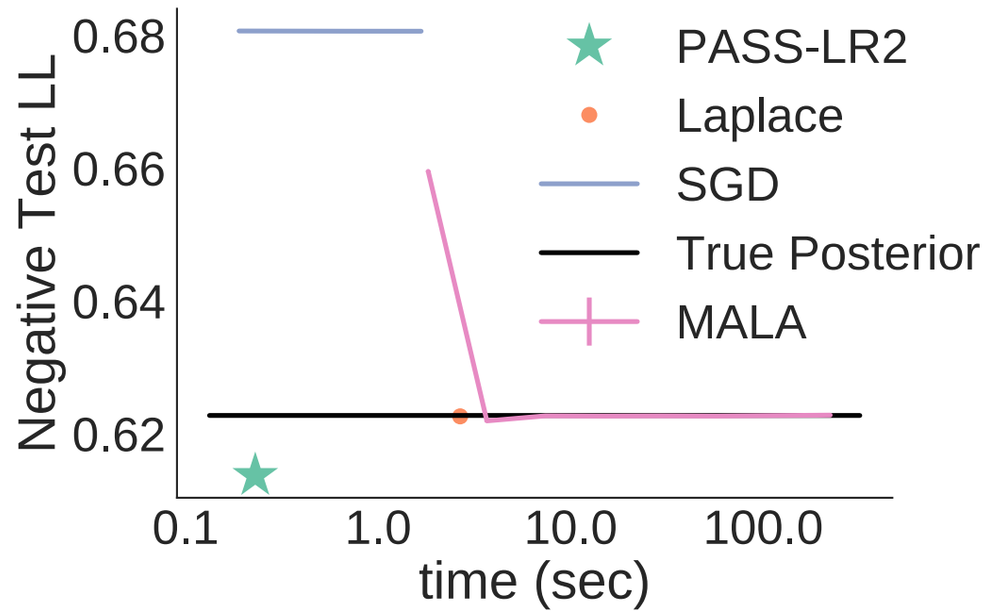
- Webspam dataset
- $N = 350{,}000$
- $d = 127$

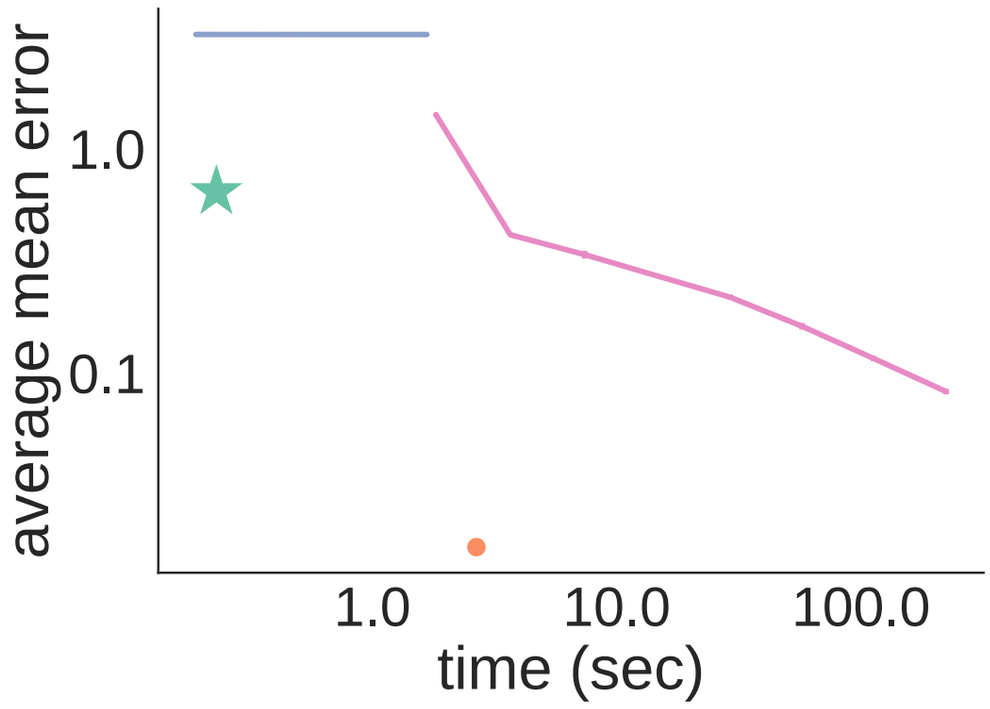


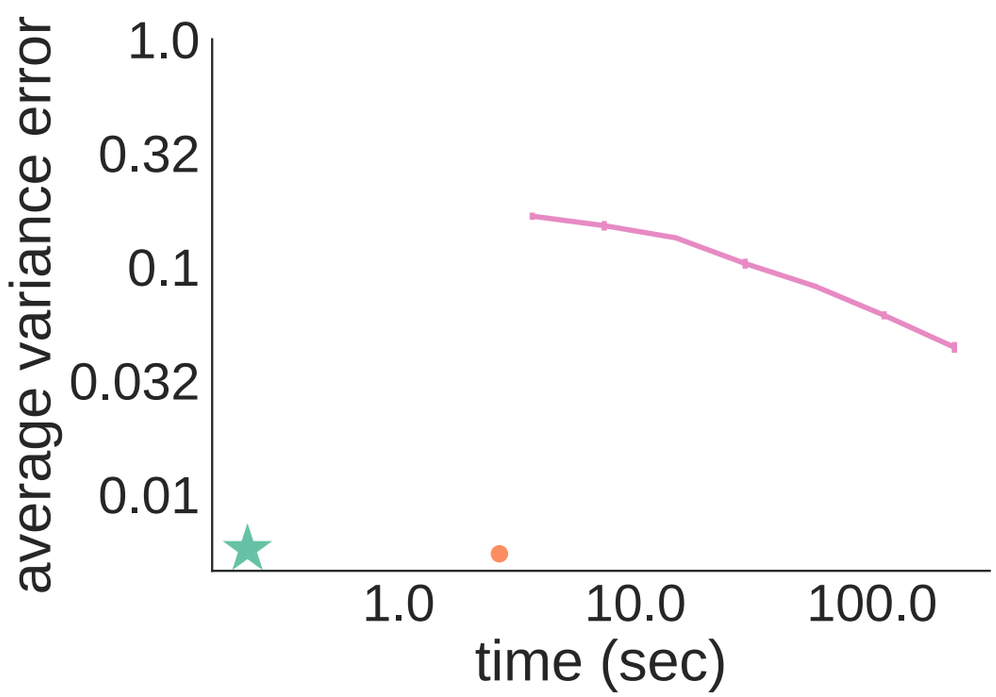


Competitive approximation performance

- Webspam dataset
- $N = 350{,}000$
- $d = 127$



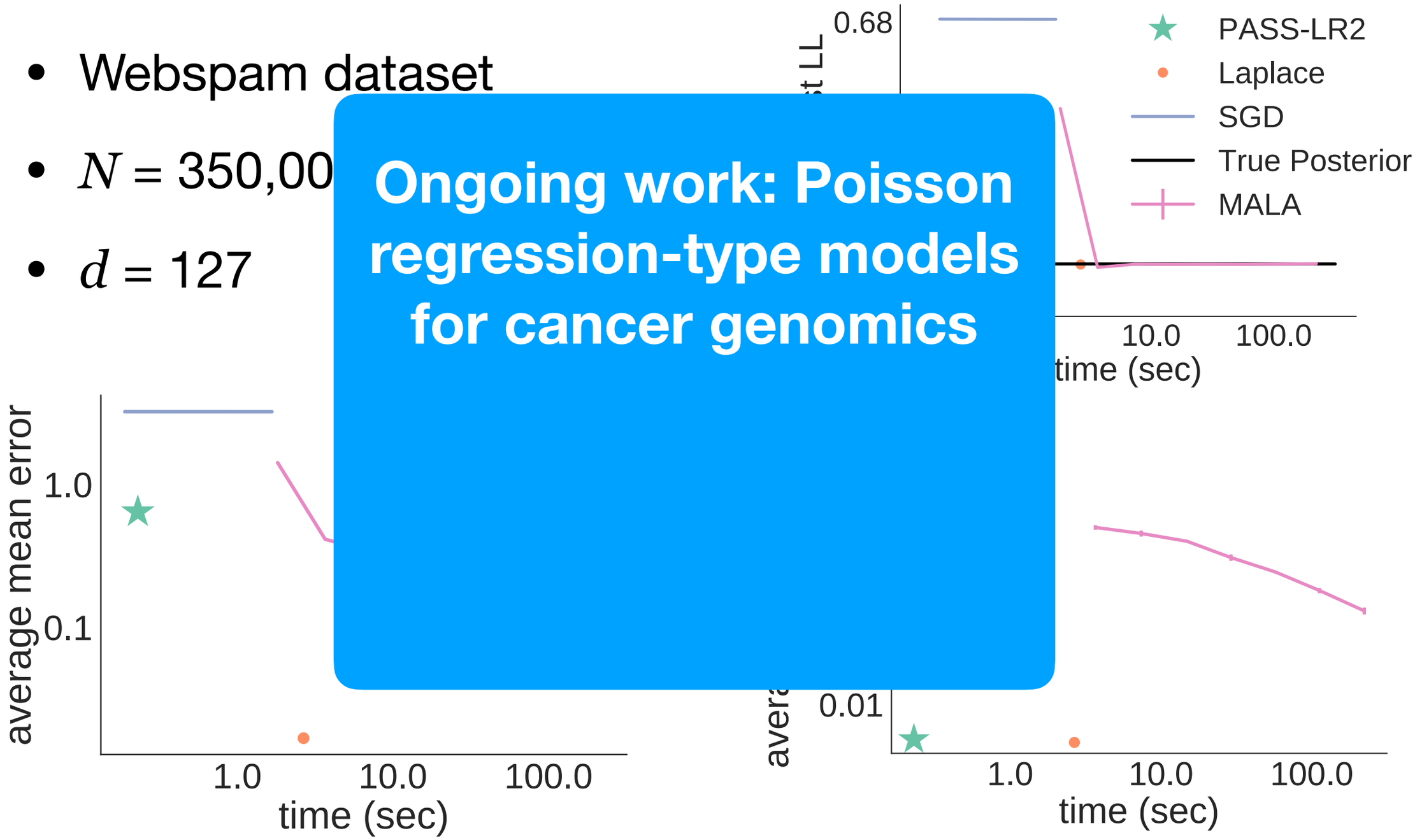




Competitive approximation performance

- Webspam dataset
- $N = 350{,}00$
- $d = 127$

Ongoing work: Poisson regression-type models for cancer genomics

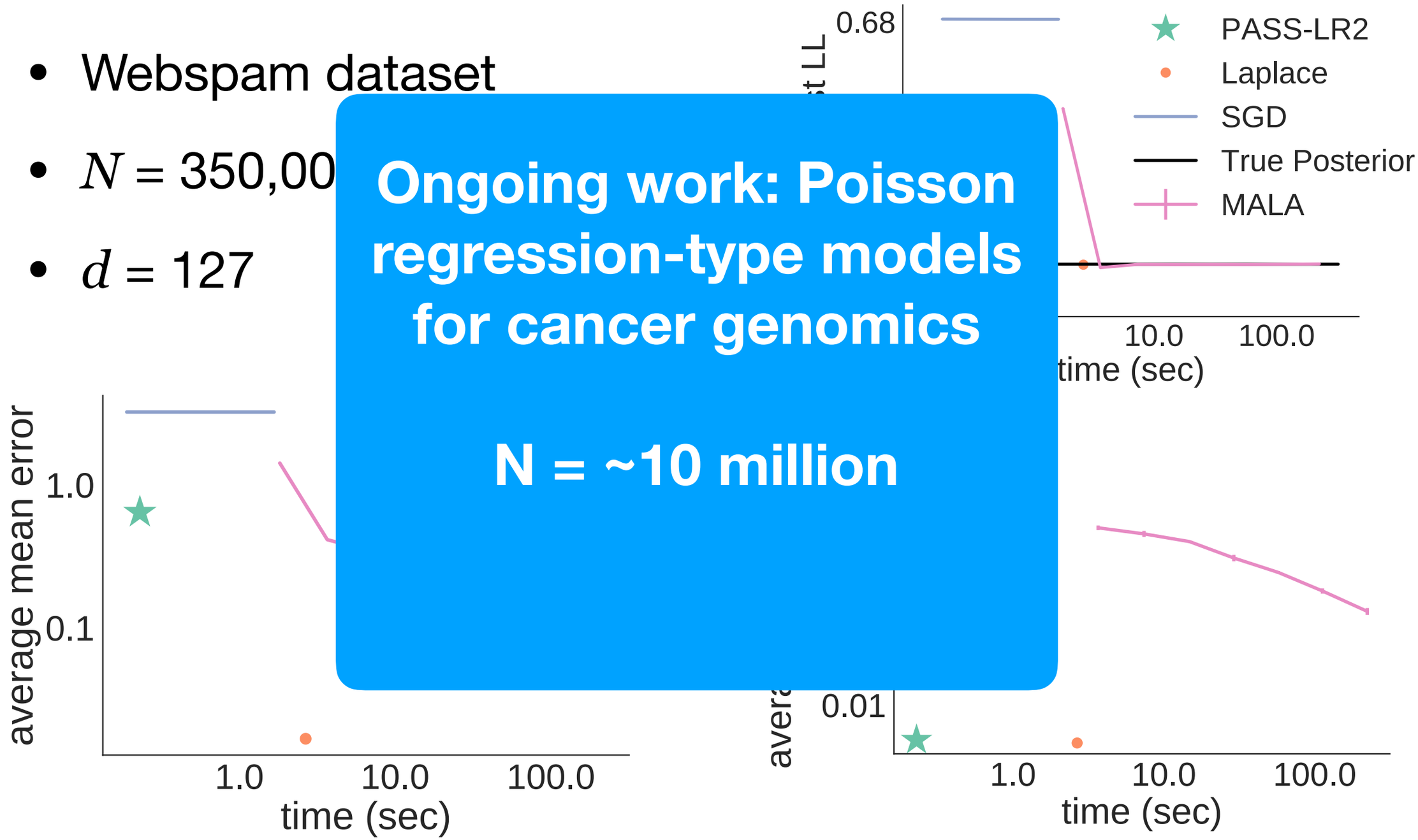


Competitive approximation performance

- Webspam dataset
- $N = 350{,}00$
- $d = 127$

Ongoing work: Poisson regression-type models for cancer genomics

N = ~10 million

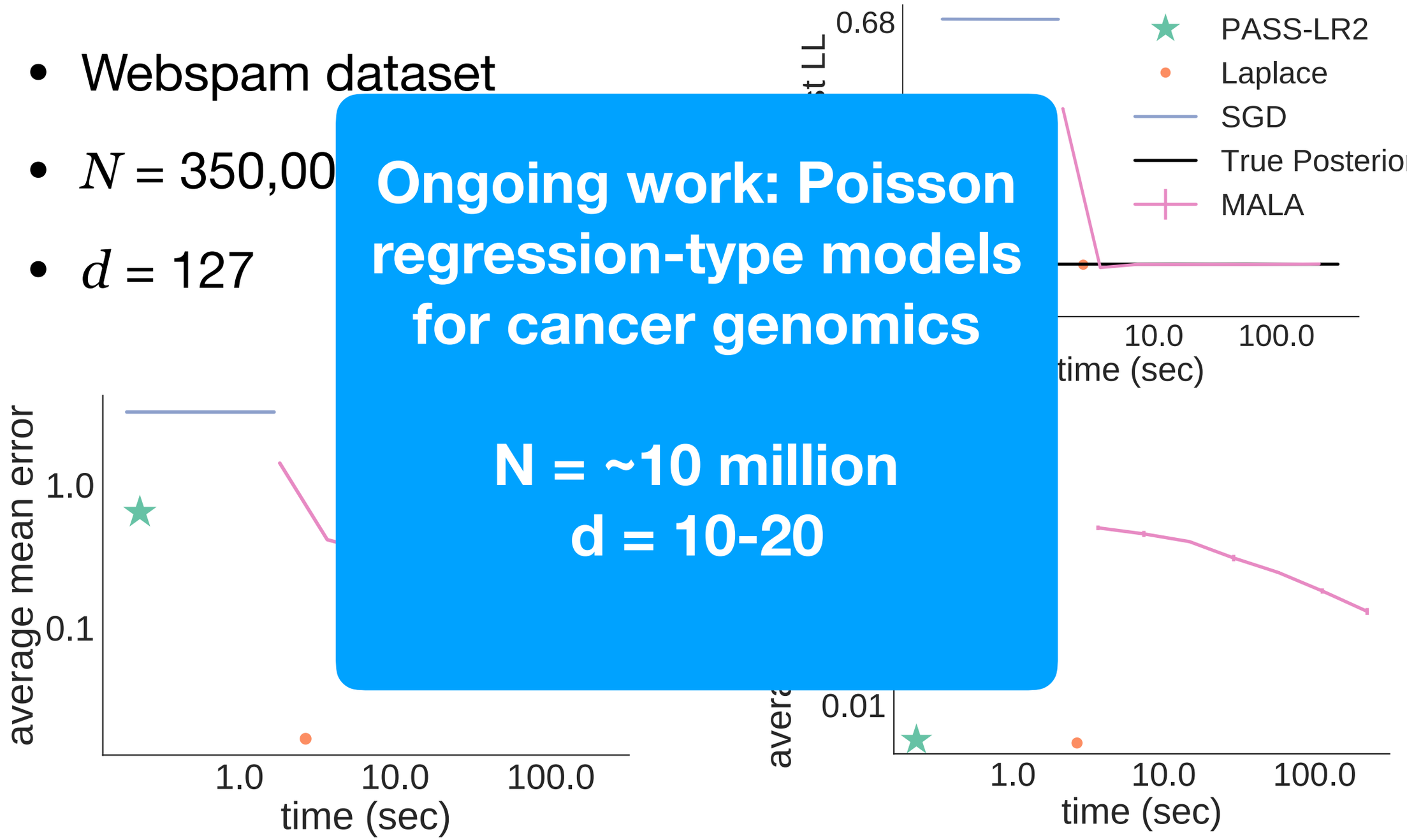


Competitive approximation performance

- Webspam dataset
- $N = 350{,}00$
- $d = 127$

Ongoing work: Poisson regression-type models for cancer genomics

N = ~10 million
d = 10-20

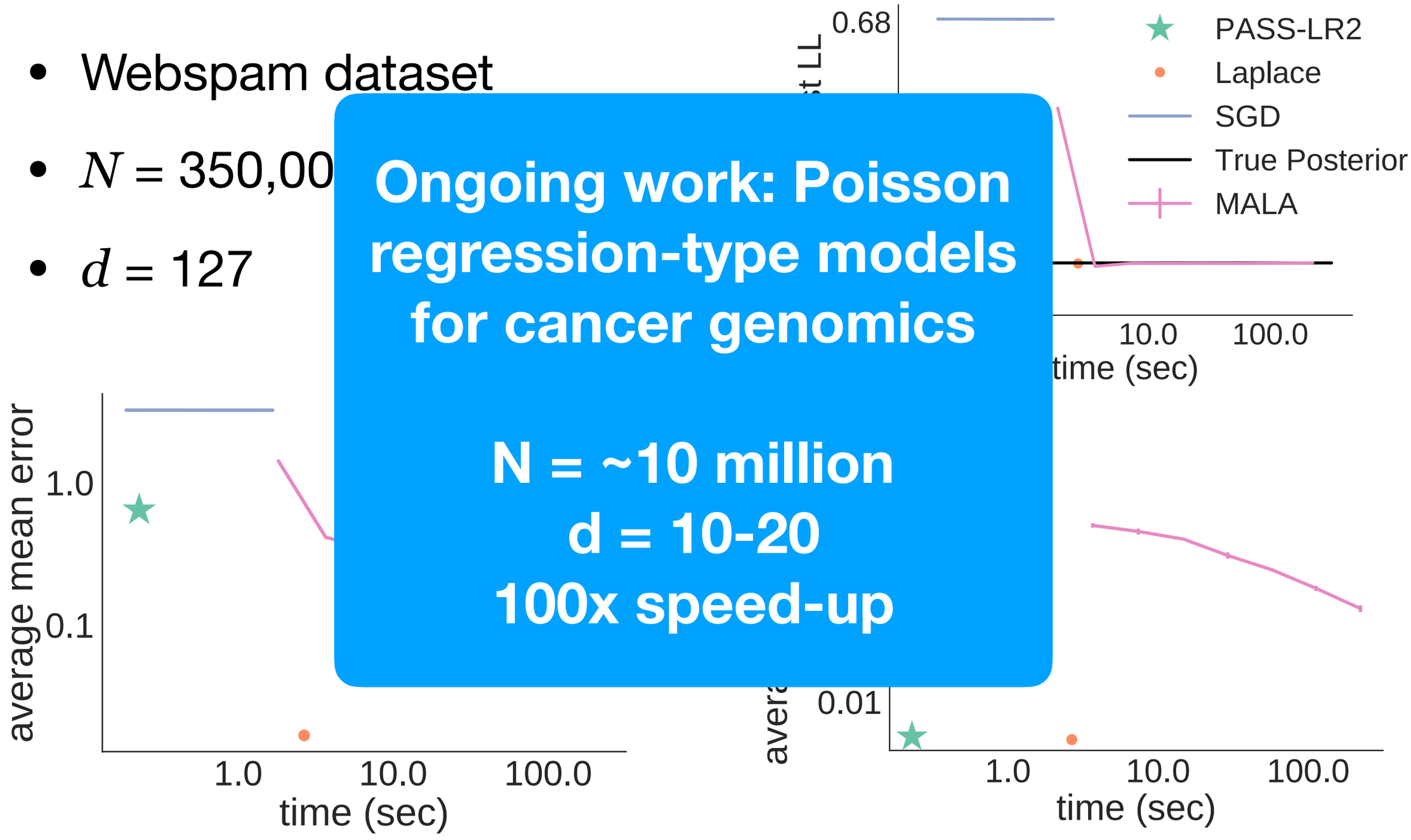


Competitive approximation performance

- Webspam dataset
- N = 350,00
- d = 127

Ongoing work: Poisson regression-type models for cancer genomics

N = ~10 million
d = 10-20
100x speed-up



Roadmap

- Approximate Bayes review
- Likelihood approximation and dataset compression
- Approximate sufficient statistics
- Accuracy guarantees

Accuracy guarantees

Accuracy guarantees

- **Problem:** existing scalable methods lack accuracy guarantees 

Accuracy guarantees

- **Problem:** existing scalable methods lack accuracy guarantees 
- **Question:** how do we measure closeness of the exact and approximate posteriors?

Meaningful

Accuracy guarantees

- **Problem:** existing scalable methods lack accuracy guarantees 
- **Question:** how do we measure closeness of the exact and approximate posteriors?

Meaningful Accuracy guarantees

- **Problem:** existing scalable methods lack accuracy guarantees 
- **Question:** how do we measure closeness of the exact and approximate posteriors?
- **Recall:** want to compute means, variances, tail probabilities, etc.

Meaningful

Accuracy guarantees

- **Problem:** existing scalable methods lack accuracy guarantees 
- **Question:** how do we measure closeness of the exact and approximate posteriors?
- **Recall:** want to compute means, variances, tail probabilities, etc.
- Good choice of measure: **1- and 2-Wasserstein distances** d_W

Meaningful

Accuracy guarantees

- **Problem:** existing scalable methods lack accuracy guarantees 
- **Question:** how do we measure closeness of the exact and approximate posteriors?
- **Recall:** want to compute means, variances, tail probabilities, etc.
- Good choice of measure: **1- and 2-Wasserstein distances** d_W
- **Why?** $d_W(p, q)$ small implies

Meaningful

Accuracy guarantees

- **Problem:** existing scalable methods lack accuracy guarantees ☠️
- **Question:** how do we measure closeness of the exact and approximate posteriors?
- **Recall:** want to compute means, variances, tail probabilities, etc.
- Good choice of measure: **1- and 2-Wasserstein distances** d_W
- **Why?** $d_W(p, q)$ small implies
  - means and variances close ✅

Meaningful

Accuracy guarantees

- **Problem:** existing scalable methods lack accuracy guarantees ☠️
- **Question:** how do we measure closeness of the exact and approximate posteriors?
- **Recall:** want to compute means, variances, tail probabilities, etc.
- Good choice of measure: **1- and 2-Wasserstein distances** d_W
- **Why?** $d_W(p, q)$ small implies
  - means and variances close ✅
  - (smoothed) tail probabilities close ✅

Wasserstein distance with approximate likelihoods

Wasserstein distance with approximate likelihoods

$$\pi(\theta|Y) = \frac{p(Y|\theta)\pi_0(\theta)}{Z}$$

Wasserstein distance with approximate likelihoods

$$\pi(\theta|Y) = \frac{p(Y|\theta)\pi_0(\theta)}{Z} \qquad \tilde{\pi}(\theta|Y) = \frac{e^{\ell(\theta, g(Y))}\pi_0(\theta)}{\tilde{Z}}$$

Wasserstein distance with approximate likelihoods

$$\pi(\theta|Y) = \frac{p(Y|\theta)\pi_0(\theta)}{Z} \qquad \tilde{\pi}(\theta|Y) = \frac{e^{\ell(\theta, g(Y))}\pi_0(\theta)}{\tilde{Z}}$$

$$\varepsilon(\theta) = \|\nabla_\theta \log p(Y|\theta) - \nabla_\theta \ell(\theta, g(Y))\|_2$$

Wasserstein distance with approximate likelihoods

$$\pi(\theta|Y) = \frac{p(Y|\theta)\pi_0(\theta)}{Z} \qquad \tilde{\pi}(\theta|Y) = \frac{e^{\ell(\theta, g(Y))}\pi_0(\theta)}{\tilde{Z}}$$

$$\varepsilon(\theta) = \|\nabla_\theta \log p(Y|\theta) - \nabla_\theta \ell(\theta, g(Y))\|_2$$

Theorem (H. & Zou 2017, H. 2018). Assume

Wasserstein distance with approximate likelihoods

$$\pi(\theta|Y) = \frac{p(Y|\theta)\pi_0(\theta)}{Z} \qquad \tilde{\pi}(\theta|Y) = \frac{e^{\ell(\theta, g(Y))}\pi_0(\theta)}{\tilde{Z}}$$

$$\varepsilon(\theta) = \|\nabla_\theta \log p(Y|\theta) - \nabla_\theta \ell(\theta, g(Y))\|_2$$

Theorem (H. & Zou 2017, H. 2018). Assume

- π is “well-behaved” and

Wasserstein distance with approximate likelihoods

$$\pi(\theta|Y) = \frac{p(Y|\theta)\pi_0(\theta)}{Z} \qquad \tilde{\pi}(\theta|Y) = \frac{e^{\ell(\theta, g(Y))}\pi_0(\theta)}{\tilde{Z}}$$

$$\varepsilon(\theta) = \|\nabla_\theta \log p(Y|\theta) - \nabla_\theta \ell(\theta, g(Y))\|_2$$

Theorem (H. & Zou 2017, H. 2018). Assume

- π is "well-behaved" and
- $\varepsilon(\theta) \leq \varepsilon$.

Wasserstein distance with approximate likelihoods

$$\pi(\theta|Y) = \frac{p(Y|\theta)\pi_0(\theta)}{Z} \qquad \tilde{\pi}(\theta|Y) = \frac{e^{\ell(\theta, g(Y))}\pi_0(\theta)}{\tilde{Z}}$$

$$\varepsilon(\theta) = \|\nabla_\theta \log p(Y|\theta) - \nabla_\theta \ell(\theta, g(Y))\|_2$$

Theorem (H. & Zou 2017, H. 2018). Assume

- π is "well-behaved" and
- $\varepsilon(\theta) \leq \varepsilon$.

Then $d_W(\pi, \tilde{\pi}) \leq c_\pi \varepsilon$.

PASS-LR provides a high-quality approximation

$\tilde{\pi}_M =$ order M PASS-LR approximate posterior

PASS-LR provides a high-quality approximation

$\tilde{\pi}_M =$ order M PASS-LR approximate posterior

Theorem (H., Adams, Broderick 2017).

PASS-LR provides a high-quality approximation

$\tilde{\pi}_M =$ order M PASS-LR approximate posterior

Theorem (H., Adams, Broderick 2017).
Assume prior and data are “well-behaved”.

PASS-LR provides a high-quality approximation

$\tilde{\pi}_M =$ order M PASS-LR approximate posterior

Theorem (H., Adams, Broderick 2017).
Assume prior and data are “well-behaved”.

Then there exist $c > 0$ and $0 < r < 1$ such that

PASS-LR provides a high-quality approximation

$\tilde{\pi}_M =$ order M PASS-LR approximate posterior

Theorem (H., Adams, Broderick 2017). Assume prior and data are "well-behaved".

Then there exist $c > 0$ and $0 < r < 1$ such that

$$d_W(\pi, \tilde{\pi}_M) \leq c d r^M$$

PASS-LR provides a high-quality approximation

$\tilde{\pi}_M =$ order M PASS-LR approximate posterior

Theorem (H., Adams, Broderick 2017). Assume prior and data are "well-behaved".

Then there exist $c > 0$ and $0 < r < 1$ such that

$$d_W(\pi, \tilde{\pi}_M) \leq cdr^M$$

- Similar results for other GLMs

Concluding thoughts

Concluding thoughts

- Empirical work

Concluding thoughts

- Empirical work
    - Hierarchical models

Concluding thoughts

- Empirical work
    - Hierarchical models
    - Applications to a wider range of GLMs

Concluding thoughts

- Empirical work
  - Hierarchical models
  - Applications to a wider range of GLMs
  - Practitioner buy-in

Concluding thoughts

- Empirical work
    - Hierarchical models
    - Applications to a wider range of GLMs
    - Practitioner buy-in
- Very- and ultra-high dimensional parameter spaces

Concluding thoughts

- Empirical work
    - Hierarchical models
    - Applications to a wider range of GLMs
    - Practitioner buy-in
- Very- and ultra-high dimensional parameter spaces
- Non-parametric models:

Concluding thoughts

- Empirical work
    - Hierarchical models
    - Applications to a wider range of GLMs
        - Practitioner buy-in
- Very- and ultra-high dimensional parameter spaces
- Non-parametric models:
    - Coresets for Gaussian processes, connections to inducing point methods

Concluding thoughts

- Empirical work
    - Hierarchical models
    - Applications to a wider range of GLMs
        - Practitioner buy-in
- Very- and ultra-high dimensional parameter spaces
- Non-parametric models:
    - Coresets for Gaussian processes, connections to inducing point methods
- Combinatorial parameter spaces

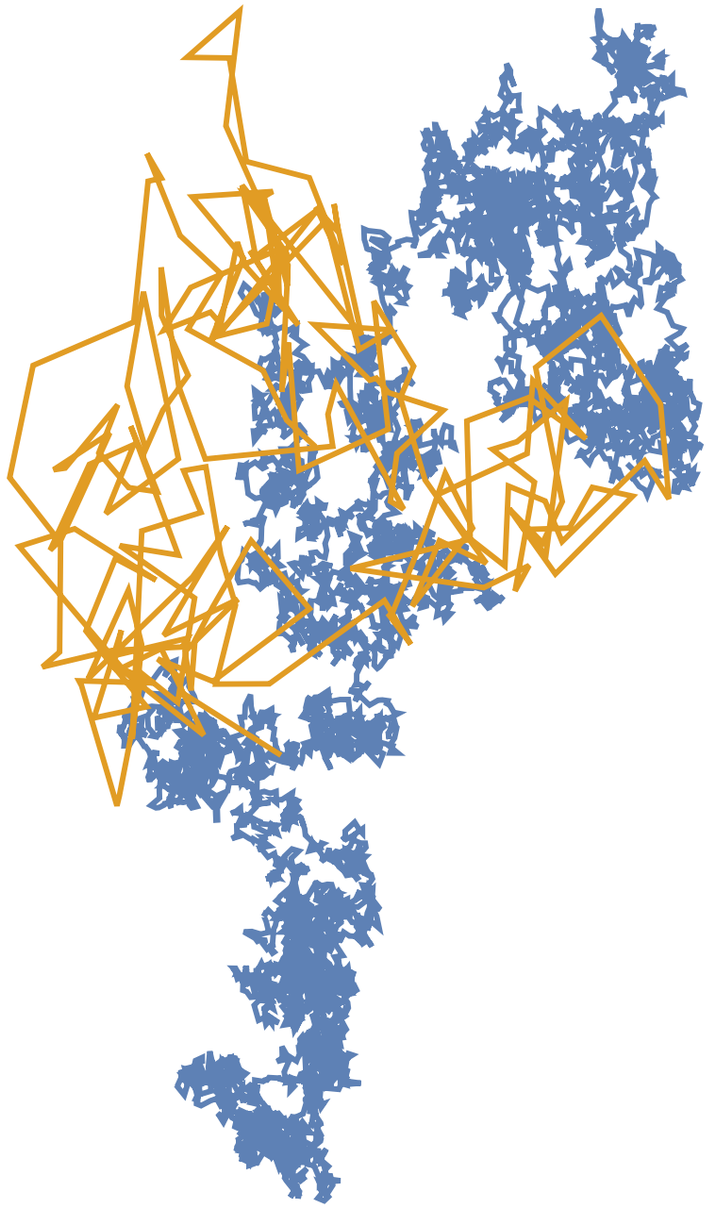
Thanks!

***J. H. Huggins**, R. P. Adams, T. Broderick. PASS-GLM: polynomial approximate sufficient statistics for scalable Bayesian GLM inference. NIPS, 2017.*

J. H. Huggins, J. Zou*. Quantifying the Accuracy of Approximate Diffusions and Markov Chains. AISTATS, 2017.*

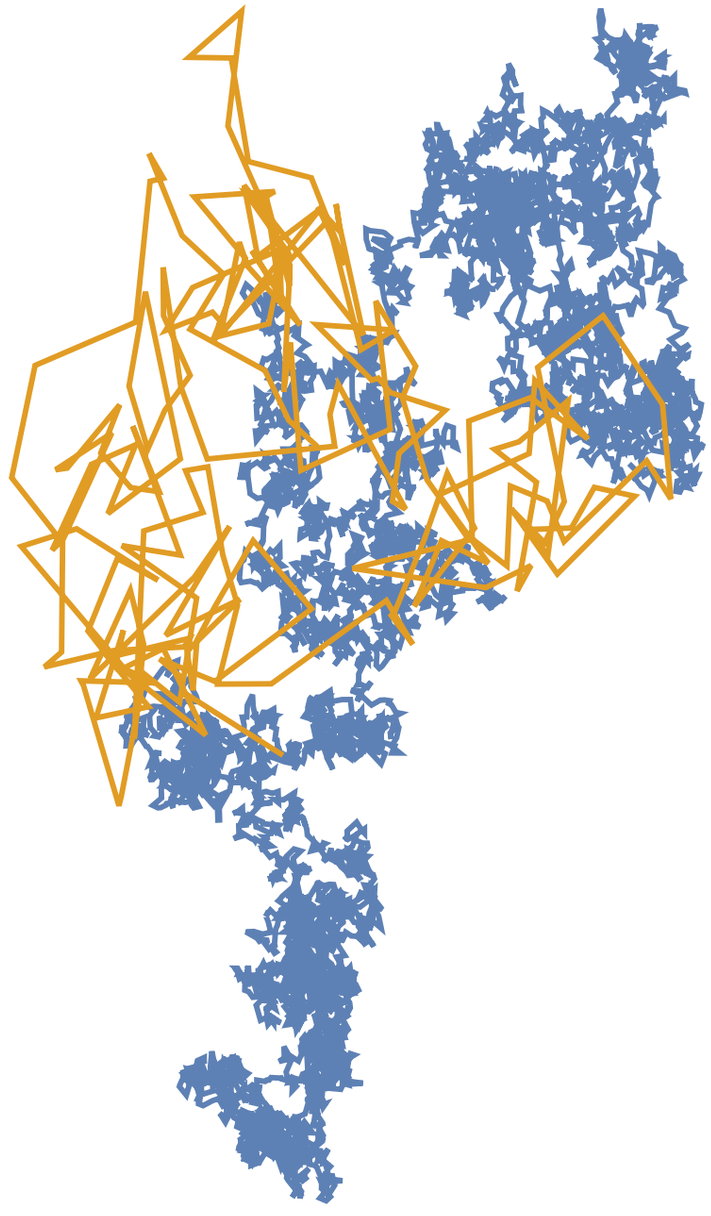
J. H. Huggins, T. Campbell, T. Broderick. Coresets for Scalable Bayesian Logistic Regression. NIPS, 2016.

Using diffusions to understand approximation error



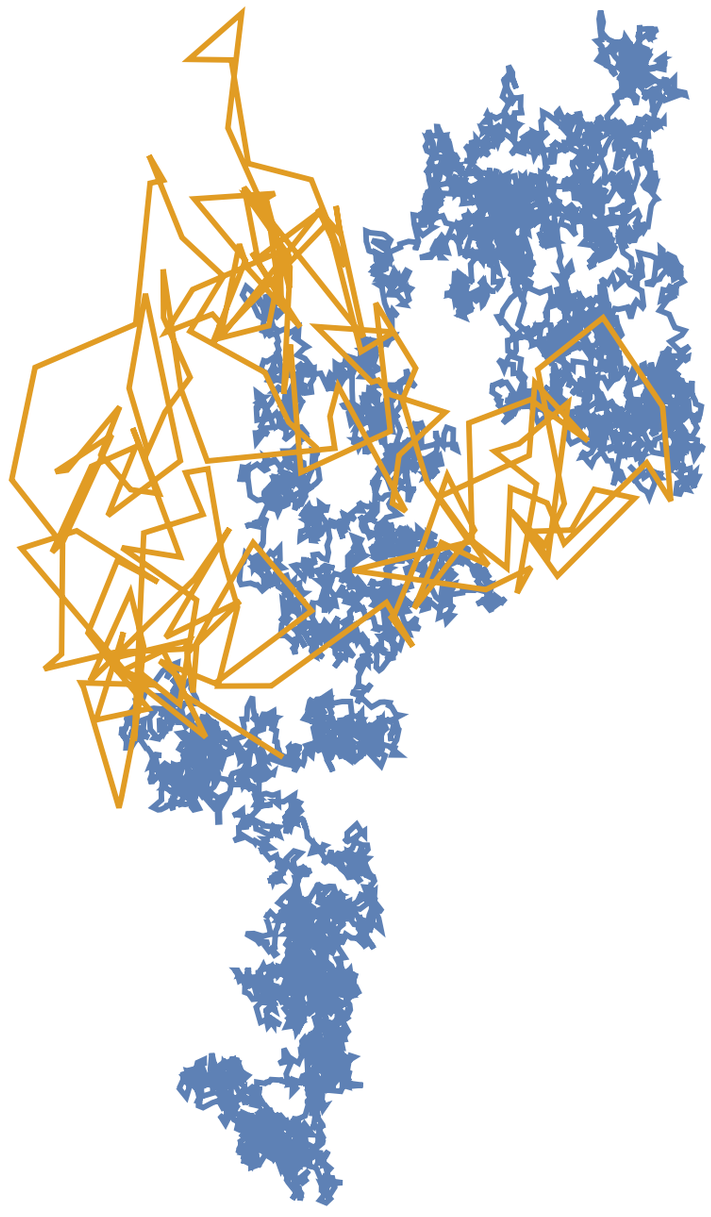
Using diffusions to understand approximation error

- **Diffusion:** continuous-time Markov process with unique stationary distribution



Using diffusions to understand approximation error

- **Diffusion:** continuous-time Markov process with unique stationary distribution



$$b(\theta) = \nabla_\theta \log \pi(\theta \,|\, Y)$$

$$\tilde{b}(\theta) = \nabla_\theta \log \tilde{\pi}(\theta \,|\, Y)$$

Using diffusions to understand approximation error

- **Diffusion:** continuous-time Markov process with unique stationary distribution

$$b(\theta) = \nabla_\theta \log \pi(\theta \,|\, Y)$$

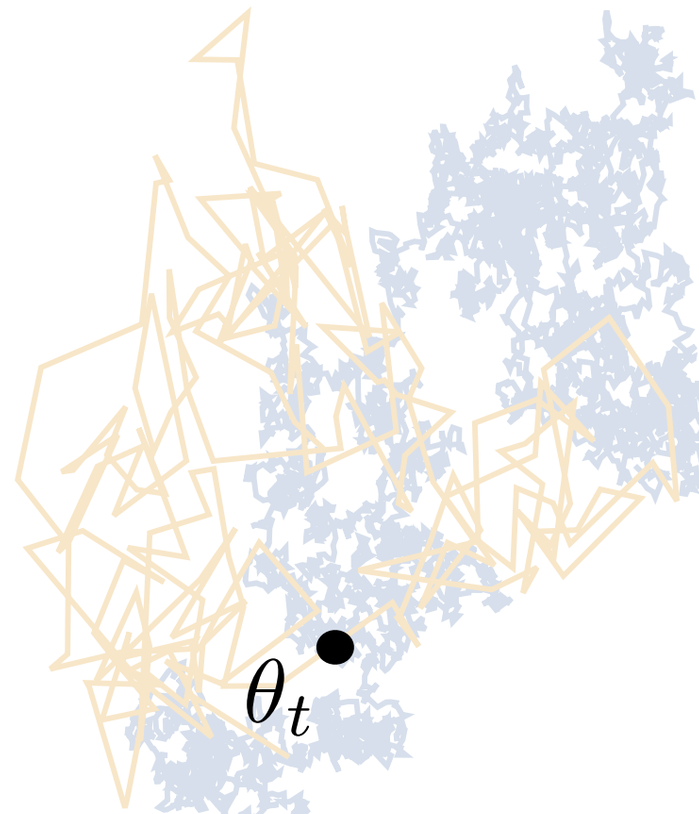
$$\tilde{b}(\theta) = \nabla_\theta \log \tilde{\pi}(\theta \,|\, Y)$$

$$\mathrm{d}\theta_t = b(\theta_t)\mathrm{d}t + \sqrt{2}\,\mathrm{d}W_t$$

$$\mathrm{d}\tilde{\theta}_t = \tilde{b}(\tilde{\theta}_t)\mathrm{d}t + \sqrt{2}\,\mathrm{d}\tilde{W}_t$$

Using diffusions to understand approximation error

- **Diffusion:** continuous-time Markov process with unique stationary distribution



$$b(\theta) = \nabla_\theta \log \pi(\theta \,|\, Y)$$

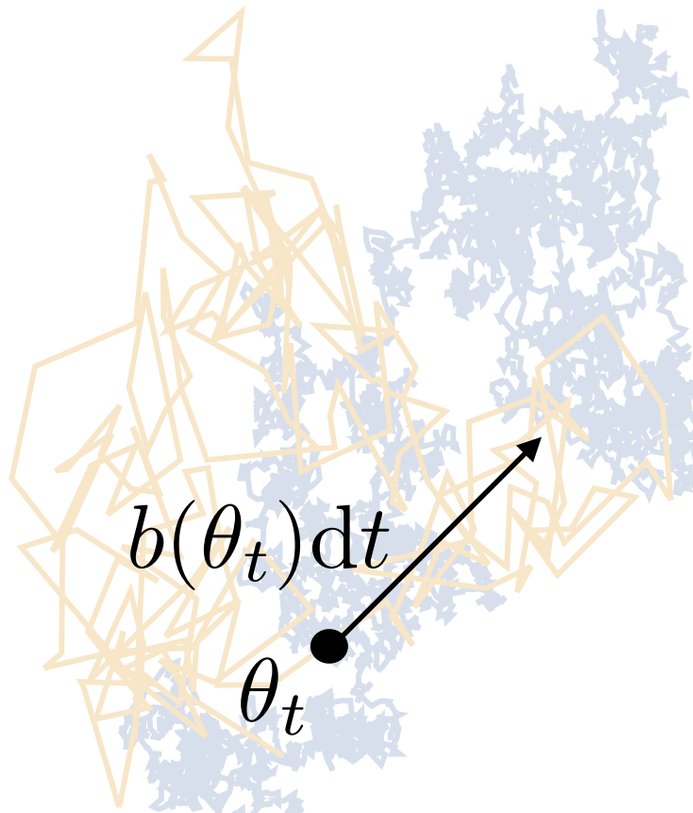
$$\tilde{b}(\theta) = \nabla_\theta \log \tilde{\pi}(\theta \,|\, Y)$$

$$\mathrm{d}\theta_t = b(\theta_t)\mathrm{d}t + \sqrt{2}\,\mathrm{d}W_t$$

$$\mathrm{d}\tilde{\theta}_t = \tilde{b}(\tilde{\theta}_t)\mathrm{d}t + \sqrt{2}\,\mathrm{d}\tilde{W}_t$$

Using diffusions to understand approximation error

- **Diffusion:** continuous-time Markov process with unique stationary distribution



$$b(\theta) = \nabla_\theta \log \pi(\theta \,|\, Y)$$

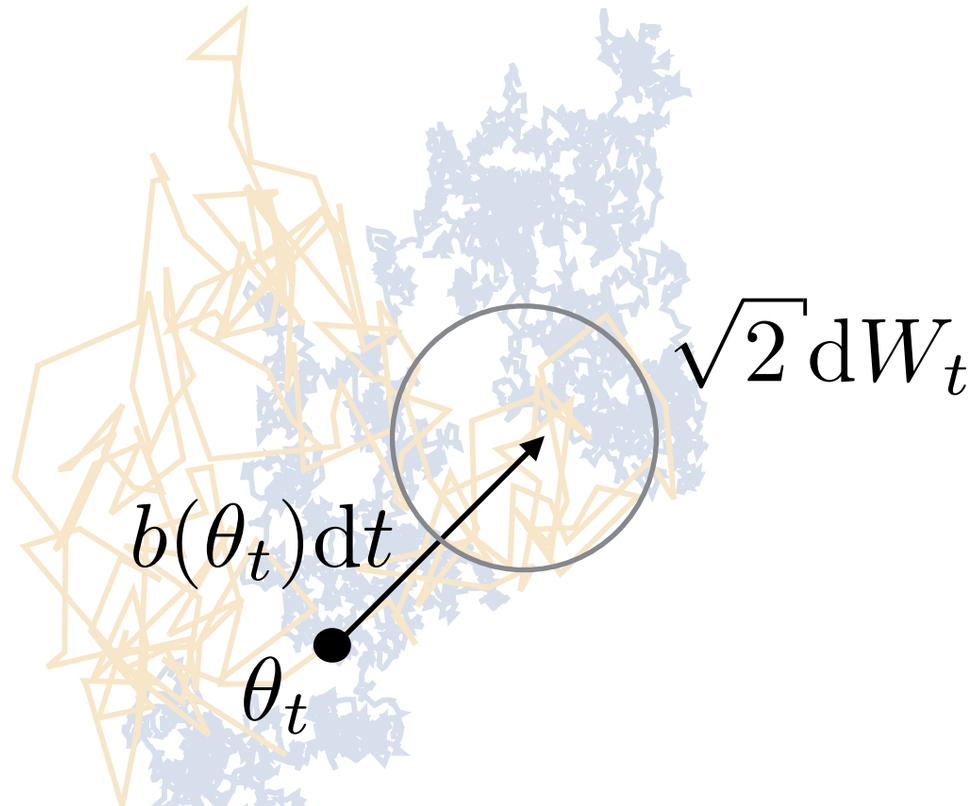
$$\tilde{b}(\theta) = \nabla_\theta \log \tilde{\pi}(\theta \,|\, Y)$$

$$\mathrm{d}\theta_t = b(\theta_t)\mathrm{d}t + \sqrt{2}\,\mathrm{d}W_t$$

$$\mathrm{d}\tilde{\theta}_t = \tilde{b}(\tilde{\theta}_t)\mathrm{d}t + \sqrt{2}\,\mathrm{d}\tilde{W}_t$$

Using diffusions to understand approximation error

- **Diffusion:** continuous-time Markov process with unique stationary distribution



$$b(\theta) = \nabla_\theta \log \pi(\theta \,|\, Y)$$

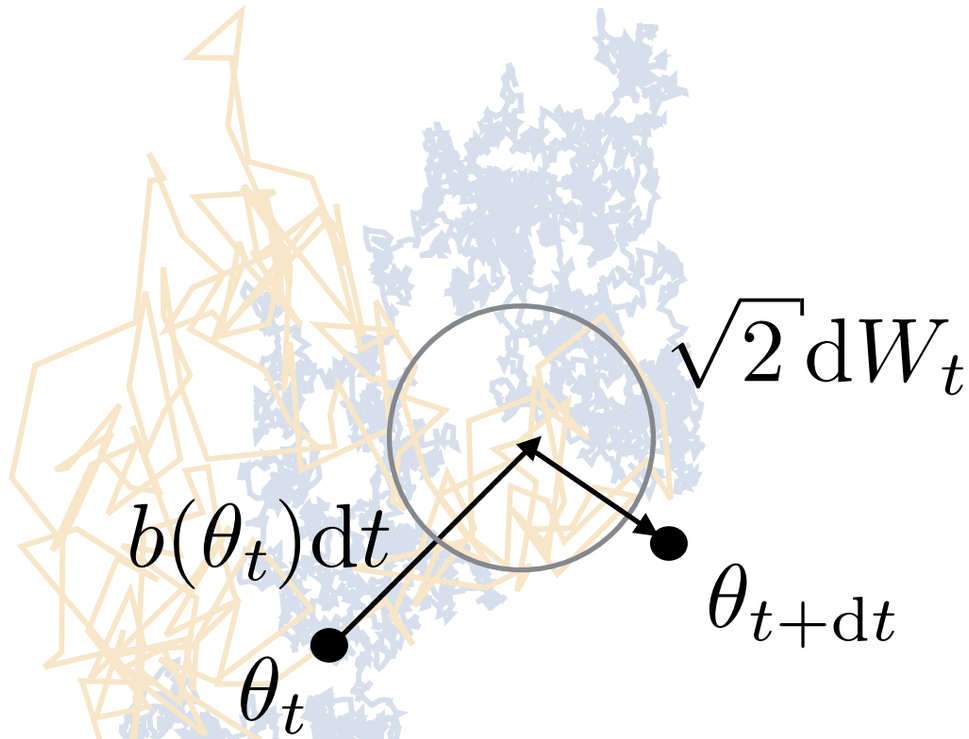
$$\tilde{b}(\theta) = \nabla_\theta \log \tilde{\pi}(\theta \,|\, Y)$$

$$\mathrm{d}\theta_t = b(\theta_t)\mathrm{d}t + \sqrt{2}\,\mathrm{d}W_t$$

$$\mathrm{d}\tilde{\theta}_t = \tilde{b}(\tilde{\theta}_t)\mathrm{d}t + \sqrt{2}\,\mathrm{d}\tilde{W}_t$$

Using diffusions to understand approximation error

- **Diffusion:** continuous-time Markov process with unique stationary distribution



$$b(\theta) = \nabla_\theta \log \pi(\theta \,|\, Y)$$

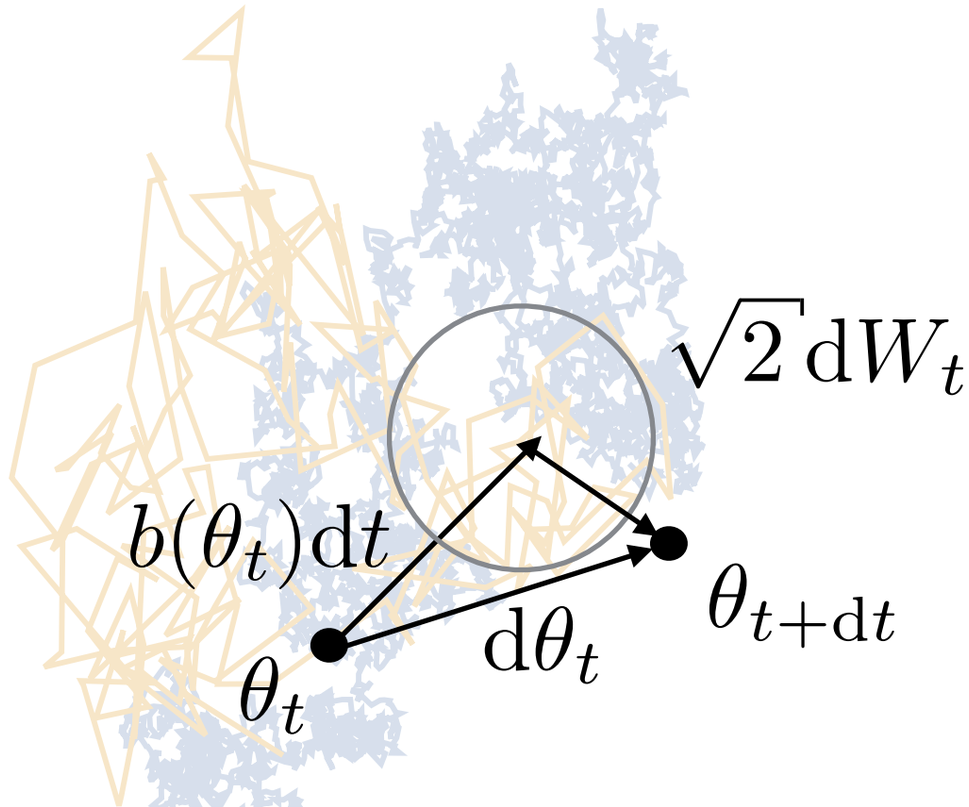
$$\tilde{b}(\theta) = \nabla_\theta \log \tilde{\pi}(\theta \,|\, Y)$$

$$\mathrm{d}\theta_t = b(\theta_t)\mathrm{d}t + \sqrt{2}\,\mathrm{d}W_t$$

$$\mathrm{d}\tilde{\theta}_t = \tilde{b}(\tilde{\theta}_t)\mathrm{d}t + \sqrt{2}\,\mathrm{d}\tilde{W}_t$$

Using diffusions to understand approximation error

- **Diffusion:** continuous-time Markov process with unique stationary distribution



$$b(\theta) = \nabla_\theta \log \pi(\theta \,|\, Y)$$

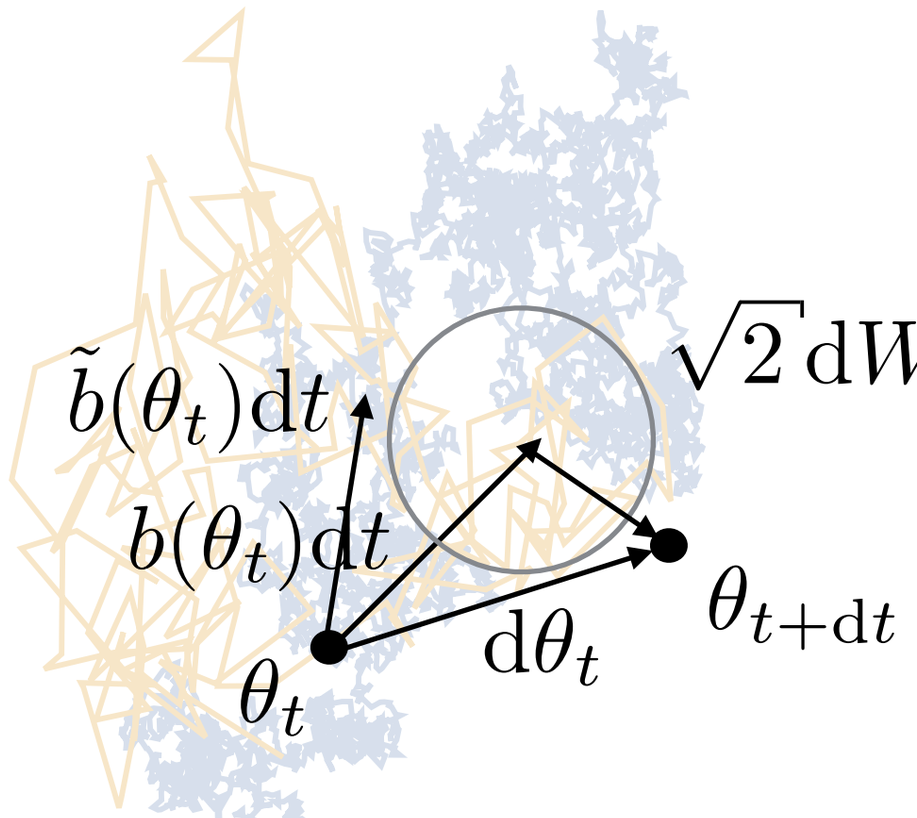
$$\tilde{b}(\theta) = \nabla_\theta \log \tilde{\pi}(\theta \,|\, Y)$$

$$\mathrm{d}\theta_t = b(\theta_t)\mathrm{d}t + \sqrt{2}\,\mathrm{d}W_t$$

$$\mathrm{d}\tilde{\theta}_t = \tilde{b}(\tilde{\theta}_t)\mathrm{d}t + \sqrt{2}\,\mathrm{d}\tilde{W}_t$$

Using diffusions to understand approximation error

- **Diffusion:** continuous-time Markov process with unique stationary distribution



$$b(\theta) = \nabla_\theta \log \pi(\theta \,|\, Y)$$

$$\tilde{b}(\theta) = \nabla_\theta \log \tilde{\pi}(\theta \,|\, Y)$$

$$\mathrm{d}\theta_t = b(\theta_t)\mathrm{d}t + \sqrt{2}\,\mathrm{d}W_t$$

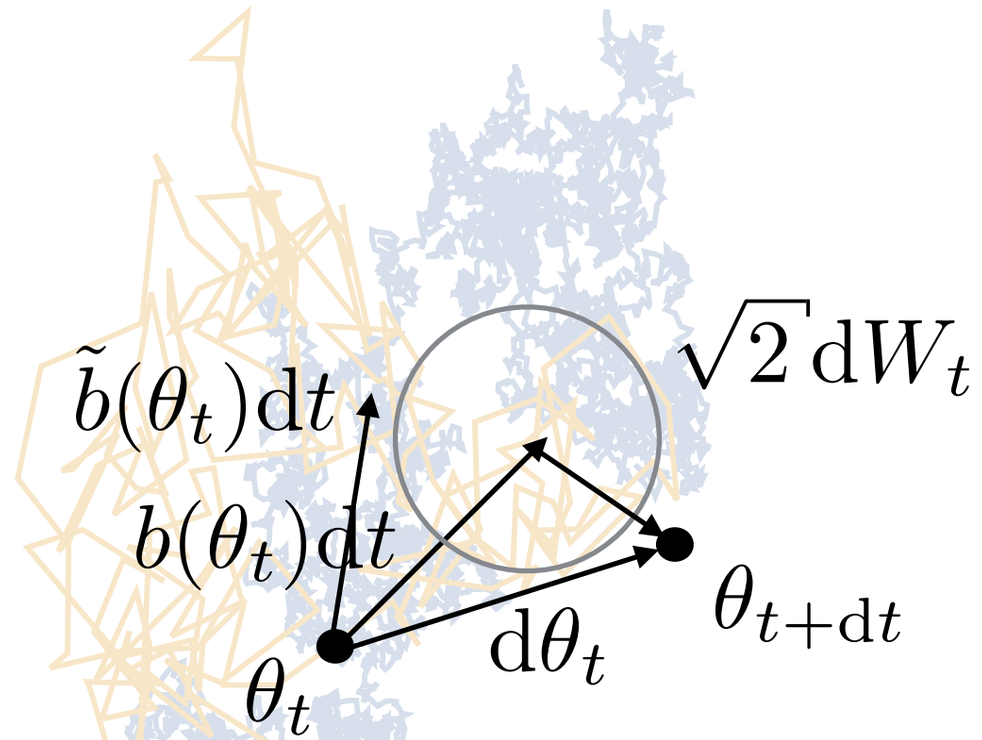
$$\mathrm{d}\tilde{\theta}_t = \tilde{b}(\tilde{\theta}_t)\mathrm{d}t + \sqrt{2}\,\mathrm{d}\tilde{W}_t$$

Using diffusions to understand approximation error

- **Diffusion:** continuous-time Markov process with unique stationary distribution
- **Intuition:** if diffusion mixes quickly, then gradient errors don't have time to build up

$$b(\theta) = \nabla_\theta \log \pi(\theta \,|\, Y)$$

$$\tilde{b}(\theta) = \nabla_\theta \log \tilde{\pi}(\theta \,|\, Y)$$



$$\mathrm{d}\theta_t = b(\theta_t)\mathrm{d}t + \sqrt{2}\,\mathrm{d}W_t$$

$$\mathrm{d}\tilde{\theta}_t = \tilde{b}(\tilde{\theta}_t)\mathrm{d}t + \sqrt{2}\,\mathrm{d}\tilde{W}_t$$

Wasserstein distance bounds from diffusions

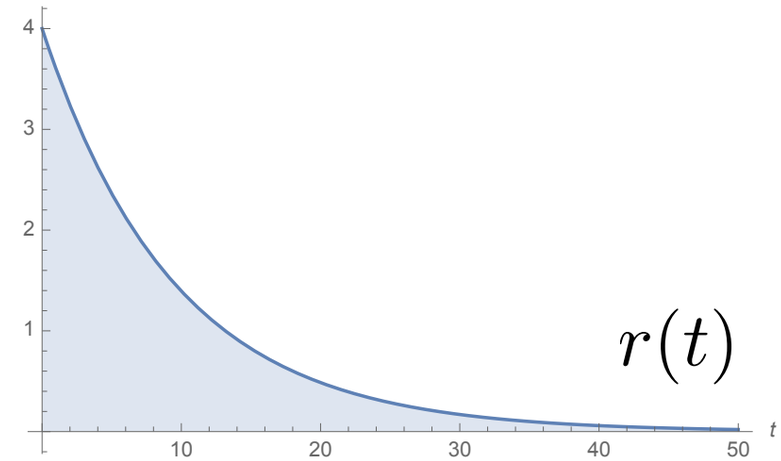
Wasserstein distance bounds from diffusions

Theorem (H. & Zou 2017, H. 2018).

Wasserstein distance bounds from diffusions

Theorem (H. & Zou 2017, H. 2018).

Assume the diffusion converges at rate $r(t)$.



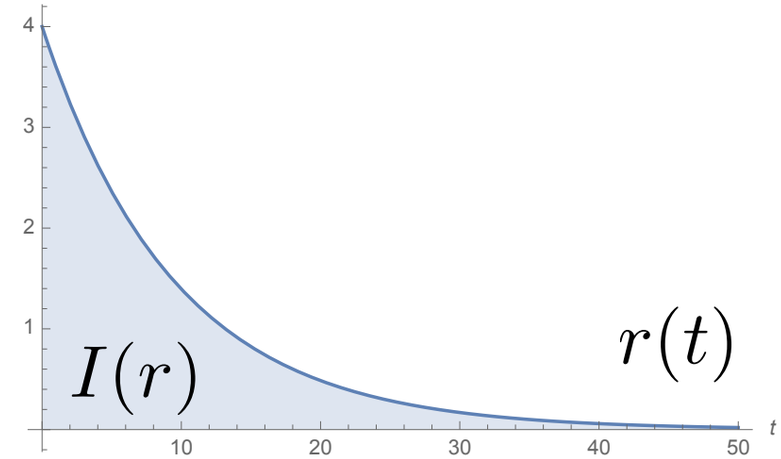
$$\mathrm{d}\theta_t = b(\theta_t)\mathrm{d}t + \sqrt{2}\,\mathrm{d}W_t$$

Wasserstein distance bounds from diffusions

Theorem (H. & Zou 2017, H. 2018).

Assume the diffusion converges at rate $r(t)$.

Let $I(r) = \int r(t)\,\mathrm{d}t$.



$$\mathrm{d}\theta_t = b(\theta_t)\mathrm{d}t + \sqrt{2}\,\mathrm{d}W_t$$

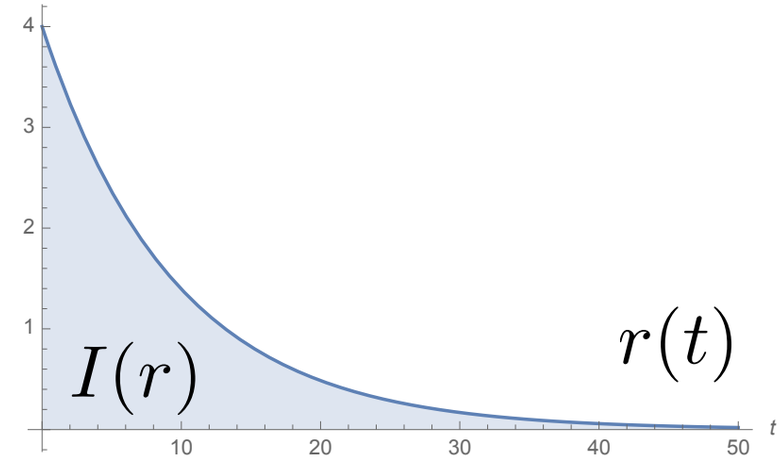
Wasserstein distance bounds from diffusions

Theorem (H. & Zou 2017, H. 2018).

Assume the diffusion converges at rate $r(t)$.

Let $I(r) = \int r(t)\,\mathrm{d}t$.

Then $d_W(\pi, \tilde{\pi}) \leq I(r)\,\mathbb{E}_{\tilde{\pi}}[\|b - \tilde{b}\|_2]$.



$$\mathrm{d}\theta_t = b(\theta_t)\mathrm{d}t + \sqrt{2}\,\mathrm{d}W_t$$

$$b(\theta) = \nabla_\theta \log \pi(\theta \mid Y)$$

$$\tilde{b}(\theta) = \nabla_\theta \log \tilde{\pi}(\theta \mid Y)$$

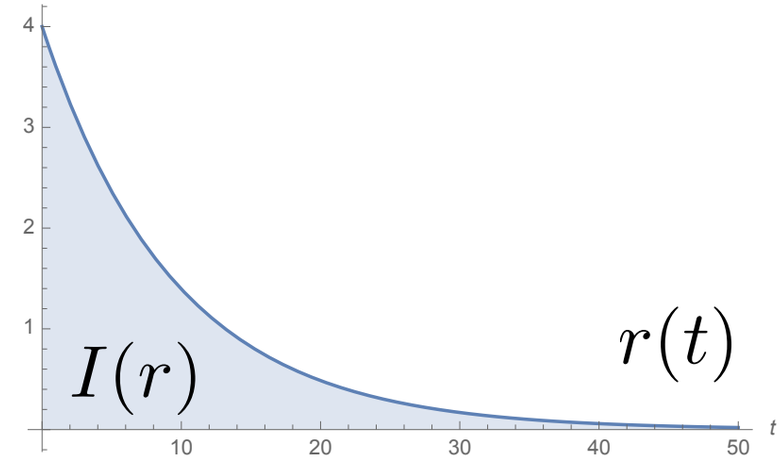
Wasserstein distance bounds from diffusions

Theorem (H. & Zou 2017, H. 2018).

Assume the diffusion converges at rate $r(t)$.

Let $I(r) = \int r(t)\,\mathrm{d}t$.

Then $d_W(\pi, \tilde{\pi}) \leq \underbrace{I(r)}_{c_\pi} \mathbb{E}_{\tilde{\pi}}[\|b - \tilde{b}\|_2]$.



$$\mathrm{d}\theta_t = b(\theta_t)\mathrm{d}t + \sqrt{2}\,\mathrm{d}W_t$$

$$b(\theta) = \nabla_\theta \log \pi(\theta \mid Y)$$

$$\tilde{b}(\theta) = \nabla_\theta \log \tilde{\pi}(\theta \mid Y)$$

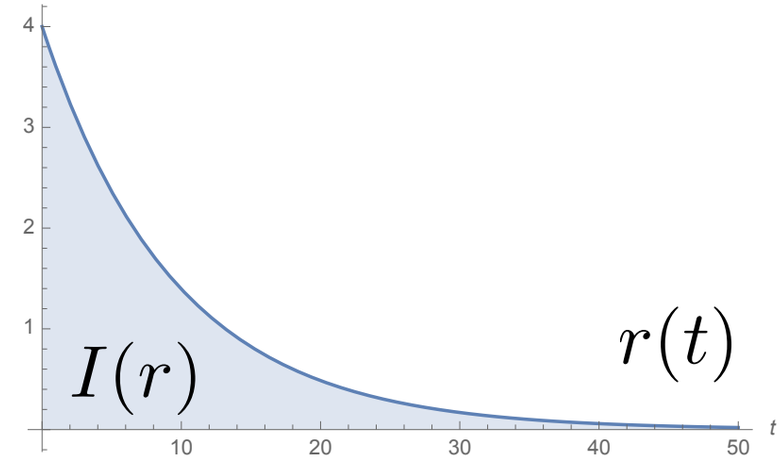
Wasserstein distance bounds from diffusions

Theorem (H. & Zou 2017, H. 2018).

Assume the diffusion converges at rate $r(t)$.

Let $I(r) = \int r(t)\,\mathrm{d}t$.

Then $d_W(\pi, \tilde{\pi}) \leq \underbrace{I(r)}_{c_\pi} \underbrace{\mathbb{E}_{\tilde{\pi}}[\|b - \tilde{b}\|_2]}_{\varepsilon}$.



$$\mathrm{d}\theta_t = b(\theta_t)\mathrm{d}t + \sqrt{2}\,\mathrm{d}W_t$$

$$b(\theta) = \nabla_\theta \log \pi(\theta \mid Y)$$

$$\tilde{b}(\theta) = \nabla_\theta \log \tilde{\pi}(\theta \mid Y)$$

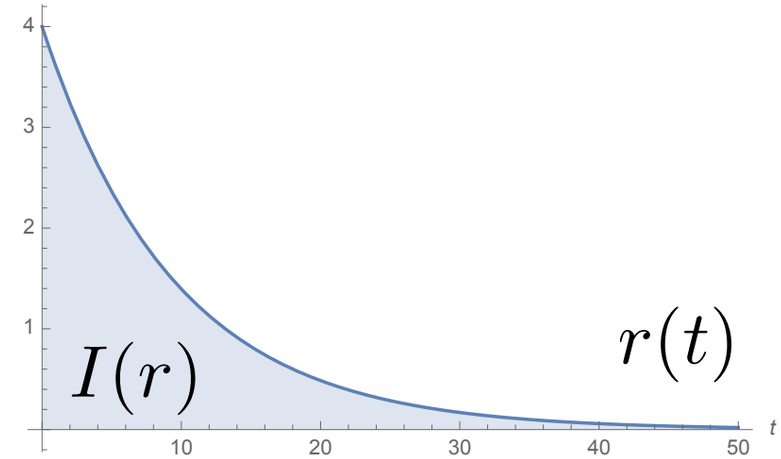
Wasserstein distance bounds from diffusions

Theorem (H. & Zou 2017, H. 2018).

Assume the diffusion converges at rate $r(t)$.

Let $I(r) = \int r(t)\,\mathrm{d}t$.

Then $d_W(\pi, \tilde{\pi}) \leq \underbrace{I(r)}_{c_\pi} \underbrace{\mathbb{E}_{\tilde{\pi}}[\|b - \tilde{b}\|_2]}_{\varepsilon}$.



$$\mathrm{d}\theta_t = b(\theta_t)\mathrm{d}t + \sqrt{2}\,\mathrm{d}W_t$$

$$b(\theta) = \nabla_\theta \log \pi(\theta \mid Y)$$

$$\tilde{b}(\theta) = \nabla_\theta \log \tilde{\pi}(\theta \mid Y)$$

- Proof techniques:

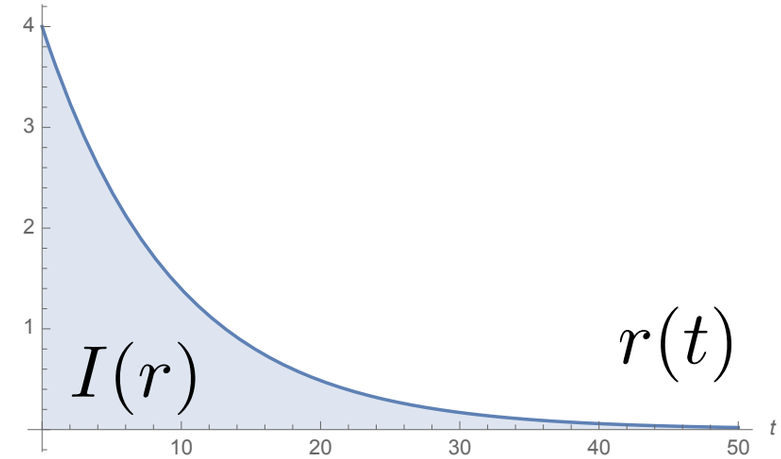
Wasserstein distance bounds from diffusions

Theorem (H. & Zou 2017, H. 2018).

Assume the diffusion converges at rate $r(t)$.

Let $I(r) = \int r(t)\,\mathrm{d}t$.

Then $d_W(\pi, \tilde{\pi}) \leq \underbrace{I(r)}_{c_\pi} \underbrace{\mathbb{E}_{\tilde{\pi}}[\|b - \tilde{b}\|_2]}_{\varepsilon}$.



$$\mathrm{d}\theta_t = b(\theta_t)\mathrm{d}t + \sqrt{2}\,\mathrm{d}W_t$$

$$b(\theta) = \nabla_\theta \log \pi(\theta \mid Y)$$

$$\tilde{b}(\theta) = \nabla_\theta \log \tilde{\pi}(\theta \mid Y)$$

- Proof techniques:
  - Stein's method (for 1-Wasserstein version)

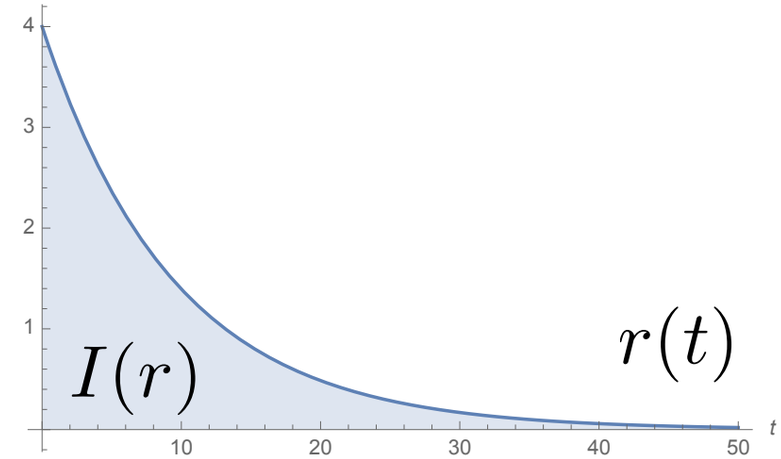
Wasserstein distance bounds from diffusions

Theorem (H. & Zou 2017, H. 2018).

Assume the diffusion converges at rate $r(t)$.

Let $I(r) = \int r(t)\,\mathrm{d}t$.

Then $d_W(\pi, \tilde{\pi}) \leq \underbrace{I(r)}_{c_\pi} \underbrace{\mathbb{E}_{\tilde{\pi}}[\|b - \tilde{b}\|_2]}_{\varepsilon}$.



$$\mathrm{d}\theta_t = b(\theta_t)\mathrm{d}t + \sqrt{2}\,\mathrm{d}W_t$$

$$b(\theta) = \nabla_\theta \log \pi(\theta \mid Y)$$

$$\tilde{b}(\theta) = \nabla_\theta \log \tilde{\pi}(\theta \mid Y)$$

- Proof techniques:
  - Stein's method (for 1-Wasserstein version)
  - A coupling argument + Ito's lemma (for 2-Wasserstein version)