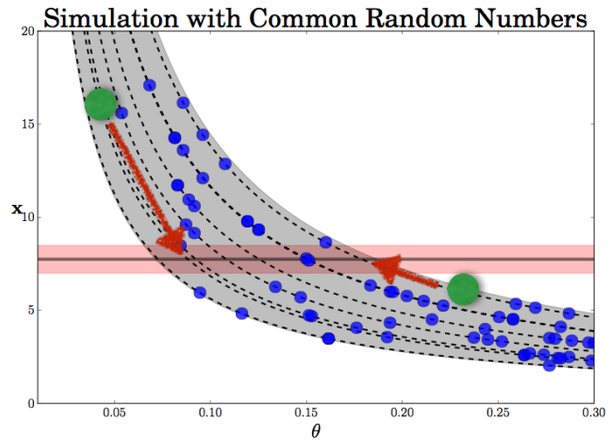
Optimization Monte Carlo



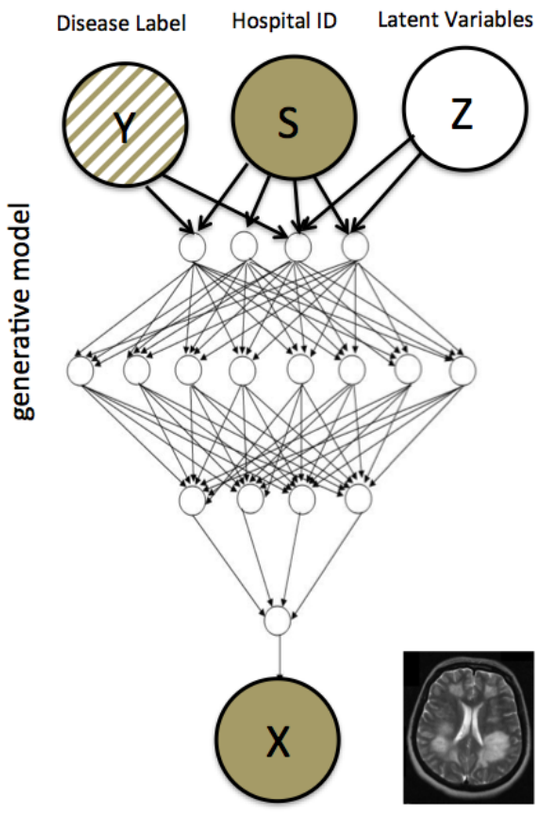
Max Welling

joint work with:

Ted Meeds

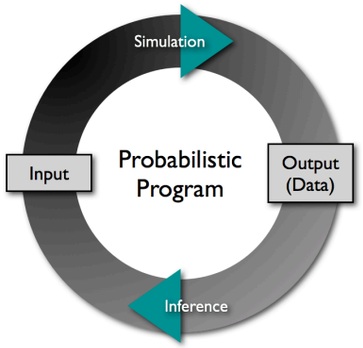
University of Amsterdam

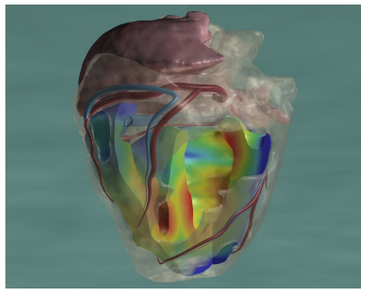
What Model Should We Consider?

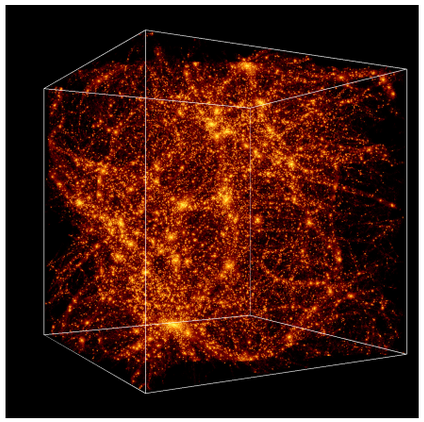


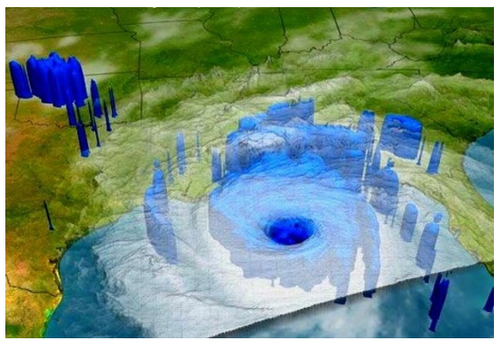
A machine learner's view of a model

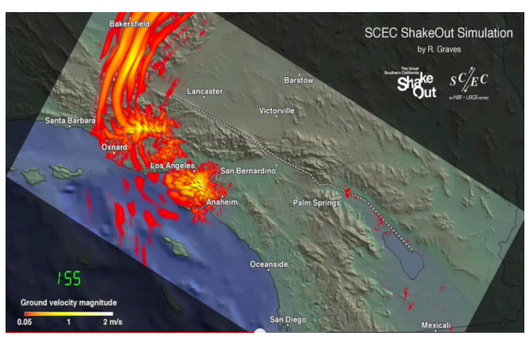
A astronomers view of a model







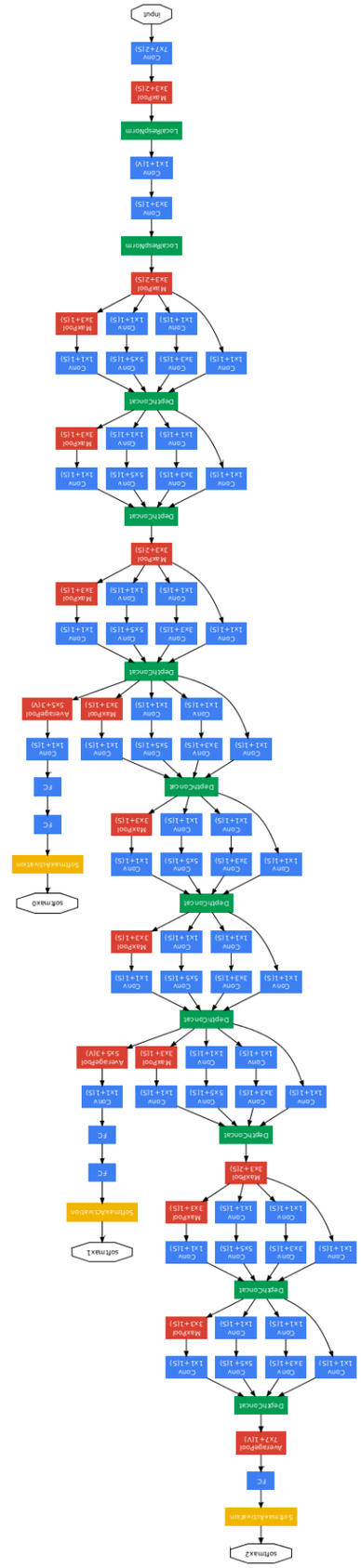




Probabilistic Generative Models

VS

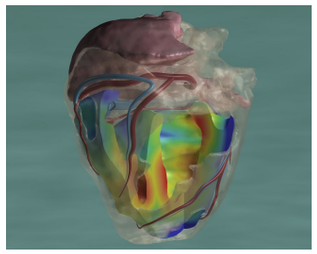
Deep Learning



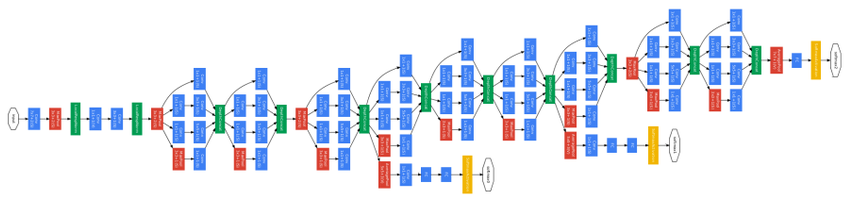
Slow vs Fast: Small N vs. Big N?

We need *statistical efficiency*

N = 100-1000



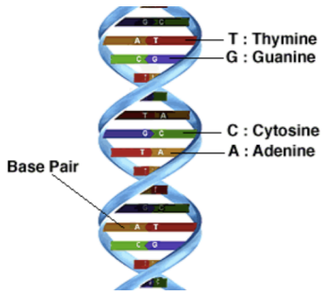
We need *computational efficiency*

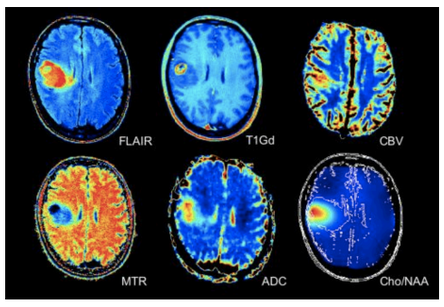


=10^8-10^9

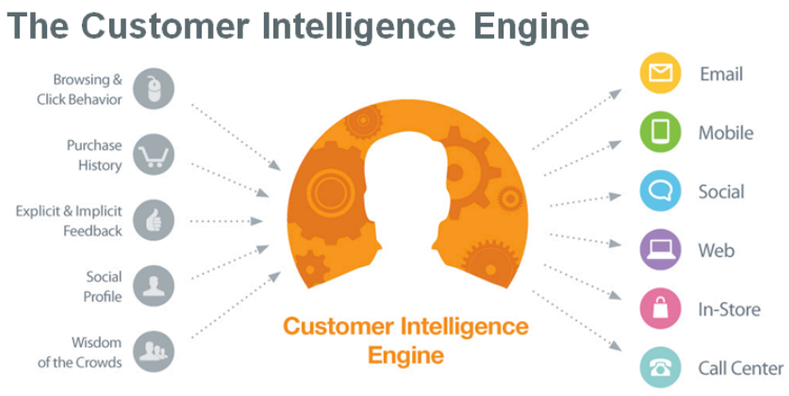
-Healthcare (p>>N)

3 Billion ($3x10^9$) base pairs (DNA)

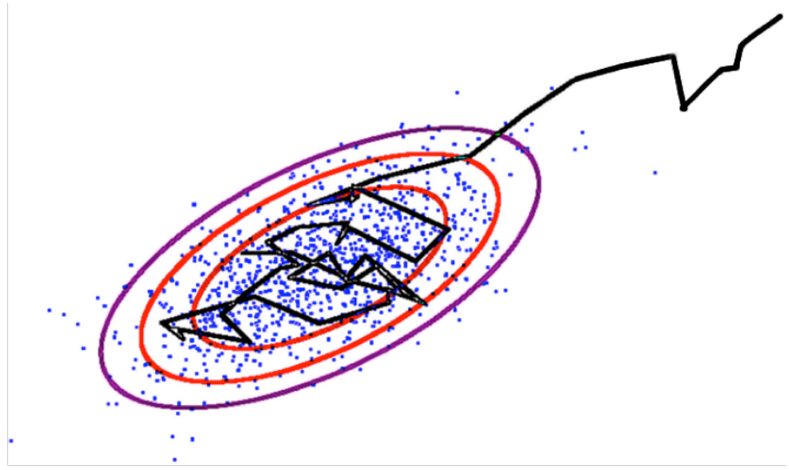




-Customer Intelligence
-Finance
-Video/image
-Internet of Things



What makes MCMC Inefficient?



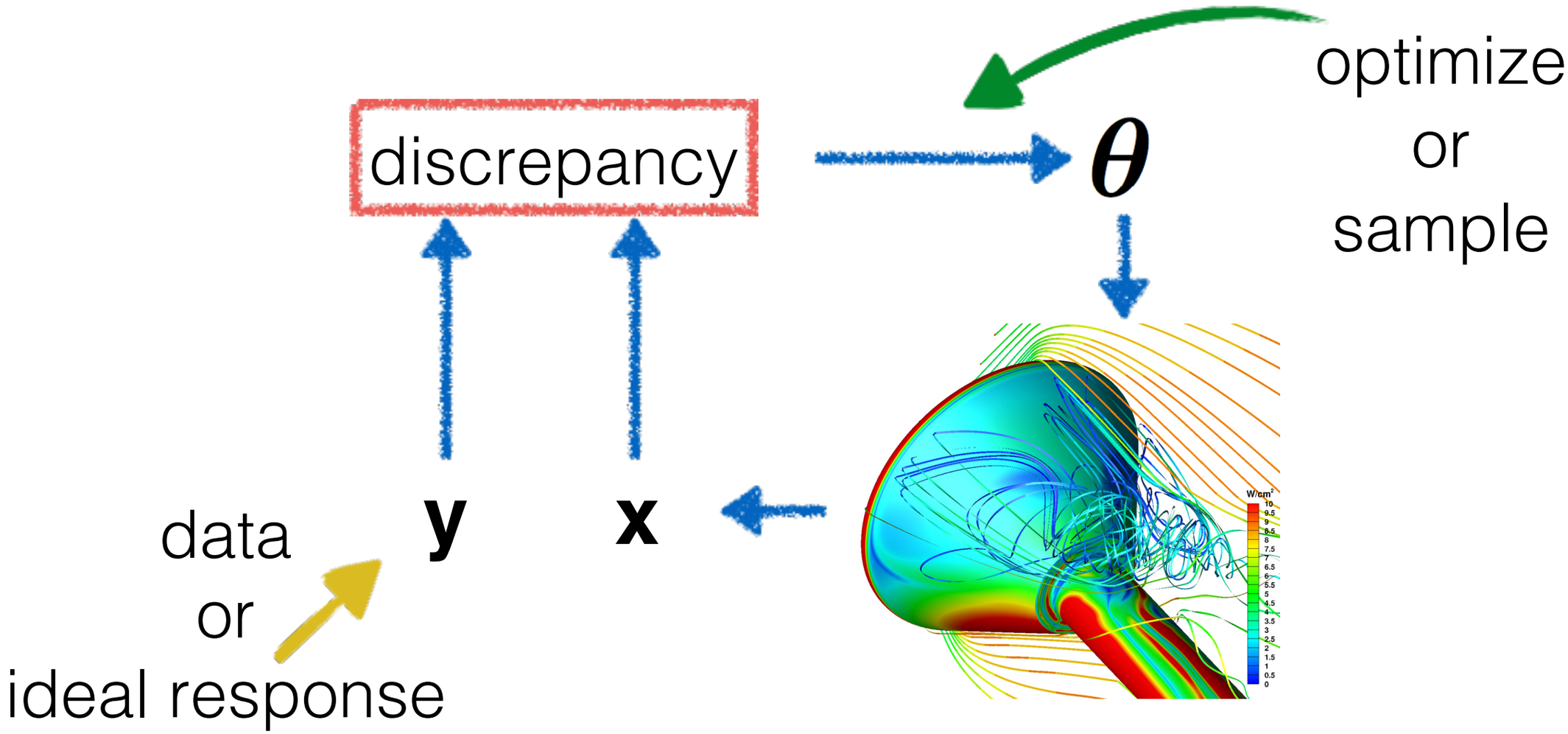
- Detailed balance
  → Optimization

- Randomness
  → Low discrepance sequence
  →Herding/QMC/Bayesian quadrature

- Sequential algorithm
  → Distributed algorithm

ABC

- Given: observational data **y**
- Inputs: parameter vector $\boldsymbol{\theta}$
- Outputs: pseudo-data **x**



Approximate Bayesian Computation

- Primary goal is Bayesian inference:

prior likelihood

posterior $\pi(\boldsymbol{\theta}|\mathbf{y}) = \dfrac{\pi(\boldsymbol{\theta})\pi(\mathbf{y}|\boldsymbol{\theta})}{\int \pi(\boldsymbol{\theta})\pi(\mathbf{y}|\boldsymbol{\theta})d\boldsymbol{\theta}}$

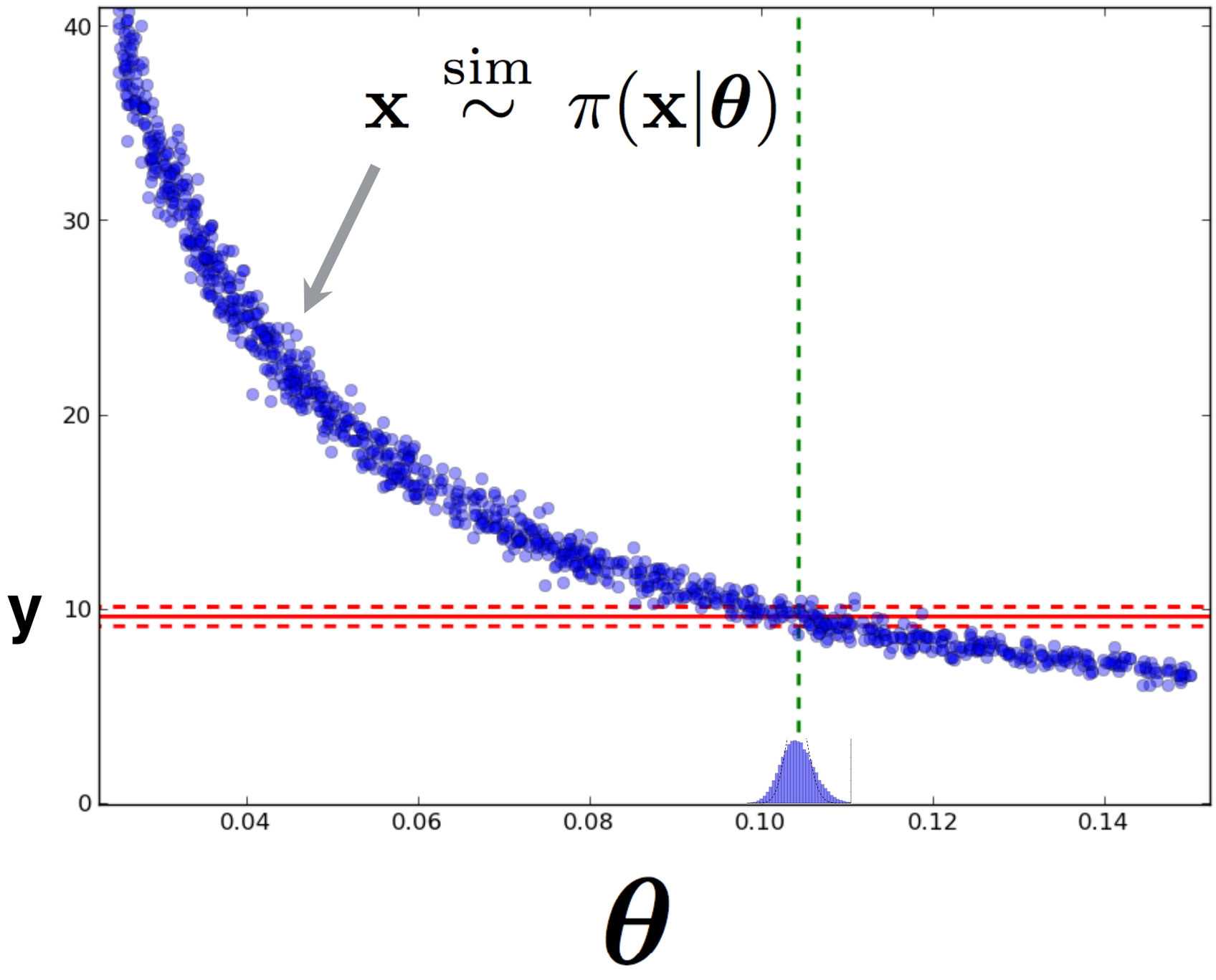
- No likelihood, only simulator.
- Discrepancy between statistics from pseudo-data and observations proxy for likelihood.

$$\pi_\epsilon(\boldsymbol{\theta}|\mathbf{y}) = \frac{\pi(\boldsymbol{\theta})}{\pi(\mathbf{y})} \int \pi_\epsilon(\mathbf{y}|\mathbf{x})\pi(\mathbf{x}|\boldsymbol{\theta})d\mathbf{x}$$

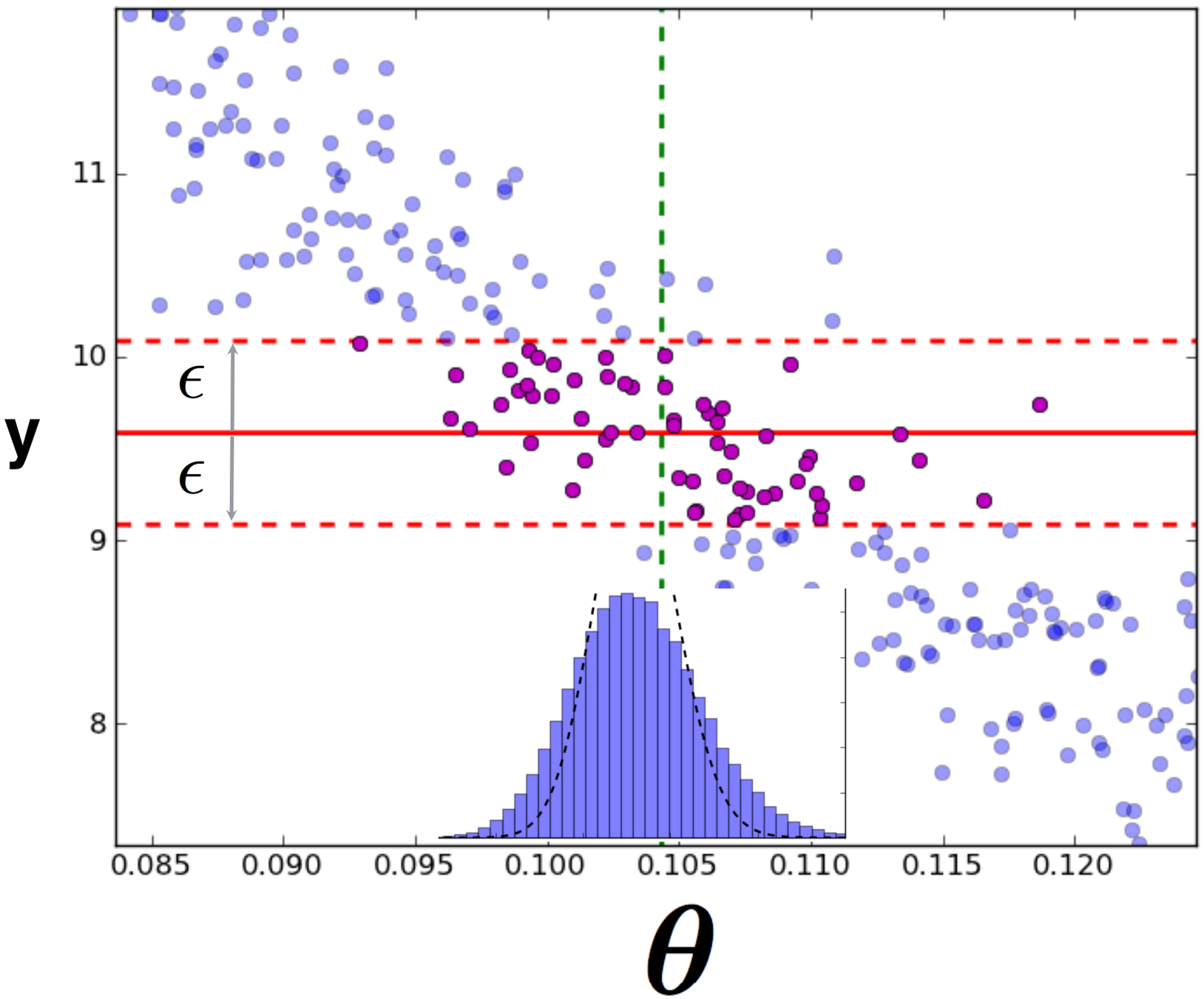
discrepancy measure

simulator $\mathbf{x} \overset{\text{sim}}{\sim} \pi(\mathbf{x}|\boldsymbol{\theta})$

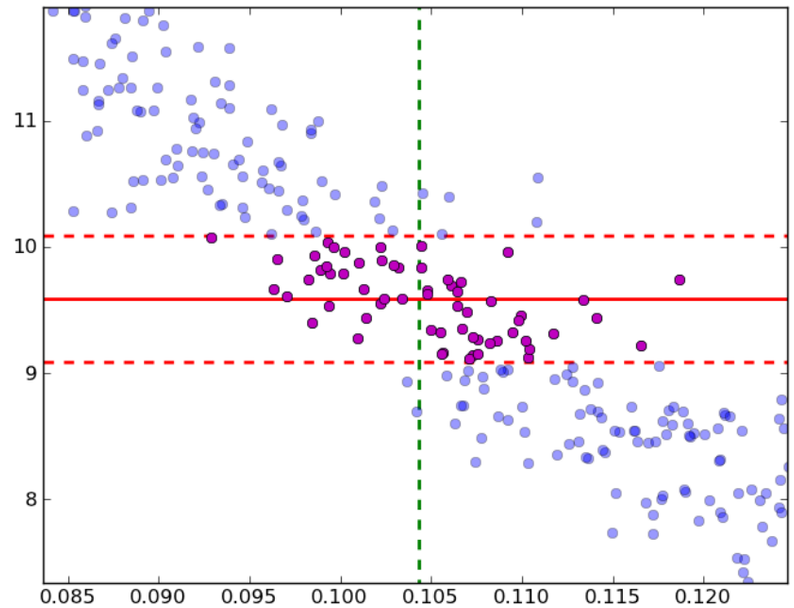
Example

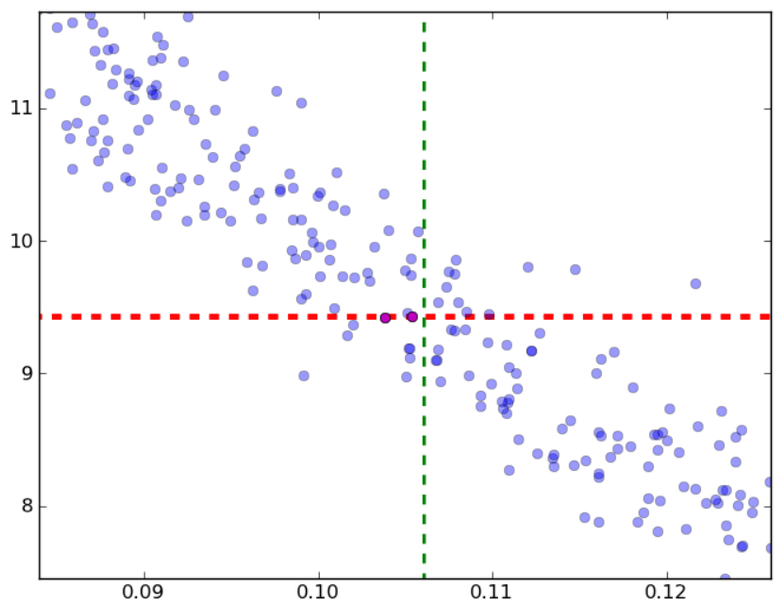

$\mathbf{x} \overset{\text{sim}}{\sim} \pi(\mathbf{x}|\boldsymbol{\theta})$
40
30
20
10
0
y
0.04
0.06
0.08
0.10
0.12
0.14
$\boldsymbol{\theta}$

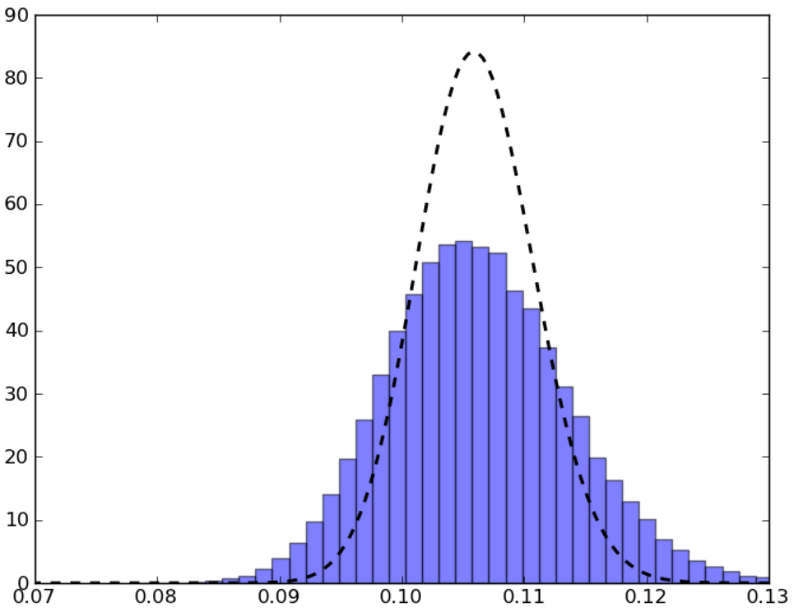
Example

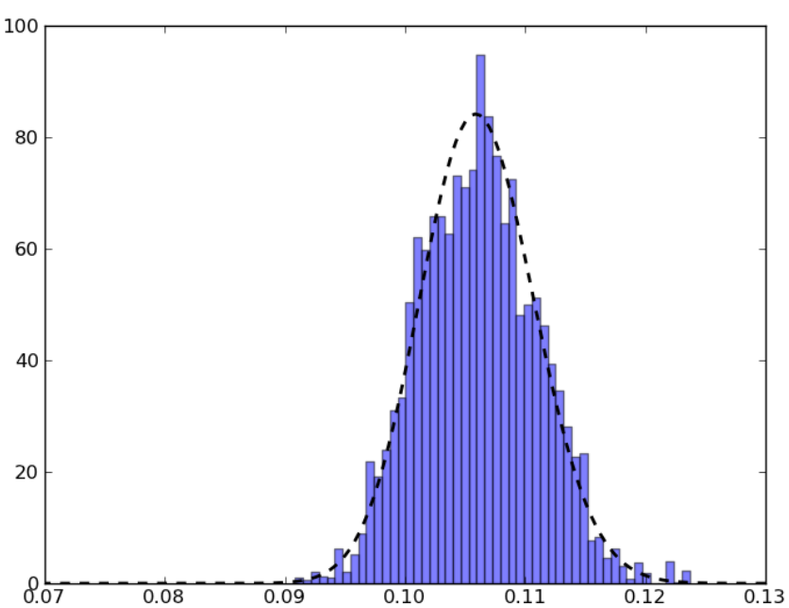

11
10
9
8
y
ϵ
ϵ
0.085
0.090
0.095
0.100
0.105
0.110
0.115
0.120
θ

Effect of Epsilon









rejection rate = 72%

rejection rate = 99.4%

A Alternative View of Inference

- Assume we have a generative model & prior: $P(X|\theta)$,$P(\theta)$
- We want to sample from:

$$P(\theta|X=y) \propto P(X=y|\theta)P(\theta) = \int \delta(y-x)P(X=x|\theta)P(\theta)\ dx$$

- Apply the "reparametrization trick":

$$= \int\int \delta(y-x)\delta(x-f(\theta,u))P(u)P(\theta)\ dx\ du$$

- Integrate out x:

$$= \int \delta(y-f(\theta,u))P(u)P(\theta)\ du$$

Alternative View Continued

- From last slide: $= \int \delta(y - f(\theta, u))P(u)P(\theta)\ du$

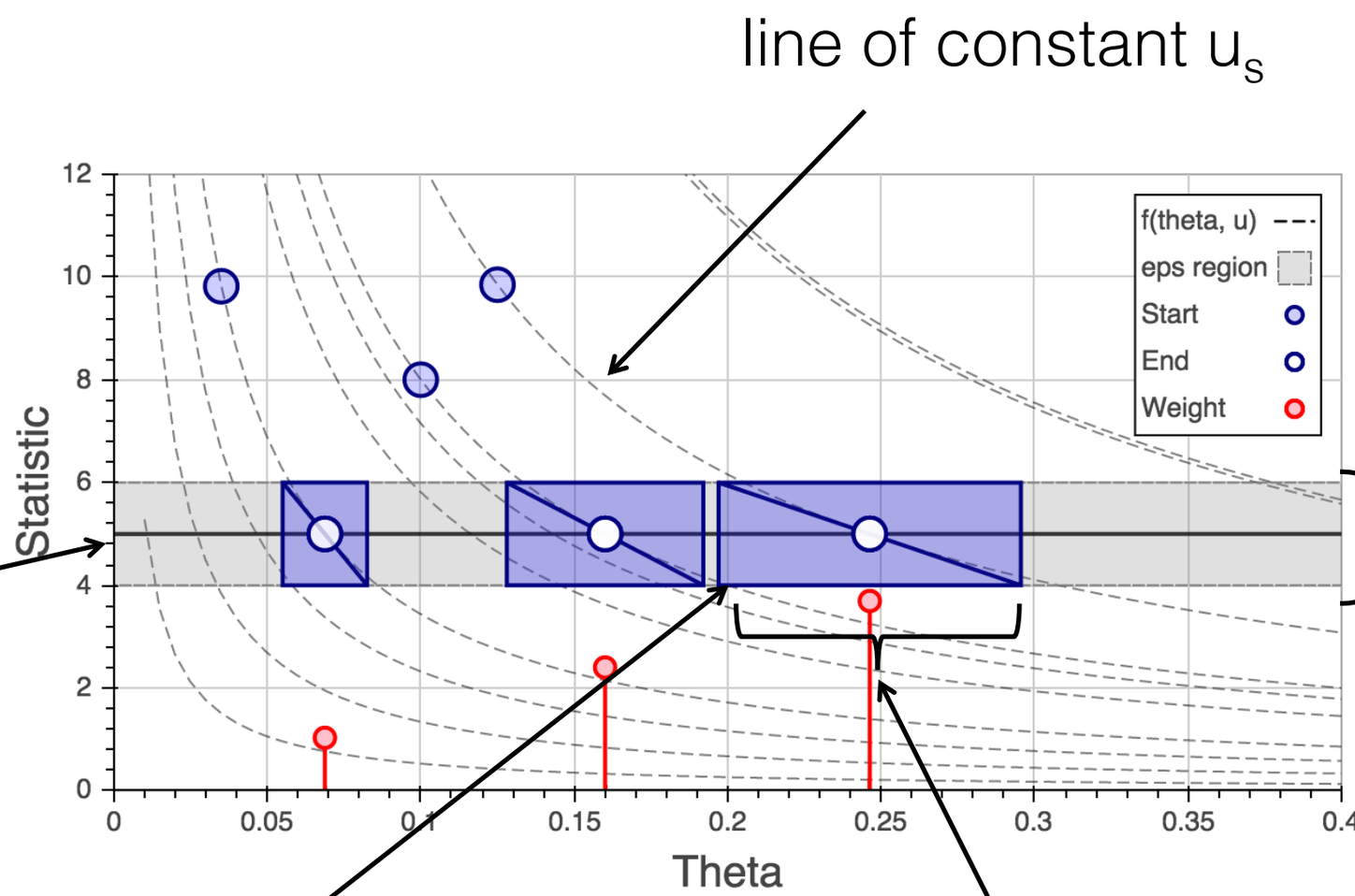
- Monte Carlo estimate: $= \frac{1}{S}\sum_{s=1}^{S} \delta(y - f(\theta, u_s))P(\theta)$

- Widen to epsilon tube: $= \frac{1}{S}\sum_{s=1}^{S} \delta_\epsilon(y - f(\theta, u_s))P(\theta)$

- Transform to density in theta: $= \frac{1}{Z}\sum_{s=1}^{S} W_s \delta_\epsilon(\theta - \theta_s^*)P(\theta_s^*)$

  with: $W_s = \frac{1}{\sqrt{J_s(\theta_s^*)^T J_s(\theta_s^*)}}$ and $f(\theta_s^*, u_s) = [y - \epsilon, y + \epsilon]$

Some Intuition



Box with parameter values that satisfy:

$$||y - f(\theta_s^*, u_s)|| \leq \epsilon$$

$$|J(\theta_s^*, u_s)| = \left| \frac{df(\theta, u_s)}{d\theta} \Big|_{\theta=\theta_s^*} \right|$$

Optimization Monte Carlo

REPEAT for s=1:S

- For observation y, sample u_s~p(u)

- Minimize: $\theta_s^* = \arg\min_\theta ||y - f(\theta, u_s)||$

- Accept when: $||y - f(\theta_s^*, u_s)|| \leq \epsilon$

- Compute weight: $W_s = \frac{1}{\sqrt{J_s(\theta_s^*)^T J_s(\theta_s^*)}}$

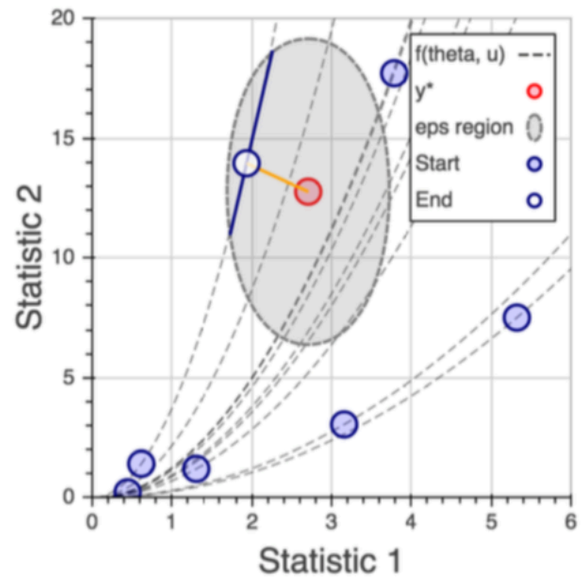
END

- Compute Monte Carlo approximation:

$$E_{p(\theta|y)}[g(\theta)] \approx \frac{1}{Z} \sum_{s=1}^{S} W_s \; P(\theta_s^*) \; g(\theta_s^*)$$

Observations about OMC

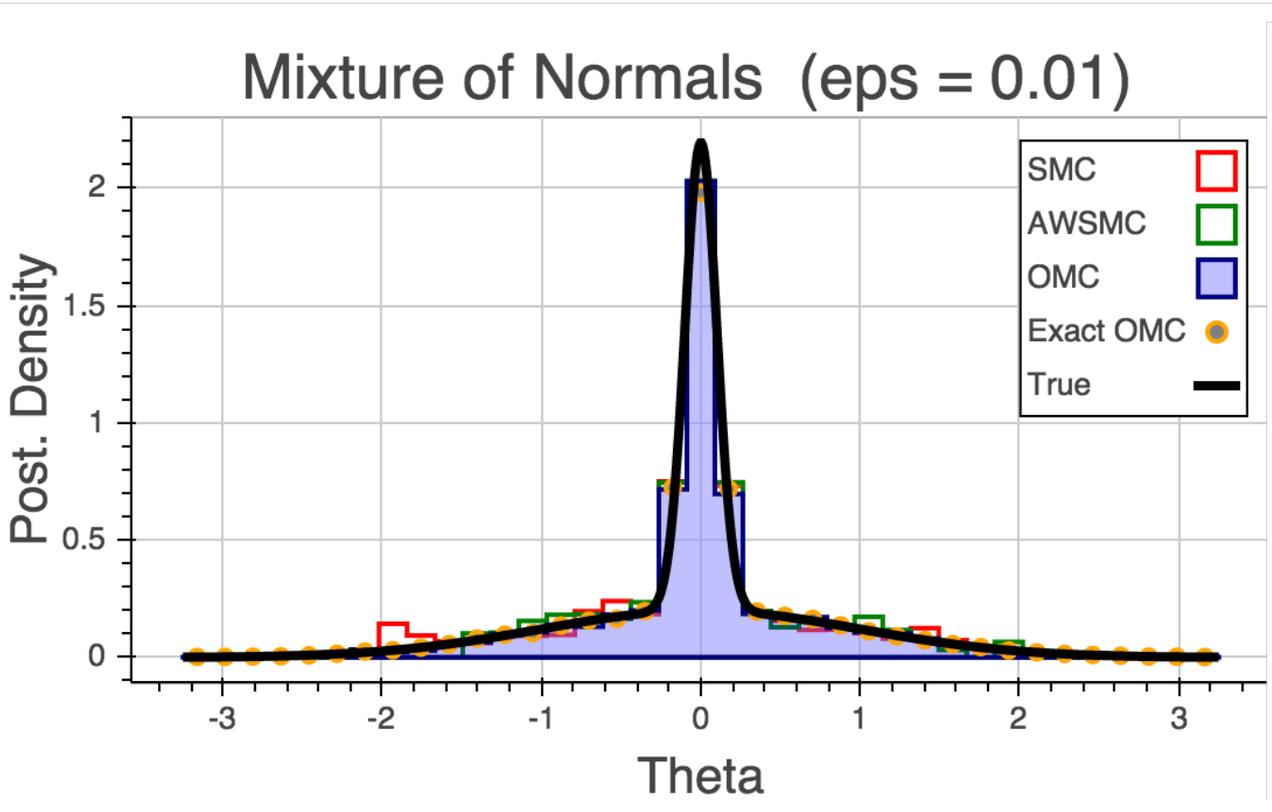
- It requires: $Dim(Y) \geq Dim(\theta)$
- It's embarrassingly parallel

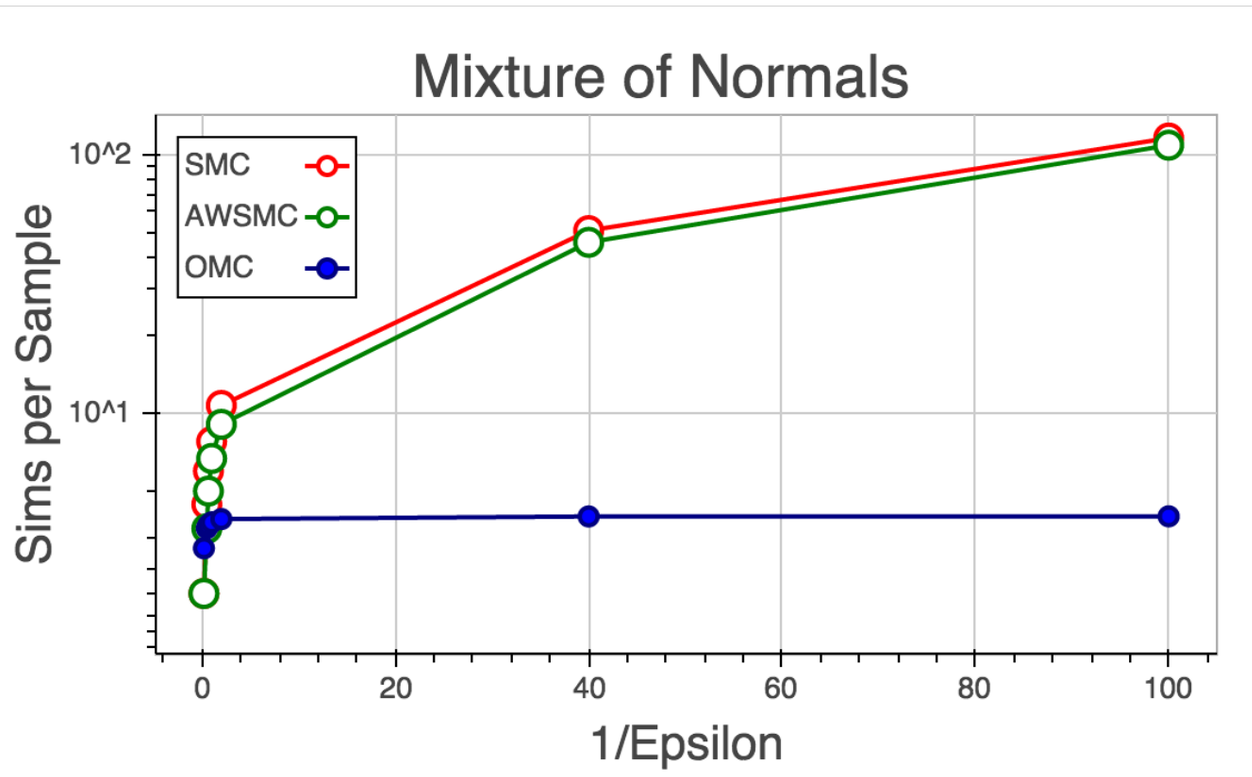


- It's based on optimization (no detailed balance etc.)
- It's random: $u_s \sim p(u)$. → low discrepance sequence?
- It's approximate because we accept close to y.
- It's anytime: we can emit the particles inside a range [y-R,y+R] and incur an approximation O(R).
- It's a natural fit to ABC problems (intractable likelihood)

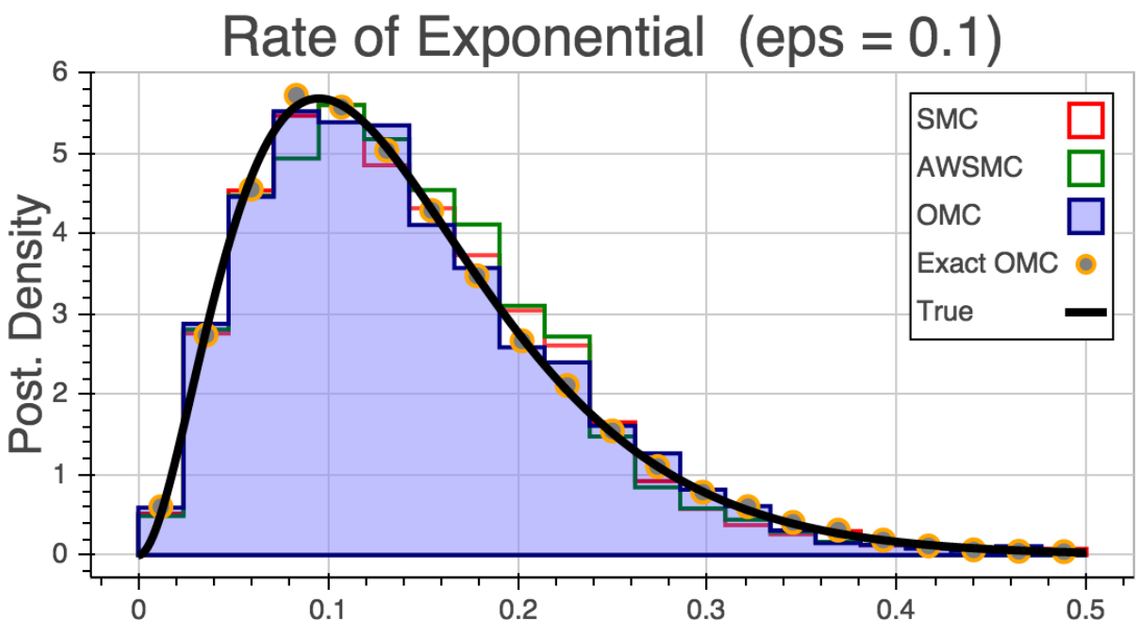
OMC Experiments

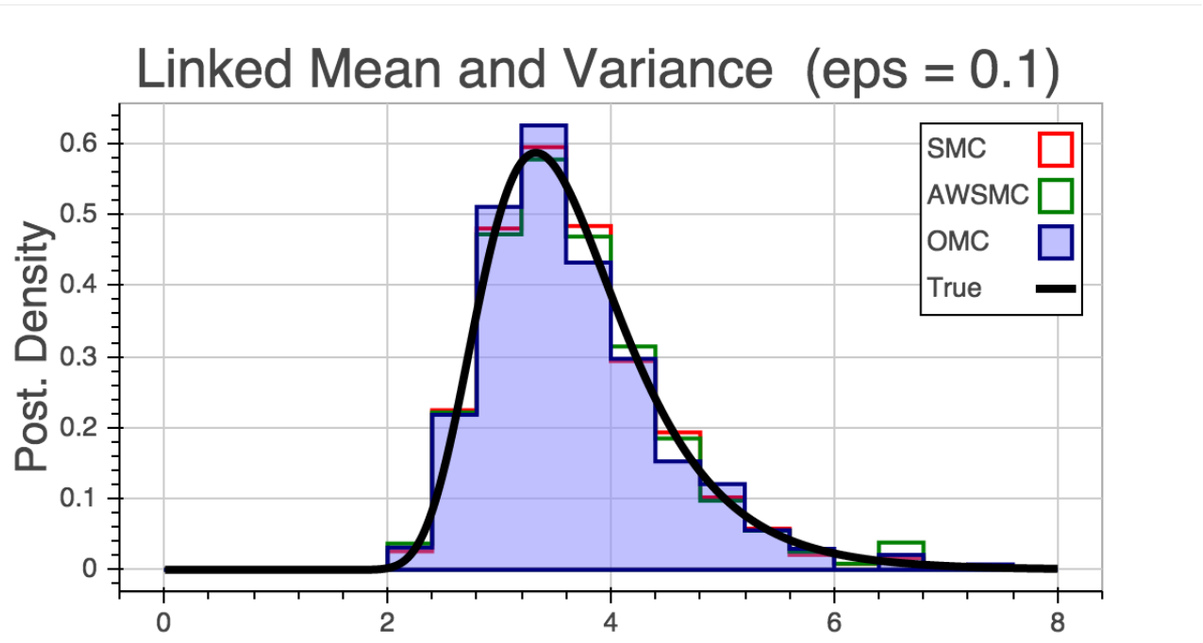
- Simulator $p(x|\theta) = \rho\,\mathcal{N}(\theta, \sigma_1^2) + (1-\rho)\,\mathcal{N}(\theta, \sigma_2^2)$
- x = $\theta + \sqrt{2}\,\mathrm{erf}(2u_2 - 1)\sigma_1^{[u_1<\rho]}\sigma_2^{[u_1\geq\rho]}$
- exact optimum: $\theta_i^o = y - R(\boldsymbol{u}_i)$ jacobian J=1

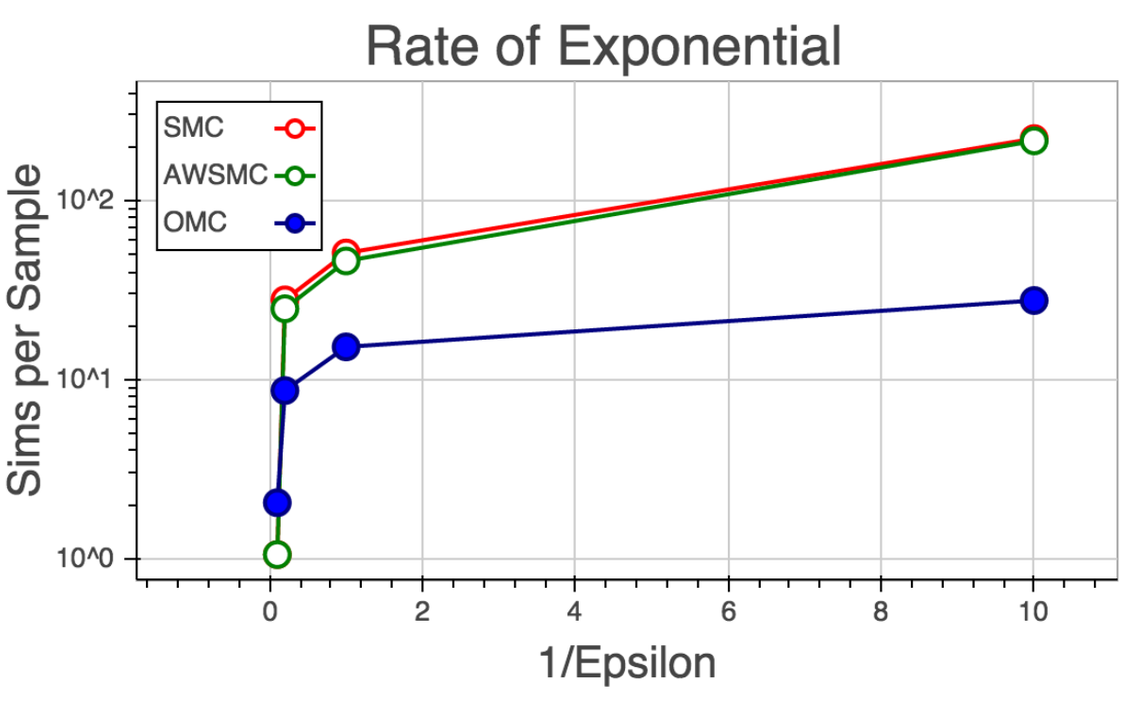


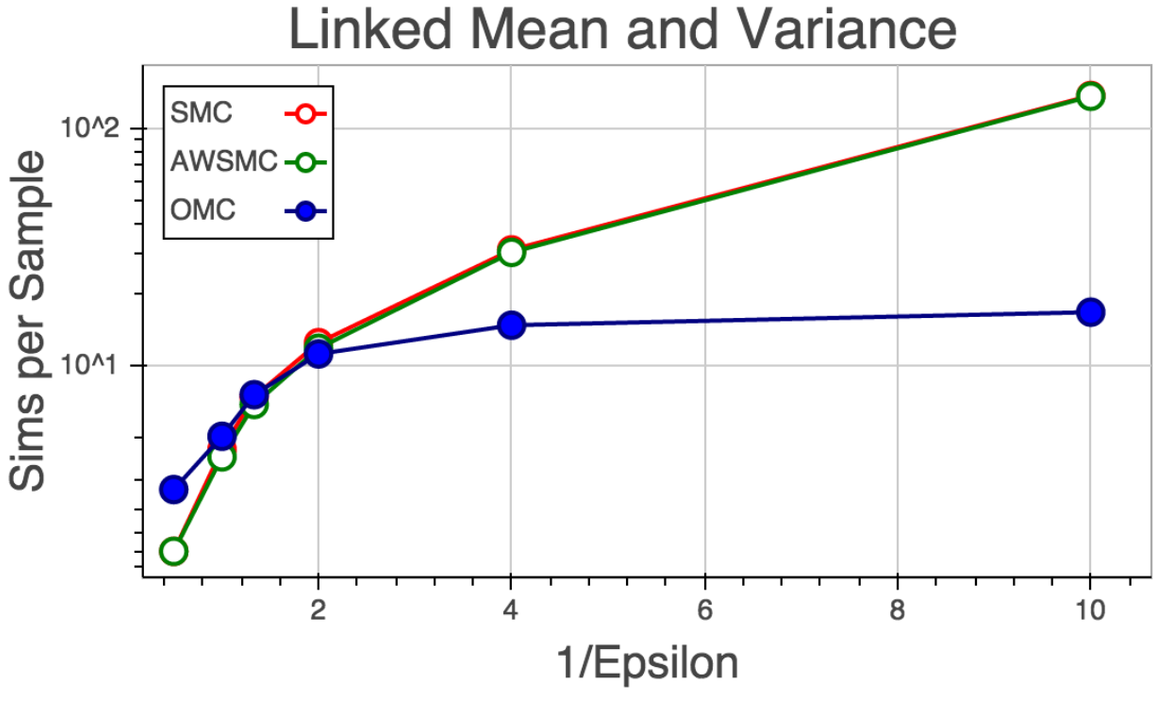


OMC Experiments

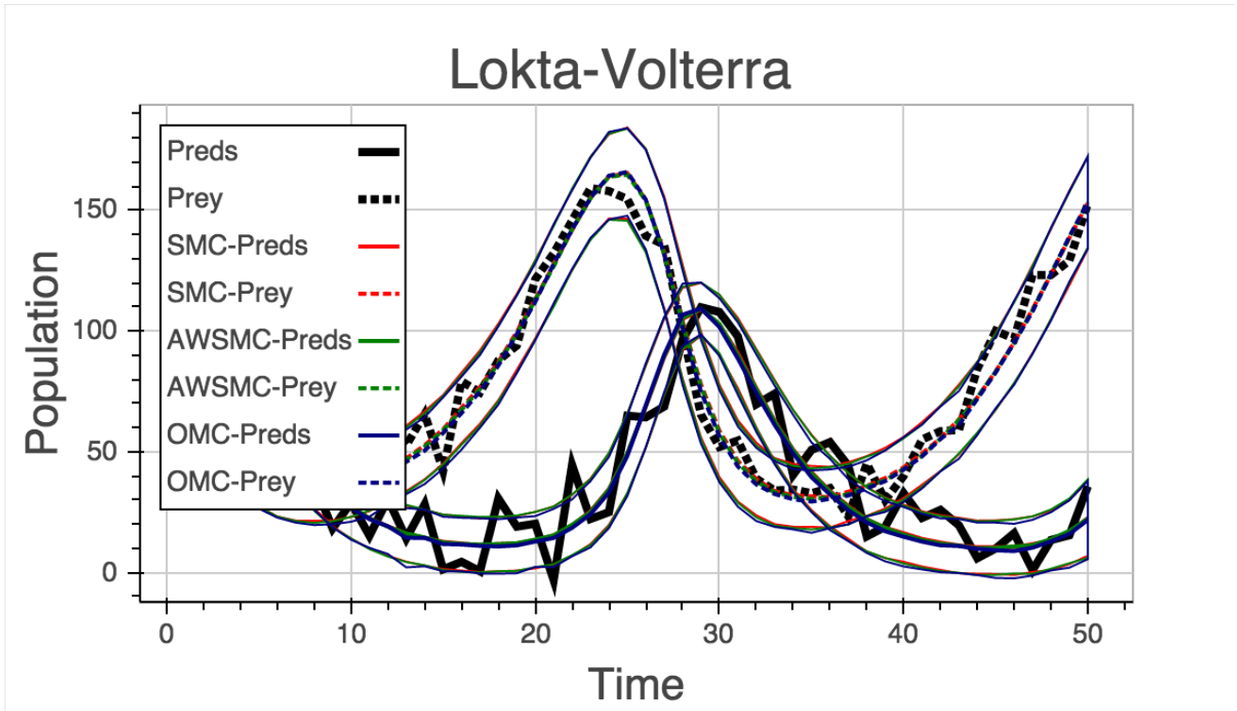


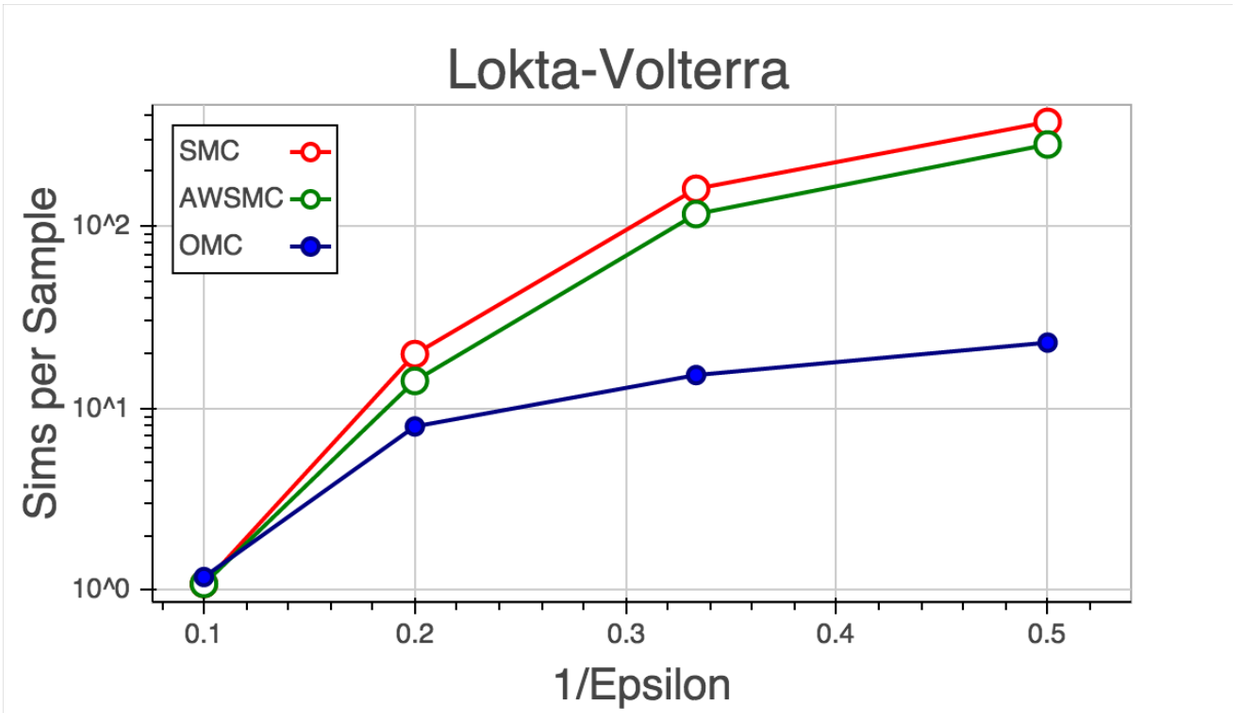


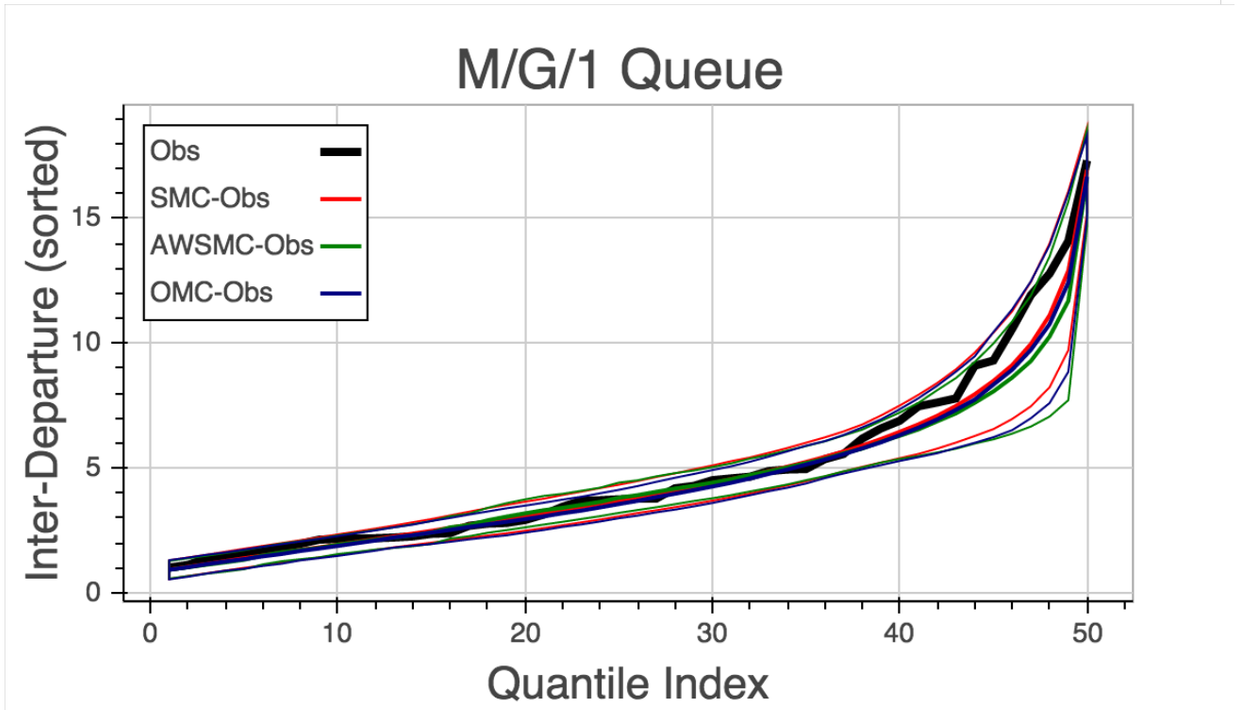


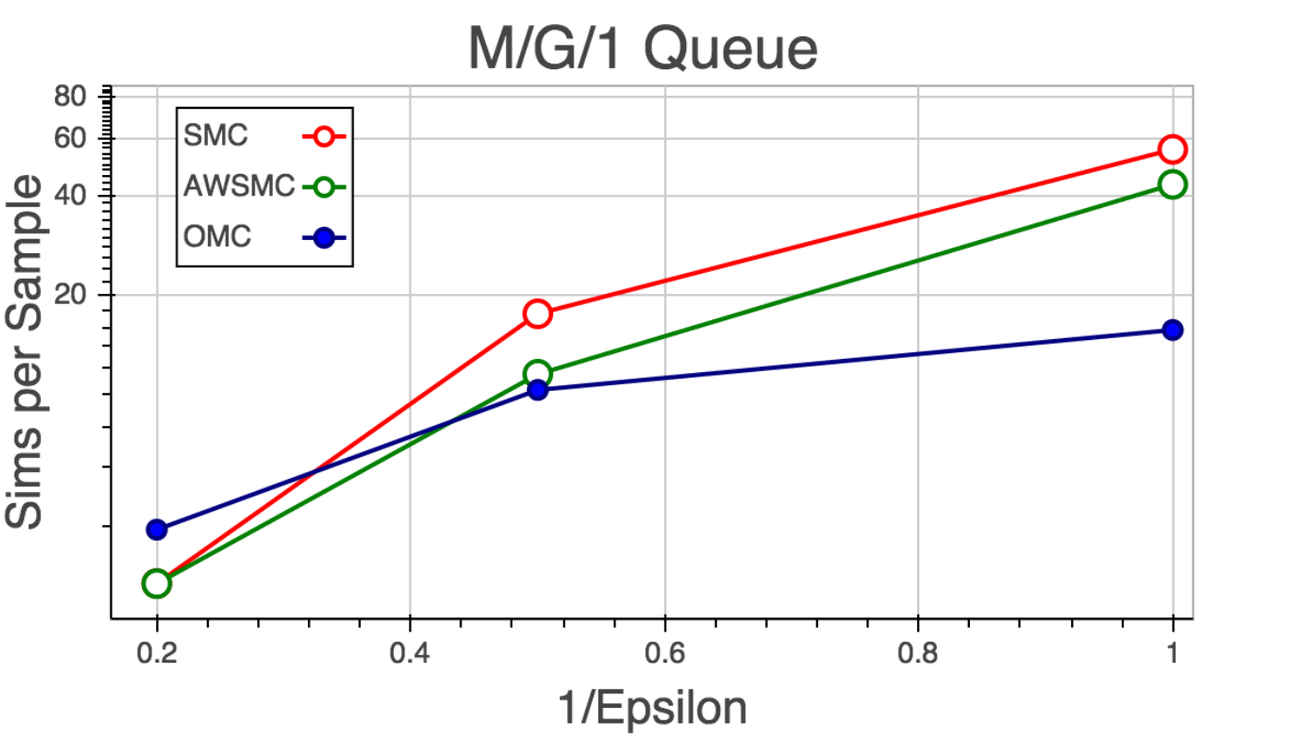


OMC Experiments





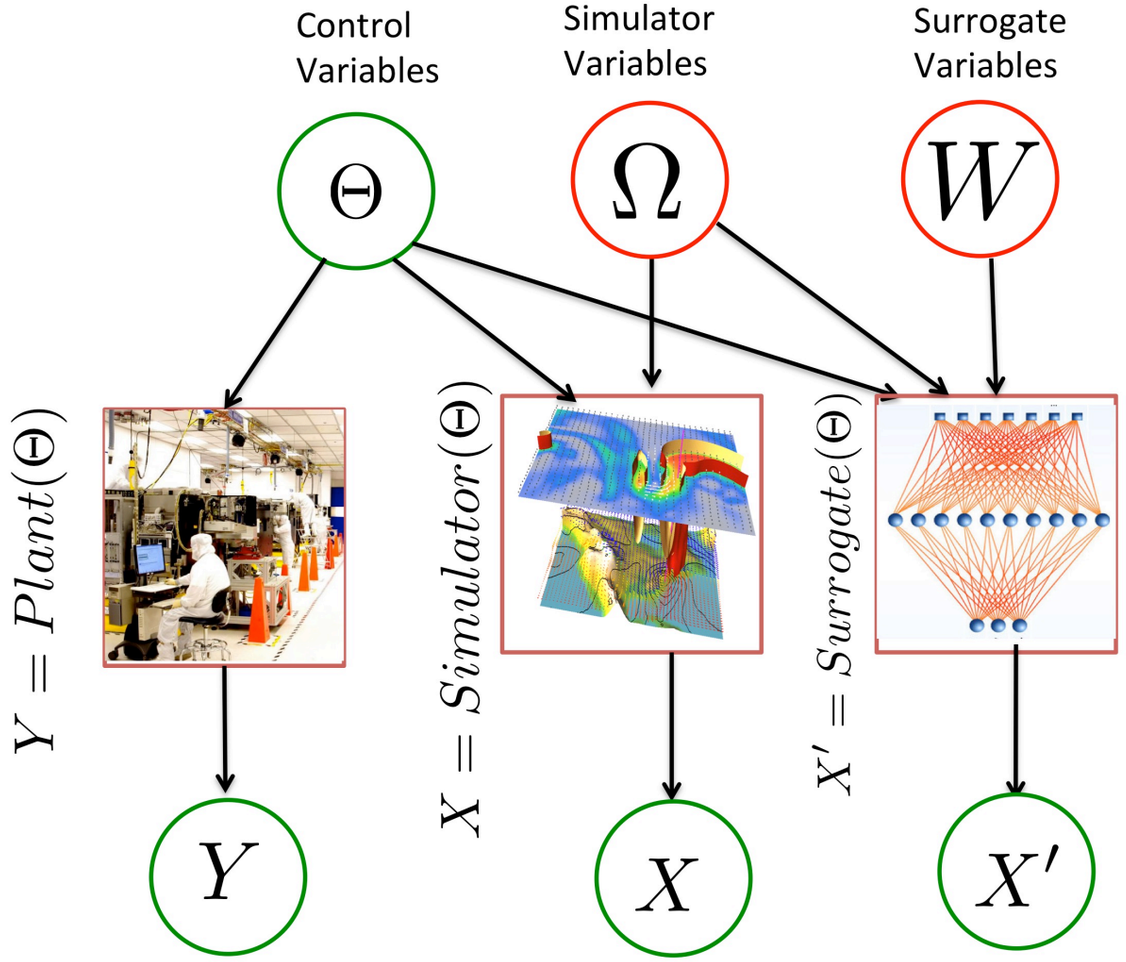




Surrogates

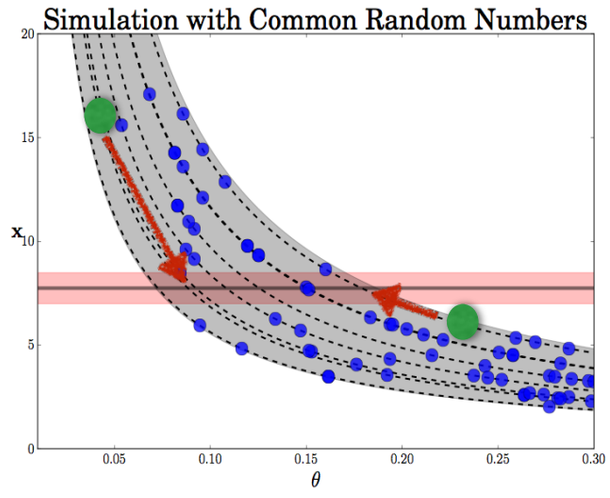
- ABC inference is generally slow due to lack of gradients.

- We can speed up ABC by training a surrogate model and use it to estimate gradients.
- Only use it when the uncertainty in the gradients is small enough.



Conclusions

- OMC is a way to replace a Markov chain with optimisation.
- Embarrassingly parallel.
- When do we give up on optimization and reject sample?
- High dimensions? Large N?
- Works for: $Dim(Y) \geq Dim(\theta)$
- Use low discrepancy sequence to sample from p(u)?
- Learn variational posteriors for Q(uly)?
- Clear connection to inference in probabilistic programs.
- Concurrent work:



[10] Forneron, J.-J. and Ng, S. (2015b). A likelihood-free reverse sampler of the posterior distribution. *arXiv preprint arXiv:1506.04017v1*.