Automating inference with Stan

2015, 2015+

Dustin Tran
Department of Statistics
Harvard University

Joint work with:
Alp Kucukelbir, Rajesh Ranganath, Andrew Gelman, David M. Blei

What is Stan?

Stan is a probabilistic programming language. [1]

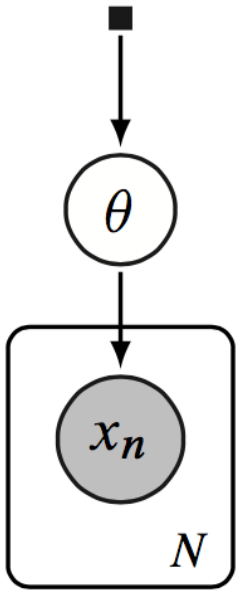
Define *programs* (models)

- declares data and parameter spaces
- define log posterior (or penalized likelihood)

Automatic inference algorithms

- Markov chain Monte Carlo (NUTS, HMC)
- Variational inference (ADVI)
- Penalized maximum likelihood estimation (L-BFGS)

[1] http://mc-stan.org




```
data {
  int N;     // number of observations
  int x[N]; // discrete-valued observations
}
parameters {
  // latent variable, must be positive
  real<lower=0> theta;
}
model {
  // non-conjugate prior for latent variable
  theta ~ exponential(1.0);

  // likelihood
  for (n in 1:N)
    increment_log_prob(poisson_log(x[n], theta));
}
```

Figure 1: Specifying a simple nonconjugate probability model in Stan.

Probabilistic programming

- Abstract modelling from inference
- Rapid prototyping of models, with no need for manual derivations
- Exciting area for research in inference. An automatic testbed across more than 150 classes of models and data sets
    - Constrained parameters (e.g., correlation matrix, simplex)
    - Multilevel generalized linear models with interacted predictors
    - Factor analysis
    - Time series
    - Mixture models
    - Gaussian processes

Stan

Support

- *Platforms*: Linux, Mac OS X, Windows
- *C++ API*: portable, standards compliant (C++03)
- *Interfaces*: Command-line, R, Python, MATLAB, Julia, Stata, Shiny

Users

- 1300 *users group* registrations
- 10,000 manual *downloads* for 2.5.0
- Largest user base for probabilistic programming (BUGS, JAGS, Church, Anglican, ...)

Setup

Given:

- Data set $\mathbf{x}$ ($\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ or streaming $\{\mathbf{x}_1, \mathbf{x}_2, \ldots\}$)
- Joint probability $p(\mathbf{x}, \mathbf{z})$

Model assumptions:

- Latent variables $\mathbf{z} = (\mathbf{z}_1, \ldots, \mathbf{z}_d)$ are continuous
- $p(\mathbf{x}, \mathbf{z})$ is differentiable w.r.t. $\mathbf{z}$

Goal:

- Compute posterior $p(\mathbf{z} \mid \mathbf{x})$

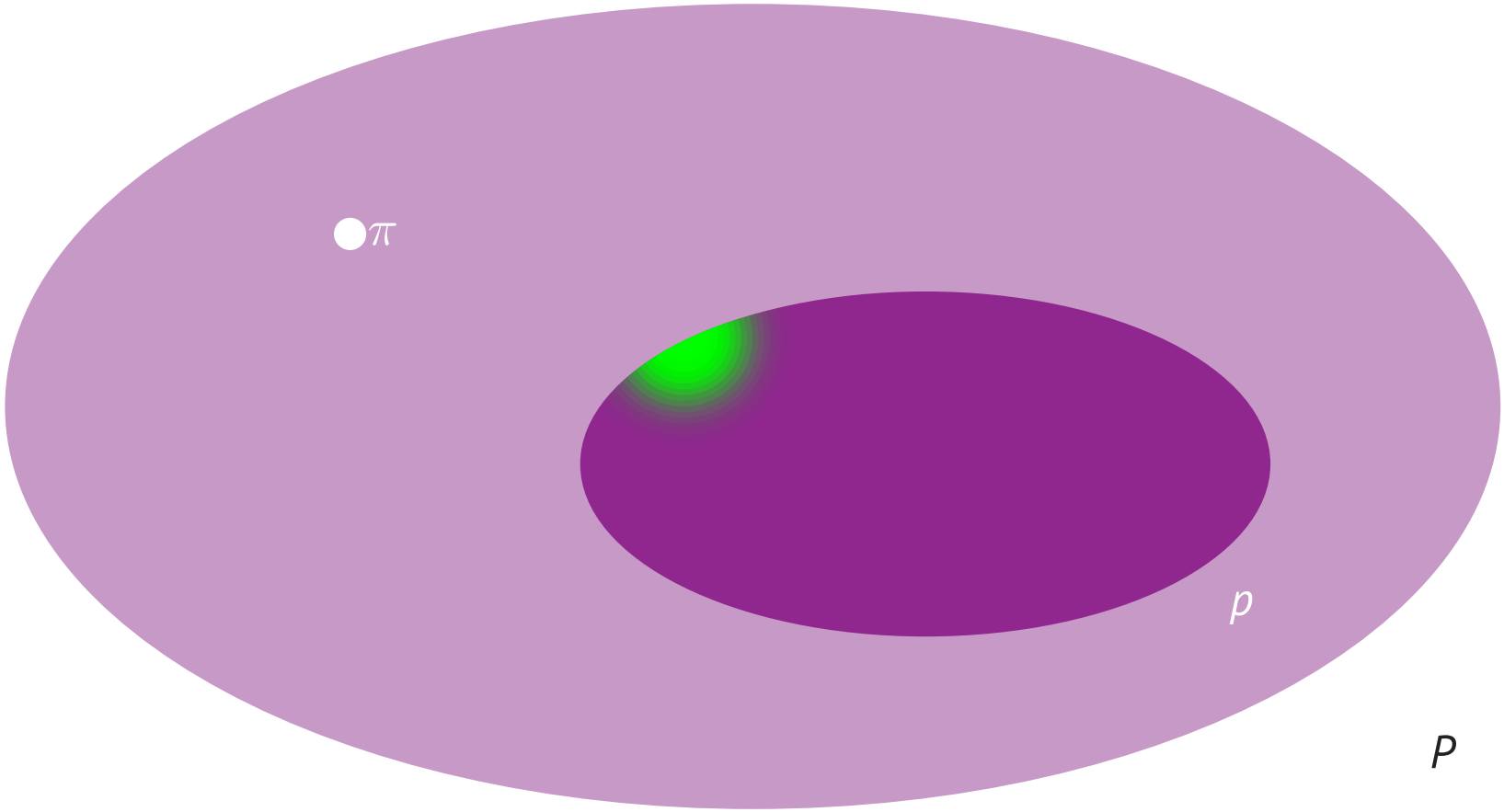
Example

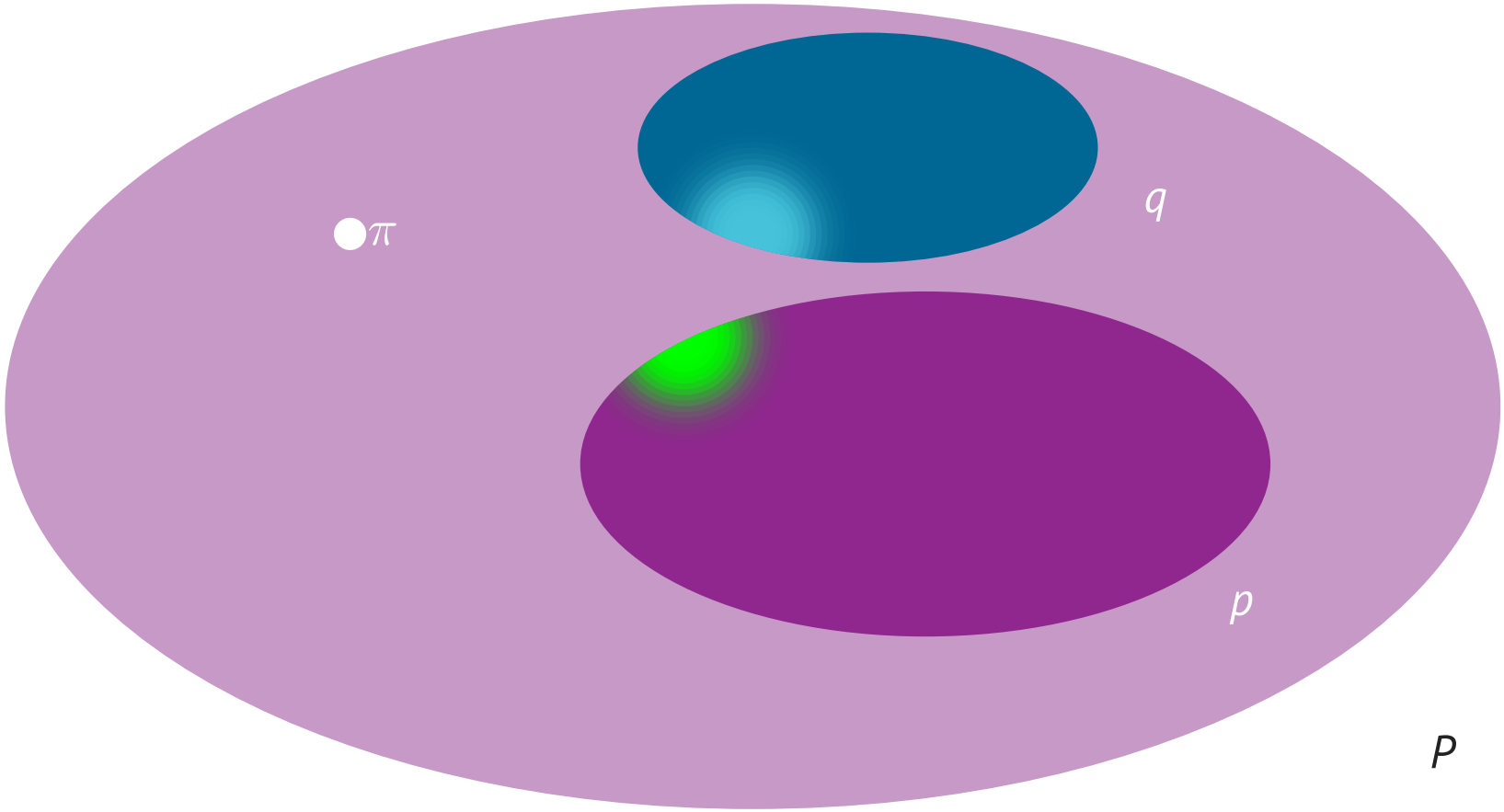
- $p(x \mid z) = \text{Pois}(x \mid z), x \in \{0, 1, 2, \ldots, \}, z \in \mathbb{R}_{>0}$
- $p(z) = \text{Expo}(z) \in \mathbb{R}_{>0}$
- $p(z \mid x) \propto p(z)p(x \mid z)$ nonconjugate posterior

Variational inference

Propose a family of distributions $\{q(\mathbf{z}; \lambda) : \lambda \in \Lambda\}$. Solve

$$\min_{\lambda \in \Lambda} \mathrm{KL}(q \parallel p) = \min_{\lambda \in \Lambda} \int q(\mathbf{z}; \lambda) \log \frac{q(\mathbf{z}; \lambda)}{p(\mathbf{z} \mid \mathbf{x})} \, \mathrm{d}\mathbf{z}$$


π
p
P


π
q
p
P

Can show that

$$\log p(\mathbf{x}) = \mathrm{KL}(q \parallel p) + \mathcal{L}(\lambda)$$

where the Evidence Lower Bound (ELBO) is

$$\mathcal{L}(\lambda) = \mathbb{E}_{q(\mathbf{z};\lambda)}\big[\log p(\mathbf{x}, \mathbf{z})\big] - \mathbb{E}_{q(\mathbf{z};\lambda)}\big[\log q(\mathbf{z};\lambda)\big]$$

Minimizing KL is equivalent to maximizing the ELBO

$$\lambda^* = \arg\max_{\lambda \in \Lambda} \mathcal{L}(\lambda) \quad \text{s.t.} \quad \mathrm{supp}(q(\mathbf{z};\lambda)) \subseteq \mathrm{supp}(p(\mathbf{z} \mid \mathbf{x}))$$

Can show that

$$\log p(\mathbf{x}) = \mathrm{KL}(q \parallel p) + \mathcal{L}(\lambda)$$

Equivalent to maximizing the ELBO

$$\lambda^* = \arg\max_{\lambda \in \Lambda} \mathcal{L}(\lambda) \quad \text{s.t.} \quad \mathrm{supp}(q(\mathbf{z}; \lambda)) \subseteq \mathrm{supp}(p(\mathbf{z} \mid \mathbf{x}))$$

$$\mathcal{L}(\lambda) = \underbrace{\mathbb{E}_{q(\mathbf{z};\lambda)}\left[\log p(\mathbf{x}, \mathbf{z})\right]}_{\text{energy}} \underbrace{-\mathbb{E}_{q(\mathbf{z};\lambda)}\left[\log q(\mathbf{z}; \lambda)\right]}_{\text{entropy}}$$

1. Transformations

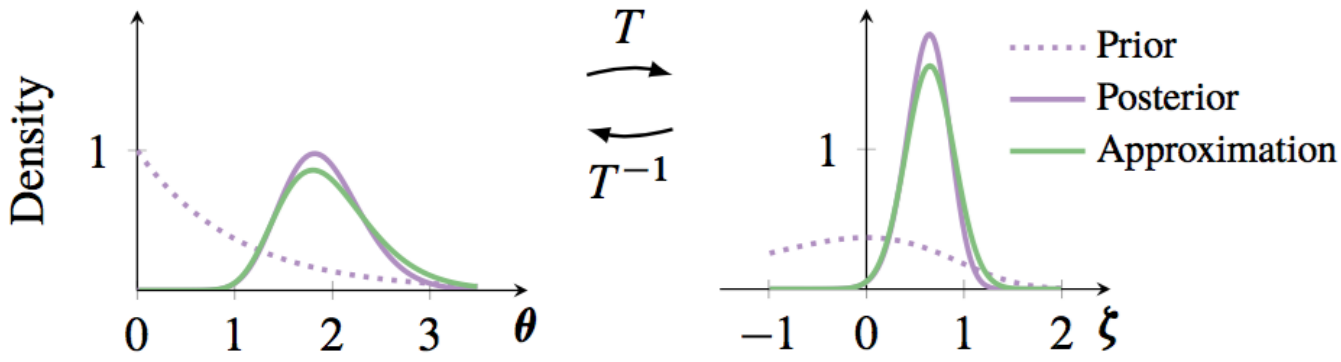
Define a one-to-one differentiable mapping

$$T : \text{supp}(p(\mathbf{z})) \to \mathbb{R}^d,$$

and let $\zeta = T(\mathbf{z})$. The transformed joint probability is

$$g(\mathbf{x}, \zeta) = p(\mathbf{x}, T^{-1}(\zeta)) \left| \det J_{T^{-1}}(\zeta) \right|$$

1. Transformations



(a) Latent variable space

(b) Real coordinate space

2. Implicit Gaussian approximation

Specify a Gaussian distribution

$$q(\zeta; \mu, \sigma^2) = \mathcal{N}(\zeta; \mu, \sigma^2 \mathbf{I}) = \prod_{i=1}^{d} \mathcal{N}(\zeta_i; \mu_i, \sigma_i^2)$$

Transformed ELBO

$$\mathcal{L}(\mu, \sigma^2) = \mathbb{E}_{q(\zeta;\mu,\sigma^2)} \left[\log p(\mathbf{x}, T^{-1}(\zeta)) + \log \left| \det J_{T^{-1}}(\zeta) \right| \right] + \underbrace{\mathbb{H}\left[q(\zeta; \mu, \sigma^2)\right]}_{\text{analytic form}}$$

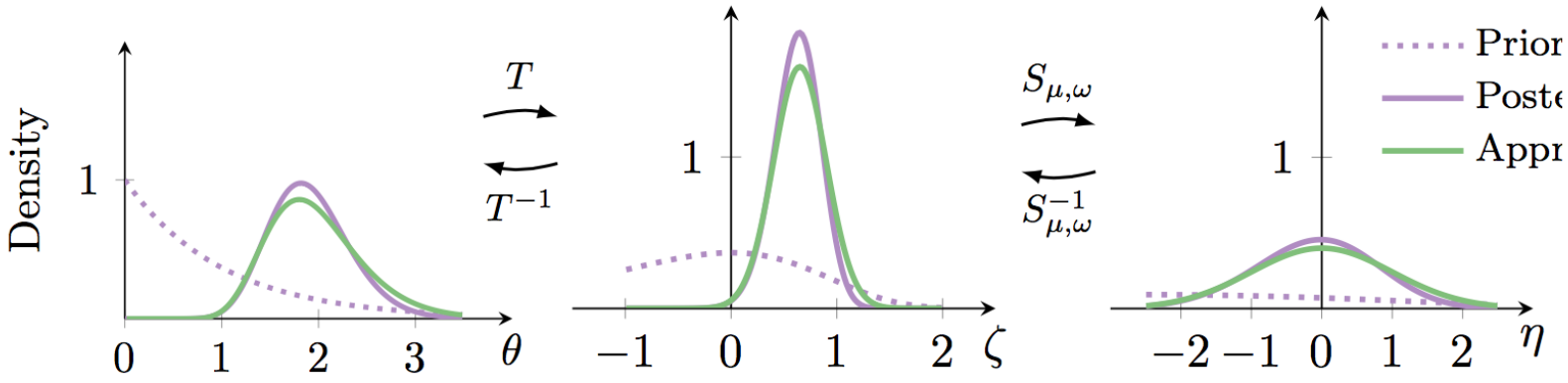
3. Reparameterization

Define reparameterization $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ such that $S(\epsilon) \sim \mathcal{N}(\mu, \sigma^2\mathbf{I})$:

$$\zeta = S(\epsilon) = \sigma^{-1}(\epsilon - \mu)$$

Fully transformed ELBO

$$\mathcal{L}(\mu, \sigma^2) = \mathbb{E}_{\mathcal{N}(\epsilon)}\left[\log p(\mathbf{x}, T^{-1}(S(\epsilon))) + \log\left|\det J_{T^{-1}}(S(\epsilon))\right|\right] + \mathbb{H}\left[q(\zeta; \mu, \sigma^2)\right]$$



(a) Latent variable space

(b) Real coordinate space

(c) Standardized space

4. Stochastic approximation

$$\nabla_\mu \mathcal{L} = \mathbb{E}_{\mathcal{N}(\epsilon)} \left[\nabla_{\mathbf{z}} \log p(\mathbf{x}, \mathbf{z}) \nabla_\zeta T^{-1}(\zeta) + \nabla_\zeta \log \left| \det J_{T^{-1}}(\zeta) \right| \right]$$

$$\nabla_{\sigma^2} \mathcal{L} = \mathbb{E}_{\mathcal{N}(\epsilon)} \Big[\left(\nabla_{\mathbf{z}} \log p(\mathbf{x}, \mathbf{z}) \nabla_\zeta T^{-1}(\zeta) + \nabla_\zeta \log \left| \det J_{T^{-1}}(\zeta) \right| \right) \epsilon^\top \operatorname{diag}(\sigma^2 \mathbf{I}) \Big] + \mathbf{1}$$

Algorithm 1: Automatic differentiation variational inference (ADVI)

Input: Dataset $\mathbf{X} = \boldsymbol{x}_{1:N}$, model $p(\mathbf{X}, \boldsymbol{\theta})$.
Set iteration counter $t = 1$.
Initialize $(\boldsymbol{\mu}, \boldsymbol{\sigma}^2)$.

while *not converged* **do**

Draw sample $\boldsymbol{\eta} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.

Calculate unbiased estimate of $\nabla_{\boldsymbol{\mu}} \mathcal{L}$.

Calculate unbiased estimate of $\nabla_{\boldsymbol{\sigma}^2} \mathcal{L}$.

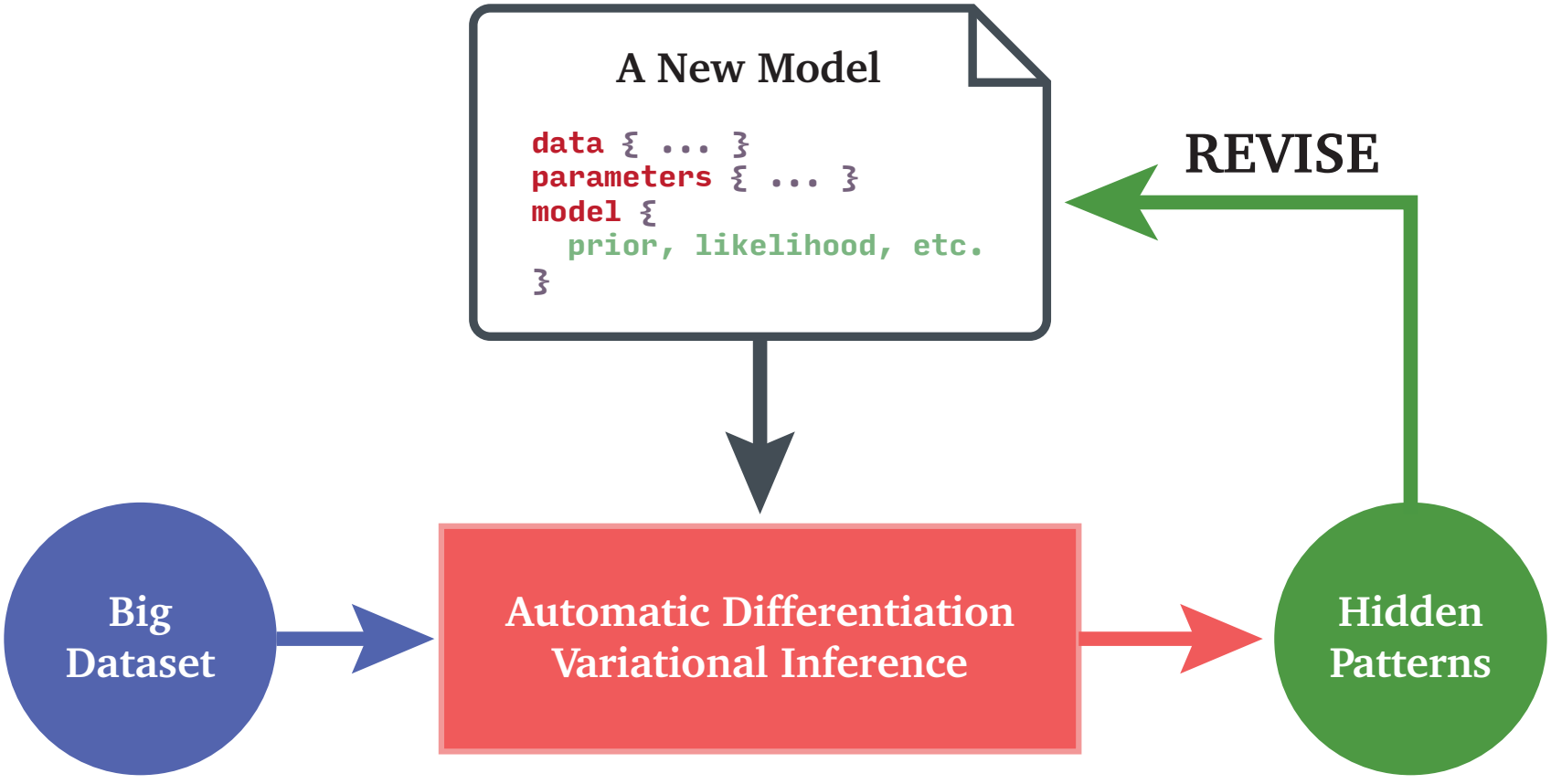
Calculate step-size $\boldsymbol{\rho}_t$.

$\boldsymbol{\mu} \longleftarrow \boldsymbol{\mu} + \boldsymbol{\rho}_t \nabla_{\boldsymbol{\mu}} \mathcal{L}$.

$\boldsymbol{\sigma}^2 \longleftarrow \boldsymbol{\sigma}^2 + \boldsymbol{\rho}_t \nabla_{\boldsymbol{\sigma}^2} \mathcal{L}$

Increment iteration counter.

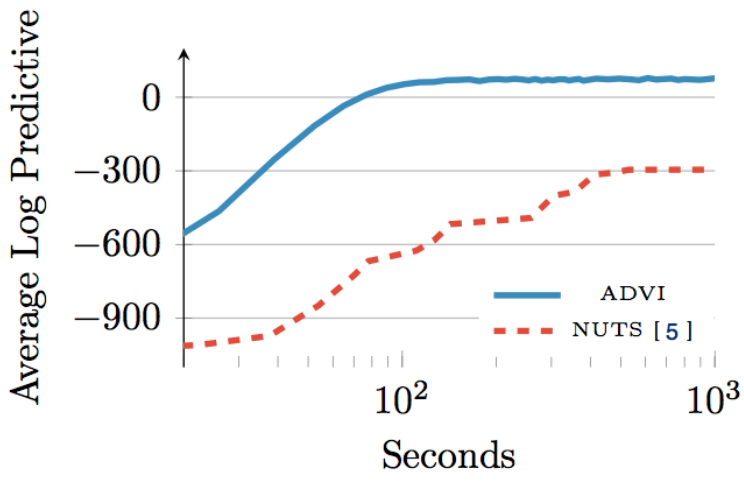
end
Return $(\boldsymbol{\mu}, \boldsymbol{\sigma}^2)$.


A New Model
data { ... }
parameters { ... }
model {
prior, likelihood, etc.
}
REVISE
Big Dataset
Automatic Differentiation Variational Inference
Hidden Patterns

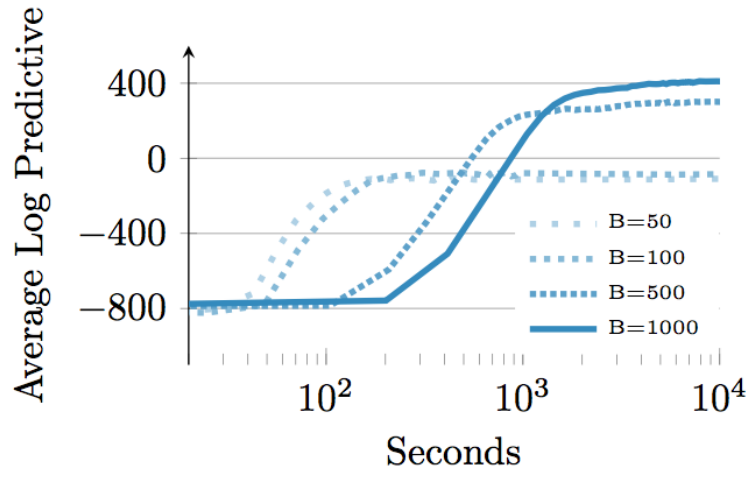
Taxi rides

- 1.7 million taxi rides in Porto, Portugal over a year
- Variable-length sequences of taxi trajectors, noisy spatial locations
- Massive amounts of missing (not at random) data


São Mamede de Infesta
Matosinhos
Rio Tinto
Parque da Cidade
São Pedro da Afurada
Alumiara
Lavadores
Canidelo
Madalena
Vila Nova de Gaia
Barrosos
Valbom
Gondomar
A 4
A 1
A 44
IP 1
EN 15-1
EN 15
EN 108
EN 222
EN 1-15
EN 208
EM 612-1
Rio Douro



(a) Subset of 1000 images



(b) Full dataset of 250 000 images

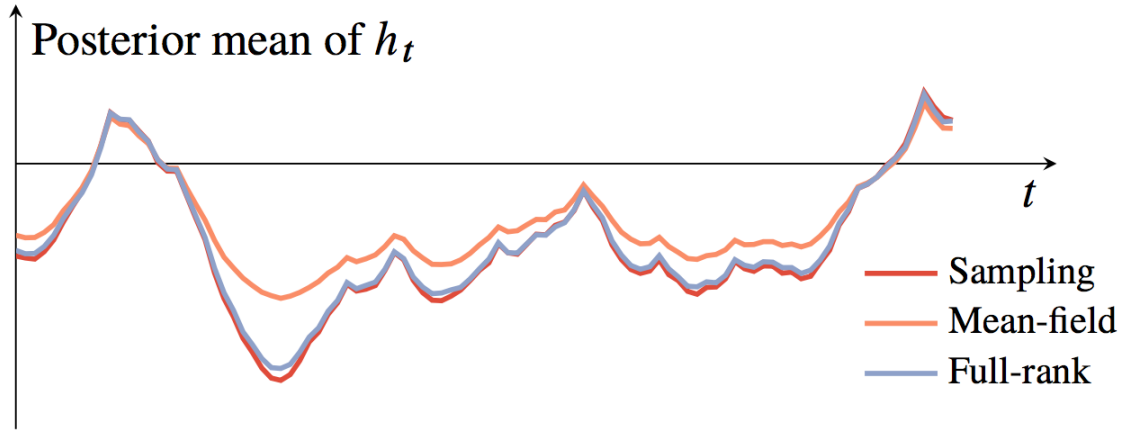
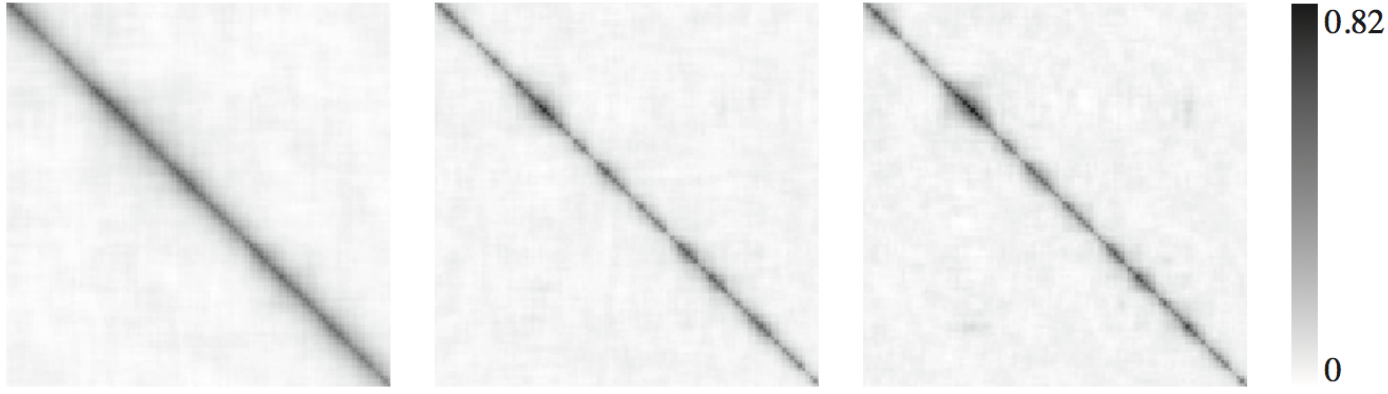


Figure 6: Comparison of posterior mean estimates of volatility.



(a) Mean-field **(b)** Full-rank **(c)** Sampling

Figure 7: Comparison of empirical posterior covariance matrices.

Alp Kucukelbir, Dustin Tran, Rajesh Ranganath, Andrew Gelman, and David M Blei. Automating variational inference. *In preparation*, 2015.

Rajesh Ranganath, Sean Gerrish, and David M Blei. Black box variational inference. In *Artificial Intelligence and Statistics*, 2014.

Rajesh Ranganath, Dustin Tran, and David M. Blei. Hierarchical variational models. In *Preprint (Under review)*, 2015.

Dustin Tran, David M. Blei, and Edoardo M. Airoldi. Variational inference with copula augmentation. In *Neural Information Processing Systems (forthcoming)*, 2015.

More details at dustintran.com