BML. 5. Large scale Bayesian Inference

DataLab CSIC

DABID

DABID (Data Age Bayesian Inference and Decisions)

Bayes in Large Scale problems

Sources

Martin, Frazier and Robert (2023 a, b)

Blei, Kucukelbir, McAuliffe (2017)

Ma, Chen, Fox (2015), Nemeth, Fearnhead (2019)

Gallego, DRI (2022)

Computational problem in Bayesian analysis

Computing the posterior

$$f(\theta|x) = \frac{f(x|\theta)\, f(\theta)}{f(x)} = \frac{f(x|\theta)\, f(\theta)}{\int f(x|\theta)\, f(\theta)\, d\theta} \propto f(x|\theta)\, f(\theta)$$

Large scale problems. The modern statistical paradigm (but not always and not for the whole problem)

What if the amount of data is large? (Big data problems)
What if the amount of parameters is large? (e.g. Deep neural nets)

Methods

Simulation based

SG-MCMC, ABC

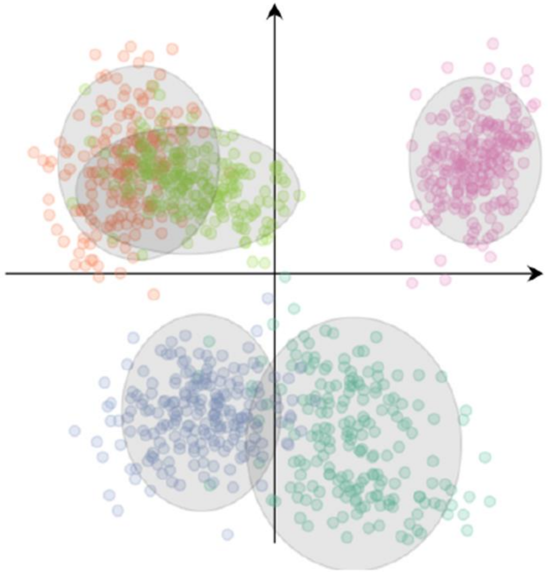
Optimization based

VB, INLA

Hybrids

Motivation. Mixtures

Mixture of gaussians



Mixture of K unit-variance (1-d) Gaussians

For each observation *x*, a cluster assignment *c*

Full model is

$$
\begin{aligned}
\mu_k &\sim \mathcal{N}(0, \sigma^2), & k &= 1, \ldots, K, \\
c_i &\sim \text{categorical}(1/K, \ldots, 1/K), & i &= 1, \ldots, n, \\
x_i \mid c_i, \mu &\sim \mathcal{N}\left(c_i^\top \mu, 1\right) & i &= 1, \ldots, n.
\end{aligned}
$$

Joint density of parameters, latent and observed variables

$$
p(\mu, \mathbf{c}, \mathbf{x}) = p(\mu) \prod_{i=1}^{n} p(c_i) p(x_i \mid c_i, \mu).
$$

Mixture of gaussians

Evidence v1

$$p(\mathbf{x}) = \int p(\mu) \prod_{i=1}^{n} \sum_{c_i} p(c_i)\, p(x_i \mid c_i, \mu)\, d\mu$$

Evidence v2

$$p(\mathbf{x}) = \sum_{\mathbf{c}} p(\mathbf{c}) \int p(\mu) \prod_{i=1}^{n} p(x_i \mid c_i, \mu)\, d\mu.$$

Problems with Gibbs sampler for mixtures

Gibbs sampler for exponential mixtures, but structure is general

1. Start with arbitrary values $(\mathbf{q}^0, \boldsymbol{\mu}^0, \mathbf{z}^0)$, $i = 0$.
2. Until convergence, iterate through
   - Generate $\mathbf{z}_j^{i+1} \sim \mathbf{z}_j | t_j, \mathbf{q}^i, \boldsymbol{\mu}^i$, $j = 1, \ldots, n_s$.
   - Generate $\mathbf{q}^{i+1} \sim \mathbf{q} | \mathbf{t}, \mathbf{z}^{i+1}$.
   - Generate $\mu_j^{i+1} \sim \mu_j | \mathbf{t}, \mathbf{z}^{i+1}$, $j = 1, \ldots, k$.
   - Set $i = i + 1$.

And similarly for MH and HMC, does not scale….
Think also in terms of large number of parameters….

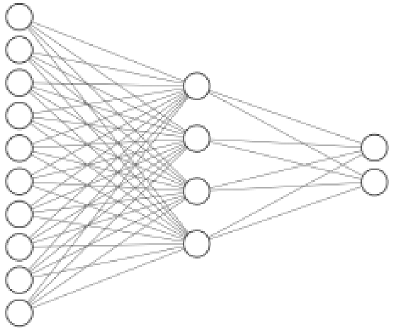
Motivation. Neural Nets

Gibbs-Met

$$y = \sum_{j=1}^{m} \beta_j \psi(x'\gamma_j) + \epsilon$$

$$\epsilon \sim N(0, \sigma^2),$$

$$\psi(\eta) = \exp(\eta)/(1 + \exp(\eta)).$$



We discuss now Bayesian approaches to shallow NNs. assuming standard priors in Bayesian hierarchical modeling, see e.g. Lavine & West (1992): $\beta_j \sim N(\mu_\beta, \sigma_\beta^2)$ and $\gamma_j \sim N(\mu_\gamma, S_\gamma^2)$, completed with priors over the hyperparameters $\mu_\beta \sim N(a_\beta, A_\beta)$, $\mu_\gamma \sim N(a_\gamma, A_\gamma)$, $\sigma_\beta^{-2} \sim Gamma(c_b/2, c_b/2)$, $S_\gamma^{-1} \sim Wish(c_\gamma, (c_\gamma C_\gamma)^{-1})$ and $\sigma^{-2} \sim Gamma(s/2, s/2)$. In the

Start with arbitrary (β, γ, ν).
while *not convergence* **do**
    Given current (γ, ν), draw β from $p(\beta|\gamma, \nu, y)$ (a multivariate normal).
    for $j = 1, \ldots, m$, *marginalizing in* β *and given* ν **do**
        Generate a candidate $\tilde{\gamma}_j \sim g_j(\gamma_j)$.
        Compute $a(\gamma_j, \tilde{\gamma}_j) = \min\left(1, \frac{p(D|\tilde{\gamma}, \nu)}{p(D|\gamma, \nu)}\right)$ with $\tilde{\gamma} = (\gamma_1, \gamma_2, \ldots, \tilde{\gamma}_j, \ldots, \gamma_m)$.
        With probability $a(\gamma_j, \tilde{\gamma}_j)$ replace γ_j by $\tilde{\gamma}_j$. If not, preserve γ_j.
    end
    Given β and γ, replace ν based on their posterior conditionals:
    $p(\mu_\beta|\beta, \sigma_\beta)$ is normal; $p(\mu_\gamma|\gamma, S_\gamma)$, multivariate normal; $p(\sigma_\beta^{-2}|\beta, \mu_\beta)$, Gamma; $p(S_\gamma^{-1}|\gamma, \mu_\gamma)$, Wishart; $p(\sigma^{-2}|\beta, \gamma, y)$, Gamma.
end

HMC

$$U(\theta) = \tau_\beta \sum_{j=1}^{m} \beta_j^2/2 + \tau_\gamma \sum_{j=1}^{d} \sum_{k=1}^{m} \gamma_{j,k}^2/2 + \tau \sum_{i=1}^{n} (y_i - f_i(\beta,\gamma))^2/2,$$

$$H(\theta, r) = U(\theta) + \frac{1}{2} \sum_{j=1}^{l} q_j^2$$

1 Start with arbitrary $\theta_0 = (\beta_0, \gamma_0)$.

2 **while** *not convergence* **do**

3 Given current θ_t and $q_t \sim \mathcal{N}(0, I)$, perform one or more leapfrog integration steps

$$q_{t+\frac{1}{2}} = q_t - \frac{\epsilon}{2} \nabla U(\theta_t)$$
$$\theta_{t+1} = \theta_t + \epsilon q_{t+\frac{1}{2}}$$
$$q_{t+1} = q_{t+\frac{1}{2}} - \frac{\epsilon}{2} \nabla U(\theta_{t+1})$$

to reach θ^* and q^*.

4 Compute $\alpha(\theta_t, \theta^*) = \min\left\{1, \frac{\exp H(\theta^*, r^*)}{\exp H(\theta_t, r_t)}\right\}$.

5 Accept θ^* as θ_{t+1} with probability $\alpha(\theta_t, \theta^*)$, else discard it.

6 **end**

Variational inference

Variational inference

- Detailed descriptions

Blei, Kucucelbir, MacAulliffe Variational inference a review for statisticians

Jordan et al. An introduction to variational methods for graphical models

Fox, Roberts. A tutorial on variational Bayesian inference

- Video by the great David Blei

https://www.youtube.com/watch?v=Dv86zdWjJKQ

VI

- Converts inference problem into one of optimization
- Faster than MCMC (as we know it today)
- Scales better for large data sets (as we know it today)
- May underestimate uncertainty, biased

VI

- Recall Laplace integration in Chapter 2

- Origin in statistical physics (mean field methods to fit neural nets)
- Generalized to probabilistic models in the 90's
- Scalable to Big Data with stochastic variants
- Main Bayesian approach to most deep ML, AI models currently

Concept

- Start with probabilistic model of observed variables **x** and latent variables **z (labels and pars)** $p(\mathbf{z}, \mathbf{x})$

- Want to estimate $p(\mathbf{z}|\mathbf{x})$ but difficult because of p(x)

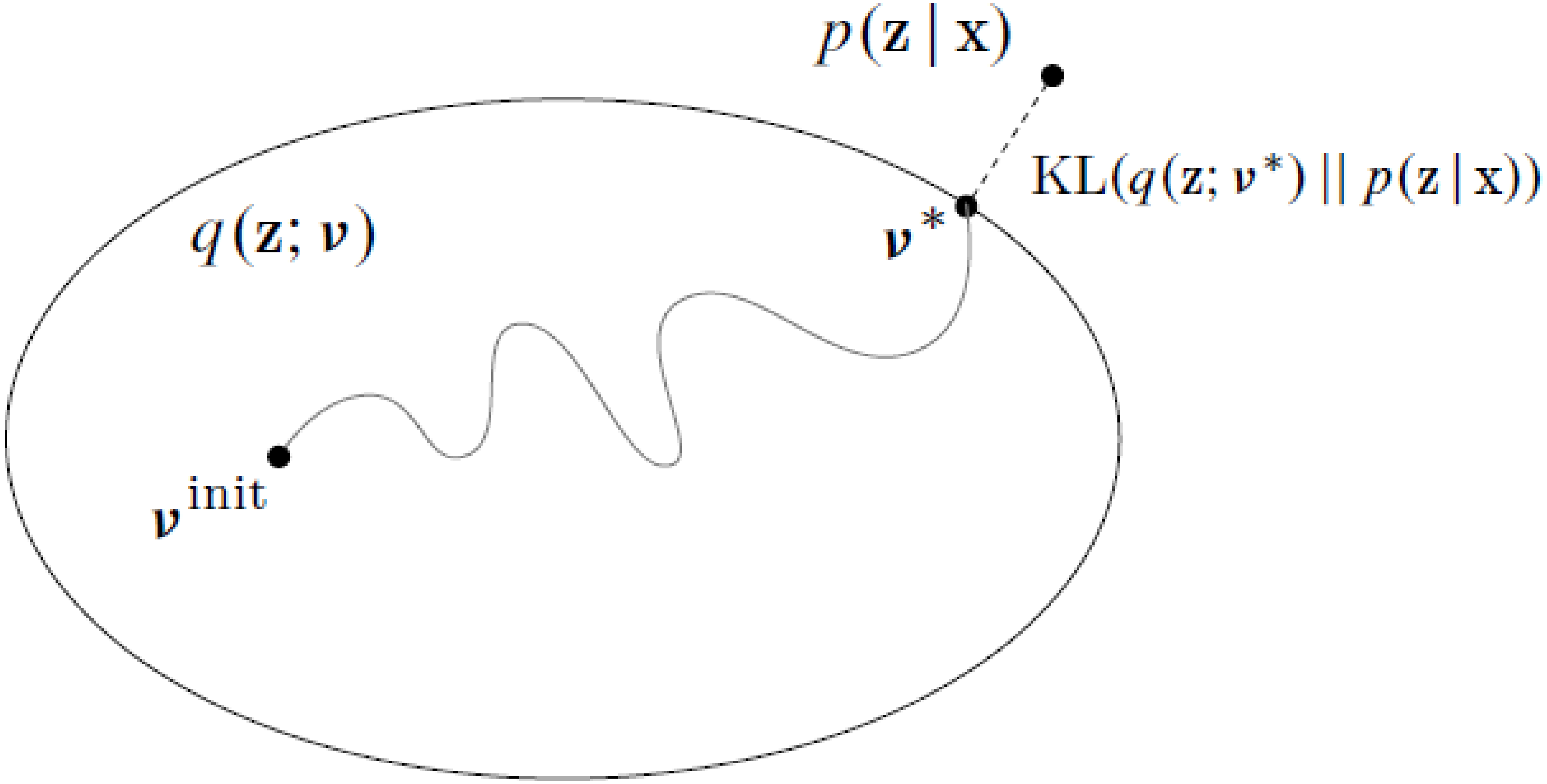
- Approximate with a distribution of efficient computation $q(\mathbf{z})$

- Minimise distance so that they resemble

- Use Kullback-Leibler divergence $$KL(q||p) = \int q(\mathbf{z}) \log \frac{q(\mathbf{z})}{p(\mathbf{z}|\mathbf{x})} d\mathbf{z}$$

Disimilarity measure
Non negative
0 iff they coincid

Concept. Inference as optimization



Variational family $q(\mathbf{z}; \boldsymbol{\nu})$

Find variational parameters $\boldsymbol{\nu}$ to approximate the posterior through KL

Concept. ELBO

But direct KL optimization impossible!!

$$KL(q||p) \equiv \mathbb{E}_q[\log q(\mathbf{z})] - \mathbb{E}_q[\log p(\mathbf{x}, \mathbf{z})] + \log p(\mathbf{x})$$

We may remove last term. Evidence lower bound

$$ELBO(q) = \mathbb{E}_q[\log p(\mathbf{x}, \mathbf{z})] - \mathbb{E}_q[\log q(\mathbf{z})]$$

Besides $\log p(\mathbf{x}) \geq ELBO(q)$

$$q^\star(\mathbf{z}) = \arg\min_{q \in \mathcal{Q}} KL(q(\mathbf{z})||p(\mathbf{z}|\mathbf{x}))$$

$$\longrightarrow$$

$$q^\star(\mathbf{z}) = \arg\max_{q \in \mathcal{Q}} ELBO(q)$$

VI algos

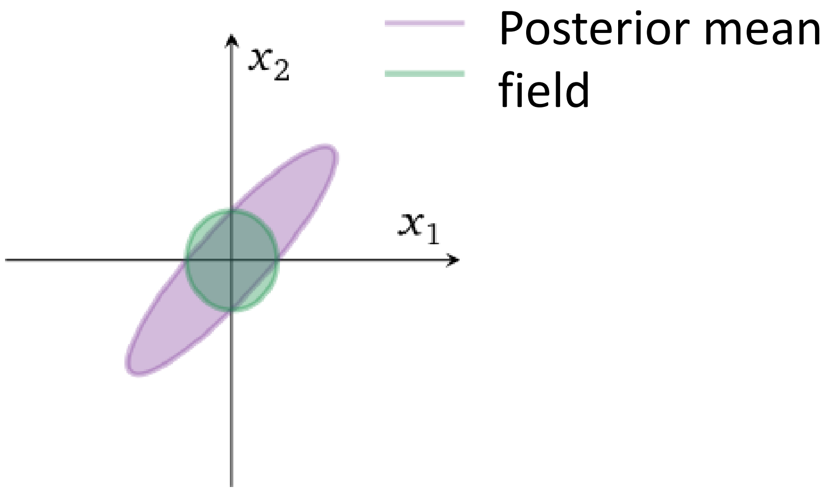
1. Formulate model

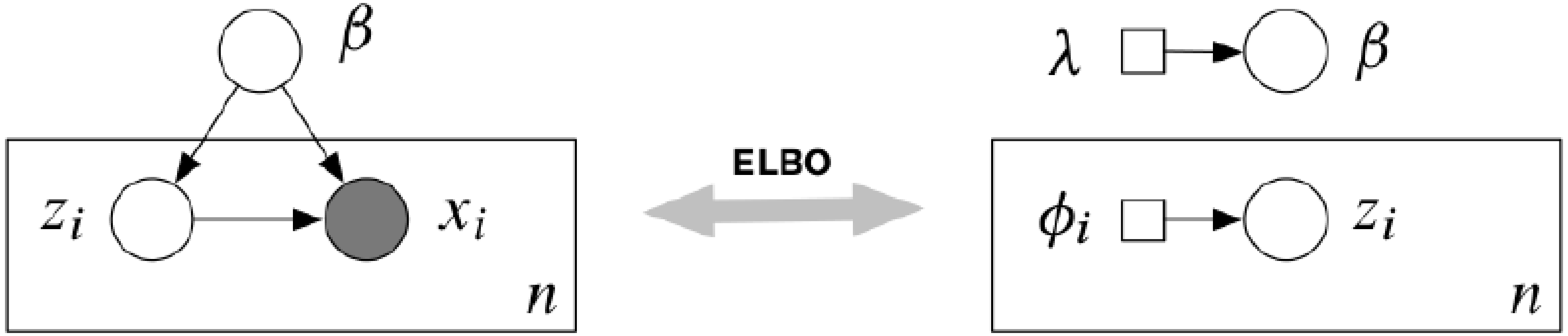
2. Formulate variational family
3. Formulate optimization problem
4. Solve optimization problem

$$q^\star(\mathbf{z}) = \arg\max_{q \in \mathcal{Q}} ELBO(q)$$

Mean field family +Coordinate ascent

$$q(\mathbf{z}) = \prod_{i=1}^{m} q_i(z_i)$$





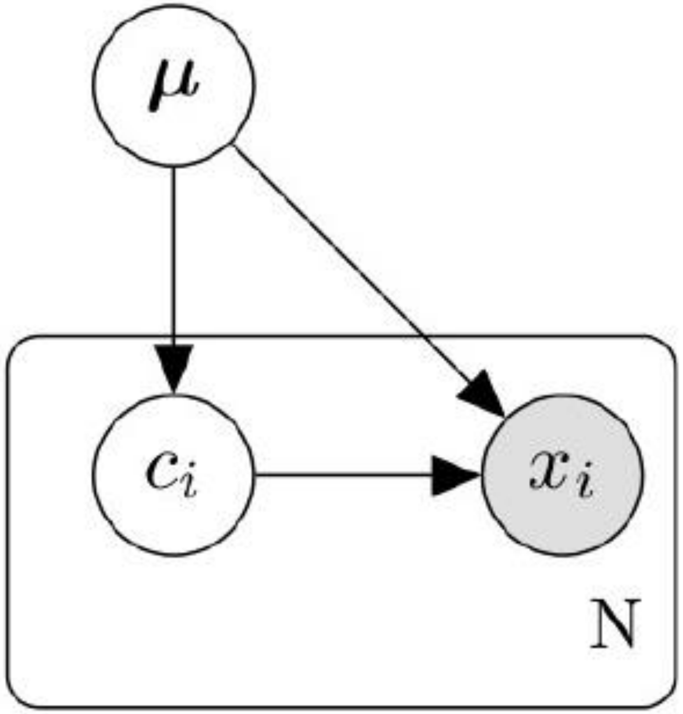
Mean field family: Latent variables independent, each governed by a distinct factor in the var density
Coordinate ascent: At each cycle, consider each coordinate and optimize in it (unidimensional)

Mean field family+coordinate ascent. CAVI

Input: Probability distribution $p(\mathbf{x}, \mathbf{z})$, data $\boldsymbol{x}$
Result: A variational PDF $q(\mathbf{z}) = \prod_{j=1}^{m} q_j(z_j)$
Initialize: Variational factors $q_j(z_j)$
while *ELBO has not converged* **do**
    for $j \in \{1..m\}$ **do**
        $q_j(z_j) \propto \exp\{\mathbb{E}_{-j}[\log p(\boldsymbol{z}, \boldsymbol{x}]\}$
    end
    $ELBO(q) = \mathbb{E}[\log p(\mathbf{z}, \mathbf{x})] + \mathbb{E}[\log q(\mathbf{z})]$
end

CAVI. Mixture of gaussians

$$q(\boldsymbol{\mu}, \mathbf{c}) = \prod_{k=1}^{K} q(\mu_k; m_k, s_k^2) \prod_{i=1}^{n} q(c_i; \phi_i)$$



$$ELBO(\mathbf{m}, \mathbf{s^2}, \boldsymbol{\phi}) = \sum_{k=1}^{K} \mathbb{E}_q[\log p(\mu_k); m_k, s_k^2]$$

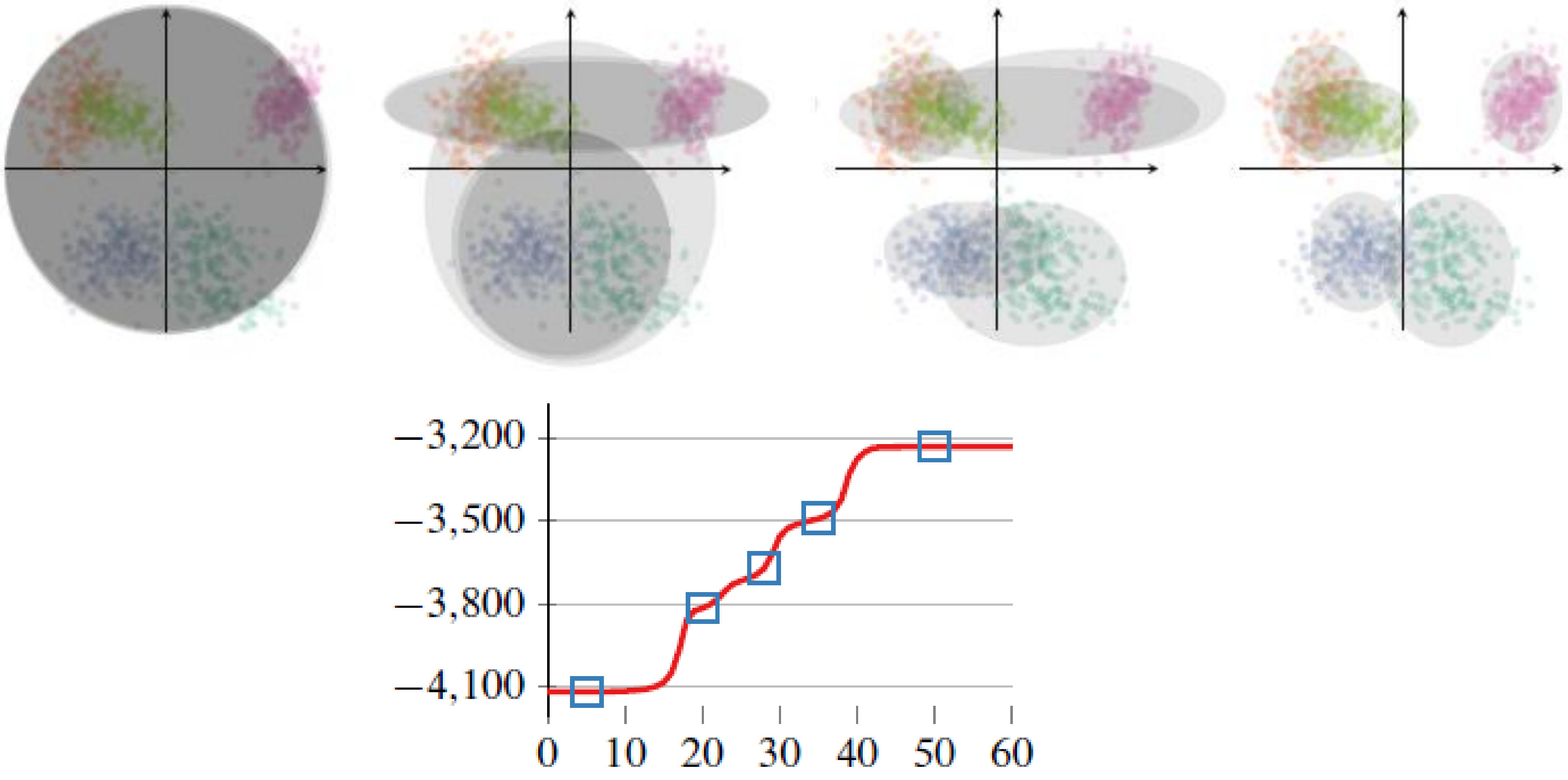
$$+ \sum_{i=1}^{n} (\mathbb{E}_q[\log p(c_i); \phi_i] + \mathbb{E}_q[\log p(x_i|c_i, \boldsymbol{\mu}); \phi_i, \mathbf{m}, \mathbf{s}^2])$$

$$- \sum_{i=1}^{n} \mathbb{E}_q[\log q(c_i; \phi)] - \sum_{k=1}^{K} \mathbb{E}_q[\log q(\mu_k; m_k, s_k^2)]$$

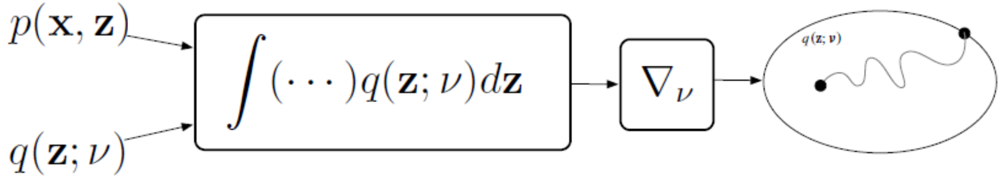
CAVI. Mixture of gaussians

Input: Data $\boldsymbol{x} = x_{1:n}$, number of components K, prior σ^2
Result: Gaussian distributions $q(\mu_k; m_k, s_k^2)$ and
categorial distributions $q(z_i, \phi_i)$
Initialize: Variational parameters $\mathbf{m} = m_{1:K}$, $\mathbf{s}^2 = s_{1:K}^2$,
$\boldsymbol{\phi} = \phi_{1:n}$
while *ELBO has not converged* **do**
  for $i \in \{1..n\}$ **do**
    for $k \in \{1..K\}$ **do**
      Set
      $\phi_{i,k} \propto \exp\{\mathbb{E}_q[\mu_k; m_k, s_k^2]x_i - \mathbb{E}_q[\mu_k^2; m_k, s_k^2]/2\}$
    end
  end
  for $k \in \{1, ...K\}$ **do**
    Set $m_k = \frac{\sum_{i=1}^n \phi_{i,k} x_i}{1/\sigma^2 + \sum_{i=1}^n \phi_{i,k}}$
    Set $s_k^2 = \frac{1}{1/\sigma^2 + \sum_{i=1}^n \phi_{i,k}}$
  end
  Compute ELBO$(\mathbf{m}, \mathbf{s}^2, \boldsymbol{\phi})$
end

CAVI. Mixture of gaussians



VI recipe



Formulate model $p(\mathbf{z}, \mathbf{x})$

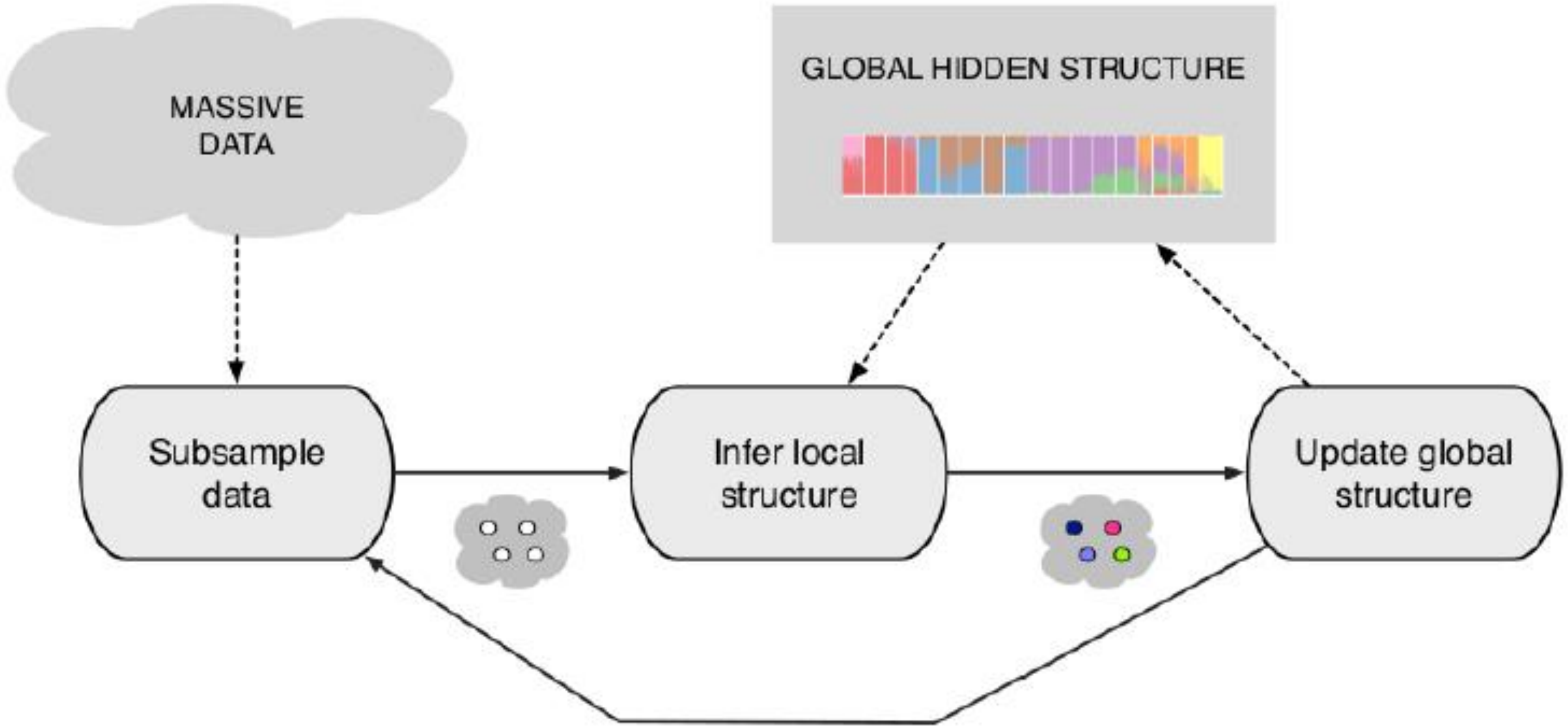
Choose variational app $q(\mathbf{z}; \boldsymbol{\nu})$

Write and assess ELBO $\mathcal{L}(\boldsymbol{\nu}) = \mathbb{E}_{q(\mathbf{z};\boldsymbol{\nu})}[\log p(\mathbf{x}, \mathbf{z}) - \log q(\mathbf{z}; \boldsymbol{\nu})]$

Compute gradient $\nabla_{\boldsymbol{\nu}} \mathcal{L}(\boldsymbol{\nu})$

Optimize $\boldsymbol{\nu}_{t+1} = \boldsymbol{\nu}_t + \rho_t \nabla_{\boldsymbol{\nu}} \mathcal{L}$

Stochastic VI



Replace gradient by cheaper estimation

Stochastic VI

- Stochastic optimization (Robbins, Monro)

Iteration $$\nu_{t+1} = \nu_t + \rho_t \hat{\nabla}_\nu \mathscr{L}(\nu_t)$$

Gradient unbiased estimate $$\mathbb{E}\left[\hat{\nabla}_\nu \mathscr{L}(\nu)\right] = \nabla_\nu \mathscr{L}(\nu)$$

- ELBO natural gradient

$$\nabla_\lambda^{\text{nat}} \mathscr{L}(\lambda) = \left(\alpha + \textstyle\sum_{i=1}^n \mathbb{E}_{\phi_i^*}[t(Z_i, x_i)]\right) - \lambda$$

- Noisy natural gradient

$$j \sim \text{Uniform}(1, \ldots, n)$$

Cheap and unbiased!!! $$\hat{\nabla}_\lambda^{\text{nat}} \mathscr{L}(\lambda) = \alpha + n\mathbb{E}_{\phi_j^*}[t(Z_j, x_j)] - \lambda$$

Stochastic VI

Input: Model $p(\mathbf{x}, \mathbf{z})$, data $\mathbf{x}$, and a sequence of step sizes ϵ_t

Result: Global variational densitioes $q_\lambda(\beta)$

Initialize: Variational parameters λ_0

while *True* **do**

  Randomy pick a data point, $t \sim Unif(1, ..., n)$

  Optimize its local variational parameters $\phi_t^\star = \mathbb{E}_\lambda[\eta(\beta, x_t)]$

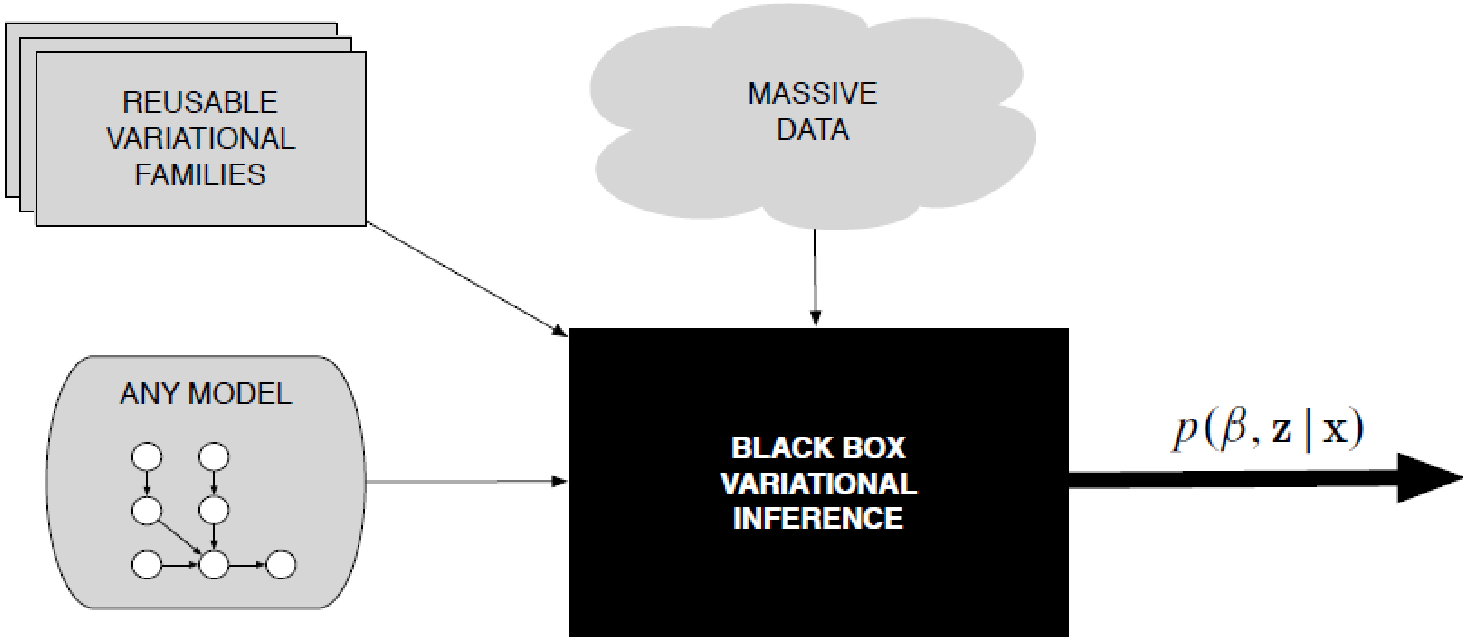
  Repeat the picked data point x_t n times, computer coordinate update $\tilde{\lambda} = \mathbb{E}_\phi[\tilde{\alpha}]$

  Update the global variational parameter: $\lambda_t = (1 - \epsilon_t)\lambda_t + \epsilon_t \tilde{\lambda}_t$

end

VI. Additional info

Black box VI



Beyond mean fields

Neural nets to approximate distributions

ADVI (Automatic variational inference)

VI. Additional info

- Systems with VI

Venture, WebPPL, Edward, **Stan**, PyMC3, Infer.net, Anglican

https://mc-stan.org/docs/cmdstan-guide/variational-inference-algorithm-advi.html

- Differentiation tools

Theano, Torch, Tensorflow, **Stan**, Caffe

SG-MCMC

MCMC in large scale problems

- Big data

Gibbs is latent variables, pass through all data (and anyways not always usable)

MH and HMC both require likelihood or log-likelihood, pass through al data

- Big number of parameters (eg deep models)

If implementable, many Gibbs steps per cycle (partly alleviated by joint samples)

MH and HMC a very long chain of computations to evaluate the factor related with posterior or log-posterior… and the gradient

- Parallelise on subsets of data and then merge (eg Scott et al, 2016) does not work well in general
- Big data and big number of parameters…. We are in deep sh****t!!!

Langevin based SG-MCMC

Sampling from the posterior $\pi(\theta) \propto \exp\{-U(\theta)\}$

e.g. with iid data $\pi(\theta) \propto p(\theta) \prod_{i=1}^{N} f(y_i|\theta)$

$$U(\theta) = \sum_{i=1}^{N} U_i(\theta)$$

$$U_i(\theta) = -\log f(y_i|\theta) - (1/N) \log p(\theta)$$

i.e. -log posterior (no normalizada)

Langevin diffusion

$$d\boldsymbol{\theta}(t) = -\frac{1}{2}\nabla U(\boldsymbol{\theta}(t))dt + dB_t$$

Converges to target if B_t brownian motion. Dynamics (with Z standard Gaussian)

$$\boldsymbol{\theta}(t+h) \approx \boldsymbol{\theta}(t) - \frac{h}{2}\nabla U(\boldsymbol{\theta}(t)) + \sqrt{h}\mathbf{Z},$$

Stochastic Gradient Langevin dynamics

$$\hat{\nabla} U(\boldsymbol{\theta})^{(n)} = \frac{N}{n}\sum_{i \in \mathcal{S}_n} \nabla U_i(\boldsymbol{\theta})$$

SGLD (Welling and Teh)

Algorithm 1: SGLD

Input: θ_0, $\{h_0, \ldots, h_K\}$.
for $k \in 1, \ldots, K$ **do**
  Draw $\mathcal{S}_n \subset \{1, \ldots, N\}$ without replacement
  Estimate $\hat{\nabla} U(\theta)^{(n)}$ using (4)
  Draw $\xi_k \sim N(0, h_k I)$
  Update $\theta_{k+1} \leftarrow \theta_k - \frac{h_k}{2} \hat{\nabla} U(\theta_k)^{(n)} + \xi_k$
end

Tricks to control the variance
Theory ensuring convergence

General framework (Ma et al, 2015)

General state ζ

Langevin $\zeta = \theta$

Hamiltonian $\zeta = (\theta, \rho)$

$$\mathrm{d}\zeta = \frac{1}{2}\mathrm{b}(\zeta)\mathrm{d}t + \sqrt{\mathrm{D}(\zeta)}\mathrm{d}B_t$$

The choice $\mathrm{Q}^\top = -\mathrm{Q}$

$$\mathrm{b}(\zeta) = -\left[\mathrm{D}(\zeta) + \mathrm{Q}(\zeta)\right]\nabla H(\zeta) + \Gamma(\zeta) \text{ and } \Gamma_i(\zeta) = \sum_{j=1}^{d} \frac{\partial}{\partial \zeta_j}(\mathrm{D}_{ij}(\zeta) + \mathrm{Q}_{ij}(\zeta))$$

makes it converge to

$$\exp\{-H(\zeta)\}$$

Discretisation

$$\zeta_{t+h} \approx \zeta_t - \frac{h}{2}\left[(\mathbf{D}(\zeta_t) + \mathbf{Q}(\zeta_t))\nabla H(\zeta_t) + \Gamma(\zeta_t)\right] + \sqrt{h}\mathbf{Z}, \quad t \geq 0$$

Replace gradient with stochastic gradient as before

Algorithm	ζ	$H(\zeta)$	$\mathbf{D}(\zeta)$	$\mathbf{Q}(\zeta)$
SGLD	$\boldsymbol{\theta}$	$U(\boldsymbol{\theta})$	$\mathbf{I}$	$\mathbf{0}$
SG-RLD	$\boldsymbol{\theta}$	$U(\boldsymbol{\theta})$	$G(\boldsymbol{\theta})^{-1}$	$\mathbf{0}$
SG-HMC	$(\boldsymbol{\theta}, \boldsymbol{\rho})$	$U(\boldsymbol{\theta}) + \frac{1}{2}\boldsymbol{\rho}^\top\boldsymbol{\rho}$	$\begin{pmatrix} 0 & 0 \\ 0 & \mathbf{C} \end{pmatrix}$	$\begin{pmatrix} 0 & -\mathbf{I} \\ \mathbf{I} & 0 \end{pmatrix}$
SG-RHMC	$(\boldsymbol{\theta}, \boldsymbol{\rho})$	$U(\boldsymbol{\theta}) + \frac{1}{2}\boldsymbol{\rho}^\top\boldsymbol{\rho}$	$\begin{pmatrix} 0 & 0 \\ 0 & G(\boldsymbol{\theta})^{-1} \end{pmatrix}$	$\begin{pmatrix} 0 & -G(\boldsymbol{\theta})^{-1/2} \\ G(\boldsymbol{\theta})^{-1/2} & 0 \end{pmatrix}$
SG-NHT	$(\boldsymbol{\theta}, \boldsymbol{\rho}, \eta)$	$U(\boldsymbol{\theta}) + \frac{1}{2}\boldsymbol{\rho}^\top\boldsymbol{\rho}$ $+\frac{1}{2d}(\eta - A)^2$	$\begin{pmatrix} 0 & 0 & 0 \\ 0 & A \cdot \mathbf{I} & 0 \\ 0 & 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & -\mathbf{I} & 0 \\ \mathbf{I} & 0 & \boldsymbol{\rho}^\top/d \\ 0 & -\boldsymbol{\rho}^\top/d & 0 \end{pmatrix}$

*

Code

https://github.com/chris-nemeth/sgmcmc-review-paper

R package   sgmcmc   (Edward in Python)

Faster than Stan in large problems

VI + MCMC

Variationally inferred sampling

Repeat

- Evolve the variational distribution a few SG-MCMC iterations
- Find new optimal variational distribution

Method	MNIST	fMNIST
Results from [24]		
UIVI	−94.09	−110.72
SIVI	−97.77	−121.53
VAE	−98.29	−126.73
Results from [29]		
VCD	−95.86	−117.65
HMC-DLGM	−96.23	−117.74
This paper		
VIS-5-10	**−82.74 ± 0.19**	**−105.08 ± 0.34**
VIS-0-10	−96.16 ± 0.17	−120.53 ± 0.59
[illegible]	[illegible]	[illegible]

https://github.com/vicgalle/vis

INLA

INLA's ancestors (Recall chapter 2)

Laplace (1774)

Tierney, Kadane (1986), Tierney, Kadane, Kass (1989)

Rue, Martino, Chopin (2009) **Integrated nested Laplace approximation**

Model

$$\mathbf{y}|\mathbf{x},\phi \sim \prod_{i=1}^{n} p(y_i|\eta_i(\mathbf{x}),\phi) \qquad \mathbf{x}|\phi \sim \mathcal{N}(0, Q^{-1}(\phi)) \qquad \phi \sim p(\phi),$$

y observation

η_i linear predictor which depends on latent Gaussian variables x

$Q(\phi)$ precision matrix

Approximate $p(\phi_j|\mathbf{y}),\ j = 1,2,..,m,$ $p(x_k|\mathbf{y}),\ k = 1,2,..,K.$

Approach

$$p(\phi|\mathbf{y}) = \frac{p(\mathbf{x},\phi|\mathbf{y})}{p(\mathbf{x}|\phi,\mathbf{y})} \propto \frac{p(\mathbf{x},\phi,\mathbf{y})}{p(\mathbf{x}|\phi,\mathbf{y})} = \frac{p(\mathbf{y}|\mathbf{x},\phi)p(\mathbf{x}|\phi)p(\phi)}{p(\mathbf{x}|\phi,\mathbf{y})},$$

approximated by

$$\widetilde{p}(\phi|\mathbf{y}) \propto \frac{p(\mathbf{y}|\widehat{\mathbf{x}}(\phi),\phi)p(\widehat{\mathbf{x}}(\phi)|\phi)p(\phi)}{p_G(\widehat{\mathbf{x}}(\phi)|\phi,\mathbf{y})}.$$

$$\widetilde{p}(\phi|\mathbf{y}) \propto p(\mathbf{y}|\widehat{\mathbf{x}}(\phi),\phi)p(\widehat{\mathbf{x}}(\phi)|\phi)p(\phi)\left|\widehat{\Sigma}(\phi)\right|^{1/2}$$

Laplace approximation of marginal (Chapter 2)

$$\widetilde{p}(x_k|\mathbf{y}) = \int_\Theta \widetilde{p}(x_k|\phi,\mathbf{y})\widetilde{p}(\phi|\mathbf{y})d\phi$$

$\widehat{\mathbf{x}}_{-k}(\phi,x_k)$ is the mode of $p(\mathbf{x}_{-k},x_k,\phi,\mathbf{y})$

$$\widetilde{p}(x_k|\phi,\mathbf{y}) \propto p(\mathbf{y}|\widehat{\mathbf{x}}_{-k}(\phi,x_k),\phi)p(\widehat{\mathbf{x}}_{-k}(\phi,x_k)|\phi)p(\phi)\left|\widehat{\Sigma}_{-k}(\phi,x_k)\right|^{1/2}$$

Laplace approximation

$\widehat{\Sigma}_{-k}(\phi,x_k)$ is the inverse of the Hessian of $-\log p(\mathbf{x}_{-k},x_k,\phi,\mathbf{y})$

$\widehat{\mathbf{x}}_{-k}(\phi,x_k)$.

$$\widetilde{p}(\phi_j|\mathbf{y}) = \int_{\Theta_{-j}} \widetilde{p}(\phi|\mathbf{y})d\phi_{-j}$$

https://www.r-inla.org/   R package for Gaussian latent models

Gómez-Rubio (2020) For pointers to extensions to other models

ABC

Basic concept

Approximate $p(\theta|\mathrm{y})$ when $p(\mathrm{y}|\theta)$ cannot be evaluated but can be simulated

Algorithm 1 Vanilla Accept/Reject ABC Algorithm

for $i = 1, \ldots, M$ do
  Simulate θ^i, $i = 1, 2, ..., M$, from $p(\theta)$, and artificial data z^i from $p(\cdot|\theta^i)$;
  Accept θ^i if $d\{\mathrm{z}^i, \mathrm{y}\} \leq \varepsilon$
end for

Draw from

$$p_\varepsilon(\theta|\mathbf{y}) = \frac{\int_{\mathcal{X}} p_\varepsilon(\theta, \mathbf{z}|\mathbf{y})\mathrm{d}\mathbf{z}}{\int_\Theta \int_{\mathcal{X}} p(\theta, \mathbf{z}|\mathbf{y})\mathrm{d}\mathbf{z}\mathrm{d}\theta}, \quad p_\varepsilon(\theta, \mathbf{z}|\mathbf{y}) = \mathbb{I}\left[d\{\mathbf{y}, \mathbf{z}\} \leq \varepsilon\right] p(\mathbf{z}|\theta)p(\theta),$$

ABC with summary statistics

Algorithm 2 Accept/Reject ABC Algorithm Based on Summary Statistics

---

for $i = 1, \ldots, M$ **do**
  Simulate $\boldsymbol{\theta}^i$, $i = 1, 2, ..., M$, from $p(\boldsymbol{\theta})$, and artificial data $\mathbf{z}^i$ from $p(\cdot|\boldsymbol{\theta}^i)$;
  Accept $\boldsymbol{\theta}^i$ if $d\{\eta(\mathbf{z}^i), \eta(\mathbf{y})\} \leq \varepsilon$.
end for

ABC-MCMC

R packages abc and abc.data

https://cran.r-project.org/web/packages/abc/vignettes/abcvignette.pdf

Final comments

Large scale Bayes

Still a pretty open area

Combinations of VI and SG-MCMC