Inference

Michele Caprio

Department of Computer Science, University of Manchester
Manchester Centre for AI Fundamentals

COMP 64101 – Reasoning and Learning under Uncertainty
Lecture 2

Topics: Variational Inference (VI)

"I know it's Tuesday. It's a good day for math!"

Max Mintz

- What is VI? (Murphy, 2023, § 10.1)
- Gradient-based VI (Murphy, 2023, § 10.2)
  - Blackbox VI (Murphy, 2023, § 10.2.3)
- Coordinate Ascent VI (Murphy, 2023, § 10.3)
  - Variational Bayes (Murphy, 2023, § 10.3.3)
- More VI Goodies (Murphy, 2023, § 10.4 – 10.7)

Topics: Monte Carlo (MC) Methods

"I know it's Monday. It's a good day for math!"

Max Mintz

- MC integration (Murphy, 2023, § 11.2)
- Importance Sampling (Murphy, 2023, § 11.5)
- Rao-Blackwellization (Murphy, 2023, § 11.6.2)

Topics: Markov Chain Monte Carlo (MCMC)

"I know it's Monday. It's a good day for math!"

Max Mintz

- Metropolis-Hastings Algorithm (Murphy, 2023, § 12.2)
- Gibbs Sampling (Murphy, 2023, § 12.3)
- Hamiltonian MC (Murphy, 2023, § 12.5.3 – 12.5.4)
- MCMC Convergence (Murphy, 2023, § 12.6)

What is VI?

- Variational Inference (VI) reduces posterior inference to optimization
- Consider a model with unknown (latent) variables z, known variables x, and fixed parameters θ

What is VI?

- Variational Inference (VI) reduces posterior inference to optimization
- Consider a model with unknown (latent) variables z, known variables x, and fixed parameters θ
  - Prior: $p(\theta)$
  - Likelihood: $p_\theta(x \mid z)$
  - (Unnormalized) Joint: $p_\theta(x, z) = p_\theta(x \mid z)p_\theta(z)$
  - Posterior:

$$p_\theta(z \mid x) = \frac{p_\theta(x, z)}{\underbrace{p_\theta(x)}_{=\int p_\theta(x,z)\mathrm{d}z,\ \text{intractable}}}$$

- Need to approximate the posterior!

What is VI?

- We approximate the posterior by the distribution $q(z)$ that minimizes the loss

$$q^\star = \arg\min_{q \in \mathcal{Q}} D_{KL}(q(z) \| p_\theta(z \mid x)) \tag{1}$$

- In practice, we pick a parametric family $\mathcal{Q}$; call ψ its variational parameters
- (1) can be rewritten as

$$\begin{aligned}
\psi^\star &= \arg\min_{\psi} D_{KL}(q_\psi(z) \| p_\theta(z \mid x)) \\
&= \arg\min_{\psi} \mathbb{E}_{q_\psi(z)} \left[\log q_\psi(z) - \log \left(\frac{p_\theta(x \mid z) p_\theta(z)}{p_\theta(x)} \right) \right] \\
&= \arg\min_{\psi} \underbrace{\mathbb{E}_{q_\psi(z)} \left[\log q_\psi(z) - \log p_\theta(x \mid z) - \log p_\theta(z) \right]}_{=: \mathcal{L}(\theta, \psi \mid x)} + \underbrace{\log p_\theta(x)}_{\text{intract. but } \perp\!\!\!\perp \psi}
\end{aligned}$$

ELBO

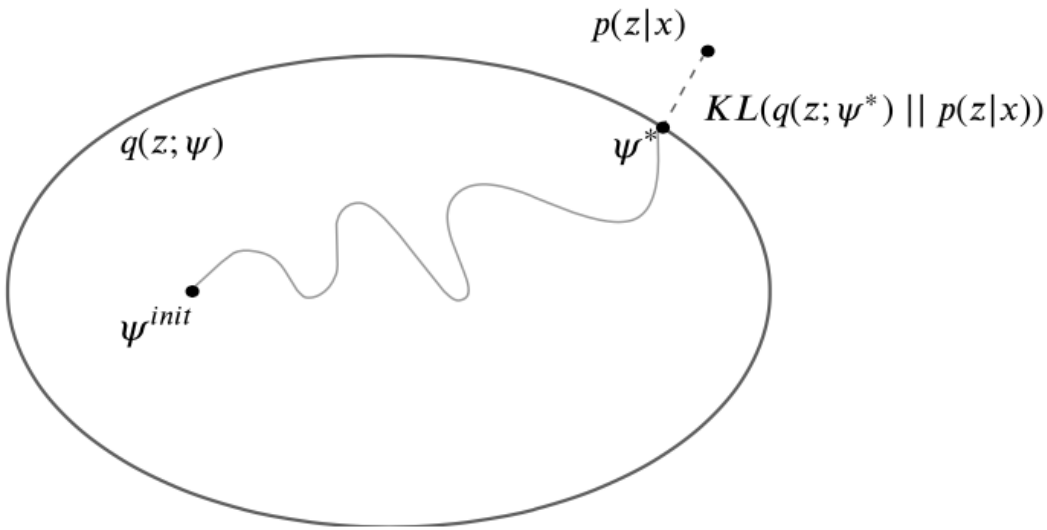
- So, we want to minimize the variational free energy

$$\mathcal{L}(\theta, \psi \mid x) := \mathbb{E}_{q_\psi(z)} \left[\log q_\psi(z) - \log p_\theta(x, z) \right]$$

- Or equivalently, maximize the evidence lower bound (ELBO)

$$\mathscr{L}(\theta, \psi \mid x) := -\mathcal{L}(\theta, \psi \mid x) = \mathbb{E}_{q_\psi(z)} \left[-\log q_\psi(z) + \log p_\theta(x, z) \right]$$

  - $\mathscr{L}(\theta, \psi \mid x) \leq \log p_\theta(x)$, the evidence



ELBO Interpretation

- $\underbrace{\mathscr{L}(\theta, \psi \mid x)}_{\text{ELBO}} = \underbrace{\mathbb{E}_{q_\psi(z)}\left[\log p_\theta(x, z)\right]}_{\text{expected log joint}} + \underbrace{H(q_\psi(z))}_{\text{entropy}}$
  - Second term encourages the posterior to be maximum entropy; first encourages it to be a joint MAP configuration

ELBO Interpretation

- $\underbrace{\mathscr{L}(\theta, \psi \mid x)}_{\text{ELBO}} = \underbrace{\mathbb{E}_{q_\psi(z)}\left[\log p_\theta(x, z)\right]}_{\text{expected log joint}} + \underbrace{H(q_\psi(z))}_{\text{entropy}}$
  - Second term encourages the posterior to be maximum entropy; first encourages it to be a joint MAP configuration
- $$\underbrace{\mathscr{L}(\theta, \psi \mid x)}_{\text{ELBO}} = \mathbb{E}_{q_\psi(z)}\left[\log p_\theta(x \mid z) + \log p_\theta(z) - \log q_\psi(z)\right]$$
$$= \underbrace{\mathbb{E}_{q_\psi(z)}\left[\log p_\theta(x \mid z)\right]}_{\text{expected log likelihood}} - \underbrace{D_{KL}(q_\psi(z) \| p_\theta(z))}_{\text{KL from posterior to prior}}$$
  - The KL term acts like a regularizer, preventing the posterior from diverging too much from the prior

Choosing the Form of the Variational Posterior

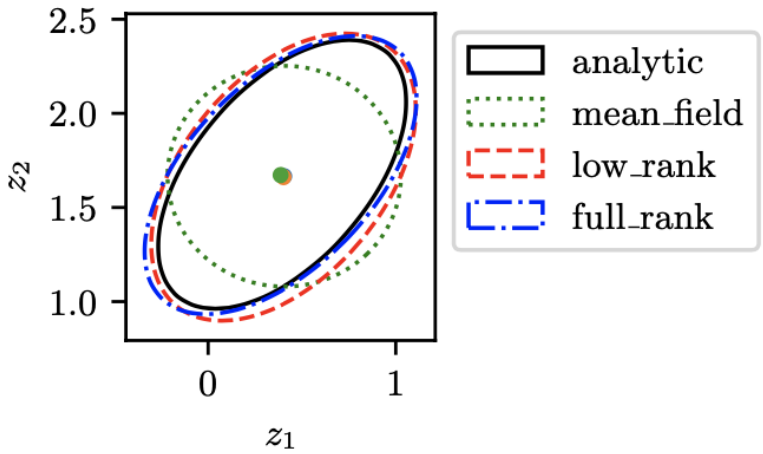
- Fixed-form VI: pick a convenient functional form (e.g. multivariate Gaussian) and optimize ELBO using gradient-based methods
- Free-form VI: make the mean field assumption that the posterior factorizes as $q_\psi(z) = \prod_{j=1}^{J} q_j(z_j)$
  - $q_j(z_j) \equiv q_{\psi_j}(z_j)$ is the posterior over the j-th group of variables
  - No need to specify the functional form for each q_j
  - Optimal distributional form derived by maximizing the ELBO wrt each group of variational parameters one at a time
    - In a coordinate ascent manner

Multivariate Normal Example

- Unknown mean, known covariance matrix
- Exact posterior can be computed analytically: Multivariate Normal conjugate prior

Multivariate Normal Example

- Unknown mean, known covariance matrix
- Exact posterior can be computed analytically: Multivariate Normal conjugate prior



- If q uses a full covariance matrix, it matches the exact posterior; intractable in high dimensions
- If q uses a diagonal covariance matrix (mean field approximation): the approximation is over confident
  - Mode-seeking nature of minimizing $D_{KL}(q\|p)$
- If q uses a rank-1 plus diagonal approximation: better approximation; can be computed efficiently

Parameter Estimation

- If the model parameters θ are not known, can we estimate them?
- Maximizing the likelihood of the dataset $D = \{x_n\}_{n=1}^N$ in the presence of latent variables $\{z_n\}_{n=1}^N$,

$$\log p(D, \{z_n\} \mid \theta) = \prod_{n=1}^{N} p(z_n \mid \theta) p(x_n \mid z_n, \theta)$$

- Since the latent variables z_n are hidden, we must marginalize them out to get the log marginal likelihood

$$\log p(x_n \mid \theta) = \log \left[\underbrace{\int p(x_n \mid z_n, \theta) p(z_n \mid \theta) \mathrm{d}z_n}_{\text{intractable: normalization constant of the exact posterior}} \right]$$

Parameter Estimation

- ELBO is a lower bound on this, $\mathscr{L}(\theta, \psi_n \mid x_n) \leq \log p(x_n \mid \theta)$
- Optimize the model parameters by maximizing

$$\mathscr{L}(\theta, \{\psi_n\} \mid D) := \sum_{n=1}^{N} \mathscr{L}(\theta, \psi_n \mid x_n) \leq \log p(D \mid \theta)$$

- Variational EM algorithm (Murphy, 2023, § 6.5.6.1): alternate between
  - Maximizing the ELBO wrt the variational parameters $\{\psi_n\}$ in the E step, to give us $q_{\psi_n}(z_n)$
  - Maximizing the ELBO (using the new ψ_n) wrt the model parameters θ in the M step
  - Faster alternative: Stochastic VI (Murphy, 2023, § 10.1.4)

Gradient-Based VI

- Choose convenient form for $q_\psi(z)$, e.g. Gaussian for continuous z, or product of Categoricals for discrete z
- Optimize the ELBO using gradient based methods
- Stochastic Gradient Descend (SGD): Suppose we have a way of computing an unbiased estimate g_t of the gradient of the ELBO, i.e.

$$\mathbb{E}(g_t) = \nabla_\theta \mathscr{L}(\theta, \psi \mid x)|_{\theta_t}$$

- Then, we can use it in a gradient descent procedure

$$\theta_{t+1} = \theta_t - \eta_t g_t,$$

where η_t is the learning rate or step size (Murphy, 2023, § 6.3.1.1)

- Same for the gradient wrt ψ

Gradient wrt Generative Parameters $\nabla_\theta \mathscr{L}(\theta, \psi \mid x)$

- Easy to compute: push gradients inside the expectation, and use a single Monte Carlo sample

$$\begin{aligned}\nabla_\theta \mathscr{L}(\theta, \psi \mid x) &= \nabla_\theta \mathbb{E}_{q_\psi(z)} \left[-\log q_\psi(z) + \log p_\theta(x, z) \right] \\ &= \mathbb{E}_{q_\psi(z)} \left[\nabla_\theta (-\log q_\psi(z) + \log p_\theta(x, z)) \right] \\ &\approx \nabla_\theta \log p_\theta(x, z^s), \quad z^s \sim q_\psi(z)\end{aligned}$$

  - This is an unbiased estimate of the gradient, so can be used with SGD

Gradient wrt Inference Parameters $\nabla_\psi \mathscr{L}(\theta, \psi \mid x)$

- Harder to compute: we cannot push gradients inside the expectation. Reparametrized VI (RVI)
- Rewrite $z \sim q_\psi(z)$ as differentiable (and invertible) transformation h of $\epsilon \sim p(\epsilon)$, does not depend on ψ, i.e. $z = h(\psi, x, \epsilon)$
  - E.g. $z \sim \mathcal{N}(\mu, \text{diag}(\sigma)) \iff z = \mu + \epsilon \odot \sigma, \quad \epsilon \sim \mathcal{N}(0, I)$

Gradient wrt Inference Parameters $\nabla_\psi \mathscr{L}(\theta, \psi \mid x)$

- Harder to compute: we cannot push gradients inside the expectation. Reparametrized VI (RVI)
- Rewrite $z \sim q_\psi(z)$ as differentiable (and invertible) transformation h of $\epsilon \sim p(\epsilon)$, does not depend on ψ, i.e. $z = h(\psi, x, \epsilon)$
  - E.g. $z \sim \mathcal{N}(\mu, \text{diag}(\sigma)) \iff z = \mu + \epsilon \odot \sigma, \quad \epsilon \sim \mathcal{N}(0, I)$
- Then, $\mathbb{E}_{q_\psi(z)}[f(z)] = \mathbb{E}_{p(\epsilon)}[f(z)]$, $z = h(\psi, x, \epsilon)$
  - Where $f(z) \equiv f_{\theta,\psi}(z) = \log p_\theta(x, z) - \log q_\psi(z)$

Gradient wrt Inference Parameters $\nabla_\psi \mathscr{L}(\theta, \psi \mid x)$

- Hence

$$\nabla_\psi \mathscr{L}(\theta, \psi \mid x) = \nabla_\psi \mathbb{E}_{q_\psi(z)}[f(z)] = \nabla_\psi \mathbb{E}_{p(\epsilon)}[f(z)] = \mathbb{E}_{p(\epsilon)}[\nabla_\psi f(z)]$$
$$\approx \mathbb{E}_{p(\epsilon)}[\nabla_\psi f(h(\psi, x, \epsilon^s))], \quad \epsilon^s \sim p(\epsilon)$$

- This is an unbiased estimate of the gradient, so can be used with SGD

Gradient wrt Inference Parameters $\nabla_\psi \mathscr{L}(\theta, \psi \mid x)$

- Hence

$$\nabla_\psi \mathscr{L}(\theta, \psi \mid x) = \nabla_\psi \mathbb{E}_{q_\psi(z)}[f(z)] = \nabla_\psi \mathbb{E}_{p(\epsilon)}[f(z)] = \mathbb{E}_{p(\epsilon)}[\nabla_\psi f(z)]$$
$$\approx \mathbb{E}_{p(\epsilon)}[\nabla_\psi f(h(\psi, x, \epsilon^s))], \quad \epsilon^s \sim p(\epsilon)$$

- This is an unbiased estimate of the gradient, so can be used with SGD
- Since we are now working with ϵ, need to use the change of variables formula

$$\log q_\psi(z) = \log p(\epsilon) - \log \left| \det \left(\frac{\partial z}{\partial \epsilon} \right) \right|$$

  - $\partial z / \partial \epsilon$ is the Jacobian, i.e. the matrix of all its first-order partial derivatives
  - We design transformation $z = h(\psi, x, \epsilon)$ s.t. this Jacobian is tractable to compute
- Examples: (Murphy, 2023, § 10.2.1)

Blackbox VI (BBVI)

- Suppose we can evaluate $\ell(\psi, z) := -\log q_\psi(z) + \log p_\theta(x, z)$ pointwise, but may not be able to take gradients of it
- To estimate the gradient of the ELBO, we use the score function estimator. Write the ELBO as

$$\mathscr{L}(\theta, \psi \mid x) = \mathbb{E}_{q_\psi(z)}[\ell(\psi, z)] = \mathbb{E}_{q_\psi(z)}[-\log q_\psi(z) + \log p_\theta(x, z)]$$

- From (Murphy, 2023, Eq. (6.58)),

$$\nabla_\psi \mathscr{L}(\theta, \psi \mid x) = \mathbb{E}_{q_\psi(z)}\left[\ell(\psi, z) \nabla_\psi \log q_\psi(z)\right]$$

Blackbox VI (BBVI)

- Suppose we can evaluate $\ell(\psi, z) := -\log q_\psi(z) + \log p_\theta(x, z)$ pointwise, but may not be able to take gradients of it
- To estimate the gradient of the ELBO, we use the score function estimator. Write the ELBO as

$$\mathscr{L}(\theta, \psi \mid x) = \mathbb{E}_{q_\psi(z)}[\ell(\psi, z)] = \mathbb{E}_{q_\psi(z)}[-\log q_\psi(z) + \log p_\theta(x, z)]$$

- From (Murphy, 2023, Eq. (6.58)),

$$\nabla_\psi \mathscr{L}(\theta, \psi \mid x) = \mathbb{E}_{q_\psi(z)}\left[\ell(\psi, z) \nabla_\psi \log q_\psi(z)\right]$$

- Compute a Monte Carlo approximation to this

$$\widehat{\nabla_\psi \mathscr{L}(\theta, \psi \mid x)}\Big|_{\psi_t} = \frac{1}{S} \sum_{s=1}^{S} \ell(\psi, z^s) \nabla_\psi \log q_\psi(z^s) \Big|_{\psi_t}, \quad z^1, \ldots, z^S \sim q_\psi(z)$$

- This is an unbiased estimate of the gradient, so can be used with SGD

Coordinate Ascent VI (CAVI)

- Mean field approximation in VI: assume that all latent variables are independent, $q_\psi(z) = \prod_{j=1}^J q_j(z_j) \equiv \prod_{j=1}^J q_{\psi_j}(z_j)$
  - J is the number of hidden variables
  - ψ_j are the variational parameters for the j-th distribution
- From previous slides, the ELBO becomes

$$\mathscr{L}(\theta, \psi \mid x) = \int q_\psi(z) \log p_\theta(x, z) \mathrm{d}z + \sum_{j=1}^J H(q_j)$$

- First term decomposes according to Markov properties of the model
- This allows us to use a coordinate ascent optimization scheme to estimate each ψ_j
  - Optimize $\mathscr{L}(\theta, \psi \mid x)$ wrt each q_j, one at a time, keeping others fixed
  - Convergence is guaranteed since the bound is concave wrt each of the factors q_j
  - Functional form of the q_j's need not be specified in advance: determined by the form of the log joint
- CAVI algorithm: (Murphy, 2023, Algorithm 10.4)

Variational Bayes

- Bayesian modeling: treat the parameters θ as latent variables
- Goal: approximate parameter posterior $p(\theta \mid D) \propto p(\theta)p(D \mid \theta)$
- No latent variables except for the shared global parameters, $p(\theta, D) = p(\theta) \prod_{n=1}^{N} p(x_n \mid \theta)$
- Fit the variational posterior by maximizing the ELBO

$$\mathscr{L}(\theta, \psi \mid D) = \mathbb{E}_{q_{\psi_\theta}(\theta)}[\log p(\theta, D)] + H(q_{\psi_\theta}(\theta))$$

- Assume the variational posterior factorizes as $q_{\psi_\theta}(\theta) = \prod_{j=1}^{J} q_{\psi_{\theta_j}}(\theta_j)$
- Update each ψ_{θ_j} via CAVI (Murphy, 2023, Algorithm 10.4)
  - Examples: (Murphy, 2023, § 10.3.4)
  - Of possible interest: Variational Bayes EM (Murphy, 2023, § 10.3.5 – 10.3.6)

More VI Goodies

- Improve tightness of ELBO lower bound – reducing KL of our posterior approximation – if we use more flexible posterior families
  - Optimizing within more flexible families may be slower, and can incur statistical error if the sample size is low (Murphy, 2023, § 10.4)

More VI Goodies

- Improve tightness of ELBO lower bound – reducing KL of our posterior approximation – if we use more flexible posterior families
  - Optimizing within more flexible families may be slower, and can incur statistical error if the sample size is low (Murphy, 2023, § 10.4)
- Improve quality of posterior approximation: optimize q wrt a bound that is a tighter approximation to the log marginal likelihood compared to ELBO (Murphy, 2023, § 10.5)

More VI Goodies

- Problem with lower bound maximization (standard VI): minimizing $D_{KL}(q\|p)$, which induces zero-forcing behavior
- This means that $q(z \mid x)$ tends to be too compact (over-confident), to avoid the situation in which $q(z \mid x) > 0$ but $p(z \mid x) = 0$,
  - Would incur infinite KL penalty
- Zero-forcing can be desirable for multi-modal posteriors (e.g. mixture models); not so reasonable for unimodal posteriors
- Avoid this problem: minimize $D_{KL}(p\|q)$, which is zero-avoiding
- Result in broad posteriors, which avoids overconfidence
- Expectation propagation: local approximation to $D_{KL}(p\|q)$ (Murphy, 2023, § 10.7)

VI for Imprecise Probabilities

Cella and Martin (2024) are the first who study VI in the context of Inferential Models. If interested, we can work together on this!

Monte Carlo Methods

- Monte Carlo methods are a stochastic approach to solving numerical integration problems

Monte Carlo Integration

- Integration of interest

$$\mathbb{E}[\varphi(x)] = \int_{\mathbb{R}^n} \varphi(x)\pi(x)\mathrm{d}x$$

  - $\varphi : \mathbb{R}^n \to \mathbb{R}^m$
  - $\pi(x)$ is the target distribution, often a posterior
- Approximate it via Monte Carlo integration

$$\mathbb{E}[\varphi(x)] \approx \frac{1}{N_s} \sum_{n=1}^{N_s} \varphi(x_n), \quad x_1, \ldots, x_{N_s} \sim \pi(x)$$

  - The function is only evaluated in places where there is non-negligible probability
  - Not need to uniformly cover the entire space
- Accuracy is independent of the dimensionality of x; only depends on the number of samples N_s

Sampling from the Target Distribution

- How to sample from $\pi(x)$?
- One possible method: Rejection Sampling (Murphy, 2023, § 11.4)
  - Drawbacks in high dimensions (Murphy, 2023, § 11.4.4)
- A better one: Importance Sampling (Murphy, 2023, § 11.5)
- In general, it is difficult to sample from $\pi(x)$
  - Sample from a proposal distribution $q(x)$
  - Adjust for this inaccuracy by associating weights with each sample
  - We end up with a weighted MC approximation

$$\mathbb{E}[\varphi(x)] \approx \sum_{n=1}^{N_s} W_n \varphi(x_n)$$

Direct Importance Sampling

- We can evaluate the normalized target distribution $\pi(x)$, but we cannot sample from it
- Instead, we sample from the proposal $q(x)$,

$$\mathbb{E}[\varphi(x)] = \int_{\mathbb{R}^n} \varphi(x)\pi(x)\mathrm{d}x = \int_{\mathbb{R}^n} \varphi(x)\frac{\pi(x)}{q(x)}\mathrm{d}x$$

- Require: $\mathrm{supp}(q) \supseteq \mathrm{supp}(\pi)$

Direct Importance Sampling

- Sample $x_1, \ldots, x_{N_s} \sim q(x)$; then,

$$\mathbb{E}[\varphi(x)] \approx \frac{1}{N_s} \sum_{n=1}^{N_s} \underbrace{\frac{\pi(x_n)}{q(x_n)}}_{=:\tilde{w}_n,\ \text{importance weights}} \varphi(x_n)$$

- Unbiased estimate of the true mean $\mathbb{E}[\varphi(x)]$
- From slide 20, $W_n = \frac{\tilde{w}_n}{N_s} = \frac{\pi(x_n)}{N_s \cdot q(x_n)}$

Self-Normalized Importance Sampling

- We can only evaluate the unnormalized target distribution $\tilde{\gamma}(x) = Z\pi(x)$, $Z = \int_{\mathbb{R}^n} \tilde{\gamma}(x)\mathrm{d}x$
- E.g. $\pi(x) = p(\theta \mid y)$, $\tilde{\gamma}(x) = p(\theta, y)$, and $Z = p(y)$
- Self-normalized importance sampling: key idea also approximate normalization constant Z with importance sampling
- Resulting estimate is a ratio of two estimates: biased
  - However as $N_s \to \infty$, the bias goes to zero, under some weak assumptions

Self-Normalized Importance Sampling

- Sample $x_1, \ldots, x_{N_s} \sim q(x)$; then,

$$\mathbb{E}[\varphi(x)] = \int_{\mathbb{R}^n} \varphi(x)\pi(x)\mathrm{d}x = \frac{\int_{\mathbb{R}^n} \varphi(x)\tilde{\gamma}(x)\mathrm{d}x}{\int_{\mathbb{R}^n} \tilde{\gamma}(x)\mathrm{d}x}$$
$$= \frac{\int_{\mathbb{R}^n} \left[\frac{\tilde{\gamma}(x)}{q(x)}\varphi(x)\right] q(x)\mathrm{d}x}{\int_{\mathbb{R}^n} \left[\frac{\tilde{\gamma}(x)}{q(x)}\right] q(x)\mathrm{d}x} \approx \frac{\frac{1}{N_s}\sum_{n=1}^{N_s} \tilde{w}_n \varphi(x_n)}{\frac{1}{N_s}\sum_{n=1}^{N_s} \tilde{w}_n},$$

  where $\tilde{w}_n = \tilde{\gamma}(x_n)/q(x_n)$
- From slide 20, $\mathbb{E}[\varphi(x)] \approx \sum_{n=1}^{N_s} W_n \varphi(x_n)$, and $W_n = \frac{\tilde{w}_n}{\sum_{n'=1}^{N_s} \tilde{w}_{n'}}$
- Approximation to the normalization constant: $Z \approx \frac{1}{N_s}\sum_{n=1}^{N_s} \tilde{w}_n =: \hat{Z}$

Choosing the Proposal

- Performance of importance sampling depends crucially on quality of proposal distribution q
- We need $\text{supp}(q) \supseteq \text{supp}(\pi)$, but we do not want $\text{supp}(q)$ to be "too large"
- $\text{supp}(q)$ should also take into account properties of the target function φ as well
- However, usually the target function φ is unknown or ignored
  - Try to find a "generally useful" approximation to the target
  - One way to come up with a good proposal is to learn one, by optimizing the ELBO

Controlling Monte Carlo Variance

- The standard error in a Monte Carlo estimate is $\mathcal{O}(1/\sqrt{S})$
  - $S \leq N_s$ is the number of (independent) samples
- It may take many samples to reduce the variance to a sufficiently small value
- Rao and Blackwell come to the rescue!

Rao-Blackwellization

- Consider two rv's, Θ and X; we want to estimate $\bar{f} = \mathbb{E}[f(\Theta, X)]$
- Naïve MC approx: $\hat{f}_{MC} = \frac{1}{S}\sum_{s=1}^{S} f(\Theta_s, X_s)$, $(\Theta_1, X_1), \ldots, (\Theta_S, X_S) \sim p(\Theta, X)$ i.i.d.
  - Unbiased, but may have high variance
- Suppose we can analytically marginalize out X, provided we know Θ, i.e., we can tractably compute

$$f_\Theta(\theta_s) = \int_{\mathbb{R}^n} f(\theta_s, X) p(X \mid \theta_s) \mathrm{d}X = \mathbb{E}[f(\Theta, X) \mid \Theta = \theta_s]$$

- Rao-Blackwellized estimator: $\hat{f}_{RB} = \frac{1}{S}\sum_{s=1}^{S} f_\Theta(\theta_s)$, $\theta_1, \ldots, \theta_S \sim p(\Theta)$
- $\hat{f}_{RB}$ is unbiased, and has lower variance than $\hat{f}_{MC}$
  - We are now sampling in a reduced dimensional space

Markov Chain Monte Carlo

- Popular method for sampling from high-dimensional distributions
- Construct a Markov chain on the state space $\mathcal{X}$ whose stationary distribution is the target density $\pi(x)$ of interest, e.g. posterior
- Perform a random walk on the state space so that the fraction of time we spend in each state x is proportional to $\pi(x)$
- Initial samples from the chain do not come from the stationary distribution: discarded
  - Mixing time: time it takes to reach stationarity

Metropolis-Hastings Algorithm

- MH Algorithm: (Murphy, 2023, Algorithm 12.1)
- At each step, we propose to move from the current state x to a new state x' w.p. $q(x' \mid x)$
  - q is the proposal distribution
  - Common proposal distributions: (Murphy, 2023, § 12.3)
  - Good proposal design depends on the form of target distribution $\pi(x)$
- The user is free to use any kind of proposal they want, subject to $\text{supp}(\pi) \subseteq \cup_{x \in \mathcal{X}} \text{supp}(q(\cdot \mid x))$
  - q is valid if it "covers" the support of the target π: MH flexible method
- Having proposed a move to x', decide whether to accept this proposal, or to reject it
  - The long-term fraction of time spent in each state is $\propto \pi(x)$
- If the proposal is accepted, the new state is x', otherwise same as the current state x (repeat the sample)

MH Algorithm: Symmetric Proposal

- Proposal is symmetric: $q(x' \mid x) = q(x \mid x')$
- Acceptance probability: $A = \min\left(1, \frac{\pi(x')}{\pi(x)}\right)$
- If x' is more probable than x, we move there, since $\pi(x')/\pi(x) > 1$
- If x' is less probable, we may still move there anyway, depending on the relative probabilities
- Instead of greedily moving to only more probable states, we occasionally allow "downhill" moves to less probable states
- This procedure ensures that the fraction of time we spend in each state x is equal to $\pi(x)$ (Murphy, 2023, § 12.2.2)

MH Algorithm: Asymmetric Proposal

- Proposal is asymmetric: $q(x' \mid x) \neq q(x \mid x')$
- Need Hastings correction

$$A = \min(1, \alpha), \quad \alpha = \frac{\pi(x')q(x \mid x')}{\pi(x)q(x' \mid x)} = \frac{\pi(x')/q(x' \mid x)}{\pi(x)/q(x \mid x')}$$

  - Correction needed: the proposal distribution itself (rather than just the target distribution) might favor certain states
- When evaluating α, we only need to know the target density up to a normalization constant

$$\pi(x) = \frac{1}{Z}\tilde{\pi}(x) \implies \alpha = \frac{(\tilde{\pi}(x')/Z)q(x \mid x')}{(\tilde{\pi}(x)/Z)q(x' \mid x)} = \frac{\tilde{\pi}(x')q(x \mid x')}{\tilde{\pi}(x)q(x' \mid x)}$$

  - We can sample from π even if Z is unknown

Stochastic MH

Very active line of research still nowadays, see e.g. Bieringer et al. (2023)

Gibbs Sampling

- Problems with MH: (i) need to choose the proposal distribution, (ii) acceptance rate may be low
- Gibbs Sampling: MH method exploiting conditional independence to automatically create a good proposal; acceptance probability 1 (Murphy, 2023, Eq.'s (12.21), (12.22))

Gibbs Sampling: Algorithm

- We begin with some initial value $x^{(0)} = (x_1^{(0)}, x_2^{(0)}, x_3^{(0)})$ sampled from a joint distribution p on $\mathbb{R}^3$
- We want the next sample $x^{(1)}$
- To do so, for each component $x_j^{(1)}$, we condition on $x^{(1)}$'s components up to $x_{j-1}^{(1)}$
  - Thereafter condition on $x^{(0)}$'s components, starting from $x_{j+1}^{(0)}$ until $x_3^{(0)}$

$$x_1^{(1)} \sim p(x_1 \mid x_2^{(0)}, x_3^{(0)})$$
$$x_2^{(1)} \sim p(x_2 \mid x_1^{(1)}, x_3^{(0)})$$
$$x_3^{(1)} \sim p(x_3 \mid x_1^{(1)}, x_2^{(1)})$$

  - This immediately generalizes to $\mathbb{R}^D$, $D > 3$ and $x^{(s)}$, $s > 1$
- $p(x_j \mid x_{-j}^{(s)})$ is called the full conditional for variable j

Gibbs Sampling as a Specias Case of MH

- Gibbs sampling is a special case of MH where we use a sequence of proposals of the form

$$q_j(x' \mid x) = p(x'_j \mid x_{-j})\mathbb{I}(x'_{-j} = x_{-j})$$

  - We move to a new state where x_j is sampled from its full conditional, but x_{-j} is left unchanged
- The fact that the acceptance rate is 100% does not necessarily mean that Gibbs will converge rapidly
  - Only updates one coordinate at a time
- If we can group together correlated variables, then we can sample them as a group: help mixing (Murphy, 2023, § 12.3)

Metropolis Within Gibbs

- What should we do if we cannot sample from full conditional?
- To sample $x_j^{(s+1)} \sim p(x_j \mid x_{1:j-1}^{(s+1)}, x_{j+1:D}^{(s)})$, we
  - Propose $x_j' \sim q(x_j' \mid x_j^{(s)})$
  - Compute acceptance probability $A_j = \min(1, \alpha_j)$,

$$\alpha_j = \frac{p(x_{1:j-1}^{(s+1)}, x_j', x_{j+1:D}^{(s)}) / q(x_j' \mid x_j^{(s)})}{p(x_{1:j-1}^{(s+1)}, x_j^{(s)}, x_{j+1:D}^{(s)}) / q(x_j^{(s)} \mid x_j')}$$

  - Sample $u \sim \mathcal{U}(0, 1)$, and set

$$x_j^{(s+1)} = \begin{cases} x_j' & \text{if } u < A_j \\ x_j^{(s)} & \text{if } u \geq A_j \end{cases}$$

Hamiltonian Monte Carlo (HMC)

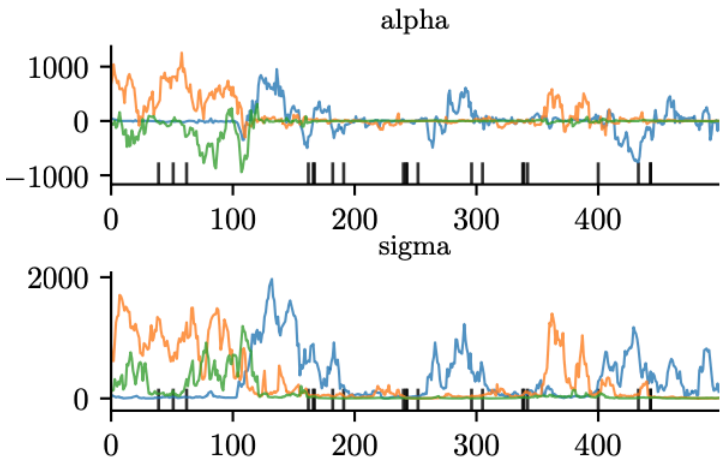
- Many MCMC algorithms perform poorly in high dimensional spaces, because they rely on a form of random search based on local perturbations
- Hamiltonian Monte Carlo leverages gradient information to guide the local moves
  - Important in applications
- Algorithm: (Murphy, 2023, § 12.5.3)
- Tuning: (Murphy, 2023, § 12.5.4)
- Implementation: BlackJAX

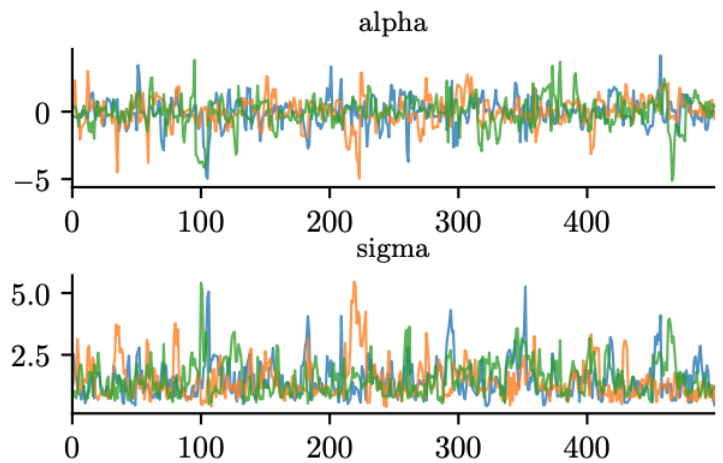
MCMC Convergence

- We start MCMC from an arbitrary initial state
- The samples will be coming from the chain's stationary distribution π only when the chain has "forgotten" where it started from
- Mixing time: amount of time it takes to enter the stationary distribution
- Samples collected before the chain has reached its stationary distribution do not come from π, and are usually thrown away
- Burn-in phase: initial period, whose samples will be ignored

MCMC Convergence Diagnostic

- Assessing if the method has converged:
  - Run multiple chains (typically 3 or 4) from very different overdispersed starting points
  - Trace plot: plot the samples of some quantity of interest
    - E.g. value of a certain component of the state vector, or some event such as the value taking on an extreme value
- If the chain has mixed, it should have "forgotten" where it started from
  - Trace plots should converge to the same distribution, and thus overlap with each other





MCMC Convergence Diagnostic

- MCMC lets us draw samples from a target distribution (assuming it has converged), but the samples are not independent
  - We may need to draw a lot of them to get a reliable estimate
- How to compute the effective sample size from a set of (possibly correlated) samples: (Murphy, 2023, § 12.6.3)
- How to improve the speed of convergence: (Murphy, 2023, § 12.6.4)

References

Sebastian Bieringer, Gregor Kasieczka, Maximilian F. Steffen, and Mathias Trabs. Statistical guarantees for stochastic Metropolis-Hastings. *Available at arXiv:2310.09335*, 2023.

Leonardo Cella and Ryan Martin. Variational approximations of possibilistic inferential models. *Available at arXiv:2404.19224*, 2024.

Kevin P. Murphy. *Probabilistic Machine Learning: Advanced Topics*. MIT Press, 2023. URL `http://probml.github.io/book2`.