Variational Inference (Wrap-up), Inference via Sampling

Piyush Rai

Topics in Probabilistic Modeling and Inference (CS698X)

Feb 27, 2019

Recap: VI using Monte-Carlo based Gradients of ELBO

- VI = ELBO optimization. Requires ELBO gradients: $\nabla_\phi \mathcal{L}(\phi) = \nabla_\phi \mathbb{E}_q[\log p(\mathbf{X}, \mathbf{Z}) - \log q(\mathbf{Z}|\phi)]$

Recap: VI using Monte-Carlo based Gradients of ELBO

- VI = ELBO optimization. Requires ELBO gradients: $\nabla_\phi \mathcal{L}(\phi) = \nabla_\phi \mathbb{E}_q[\log p(\mathbf{X}, \mathbf{Z}) - \log q(\mathbf{Z}|\phi)]$
- Looked at two approaches that optimize ELBO using its Monte-Carlo based gradients

Recap: VI using Monte-Carlo based Gradients of ELBO

- VI = ELBO optimization. Requires ELBO gradients: $\nabla_\phi \mathcal{L}(\phi) = \nabla_\phi \mathbb{E}_q[\log p(\mathbf{X}, \mathbf{Z}) - \log q(\mathbf{Z}|\phi)]$
- Looked at two approaches that optimize ELBO using its Monte-Carlo based gradients
  - Black-box VI (a.k.a. score-function gradients): No model-specific gradient calculations required

$$
\begin{aligned}
\mathbf{Z}_s &\sim q(\mathbf{Z}|\phi) \quad s = 1, \ldots, S \\
\nabla_\phi \mathcal{L}(q) &\approx \frac{1}{S} \sum_{s=1}^{S} \nabla_\phi \log q(\mathbf{Z}_s|\phi)[\log p(\mathbf{X}, \mathbf{Z}_s) - \log q(\mathbf{Z}_s|\phi)]
\end{aligned}
$$

Recap: VI using Monte-Carlo based Gradients of ELBO

- VI = ELBO optimization. Requires ELBO gradients: $\nabla_\phi \mathcal{L}(\phi) = \nabla_\phi \mathbb{E}_q[\log p(\mathbf{X}, \mathbf{Z}) - \log q(\mathbf{Z}|\phi)]$
- Looked at two approaches that optimize ELBO using its Monte-Carlo based gradients
  - Black-box VI (a.k.a. score-function gradients): No model-specific gradient calculations required

$$
\begin{aligned}
\mathbf{Z}_s &\sim q(\mathbf{Z}|\phi) \quad s = 1, \ldots, S \\
\nabla_\phi \mathcal{L}(q) &\approx \frac{1}{S}\sum_{s=1}^{S} \nabla_\phi \log q(\mathbf{Z}_s|\phi)[\log p(\mathbf{X}, \mathbf{Z}_s) - \log q(\mathbf{Z}_s|\phi)]
\end{aligned}
$$

  - Reparametrization trick (a.k.a. pathwise gradients)

$$
\begin{aligned}
\mathbf{Z} &= g(\epsilon, \phi) \\
\epsilon_s &\sim p(\epsilon) \quad s = 1, \ldots, S \\
\nabla_\phi \mathcal{L}(q) &\approx \frac{1}{S}\sum_{s=1}^{S}[\nabla_\phi \log p(\mathbf{X}, g(\epsilon_s, \phi)) - \nabla_\phi \log q_\phi(g(\epsilon_s, \phi))]
\end{aligned}
$$

Recap: VI using Monte-Carlo based Gradients of ELBO

- VI = ELBO optimization. Requires ELBO gradients: $\nabla_\phi \mathcal{L}(\phi) = \nabla_\phi \mathbb{E}_q[\log p(\mathbf{X}, \mathbf{Z}) - \log q(\mathbf{Z}|\phi)]$
- Looked at two approaches that optimize ELBO using its Monte-Carlo based gradients
  - Black-box VI (a.k.a. score-function gradients): No model-specific gradient calculations required

$$\begin{aligned}\mathbf{Z}_s &\sim q(\mathbf{Z}|\phi) \quad s = 1, \ldots, S \\ \nabla_\phi \mathcal{L}(q) &\approx \frac{1}{S}\sum_{s=1}^{S} \nabla_\phi \log q(\mathbf{Z}_s|\phi)[\log p(\mathbf{X}, \mathbf{Z}_s) - \log q(\mathbf{Z}_s|\phi)]\end{aligned}$$

  - Reparametrization trick (a.k.a. pathwise gradients)

$$\begin{aligned}\mathbf{Z} &= g(\epsilon, \phi) \\ \epsilon_s &\sim p(\epsilon) \quad s = 1, \ldots, S \\ \nabla_\phi \mathcal{L}(q) &\approx \frac{1}{S}\sum_{s=1}^{S} [\nabla_\phi \log p(\mathbf{X}, g(\epsilon_s, \phi)) - \nabla_\phi \log q_\phi(g(\epsilon_s, \phi))]\end{aligned}$$

- Note: We can use minibatches of data (instead of all $\mathbf{X}$) to compute the above gradients

Automatic Differentiation Variational Inference (ADVI)

- Auto. Diff. (AD): A way to automate differentiation of functions with unconstrained variables

*Automatic Differentiation Variational Inference (Kucukelbir et al, 2017)

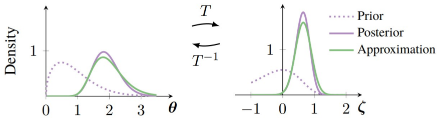
Automatic Differentiation Variational Inference (ADVI)

- Auto. Diff. (AD): A way to automate differentiation of functions with unconstrained variables
- VI is also optimization. However, often the variables are constrained, e.g.,
  - Gamma's shape and scale can only be non-negative
  - Beta's parameters can only be non-negative
  - Dirichlet's probability parameter sums to one

* Automatic Differentiation Variational Inference (Kucukelbir et al, 2017)

Automatic Differentiation Variational Inference (ADVI)

- Auto. Diff. (AD): A way to automate differentiation of functions with unconstrained variables
- VI is also optimization. However, often the variables are constrained, e.g.,
  - Gamma's shape and scale can only be non-negative
  - Beta's parameters can only be non-negative
  - Dirichlet's probability parameter sums to one
- If we can somehow transform our distributions to unconstrained ones, we can use AD for VI



(a) Latent variable space **(b)** Real coordinate space

$$T : \text{supp}(p(\boldsymbol{\theta})) \to \mathbb{R}^K$$

$$\boldsymbol{\zeta} = T(\boldsymbol{\theta})$$

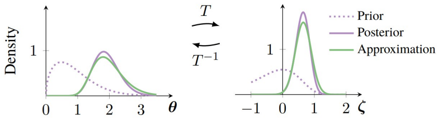
$$p(\mathbf{x}, \boldsymbol{\zeta}) = p\left(\mathbf{x}, T^{-1}(\boldsymbol{\zeta})\right) \left| \det J_{T^{-1}}(\boldsymbol{\zeta}) \right|$$

Transformed density — Original density — Jacobian of inverse of T

[*] Automatic Differentiation Variational Inference (Kucukelbir et al, 2017)

Automatic Differentiation Variational Inference (ADVI)

- Auto. Diff. (AD): A way to automate differentiation of functions with unconstrained variables
- VI is also optimization. However, often the variables are constrained, e.g.,
  - Gamma's shape and scale can only be non-negative
  - Beta's parameters can only be non-negative
  - Dirichlet's probability parameter sums to one
- If we can somehow transform our distributions to unconstrained ones, we can use AD for VI



(a) Latent variable space (b) Real coordinate space

$$T : \text{supp}(p(\boldsymbol{\theta})) \to \mathbb{R}^K$$

$$\boldsymbol{\zeta} = T(\boldsymbol{\theta})$$

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

Transformed density — Original density — Jacobian of inverse of T

- ADVI transforms the variables to real-valued and then does VI with Gaussian variational approx.

*Automatic Differentiation Variational Inference (Kucukelbir et al, 2017)

Amortized Variational Inference

Amortized Variational Inference

- Many latent variable models have one latent variable $\boldsymbol{z}_n$ for each data point $\boldsymbol{x}_n$

Amortized Variational Inference

- Many latent variable models have one latent variable $\boldsymbol{z}_n$ for each data point $\boldsymbol{x}_n$
- VI finds the optimal ϕ_n for each $q(\boldsymbol{z}_n|\phi_n)$

Amortized Variational Inference

- Many latent variable models have one latent variable $\boldsymbol{z}_n$ for each data point $\boldsymbol{x}_n$
- VI finds the optimal ϕ_n for each $q(\boldsymbol{z}_n|\phi_n)$
- This can be expensive for large datasets (a similar issue which motivated SVI)

Amortized Variational Inference

- Many latent variable models have one latent variable $\boldsymbol{z}_n$ for each data point $\boldsymbol{x}_n$
- VI finds the optimal ϕ_n for each $q(\boldsymbol{z}_n|\phi_n)$
- This can be expensive for large datasets (a similar issue which motivated SVI)
- Also slow at test time: Given a new $\boldsymbol{x}_*$, finding ϕ_* requires iterative updates

Amortized Variational Inference

- Many latent variable models have one latent variable $\boldsymbol{z}_n$ for each data point $\boldsymbol{x}_n$
- VI finds the optimal ϕ_n for each $q(\boldsymbol{z}_n|\phi_n)$
- This can be expensive for large datasets (a similar issue which motivated SVI)
- Also slow at test time: Given a new $\boldsymbol{x}_*$, finding ϕ_* requires iterative updates
  - Update local ϕ_*, update global λ, and repeat until convergence

Amortized Variational Inference

- Many latent variable models have one latent variable $\boldsymbol{z}_n$ for each data point $\boldsymbol{x}_n$
- VI finds the optimal ϕ_n for each $q(\boldsymbol{z}_n|\phi_n)$
- This can be expensive for large datasets (a similar issue which motivated SVI)
- Also slow at test time: Given a new $\boldsymbol{x}_*$, finding ϕ_* requires iterative updates
  - Update local ϕ_*, update global λ, and repeat until convergence
- Amortized VI : Learn an "inference network" or "recognition model" to directly get ϕ_n

Amortized Variational Inference

- Many latent variable models have one latent variable $\boldsymbol{z}_n$ for each data point $\boldsymbol{x}_n$
- VI finds the optimal ϕ_n for each $q(\boldsymbol{z}_n|\phi_n)$
- This can be expensive for large datasets (a similar issue which motivated SVI)
- Also slow at test time: Given a new $\boldsymbol{x}_*$, finding ϕ_* requires iterative updates
  - Update local ϕ_*, update global λ, and repeat until convergence
- Amortized VI : Learn an "inference network" or "recognition model" to directly get ϕ_n, e.g.,
  - A neural network to directly map $\boldsymbol{x}_n$ to ϕ_n

$$q(\boldsymbol{z}_n|\phi_n) \approx q(\boldsymbol{z}_n|\hat{\phi}_n) \quad \text{where} \quad \hat{\phi}_n = \text{NN}_\phi(\boldsymbol{x}_n)$$

Amortized Variational Inference

- Many latent variable models have one latent variable $\boldsymbol{z}_n$ for each data point $\boldsymbol{x}_n$
- VI finds the optimal ϕ_n for each $q(\boldsymbol{z}_n|\phi_n)$
- This can be expensive for large datasets (a similar issue which motivated SVI)
- Also slow at test time: Given a new $\boldsymbol{x}_*$, finding ϕ_* requires iterative updates
  - Update local ϕ_*, update global λ, and repeat until convergence
- Amortized VI : Learn an "inference network" or "recognition model" to directly get ϕ_n, e.g.,
  - A neural network to directly map $\boldsymbol{x}_n$ to ϕ_n

$$q(\boldsymbol{z}_n|\phi_n) \approx q(\boldsymbol{z}_n|\hat{\phi}_n) \quad \text{where} \quad \hat{\phi}_n = \text{NN}_\phi(\boldsymbol{x}_n)$$

- The inference network params ϕ can be learned along with the other global vars

Amortized Variational Inference

- Many latent variable models have one latent variable $\boldsymbol{z}_n$ for each data point $\boldsymbol{x}_n$
- VI finds the optimal ϕ_n for each $q(\boldsymbol{z}_n|\phi_n)$
- This can be expensive for large datasets (a similar issue which motivated SVI)
- Also slow at test time: Given a new $\boldsymbol{x}_*$, finding ϕ_* requires iterative updates
  - Update local ϕ_*, update global λ, and repeat until convergence
- Amortized VI : Learn an "inference network" or "recognition model" to directly get ϕ_n, e.g.,
  - A neural network to directly map $\boldsymbol{x}_n$ to ϕ_n

$$q(\boldsymbol{z}_n|\phi_n) \approx q(\boldsymbol{z}_n|\hat{\phi}_n) \quad \text{where} \quad \hat{\phi}_n = \text{NN}_\phi(\boldsymbol{x}_n)$$

- The inference network params ϕ can be learned along with the other global vars
- Popular in deep probabilistic models such as variational autoencoders, deep Gaussian Processes, etc

Structured Variational Inference

Structured Variational Inference

- Here "structured" may refer to anything that makes the VI approximation more expressive, e.g.,
  - Removing the independence assumption of mean-field VI
  - Learning more complex forms variational distributions

Structured Variational Inference

- Here "structured" may refer to anything that makes the VI approximation more expressive, e.g.,
  - Removing the independence assumption of mean-field VI
  - Learning more complex forms variational distributions
- To remove the mean-field assumption, various approaches exist

Structured Variational Inference

- Here "structured" may refer to anything that makes the VI approximation more expressive, e.g.,
  - Removing the independence assumption of mean-field VI
  - Learning more complex forms variational distributions
- To remove the mean-field assumption, various approaches exist
  - Structured mean-field (Saul et al, 1996)

Structured Variational Inference

- Here "structured" may refer to anything that makes the VI approximation more expressive, e.g.,
  - Removing the independence assumption of mean-field VI
  - Learning more complex forms variational distributions
- To remove the mean-field assumption, various approaches exist
  - Structured mean-field (Saul et al, 1996)
  - Hierarchical VI (Ranganath et al, 2016): Variational params $\phi_1, \ldots, \phi_M$ "tied" via a shared prior

$$q(\boldsymbol{z}_1, \ldots, \boldsymbol{z}_M | \theta) = \int \left[\prod_{m=1}^{M} q(\boldsymbol{z}_m | \phi_m) \right] p(\boldsymbol{\phi} | \theta) d\boldsymbol{\phi}$$

Structured Variational Inference

- Here "structured" may refer to anything that makes the VI approximation more expressive, e.g.,
  - Removing the independence assumption of mean-field VI
  - Learning more complex forms variational distributions
- To remove the mean-field assumption, various approaches exist
  - Structured mean-field (Saul et al, 1996)
  - Hierarchical VI (Ranganath et al, 2016): Variational params $\phi_1, \ldots, \phi_M$ "tied" via a shared prior

$$q(\boldsymbol{z}_1, \ldots, \boldsymbol{z}_M|\theta) = \int \left[\prod_{m=1}^{M} q(\boldsymbol{z}_m|\phi_m)\right] p(\phi|\theta)d\phi$$

- To learn more expressive variational approximations, various approaches exist

Structured Variational Inference

- Here "structured" may refer to anything that makes the VI approximation more expressive, e.g.,
  - Removing the independence assumption of mean-field VI
  - Learning more complex forms variational distributions
- To remove the mean-field assumption, various approaches exist
  - Structured mean-field (Saul et al, 1996)
  - Hierarchical VI (Ranganath et al, 2016): Variational params $\phi_1, \ldots, \phi_M$ "tied" via a shared prior

$$q(\boldsymbol{z}_1, \ldots, \boldsymbol{z}_M|\theta) = \int \left[\prod_{m=1}^{M} q(\boldsymbol{z}_m|\phi_m) \right] p(\boldsymbol{\phi}|\theta) d\boldsymbol{\phi}$$

- To learn more expressive variational approximations, various approaches exist, e.g.,
  - Boosting or mixture of simpler distributions, e.g., $q(\boldsymbol{z}) = \sum_{c=1}^{C} \rho_c q_c(\boldsymbol{z}|\phi_c)$

Structured Variational Inference

- Here "structured" may refer to anything that makes the VI approximation more expressive, e.g.,
  - Removing the independence assumption of mean-field VI
  - Learning more complex forms variational distributions
- To remove the mean-field assumption, various approaches exist
  - Structured mean-field (Saul et al, 1996)
  - Hierarchical VI (Ranganath et al, 2016): Variational params $\phi_1, \ldots, \phi_M$ "tied" via a shared prior

$$q(\boldsymbol{z}_1, \ldots, \boldsymbol{z}_M | \theta) = \int \left[\prod_{m=1}^{M} q(\boldsymbol{z}_m | \phi_m) \right] p(\boldsymbol{\phi}|\theta) d\boldsymbol{\phi}$$

- To learn more expressive variational approximations, various approaches exist, e.g.,
  - Boosting or mixture of simpler distributions, e.g., $q(\boldsymbol{z}) = \sum_{c=1}^{C} \rho_c q_c(\boldsymbol{z}|\phi_c)$
  - Normalizing flows:' Turn a simple $q(\boldsymbol{z})$ into a complex one via series of invertible transformations

Other Divergence Measures

Other Divergence Measures

- VI minimizes $KL(q||p)$ but other divergences can be minimized as well

Other Divergence Measures

- VI minimizes $KL(q||p)$ but other divergences can be minimized as well
- A general form of divergence is Renyi's α-divergence defined as

$$D_\alpha^R(p(\boldsymbol{x})||q(\boldsymbol{x})) = \frac{1}{\alpha - 1} \log \int p(\boldsymbol{x})^\alpha q(\boldsymbol{x})^{1-\alpha} d\boldsymbol{x}$$

Other Divergence Measures

- VI minimizes $KL(q||p)$ but other divergences can be minimized as well
- A general form of divergence is Renyi's α-divergence defined as

$$D_\alpha^R(p(\boldsymbol{x})||q(\boldsymbol{x})) = \frac{1}{\alpha - 1}\log\int p(\boldsymbol{x})^\alpha q(\boldsymbol{x})^{1-\alpha}d\boldsymbol{x}$$

- $KL(p||q)$ is a special case with $\alpha \to 1$ (can verify using L'Hopital rule of taking limits)

Other Divergence Measures

- VI minimizes $KL(q||p)$ but other divergences can be minimized as well
- A general form of divergence is Renyi's α-divergence defined as

$$D_\alpha^R(p(\boldsymbol{x})||q(\boldsymbol{x})) = \frac{1}{\alpha - 1}\log \int p(\boldsymbol{x})^\alpha q(\boldsymbol{x})^{1-\alpha} d\boldsymbol{x}$$

- $KL(p||q)$ is a special case with $\alpha \to 1$ (can verify using L'Hopital rule of taking limits)
- An even more general form of divergece is f-Divergence

$$D_f(p(\boldsymbol{x})||q(\boldsymbol{x})) = \int q(\boldsymbol{x}) f\left(\frac{p(\boldsymbol{x})}{q(\boldsymbol{x})}\right) d\boldsymbol{x}$$

Other Divergence Measures

- VI minimizes $KL(q||p)$ but other divergences can be minimized as well
- A general form of divergence is Renyi's α-divergence defined as

$$D_\alpha^R(p(\boldsymbol{x})||q(\boldsymbol{x})) = \frac{1}{\alpha - 1} \log \int p(\boldsymbol{x})^\alpha q(\boldsymbol{x})^{1-\alpha} d\boldsymbol{x}$$

- $KL(p||q)$ is a special case with $\alpha \rightarrow 1$ (can verify using L'Hopital rule of taking limits)
- An even more general form of divergece is f-Divergence

$$D_f(p(\boldsymbol{x})||q(\boldsymbol{x})) = \int q(\boldsymbol{x}) f\left(\frac{p(\boldsymbol{x})}{q(\boldsymbol{x})}\right) d\boldsymbol{x}$$

- Many recent inference algorithms are based on minimizing such divergences

Variational Inference: Some Comments

- Many probabilistic models (deep/non-deep) nowadays rely on VI to do tractable inference

Variational Inference: Some Comments

- Many probabilistic models (deep/non-deep) nowadays rely on VI to do tractable inference
- Even mean-field for locally-conjugate models has many applications in lots of probabilistic models
  - This + SVI gives excellent scalability

Variational Inference: Some Comments

- Many probabilistic models (deep/non-deep) nowadays rely on VI to do tractable inference
- Even mean-field for locally-conjugate models has many applications in lots of probabilistic models
  - This + SVI gives excellent scalability
- Stoch. opt., auto. diff., Monte-Carlo gradient of ELBO, contributed immensely to the success

Variational Inference: Some Comments

- Many probabilistic models (deep/non-deep) nowadays rely on VI to do tractable inference
- Even mean-field for locally-conjugate models has many applications in lots of probabilistic models
  - This + SVI gives excellent scalability
- Stoch. opt., auto. diff., Monte-Carlo gradient of ELBO, contributed immensely to the success
- Note: Most of these ideas apply also to Variational EM

Variational Inference: Some Comments

- Many probabilistic models (deep/non-deep) nowadays rely on VI to do tractable inference
- Even mean-field for locally-conjugate models has many applications in lots of probabilistic models
  - This + SVI gives excellent scalability
- Stoch. opt., auto. diff., Monte-Carlo gradient of ELBO, contributed immensely to the success
- Note: Most of these ideas apply also to Variational EM
- Many VI and advanced VI algorithms are implemented in probabilistic programming packages (e.g., Stan, Tensorflow Probability, etc), making VI a painless exercise even for complex models

Variational Inference: Some Comments

- Many probabilistic models (deep/non-deep) nowadays rely on VI to do tractable inference
- Even mean-field for locally-conjugate models has many applications in lots of probabilistic models
  - This + SVI gives excellent scalability
- Stoch. opt., auto. diff., Monte-Carlo gradient of ELBO, contributed immensely to the success
- Note: Most of these ideas apply also to Variational EM
- Many VI and advanced VI algorithms are implemented in probabilistic programming packages (e.g., Stan, Tensorflow Probability, etc), making VI a painless exercise even for complex models
- Still a very active area of research, especially for doing VI in complex models
  - Models with discrete latent variables
  - Reducing the variance in Monte-Carlo estimate of ELBO gradients

Inference via Sampling

(Note that we have already seen Gibbs sampling)

Sampling for Approximate Inference

- Some typical inference tasks

Sampling for Approximate Inference

- Some typical inference tasks
  - Compute a (possibly intractable) posterior distribution: $p(\theta|\mathcal{D}) = \frac{p(\mathcal{D}|\theta)p(\theta)}{p(\mathcal{D})} = \frac{p(\mathcal{D}|\theta)p(\theta)}{\int p(\mathcal{D}|\theta)p(\theta)d\theta}$

Sampling for Approximate Inference

- Some typical inference tasks
  - Compute a (possibly intractable) posterior distribution: $p(\theta|\mathcal{D}) = \frac{p(\mathcal{D}|\theta)p(\theta)}{p(\mathcal{D})} = \frac{p(\mathcal{D}|\theta)p(\theta)}{\int p(\mathcal{D}|\theta)p(\theta)d\theta}$
  - Compute a difficult expectation of a random quantity w.r.t. a distribution (an integral), e.g.,

Sampling for Approximate Inference

- Some typical inference tasks
  - Compute a (possibly intractable) posterior distribution: $p(\theta|\mathcal{D}) = \frac{p(\mathcal{D}|\theta)p(\theta)}{p(\mathcal{D})} = \frac{p(\mathcal{D}|\theta)p(\theta)}{\int p(\mathcal{D}|\theta)p(\theta)d\theta}$
  - Compute a difficult expectation of a random quantity w.r.t. a distribution (an integral), e.g.,
    - The posterior predictive (an expectation w.r.t the posterior over θ)

$$p(\mathcal{D}^{new}|\mathcal{D}) = \int p(\mathcal{D}^{new}|\theta)p(\theta|\mathcal{D})d\theta$$

Sampling for Approximate Inference

- Some typical inference tasks
  - Compute a (possibly intractable) posterior distribution: $p(\theta|\mathcal{D}) = \frac{p(\mathcal{D}|\theta)p(\theta)}{p(\mathcal{D})} = \frac{p(\mathcal{D}|\theta)p(\theta)}{\int p(\mathcal{D}|\theta)p(\theta)d\theta}$
  - Compute a difficult expectation of a random quantity w.r.t. a distribution (an integral), e.g.,
    - The posterior predictive (an expectation w.r.t the posterior over θ)

$$p(\mathcal{D}^{new}|\mathcal{D}) = \int p(\mathcal{D}^{new}|\theta)p(\theta|\mathcal{D})d\theta = \mathbb{E}_{p(\theta|\mathcal{D})}[p(\mathcal{D}^{new}|\theta)]$$

Sampling for Approximate Inference

- Some typical inference tasks
  - Compute a (possibly intractable) posterior distribution: $p(\theta|\mathcal{D}) = \frac{p(\mathcal{D}|\theta)p(\theta)}{p(\mathcal{D})} = \frac{p(\mathcal{D}|\theta)p(\theta)}{\int p(\mathcal{D}|\theta)p(\theta)d\theta}$
  - Compute a difficult expectation of a random quantity w.r.t. a distribution (an integral), e.g.,
    - The posterior predictive (an expectation w.r.t the posterior over θ)

$$p(\mathcal{D}^{new}|\mathcal{D}) = \int p(\mathcal{D}^{new}|\theta)p(\theta|\mathcal{D})d\theta = \mathbb{E}_{p(\theta|\mathcal{D})}[p(\mathcal{D}^{new}|\theta)]$$

    - The marginal likelihood or "evidence" (an expectation over the prior)

$$p(\mathcal{D}|m) = \int p(\mathcal{D}|\theta)p(\theta|m)d\theta$$

Sampling for Approximate Inference

- Some typical inference tasks
  - Compute a (possibly intractable) posterior distribution: $p(\theta|\mathcal{D}) = \frac{p(\mathcal{D}|\theta)p(\theta)}{p(\mathcal{D})} = \frac{p(\mathcal{D}|\theta)p(\theta)}{\int p(\mathcal{D}|\theta)p(\theta)d\theta}$
  - Compute a difficult expectation of a random quantity w.r.t. a distribution (an integral), e.g.,
    - The posterior predictive (an expectation w.r.t the posterior over θ)

$$p(\mathcal{D}^{new}|\mathcal{D}) = \int p(\mathcal{D}^{new}|\theta)p(\theta|\mathcal{D})d\theta = \mathbb{E}_{p(\theta|\mathcal{D})}[p(\mathcal{D}^{new}|\theta)]$$

    - The marginal likelihood or "evidence" (an expectation over the prior)

$$p(\mathcal{D}|m) = \int p(\mathcal{D}|\theta)p(\theta|m)d\theta = \mathbb{E}_{p(\theta|m)}[p(\mathcal{D}|\theta)]$$

Sampling for Approximate Inference

- Some typical inference tasks
  - Compute a (possibly intractable) posterior distribution: $p(\theta|\mathcal{D}) = \frac{p(\mathcal{D}|\theta)p(\theta)}{p(\mathcal{D})} = \frac{p(\mathcal{D}|\theta)p(\theta)}{\int p(\mathcal{D}|\theta)p(\theta)d\theta}$
  - Compute a difficult expectation of a random quantity w.r.t. a distribution (an integral), e.g.,
    - The posterior predictive (an expectation w.r.t the posterior over θ)

$$p(\mathcal{D}^{new}|\mathcal{D}) = \int p(\mathcal{D}^{new}|\theta)p(\theta|\mathcal{D})d\theta = \mathbb{E}_{p(\theta|\mathcal{D})}[p(\mathcal{D}^{new}|\theta)]$$

    - The marginal likelihood or "evidence" (an expectation over the prior)

$$p(\mathcal{D}|m) = \int p(\mathcal{D}|\theta)p(\theta|m)d\theta = \mathbb{E}_{p(\theta|m)}[p(\mathcal{D}|\theta)]$$

    - The expected complete data log-likelihood needed for doing MLE/MAP in LVMs (recall EM)

$$\text{Exp-CLL} = \int p(\boldsymbol{z}|\theta, \boldsymbol{x})p(\boldsymbol{x}, \boldsymbol{z}|\theta)d\boldsymbol{z}$$

Sampling for Approximate Inference

- Some typical inference tasks
  - Compute a (possibly intractable) posterior distribution: $p(\theta|\mathcal{D}) = \frac{p(\mathcal{D}|\theta)p(\theta)}{p(\mathcal{D})} = \frac{p(\mathcal{D}|\theta)p(\theta)}{\int p(\mathcal{D}|\theta)p(\theta)d\theta}$
  - Compute a difficult expectation of a random quantity w.r.t. a distribution (an integral), e.g.,
    - The posterior predictive (an expectation w.r.t the posterior over θ)

$$p(\mathcal{D}^{new}|\mathcal{D}) = \int p(\mathcal{D}^{new}|\theta)p(\theta|\mathcal{D})d\theta = \mathbb{E}_{p(\theta|\mathcal{D})}[p(\mathcal{D}^{new}|\theta)]$$

    - The marginal likelihood or "evidence" (an expectation over the prior)

$$p(\mathcal{D}|m) = \int p(\mathcal{D}|\theta)p(\theta|m)d\theta = \mathbb{E}_{p(\theta|m)}[p(\mathcal{D}|\theta)]$$

    - The expected complete data log-likelihood needed for doing MLE/MAP in LVMs (recall EM)

$$\text{Exp-CLL} = \int p(\boldsymbol{z}|\theta,\boldsymbol{x})p(\boldsymbol{x},\boldsymbol{z}|\theta)d\boldsymbol{z} = \mathbb{E}_{p(\boldsymbol{z}|\theta,\boldsymbol{x})}[p(\boldsymbol{x},\boldsymbol{z}|\theta)]$$

Sampling for Approximate Inference

- Some typical inference tasks
  - Compute a (possibly intractable) posterior distribution: $p(\theta|\mathcal{D}) = \frac{p(\mathcal{D}|\theta)p(\theta)}{p(\mathcal{D})} = \frac{p(\mathcal{D}|\theta)p(\theta)}{\int p(\mathcal{D}|\theta)p(\theta)d\theta}$
  - Compute a difficult expectation of a random quantity w.r.t. a distribution (an integral), e.g.,
    - The posterior predictive (an expectation w.r.t the posterior over θ)

$$p(\mathcal{D}^{new}|\mathcal{D}) = \int p(\mathcal{D}^{new}|\theta)p(\theta|\mathcal{D})d\theta = \mathbb{E}_{p(\theta|\mathcal{D})}[p(\mathcal{D}^{new}|\theta)]$$

    - The marginal likelihood or "evidence" (an expectation over the prior)

$$p(\mathcal{D}|m) = \int p(\mathcal{D}|\theta)p(\theta|m)d\theta = \mathbb{E}_{p(\theta|m)}[p(\mathcal{D}|\theta)]$$

    - The expected complete data log-likelihood needed for doing MLE/MAP in LVMs (recall EM)

$$\text{Exp-CLL} = \int p(\boldsymbol{z}|\theta, \boldsymbol{x})p(\boldsymbol{x}, \boldsymbol{z}|\theta)d\boldsymbol{z} = \mathbb{E}_{p(\boldsymbol{z}|\theta,\boldsymbol{x})}[p(\boldsymbol{x}, \boldsymbol{z}|\theta)]$$

    - The ELBO in variational inference

$$\mathcal{L}(q) = \mathbb{E}_q[\log p(\boldsymbol{x}, \boldsymbol{z})] - \mathbb{E}_q[\log p(\boldsymbol{z})[$$

Sampling for Approximate Inference

- Some typical inference tasks
  - Compute a (possibly intractable) posterior distribution: $p(\theta|\mathcal{D}) = \frac{p(\mathcal{D}|\theta)p(\theta)}{p(\mathcal{D})} = \frac{p(\mathcal{D}|\theta)p(\theta)}{\int p(\mathcal{D}|\theta)p(\theta)d\theta}$
  - Compute a difficult expectation of a random quantity w.r.t. a distribution (an integral), e.g.,
    - The posterior predictive (an expectation w.r.t the posterior over θ)

$$p(\mathcal{D}^{new}|\mathcal{D}) = \int p(\mathcal{D}^{new}|\theta)p(\theta|\mathcal{D})d\theta = \mathbb{E}_{p(\theta|\mathcal{D})}[p(\mathcal{D}^{new}|\theta)]$$

    - The marginal likelihood or "evidence" (an expectation over the prior)

$$p(\mathcal{D}|m) = \int p(\mathcal{D}|\theta)p(\theta|m)d\theta = \mathbb{E}_{p(\theta|m)}[p(\mathcal{D}|\theta)]$$

    - The expected complete data log-likelihood needed for doing MLE/MAP in LVMs (recall EM)

$$\text{Exp-CLL} = \int p(\boldsymbol{z}|\theta,\boldsymbol{x})p(\boldsymbol{x},\boldsymbol{z}|\theta)d\boldsymbol{z} = \mathbb{E}_{p(\boldsymbol{z}|\theta,\boldsymbol{x})}[p(\boldsymbol{x},\boldsymbol{z}|\theta)]$$

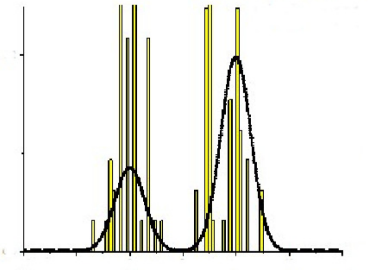
    - The ELBO in variational inference

$$\mathcal{L}(q) = \mathbb{E}_q[\log p(\boldsymbol{x},\boldsymbol{z})] - \mathbb{E}_q[\log p(\boldsymbol{z})[$$

- Sampling methods provide a general way to (approximately) solve these problems

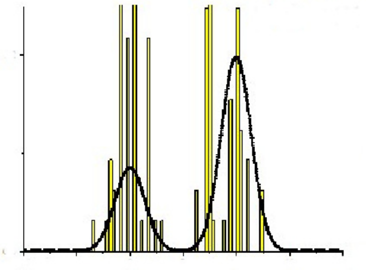
The Basic Idea

- Can approximate any distribution using a set of randomly drawn samples from it



The Basic Idea

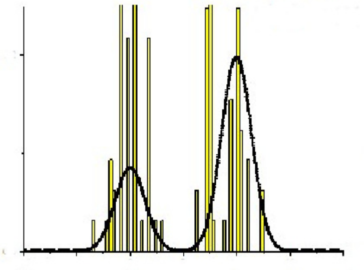
- Can approximate any distribution using a set of randomly drawn samples from it



- The samples can also be used for computing expectations (Monte-Carlo averaging)

The Basic Idea

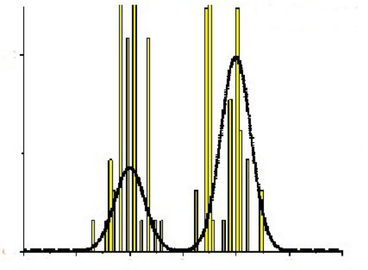
- Can approximate any distribution using a set of randomly drawn samples from it



- The samples can also be used for computing expectations (Monte-Carlo averaging)
- Usually straightforward to generate samples if it is a simple/standard distribution

The Basic Idea

- Can approximate any distribution using a set of randomly drawn samples from it



- The samples can also be used for computing expectations (Monte-Carlo averaging)
- Usually straightforward to generate samples if it is a simple/standard distribution
- **The interesting bit:** Even if the distribution is "difficult" (e.g., an intractable posterior), it is often possible to generate random samples from such a distribution, as we will see..

Empirical Distribution

- Sampling based approximation of a distribution can be represented using an empirical distribution

Empirical Distribution

- Sampling based approximation of a distribution can be represented using an empirical distribution
- Given L "points" $\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(L)}$, the empirical distribution of these points is defined as

Empirical Distribution

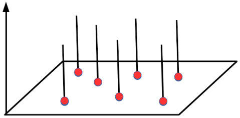
- Sampling based approximation of a distribution can be represented using an empirical distribution
- Given L "points" $\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(L)}$, the empirical distribution of these points is defined as

$$p_L(A) = \sum_{\ell=1}^{L} w_\ell \delta_{\boldsymbol{z}^{(\ell)}}(A)$$

Empirical Distribution

- Sampling based approximation of a distribution can be represented using an empirical distribution
- Given L "points" $\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(L)}$, the empirical distribution of these points is defined as

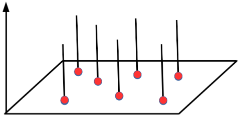
$$p_L(A) = \sum_{\ell=1}^{L} w_\ell \delta_{\boldsymbol{z}^{(\ell)}}(A)$$



Empirical Distribution

- Sampling based approximation of a distribution can be represented using an empirical distribution
- Given L "points" $\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(L)}$, the empirical distribution of these points is defined as

$$p_L(A) = \sum_{\ell=1}^{L} w_\ell \delta_{\boldsymbol{z}^{(\ell)}}(A)$$

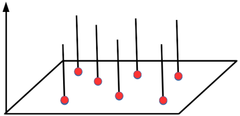


- Here $w_1, \ldots, w_L$ are weights that sum to 1, i.e., $\sum_{\ell=1}^{L} w_\ell = 1$ (for uniform weights, $w_\ell = 1/L$)

Empirical Distribution

- Sampling based approximation of a distribution can be represented using an empirical distribution
- Given L "points" $\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(L)}$, the empirical distribution of these points is defined as

$$p_L(A) = \sum_{\ell=1}^{L} w_\ell \delta_{\boldsymbol{z}^{(\ell)}}(A)$$



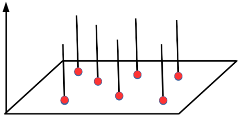
- Here $w_1, \ldots, w_L$ are weights that sum to 1, i.e., $\sum_{\ell=1}^{L} w_\ell = 1$ (for uniform weights, $w_\ell = 1/L$)
- Here $\delta_{\boldsymbol{z}}(A)$ denotes the Dirac distribution defined as

$$\delta_{\boldsymbol{z}}(A) = \begin{cases} 0 & \text{if} \quad \boldsymbol{z} \notin A \\ 1 & \text{if} \quad \boldsymbol{z} \in A \end{cases}$$

Empirical Distribution

- Sampling based approximation of a distribution can be represented using an empirical distribution
- Given L "points" $\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(L)}$, the empirical distribution of these points is defined as

$$p_L(A) = \sum_{\ell=1}^{L} w_\ell \delta_{\boldsymbol{z}^{(\ell)}}(A)$$



- Here $w_1, \ldots, w_L$ are weights that sum to 1, i.e., $\sum_{\ell=1}^{L} w_\ell = 1$ (for uniform weights, $w_\ell = 1/L$)
- Here $\delta_{\boldsymbol{z}}(A)$ denotes the Dirac distribution defined as

$$\delta_{\boldsymbol{z}}(A) = \begin{cases} 0 & \text{if} \quad \boldsymbol{z} \notin A \\ 1 & \text{if} \quad \boldsymbol{z} \in A \end{cases}$$

- $p_L(A)$ is a discrete distribution with finite support $\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(L)}$ (can think of it as a histogram)

Approximate Inference: VI vs Sampling-based

- VI approximates a posterior distribution $p(\mathbf{Z}|\mathbf{X})$ by another distribution $q(\mathbf{Z}|\phi)$
- Sampling uses S (typically large number) samples $\{\mathbf{Z}_s\}_{s=1}^{S}$ to approximate $p(\mathbf{Z}|\mathbf{X})$

Approximate Inference: VI vs Sampling-based

- VI approximates a posterior distribution $p(\mathbf{Z}|\mathbf{X})$ by another distribution $q(\mathbf{Z}|\phi)$
- Sampling uses S (typically large number) samples $\{\mathbf{Z}_s\}_{s=1}^S$ to approximate $p(\mathbf{Z}|\mathbf{X})$
- Sampling can be used within VI (already saw ELBO approximations using Monte-Carlo)

Approximate Inference: VI vs Sampling-based

- VI approximates a posterior distribution $p(\mathbf{Z}|\mathbf{X})$ by another distribution $q(\mathbf{Z}|\phi)$
- Sampling uses S (typically large number) samples $\{\mathbf{Z}_s\}_{s=1}^S$ to approximate $p(\mathbf{Z}|\mathbf{X})$
- Sampling can be used within VI (already saw ELBO approximations using Monte-Carlo)
- Also possible (though less common) to use VI in sampling algorithms (will talk about it later)

Approximate Inference: VI vs Sampling-based

- VI approximates a posterior distribution $p(\mathbf{Z}|\mathbf{X})$ by another distribution $q(\mathbf{Z}|\phi)$
- Sampling uses S (typically large number) samples $\{\mathbf{Z}_s\}_{s=1}^S$ to approximate $p(\mathbf{Z}|\mathbf{X})$
- Sampling can be used within VI (already saw ELBO approximations using Monte-Carlo)
- Also possible (though less common) to use VI in sampling algorithms (will talk about it later)
- In terms of "comparison" between VI and sampling, a few things to be noted

Approximate Inference: VI vs Sampling-based

- VI approximates a posterior distribution $p(\mathbf{Z}|\mathbf{X})$ by another distribution $q(\mathbf{Z}|\phi)$
- Sampling uses S (typically large number) samples $\{\mathbf{Z}_s\}_{s=1}^S$ to approximate $p(\mathbf{Z}|\mathbf{X})$
- Sampling can be used within VI (already saw ELBO approximations using Monte-Carlo)
- Also possible (though less common) to use VI in sampling algorithms (will talk about it later)
- In terms of "comparison" between VI and sampling, a few things to be noted
  - Convergence: VI only has local convergece, sampling (in theory) can give posterior (more on it later)

Approximate Inference: VI vs Sampling-based

- VI approximates a posterior distribution $p(\mathbf{Z}|\mathbf{X})$ by another distribution $q(\mathbf{Z}|\phi)$
- Sampling uses S (typically large number) samples $\{\mathbf{Z}_s\}_{s=1}^S$ to approximate $p(\mathbf{Z}|\mathbf{X})$
- Sampling can be used within VI (already saw ELBO approximations using Monte-Carlo)
- Also possible (though less common) to use VI in sampling algorithms (will talk about it later)
- In terms of "comparison" between VI and sampling, a few things to be noted
  - Convergence: VI only has local convergece, sampling (in theory) can give posterior (more on it later)
  - Storage requirements: Sampling-based approximation requires more storage (why?)

Approximate Inference: VI vs Sampling-based

- VI approximates a posterior distribution $p(\mathbf{Z}|\mathbf{X})$ by another distribution $q(\mathbf{Z}|\phi)$
- Sampling uses S (typically large number) samples $\{\mathbf{Z}_s\}_{s=1}^{S}$ to approximate $p(\mathbf{Z}|\mathbf{X})$
- Sampling can be used within VI (already saw ELBO approximations using Monte-Carlo)
- Also possible (though less common) to use VI in sampling algorithms (will talk about it later)
- In terms of "comparison" between VI and sampling, a few things to be noted
  - Convergence: VI only has local convergece, sampling (in theory) can give posterior (more on it later)
  - Storage requirements: Sampling-based approximation requires more storage (why?)
  - Prediction time cost (also related to storage requirement): Sampling always requires Monte-Carlo averaging for posterior predictive; with VI, sometimes we can get closed form posterior predictive

Approximate Inference: VI vs Sampling-based

- VI approximates a posterior distribution $p(\mathbf{Z}|\mathbf{X})$ by another distribution $q(\mathbf{Z}|\phi)$
- Sampling uses S (typically large number) samples $\{\mathbf{Z}_s\}_{s=1}^S$ to approximate $p(\mathbf{Z}|\mathbf{X})$
- Sampling can be used within VI (already saw ELBO approximations using Monte-Carlo)
- Also possible (though less common) to use VI in sampling algorithms (will talk about it later)
- In terms of "comparison" between VI and sampling, a few things to be noted
  - Convergence: VI only has local convergece, sampling (in theory) can give posterior (more on it later)
  - Storage requirements: Sampling-based approximation requires more storage (why?)
  - Prediction time cost (also related to storage requirement): Sampling always requires Monte-Carlo averaging for posterior predictive; with VI, sometimes we can get closed form posterior predictive
    - Sampling based posterior predictive: $p(x_*|\mathbf{X}) \approx \frac{1}{S}\sum_{s=1}^S p(x_*|\theta_s)p(\theta_s|\mathbf{X})$

Approximate Inference: VI vs Sampling-based

- VI approximates a posterior distribution $p(\mathbf{Z}|\mathbf{X})$ by another distribution $q(\mathbf{Z}|\phi)$
- Sampling uses S (typically large number) samples $\{\mathbf{Z}_s\}_{s=1}^S$ to approximate $p(\mathbf{Z}|\mathbf{X})$
- Sampling can be used within VI (already saw ELBO approximations using Monte-Carlo)
- Also possible (though less common) to use VI in sampling algorithms (will talk about it later)
- In terms of "comparison" between VI and sampling, a few things to be noted
  - Convergence: VI only has local convergece, sampling (in theory) can give posterior (more on it later)
  - Storage requirements: Sampling-based approximation requires more storage (why?)
  - Prediction time cost (also related to storage requirement): Sampling always requires Monte-Carlo averaging for posterior predictive; with VI, sometimes we can get closed form posterior predictive
    - Sampling based posterior predictive: $p(x_*|\mathbf{X}) \approx \frac{1}{S}\sum_{s=1}^S p(x_*|\theta_s)p(\theta_s|\mathbf{X})$
    - VI based posterior predictive: $p(x_*|\mathbf{X}) \approx \int p(x_*|\theta)q(\theta|\phi)d\theta$

Approximate Inference: VI vs Sampling-based

- VI approximates a posterior distribution $p(\mathbf{Z}|\mathbf{X})$ by another distribution $q(\mathbf{Z}|\phi)$
- Sampling uses S (typically large number) samples $\{\mathbf{Z}_s\}_{s=1}^S$ to approximate $p(\mathbf{Z}|\mathbf{X})$
- Sampling can be used within VI (already saw ELBO approximations using Monte-Carlo)
- Also possible (though less common) to use VI in sampling algorithms (will talk about it later)
- In terms of "comparison" between VI and sampling, a few things to be noted
  - Convergence: VI only has local convergece, sampling (in theory) can give posterior (more on it later)
  - Storage requirements: Sampling-based approximation requires more storage (why?)
  - Prediction time cost (also related to storage requirement): Sampling always requires Monte-Carlo averaging for posterior predictive; with VI, sometimes we can get closed form posterior predictive
    - Sampling based posterior predictive: $p(x_*|\mathbf{X}) \approx \frac{1}{S}\sum_{s=1}^S p(x_*|\theta_s)p(\theta_s|\mathbf{X})$
    - VI based posterior predictive: $p(x_*|\mathbf{X}) \approx \int p(x_*|\theta)q(\theta|\phi)d\theta$
  - There is some work on "compressing" sampling-based approximations

Approximate Inference: VI vs Sampling-based

- VI approximates a posterior distribution $p(\mathbf{Z}|\mathbf{X})$ by another distribution $q(\mathbf{Z}|\phi)$
- Sampling uses S (typically large number) samples $\{\mathbf{Z}_s\}_{s=1}^S$ to approximate $p(\mathbf{Z}|\mathbf{X})$
- Sampling can be used within VI (already saw ELBO approximations using Monte-Carlo)
- Also possible (though less common) to use VI in sampling algorithms (will talk about it later)
- In terms of "comparison" between VI and sampling, a few things to be noted
  - Convergence: VI only has local convergece, sampling (in theory) can give posterior (more on it later)
  - Storage requirements: Sampling-based approximation requires more storage (why?)
  - Prediction time cost (also related to storage requirement): Sampling always requires Monte-Carlo averaging for posterior predictive; with VI, sometimes we can get closed form posterior predictive
    - Sampling based posterior predictive: $p(x_*|\mathbf{X}) \approx \frac{1}{S}\sum_{s=1}^S p(x_*|\theta_s)p(\theta_s|\mathbf{X})$
    - VI based posterior predictive: $p(x_*|\mathbf{X}) \approx \int p(x_*|\theta)q(\theta|\phi)d\theta$
  - There is some work on "compressing" sampling-based approximations (e.g., see "Compact approximations to Bayesian predictive distributions" by Snelson and Ghaharamani, 2005; and "Bayesian Dark Knowledge" by Korattikara et al, 2015)

Sampling: Some Basic Methods

- Most of these basic methods are based on the idea of transformation

Sampling: Some Basic Methods

- Most of these basic methods are based on the idea of transformation
- Given a sample x from an "easy" distribution $p(x)$, transform it into a random sample z from a "less easy" distribution $p(z)$

Sampling: Some Basic Methods

- Most of these basic methods are based on the idea of transformation
- Given a sample x from an "easy" distribution $p(x)$, transform it into a random sample z from a "less easy" distribution $p(z)$
- Some popular examples of transformation methods

Sampling: Some Basic Methods

- Most of these basic methods are based on the idea of transformation
- Given a sample x from an "easy" distribution $p(x)$, transform it into a random sample z from a "less easy" distribution $p(z)$
- Some popular examples of transformation methods
  - Inverse CDF method

$$x \sim \text{Unif}(0,1) \Rightarrow z = \text{Inv-CDF}_{p(z)}(x) \sim p(z)$$

Sampling: Some Basic Methods

- Most of these basic methods are based on the idea of transformation
- Given a sample x from an "easy" distribution $p(x)$, transform it into a random sample z from a "less easy" distribution $p(z)$
- Some popular examples of transformation methods
  - Inverse CDF method

$$x \sim \text{Unif}(0,1) \Rightarrow z = \text{Inv-CDF}_{p(z)}(x) \sim p(z)$$

  - Reparametrization method

$$x \sim \mathcal{N}(0,1) \Rightarrow z = \mu + \sigma x$$

Sampling: Some Basic Methods

- Most of these basic methods are based on the idea of transformation
- Given a sample x from an "easy" distribution $p(x)$, transform it into a random sample z from a "less easy" distribution $p(z)$
- Some popular examples of transformation methods
  - Inverse CDF method

$$x \sim \text{Unif}(0,1) \Rightarrow z = \text{Inv-CDF}_{p(z)}(x) \sim p(z)$$

  - Reparametrization method

$$x \sim \mathcal{N}(0,1) \Rightarrow z = \mu + \sigma x \sim \mathcal{N}(\mu, \sigma^2)$$

Sampling: Some Basic Methods

- Most of these basic methods are based on the idea of transformation
- Given a sample x from an "easy" distribution $p(x)$, transform it into a random sample z from a "less easy" distribution $p(z)$
- Some popular examples of transformation methods
  - Inverse CDF method

$$x \sim \text{Unif}(0,1) \Rightarrow z = \text{Inv-CDF}_{p(z)}(x) \sim p(z)$$

  - Reparametrization method

$$x \sim \mathcal{N}(0,1) \Rightarrow z = \mu + \sigma x \sim \mathcal{N}(\mu, \sigma^2)$$

  - Box-Muller method: Given (x_1, x_2) from $\text{Unif}(-1,+1)$, generate (z_1, z_2) from 2D Gaussian $\mathcal{N}(0, \mathbf{I})$

Sampling: Some Basic Methods

- Most of these basic methods are based on the idea of transformation
- Given a sample x from an "easy" distribution $p(x)$, transform it into a random sample z from a "less easy" distribution $p(z)$
- Some popular examples of transformation methods
  - Inverse CDF method

$$x \sim \text{Unif}(0,1) \Rightarrow z = \text{Inv-CDF}_{p(z)}(x) \sim p(z)$$

  - Reparametrization method

$$x \sim \mathcal{N}(0,1) \Rightarrow z = \mu + \sigma x \sim \mathcal{N}(\mu, \sigma^2)$$

  - Box-Muller method: Given (x_1, x_2) from $\text{Unif}(-1,+1)$, generate (z_1, z_2) from 2D Gaussian $\mathcal{N}(0, \mathbf{I})$
- Transformation Methods are simple but have limitations

Sampling: Some Basic Methods

- Most of these basic methods are based on the idea of transformation
- Given a sample x from an "easy" distribution $p(x)$, transform it into a random sample z from a "less easy" distribution $p(z)$
- Some popular examples of transformation methods
  - Inverse CDF method

$$x \sim \text{Unif}(0,1) \Rightarrow z = \text{Inv-CDF}_{p(z)}(x) \sim p(z)$$

  - Reparametrization method

$$x \sim \mathcal{N}(0,1) \Rightarrow z = \mu + \sigma x \sim \mathcal{N}(\mu, \sigma^2)$$

  - Box-Muller method: Given (x_1, x_2) from $\text{Unif}(-1,+1)$, generate (z_1, z_2) from 2D Gaussian $\mathcal{N}(0, \mathbf{I})$
- Transformation Methods are simple but have limitations
  - Mostly limited to standard distributions and/or distributions with very few variables

Rejection Sampling

- Want to sample from $p(\boldsymbol{z}) = \frac{\tilde{p}(\boldsymbol{z})}{Z_p}$.

Rejection Sampling

- Want to sample from $p(\boldsymbol{z}) = \frac{\tilde{p}(\boldsymbol{z})}{Z_p}$. Suppose we can only evaluate the numerator $\tilde{p}(\boldsymbol{z})$ at any $\boldsymbol{z}$

Rejection Sampling

- Want to sample from $p(\boldsymbol{z}) = \frac{\tilde{p}(\boldsymbol{z})}{Z_p}$. Suppose we can only evaluate the numerator $\tilde{p}(\boldsymbol{z})$ at any $\boldsymbol{z}$
- Suppose we have a proposal distribution $q(\boldsymbol{z})$ that we can generate samples from, and

$$Mq(\boldsymbol{z}) \geq \tilde{p}(\boldsymbol{z}) \qquad \forall \boldsymbol{z} \quad \text{(where } M > 0 \text{ is some const.)}$$

Rejection Sampling

- Want to sample from $p(\boldsymbol{z}) = \frac{\tilde{p}(\boldsymbol{z})}{Z_p}$. Suppose we can only evaluate the numerator $\tilde{p}(\boldsymbol{z})$ at any $\boldsymbol{z}$
- Suppose we have a proposal distribution $q(\boldsymbol{z})$ that we can generate samples from, and

$$Mq(\boldsymbol{z}) \geq \tilde{p}(\boldsymbol{z}) \qquad \forall \boldsymbol{z} \quad \text{(where } M > 0 \text{ is some const.)}$$

- Basic idea: Generate samples from the proposal $q(\boldsymbol{z})$

Rejection Sampling

- Want to sample from $p(\boldsymbol{z}) = \frac{\tilde{p}(\boldsymbol{z})}{Z_p}$. Suppose we can only evaluate the numerator $\tilde{p}(\boldsymbol{z})$ at any $\boldsymbol{z}$
- Suppose we have a proposal distribution $q(\boldsymbol{z})$ that we can generate samples from, and

$$Mq(\boldsymbol{z}) \geq \tilde{p}(\boldsymbol{z}) \qquad \forall \boldsymbol{z} \quad \text{(where } M > 0 \text{ is some const.)}$$

- Basic idea: Generate samples from the proposal $q(\boldsymbol{z})$ and accept/reject based on some condition

Rejection Sampling

- Want to sample from $p(\boldsymbol{z}) = \frac{\tilde{p}(\boldsymbol{z})}{Z_p}$. Suppose we can only evaluate the numerator $\tilde{p}(\boldsymbol{z})$ at any $\boldsymbol{z}$
- Suppose we have a proposal distribution $q(\boldsymbol{z})$ that we can generate samples from, and

$$Mq(\boldsymbol{z}) \geq \tilde{p}(\boldsymbol{z}) \qquad \forall \boldsymbol{z} \quad \text{(where } M > 0 \text{ is some const.)}$$

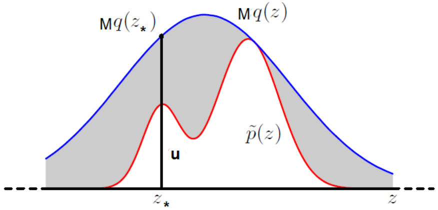
- Basic idea: Generate samples from the proposal $q(\boldsymbol{z})$ and accept/reject based on some condition
  1. Sample an r.v. $\boldsymbol{z}_*$ from $q(\boldsymbol{z})$

Rejection Sampling

- Want to sample from $p(\boldsymbol{z}) = \frac{\tilde{p}(\boldsymbol{z})}{Z_p}$. Suppose we can only evaluate the numerator $\tilde{p}(\boldsymbol{z})$ at any $\boldsymbol{z}$
- Suppose we have a proposal distribution $q(\boldsymbol{z})$ that we can generate samples from, and

$$Mq(\boldsymbol{z}) \geq \tilde{p}(\boldsymbol{z}) \qquad \forall \boldsymbol{z} \quad \text{(where } M > 0 \text{ is some const.)}$$

- Basic idea: Generate samples from the proposal $q(\boldsymbol{z})$ and accept/reject based on some condition
  1. Sample an r.v. $\boldsymbol{z}_*$ from $q(\boldsymbol{z})$
  2. Sampling a uniform r.v. $u \sim \text{Unif}[0, Mq(\boldsymbol{z}_*)]$

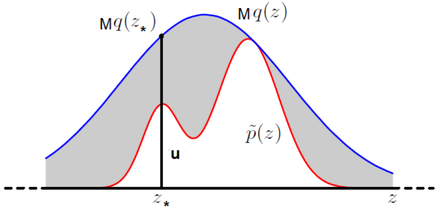


Rejection Sampling

- Want to sample from $p(\boldsymbol{z}) = \frac{\tilde{p}(\boldsymbol{z})}{Z_p}$. Suppose we can only evaluate the numerator $\tilde{p}(\boldsymbol{z})$ at any $\boldsymbol{z}$
- Suppose we have a proposal distribution $q(\boldsymbol{z})$ that we can generate samples from, and

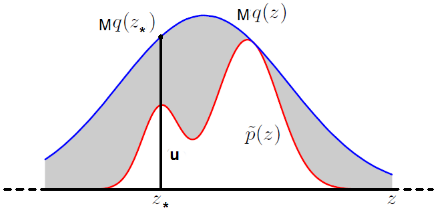
$$Mq(\boldsymbol{z}) \geq \tilde{p}(\boldsymbol{z}) \qquad \forall \boldsymbol{z} \quad \text{(where } M > 0 \text{ is some const.)}$$

- Basic idea: Generate samples from the proposal $q(\boldsymbol{z})$ and accept/reject based on some condition
  1. Sample an r.v. $\boldsymbol{z}_*$ from $q(\boldsymbol{z})$
  2. Sampling a uniform r.v. $u \sim \text{Unif}[0, Mq(\boldsymbol{z}_*)]$



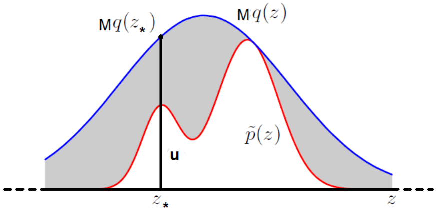
  3. If $u \leq \tilde{p}(\boldsymbol{z}_*)$ then accept $\boldsymbol{z}_*$ else reject

Rejection Sampling



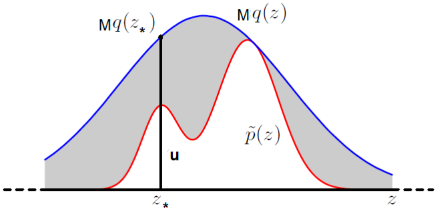
- Why $\boldsymbol{z} \sim q(\boldsymbol{z})$ + accept/reject rule is equivalent to $\boldsymbol{z} \sim p(\boldsymbol{z})$?

Rejection Sampling



- Why $\boldsymbol{z} \sim q(\boldsymbol{z})$ + accept/reject rule is equivalent to $\boldsymbol{z} \sim p(\boldsymbol{z})$?
- Let's look at the pdf of $\boldsymbol{z}$'s that were accepted, i.e., $p(\boldsymbol{z}|\text{accept})$

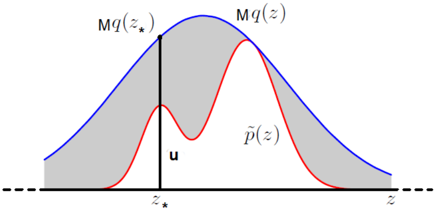
Rejection Sampling



- Why $\boldsymbol{z} \sim q(\boldsymbol{z})$ + accept/reject rule is equivalent to $\boldsymbol{z} \sim p(\boldsymbol{z})$?
- Let's look at the pdf of $\boldsymbol{z}$'s that were accepted, i.e., $p(\boldsymbol{z}|\text{accept})$

$$p(\text{accept}|\boldsymbol{z})$$

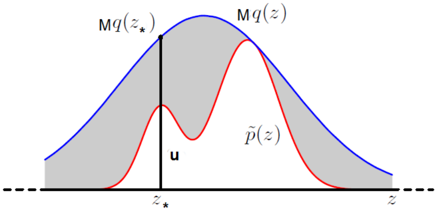
Rejection Sampling



- Why $\boldsymbol{z} \sim q(\boldsymbol{z})$ + accept/reject rule is equivalent to $\boldsymbol{z} \sim p(\boldsymbol{z})$?
- Let's look at the pdf of $\boldsymbol{z}$'s that were accepted, i.e., $p(\boldsymbol{z}|\text{accept})$

$$p(\text{accept}|\boldsymbol{z}) = \int_0^{\tilde{p}(\boldsymbol{z})} \frac{1}{Mq(\boldsymbol{z})} du = \frac{\tilde{p}(\boldsymbol{z})}{Mq(\boldsymbol{z})}$$

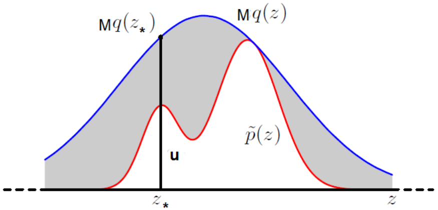
Rejection Sampling



- Why $\boldsymbol{z} \sim q(\boldsymbol{z})$ + accept/reject rule is equivalent to $\boldsymbol{z} \sim p(\boldsymbol{z})$?
- Let's look at the pdf of $\boldsymbol{z}$'s that were accepted, i.e., $p(\boldsymbol{z}|\text{accept})$

$$
\begin{aligned}
p(\text{accept}|\boldsymbol{z}) &= \int_0^{\tilde{p}(z)} \frac{1}{Mq(\boldsymbol{z})} du = \frac{\tilde{p}(\boldsymbol{z})}{Mq(\boldsymbol{z})} \\
p(\boldsymbol{z}, \text{accept}) &= q(\boldsymbol{z}) p(\text{accept}|\boldsymbol{z}) = \frac{\tilde{p}(\boldsymbol{z})}{M}
\end{aligned}
$$

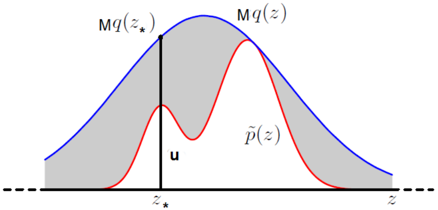
Rejection Sampling



- Why $\boldsymbol{z} \sim q(\boldsymbol{z})$ + accept/reject rule is equivalent to $\boldsymbol{z} \sim p(\boldsymbol{z})$?
- Let's look at the pdf of $\boldsymbol{z}$'s that were accepted, i.e., $p(\boldsymbol{z}|\text{accept})$

$$
\begin{aligned}
p(\text{accept}|\boldsymbol{z}) &= \int_0^{\tilde{p}(\boldsymbol{z})} \frac{1}{Mq(\boldsymbol{z})} du = \frac{\tilde{p}(\boldsymbol{z})}{Mq(\boldsymbol{z})} \\
p(\boldsymbol{z}, \text{accept}) &= q(\boldsymbol{z}) p(\text{accept}|\boldsymbol{z}) = \frac{\tilde{p}(\boldsymbol{z})}{M} \\
p(\text{accept}) &= \int \frac{\tilde{p}(\boldsymbol{z})}{M} d\boldsymbol{z} = \frac{Z_p}{M}
\end{aligned}
$$

Rejection Sampling



- Why $\boldsymbol{z} \sim q(\boldsymbol{z})$ + accept/reject rule is equivalent to $\boldsymbol{z} \sim p(\boldsymbol{z})$?
- Let's look at the pdf of $\boldsymbol{z}$'s that were accepted, i.e., $p(\boldsymbol{z}|\text{accept})$

$$
\begin{aligned}
p(\text{accept}|\boldsymbol{z}) &= \int_0^{\tilde{p}(\boldsymbol{z})} \frac{1}{Mq(\boldsymbol{z})} du = \frac{\tilde{p}(\boldsymbol{z})}{Mq(\boldsymbol{z})} \\
p(\boldsymbol{z}, \text{accept}) &= q(\boldsymbol{z})p(\text{accept}|\boldsymbol{z}) = \frac{\tilde{p}(\boldsymbol{z})}{M} \\
p(\text{accept}) &= \int \frac{\tilde{p}(\boldsymbol{z})}{M} d\boldsymbol{z} = \frac{Z_p}{M} \\
p(\boldsymbol{z}|\text{accept}) &= \frac{p(\boldsymbol{z}, \text{accept})}{p(\text{accept})} = \frac{\tilde{p}(\boldsymbol{z})}{Z_p} = p(\boldsymbol{z})
\end{aligned}
$$

Sampling for Approximating Expectations

- Suppose $f(\boldsymbol{z})$ is function of a random variable $\boldsymbol{z} \sim p(\boldsymbol{z})$

Sampling for Approximating Expectations

- Suppose $f(\boldsymbol{z})$ is function of a random variable $\boldsymbol{z} \sim p(\boldsymbol{z})$
- Wish to compute $\mathbb{E}[f] = \mathbb{E}_{p(\boldsymbol{z})}[f(\boldsymbol{z})] = \int f(\boldsymbol{z})p(\boldsymbol{z})d\boldsymbol{z}$

Sampling for Approximating Expectations

- Suppose $f(\boldsymbol{z})$ is function of a random variable $\boldsymbol{z} \sim p(\boldsymbol{z})$
- Wish to compute $\mathbb{E}[f] = \mathbb{E}_{p(\boldsymbol{z})}[f(\boldsymbol{z})] = \int f(\boldsymbol{z})p(\boldsymbol{z})d\boldsymbol{z}$
- Given L independent samples $\{\boldsymbol{z}^{(\ell)}\}_{\ell=1}^{L}$ from $p(\boldsymbol{z})$, we can approximate the above as

$$\mathbb{E}[f] \approx \frac{1}{L}\sum_{\ell=1}^{L} f(\boldsymbol{z}^{(\ell)}) \quad \text{(Monte Carlo sampling)}$$

Sampling for Approximating Expectations

- Suppose $f(\boldsymbol{z})$ is function of a random variable $\boldsymbol{z} \sim p(\boldsymbol{z})$
- Wish to compute $\mathbb{E}[f] = \mathbb{E}_{p(\boldsymbol{z})}[f(\boldsymbol{z})] = \int f(\boldsymbol{z})p(\boldsymbol{z})d\boldsymbol{z}$
- Given L independent samples $\{\boldsymbol{z}^{(\ell)}\}_{\ell=1}^{L}$ from $p(\boldsymbol{z})$, we can approximate the above as

$$\mathbb{E}[f] \approx \frac{1}{L}\sum_{\ell=1}^{L} f(\boldsymbol{z}^{(\ell)}) \quad \text{(Monte Carlo sampling)}$$

- What if we can't generate samples from $p(\boldsymbol{z})$?

Sampling for Approximating Expectations

- Suppose $f(\boldsymbol{z})$ is function of a random variable $\boldsymbol{z} \sim p(\boldsymbol{z})$
- Wish to compute $\mathbb{E}[f] = \mathbb{E}_{p(\boldsymbol{z})}[f(\boldsymbol{z})] = \int f(\boldsymbol{z})p(\boldsymbol{z})d\boldsymbol{z}$
- Given L independent samples $\{\boldsymbol{z}^{(\ell)}\}_{\ell=1}^{L}$ from $p(\boldsymbol{z})$, we can approximate the above as

$$\mathbb{E}[f] \approx \frac{1}{L}\sum_{\ell=1}^{L} f(\boldsymbol{z}^{(\ell)}) \quad \text{(Monte Carlo sampling)}$$

- What if we can't generate samples from $p(\boldsymbol{z})$? Answer: Use Importance Sampling

Sampling for Approximating Expectations

- Suppose $f(\boldsymbol{z})$ is function of a random variable $\boldsymbol{z} \sim p(\boldsymbol{z})$
- Wish to compute $\mathbb{E}[f] = \mathbb{E}_{p(\boldsymbol{z})}[f(\boldsymbol{z})] = \int f(\boldsymbol{z})p(\boldsymbol{z})d\boldsymbol{z}$
- Given L independent samples $\{\boldsymbol{z}^{(\ell)}\}_{\ell=1}^{L}$ from $p(\boldsymbol{z})$, we can approximate the above as

$$\mathbb{E}[f] \approx \frac{1}{L}\sum_{\ell=1}^{L} f(\boldsymbol{z}^{(\ell)}) \quad \text{(Monte Carlo sampling)}$$

- What if we can't generate samples from $p(\boldsymbol{z})$? Answer: Use Importance Sampling
  - If we can generate L indep. samples $\{\boldsymbol{z}^{(\ell)}\}_{\ell=1}^{L}$ from a different "proposal" distribution $q(\boldsymbol{z})$ then

$$\mathbb{E}[f] = \int f(\boldsymbol{z})p(\boldsymbol{z})d\boldsymbol{z}$$

Sampling for Approximating Expectations

- Suppose $f(\boldsymbol{z})$ is function of a random variable $\boldsymbol{z} \sim p(\boldsymbol{z})$
- Wish to compute $\mathbb{E}[f] = \mathbb{E}_{p(\boldsymbol{z})}[f(\boldsymbol{z})] = \int f(\boldsymbol{z})p(\boldsymbol{z})d\boldsymbol{z}$
- Given L independent samples $\{\boldsymbol{z}^{(\ell)}\}_{\ell=1}^{L}$ from $p(\boldsymbol{z})$, we can approximate the above as

$$\mathbb{E}[f] \approx \frac{1}{L}\sum_{\ell=1}^{L} f(\boldsymbol{z}^{(\ell)}) \quad \text{(Monte Carlo sampling)}$$

- What if we can't generate samples from $p(\boldsymbol{z})$? Answer: Use Importance Sampling
  - If we can generate L indep. samples $\{\boldsymbol{z}^{(\ell)}\}_{\ell=1}^{L}$ from a different "proposal" distribution $q(\boldsymbol{z})$ then

$$\mathbb{E}[f] = \int f(\boldsymbol{z})p(\boldsymbol{z})d\boldsymbol{z} = \int f(\boldsymbol{z})\frac{p(\boldsymbol{z})}{q(\boldsymbol{z})}q(\boldsymbol{z})d\boldsymbol{z}$$

Sampling for Approximating Expectations

- Suppose $f(\boldsymbol{z})$ is function of a random variable $\boldsymbol{z} \sim p(\boldsymbol{z})$
- Wish to compute $\mathbb{E}[f] = \mathbb{E}_{p(\boldsymbol{z})}[f(\boldsymbol{z})] = \int f(\boldsymbol{z})p(\boldsymbol{z})d\boldsymbol{z}$
- Given L independent samples $\{\boldsymbol{z}^{(\ell)}\}_{\ell=1}^{L}$ from $p(\boldsymbol{z})$, we can approximate the above as

$$\mathbb{E}[f] \approx \frac{1}{L}\sum_{\ell=1}^{L} f(\boldsymbol{z}^{(\ell)}) \quad \text{(Monte Carlo sampling)}$$

- What if we can't generate samples from $p(\boldsymbol{z})$? Answer: Use Importance Sampling
  - If we can generate L indep. samples $\{\boldsymbol{z}^{(\ell)}\}_{\ell=1}^{L}$ from a different "proposal" distribution $q(\boldsymbol{z})$ then

$$\mathbb{E}[f] = \int f(\boldsymbol{z})p(\boldsymbol{z})d\boldsymbol{z} = \int f(\boldsymbol{z})\frac{p(\boldsymbol{z})}{q(\boldsymbol{z})}q(\boldsymbol{z})d\boldsymbol{z} \approx \frac{1}{L}\sum_{\ell=1}^{L} f(\boldsymbol{z}^{(\ell)})\frac{p(\boldsymbol{z}^{(\ell)})}{q(\boldsymbol{z}^{(\ell)})}$$

Sampling for Approximating Expectations

- Suppose $f(\boldsymbol{z})$ is function of a random variable $\boldsymbol{z} \sim p(\boldsymbol{z})$
- Wish to compute $\mathbb{E}[f] = \mathbb{E}_{p(\boldsymbol{z})}[f(\boldsymbol{z})] = \int f(\boldsymbol{z})p(\boldsymbol{z})d\boldsymbol{z}$
- Given L independent samples $\{\boldsymbol{z}^{(\ell)}\}_{\ell=1}^{L}$ from $p(\boldsymbol{z})$, we can approximate the above as

$$\mathbb{E}[f] \approx \frac{1}{L}\sum_{\ell=1}^{L} f(\boldsymbol{z}^{(\ell)}) \quad \text{(Monte Carlo sampling)}$$

- What if we can't generate samples from $p(\boldsymbol{z})$? Answer: Use Importance Sampling
  - If we can generate L indep. samples $\{\boldsymbol{z}^{(\ell)}\}_{\ell=1}^{L}$ from a different "proposal" distribution $q(\boldsymbol{z})$ then

$$\mathbb{E}[f] = \int f(\boldsymbol{z})p(\boldsymbol{z})d\boldsymbol{z} = \int f(\boldsymbol{z})\frac{p(\boldsymbol{z})}{q(\boldsymbol{z})}q(\boldsymbol{z})d\boldsymbol{z} \approx \frac{1}{L}\sum_{\ell=1}^{L} f(\boldsymbol{z}^{(\ell)})\frac{p(\boldsymbol{z}^{(\ell)})}{q(\boldsymbol{z}^{(\ell)})}$$

  - IS only requires that we can evaluate $p(\boldsymbol{z})$ at any $\boldsymbol{z}$

Sampling for Approximating Expectations

- Suppose $f(\boldsymbol{z})$ is function of a random variable $\boldsymbol{z} \sim p(\boldsymbol{z})$
- Wish to compute $\mathbb{E}[f] = \mathbb{E}_{p(\boldsymbol{z})}[f(\boldsymbol{z})] = \int f(\boldsymbol{z})p(\boldsymbol{z})d\boldsymbol{z}$
- Given L independent samples $\{\boldsymbol{z}^{(\ell)}\}_{\ell=1}^{L}$ from $p(\boldsymbol{z})$, we can approximate the above as

$$\mathbb{E}[f] \approx \frac{1}{L}\sum_{\ell=1}^{L} f(\boldsymbol{z}^{(\ell)}) \quad \text{(Monte Carlo sampling)}$$

- What if we can't generate samples from $p(\boldsymbol{z})$? Answer: Use Importance Sampling
  - If we can generate L indep. samples $\{\boldsymbol{z}^{(\ell)}\}_{\ell=1}^{L}$ from a different "proposal" distribution $q(\boldsymbol{z})$ then

$$\mathbb{E}[f] = \int f(\boldsymbol{z})p(\boldsymbol{z})d\boldsymbol{z} = \int f(\boldsymbol{z})\frac{p(\boldsymbol{z})}{q(\boldsymbol{z})}q(\boldsymbol{z})d\boldsymbol{z} \approx \frac{1}{L}\sum_{\ell=1}^{L} f(\boldsymbol{z}^{(\ell)})\frac{p(\boldsymbol{z}^{(\ell)})}{q(\boldsymbol{z}^{(\ell)})}$$

  - IS only requires that we can evaluate $p(\boldsymbol{z})$ at any $\boldsymbol{z}$ (in fact, with a small modification to the above, IS works even when we can evaluate $p(\boldsymbol{z})$ only up to a proportionality constant)

Sampling for Approximating Expectations

- Suppose $f(\boldsymbol{z})$ is function of a random variable $\boldsymbol{z} \sim p(\boldsymbol{z})$
- Wish to compute $\mathbb{E}[f] = \mathbb{E}_{p(\boldsymbol{z})}[f(\boldsymbol{z})] = \int f(\boldsymbol{z})p(\boldsymbol{z})d\boldsymbol{z}$
- Given L independent samples $\{\boldsymbol{z}^{(\ell)}\}_{\ell=1}^{L}$ from $p(\boldsymbol{z})$, we can approximate the above as

$$\mathbb{E}[f] \approx \frac{1}{L}\sum_{\ell=1}^{L} f(\boldsymbol{z}^{(\ell)}) \quad \text{(Monte Carlo sampling)}$$

- What if we can't generate samples from $p(\boldsymbol{z})$? Answer: Use Importance Sampling
  - If we can generate L indep. samples $\{\boldsymbol{z}^{(\ell)}\}_{\ell=1}^{L}$ from a different "proposal" distribution $q(\boldsymbol{z})$ then

$$\mathbb{E}[f] = \int f(\boldsymbol{z})p(\boldsymbol{z})d\boldsymbol{z} = \int f(\boldsymbol{z})\frac{p(\boldsymbol{z})}{q(\boldsymbol{z})}q(\boldsymbol{z})d\boldsymbol{z} \approx \frac{1}{L}\sum_{\ell=1}^{L} f(\boldsymbol{z}^{(\ell)})\frac{p(\boldsymbol{z}^{(\ell)})}{q(\boldsymbol{z}^{(\ell)})}$$

  - IS only requires that we can evaluate $p(\boldsymbol{z})$ at any $\boldsymbol{z}$ (in fact, with a small modification to the above, IS works even when we can evaluate $p(\boldsymbol{z})$ only up to a proportionality constant)
  - Note: IS is NOT a sampling method (doesn't generate samples from a desired distribution; just a way to approximate expectations)

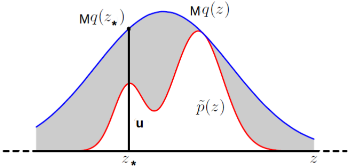
Limitations of Basic Sampling Methods

Limitations of Basic Sampling Methods

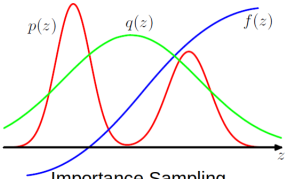
- Transformation based methods: Usually limited to drawing from standard distributions

Limitations of Basic Sampling Methods

- Transformation based methods: Usually limited to drawing from standard distributions
- Rejection Sampling and Importance Sampling: Require good proposal distributions



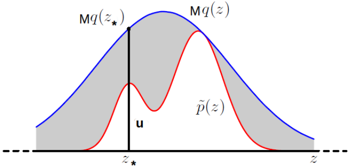
Rejection Sampling



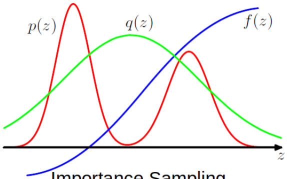
Importance Sampling

Limitations of Basic Sampling Methods

- Transformation based methods: Usually limited to drawing from standard distributions
- Rejection Sampling and Importance Sampling: Require good proposal distributions



Rejection Sampling

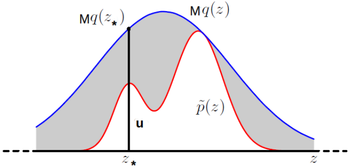


Importance Sampling

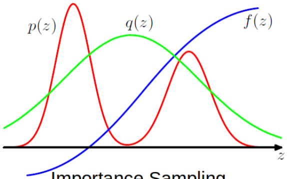
- Difficult to find good prop. distr. especially when $\boldsymbol{z}$ is high-dim. (e.g., models with many params)

Limitations of Basic Sampling Methods

- Transformation based methods: Usually limited to drawing from standard distributions
- Rejection Sampling and Importance Sampling: Require good proposal distributions



Rejection Sampling

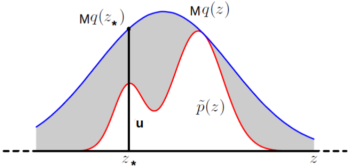


Importance Sampling

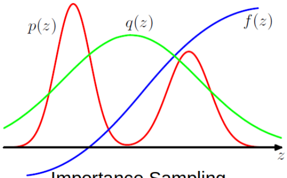
- Difficult to find good prop. distr. especially when $\boldsymbol{z}$ is high-dim. (e.g., models with many params)
  - In high dimensions, most of the mass of $p(\boldsymbol{z})$ is concentrated in a tiny region of the $\boldsymbol{z}$ space

Limitations of Basic Sampling Methods

- Transformation based methods: Usually limited to drawing from standard distributions
- Rejection Sampling and Importance Sampling: Require good proposal distributions



Rejection Sampling



Importance Sampling

- Difficult to find good prop. distr. especially when $\boldsymbol{z}$ is high-dim. (e.g., models with many params)
  - In high dimensions, most of the mass of $p(\boldsymbol{z})$ is concentrated in a tiny region of the $\boldsymbol{z}$ space
  - Difficult to *a priori* know what those regions are, thus difficult to come up with good proposal dist.
- A solution to these: MCMC methods