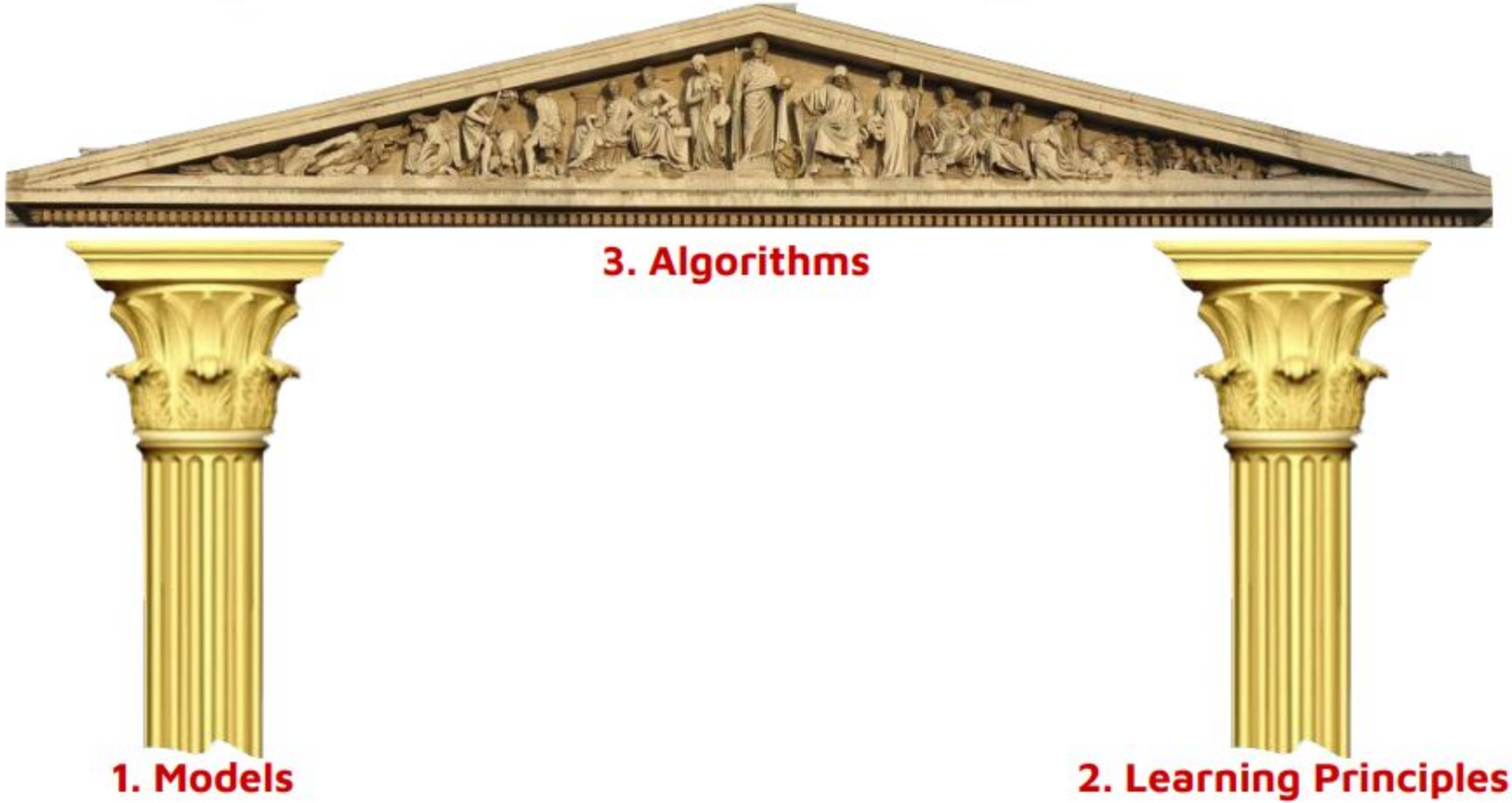
Variational Inference (VI)
and
Discrete Latent Structure

Fanyun.Sun 2019.04.12

Main Source:

- https://www.shakirm.com/slides/DeepGenModelsTutorial.pdf
- http://www.cs.columbia.edu/~blei/talks/2016_NIPS_VI_tutorial.pdf
- https://duvenaud.github.io/learn-discrete/

Thinking about Machine Learning



Combining Models and Inference

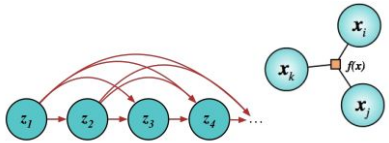
3. Algorithms

A given model and learning principle can be implemented in many ways.

1. Models

Fully-observed models

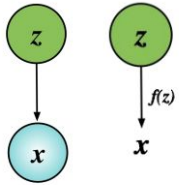
Model observed data directly without introducing any new unobserved local variables.



Latent Variable Models

Introduce an unobserved random variable for every observed data point to explain hidden causes.

- **Prescribed models:** Use observer likelihoods and assume observation noise.
- **Implicit models:** Likelihood-free models.



Choice of Learning Principles

For a given model, there are many competing inference methods.

- Exact methods (conjugacy, enumeration)
- Numerical integration (Quadrature)
- Generalised method of moments
- **Maximum likelihood (ML)**
- Maximum a posteriori (MAP)
- Laplace approximation
- Integrated nested Laplace approximations (INLA)
- **Expectation Maximisation (EM)**
- Monte Carlo methods (MCMC, SMC, ABC)
- Contrastive estimation (NCE)
- Cavity Methods (EP)
- **Variational methods**

2. Learning Principles

Variational Methods
+
Latent Variable Models !!!

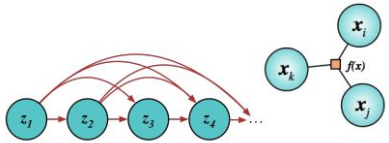
Combining Models and Inference

A given model and learning principle can be implemented in many ways.

1. Models

Fully-observed models

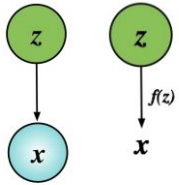
Model observed data directly without introducing any new unobserved local variables.



Latent Variable Models

Introduce an unobserved random variable for every observed data point to explain hidden causes.

- **Prescribed models:** Use observer likelihoods and assume observation noise.
- **Implicit models:** Likelihood-free models.



Choice of Learning Principles

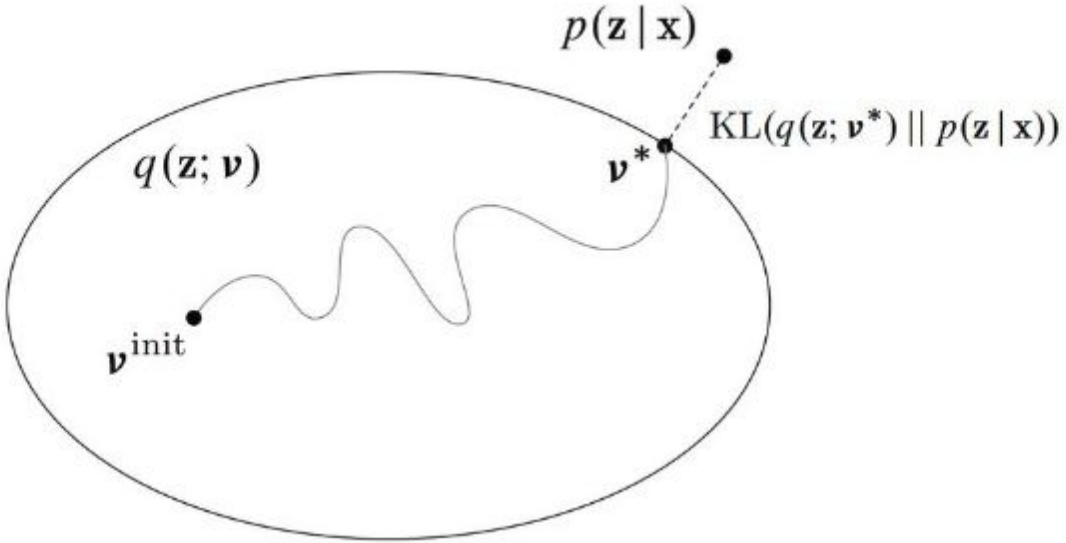
For a given model, there are many competing inference methods.

- Exact methods (conjugacy, enumeration)
- Numerical integration (Quadrature)
- Generalised method of moments
- **Maximum likelihood (ML)**
- Maximum a posteriori (MAP)
- Laplace approximation
- Integrated nested Laplace approximations (INLA)
- **Expectation Maximisation (EM)**
- Monte Carlo methods (MCMC, SMC, ABC)
- Contrastive estimation (NCE)
- Cavity Methods (EP)
- **Variational methods**

2. Learning Principles

Advantages of latent variable models

- Model checking by sampling
- Natural way to specify models
- Compact representations
- Semi-Supervised learning
- Understanding factors of variation in data



- VI turns **inference into optimization**.
- Posit a **variational family** of distributions over the latent variables,

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

- Fit the **variational parameters** $\boldsymbol{\nu}$ to be close (in KL) to the exact posterior.
  (There are alternative divergences, which connect to algorithms like EP, BP, and others.)

Variational Inference Recipe

- Start with data ***X*** and a model ***p(z, x)***, we are interested in ***p(z | x)***
- Choose a variational approximation ***q(z | x; v)*** **(approximate posteriors)**
- By maximizing likelihood ***log p(X)***, we derive ELBO

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

- How to obtain gradients? **(variational optimization)**

Example Using Gaussian mixture Model: https://zhiyzuo.github.io/VI/#general-situation

Design Choices

Choice of Model

Computation graphs, Renderers, simulators and environments

Variational Optimisation

- Variational EM
- Stochastic VEM
- Monte Carlo gradient estimators

Approximate Posteriors

- Mean-field
- Structured approx
- Aux. variable methods

Design Choices

Choice of Model

Computation graphs, Renderers, simulators and environments

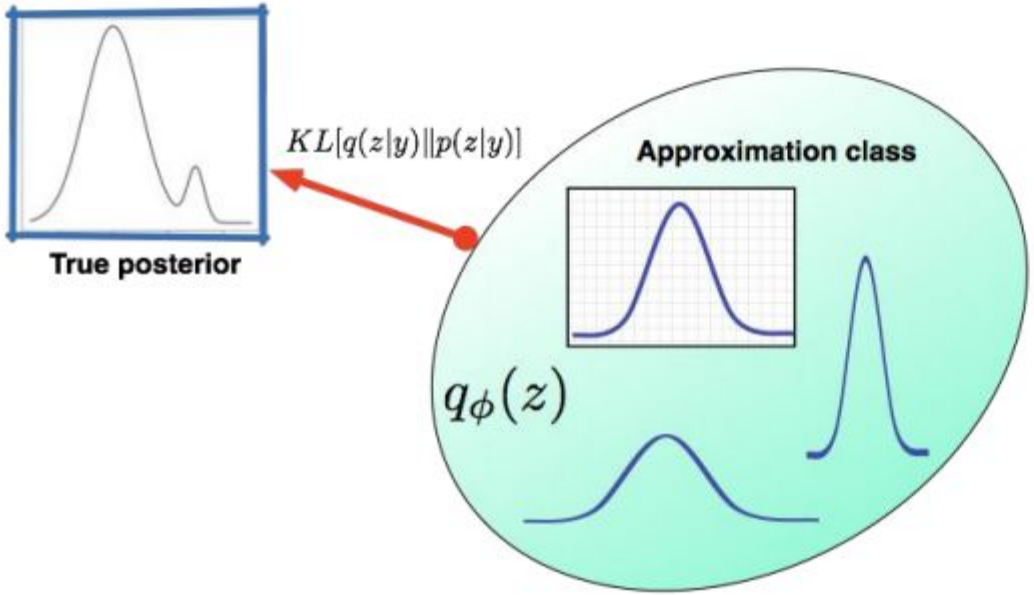
Variational Optimisation

- Variational EM
- Stochastic VEM
- Monte Carlo gradient estimators

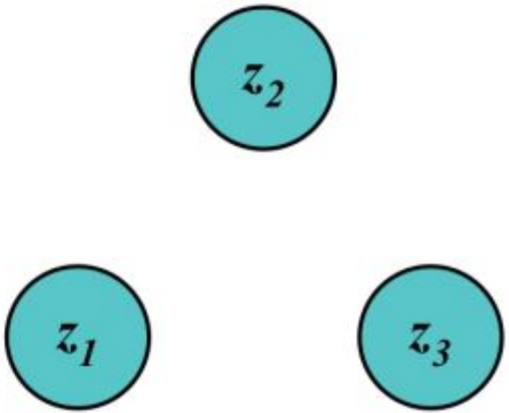
Approximate Posteriors

- Mean-field
- Structured approx
- Aux. variable methods

Mean Field Approximations



Fully-factorised

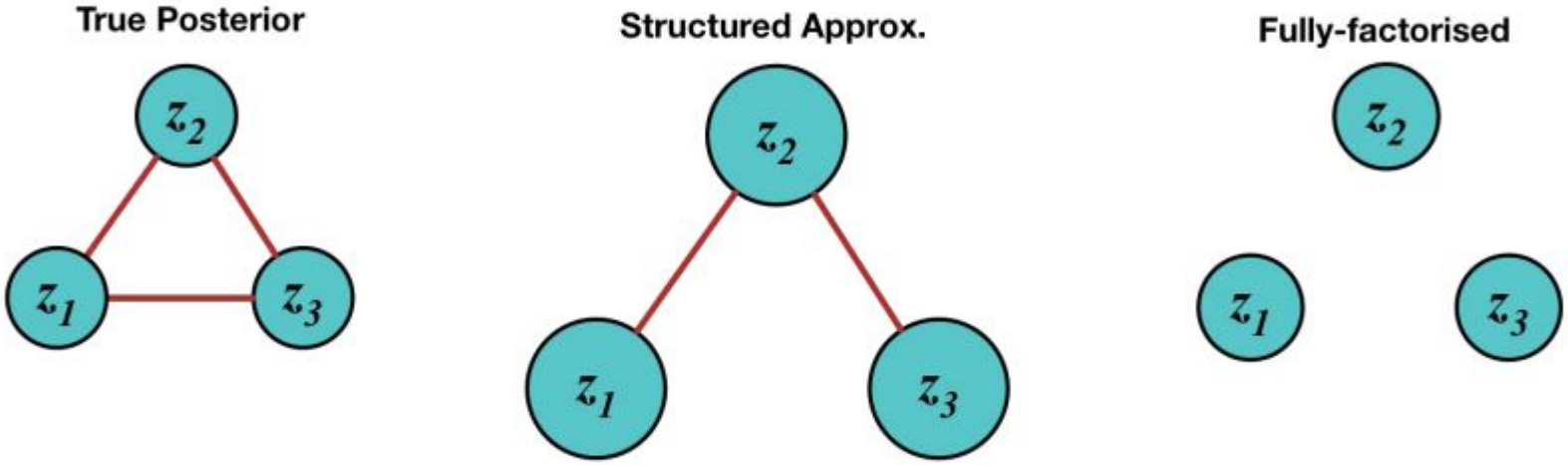


$$q_{MF}(z|x) = \prod_k q(z_k)$$

Key part of variational inference is choice of approximate posterior distribution q.

$$\mathcal{F}(q, \boldsymbol{\theta}) = \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(\mathbf{x}|\mathbf{z})] - \mathrm{KL}[q_\phi(\mathbf{z}|\mathbf{x}|)||p(\mathbf{z})]$$

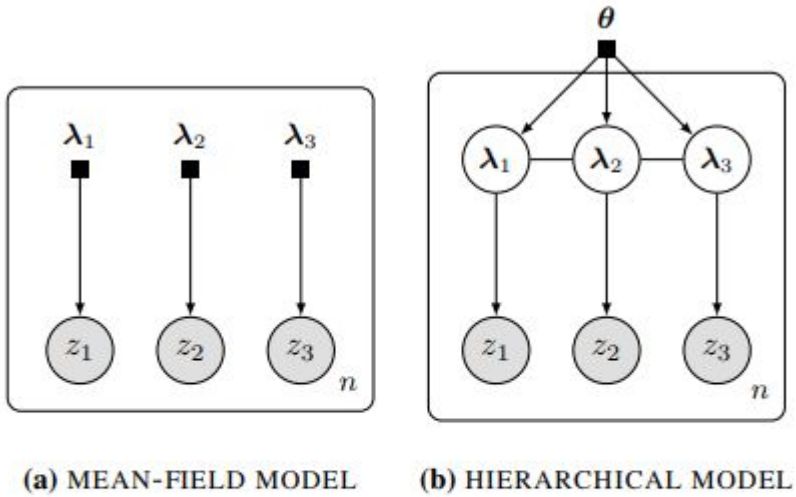
Structured Approximations



Most Expressive ← → *Least Expressive*

$$q^*(z|x) \propto p(x|z)p(z) \qquad q(z) = \prod_k q_k(z_k|\{z_j\}_{j\neq k}) \qquad q_{MF}(z|x) = \prod_k q(z_k)$$

Hierarchical Variational Models Rajesh Ranganath, Dustin Tran, David M. Blei ICML 2016



(a) MEAN-FIELD MODEL **(b)** HIERARCHICAL MODEL

Figure 1. Graphical model representation. **(a)** In mean-field models, the latent variables are strictly independent. **(b)** In hierarchical variational models, the latent variables are governed by a prior distribution on their parameters, which induces arbitrarily complex structure.

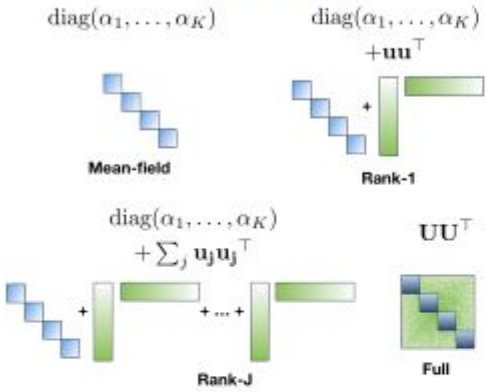
Hierarchical variational models: prior $q(\lambda; \theta)$, likelihood $\prod_i q(\mathbf{z}_i \mid \lambda_i)$.

$$q(\mathbf{z}; \theta) = \int \Big[\prod_i q(\mathbf{z}_i \mid \lambda_i) \Big] q(\lambda; \theta) \, \mathrm{d}\lambda$$

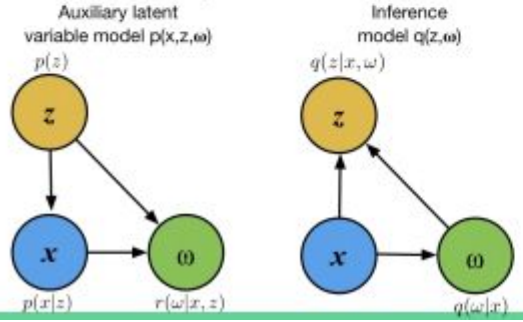
- Hierarchical variational models unify other expressive approximations (mixture, structured, MCMC, copula,...).
- Their expressiveness is determined by the complexity of the prior $q(\lambda)$.

Families of Approximate Posteriors

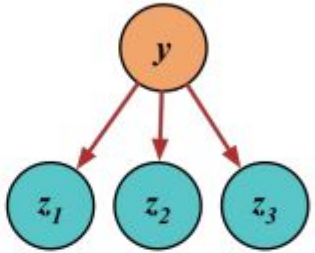
Covariance Models



Auxiliary Variable Models

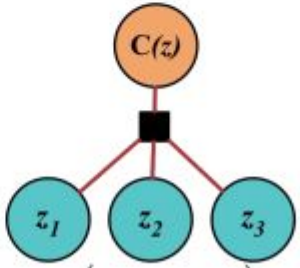


Mixture model



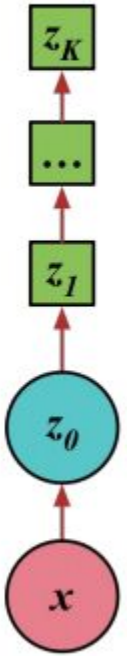
$$q_{mm}(\mathbf{z};\boldsymbol{\nu}) = \sum_r \rho_r q_r(\mathbf{z}_r|\boldsymbol{\nu}_r)$$

Copula Methods



$$q_{lm}(\mathbf{z};\boldsymbol{\nu}) = \left(\prod_k q_k(z_k|\boldsymbol{\nu}_k)\right) C(\mathbf{z};\boldsymbol{\nu}_{k+1})$$

Normalising Flows



Design Choices

Choice of Model

Computation graphs, Renderers, simulators and environments

Variational Optimisation

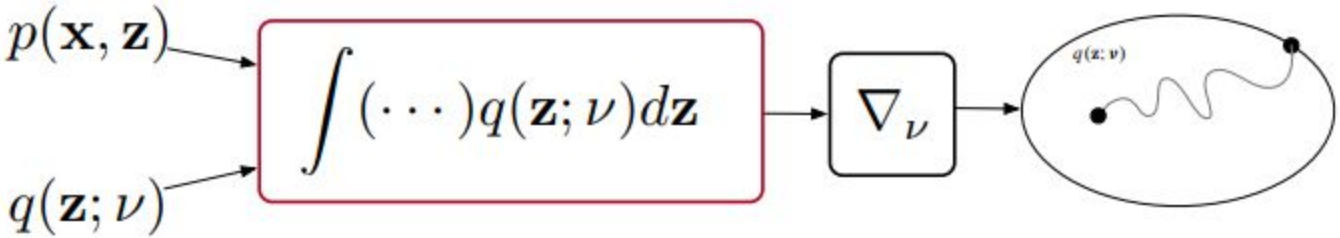
- Variational EM
- Stochastic VEM
- Monte Carlo gradient estimators

Approximate Posteriors

- Mean-field
- Structured approx
- Aux. variable methods

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

Classical inference →very rarely can this gradient be computed analytically

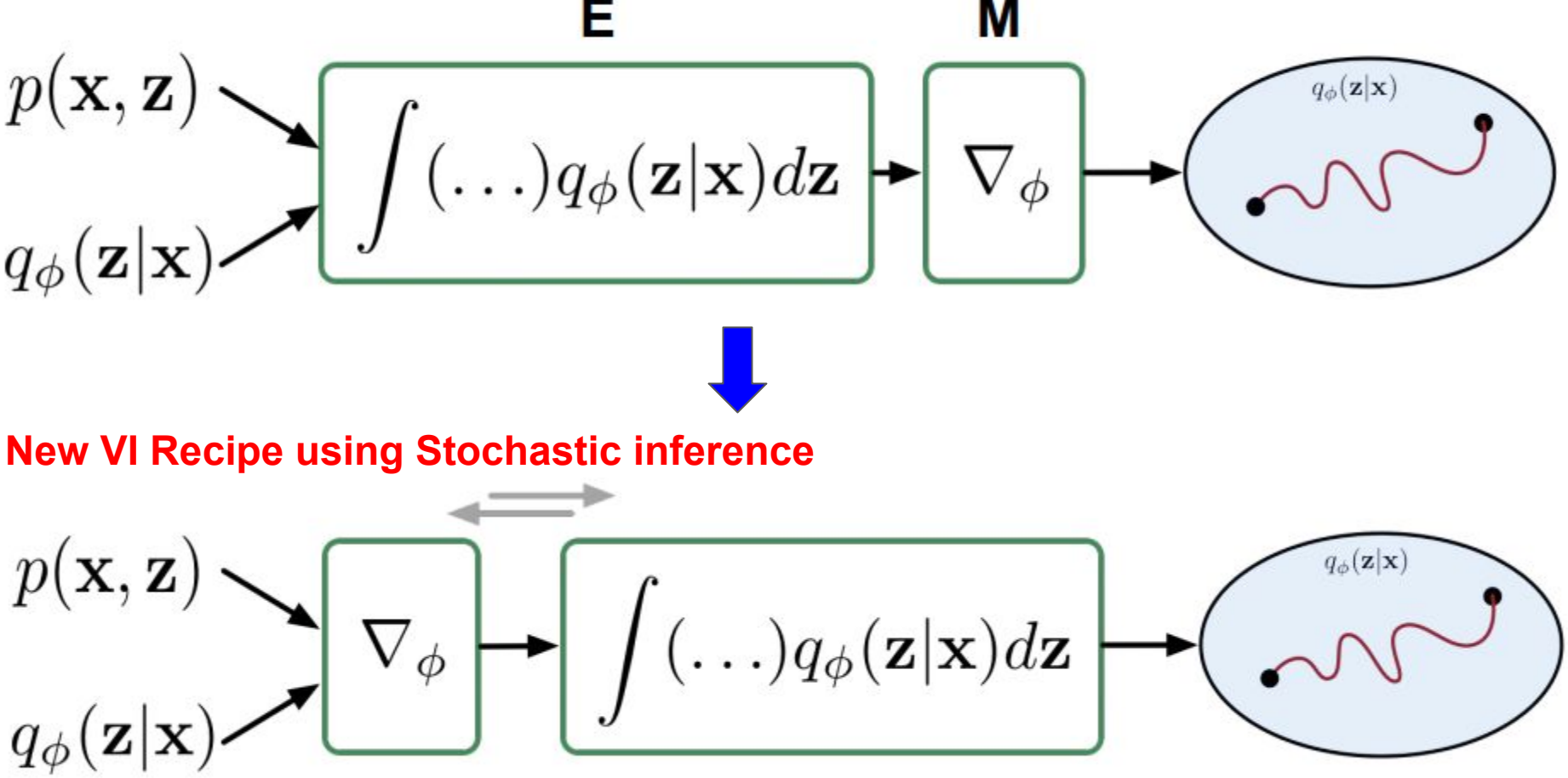


→ so we **estimate** !!!

$$\mathcal{L}(\theta) = \mathbb{E}_{p(b|\theta)}[f(b)]$$

- Unbiased
- Low variance
- Usable when ***f(b)*** is unknown (RL)
- Usable when ***p(b|θ)*** is discrete

$$\text{argmax}_\theta E_{\tau \sim \pi(\tau|\theta)}[R(\tau)]$$


E
M
$p(\mathbf{x}, \mathbf{z})$
$q_\phi(\mathbf{z}|\mathbf{x})$
$\int (\ldots) q_\phi(\mathbf{z}|\mathbf{x}) d\mathbf{z}$
∇_ϕ
$q_\phi(\mathbf{z}|\mathbf{x})$
New VI Recipe using Stochastic inference
$p(\mathbf{x}, \mathbf{z})$
$q_\phi(\mathbf{z}|\mathbf{x})$
∇_ϕ
$\int (\ldots) q_\phi(\mathbf{z}|\mathbf{x}) d\mathbf{z}$
$q_\phi(\mathbf{z}|\mathbf{x})$

Machine Learning

Latent Variable Model + Variational Inference

Variational Inference Recipe

Approximate Posteriors

- Mean-field
- Structured Approx.
- Hierarchical variational methods
- ...

Variational Optimization

Stochastic Optimization
(Stochastic Gradient Estimator)

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

$$\nabla_\phi \mathbb{E}_{q_\phi(\mathbf{z})}[f_\theta(\mathbf{z})]$$

Score Function Estimator

Pathwise Derivative Estimator

Stochastic Gradient Estimators

$$\nabla_\phi \mathbb{E}_{q_\phi(\mathbf{z})}[f_\theta(\mathbf{z})] = \nabla \int q_\phi(\mathbf{z}) f_\theta(\mathbf{z}) d\mathbf{z}$$

Score-function estimator:
Differentiate the density $q(\mathbf{z}|\mathbf{x})$

Pathwise gradient estimator:
Differentiate the function $f(\mathbf{z})$

Typical problem areas:

- Generative models and inference
- Reinforcement learning and control
- Operations research and inventory control
- Monte Carlo simulation
- Finance and asset pricing
- Sensitivity estimation

Score Function Estimators

$$\nabla_\phi \mathbb{E}_{q_\phi(\mathbf{z})}[f_\theta(\mathbf{z})] = \nabla \int q_\phi(\mathbf{z}) f_\theta(\mathbf{z}) d\mathbf{z}$$

$$= \mathbb{E}_{q(z)}[f_\theta(\mathbf{z}) \nabla_\phi \log q_\phi(\mathbf{z}))]$$

Gradient reweighted by the value of the function

Other names:

- Likelihood-ratio trick
- Radon-Nikodym derivative
- REINFORCE and policy gradients
- Automated inference
- Black-box inference

When to use:

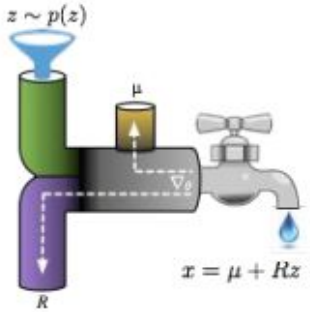
- Function is not differentiable.
- Distribution q is easy to sample from.
- Density q is known and differentiable.

REINFORCE (Williams, 1992)

$$\hat{g}_{\text{REINFORCE}}[f] = f\,(b)\,\frac{\partial}{\partial\theta}\log p(b|\theta), \qquad b \sim p(b|\theta)$$

- Unbiased
- Has few requirements
- Easy to compute
- Suffers from high variance

Pathwise Derivative Estimator



$$\mathbf{z} = g(\epsilon, \phi) \quad \epsilon \sim p(\epsilon)$$

$$\nabla_\phi \mathbb{E}_{q_\phi(\mathbf{z})}[f_\theta(\mathbf{z})] = \nabla \int q_\phi(\mathbf{z}) f_\theta(\mathbf{z}) d\mathbf{z}$$

$$= \mathbb{E}_{p(\epsilon)}[\nabla_\phi f_\theta(g(\epsilon, \phi))]$$

Other names:

- Reparameterisation trick
- Stochastic backpropagation
- Perturbation analysis
- Affine-independent inference
- Doubly stochastic estimation
- Hierarchical non-centred parameterisations.

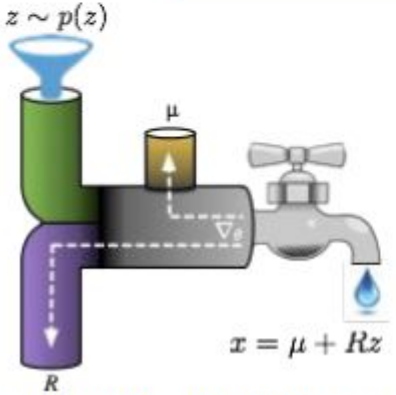
When to use

- Function *f* is differentiable
- Density *q* can be described using a simpler base distribution: inverse CDF, location-scale transform, or other co-ordinate transform.
- Easy to sample from base distribution.

Reparameterisation

$$\nabla_\phi \mathbb{E}_{q_\phi(\mathbf{z})}[f_\theta(\mathbf{z})] = \nabla \int q_\phi(\mathbf{z}) f_\theta(\mathbf{z}) d\mathbf{z}$$

Find an invertible function *g(.)* that expresses z as a transformation of a base distribution .



$$\mathbf{z} = g_\phi(\boldsymbol{\epsilon}) \qquad \boldsymbol{\epsilon} \sim p(\boldsymbol{\epsilon})$$

$$\mathbb{E}_{q_\phi(z|x)}[f(z)] = \mathbb{E}_{p(\epsilon)}[f(g_\phi(x, \epsilon))]$$

Kingma and Welling, 2014, Rezende et al., 2014

Score Function Estimator vs. Pathwise Estimator

Score Function

- Differentiates the density $\nabla_{\nu} q(z; \nu)$
- Works for discrete and continuous models
- Works for large class of variational approximations
- Variance can be a big problem

Pathwise

- Differentiates the function $\nabla_z [\log p(\mathbf{x}, \mathbf{z}) - \log q(\mathbf{z}; \nu)]$
- Requires differentiable models
- Requires variational approximation to have form $\mathbf{z} = t(\epsilon, \nu)$
- Generally better behaved variance

Machine Learning

Latent Variable Model + Variational Inference

Variational Inference Recipe

Approximate Posteriors

- Mean-field
- Structured Approx.
- ...

Variational Optimization

Stochastic Optimization
(Stochastic Gradient Estimator)

Score Function Estimator

- REINFORCE
- Black box Inference

Pathwise Derivative Estimator

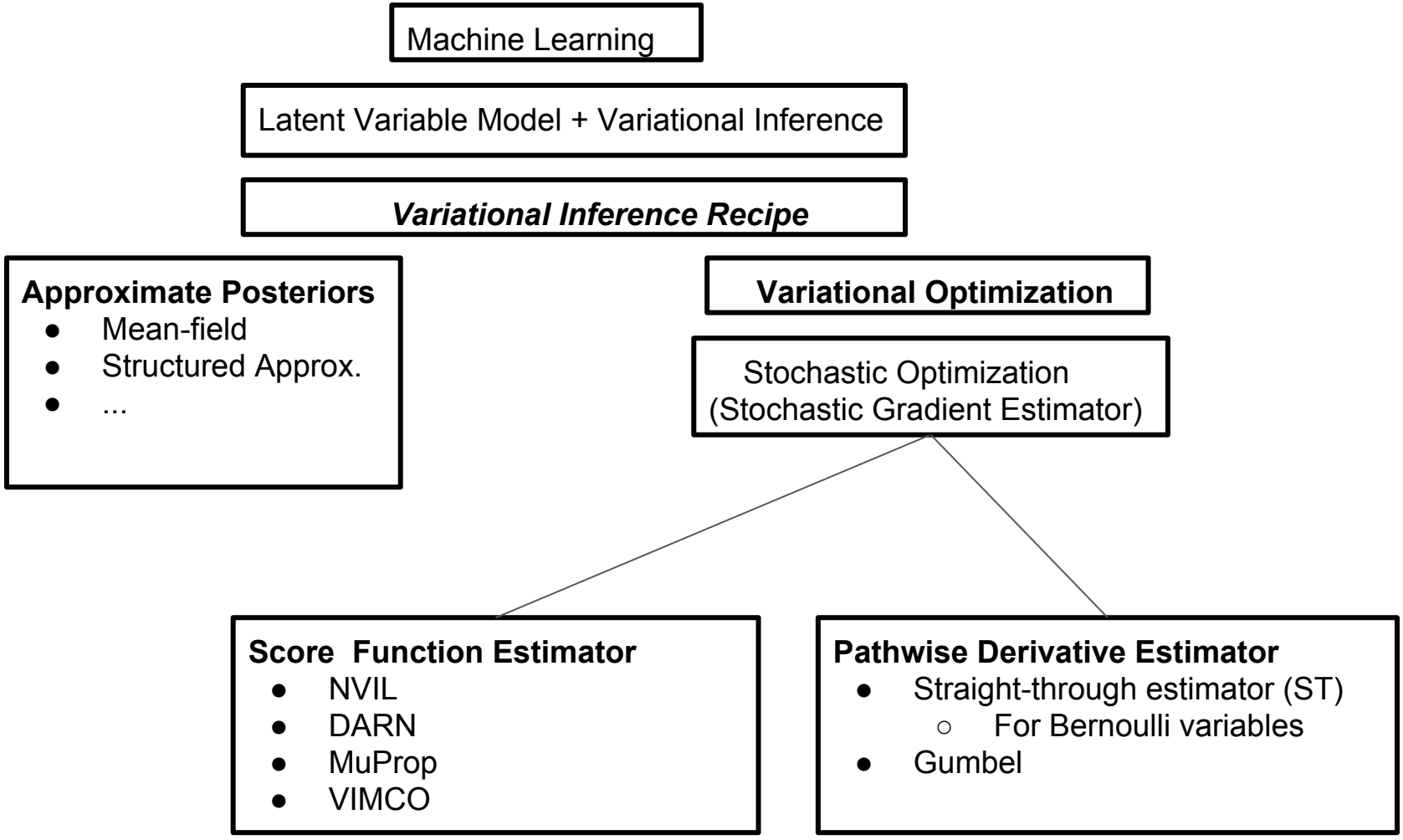
- Reparameterization Trick

Why discrete latent structure?

- **Computational efficiency** - Making models fully differentiable sometimes requires us to sum over all possibilities to compute gradients, for instance in soft attention models. Making hard choices about which computation to perform breaks differentiability, but is faster and requires less memory.

- **Reinforcement learning** - In many domains, the set of possible actions is discrete. Planning and learning in these domains requires integrating over possible future actions.

- **Interpretability and Communication** - Models with millions of continuous parameters, or vector-valued latent states, are usually hard to interpret. Discrete structure is easier to communicate using language. Conversely, communicating using words is an example of learning and planning in a discrete domain.


Machine Learning
Latent Variable Model + Variational Inference
Variational Inference Recipe
Approximate Posteriors
Mean-field
Structured Approx.
...
Variational Optimization
Stochastic Optimization
(Stochastic Gradient Estimator)
Score Function Estimator
NVIL
DARN
MuProp
VIMCO
Pathwise Derivative Estimator
Straight-through estimator (ST)
For Bernoulli variables
Gumbel

Score Function Estimator

- NVIL
- DARN
- MuProp
- VIMCO(multi-sample)

→ Control variates !

→ Variance reduction technique used in Monte Carlo Methods

Control Variates

- We are interested in $\mathbb{E}[m]$ while reducing $\mathrm{Var}(m)$

- Introduce another statistic such that $\mathbb{E}[t] = \tau$ and let $m^\star = m + c(t - \tau)$

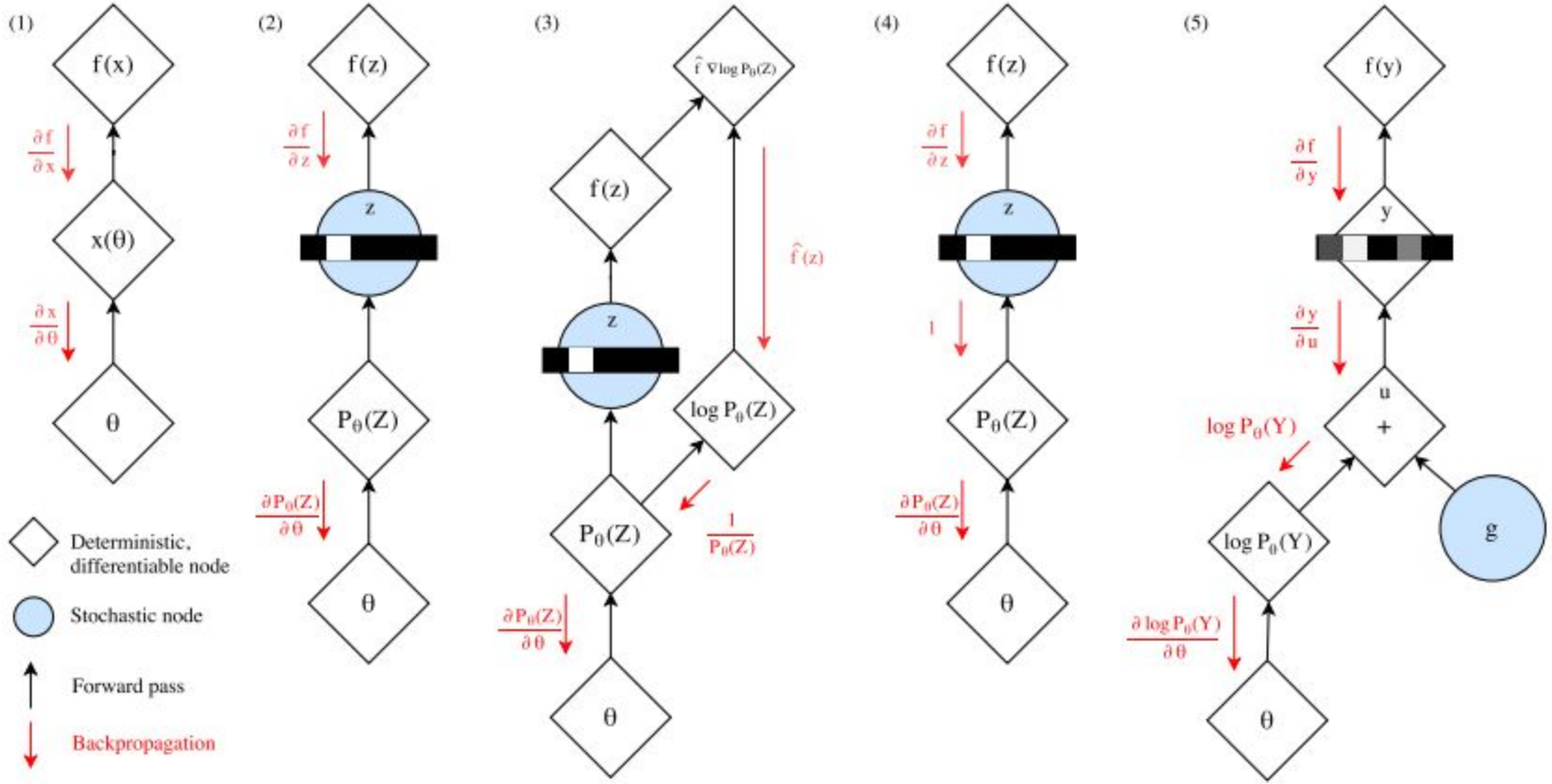
$$\mathrm{Var}(m^\star) = \mathrm{Var}(m) + c^2 \mathrm{Var}(t) + 2c \mathrm{Cov}(m, t).$$

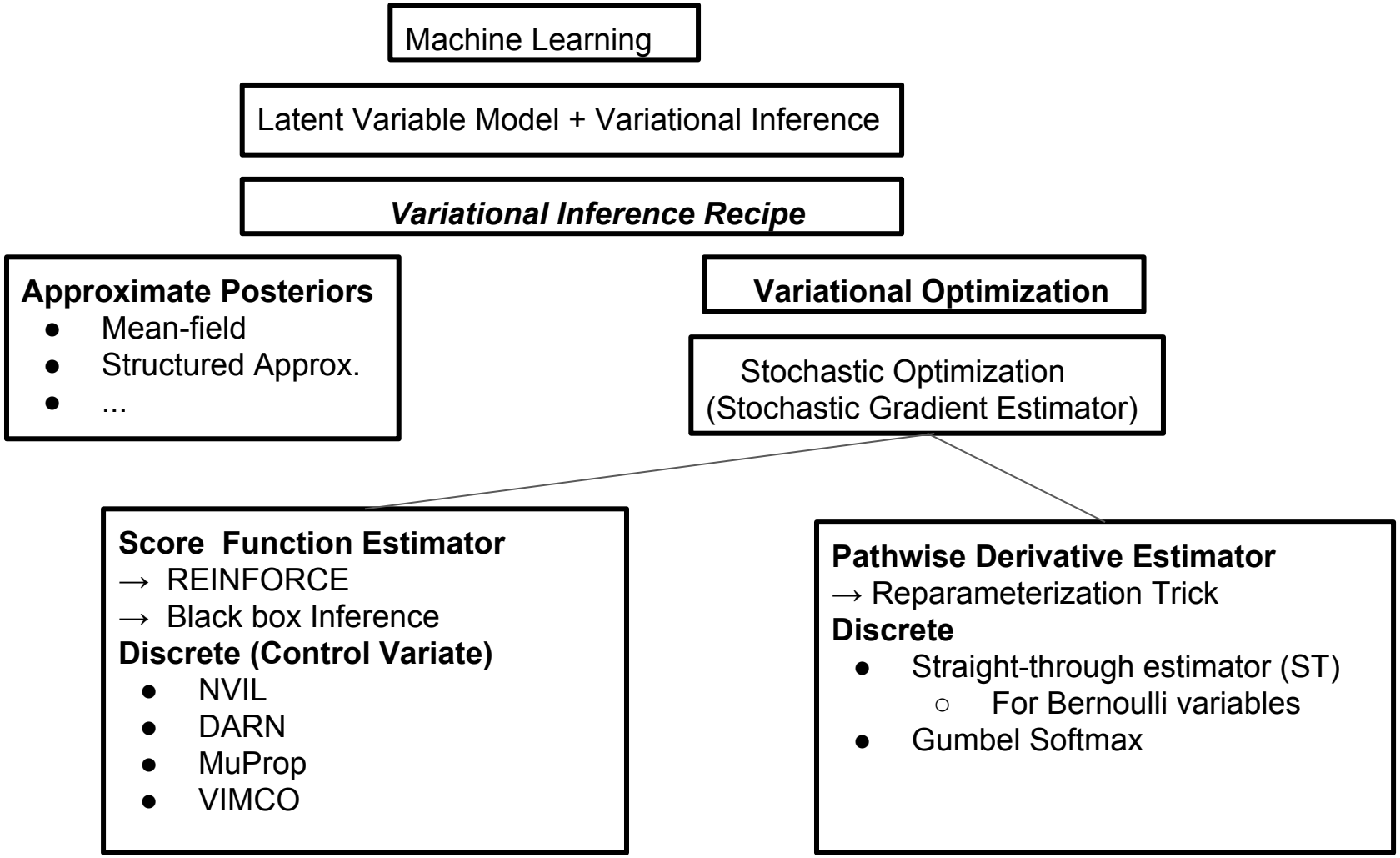
$$c^\star = -\frac{\mathrm{Cov}(m, t)}{\mathrm{Var}(t)}$$

$$\begin{aligned}\mathrm{Var}(m^\star) &= \mathrm{Var}(m) - \frac{[\mathrm{Cov}(m, t)]^2}{\mathrm{Var}(t)} \\ &= \left(1 - \rho_{m,t}^2\right) \mathrm{Var}(m)\end{aligned}$$

where

$$\rho_{m,t} = \mathrm{Corr}(m, t)$$


(1)
f(x)
$\frac{\partial f}{\partial x}$
x(θ)
$\frac{\partial x}{\partial \theta}$
θ
(2)
f(z)
$\frac{\partial f}{\partial z}$
z
$P_\theta(Z)$
$\frac{\partial P_\theta(Z)}{\partial \theta}$
θ
(3)
$\hat{f} \nabla \log P_\theta(Z)$
f(z)
$\hat{f}(z)$
z
$\log P_\theta(Z)$
$P_\theta(Z)$
$\frac{1}{P_\theta(Z)}$
$\frac{\partial P_\theta(Z)}{\partial \theta}$
θ
(4)
f(z)
$\frac{\partial f}{\partial z}$
z
1
$P_\theta(Z)$
$\frac{\partial P_\theta(Z)}{\partial \theta}$
θ
(5)
f(y)
$\frac{\partial f}{\partial y}$
y
$\frac{\partial y}{\partial u}$
u
+
$\log P_\theta(Y)$
g
$\log P_\theta(Y)$
$\frac{\partial \log P_\theta(Y)}{\partial \theta}$
θ
Deterministic, differentiable node
Stochastic node
Forward pass
Backpropagation


Machine Learning
Latent Variable Model + Variational Inference
Variational Inference Recipe
Approximate Posteriors
● Mean-field
● Structured Approx.
● ...
Variational Optimization
Stochastic Optimization
(Stochastic Gradient Estimator)
Score Function Estimator
→ REINFORCE
→ Black box Inference
Discrete (Control Variate)
● NVIL
● DARN
● MuProp
● VIMCO
Pathwise Derivative Estimator
→ Reparameterization Trick
Discrete
● Straight-through estimator (ST)
○ For Bernoulli variables
● Gumbel Softmax

References

- https://www.shakirm.com/slides/DeepGenModelsTutorial.pdf
- http://www.cs.columbia.edu/~blei/talks/2016_NIPS_VI_tutorial.pdf
- https://duvenaud.github.io/learn-discrete/
- Back propagation through void
- Advances in Variational Inference