AI and Research

Bayesian Learning

Basics and Advances in Modeling and Inference

Chang Liu

<changliu@microsoft.com>

Microsoft

Outline

- Overview
  - Bayesian Models
  - Bayesian Inference
- Bayesian Inference
  - Variational Inference
    - Parametric Variational Inference
    - Particle-Based Variational Inference
  - MCMC
  - Amortized Inference
- Bayesian Models
  - Bayesian Networks
    - Sigmoid Belief Networks
    - Topic Models
    - Variational Auto-Encoders
  - Bayesian Neural Networks
  - Markov Random Fields
- Topics
  - Online inference
  - Asymmetry of KL
  - Bayesian Reinforcement Learning
  - Causality
- Future Work

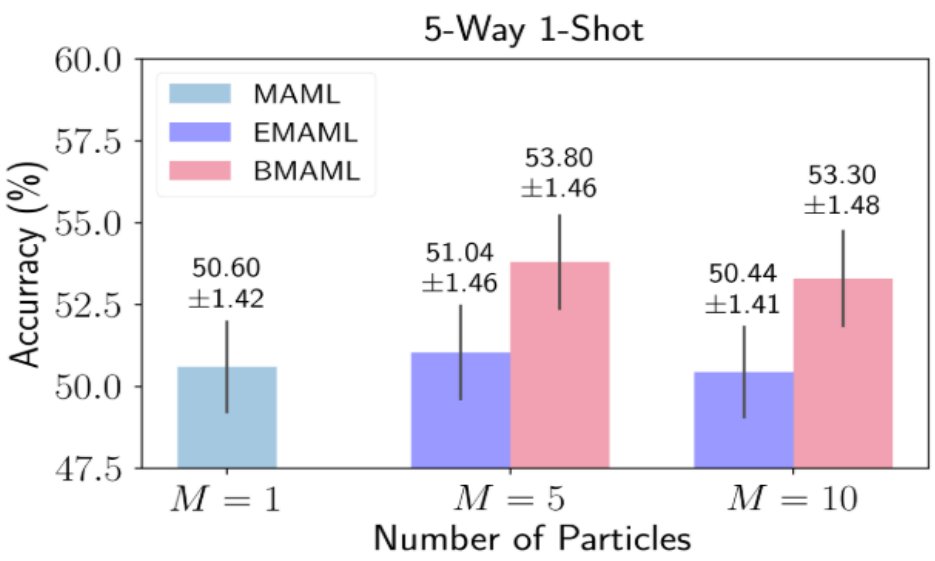
Overview

Bayesian Models

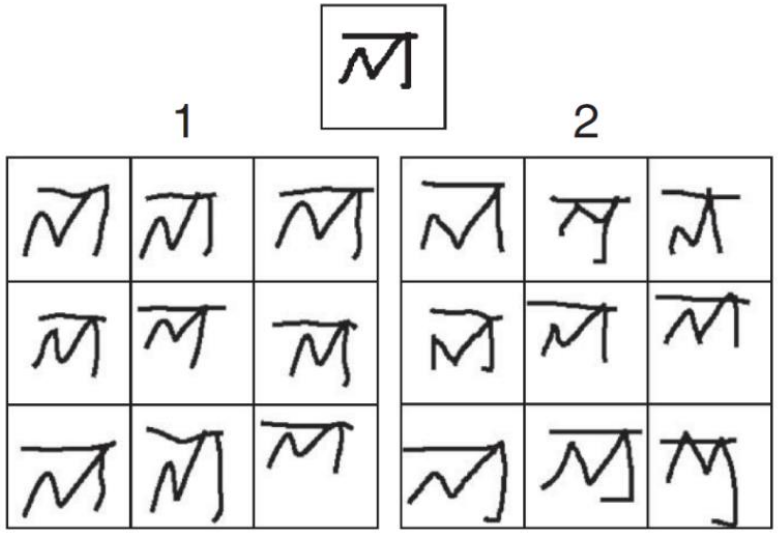
Bayesian Model: Overview

The dependence between latent variable and data is probabilistic.

- Handles model complexity and uncertainty
- Robust to small data...



Meta-learning [KYD+18]

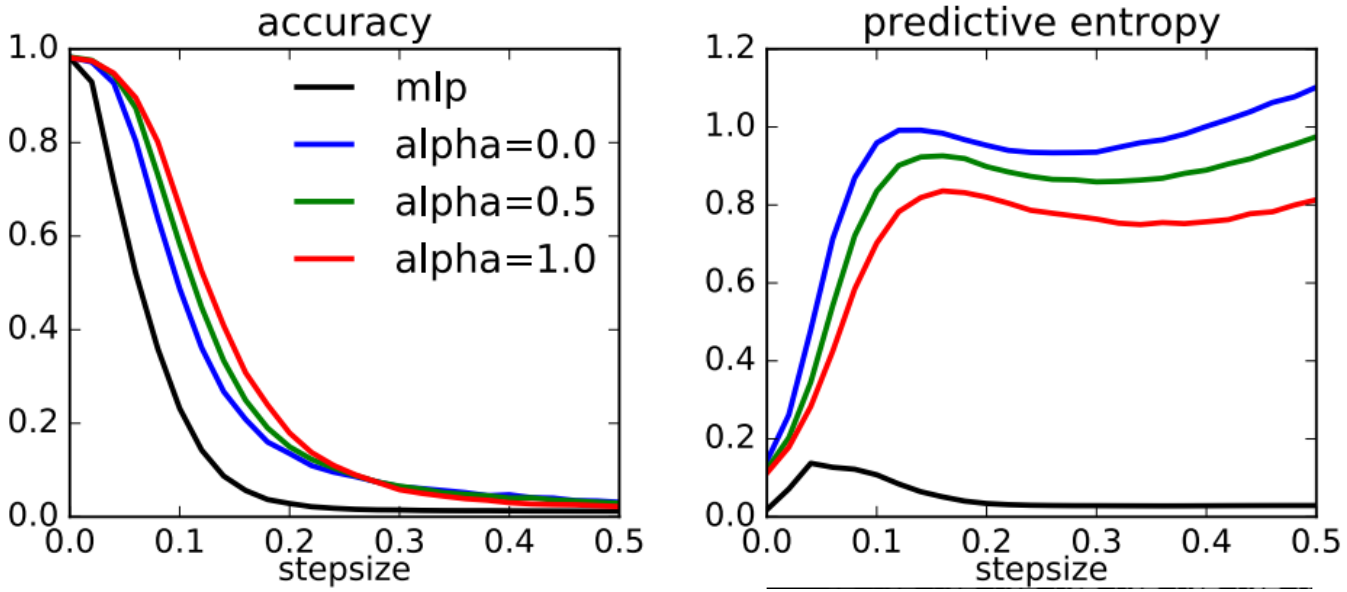


One-shot generation [LST15]

Bayesian Model: Overview

The dependence between latent variable and data is probabilistic.

- Handles model complexity and uncertainty
- Robust to small data and adversarial attack

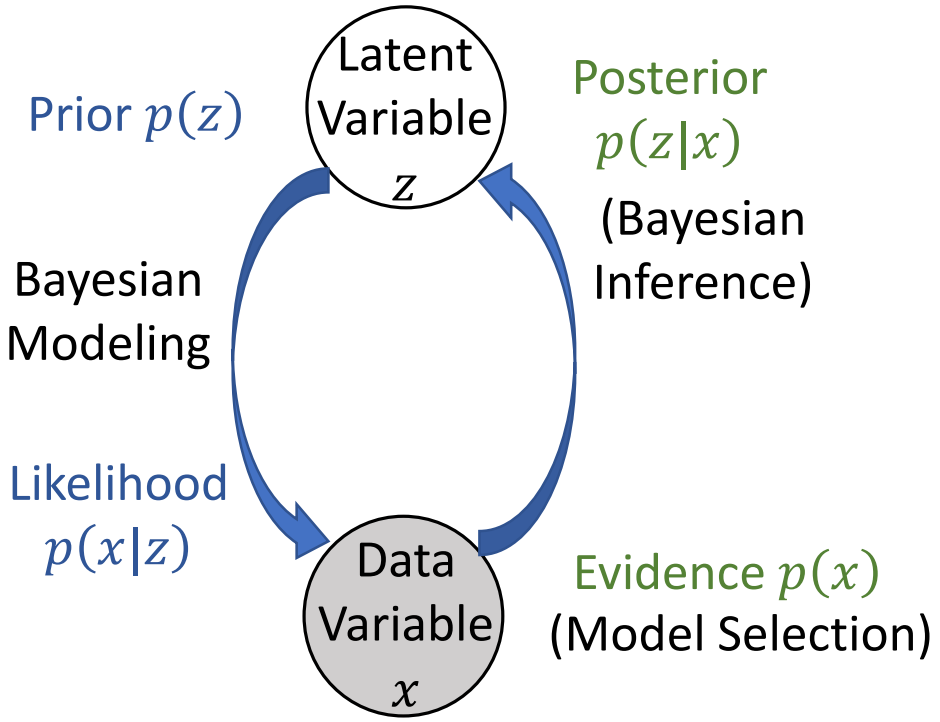


Adversarial robustness [LG17]

Bayesian Model: Overview

The dependence between latent variable and data is probabilistic.

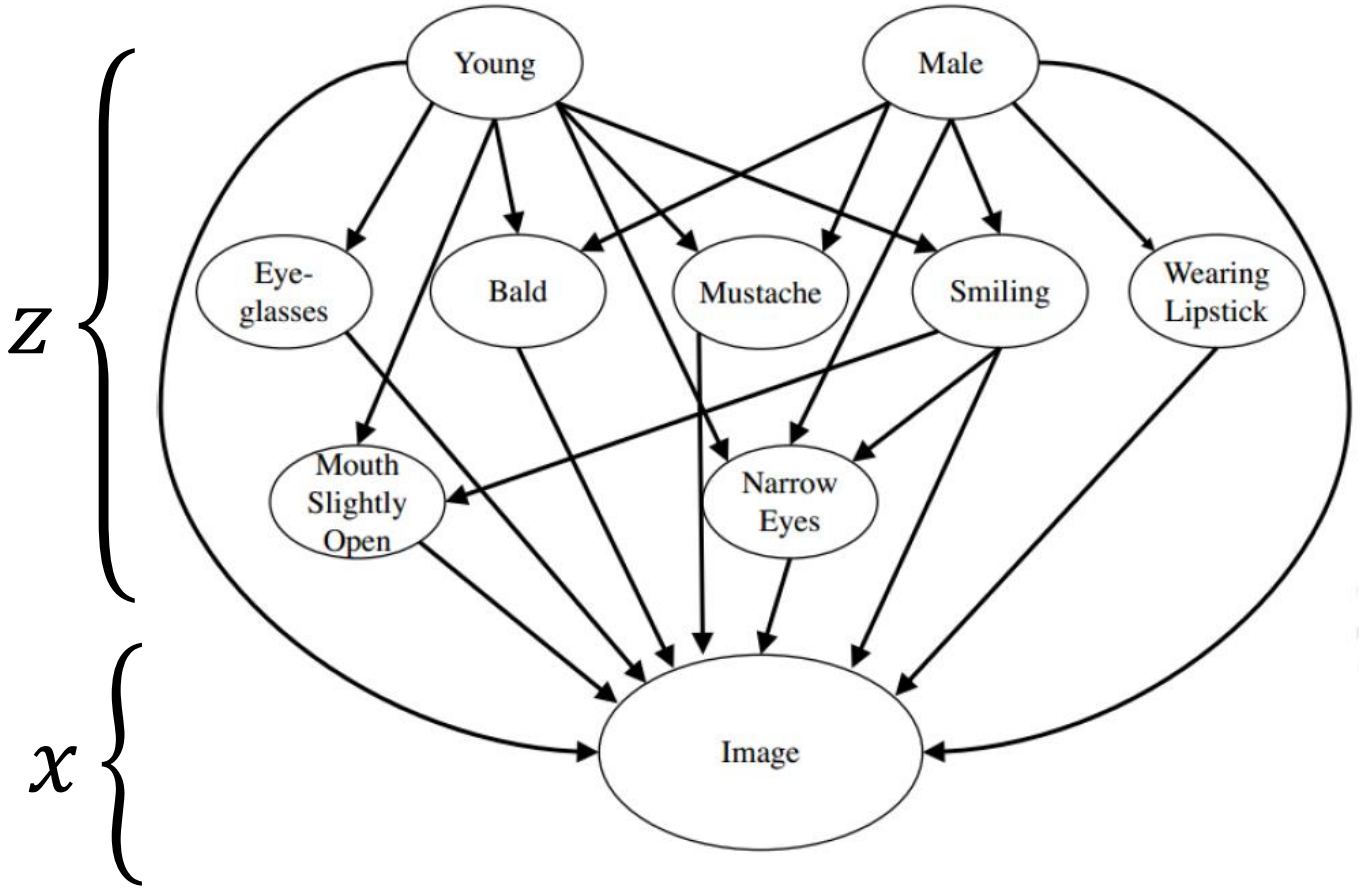
- Stable training process
- Principled and natural inference $p(z|x)$ via Bayes' rule



Bayesian Model: Overview

The dependence between latent variable and data is probabilistic.

- Natural to incorporate prior knowledge



Bald

Mustache

[KSD+18]

Bayesian Model: Overview

The dependence between latent variable and data is probabilistic.

- Latent variable: Compact representation of the dependency.
  - De Finetti's Theorem (1955)

    If $(x_1, x_2, \dots)$ are *infinitely exchangeable*, then ∃ r.v. z and $p(\cdot\,|z)$ s.t. $\forall n$,

$$p(x_1, \dots, x_n) = \int \left(\prod_{i=1}^{n} p(x_i|z) \right) p(z)\, \mathrm{d}z\,.$$

$$p\left(x_1 \; x_2 \; \cdots \; x_n \right) = \int_z p\left(\begin{array}{c} z \\ x_1 \; x_2 \; \cdots \; x_n \end{array} \right)$$

    Infinite exchangeability:

    For all n and permutation σ, $p(x_1, \dots, x_n) = p\left(x_{\sigma(1)}, \dots, x_{\sigma(n)}\right)$.

  - For supervised modeling, relation between x and y can be modeled by z via e.g., $p_z(y|x)$ or $y = f_z(x)$.

Bayesian Model: Overview

The dependence between latent variable and data is probabilistic.

- Latent variable: Abstract knowledge of the data.
  - Structured model: x (documents) <========> z (topics)

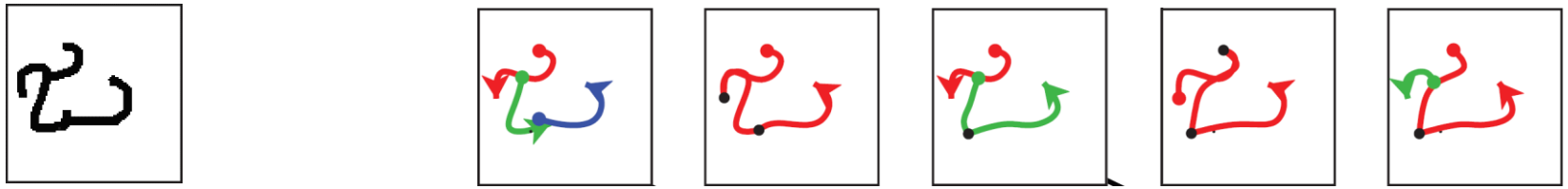
"ENGINES"	speed	product	introduced	designs	fuel
"ROYAL"	britain	queen	sir	earl	died
"ARMY"	commander	forces	war	general	military
"STUDY"	analysis	space	program	user	research
"PARTY"	act	office	judge	justice	legal
"DESIGN"	size	glass	device	memory	engine
"PUBLIC"	report	health	community	industry	conference
"CHURCH"	prayers	communion	religious	faith	historical

[PT13]

Bayesian Model: Overview

The dependence between latent variable and data is probabilistic.

- Latent variable: Abstract knowledge of the data.
  - Structured model: x (character image) <========> z (strokes)

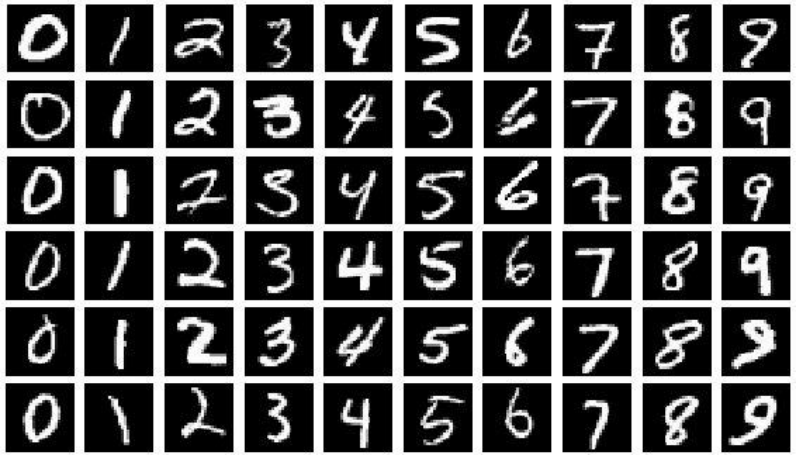


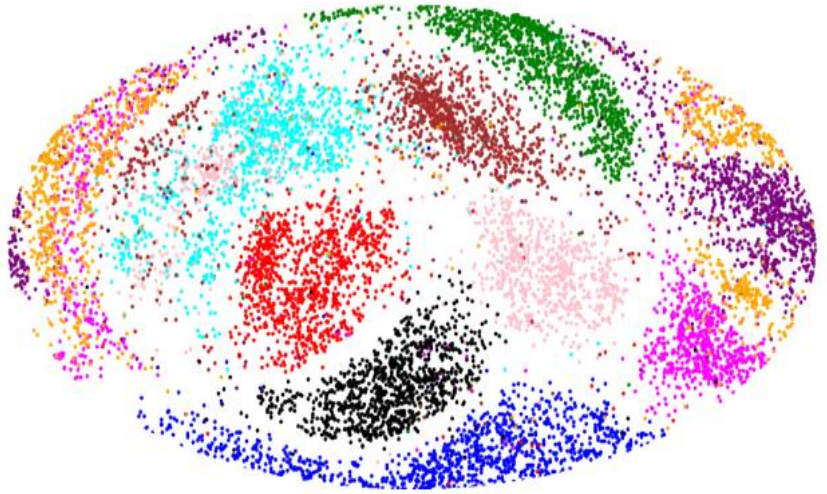
[LST15]

Bayesian Model: Overview

The dependence between latent variable and data is probabilistic.

- Latent variable: Abstract knowledge of the data.
  - Universal model: x (general data) <========> z (general semantic feature)



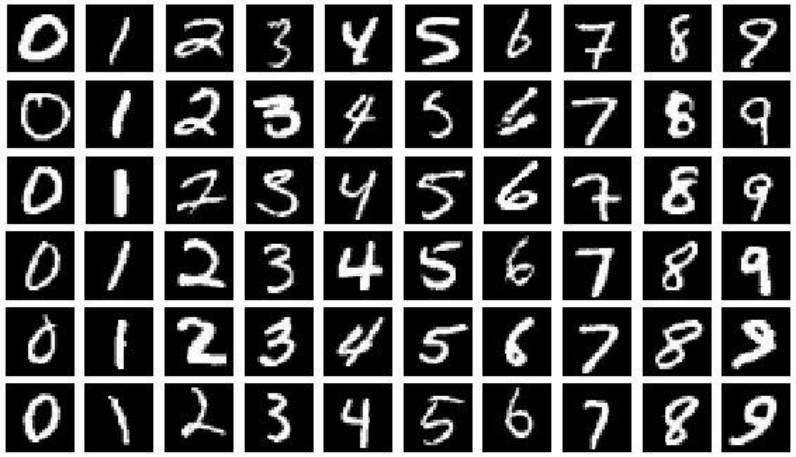


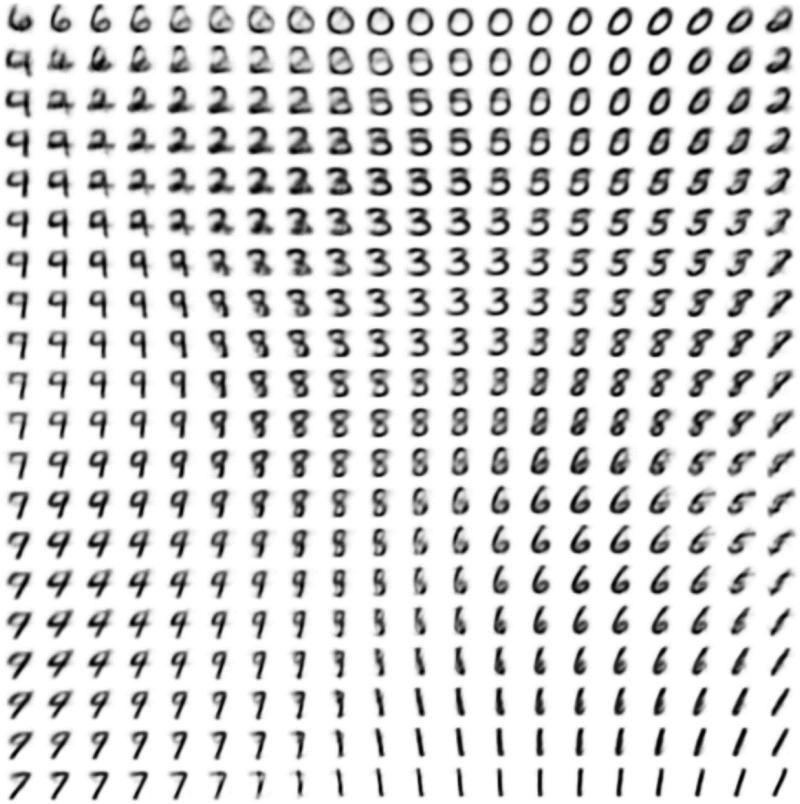
[DFD+18]

Bayesian Model: Overview

The dependence between latent variable and data is probabilistic.

- Latent variable: Abstract knowledge of the data.
  - Universal model: x (general data) <========> z (general semantic feature)





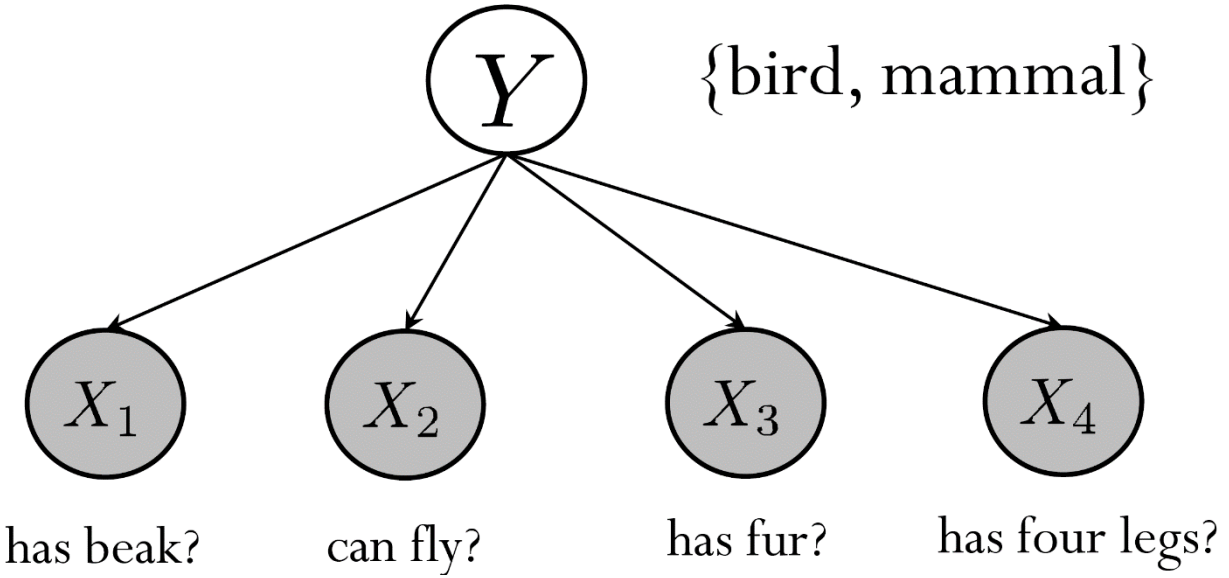
[KW14]

Bayesian Model: Overview

The dependence between latent variable and data is probabilistic.

- Latent variable: Abstract knowledge of the data.
  - Supervised tasks: (x, y) <=======> $z = y$

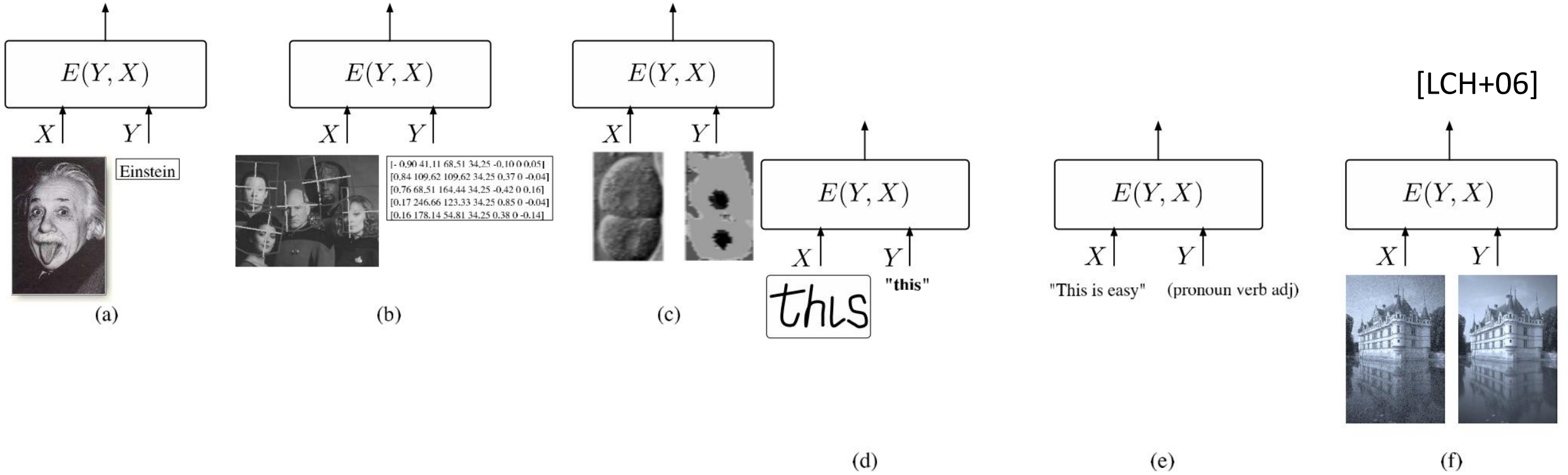
[Naive Bayes]



Bayesian Model: Overview

The dependence between latent variable and data is probabilistic.

- Latent variable: Abstract knowledge of the data.
  - Supervised tasks: (x, y) <=======> $z = y$



[LCH+06]

Bayesian Model: Overview

The dependence between latent variable and data is probabilistic.

- Latent variable: Abstract knowledge of the data.
  - Supervised tasks: (x, y) <=======> z

z: topics

"ENGINES"	speed	product	introduced
"ROYAL"	britain	queen	sir
"ARMY"	commander	forces	war
"STUDY"	analysis	space	program

x_1: document 1

y_1: science & tech

x_2: document 2

y_2: politics

x_1: document 1

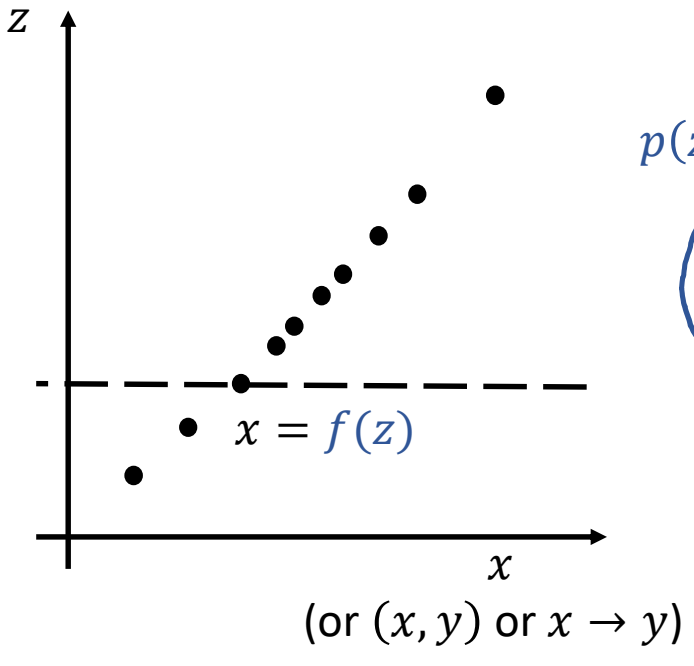
.....

$y = f_z(x)$,
or
$y \sim p_z(y|x)$.

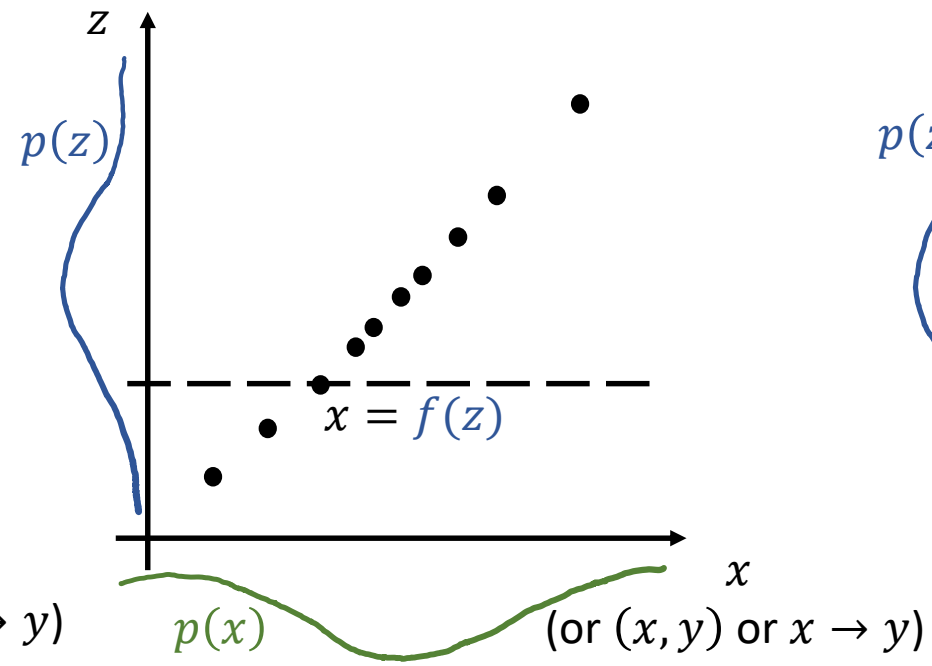
y_1: science & tech .....

Bayesian Model: Overview

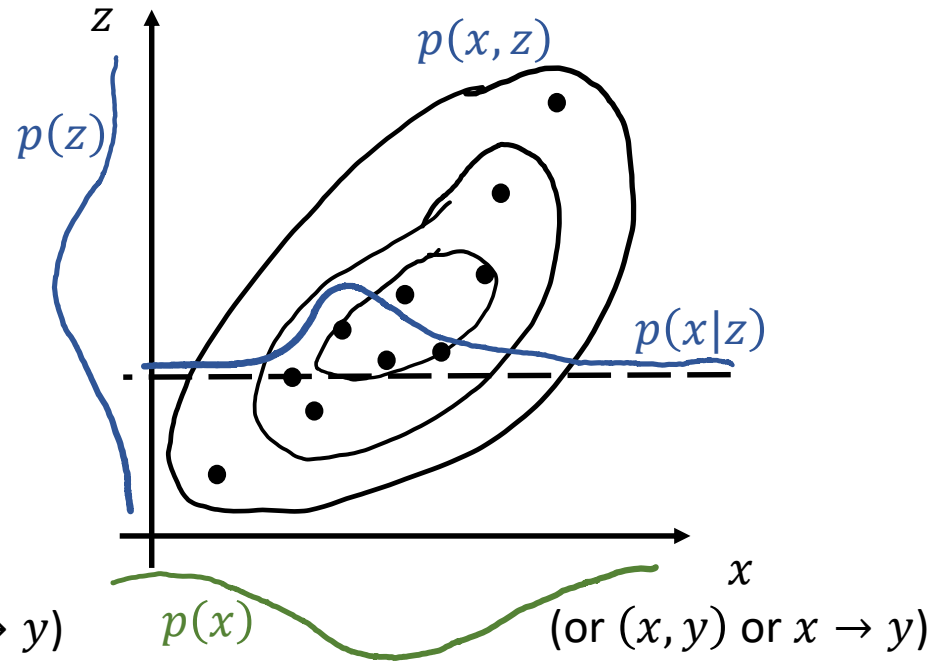
The dependence between latent variable and data is probabilistic.



Autoencoders
(or Feedforward NNs)



Generative Adversarial Nets,
Flow-Based Generative Models

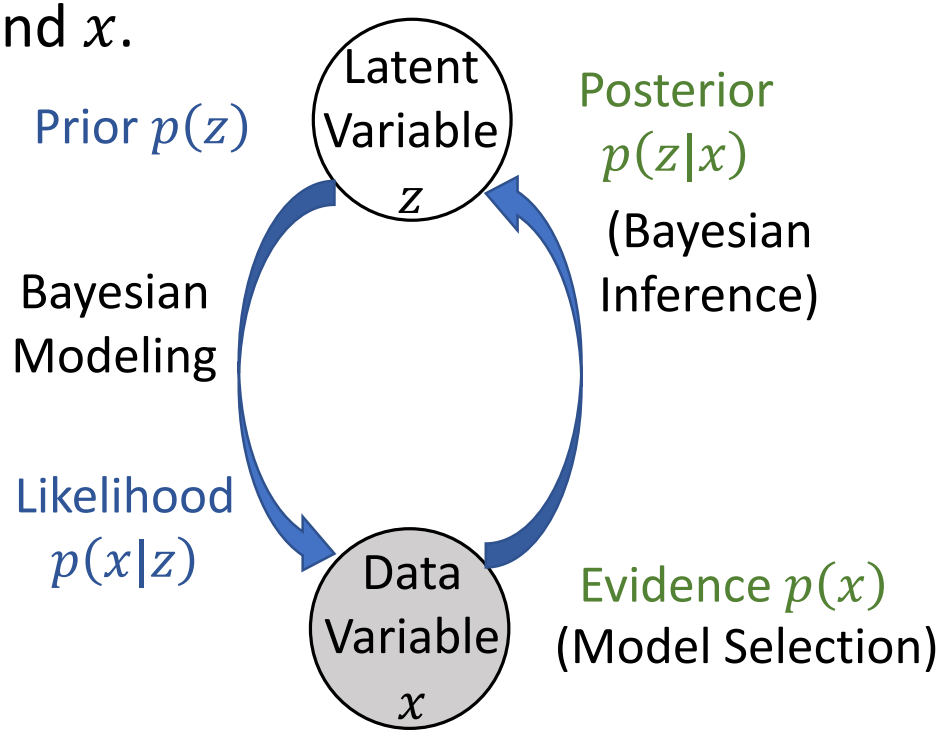


Bayesian Models

Bayesian Model: Overview

The dependence between latent variable and data is probabilistic.

- Model structure: *Prior* $p(z)$, *Likelihood* $p(x|z)$.
- How it works:
  - $p(z)$ models the *initial* belief of z.
  - $p(x|z)$ models the dependence between z and x.
  - $p(z,x) = p(z)p(x|z)$ then models all the information about x and z. Particularly,
  - *Evidence* $p(x) = \int p(z,x)\,\mathrm{d}z$ gives the belief on x.
  - *Posterior* $p(z|x) = \frac{p(z,x)}{p(x)} = \frac{p(z)p(x|z)}{\int p(z,x)\,\mathrm{d}z}$ (by Bayes' rule) gives the *updated* information that observation x conveys to z.



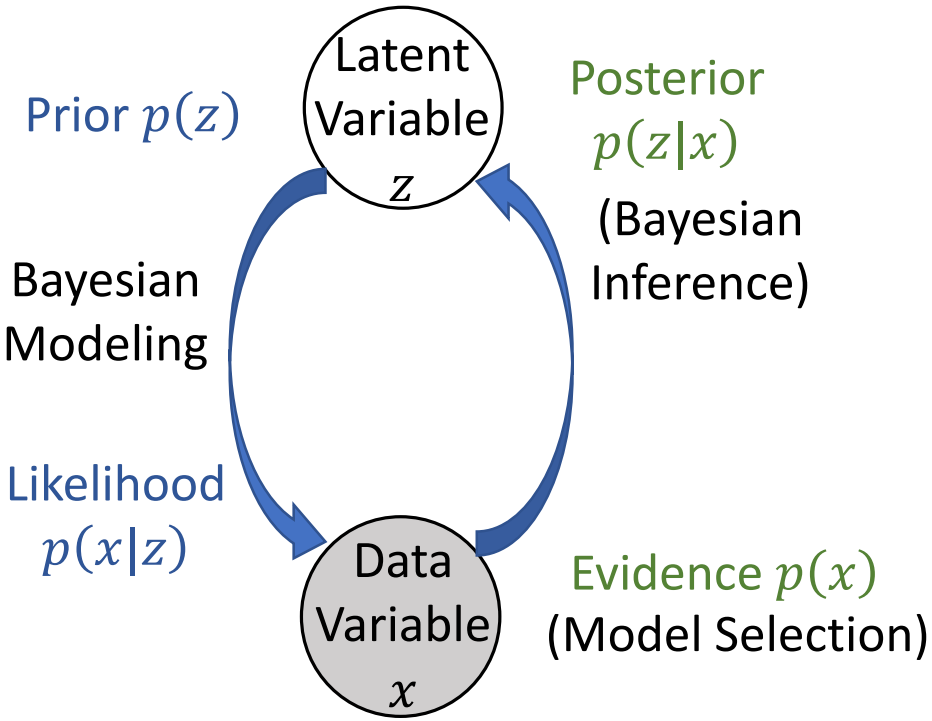
Bayesian Model: Overview

The dependence between latent variable and data is probabilistic.

- Model structure: *Prior* $p(z)$, *Likelihood* $p(x|z)$.
- How it learns (*model selection*):
  - Maximum likelihood estimation (equiv. to $\min_\theta \mathrm{KL}(\hat{p}, p_\theta)$):

$$\theta^* = \arg\max_\theta \mathbb{E}_{\hat{p}(x)}[\log p_\theta(x)]$$
$$\approx \arg\max_\theta \frac{1}{N}\sum_{n=1}^{N} \log p_\theta(x^{(n)}).$$

- How it serves:
  - Feature/representation learning, prediction: $p(z|x)$ (*Bayesian Inference*).
  - Generation, prediction: $z_{\text{new}} \sim p(z|x), x_{\text{new}} \sim p(x|z_{\text{new}})$.



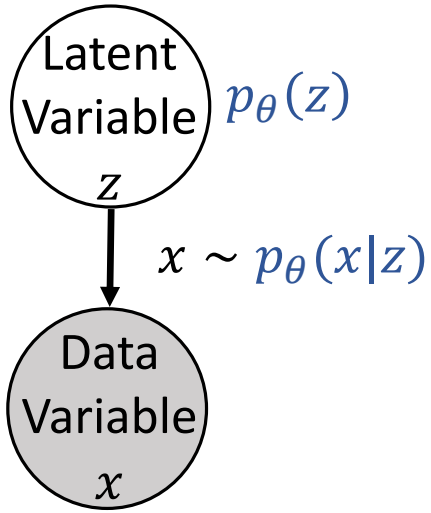
Bayesian Model: Taxonomy

The dependence between latent variable and data is probabilistic.

Bayesian Network (BayesNet):

$p(x, z)$ is specified via $p(z)$ and $p(x|z)$.

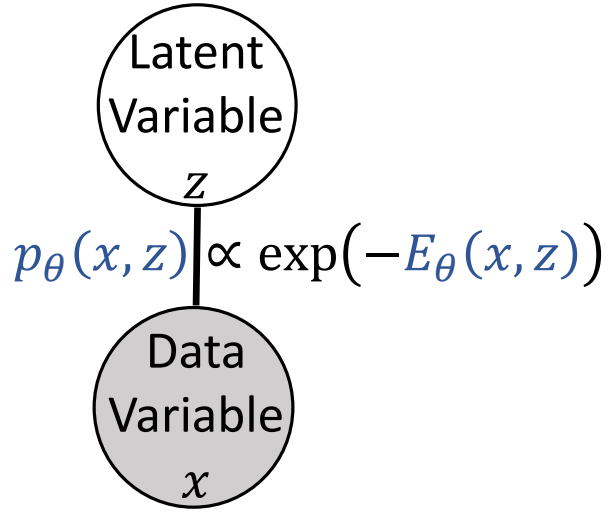
- Synonyms: Causal Networks, Directed Graphical Model
- Prior-likelihood interpretation
- Causality information encoded



Markov Random Field (MRF):

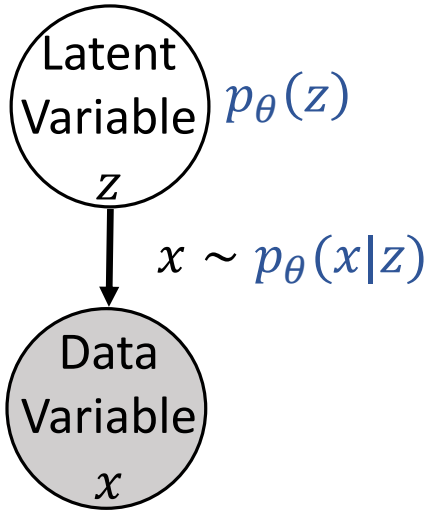
$p(x, z)$ is specified by an Energy function $E_\theta(x, z)$: $p(x, z) \propto \exp\left(-E_\theta(x, z)\right)$.

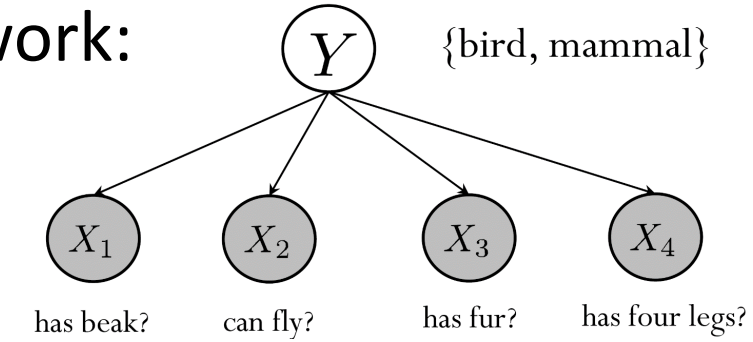
- Synonyms: Energy-Based Model, Undirected Graphical Model
- Focus on modeling the relationship

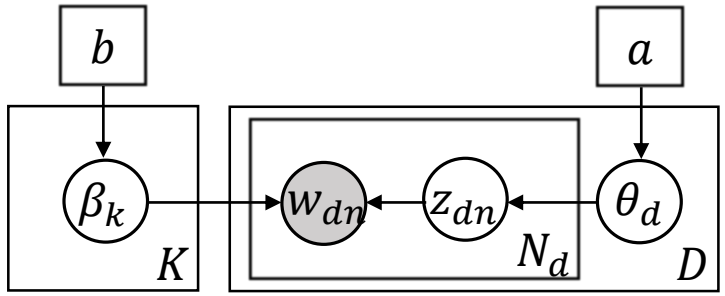


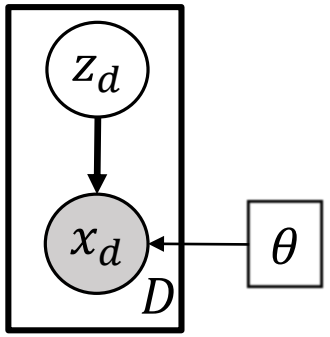
Bayesian Model: Taxonomy

- Bayesian Network:









Naive Bayes $z = y$:
$p(y), p(x_i|y)$: $\mathrm{Bern}(\pi)$.

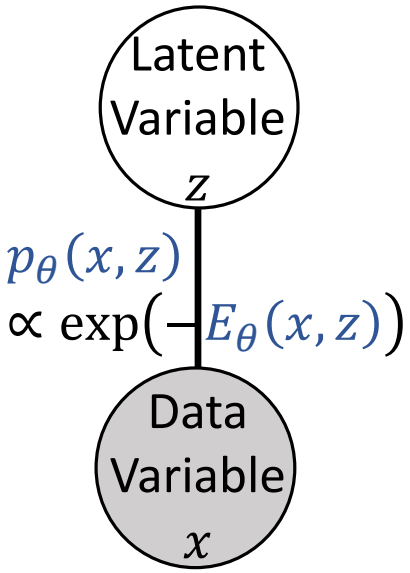
Latent Dirichlet Allocation:
- Prior: $p(\beta_k|b) = \mathrm{Dir}(b)$, $p(\theta_d|a) = \mathrm{Dir}(a), p(z_{dn}|\theta_d) = \mathrm{Mult}(\theta_d)$.
- Likelihood: $p(w_{dn}|z_{dn}, \beta) = \mathrm{Mult}(\beta_{z_{dn}})$.

Variational Auto-Encoder
- Prior: $p(z) = \mathcal{N}(0, I)$.
- Likelihood:

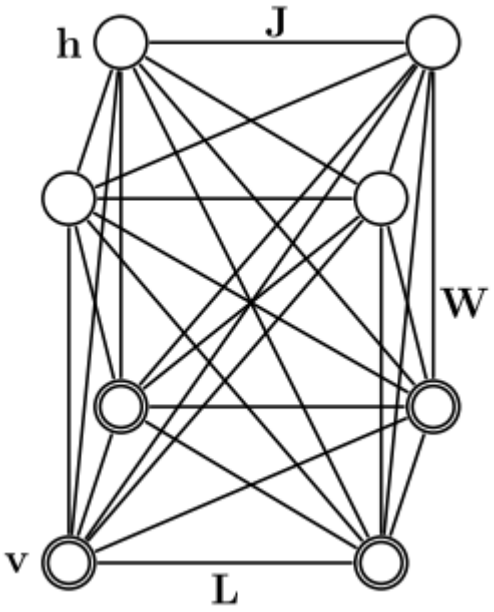
$$p(x_d|z_d) = \mathcal{N}\left(\mathrm{NN}_\theta^{(1)}(z_d), \mathrm{NN}_\theta^{(2)}(z_d)\right).$$

Bayesian Model: Taxonomy

- Markov Random Field

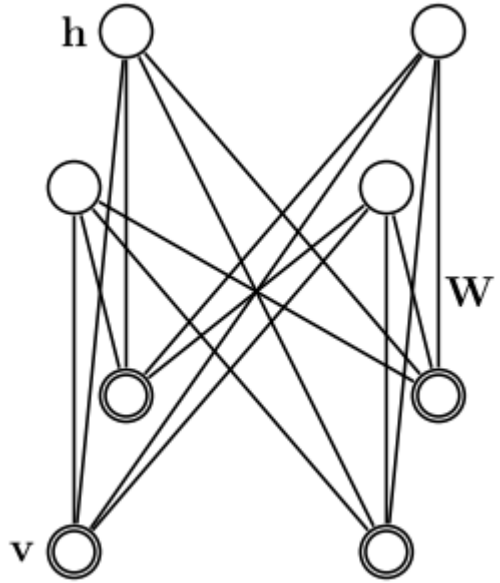


Boltzmann Machine [HS83]



$$E_\theta(v,h) = -\frac{1}{2}v^\top Lv - \frac{1}{2}h^\top Jh - v^\top Wh$$

Restricted Boltzmann Machine (*Harmonium*) [Smo86]



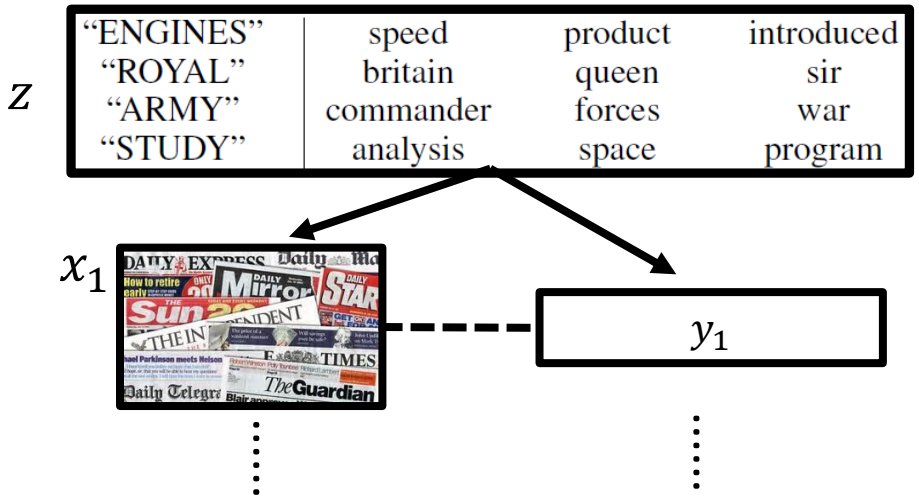
$$E_\theta(v,h) = -v^\top Wh$$

(may has a bias term $b^{(v)^\top}v + b^{(h)^\top}h$)

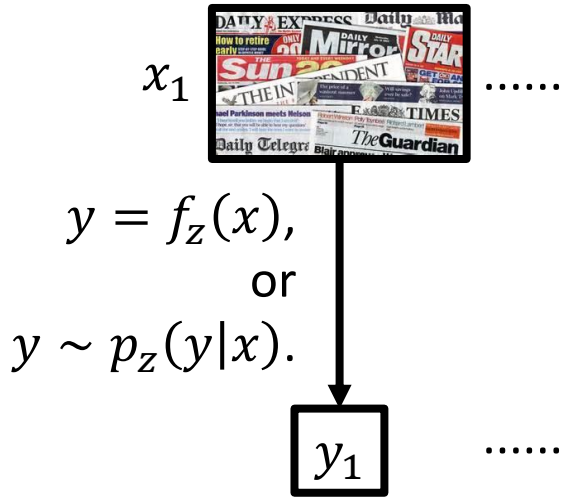
Bayesian Model: Taxonomy

For supervised tasks,

- Generative model:
  - Model all the data variable.
  - $p(x, y)$ is available.



- Non-generative model:
  - Directly model the dependency of y on x.
  - Only $p(y|x)$ is available.



Bayesian Model: Taxonomy

For supervised tasks,

- Generative model:
  - Responsible knowledge of the data.

    "What I cannot create, I do not understand." —*Richard Feynman*
  - More data-efficient.

    For logistic regression (discriminative) and naive Bayes (generative) [NJ01],

    $\epsilon_{\mathrm{Dis},N} \leq \epsilon_{\mathrm{Dis},\infty} + O\left(\sqrt{\frac{d}{N}}\right), \epsilon_{\mathrm{Gen},N} \leq \epsilon_{\mathrm{Gen},\infty} + O\left(\sqrt{\frac{\log d}{N}}\right)$. ($N$: data size. d: dimension)
  - Can leverage unlabeled data (semi-supv. learning).
- Non-generative model:
  - Better results with big enough data.

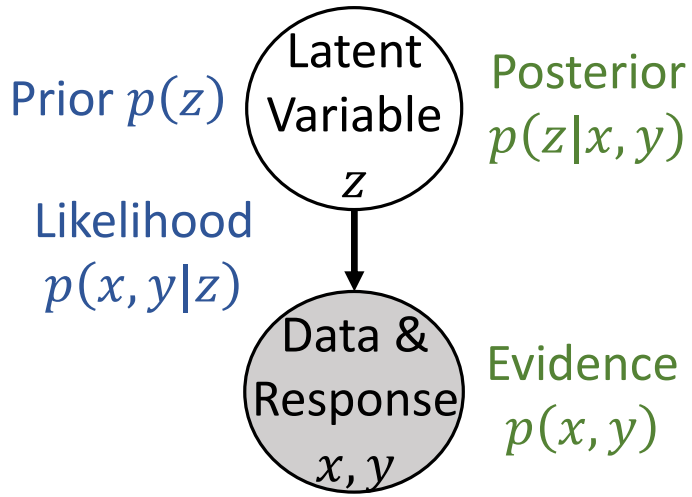
    [NJ01] If $x_1, \ldots, x_N$ are independent given y, then $\epsilon_{\mathrm{Dis},\infty} \approx \epsilon_{\mathrm{Gen},\infty}$.

    If not, then $\epsilon_{\mathrm{Dis},\infty} < \epsilon_{\mathrm{Gen},\infty}$.

Bayesian Model: Taxonomy

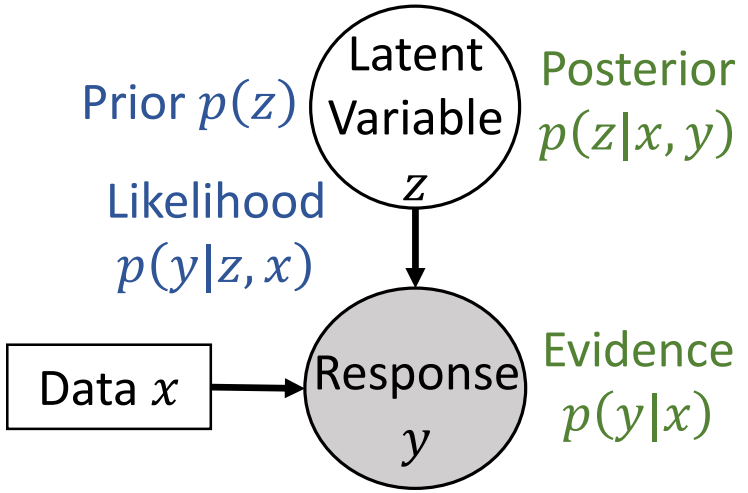
For supervised tasks,

Generative Bayesian Models



$$p(y^*|x^*,x,y) = \int p(y^*|z,x^*)p(z|x^*,x,y)\,\mathrm{d}z.$$

Non-Generative Bayesian Models



$$p(y^*|x^*,x,y) = \int p(y^*|z,x^*)p(z|x,y)\,\mathrm{d}z.$$

Bayesian Model: Taxonomy

For supervised tasks,

Bayesian Models

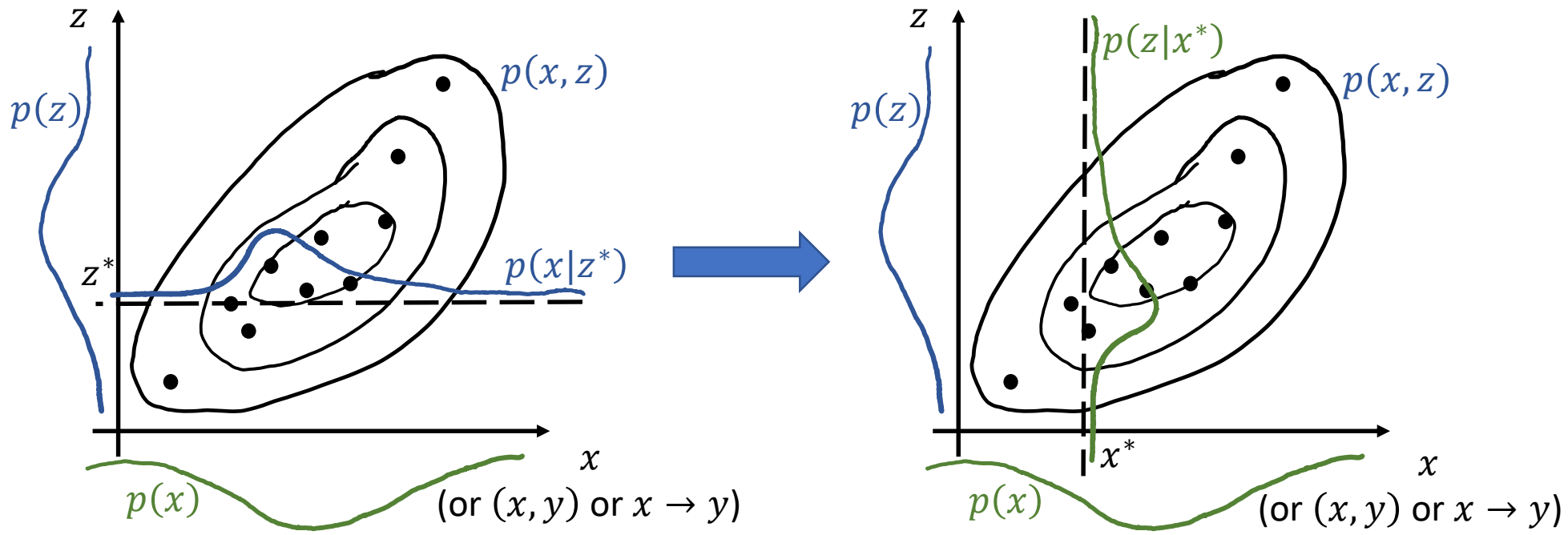
	Generative	Non-generative
Supervised	NB, sLDA, MedLDA, sVAE	BLR, BNN
Unsupervised	BayesNets (LDA, HMM, SBN, VAE), MRFs (BM, RBM, DBM, PoE)	(invalid task)

Overview

Bayesian Inference

Bayesian Inference

Estimate the posterior $p(z|x)$.



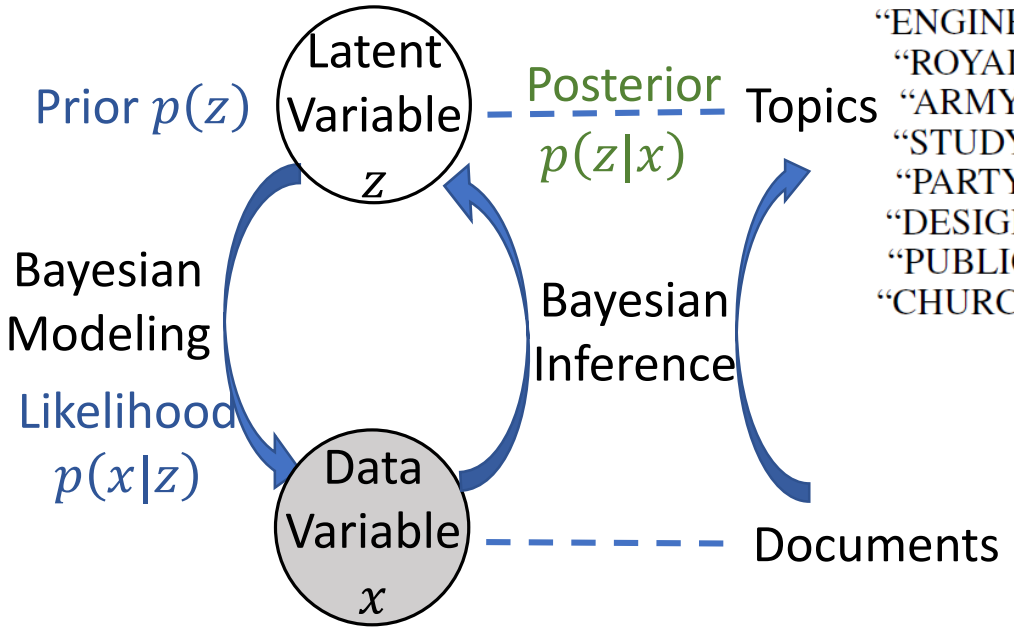
Bayesian Modeling

Bayesian Inference

Bayesian Inference

Estimate the posterior $p(z|x)$.

- Motivation 1: extract knowledge/representation from data.



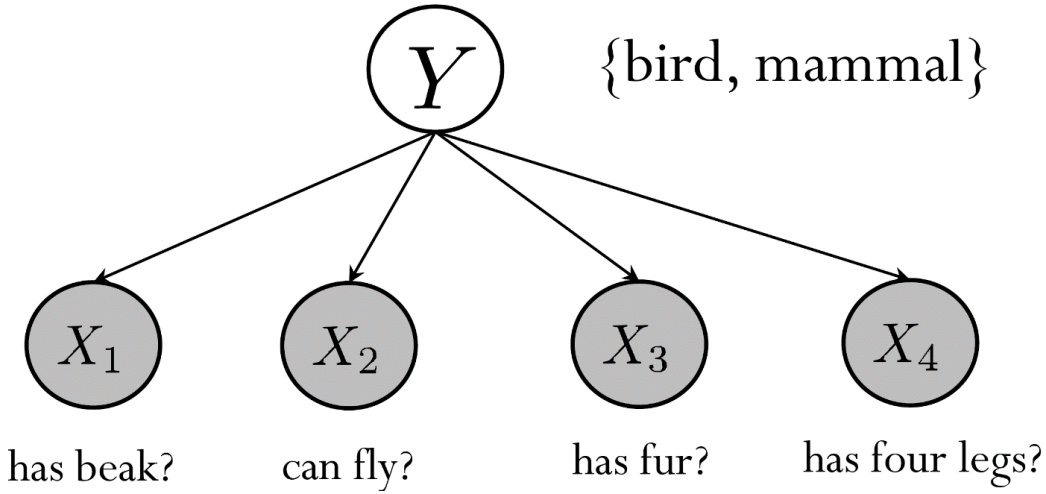
"ENGINES"	speed	product	introduced	designs	fuel
"ROYAL"	britain	queen	sir	earl	died
"ARMY"	commander	forces	war	general	military
"STUDY"	analysis	space	program	user	research
"PARTY"	act	office	judge	justice	legal
"DESIGN"	size	glass	device	memory	engine
"PUBLIC"	report	health	community	industry	conference
"CHURCH"	prayers	communion	religious	faith	historical

Bayesian Inference

Estimate the posterior $p(z|x)$.

- Motivation 1: extract knowledge/representation from data.

Naive Bayes: $z = y$.



$$p(y = 0|x) = \frac{p(x|y = 0)p(y = 0)}{p(x|y = 0)p(y = 0) + p(x|y = 1)p(y = 1)}$$

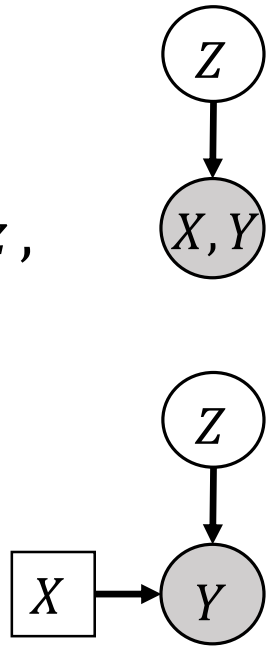
$f(x) = \arg\max_y p(y|x)$ achieves the lowest error $\int p\big(y = \big(1 - f(x)\big)|x\big)\, p(x)\, \mathrm{d}x$.

Bayesian Inference

Estimate the posterior $p(z|x)$.

- Motivation 2: prediction.

$$p(y^*|x^*,x,y) = \begin{cases} \int p(y^*|z,x^*)p(z|x^*,x,y)\,\mathrm{d}z\,, \\ \int p(y^*|z,x^*\,)p(z|x,y)\,\mathrm{d}z\,. \end{cases}$$



Bayesian Inference

Estimate the posterior $p(z|x)$.

- Motivation 3: facilitate model learning: $\max_\theta \frac{1}{N}\sum_{n=1}^{N} \log p_\theta(x^{(n)})$.
  - $p_\theta(x) = \int p_\theta(x|z)p_\theta(z)\,\mathrm{d}z$ is *hard* to evaluate:
    - Closed-form integration is generally unavailable.
    - Numerical integration
      - Curse of dimensionality
      - Hard to optimize.
  - $\log p_\theta(x) = \log \mathbb{E}_{p(z)}[p_\theta(x|z)] \approx \log \frac{1}{N}\sum_n p_\theta\left(x\middle|z^{(n)}\right), \left\{z^{(n)}\right\} \sim p(z)$.
    - Hard for $p(z)$ to cover regions where $p_\theta(x|z)$ is large.
    - $\log \frac{1}{N}\sum_n p_\theta\left(x\middle|z^{(n)}\right)$ is biased:

      $\mathbb{E}\left[\log \frac{1}{N}\sum_n p_\theta\left(x\middle|z^{(n)}\right)\right] \le \log \mathbb{E}\left[\frac{1}{N}\sum_n p_\theta\left(x\middle|z^{(n)}\right)\right] = \log p_\theta(x)$.

Bayesian Inference

Estimate the posterior $p(z|x)$.

- Motivation 3: facilitate model learning: $\max_\theta \frac{1}{N}\sum_{n=1}^{N} \log p_\theta(x^{(n)})$.

  An effective and practical learning approach:

  - Introduce a *variational distribution* $q(z)$:

$$\log p_\theta(x) = \mathcal{L}_\theta[q(z)] + \mathrm{KL}\big(q(z), p_\theta(z|x)\big),$$
$$\mathcal{L}_\theta[q(z)] \coloneqq \mathbb{E}_{q(z)}[\log p_\theta(z,x)] - \mathbb{E}_{q(z)}[\log q(z)].$$

    - $\mathcal{L}_\theta[q(z)] \le \log p_\theta(x)$ ➔ Evidence Lower BOund (ELBO)!
    - $\mathcal{L}_\theta[q(z)]$ is easier to compute ($p_\theta(z,x)$ is available for BayesNets).
  - (Variational) Expectation-Maximization Algorithm:

    (a) E-step: Let $\mathcal{L}_\theta[q(z)] \approx \log p_\theta(x)\,, \forall\theta \Leftrightarrow \overbrace{\min_{q\in\mathcal{Q}} \mathrm{KL}\big(q(z), p_\theta(z|x)\big)}^{\text{Bayesian Inference}}$;

    (b) M-step: $\max_\theta \mathcal{L}_\theta[q(z)]$.

Bayesian Inference

Estimate the posterior $p(z|x)$.

- It is a hard problem
  - Closed form of $p(z|x) \propto p(z)p(x|z)$ is generally intractable.
    - Numerical integration suffers from the curse of dimensionality.

Bayesian Inference

Estimate the posterior $p(z|x)$.

- It is a hard problem
  - We care about *expectations* w.r.t $p(z|x)$ (prediction, computing ELBO).
    - So that even if we know the closed form (e.g., by numerical integration), downstream tasks are still hard.
    - So that the Maximum *a Posteriori* (MAP) estimate

$$\arg\max_z \log p\left(z \middle| \{x^{(n)}\}_{n=1}^N\right) = \arg\max_z \log p(z) + \sum_{n=1}^N \log p(x^{(n)}|z)$$

does not help much for Bayesian tasks.

Modeling Method	Mathematical Problem
Parametric Method	Optimization
Bayesian Method	Bayesian Inference

Bayesian Inference

Estimate the posterior $p(z|x)$.

- Variational inference (VI)

  Use a *tractable* variational distribution $q(z)$ to approximate $p(z|x)$:

  $$\min_{q \in \mathcal{Q}} \mathrm{KL}\big(q(z), p(z|x)\big).$$

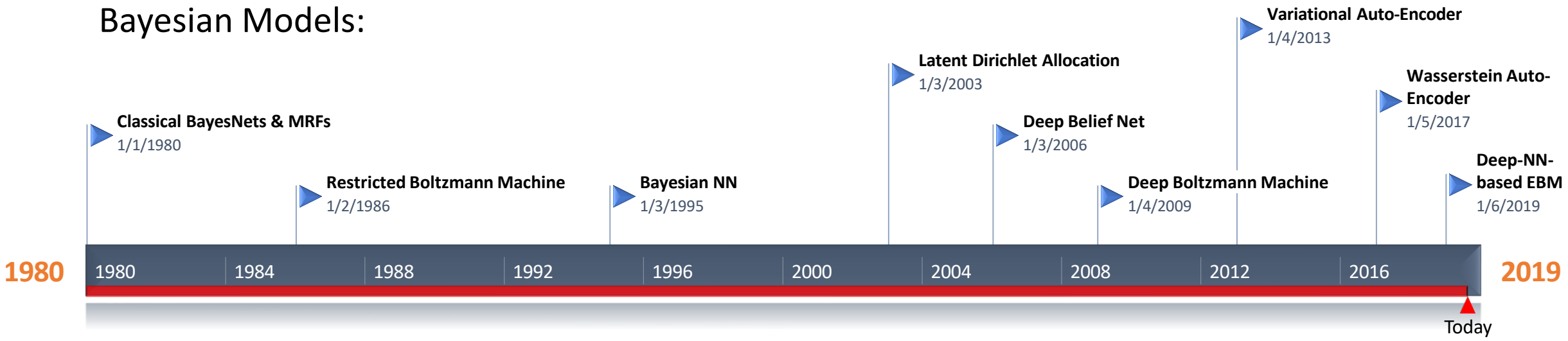
  Tractability: known density function, or samples are easy to draw.

  - Parametric VI: use a parameter ϕ to represent $q_\phi(z)$.
  - Particle-based VI: use a set of particles $\{z^{(i)}\}_{i=1}^{N}$ to represent $q(z)$.
- Monte Carlo (MC)
  - Draw samples from $p(z|x)$.
  - Typically by simulating a *Markov chain* (i.e., MCMC) to release requirements on $p(z|x)$.

Bayesian Learning

- Timelines

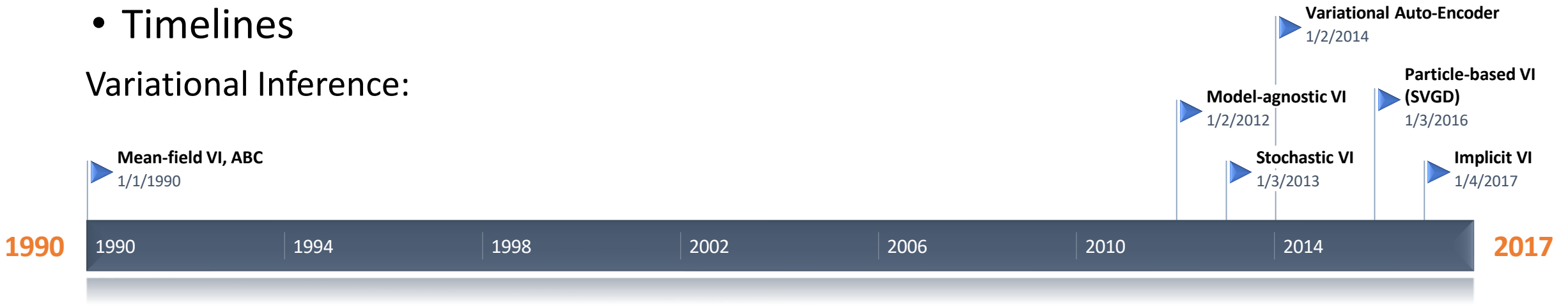
Bayesian Models:



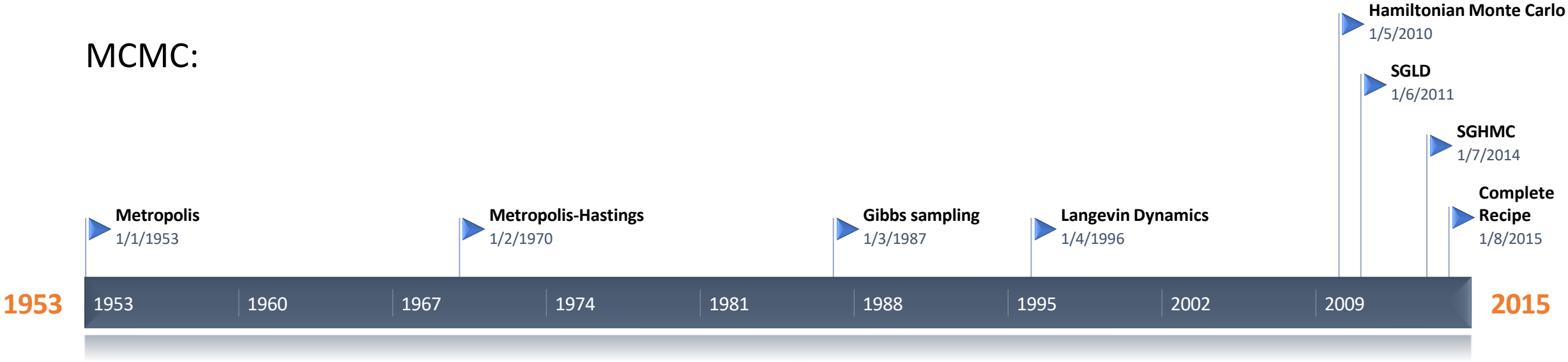
Bayesian Learning

- Timelines

Variational Inference:



MCMC:



Bayesian Inference

Estimate the posterior $p(z|x)$.

- Parametric Variational Inference

Bayesian Inference: Overview Recap.

Estimate the posterior $p(z|x)$.

- Variational inference (VI)

  Use a *tractable* variational distribution $q(z)$ to approximate $p(z|x)$:

  $$\min_{q \in \mathcal{Q}} \mathrm{KL}\big(q(z), p(z|x)\big).$$

  Tractability: known density function, or samples are easy to draw.

  - Parametric VI: use a parameter ϕ to represent $q_\phi(z)$.
  - Particle-based VI: use a set of particles $\{z^{(i)}\}_{i=1}^{N}$ to represent $q(z)$.
- Monte Carlo (MC)
  - Draw samples from $p(z|x)$.
  - Typically by simulating a *Markov chain* (i.e., MCMC) to release requirements on $p(z|x)$.

Variational Inference

$$\min_{q \in \mathcal{Q}} \mathrm{KL}\big(q(z), p(z|x)\big).$$

- Parametric variational inference: use a parameter ϕ to represent $q_\phi(z)$.

But $\mathrm{KL}\left(q_\phi(z), p_\theta(z|x)\right)$ is hard to compute…

Recall $\log p_\theta(x) = \mathcal{L}_\theta[q(z)] + \mathrm{KL}\big(q(z), p_\theta(z|x)\big),$

so $\min_\phi \mathrm{KL}\left(q_\phi(z), p(z|x)\right) \Leftrightarrow \max_\phi \mathcal{L}_\theta\left[q_\phi(z)\right].$

The ELBO $\mathcal{L}_\theta\left[q_\phi(z)\right] = \mathbb{E}_{q_\phi(z)}[\log p_\theta(z,x)] - \mathbb{E}_{q_\phi(z)}\left[\log q_\phi(z)\right]$ is easier to compute.

Variational Inference

$$\min_{q \in \mathcal{Q}} \mathrm{KL}\big(q(z), p(z|x)\big).$$

- Parametric variational inference: use a parameter ϕ to represent $q_\phi(z)$.
  - Information theory perspective of the ELBO: Bits-Back Coding [HV93].
    - Average coding length for communicating x after communicating its code z:
      $$\mathbb{E}_{q(z|x)}[-\log p(x|z)].$$
    - Average coding length for communicating z under the bits-back coding:
      $$\mathbb{E}_{q(z|x)}[-\log p(z)] - \mathbb{E}_{q(z|x)}[-\log q(z|x)].$$
      The second term: the receiver knowns the encoder $q(z|x)$ that the sender uses.
    - Average coding length for communicating x with the help of z:
      $$\mathbb{E}_{q(z|x)}[-\log p(x|z) - \log p(z) + \log q(z|x)].$$
      This coincides with the negative ELBO!

      Maximize ELBO = Minimize averaged coding length under the bits-back scheme.

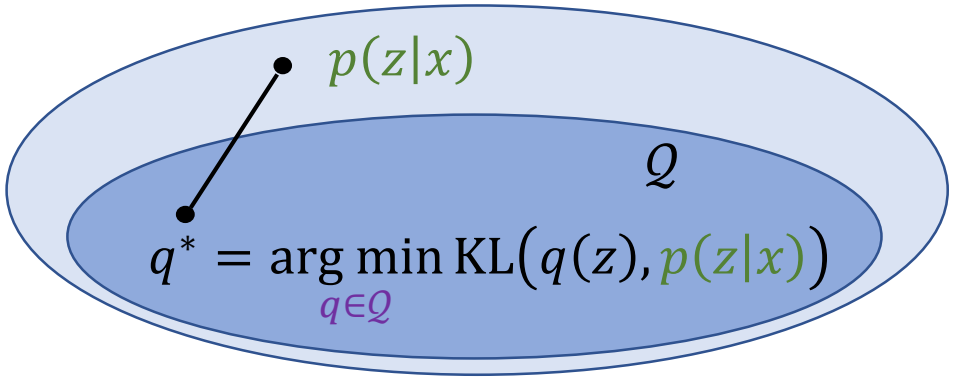
Variational Inference

$$\min_{q \in \mathcal{Q}} \mathrm{KL}\big(q(z), p(z|x)\big).$$

- Parametric variational inference: use a parameter ϕ to represent $q_\phi(z)$.

The Main Challenge:

- $\mathcal{Q}$ needs to enable practical computation of the ELBO,
- and needs to be as large/general/flexible as possible.



Variational Inference

- Parametric variational inference: use a parameter ϕ to represent $q_\phi(z)$.

$$\max_{\phi} \Big(\mathcal{L}_\theta\big[q_\phi(z)\big] = \mathbb{E}_{q_\phi(z)}[\log p_\theta(z,x)] - \mathbb{E}_{q_\phi(z)}\big[\log q_\phi(z)\big] \Big).$$

  - Explicit variational inference: specify the form of the density function $q_\phi(z)$.
    - Model-specific $q_\phi(z)$: $\mathrm{ELBO}(\theta,\phi)$ has closed form
      (e.g., [SJJ96] for SBN, [BNJ03] for LDA).

Variational Inference

- Parametric variational inference: use a parameter ϕ to represent $q_\phi(z)$.

$$\max_\phi \Big(\mathcal{L}_\theta[q_\phi(z)] = \mathbb{E}_{q_\phi(z)}[\log p_\theta(z,x)] - \mathbb{E}_{q_\phi(z)}\big[\log q_\phi(z)\big] \Big).$$

  - Explicit variational inference: specify the form of the density function $q_\phi(z)$.

    To be applicable to any model (model-agnostic $q_\phi(z)$):

    - [GHB12]: mixture of Gaussian $q_\phi(z) = \frac{1}{N}\sum_{n=1}^N \mathcal{N}(z|\mu_n, \sigma_n^2 I)$.

      $\text{Blue} = \frac{1}{N}\sum_{n=1}^N \mathbb{E}_{\mathcal{N}(\mu_n,\sigma_n^2 I)}[f(z)]$

      $\approx \frac{1}{N}\sum_{n=1}^N \mathbb{E}_{\mathcal{N}(\mu_n,\sigma_n^2 I)}[\text{Taylor}_2(f,\mu_n)] = \frac{1}{N}\sum_{n=1}^N f(\mu_n) + \frac{\sigma_n^2}{2}\text{tr}\big(\nabla^2 f(\mu_n)\big),$

      $\text{Red} \geq -\frac{1}{N}\sum_{n=1}^N \log \sum_{j=1}^N \mathcal{N}\big(\mu_n|\mu_j, (\sigma_n^2 + \sigma_j^2)I\big) + \log N.$

    - [RGB14]: mean-field $q_\phi(z) = \prod_{d=1}^D q_{\phi_d}(z_d)$.
      - $\nabla_\theta \mathcal{L}_\theta[q_\phi] = \mathbb{E}_{q_\phi(z)}[\nabla_\theta \log p_\theta(z,x)]$.
      - $\nabla_\phi \mathcal{L}_\theta[q_\phi] = \mathbb{E}_{q_\phi(z)}\big[\big(\nabla_\phi \log q_\phi(z)\big)\big(\log p_\theta(z,x) - \log q_\phi(z)\big)\big]$

        (similar to REINFORCE [Wil92]) (with variance reduction).

Variational Inference

- Parametric variational inference: use a parameter ϕ to represent $q_\phi(z)$.

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

  - Explicit variational inference: specify the form of the density function $q_\phi(z)$.

    To be more flexible and model-agnostic:

    - [RM15, KSJ+16]: define $q_\phi(z)$ by a generative model:

$$z \sim q_\phi(z) \Leftrightarrow z = g_\phi(\epsilon), \epsilon \sim q(\epsilon),$$

      where g_ϕ is invertible (flow model).

      Density function $q_\phi(z)$ is known!

$$q_\phi(z) = q\left(\epsilon = g_\phi^{-1}(z)\right) \left| \frac{\partial g_\phi^{-1}}{\partial z} \right|. \quad \text{(rule of change of variables)}$$

$$\mathcal{L}_\theta[q_\phi] = \mathbb{E}_{q(\epsilon)} \left[\log p_\theta(z,x) \Big|_{z=g_\phi(\epsilon)} - \log q_\phi(z) \Big|_{z=g_\phi(\epsilon)} \right].$$

Variational Inference

$$\min_{q \in \mathcal{Q}} \mathrm{KL}\big(q(z), p(z|x)\big).$$

- Parametric variational inference: use a parameter ϕ to represent $q_\phi(z)$.
  - Expectation Propagation [Min01]:

    $p\left(z \middle| \{x^{(n)}\}_{n=1}^N\right) \propto p(z) \prod_{n=1}^N p\big(x^{(n)}|z\big)$, so let $q(z) = p(z) \prod_{n=1}^N q_{\phi_n}(z)$, and

$$\arg\min_{\phi_n} \mathrm{KL}\big(\tilde{q}_n(z), q(z)\big) = \arg\min_{\phi_n} \mathbb{E}_{\tilde{q}_n}\big[\log q_{\phi_n}(z)\big],$$

    where $\tilde{q}_n(z) \propto \underbrace{\frac{q(z)}{q_{\phi_n}(z)}}_{\text{cavity distr.}} p\big(x^{(n)}|z\big) = p(z) p\big(x^{(n)}|z\big) \prod_{n' \neq n} q_{\phi_{n'}}(z).$

    - Local objective. Also has a global objective reformulation.
    - For $q_{\phi_n}(z) \propto h(z) \exp\big(\phi_n^\top T_n(z)\big)$, EP ➔ Solve $\mathbb{E}_{\tilde{q}_n}[T_n(z)] = \mathbb{E}_{q_{\phi_n}}[T_n(z)]$ for ϕ_n.
    - Stochastic EP [LHT15].

Variational Inference

- Parametric variational inference: use a parameter ϕ to represent $q_\phi(z)$.

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

  - Implicit variational inference: define $q_\phi(z)$ by a generative model:

$$z \sim q_\phi(z) \Leftrightarrow z = g_\phi(\epsilon), \epsilon \sim q(\epsilon),$$

    where g_ϕ is a general function.
    - More flexible than explicit VIs.
    - Samples are easy to draw, but density function $q_\phi(z)$ is unavailable.
    - $\mathcal{L}_\theta[q_\phi(z)] = \mathbb{E}_{q(\epsilon)}\left[\log p_\theta(x|z)\,|_{z=g_\phi(\epsilon)}\right] - \mathbb{E}_{q(\epsilon)}\left[\log \frac{q_\phi(z)}{p(z)}\,|_{z=g_\phi(\epsilon)}\right].$

    Key Problem:

    - Density Ratio Estimation $r(z) := \frac{q_\phi(z)}{p(z)}$.
    - Gradient Estimation $\nabla \log q(z)$.

Variational Inference

- Parametric variational inference: use a parameter ϕ to represent $q_\phi(z)$.

$$\max_{\phi}\Big(\mathcal{L}_\theta\big[q_\phi(z)\big] = \mathbb{E}_{q_\phi(z)}[\log p_\theta(z,x)] - \mathbb{E}_{q_\phi(z)}\big[\log q_\phi(z)\big]\Big).$$

  - Implicit variational inference

    Density Ratio Estimation:

    - [MNG17]: $\log r = \arg\max_T \mathbb{E}_{q_\phi(z)}\big[\log\sigma\big(T(z)\big)\big] + \mathbb{E}_{p(z)}\Big[\log\Big(1-\sigma\big(T(z)\big)\Big)\Big]$.

      Also used in [MSJ+15, Hus17, TRB17].

    - [SSZ18a]: $r = \arg\min_{\hat{r}} \frac{1}{2}\mathbb{E}_p[(\hat{r}-r)^2]$, so

$$r \approx \arg\min_{\hat{r}\in\mathcal{H}} \frac{1}{2}\mathbb{E}_p[(\hat{r}-r)^2] + \frac{\lambda}{2}\|\hat{r}\|_{\mathcal{H}}^2 \approx \frac{1}{\lambda N_q}\mathbf{1}^\top K_q - \frac{1}{\lambda N_p N_q}\mathbf{1}^\top K_{qp}\left(\frac{1}{N_p}K_{pp} + \lambda I\right)^{-1} K_p,$$

      where $K_p(z)_j = K\left(z_j^{(p)}, z\right)$, $\left(K_{qp}\right)_{ij} = K\left(z_i^{(q)}, z_j^{(p)}\right)$, $\left\{z_i^{(q)}\right\}_{i=1}^{N_q} \sim q_\phi(z)$, $\left\{z_j^{(p)}\right\}_{j=1}^{N_p} \sim p(z)$.

    - Other techniques: (Bickel et al., 2007), KLIEP (Sugiyama et al., 2008), LSIF (Kanamori et al., 2009), (Hido et al., 2011), (Sugiyama et al., 2012).

Variational Inference

- Parametric variational inference: use a parameter ϕ to represent $q_\phi(z)$.

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

  - Implicit variational inference

    Gradient Estimation:

    - [VLBM08]:

$$\begin{aligned} v^*(x,z) &= \operatorname{argmin}_v \mathbb{E}_{x,z\sim q(x,z),\eta\sim\mathcal{N}_{\sigma_n}} \|v(x,z+\eta)-z\|^2 \\ &\approx z + \sigma_n^2 \frac{\partial \log q(z|x)}{\partial z} \qquad \text{as } \sigma_n \to 0 \qquad (7) \end{aligned}$$

Variational Inference

- Parametric variational inference: use a parameter ϕ to represent $q_\phi(z)$.

$$\max_{\phi}\Big(\mathcal{L}_\theta\big[q_\phi(z)\big] = \mathbb{E}_{q_\phi(z)}[\log p_\theta(z,x)] - \mathbb{E}_{q_\phi(z)}\big[\log q_\phi(z)\big]\Big).$$

  - Implicit variational inference

    Gradient Estimation:

    - [LT18]:
      - Stein's identity: w/ bound. cond. for $\boldsymbol{h}$: $\mathbb{R}^d \to \mathbb{R}^{d'}$, $\mathbb{E}_q[\boldsymbol{h}(\boldsymbol{x})\nabla_{\boldsymbol{x}} \log q(\boldsymbol{x})^{\mathrm{T}} + \nabla_{\boldsymbol{x}}\boldsymbol{h}(\boldsymbol{x})] = \mathbf{0}$
      - MC approx.: $-\frac{1}{K}\mathbf{H}\mathbf{G} + \mathrm{err} = \overline{\nabla_{\boldsymbol{x}}\boldsymbol{h}}$.

        So define the approximator: $\hat{\mathbf{G}}_V^{\mathrm{Stein}} := \arg\min_{\hat{\mathbf{G}}\in\mathbb{R}^{K\times d}} ||\overline{\nabla_{\boldsymbol{x}}\boldsymbol{h}} + \frac{1}{K}\mathbf{H}\hat{\mathbf{G}}||_F^2 + \frac{\eta}{K^2}||\hat{\mathbf{G}}||_F^2$

        Solution: $\hat{\mathbf{G}}_V^{\mathrm{Stein}} = -(\mathbf{K} + \eta\boldsymbol{I})^{-1}\langle\nabla, \mathbf{K}\rangle$ where $\mathbf{K} \coloneqq \mathbf{H}^{\mathrm{T}}\mathbf{H}$, $\langle\nabla, \mathbf{K}\rangle_{ij} \coloneqq \sum_{k=1}^{K} \nabla_{x_j^k} \mathbf{K}_{ik}$.
      - Directly select a kernel (*e.g.*, RBF kernel), instead of $\boldsymbol{h}$.
      - Out-of-sample prediction:

$$\nabla_{\boldsymbol{y}} \log q(\boldsymbol{y})^{\mathrm{T}} \approx -\left(\mathbf{K}_{\boldsymbol{y}\boldsymbol{y}} + \eta - \mathbf{K}_{\boldsymbol{y}\mathbf{X}}(\mathbf{K} + \eta\mathbf{I})^{-1}\mathbf{K}_{\mathbf{X}\boldsymbol{y}}\right)^{-1}\left(\mathbf{K}_{\boldsymbol{y}\mathbf{X}}\hat{\mathbf{G}}_V^{\mathrm{Stein}} - \left(\mathbf{K}_{\boldsymbol{y}\mathbf{X}}(\mathbf{K} + \eta\mathbf{I})^{-1} + \mathbf{1}^{\mathrm{T}}\right)\nabla_{\boldsymbol{y}}\mathcal{K}(\cdot, \boldsymbol{y})\right).$$

Variational Inference

- Parametric variational inference: use a parameter ϕ to represent $q_\phi(z)$.

$$\max_{\phi}\Big(\mathcal{L}_\theta\big[q_\phi(z)\big] = \mathbb{E}_{q_\phi(z)}[\log p_\theta(z,x)] - \mathbb{E}_{q_\phi(z)}\big[\log q_\phi(z)\big]\Big).$$

  - Implicit variational inference

    Gradient Estimation:

    - [SSZ18b]:
      - The Nystrom Method: approximation to the eigenfunctions $\{\psi_i\}_{i=1}^{\infty}$ of kernel k in L_q^2: $\hat{\psi}_i(x) = \frac{\sqrt{M}}{\lambda_i}\sum_{m=1}^{M} k\big(x, x^{(m)}\big)u_{im}$, where $\{x^{(m)}\}_{m=1}^{M} \sim q$, and $\{u_i\}_{i=1}^{M}, \{\lambda_i\}_{i=1}^{M}$ are the eigenvectors and eigenvalues of the Gram matrix $\Big(k\big(x^{(m)}, x^{(n)}\big)\Big)_{m,n}$.

Assumed in L_q^2

      - $\partial_d \log q(x) = \sum_{j=1}^{\infty}\beta_{dj}\,\psi_j(x), \beta_{dj} = \mathbb{E}_q\big[\psi_j \partial_d \log q\big] = -\mathbb{E}_q\big[\partial_d \psi_j\big]$. So: $\partial_d \log q(x) \approx \sum_{j=1}^{K}\hat{\beta}_{dj}\hat{\psi}_j(x), \hat{\beta}_{dj} = -\frac{1}{M}\sum_{n=1}^{M}\partial_d\hat{\psi}_j\big(x^{(n)}\big)$.
      - Faster out-of-sample prediction than [LT18].

boundary cond. on k

Variational Inference

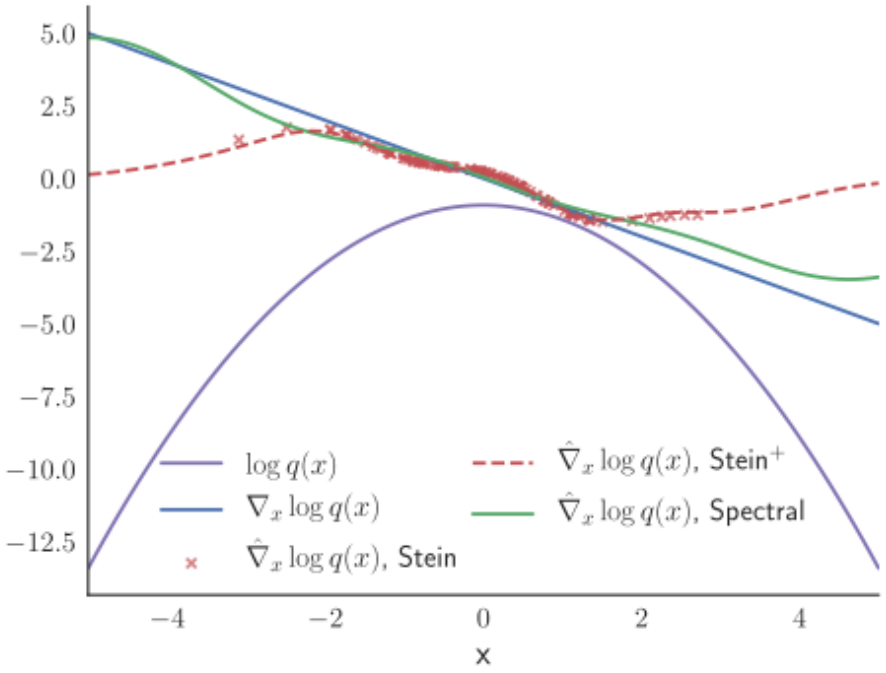
- Parametric variational inference: use a parameter ϕ to represent $q_\phi(z)$.

$$\max_{\phi}\Big(\mathcal{L}_\theta[q_\phi(z)] = \mathbb{E}_{q_\phi(z)}[\log p_\theta(z,x)] - \mathbb{E}_{q_\phi(z)}[\log q_\phi(z)]\Big).$$

  - Implicit variational inference

    Gradient Estimation:

    - [SSZ18b]:



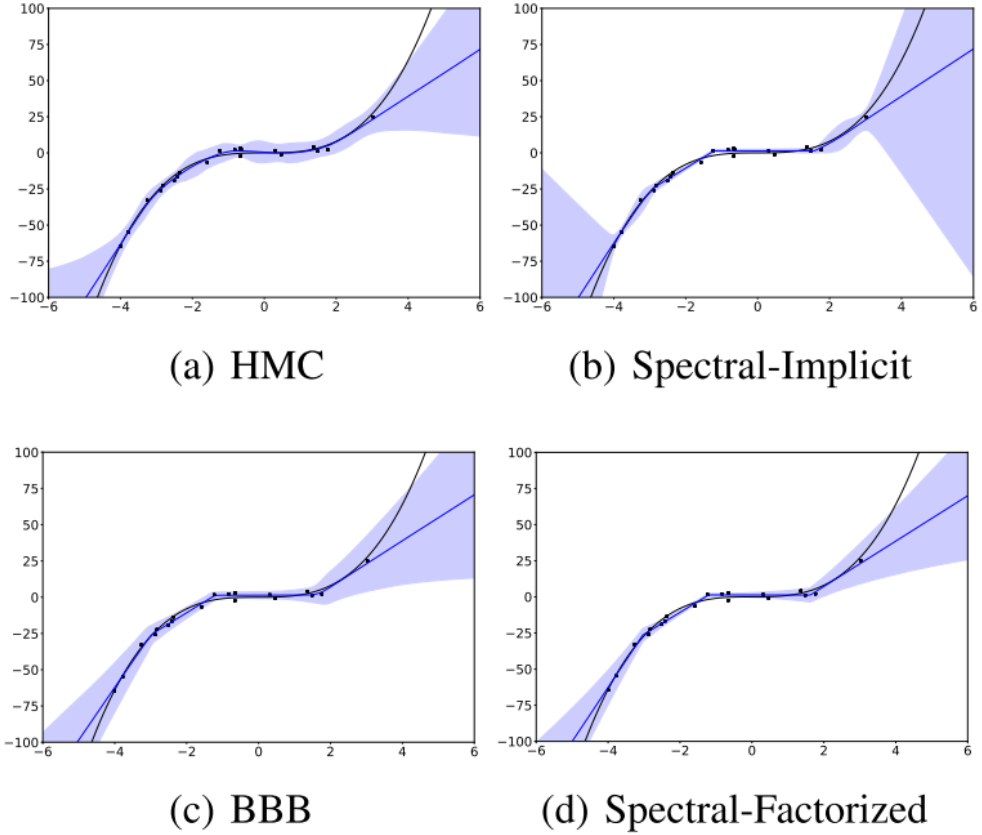


Figure 1. Gradient estimates of the log density of $\mathcal{N}(0,1)$.

Bayesian Inference

Estimate the posterior $p(z|x)$.

- Particle-Based Variational Inference

Variational Inference

$$\min_{q \in \mathcal{Q}} \mathrm{KL}(q, p)\,.$$

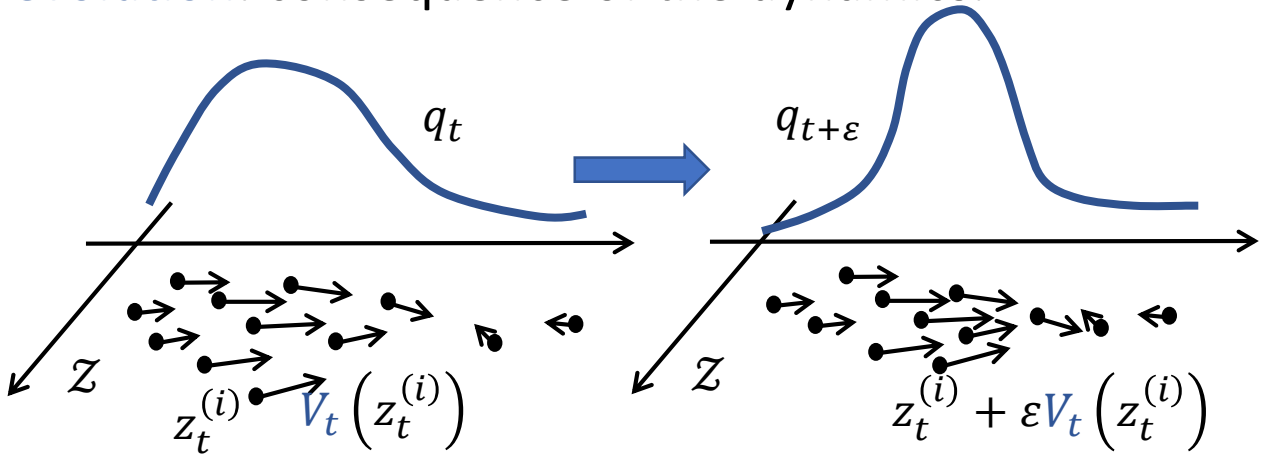
- Particle-based variational inference: use particles $\{z^{(i)}\}_{i=1}^{N}$ to represent $q(z)$.

  Non-parametric q: more particles, more flexible.

  - Stein Variational Gradient Descent (SVGD) [LW16]:

    Update the particles by a dynamics $\frac{\mathrm{d}z_t}{\mathrm{d}t} = V_t(z_t)$ so that KL decreases.

    - Distribution evolution: consequence of the dynamics.



$\partial_t q_t = -\mathrm{div}(q_t V_t) = -\nabla \cdot (q_t V_t)$ (continuity Eq.)

Variational Inference

$$\min_{q \in \mathcal{Q}} \mathrm{KL}(q, p)\,.$$

- Particle-based variational inference: use particles $\{z^{(i)}\}_{i=1}^{N}$ to represent $q(z)$.
  - Stein Variational Gradient Descent (SVGD) [LW16]:

    Update the particles by a dynamics $\frac{\mathrm{d}z_t}{\mathrm{d}t} = V_t(z_t)$ so that KL decreases.
    - Decrease KL:

$$V_t^* := \arg\max_{V_t} \left\{ -\frac{\mathrm{d}}{\mathrm{d}t}\mathrm{KL}(q_t, p) = \mathbb{E}_{q_t} \underbrace{[V_t \cdot \nabla \log p + \nabla \cdot V_t]}_{\text{Stein Operator } \mathcal{A}_p[V_t]} \right\}.$$

    For tractability,

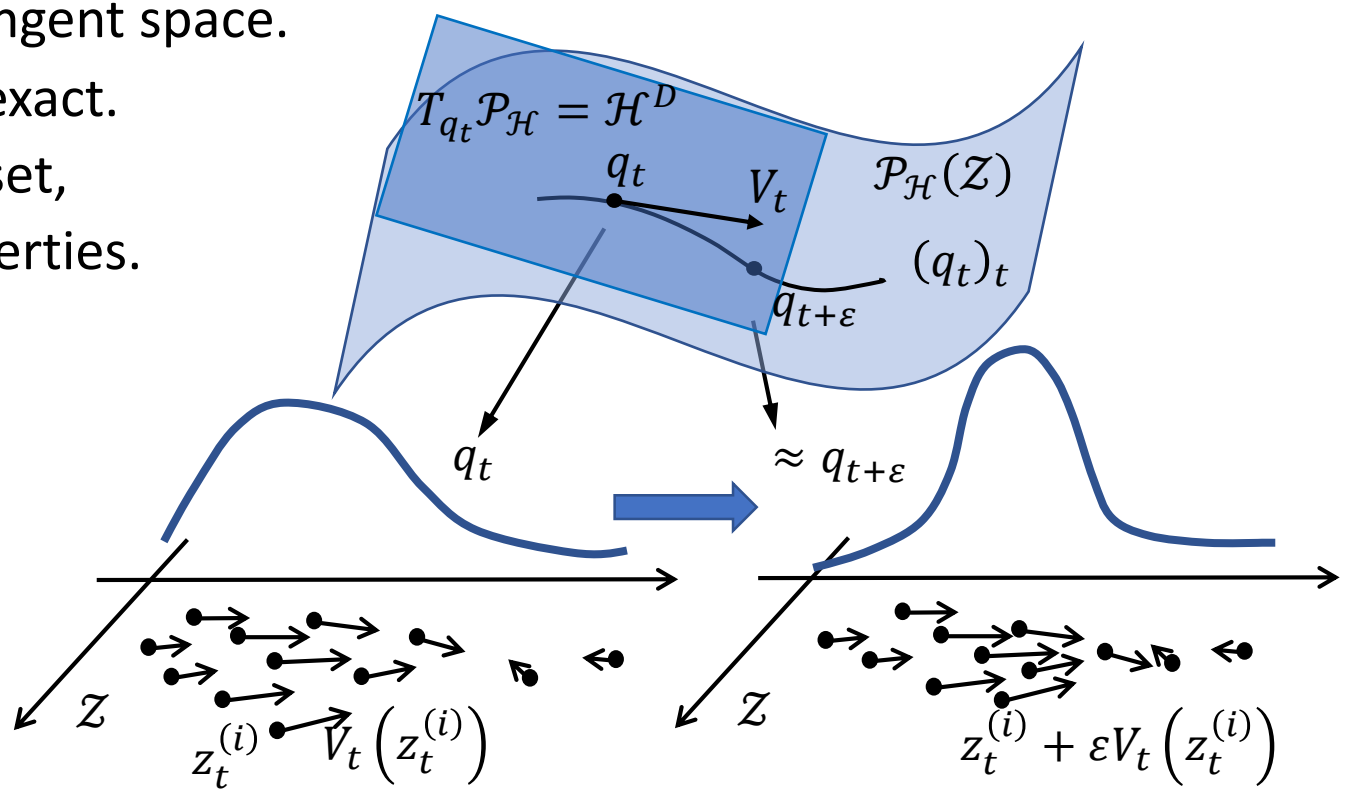
$$V_t^{\mathrm{SVGD}} := \max\cdot \arg \max_{V_t \in \mathcal{H}^D, \|V_t\|=1} \mathbb{E}_{q_t}[V_t \cdot \nabla \log p + \nabla \cdot V_t]$$
$$= \mathbb{E}_{q(z')}[K(z', \cdot)\nabla_{z'} \log p(z') + \nabla_{z'} K(z', \cdot)].$$

    Update rule: $z^{(i)} \mathrel{+}= \varepsilon\left[\sum_j K_{ij}\, \nabla_{z^{(j)}} \log p\left(z^{(j)}\right) + \sum_j \nabla_{z^{(j)}} K_{ij}\right]$.

$= \sum_j \left(z^{(i)} - z^{(j)}\right) K_{ij}$
for Gaussian Kernel:
Repulsive force!

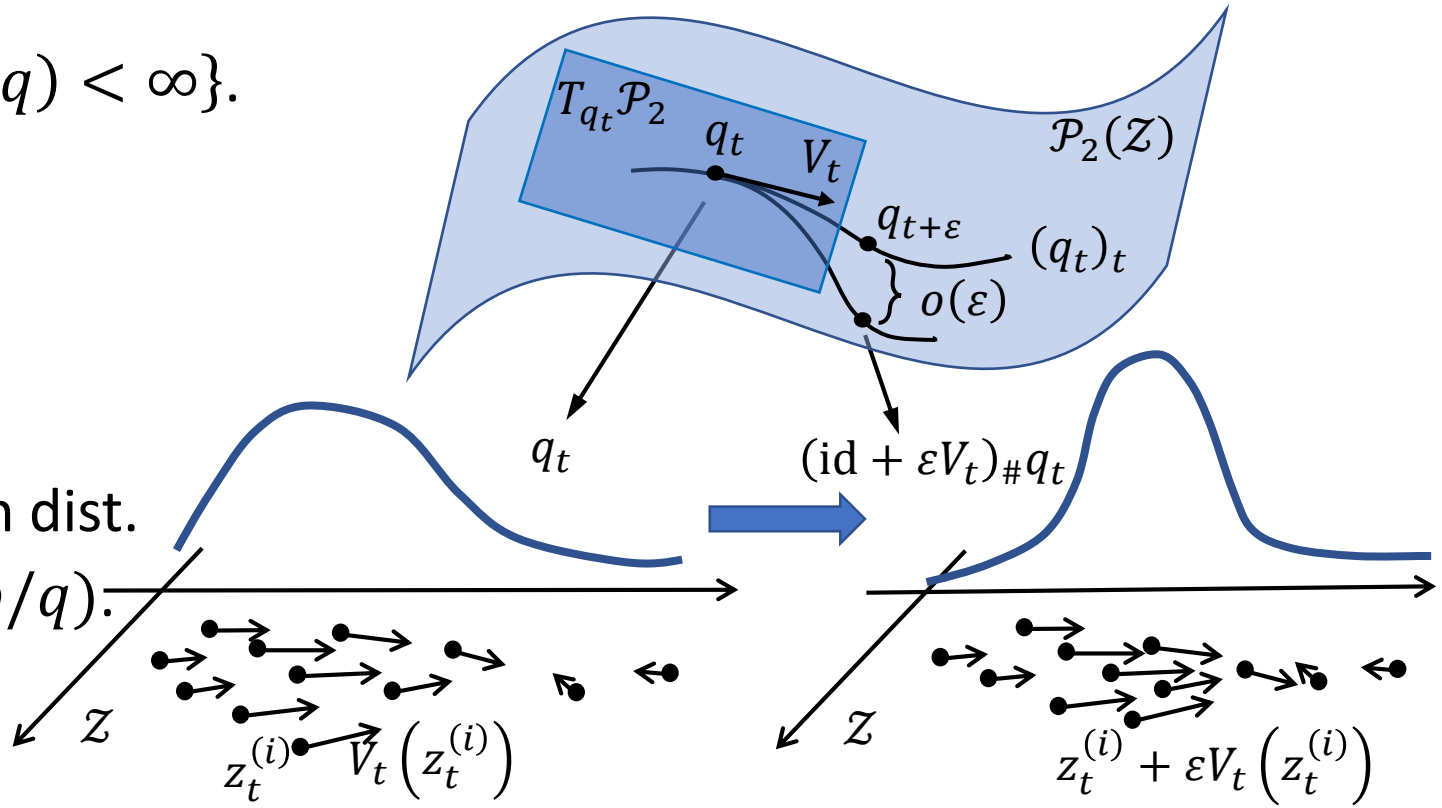
Variational Inference

- Particle-based variational inference: use particles $\{z^{(i)}\}_{i=1}^{N}$ to represent $q(z)$.
  - SVGD as Gradient Flow [Liu17]:
    V_t^{SVGD} is the gradient of $\mathrm{KL}(q_t, p)$ on $\mathcal{P}_{\mathcal{H}}(\mathcal{Z})$ at q_t.
    - $\mathcal{P}_{\mathcal{H}}(\mathcal{Z})$: has $\mathcal{H}^D$ as its tangent space.
    - Explanation of V_t^{SVGD} is exact.
    - $\mathcal{P}_{\mathcal{H}}(\mathcal{Z})$ is unknown as a set, and lacks appealing properties.



Variational Inference

- Particle-based variational inference: use particles $\{z^{(i)}\}_{i=1}^{N}$ to represent $q(z)$.
  - Wasserstein Gradient Flow [BB00, Ott01, Vil08, AGS08]:
    - Wasserstein space:
      $\mathcal{P}_2(\mathcal{Z}) := \{\text{distr. on } \mathcal{Z} | \text{Var}(q) < \infty\}$.
    - Wasserstein tangent vector
      $\Leftrightarrow$ vector field on $\mathcal{Z}$.
    - $T_q\mathcal{P}_2 = \overline{\{\nabla\varphi | \varphi \in \mathcal{C}_c^\infty\}}^{\mathcal{L}^2_q}$,
      $\langle V, U\rangle_{T_q\mathcal{P}_2} = \mathbb{E}_q[V \cdot U]$.
    - Consistent with Wasserstein dist.
    - $-\text{grad}_q \text{KL}(q,p) = \nabla \log(p/q)$.
    - $\text{Exp}_q(V) = (\text{id} + V)_\# q$.



Variational Inference

- Particle-based variational inference: use particles $\{z^{(i)}\}_{i=1}^{N}$ to represent $q(z)$.
  - Wasserstein Gradient Flow (WGF)

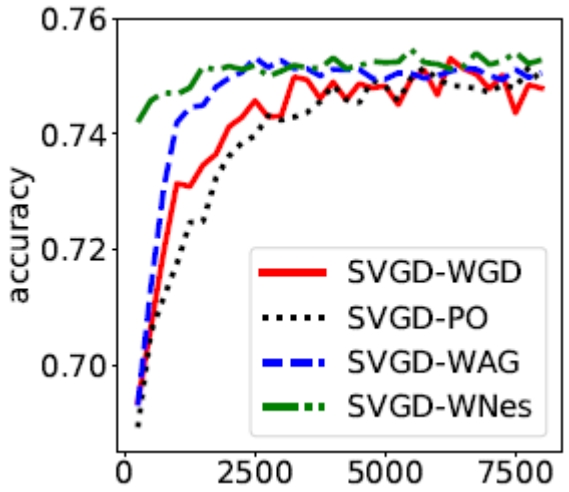
$$V := -\mathrm{grad}_q \,\mathrm{KL}(q,p) = \nabla \log(p/q)\,,$$
$$z^{(i)} \mathrel{+}= \varepsilon V\left(z^{(i)}\right).$$

  $V\left(z^{(i)}\right) \approx$

  - Blob [CZW+18]: $\nabla_{z^{(i)}} \log p\left(z^{(i)}\right) - \frac{\sum_j \nabla_{z^{(i)}} K_{ij}}{\sum_k K_{ik}} - \sum_j \frac{\nabla_{z^{(i)}} K_{ij}}{\sum_k K_{jk}}$.
  - GFSD [LZC+19]: $\nabla_{z^{(i)}} \log p\left(z^{(i)}\right) - \frac{\sum_j \nabla_{z^{(i)}} K_{ij}}{\sum_k K_{ik}}$.
  - GFSF [LZC+19]: $\nabla_{z^{(i)}} \log p\left(z^{(i)}\right) + \sum_{j,k} (K^{-1})_{ik} \nabla_{z^{(j)}} K_{kj}$.
  - [LZC+19]: particle-based VIs approximate WGF with a *compulsory* smoothing assumption, in either of the two *equivalent* forms of *smoothing the density* (Blob, GFSD) or *smoothing functions* (SVGD, GFSF).

Variational Inference

- Particle-based variational inference: use particles $\{z^{(i)}\}_{i=1}^{N}$ to represent $q(z)$.
  - Acceleration on the Wasserstein space [LZC+19]:
    - Apply Riemannian Nesterov's methods to $\mathcal{P}_2(\mathcal{Z})$.



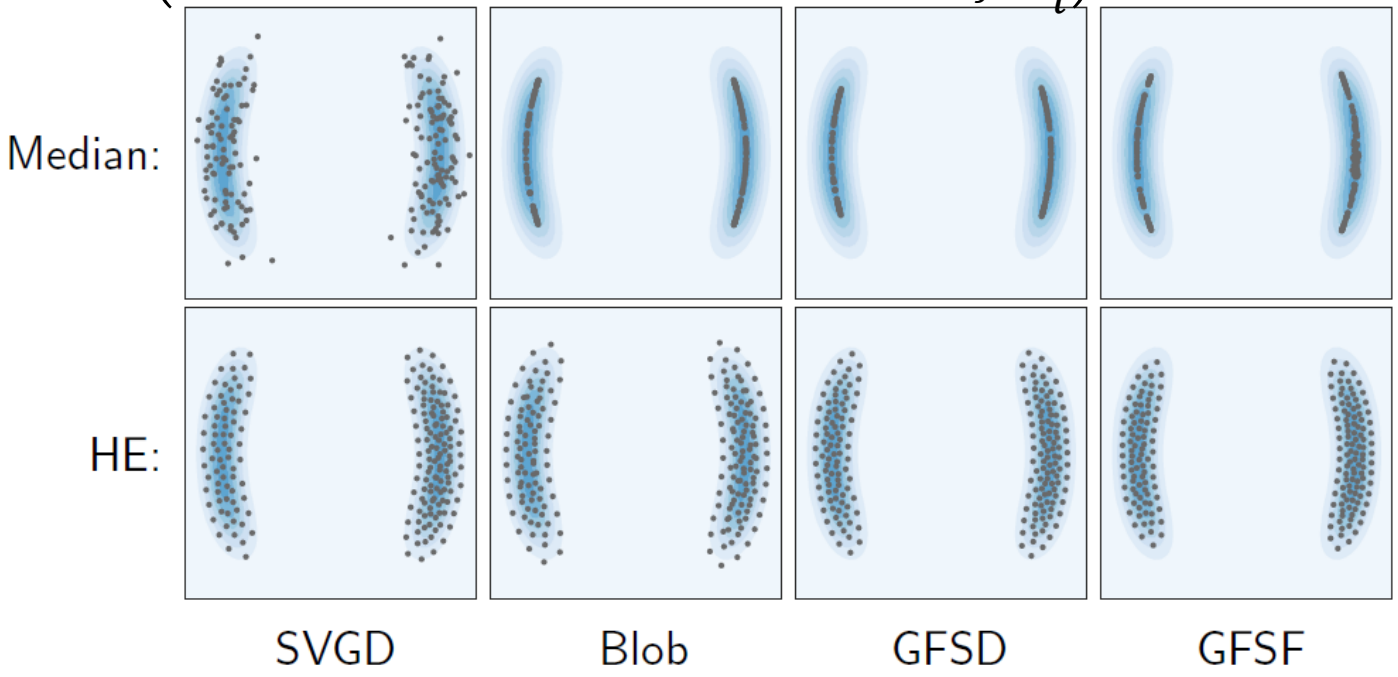
Inference for Bayesian logistic regression

Algorithm 1 The acceleration framework with Wasserstein Accelerated Gradient (WAG) and Wasserstein Nesterov's method (WNes)

WAG: select acceleration factor $\alpha > 3$;
WNes: select or calculate $c_1, c_2 \in \mathbb{R}^+$ (Appendix C.2);
Initialize $\{x_0^{(i)}\}_{i=1}^N$ distinctly; let $y_0^{(i)} = x_0^{(i)}$;
for $k = 1, 2, \cdots, k_{\max}$, **do**
  for $i = 1, \cdots, N$, **do**
    Find $v(y_{k-1}^{(i)})$ by SVGD/Blob/GFSD/GFSF;
    $x_k^{(i)} = y_{k-1}^{(i)} + \varepsilon v(y_{k-1}^{(i)})$;
    $y_k^{(i)} = x_k^{(i)} +$
    $\begin{cases} \text{WAG: } \frac{k-1}{k}(y_{k-1}^{(i)} - x_{k-1}^{(i)}) + \frac{k+\alpha-2}{k}\varepsilon v(y_{k-1}^{(i)}); \\ \text{WNes: } c_1(c_2-1)(x_k^{(i)} - x_{k-1}^{(i)}); \end{cases}$
  end for
end for
Return $\{x_{k_{\max}}^{(i)}\}_{i=1}^N$.

Variational Inference

- Particle-based variational inference: use particles $\left\{z^{(i)}\right\}_{i=1}^{N}$ to represent $q(z)$.
  - Kernel bandwidth selection:
    - Median [LW16]: median of pairwise distances of the particles.
    - HE [LZC+19]: the two approx. to $q_{t+\varepsilon}(x)$, i.e., $\tilde{q}\left(x;\left\{x^{(j)}\right\}_j\right)+\varepsilon\Delta_x\tilde{q}\left(x;\left\{x^{(j)}\right\}_j\right)$ (Heat Eq.) and $\tilde{q}\left(x;\left\{x^{(i)}-\varepsilon\nabla_{x^{(i)}}\log\tilde{q}\left(x^{(i)};\left\{x^{(j)}\right\}_j\right)\right\}_i\right)$ (particle evolution), should match.



Variational Inference

- Particle-based variational inference: use particles $\{z^{(i)}\}_{i=1}^{N}$ to represent $q(z)$.
  - Add particles dynamically: [CMG18, FCSS18].
  - Solve the Wasserstein gradient by optimal transport: [CZ17, CZW+18].
  - Manifold support space: [LZ18].
  - Asymptotic analysis: SVGD [Liu17] ($N \to \infty, \varepsilon \to 0$).
  - Non-asymptotic analysis
    - w.r.t ε: e.g., [RT96] (as WGF).
    - w.r.t N: [CMG18, FCSS18, ZZC18].

Bayesian Inference

Estimate the posterior $p(z|x)$.

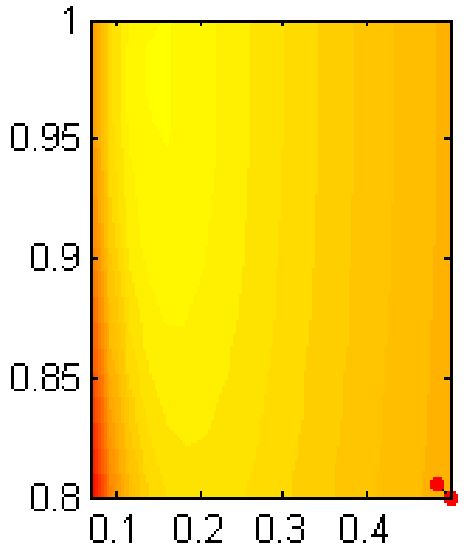
- MCMC

MCMC

- Monte Carlo
  - Directly draw (i.i.d.) samples from $p(z|x)$.
  - Almost always impossible to directly do so.
- Markov Chain Monte Carlo (MCMC):

  Simulate a Markov chain whose stationary distribution is $p(z|x)$.
  - Easier to implement: only requires unnormalized $p(z|x)$ (*e.g.*, $p(z,x)$).
  - Asymptotically accurate.
  - Drawback/Challenge: sample auto-correlation.

    Less effective than i.i.d. samples.



[GC11]

MCMC

- A fantastic MCMC animation site: https://chi-feng.github.io/mcmc-demo/

The Markov-chain Monte Carlo Interactive Gallery

Click on an algorithm below to view interactive demo:

- Random Walk Metropolis Hastings
- Adaptive Metropolis Hastings [1]
- Hamiltonian Monte Carlo [2]
- No-U-Turn Sampler [2]
- Metropolis-adjusted Langevin Algorithm (MALA) [3]
- Hessian-Hamiltonian Monte Carlo (H2MC) [4]
- Stein Variational Gradient Descent (SVGD) [5]
- Nested Sampling with RadFriends (RadFriends-NS) [6]

View the source code on github: https://github.com/chi-feng/mcmc-demo.

MCMC

Classical MCMC

- Metropolis-Hastings framework [MRR+53, Has70]:

  Draw $z^* \sim q\left(z^* | z^{(k)}\right)$ and take $z^{(k+1)}$ as z^* with probability

$$\min\left\{1, \frac{q\left(z^{(k)} \middle| z^*\right) p(z^* | x)}{q\left(z^* \middle| z^{(k)}\right) p\left(z^{(k)} \middle| x\right)}\right\},$$

  else take $z^{(k+1)}$ as $z^{(k)}$.

- Importance sampling [Nea01, ADDJ03]:

$$\mathbb{E}_{p(z|x)}[f(z)] = \mathbb{E}_{q(z)}\left[f(z)\frac{p(z|x)}{q(z)}\right].$$

  Draw sample $z^{(i)}$ from $q(z)$ and assign it with importance/weight

$$\frac{p\left(z^{(i)} \middle| x\right)}{q(z^{(i)})} \propto \frac{p\left(z^{(i)}, x\right)}{q(z^{(i)})}.$$

MCMC

Classical MCMC

- Gibbs sampling [GG87]:

  Iteratively sample from conditional distributions, which are easier to draw:

$$
\begin{aligned}
z_1^{(1)} &\sim p\left(z_1 \middle| \quad z_2^{(0)}, z_3^{(0)}, \dots, z_d^{(0)}, x\right), \\
z_2^{(1)} &\sim p\left(z_2 \middle| z_1^{(1)}, \quad z_3^{(0)}, \dots, z_d^{(0)}, x\right), \\
z_3^{(1)} &\sim p\left(z_3 \middle| z_1^{(1)}, z_2^{(1)}, \quad \dots, z_d^{(0)}, x\right), \\
&\dots, \\
z_i^{(k+1)} &\sim p\left(z_i \middle| z_1^{(k+1)}, \dots, z_{i-1}^{(k+1)}, \quad z_{i+1}^{(k)}, \dots, z_d^{(k)}, x\right).
\end{aligned}
$$

MCMC

Dynamics-based MCMC

- Simulates a jump-free continuous-time Markov process (dynamics):

$$\mathrm{d}z = \underbrace{b(z)\,\mathrm{d}t}_{\text{drift}} + \underbrace{\sqrt{2D(z)}\,\mathrm{d}B_t(z)}_{\text{diffusion}},$$

$$\Delta z = b(z)\varepsilon + \mathcal{N}(0, 2D(z)\varepsilon) + o(\varepsilon),$$

Pos. semi-def. matrix

Brownian motion

  with appropriate $b(z)$ and $D(z)$ so that $p(z|x)$ is kept stationary/invariant.

- Informative transition using gradient $\nabla_z \log p(z|x)$.
- Some are compatible with *stochastic gradient* (SG): more efficient.

$$\nabla_z \log p(z|x) = \nabla_z \log p(z) + \sum_{d \in \mathcal{D}} \nabla_z \log p(x_d|z)\,,$$

$$\tilde{\nabla}_z \log p(z|x) = \nabla_z \log p(z) + \frac{|\mathcal{D}|}{|\mathcal{S}|} \sum_{d \in \mathcal{S}} \nabla_z \log p(x_d|z)\,, \mathcal{S} \subset \mathcal{D}.$$

MCMC

Dynamics-based MCMC

- Langevin dynamics [Lan08]:

$$\mathrm{d}z = \nabla \log p \, \mathrm{d}t + \sqrt{2} \, \mathrm{d}B_t(z).$$

Algorithm (also called Metropolis Adapted Langevin Algorithm) [RS02]:

$$z^{(k+1)} = z^{(k)} + \varepsilon \nabla \log p\left(z^{(k)} \middle| x\right) + \mathcal{N}(0, 2\varepsilon),$$

followed by an MH step.

MCMC

Dynamics-based MCMC

- Hamiltonian Dynamics: $\begin{cases} \mathrm{d}z = \Sigma^{-1} r \, \mathrm{d}t, \\ \mathrm{d}r = \nabla \log p \ \mathrm{d}t. \end{cases}$
- Algorithm: Hamiltonian Monte Carlo [DKPR87, Nea11, Bet17]

  Draw $r \sim \mathcal{N}(0, \Sigma)$ and simulate K steps:

$$\begin{cases} r^{(k+1/2)} = r^{(k)} + (\varepsilon/2)\nabla_z \log p\left(z^{(k)} \middle| x\right), \\ z^{(k+1)} = z^{(k)} + \varepsilon \Sigma^{-1} r^{(k+1/2)}, \\ r^{(k+1)} = r^{(k+1/2)} + (\varepsilon/2)\nabla_z \log p\left(z^{(k+1)} \middle| x\right), \end{cases}$$

  and do an MH step, for one sample of z.

  - Störmer-Verlet (leap-frog) integrator:
    - Makes MH ratio close to 1.
    - Higher-order simulation error.
  - More distant exploration than LD (less auto-correlation).

MCMC

Dynamics-based MCMC: using stochastic gradient (SG).

- Langevin dynamics is compatible with SG [WT11, CDC15, TTV16].
- Hamiltonian Monte Carlo is incompatible with SG [CFG14, Bet15]:

  the stationary distribution is changed.
- Stochastic Gradient Hamiltonian Monte Carlo [CFG14]:

$$\begin{cases} \mathrm{d}z = \Sigma^{-1} r \, \mathrm{d}t, \\ \mathrm{d}r = \nabla \log p \, \mathrm{d}t - C\Sigma^{-1} r \, \mathrm{d}t + \sqrt{2C} \, \mathrm{d}B_t(r). \end{cases}$$

  - Asymptotically, stationary distribution is p.
  - Non-asymptotically (with Euler integrator), gradient noise is of higher-order of Brownian-motion noise.

MCMC

Dynamics-based MCMC: using stochastic gradient.

- Stochastic Gradient Nose-Hoover Thermostats [DFB+14] (scalar C):

$$\begin{cases} \mathrm{d}z = \Sigma^{-1} r \, \mathrm{d}t, \\ \mathrm{d}r = \nabla \log p \, \mathrm{d}t - \xi r \, \mathrm{d}t + \sqrt{2C\Sigma} \, \mathrm{d}B_t(r), \\ \mathrm{d}\xi = \left(\frac{1}{D} r^{\top} \Sigma^{-1} r - 1 \right) \mathrm{d}t. \end{cases}$$

  - Thermostats $\xi \in \mathbb{R}$: adaptively balance the gradient noise and the Brownian-motion noise.

MCMC

Dynamics-based MCMC

- For manifold support space:
  - LD: [GC11]
  - HMC: [GC11, BSU12, BG13, LSSG15]
  - SGLD: [PT13]
  - SGHMC: [MCF15, LZS16]
  - SGNHT: [LZS16]
- Different kinetic energy (other than Gaussian):
  - Monomial Gamma [ZWC+16, ZCG+17].
- Fancy Dynamics:
  - Relativistic: [LPH+16]
  - Magnetic: [TRGT17]

MCMC

Dynamics-based MCMC

- Complete recipe for the dynamics [MCF15]:

  For any skew-symmetric matrix Q and pos. semi-def. matrix D, the dynamics

$$\mathrm{d}z = b(z)\,\mathrm{d}t + \sqrt{2D(z)}\,\mathrm{d}B_t(z),$$
$$b_i = \frac{1}{p}\sum_j \partial_j \Big(p\big(D_{ij} + Q_{ij}\big)\Big),$$

  keeps p stationary/invariant.

  - The inverse also holds:

    Any dynamics that keeps p stationary can be cast into this form.
  - If D is pos. def., p is the unique stationary distribution.

- Integrators and their non-asymptotic analysis (with SG): [CDC15].

MCMC and Particle-Based VI

- MCMC: dynamics that keeps p invariant.

  ParVI: dynamics that drives q_t to minimize $\mathrm{KL}(q_t, p)$.

- LD: induces gradient flow of KL on $\mathcal{P}_2$ [JKO98].

  ParVI: simulates gradient flow of KL on $\mathcal{P}_2$ [CZW+18, LZC+19].

  - Convergence intuition
    - Exponential asymptotic convergence for convex KL (log-concave p) [e.g., RT96, CB17].
  - Robust to SG.

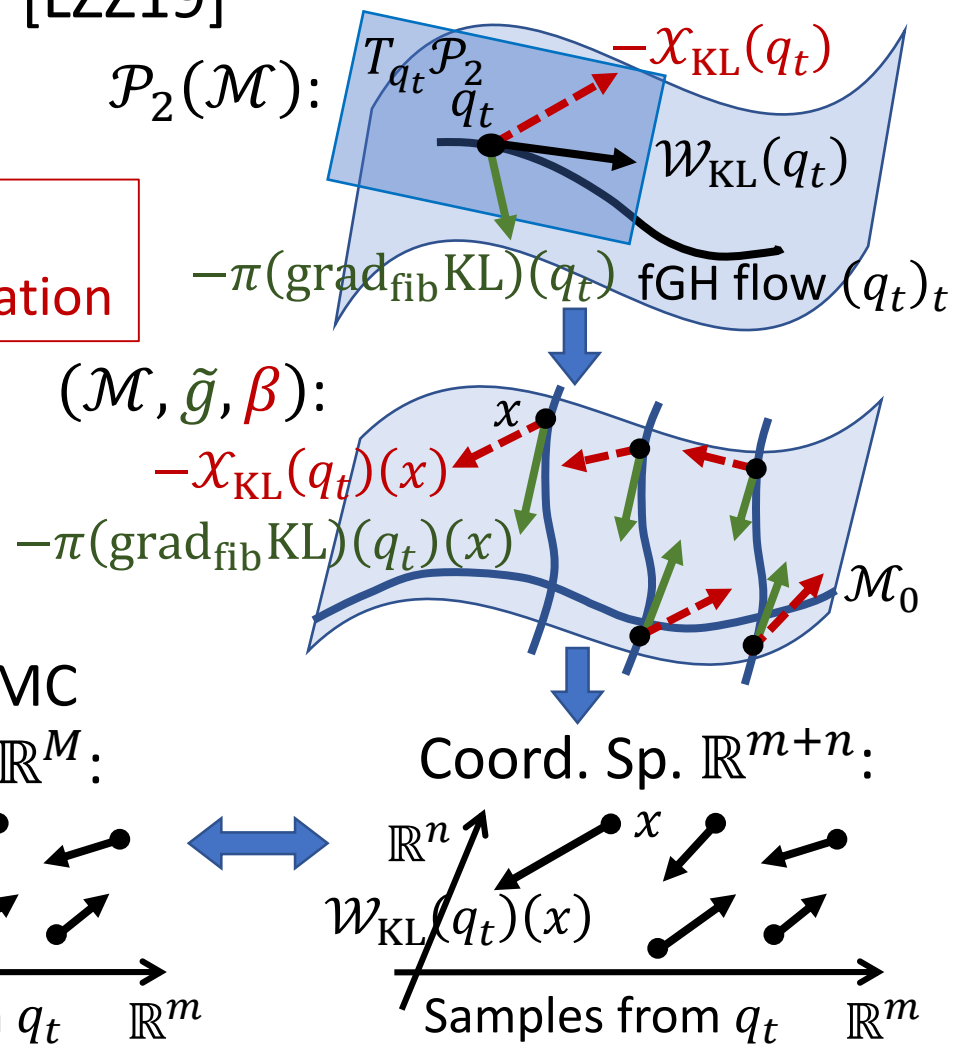
MCMC and Particle-Based VI

- General MCMC: fiber-Gradient Hamiltonian flow [LZZ19]

$$\mathcal{W}_{\mathrm{KL}}(q_t) = -\pi(\mathrm{grad}_{\mathrm{fib}}\mathrm{KL})(q_t) - \mathcal{X}_{\mathrm{KL}}(q_t)$$

Hamiltonian flow of KL on $\mathcal{P}_2$: keeps KL invariant; helps exploration

fiber-gradient flow of KL on $\mathcal{P}_2$: minimizes KL on each fiber (matches conditional distributions)



- Convergence intuitions for general MCMCs.
- New ParVIs using more efficient dynamics other than LD.

Bayesian Inference: Comparison

	Parametric VI	Particle-Based VI	MCMC
Asymptotic Accuracy	No	Yes	Yes
Approximation Flexibility	Limited	Unlimited	Unlimited
Empirical Convergence Speed	High	High	Low
Particle Efficiency	(Do not apply)	High	Low
High-Dimensional Efficiency	High	Low	High

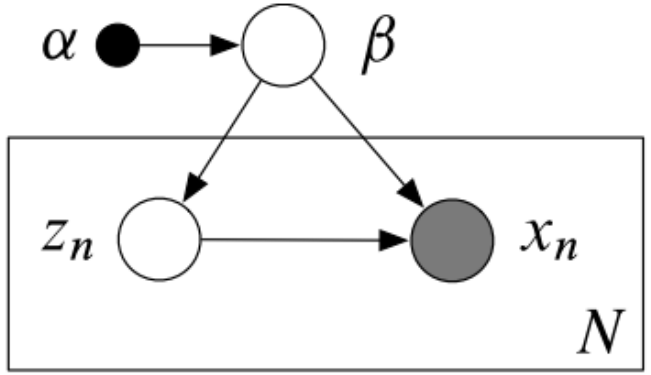
Bayesian Inference

Estimate the posterior $p(z|x)$.

- Amortized Inference

Amortized Inference

- Latent Variable: local and global.



$$p(x, z, \beta|\alpha) = p(\beta|\alpha) \prod_{n=1}^{N} p(x_n, z_n|\beta).$$

- For β to be global and z to be local, $p(x_n, z_n|x_{-n}, z_{-n}, \beta, \alpha) = p(x_n, z_n|\beta, \alpha)$. [HBWP13]
- Typical inference task: $p(\beta, \{z_n\}_{n=1}^N | \{x_n\}_{n=1}^N)$.
- More concerned task:
  - $p(\beta | \{x_n\}_{n=1}^N)$: global representation of the whole data set.
  - $p(z_n|x_n, \beta)$: fast $x_n \to z_n$ (feature map) desired (also global information).

Amortized Inference

- Latent Variable: local and global.

$$p(x, z, \beta|\alpha) = p(\beta|\alpha) \prod_{n=1}^{N} p(x_n, z_n|\beta)\,.$$

- Amortized variational inference: parametric, explicit
  - [HBWP13]: stochastic variational inference.
    - Assume $p(\beta \mid x, z, \alpha) = h(\beta) \exp\{\eta_g(x, z, \alpha)^\top t(\beta) - a_g(\eta_g(x, z, \alpha))\}$

$$p(z_{nj} \mid x_n, z_{n,-j}, \beta) = h(z_{nj}) \exp\{\eta_\ell(x_n, z_{n,-j}, \beta)^\top t(z_{nj}) - a_\ell(\eta_\ell(x_n, z_{n,-j}, \beta))\}.$$

    - Take variational distribution:

$$q(z, \beta) = q(\beta \mid \lambda) \prod_{n=1}^{N} \prod_{j=1}^{J} q(z_{nj} \mid \phi_{nj}).$$

$$q(\beta \mid \lambda) = h(\beta) \exp\{\lambda^\top t(\beta) - a_g(\lambda)\}$$

$$q(z_{nj} \mid \phi_{nj}) = h(z_{nj}) \exp\{\phi_{nj}^\top t(z_{nj}) - a_\ell(\phi_{nj})\}.$$

Then: $\nabla_\lambda \mathcal{L} = \nabla_\lambda^2 a_g(\lambda)(\mathbb{E}_q[\eta_g(x, z, \alpha)] - \lambda)$. $\nabla_{\phi_{nj}} \mathcal{L} = \nabla_{\phi_{nj}}^2 a_\ell(\phi_{nj})(\mathbb{E}_q[\eta_\ell(x_n, z_{n,-j}, \beta)] - \phi_{nj})$.

Algorithm:
- Sample a data point x_n uniformly from the data set.
- $\phi_{nj} = \mathbb{E}_{q_\lambda q_{\phi_{n,-j}}}\left[\eta_\ell\left(x_n, z_{n,-j}, \beta\right)\right], j = 1, \ldots, J; \hat{\lambda} = \mathbb{E}_{q_{\phi n}}\left[\eta_g\left(x_n^{(N)}, z_n^{(N)}, \alpha\right)\right]$.
- $\lambda \leftarrow (1 - \rho_t)\lambda + \rho_t \hat{\lambda}$.

Amortized Inference

- Latent Variable: local and global.

$$p(x, z, \beta|\alpha) = p(\beta|\alpha) \prod_{n=1}^{N} p(x_n, z_n|\beta)\,.$$

- Amortized variational inference: parametric, explicit
  - [KW14, RMW14, RM15]: use a reparameterizable encoder for $q_\phi(z_n|x_n)$ with available p.d.f.:

$$z_n = g_\phi(x_n, \epsilon), \epsilon \sim q(\epsilon).$$

  - Objective for inferring z:

$$\begin{aligned}
&\mathbb{E}_{\hat{p}(x)}[\mathrm{ELBO}(x)] \\
&\approx \frac{1}{N} \sum_{n=1}^{N} \mathbb{E}_{q_\phi(z_n|x_n)}[\log p_\theta(x_n, z_n, \beta)] - \mathbb{E}_{q_\phi(z_n|x_n)}\left[\log q_\phi(z_n|x_n)\right] \\
&= \frac{1}{N} \sum_{n=1}^{N} \mathbb{E}_{q(\epsilon)}\left[\log p_\theta\left(x_n, g_\phi(x_n, \epsilon), \beta\right) - \log q_\phi\left(g_\phi(x_n, \epsilon)\middle|x_n\right)\right].
\end{aligned}$$

    Estimate the gradient by samples of $q(\epsilon)$.

Amortized Inference

- Latent Variable: local and global.

$$p(x, z, \beta|\alpha) = p(\beta|\alpha) \prod_{n=1}^{N} p(x_n, z_n|\beta)\,.$$

- Amortized variational inference: parametric, explicit
  - [KW14, RMW14, RM15]: use a reparameterizable encoder for $q_\phi(z_n|x_n)$ with available p.d.f.:

$$z_n = g_\phi(x_n, \epsilon), \epsilon \sim q(\epsilon).$$

    - For inferring β,

$$\mathcal{L}[q(\beta)] = \mathbb{E}_{q(\beta)}[\log p(\{x_n\}_{n=1}^N, \beta) - \log q(\beta)]$$

$$\geq \mathbb{E}_{q(\beta)}\left[\sum_{n=1}^{N} \mathrm{ELBO}(x_n) - \log q(\beta)\right].$$

Amortized Inference

- Latent Variable: local and global.

$$p(x, z, \beta|\alpha) = p(\beta|\alpha) \prod_{n=1}^{N} p(x_n, z_n|\beta)\,.$$

- Amortized variational inference: parametric, implicit
  - [MNG17]:

$$\mathbb{E}_{\hat{p}(x)}[\mathrm{ELBO}(x)] = \mathbb{E}_{\hat{p}(x)q_\phi(z|x)}\left[\log p_\theta(x|z) - \log \frac{q_\phi(z|x)}{p_\theta(z)}\right].$$

  - [Hus17, TRB17]:

$$\mathbb{E}_{\hat{p}(x)}[\mathrm{ELBO}(x)] = -\mathbb{E}_{\hat{p}(x)q_\phi(z|x)}\left[\log \frac{\hat{p}(x)q_\phi(z|x)}{p_\theta(z, x)}\right] + \mathbb{E}_{\hat{p}(x)}[\log \hat{p}(x)].$$

  $q_\phi(z|x)$ is reparameterizable.

  Estimate the density ratio by adversarial learning.

Amortized Inference

- Latent Variable: local and global.

$$p(x, z, \beta|\alpha) = p(\beta|\alpha) \prod_{n=1}^{N} p(x_n, z_n|\beta)\,.$$

- Amortized MCMC:
  - [LTL17]: use MCMC update rule to guide a generator $q_\phi(z|x)$:

$$\phi_{t+1} = \arg\min_{\phi} \sum_{N=1}^{N} D\left(\underbrace{(\mathcal{K}_T)_{\#} q_{\phi_t}(z_n|x_n)}_{q_T}, q_{\phi_t}(z_n|x_n)\right),$$

    where $\mathcal{K}_T$ is the T-step transition of an MCMC method.

    q_T is easy to sample.

    - $D = \mathrm{KL}$: $\phi_{t+1} = \arg\min_{\phi} -\sum_{n=1}^{N} \mathbb{E}_{q_T}\left[\log q_\phi(z_n|x_n)\right]$.
    - $D = \mathrm{JS}$: adversarial learning

$$\mathrm{D_{JS}}[q_T||q] \geq \mathrm{D_{adv}}[\{\boldsymbol{z}_T^k\}||\{\boldsymbol{z}_0^k\}] = \frac{1}{K}\sum_{k=1}^{K} \log \sigma(d_{\boldsymbol{\psi}}(\boldsymbol{z}_T^k|\boldsymbol{x})) + \frac{1}{K}\sum_{k=1}^{K} \log(1 - \sigma(d_{\boldsymbol{\psi}}(\boldsymbol{z}_0^k|\boldsymbol{x}))),$$

Amortized Inference

- Latent Variable: local and global.

$$p(x, z, \beta|\alpha) = p(\beta|\alpha) \prod_{n=1}^{N} p(x_n, z_n|\beta) .$$

- Amortized particle-based VI:
  - [FWL17]: generator $q_\phi(z|x)$: $z = g_\phi(\epsilon, x), \epsilon \sim q(\epsilon)$.

$$\phi_{t+1} = \arg\min_{\phi} \sum_{n=1}^{N} \mathbb{E}_{q(\epsilon)} \left[\left\| g_\phi(x_n, \epsilon) - K_1 g_{\phi_t}(x_n, \epsilon) \right\|_2^2 \right],$$

  where K_1 is the one-step update of SVGD.

$$\phi_{t+1} \approx \phi_t + \varepsilon \sum_{n=1}^{N} \sum_{i=1}^{M} \nabla_\phi g_{\phi_t}(x_n, \epsilon_i) \, V^{\mathrm{SVGD}} \left(g_{\phi_t}(x_n, \epsilon_i) \right).$$

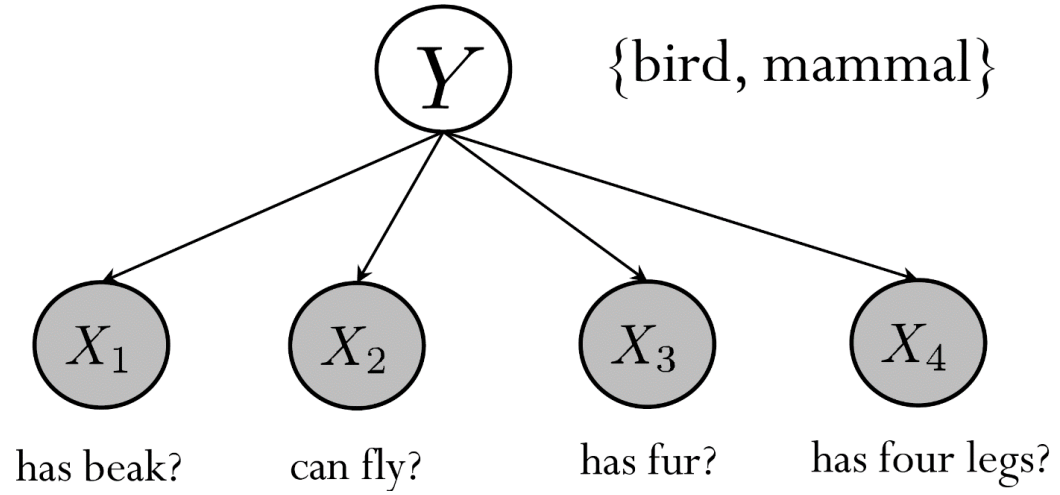
  For β, $\nabla_\beta \log p(\beta | \{x_n\}_{n=1}^N) = \sum_{n=1}^{N} \mathbb{E}_{p(z_n|\beta, x_n)} \left[\nabla_\beta \log p(x_n, z_n, \beta) \right]$.

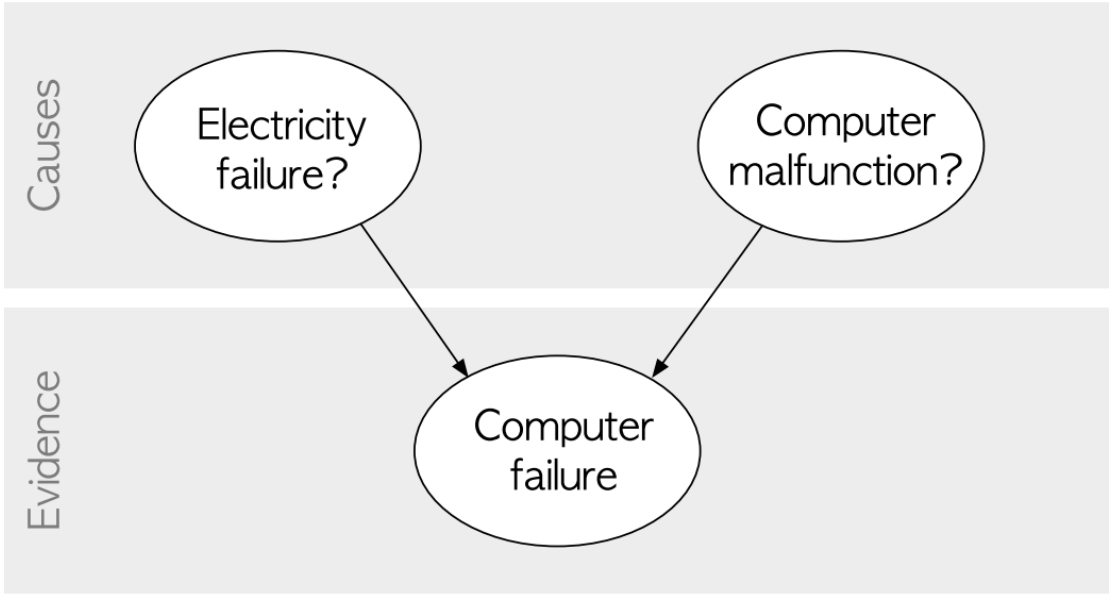
Bayesian Models

Bayesian Networks

Bayesian Networks

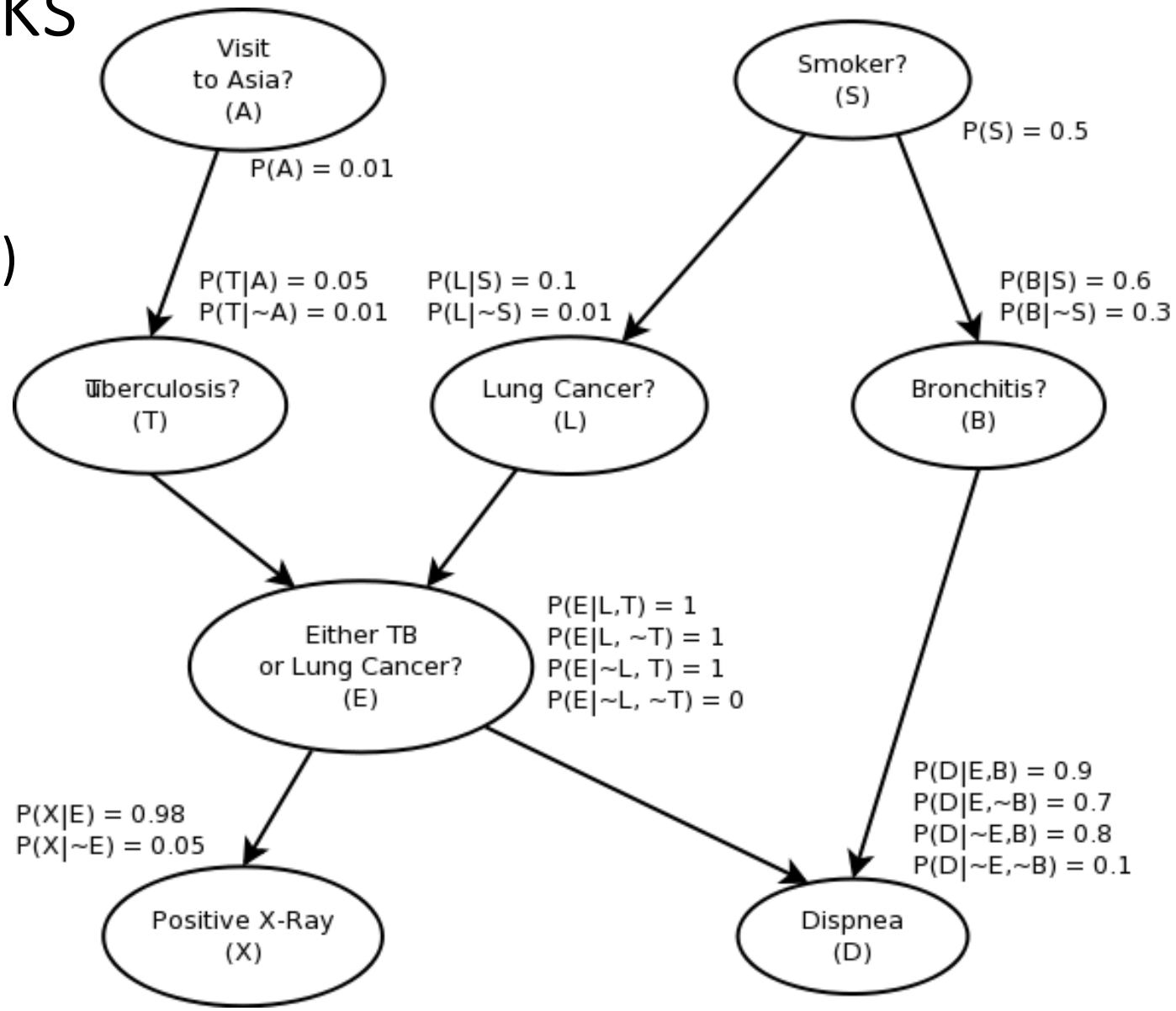
- Classical BayesNets
  (Causal Networks /
  Directed Graphical Models)





Bayesian Networks

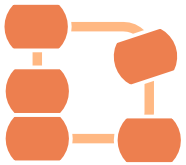
- Classical BayesNets (Causal Networks / Directed Graphical Models)



[http://sujitpal.blogspot.com/2013/07/bayesian-network-inference-with-r-and.html]

Bayesian Networks

- Classical BayesNets
  - Interpretable and incorporates domain knowledge.
  - Variable Dependency <=====> Graph Structure
    - Markov blanket, independency map, ...
  - Approximate Bayes Computation (ABC) for large networks
    - Message passing, belief propagation, expectation propagation, ...
- General BayesNets:
  - Simple graph structure, but complex likelihood.
  - Reduce human knowledge requirement, increase model flexibility.
  - Sigmoid Belief Networks, Latent Dirichlet Allocation, Variational Auto-Encoders, ...
  - Programming Tools:
    - ZhuSuan [SCZ+17]: probabilistic programming library.
      - Easy Bayesian modeling, especially with deep neural nets.
      - Various convenient Bayesian inference tools.
      - Python interface.



ZHUSUAN

Bayesian Models

Bayesian Networks

- Sigmoid Belief Networks

Sigmoid Belief Networks

- Model structure [Nea92]:

$$P(S_i = s_i \mid S_j = s_j : j < i) = \sigma\left(s_i^* \sum_{j<i} s_j w_{ij}\right).$$

$$P(\widetilde{S} = \widetilde{s}) = \prod_i P(S_i = s_i \mid S_j = s_j : j < i)$$

$$= \prod_i \sigma\left(s_i^* \sum_{j<i} s_j w_{ij}\right).$$

  - Inference with MCMC: Gibbs sampling from $P_w(h|v)$ ($s = (h, v)$) [Nea92].

$$P(S_i = x \mid S_j = s_j : j \neq i)$$

$$\propto \sigma\left(x^* \sum_{j<i} s_j w_{ij}\right) \prod_{j>i} \sigma\left(s_j^* \left(x w_{ji} + \sum_{k<j, k\neq i} s_k w_{jk}\right)\right).$$

  - Learning [Nea92]: $\nabla_w \log P_w(v) = \mathbb{E}_{P_w(h|v)}[\nabla_w \log P_w(h, v)]$.

Gibbs sampling

Sigmoid Belief Networks

- Sigmoid Belief Networks [Nea92]

$$P(S_i = s_i \mid S_j = s_j : j < i) = \sigma\left(s_i^* \sum_{j<i} s_j w_{ij}\right).$$

  - Variational Inference: mean-field VI [SJJ96]
    - Mean-field variational distribution: $Q(H|V) = \prod_{i \in H} \mu_i^{S_i} (1-\mu_i)^{1-S_i}$
    - ELBO:

$$\ln P(V) \geq \sum_{ij} J_{ij} \mu_i \mu_j + \sum_i h_i \mu_i - \sum_i \left\langle \ln \left[1 + e^{\sum_j J_{ij} S_j + h_i} \right] \right\rangle - \sum_i \left[\mu_i \ln \mu_i + (1-\mu_i) \ln(1-\mu_i) \right],$$

where $\langle \cdot \rangle$ indicates an expectation value over the mean field distributi

With further relaxation, the gradient can be estimated.

Sigmoid Belief Networks

- Sigmoid Belief Networks [Nea92]

$$P(S_i = s_i \mid S_j = s_j : j < i) = \sigma\left(s_i^* \sum_{j<i} s_j w_{ij}\right).$$

  - Variational Inference: wake-sleep [HDFN95]
    - Variational distribution: recognition model $Q_\phi(h|x)$ (*e.g.,* also an SBN).
    - Wake phase: update the original model.

$$\nabla_\theta \mathcal{L}(x) \approx \frac{1}{n} \sum_{i=1}^{n} \nabla_\theta \log P_\theta(x, h^{(i)})$$

Same as grad. of ELBO

      where $h^{(i)} \sim Q_\phi(h|x)$.
    - Sleep phase: update the recognition model.

$$\nabla_\phi \mathcal{L}(x) = E_{P_\theta(x,h)} \left[\nabla_\phi \log Q_\phi(h|x)\right].$$

Different from grad. of ELBO

    - Intuition: learn the two models symmetrically; coordinate-descent.
    - *"... does not optimize a well-defined objective function and is not guaranteed to converge."* -- [MG14]

Sigmoid Belief Networks

- Sigmoid Belief Networks [Nea92]

$$P(S_i = s_i \mid S_j = s_j : j < i) = \sigma\left(s_i^* \sum_{j<i} s_j w_{ij}\right). \quad (11)$$

  - Variational Inference: NVIL [MG14]
    - Variational distribution: recognition model $Q_\phi(h|x)$ (*e.g.,* also an SBN).
    - Learning: same as the wake phase

$$\nabla_\theta \mathcal{L}(x) \approx \frac{1}{n} \sum_{i=1}^{n} \nabla_\theta \log P_\theta(x, h^{(i)})$$

    - Inference: grad. of the ELBO:

$$\nabla_\phi \mathcal{L}(x) = E_Q[(\log P_\theta(x,h) - \log Q_\phi(h|x)) \times \nabla_\phi \log Q_\phi(h|x)],$$

(since $\nabla_\phi \mathbb{E}_{q_\phi}[f] = \mathbb{E}_{q_\phi}[f \nabla_\phi \log q_\phi]$.)

- Variance reduction required.

  Subtract the mean $C_\psi(x)$, divide by the variance, simplification using the structure.

Bayesian Models

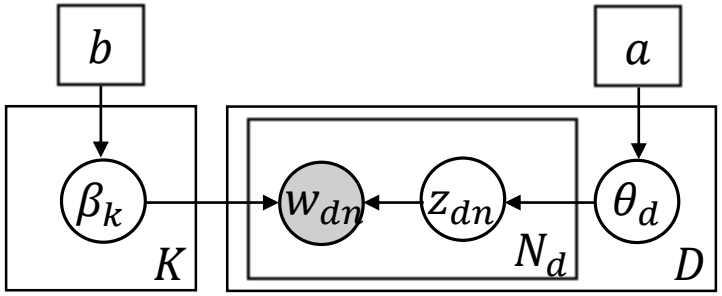
Bayesian Networks

- Topic Models

Latent Dirichlet Allocation

Separate *global* latent feature of dataset and *local* representation of each datum.

- Model Structure [BNJ03]:



  - Data variable: Words/Documents $w = \{w_{dn}\}_{n=1:N_d, d=1:D}$.
  - Latent variables:
    - Global: topics $\beta = \{\beta_k\}$.
    - Local: topic proportions $\theta = \{\theta_d\}$, topic assignments $z = \{z_{dn}\}$.
  - Prior: $p(\beta_k|b) = \mathrm{Dir}(b), p(\theta_d|a) = \mathrm{Dir}(a), p(z_{dn}|\theta_d) = \mathrm{Mult}(\theta_d)$.
  - Likelihood: $p(w_{dn}|z_{dn}, \beta) = \mathrm{Mult}(\beta_{z_{dn}})$.

Latent Dirichlet Allocation

Variational inference [BNJ03]:

- Take variational distribution (mean-field approximation):

$$q_{\lambda,\gamma,\phi}(\beta,\theta,z) \coloneqq \prod_{k=1}^{K} \mathrm{Dir}(\beta_k|\lambda_k) \prod_{d=1}^{D} \mathrm{Dir}(\theta_d|\gamma_d) \prod_{n=1}^{n_d} \mathrm{Mult}(z_{dn}|\phi_{dn}) .$$

- ELBO$(\lambda,\gamma,\phi;a,b)$ is available in closed form.
- E-step: update λ,γ,ϕ by maximizing ELBO;
- M-step: update a,b by maximizing ELBO.

Latent Dirichlet Allocation

MCMC: Gibbs sampling [GS04]

$$p(\beta, \theta, z, w) = AB\left(\prod_{k,w} \beta_{kw}^{N_{kw}+b_w-1}\right)\left(\prod_{d,k} \theta_{dk}^{N_{kd}+a_k-1}\right),$$

$$N_{kw} = \sum_{d=1}^{D}\sum_{n=1}^{N_d} \mathbb{I}(w_{dn} = w, z_{dn} = k)\,, N_{kd} = \sum_{n=1}^{N_d} \mathbb{I}(z_{dn} = k)\,.$$

- β and θ can be collapsed:

$$p(z, w) = \iint p(\beta, \theta, z, w)\,\mathrm{d}\beta\,\mathrm{d}\theta$$
$$= AB\left(\prod_{k} \frac{\prod_w \Gamma(N_{kw} + b_w)}{\Gamma\left(N_k + W\bar{b}\right)}\right)\left(\prod_{d} \frac{\prod_k \Gamma(N_{kd} + a_k)}{\Gamma(N_d + K\bar{a})}\right).$$

- Unacceptable cost to directly compute $p(z|w) = p(z|w)/p(w)$!

Latent Dirichlet Allocation

MCMC: Gibbs sampling [GS04]

$$p(z,w) = AB\left(\prod_k \frac{\prod_w \Gamma(N_{kw}+b_w)}{\Gamma\left(N_k + W\bar{b}\right)}\right)\left(\prod_d \frac{\prod_k \Gamma(N_{kd}+a_k)}{\Gamma(N_d + K\bar{a})}\right).$$

- Use Gibbs sampling: iteratively sample from

$$p\left(z_{11}^{(1)}\middle|\quad z_{12}^{(0)}, z_{13}^{(0)}, z_{14}^{(0)}, \dots, z_{DN_D}^{(0)}, w\right),$$
$$p\left(z_{12}^{(1)}\middle| z_{11}^{(1)},\quad z_{13}^{(0)}, z_{14}^{(0)}, \dots, z_{DN_D}^{(0)}, w\right),$$
$$p\left(z_{13}^{(1)}\middle| z_{11}^{(1)}, z_{12}^{(1)},\quad z_{14}^{(0)}, \dots, z_{DN_D}^{(0)}, w\right),$$
$$\dots,$$
$$p\left(z_{dn}^{(l+1)}\middle| z_{11}^{(l+1)}, \dots, z_{d(n-1)}^{(l+1)},\quad z_{d(n+1)}^{(l)}, \dots, w\right) =: p\left(z_{dn}\middle| z^{-dn}, w\right).$$

$$p\left(z_{dn} = k\middle| z^{-dn}, w\right) \propto \frac{N_{kw}^{-dn} + b_w}{N_k^{-dn} + W\bar{b}}\left(N_{kd}^{-dn} + a_k\right).$$

Latent Dirichlet Allocation

MCMC: Gibbs sampling [GS04]

- For β, use the MAP estimate:

$$\hat{\beta} = \arg\max_{\beta} \log p(\beta|w)\,.$$

Estimate $p(\beta|w) = \mathbb{E}_{p(z|w)}[p(\beta, z, w)]$ with one sample of z from $p(z|w)$:

$$\Longrightarrow \hat{\beta}_k = \frac{N_{kw} + b_w - 1}{N_k + W\bar{b} - W} \approx \frac{N_{kw} + b_w}{N_k + W\bar{b}}\,.$$

- For θ, use the MAP estimate:

$$\hat{\theta}_{dk} = \frac{N_{kd} + a_k - 1}{N_d + K\bar{a} - K} \approx \frac{N_{kd} + a_k}{N_d + K\bar{a}}\,.$$

Latent Dirichlet Allocation

MCMC: Stochastic Gradient Riemannian Langevin Dynamics [PT13]

$$\mathrm{d}x = G^{-1}\nabla \log p \ \mathrm{d}t + \nabla \cdot G^{-1}\,\mathrm{d}t + \mathcal{N}(0, 2G^{-1}\,\mathrm{d}t).$$

- To draw from $p(\beta|w)$,

$$\nabla_\beta \log p(\beta|w) = \frac{1}{p(\beta|w)} \nabla_\beta \int p(\beta, z|w)\,\mathrm{d}z = \int \frac{1}{p(\beta|w)} \nabla_\beta p(\beta, z|w)\,\mathrm{d}z$$

$$= \int \frac{p(\beta, z|w)}{p(\beta|w)} \frac{\nabla_\beta p(\beta, z|w)}{p(\beta, z|w)}\,\mathrm{d}z = \mathbb{E}_{p(z|\beta,w)}\left[\nabla_\beta \log p(\beta, z, w)\right].$$

  - $p(\beta, z, w)$ is available in closed form.
  - $p(z|\beta, w)$ can be drawn using Gibbs sampling.
- Each β_k is on a simplex: use reparameterization to convert to the Euclidean space (that's where G comes from), e.g., $\beta_{kw} = \frac{\pi_{kw}}{\sum_w \pi_{kw}}$.

Latent Dirichlet Allocation

MCMC: Stochastic Gradient Riemannian Langevin Dynamics [PT13]

$$\mathrm{d}x = G^{-1}\nabla \log p \ \mathrm{d}t + \nabla \cdot G^{-1}\,\mathrm{d}t + \mathcal{N}(0, 2G^{-1}\,\mathrm{d}t).$$

- Various parameterizations:

Parameterisation	Reduced-Mean	Reduced-Natural	Expanded-Mean	Expanded-Natural
θ	$\theta_k = \pi_k$	$\theta_k = \log \frac{\pi_k}{1-\sum_k^{K-1}\pi_k}$	$\pi_k = \frac{\lvert\theta_k\rvert}{\sum_{k=1}\lvert\theta_k\rvert}$	$\pi_k = \frac{\mathrm{e}^{\theta_k}}{\sum_{k=1}\mathrm{e}^{\theta_k}}$
$\nabla_\theta \log p(\theta\lvert\mathbf{x})$	$\frac{n+\alpha}{\theta} - \mathbf{1}\frac{n_K+\alpha-1}{\pi_K}$	$n+\alpha-(n.+K\alpha)\,\pi$	$\frac{n+\alpha-1}{\theta} - \frac{n.}{\theta.} - \mathbf{1}$	$n+\alpha-n.\pi-\mathrm{e}^{\theta}$
$G(\theta)$	$n.\left(\mathrm{diag}(\theta)^{-1} + \frac{1}{1-\sum_k\theta_k}\mathbf{1}\mathbf{1}^T\right)$	$\frac{1}{n.}\left(\mathrm{diag}(\pi) - \pi\pi^T\right)$	$\mathrm{diag}\,(\theta)^{-1}$	$\mathrm{diag}\left(\mathrm{e}^{\theta}\right)$
$G^{-1}(\theta)$	$\frac{1}{n.}\left(\mathrm{diag}(\theta) - \theta\theta^T\right)$	$n.\left(\mathrm{diag}(\pi)^{-1} + \frac{1}{1-\sum_k\pi_k}\mathbf{1}\mathbf{1}^T\right)$	$\mathrm{diag}\,(\theta)$	$\mathrm{diag}\left(\mathrm{e}^{-\theta}\right)$
$\sum_{k=1}^{D}\left(G^{-1}\frac{\partial G}{\partial\theta_k}G^{-1}\right)_{jk}$	$K\theta_j - 1$	$\frac{1}{\pi_j^2} - \frac{K-1}{(1-\sum_k\pi_k)^2}$	-1	$\mathrm{e}^{-\theta_j}$
$\sum_{k=1}^{D}\left(G^{-1}(\theta)\right)_{jk}\mathrm{Tr}\left(G^{-1}(\theta)\frac{\partial G}{\partial\theta_k}\right)$	$K\theta_j - 1$	$\frac{1}{\pi_j^2} - \frac{K-1}{(1-\sum_k\pi_k)^2}$	-1	$\mathrm{e}^{-\theta_j}$

Latent Dirichlet Allocation

$$\nabla_\beta \log p(\beta|w) = \mathbb{E}_{p(z|\beta,w)}\left[\nabla_\beta \log p(\beta, z, w)\right].$$

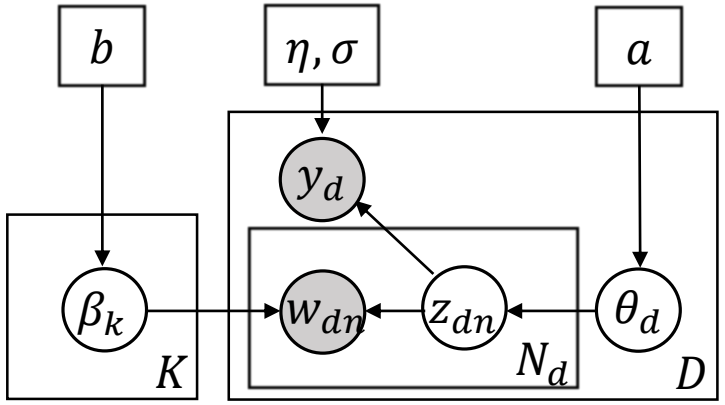
Gibbs Sampling

Closed-form known

- This makes dynamics-based MCMCs and particle-based VIs applicable.
  - Stochastic Gradient Nose-Hoover Thermostats [DFB+14], Stochastic Gradient Riemann Hamiltonian Monte Carlo [MCF15].
  - Accelerated and Hamiltonian particle-based VI [LZC+19, LZZ19]

Supervised Latent Dirichlet Allocation

Model structure [MB08]:



- Generating process:
  - Draw topics: $\beta_k \sim \mathrm{Dir}(b), k = 1, \ldots, K;$
  - For each document d,
    - Draw topic proportion $\theta_d \sim \mathrm{Dir}(a);$
    - For each word n in document d,
      - Draw topic assignment $z_{dn} \sim \mathrm{Mult}(\theta_d);$
      - Draw word $w_{dn} \sim \mathrm{Mult}(z_{dn})$.
    - Draw the response $y_d \sim \mathcal{N}(\eta^\top \bar{z}_d, \sigma^2), \bar{z}_d \coloneqq \frac{1}{N_d}\sum_{n=1}^{N_d} z_{dn}$ (one-hot).

$$p(\beta, \theta, z, w, \mathrm{y})$$

$$= \left(\prod_{k=1}^{K} \mathrm{Dir}(\beta_k|b)\right)\left(\prod_{d=1}^{D} \mathrm{Dir}\left(\prod_{n=1}^{N_d} \mathrm{Mult}(z_{dn}|\theta_d)\mathrm{Mult}\left(w_{dn}\middle|\beta_{z_{dn}}\right)\right)\mathcal{N}(y_d|\eta^\top \bar{z}_d, \sigma^2)\right).$$

Supervised Latent Dirichlet Allocation

Variational inference [MB08]: similar to LDA.

- Same variational distribution

$$q_{\lambda,\gamma,\phi}(\beta,\theta,z) \coloneqq \prod_{k=1}^{K} \mathrm{Dir}(\beta_k|\lambda_k) \prod_{d=1}^{D} \mathrm{Dir}(\theta_d|\gamma_d) \prod_{n=1}^{n_d} \mathrm{Mult}(z_{dn}|\phi_{dn}) .$$

  ELBO$(\lambda, \gamma, \phi; a, b, \eta, \sigma^2)$ is available in closed form.

- E-step: update λ, γ, ϕ by maximizing ELBO.
- M-step: update a, b, η, σ^2 by maximizing ELBO.
- Prediction: given a new document w_d,

$$\hat{y}_d \coloneqq \mathbb{E}_{p(y_d|w_d)}[y_d] = \eta^\top \mathbb{E}_{p(z_d|w_d)}[\bar{z}_d] \approx \eta^\top \mathbb{E}_{q(z_d|w_d)}[\bar{z}_d].$$

  First do inference (find $q(z_d|w_d)$ i.e. ϕ_d), then estimate $\hat{y}_d$.

Supervised Latent Dirichlet Allocation

Variational inference with posterior regularization [ZAX12]

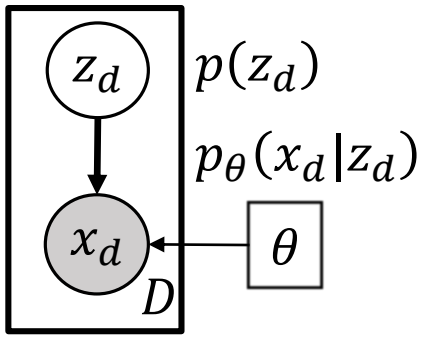
- Regularized Bayesian Posterior (RegBayes) [ZCX14]:
  - Recall: $p\left(z\middle|\left\{x^{(n)},y^{(n)}\right\}\right)$
    $= \arg\min_{q(z)}\left\{-\mathcal{L}[q] = \mathrm{KL}\left(q(z),p(z)\right) - \sum_n \mathbb{E}_q\left[\log p\left(x^{(n)},y^{(n)}|z\right)\right]\right\}.$
  - Regularize posterior towards better prediction:
    $\min_{q(z)} \mathrm{KL}\left(q(z),p(z)\right) - \sum_n \mathbb{E}_q\left[\log p\left(x^{(n)},y^{(n)}|z\right)\right] + \lambda\ell\left(q(z);\left\{x^{(n)},y^{(n)}\right\}\right).$
- Maximum entropy discrimination LDA (MedLDA) [ZAX12]:
  - $\ell\left(q;\left\{w^{(n)},y^{(n)}\right\}\right) = \sum_n \ell_\varepsilon\left(y^{(n)} - \hat{y}^{(n)}\left(q,w^{(n)}\right)\right)$
    $= \sum_n \ell_\varepsilon\left(y^{(n)} - \eta^\top \mathbb{E}_{q\left(z^{(n)}\middle|w^{(n)}\right)}\left[\bar{z}^{(n)}\right]\right),$
    where $\ell_\varepsilon(r) = \max\{0,|r|-\varepsilon\}$ is the hinge (max-margin) loss.
  - Facilitates both prediction and topic representation.

Bayesian Models

Bayesian Networks

- Variational Auto-Encoders

Variational Auto-Encoder



More *flexible* Bayesian model using *deep learning* tools.

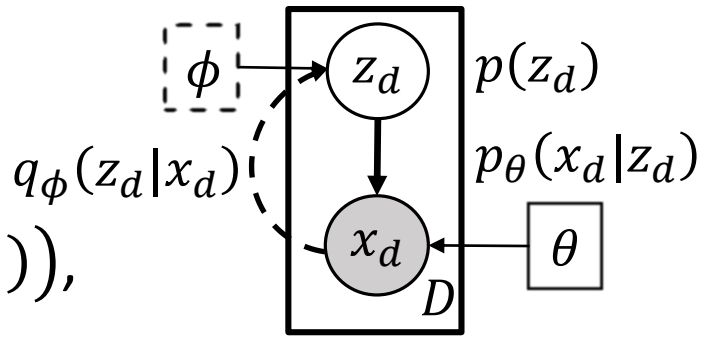
- Model structure (decoder) [KW14]:

  $z_d \sim p(z_d) = \mathcal{N}(z_d|0, I)$,

  $x_d \sim p_\theta(x_d|z_d) = \mathcal{N}\left(x_d|\mu_\theta(z_d), \Sigma_\theta(z_d)\right)$,

  where $\mu_\theta(z_d)$ and $\Sigma_\theta(z_d)$ are modeled by neural networks.

Variational Auto-Encoder



- Variational inference (encoder) [KW14]:

  $q_\phi(z|x) \coloneqq \prod_{d=1}^{D} q_\phi(z_d|x_d) = \prod_{d=1}^{D} \mathcal{N}\left(z_d \middle| \nu_\phi(x_d), \Gamma_\phi(x_d)\right)$,

  where $\nu_\phi(x_d), \Gamma_\phi(x_d)$ are also NNs.

  - Amortized inference: approximate local posteriors $\{p(z_d|x_d)\}_{d=1}^{D}$ globally by ϕ.
  - Objective:

$$\mathbb{E}_{\hat{p}(x)}[\mathrm{ELBO}(x)] \approx \frac{1}{D}\sum_{d=1}^{D} \mathbb{E}_{q_\phi(z_d|x_d)}\left[\log p_\theta(z_d)p_\theta(x_d|z_d) - \log q_\phi(z_d|x_d)\right].$$

  - Gradient estimation with the reparameterization trick:

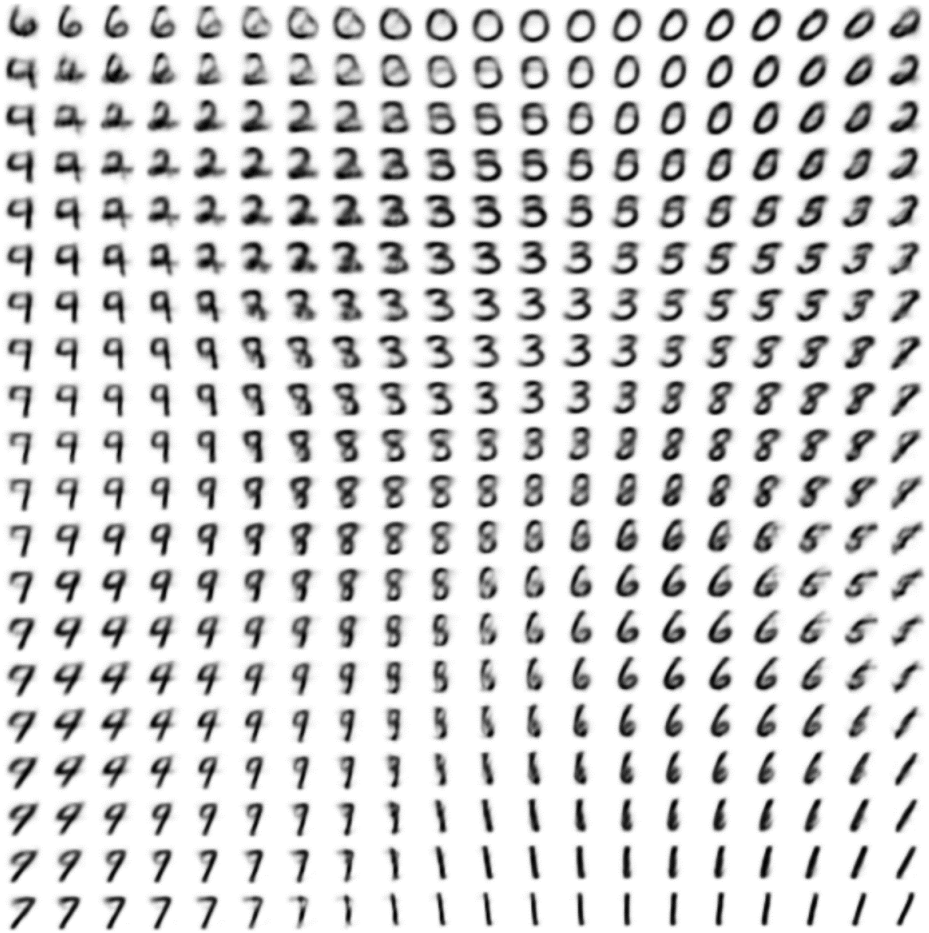
    $z_d \sim q_\phi(z_d|x_d) \Leftrightarrow z_d = g_\phi(x_d, \epsilon) \coloneqq \nu_\phi(x_d) + \epsilon\sqrt{\Gamma_\phi(x_d)}, \epsilon \sim q(\epsilon) = \mathcal{N}(\epsilon|0, I).$

    Smaller variance than REINFORCE-like estimations.

$$\nabla_{\phi,\theta}\mathbb{E}_{\hat{p}(x)}[\mathrm{ELBO}(x)] \approx \frac{1}{D}\sum_{d=1}^{D} \mathbb{E}_{q(\epsilon)}\left[\nabla_{\phi,\theta}\left(\log p_\theta(z_d)p_\theta(x_d|z_d) - \log q_\phi(z_d|x_d)\,|_{z_d=g_\phi(x_d,\epsilon)}\right)\right].$$

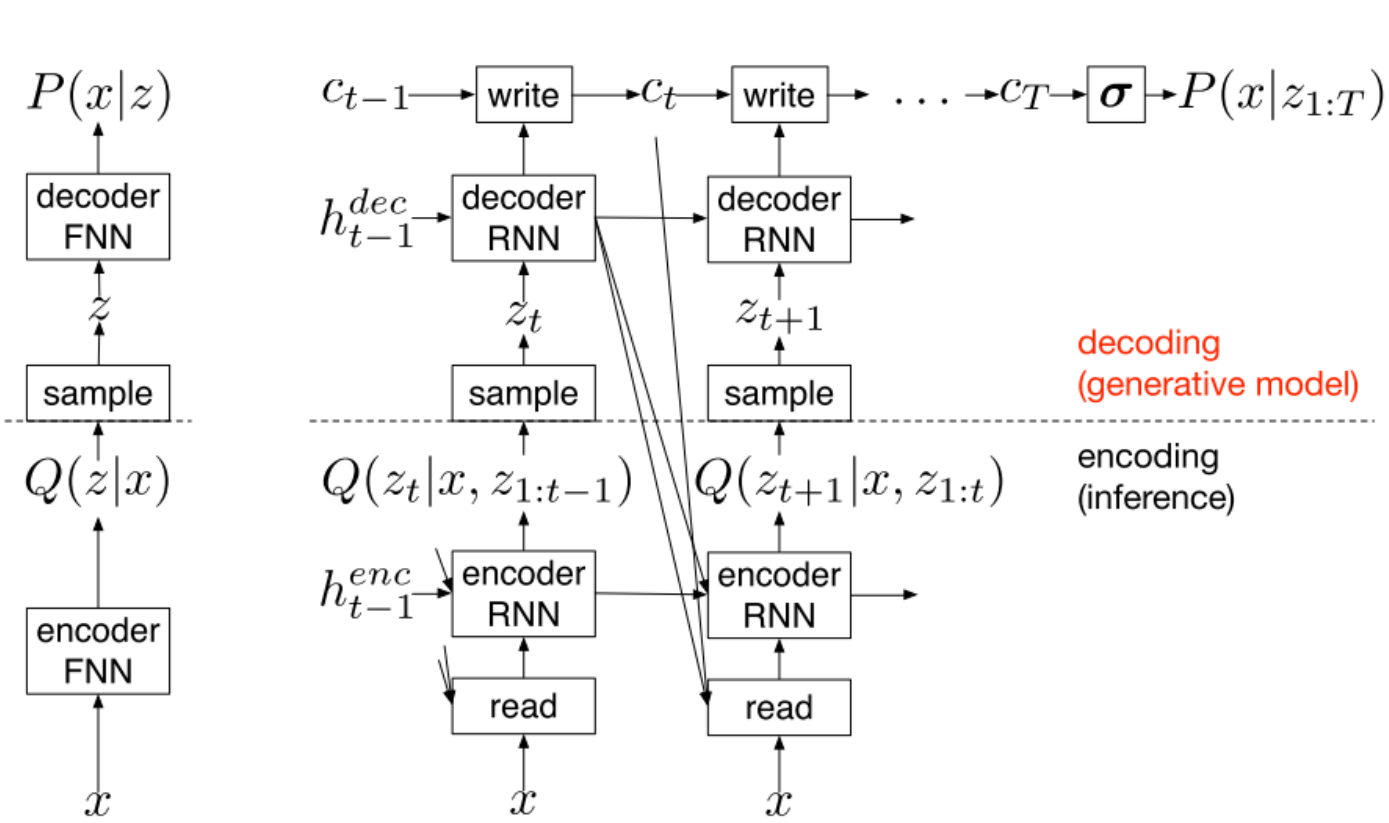
Variational Auto-Encoder

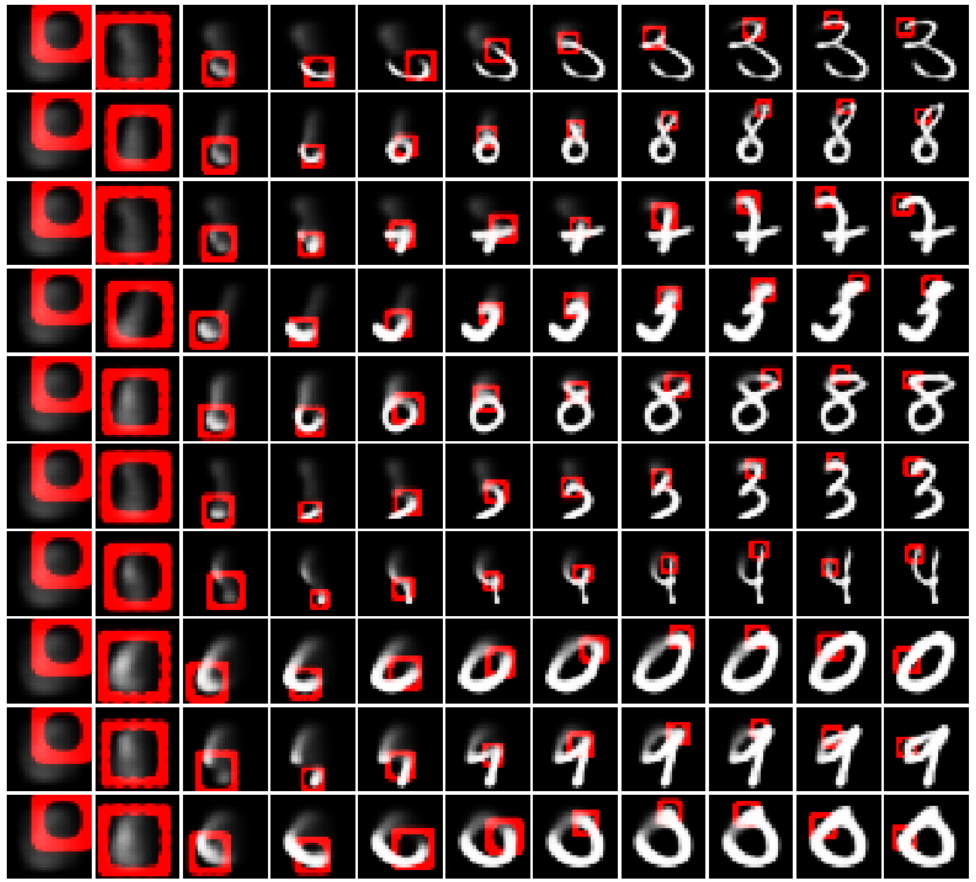
- Generation results [KW14]



Variational Auto-Encoder

- With spatial attention structure [GDG+15]





Variational Auto-Encoder

- Inference with importance-weighted ELBO [BGS15]
  - ELBO: $\mathcal{L}_\theta[q_\phi(z)] = \mathbb{E}_{q_\phi(z)}[\log p_\theta(z,x)] - \mathbb{E}_{q_\phi(z)}[\log q_\phi(z)]$.
  - A tighter lower bound:

$$\mathcal{L}_\theta^{(k)}[q_\phi] \coloneqq \mathbb{E}_{z^{(1)},\ldots,z^{(k)} \sim \text{i.i.d.} q_\phi}\left[\log \frac{1}{k}\sum_{i=1}^{k} \frac{p_\theta(z^{(k)},x)}{q_\phi(z^{(k)})}\right].$$

Ordering relation:

$$\mathcal{L}_\theta[q_\phi] = \mathcal{L}_\theta^{(1)}[q_\phi] \le \mathcal{L}_\theta^{(2)}[q_\phi] \le \cdots \le \mathcal{L}_\theta^{(\infty)}[q_\phi] = \log p_\theta(x).$$

If $\frac{p(z,x)}{q(z|x)}$ is bounded.

Variational Auto-Encoder

- Parametric Variational Inference: towards more flexible approximations.
  - Explicit VI:

    Normalizing flows [RM15, KSJ+16]

    (concatenation of a sequence of invertible mappings).
  - Implicit VI:

    Adversarial Auto-Encoder [MSJ+15], Adversarial Variational Bayes [MNG17], Wasserstein Auto-Encoder [TBGS17], [SSZ18a], [LT18], [SSZ18b].

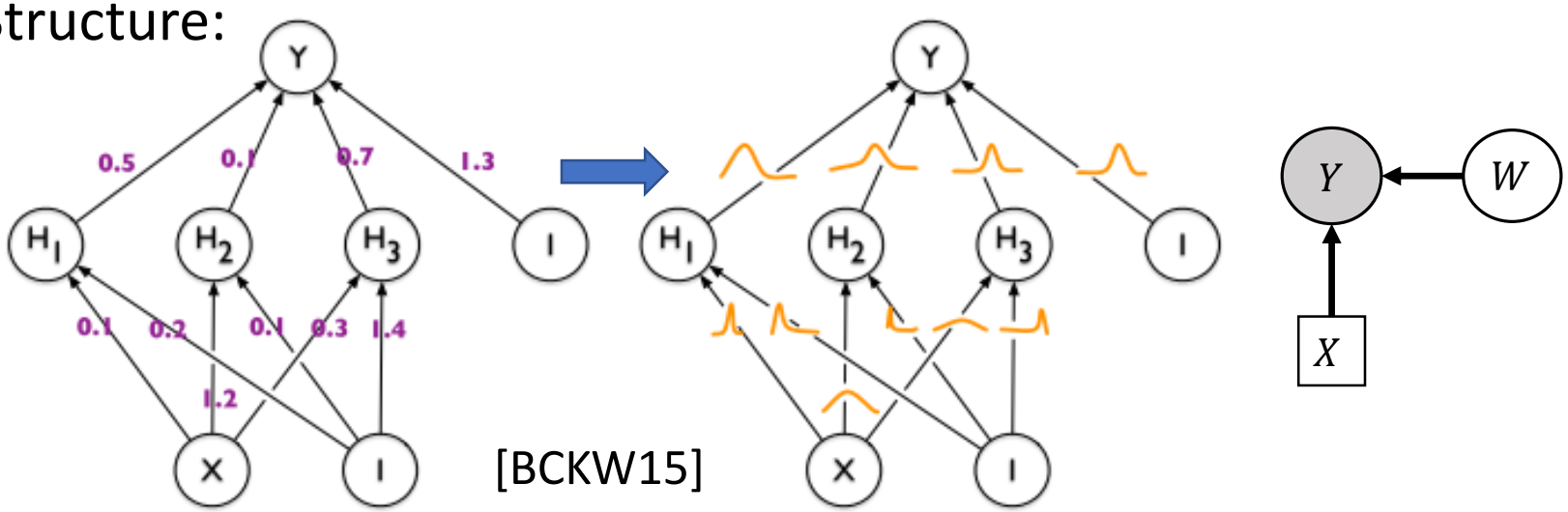
- MCMC [LTL17] and Particle-Based VI [FWL17, PGH+17]:
  - Train the encoder as a sample generator.
  - Amortize the update on samples to ϕ.

Bayesian Models

Bayesian Neural Networks

Bayesian Neural Networks

Model Structure:



[BCKW15]

- Latent variable: W, the parameter (weights) of an NN.
- Prior: $p(W)$ (*e.g.*, diagonal Gaussian).
- Likelihood: $p(Y|W,X) = \prod_{d=1}^{D} p(Y_d|W,X_d)$, $p(Y_d|W,X_d)$ is specified by the NN.
- Equiv. to Gaussian process for infinite width [Nea95].
- Benefits: avoid overfitting (similar to dropout [SHK+14]); robust to adversarial attack.

Bayesian Neural Networks

Variational Inference: parametric, explicit

Bayes by Backprop [BCKW15]

- Take $q_\phi(W)$ s.t. $w \sim q_\phi(W) \Leftrightarrow W = f_\phi(\epsilon), \epsilon \sim q(\epsilon)$.

  The form of $q_\phi(W)$ is required.
  - Example: $W \sim \mathcal{N}\left(\phi_1, e^{2\phi_2}\right) \Leftrightarrow W = \phi_1 + e^{\phi_2}\epsilon, \epsilon \sim \mathcal{N}(0, I)$.
- $\mathrm{ELBO}(\phi) = \mathbb{E}_{q_\phi(W)}[L(W, \phi)] = \mathbb{E}_{q(\epsilon)}\left[L\left(f_\phi(\epsilon), \phi\right)\right]$,
  where $L(W, \phi) = \log p(W) + \sum_{d=1}^{D} \log p(Y_d | W, X_d) - \log q_\phi(W)$ is available in closed form.

$$\nabla_\phi \mathcal{L}(\phi) = \mathbb{E}_{q(\epsilon)}\left[\nabla_W L \cdot \nabla_\phi f_\phi(\epsilon) + \nabla_\phi L\right].$$

Bayesian Neural Networks

Variational Inference: parametric, implicit

- [MNG17, SSZ18a, LT18, SSZ18b, ...]
- Using EP: [HA15]

Variational Inference: particle-based

- All required is $\nabla_W \log p(W|Y,X) = \nabla_W \log p(W) + \sum_{d=1}^{D} \nabla_W \log p(Y_d|W,X_d)$.
- [LW16, CZW+18, LZC+19, LZZ19, ...]

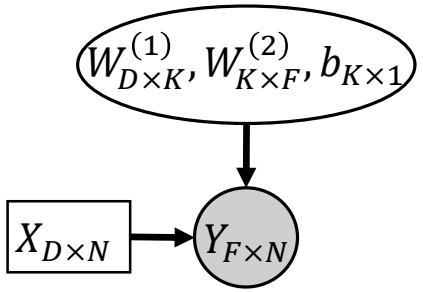
MCMC:

- For dynamics-based MCMC, all required is $\nabla_W \log p(W|Y,X) = \nabla_W \log p(W) + \sum_{d=1}^{D} \nabla_W \log p(Y_d|W,X_d)$.
- [Nea11, CFG14, DFB+14, ...]

Bayesian Neural Networks

- Dropout as VI for BNN [GG15]

BNN model:



$$p\left(W^{(1)}\right)=\mathcal{N}(0,I), p\left(W^{(2)}\right)=\mathcal{N}(0,I), p(b)=\mathcal{N}(0,I),$$
$$p\left(Y\middle|W^{(1)},W^{(2)},b,X\right)=\mathcal{N}\left(\frac{1}{\sqrt{K}}W^{(2)^{\top}}\sigma\left(W^{(1)^{\top}}X+b\right),\frac{1}{\tau}I\right).$$

Variational Distribution:

$$q\left(W^{(1)},W^{(2)},b\right)=q(b)\prod_{d=1}^{D}q\left(W_d^{(1)}\right)\prod_{k=1}^{K}q\left(W_k^{(2)}\right),$$

- $b=m+\delta\varepsilon^{(3)}, \varepsilon^{(3)}\sim\mathcal{N}(0,I_K),$
- $W_d^{(1)}=z_d^{(1)}\odot\left(M_d^{(1)}+\delta\varepsilon_d^{(1)}\right)+\left(1-z_d^{(1)}\right)\odot\delta\varepsilon_d^{(1)},$
  $\varepsilon_d^{(1)}\sim\mathcal{N}(0,I_K), z_d^{(1)}\sim\left[\mathrm{Bern}\left(p^{(1)}\right)\right]_{1\times K},$
- $W_k^{(2)}=z_k^{(2)}\odot\left(M_k^{(2)}+\delta\varepsilon_k^{(2)}\right)+\left(1-z_k^{(2)}\right)\odot\delta\varepsilon_k^{(2)},$
  $\varepsilon_k^{(2)}\sim\mathcal{N}(0,I_F), z_k^{(2)}\sim\left[\mathrm{Bern}\left(p^{(2)}\right)\right]_{1\times F}.$

$$\mathrm{ELBO}\left(M^{(1)},M^{(2)},m\right)=\mathbb{E}_q\left[\log p\left(Y\middle|W^{(1)},W^{(2)},b,X\right)\right]-\mathrm{KL}\left(q,p\left(W^{(1)}\right)p\left(W^{(2)}\right)p(b)\right)$$

$$=-\frac{\tau}{2}\sum_{n=1}^{N}\left\|Y_n-\hat{Y}_n\right\|_2^2-\frac{p^{(1)}}{2}\left\|M^{(1)}\right\|_F^2-\frac{p^{(2)}}{2}\left\|M^{(2)}\right\|_F^2-\frac{1}{2}\|m\|_2^2+C(\tau,\sigma),$$

$K\gg 1, \delta\to 0$

Obj. for training with dropout!

$$\hat{Y}_n=\frac{1}{\sqrt{K}}\widehat{W}^{(2)^{\top}}\sigma\left(\widehat{W}^{(1)^{\top}}X_n+\hat{b}\right)\xrightarrow{\delta\to 0}\frac{1}{\sqrt{K}}\left(\hat{z}^{(2)}\odot M^{(2)}\right)^{\top}\sigma\left(\left(\hat{z}^{(1)}\odot M^{(1)}\right)^{\top}X_n+m\right),$$

where $\widehat{W}^{(1)}, \widehat{W}^{(2)}, \hat{b}\sim q, \hat{z}\sim\mathrm{Bern}.$

Bayesian Models

Markov Random Fields

Markov Random Fields

Specify the joint distribution $p_\theta(x, z)$ by an energy function $E_\theta(x, z)$.

$$p_\theta(x, z) = \frac{1}{Z_\theta} \exp\big(-E_\theta(x, z)\big), Z_\theta = \int \exp\big(-E_\theta(x', z')\big)\ \mathrm{d}x'\mathrm{d}z' .$$

- Only relationship and no causality: $p(x, z)$ is either $p(z)p(x|z)$ or $p(x)p(z|x)$.
  - Simpler modeling, yet still flexible.
  - Learning is harder.
- Synonyms: Energy-Based Model, Undirected Graphical Model.
- Etymology: Statistical Mechanics.

  The distribution over the microstate s of a canonical ensemble (system with fixed temperature T, volume and particle number) is called the Boltzmann distribution:

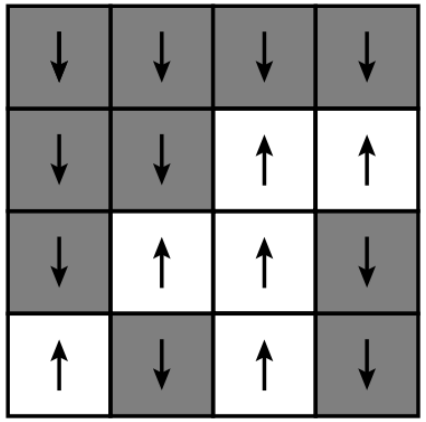
$$p(s) \propto \exp\left(-\frac{1}{k_B T} H(s)\right),$$

  where $H(s)$ is the Hamiltonian of the system, which almost always represents the energy of the system. [Set06]

Markov Random Fields

- Gallery (alternative symbols: $v \Leftrightarrow x$, $h \Leftrightarrow z$, $s \Leftrightarrow x$ or z)

Ising model [Isi25]

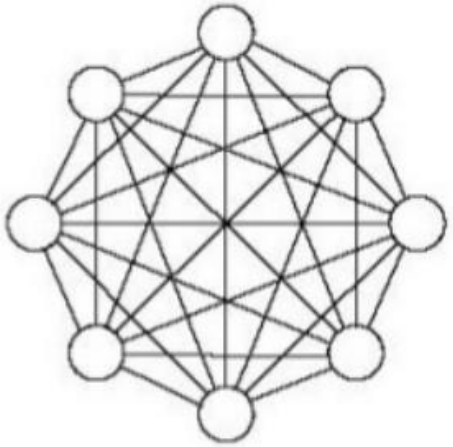


$$\mathcal{H} = -\sum_{\langle ij \rangle} J s_i s_j - H \sum_i s_i.$$

[Set06]

Sum over nearest neighbors

Hopfield Network [Hop82]



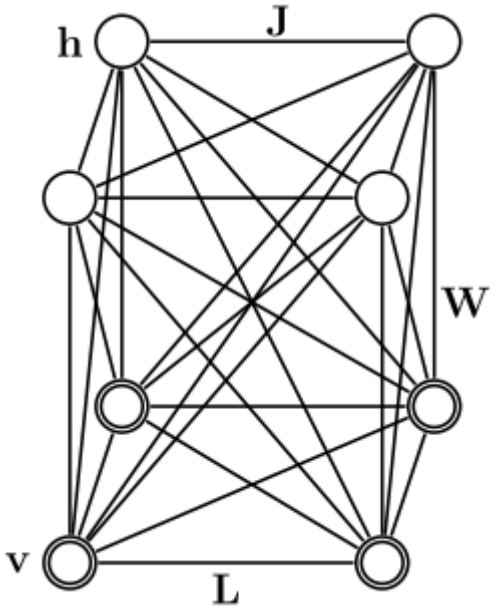
$$E = -\frac{1}{2}\sum_{i,j} w_{ij} s_i s_j + \sum_i \theta_i s_i$$

(Slides by Kharagorgiev)

Markov Random Fields

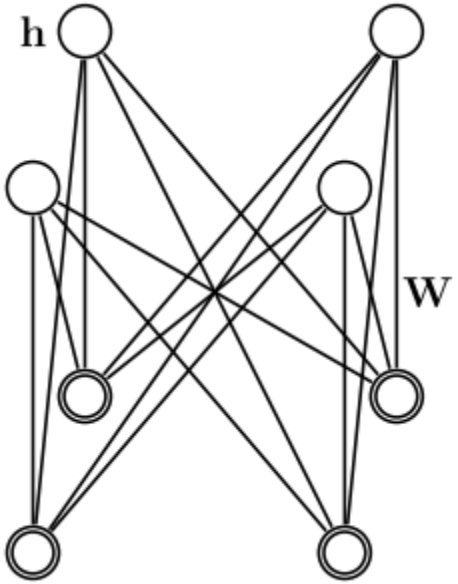
- Gallery (alternative symbols: $v \Leftrightarrow x$, $h \Leftrightarrow z$, $s \Leftrightarrow x$ or z)

Boltzmann Machine [HS83]



$$E_\theta(v,h) = -\frac{1}{2}v^\top Lv - \frac{1}{2}h^\top Jh - v^\top Wh$$

Restricted Boltzmann Machine (*Harmonium*) [Smo86]



$$E_\theta(v,h) = -v^\top Wh$$

(may has a bias term $b^{(v)^\top}v + b^{(h)^\top}h$)

Products of Experts [Hin02]

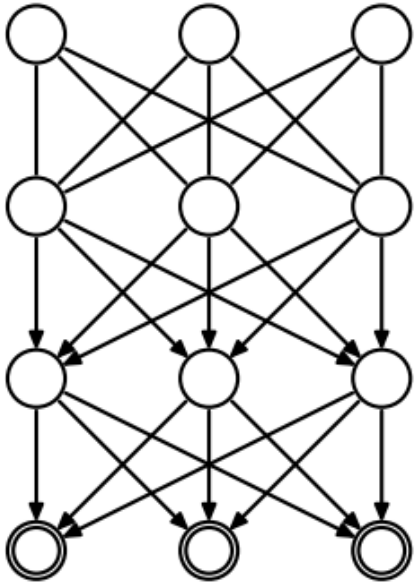
$$p(\mathbf{d} \mid \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_n) = \frac{\prod_m f_m(\mathbf{d} \mid \boldsymbol{\theta}_m)}{\sum_{\mathbf{c}} \prod_m f_m(\mathbf{c} \mid \boldsymbol{\theta}_m)},$$

Each $f_m(d|\theta_m)$ is an expert.

Markov Random Fields

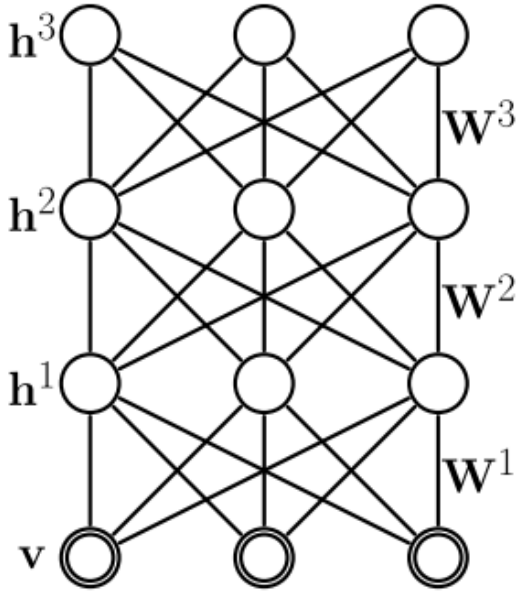
- Gallery (alternative symbols: $v \Leftrightarrow x$, $h \Leftrightarrow z$, $s \Leftrightarrow x$ or z)

Deep Belief Network [HOT06]
(hybrid of directed and undirected)



$$p\left(v, h^{(1)}, \dots, h^{(L)}\right)$$
$$= p\left(v \middle| h^{(1)}\right) p\left(h^{(1)} \middle| h^{(2)}\right)$$
$$\dots p\left(h^{(L-2)} \middle| h^{(L-1)}\right) p\left(h^{(L-1)}, h^{(L)}\right)$$

Deep Boltzmann Machine [SH09]



$$E_\theta\left(v, h^{(1)}, \dots, h^{(L)}\right)$$
$$= E_{W^{(1)}}\left(v, h^{(1)}\right) + \sum_{l=2}^{L} E_{W^{(l)}}\left(h^{(l-1)}, h^{(l)}\right)$$

Markov Random Fields

Specify the joint distribution $p_\theta(x, z)$ by an energy function $E_\theta(x, z)$.

$$p_\theta(x, z) = \frac{1}{Z_\theta} \exp\big(-E_\theta(x, z)\big), Z_\theta = \int \exp\big(-E_\theta(x', z')\big)\ \mathrm{d}x'\mathrm{d}z' .$$

- Bayesian Inference: generally same as BayesNets.
  - Some MRFs can do this *exactly*, *e.g.*, RBM:

$$p(\boldsymbol{x}, \boldsymbol{h}; \boldsymbol{\theta}) = \frac{\exp(\boldsymbol{x}^T \mathbf{W} \boldsymbol{h} + \boldsymbol{x}^T \boldsymbol{b}^{(1)} + \boldsymbol{h}^T \boldsymbol{b}^{(2)})}{\mathcal{Z}(\boldsymbol{\theta})}$$

$$p(x_k|\boldsymbol{h}; \boldsymbol{\theta}) = \mathrm{Bern}(\sigma(\mathbf{W}_{k:}\boldsymbol{h} + b_k^{(1)}))$$
$$p(h_k|\boldsymbol{x}; \boldsymbol{\theta}) = \mathrm{Bern}(\sigma(\boldsymbol{x}^T \mathbf{W}_{:k} + b_k^{(2)}))$$

Markov Random Fields

Specify the joint distribution $p_\theta(x, z)$ by an energy function $E_\theta(x, z)$.

$$p_\theta(x, z) = \frac{1}{Z_\theta} \exp\big(-E_\theta(x, z)\big), Z_\theta = \int \exp\big(-E_\theta(x', z')\big)\, \mathrm{d}x' \mathrm{d}z'.$$

- Learning: MLE, i.e., $\max_\theta \mathbb{E}_{\hat{p}(x)}[\log p_\theta(x)]$.

  Harder than BayesNets! Even $p_\theta(x, z)$ is unavailable.

  $\nabla_\theta \mathbb{E}_{\hat{p}(x)}[\log p_\theta(x)] = -\mathbb{E}_{\hat{p}(x)p_\theta(z|x)}[\nabla_\theta E_\theta(x, z)] + \mathbb{E}_{p_\theta(x,z)}[\nabla_\theta E_\theta(x, z)]$.

  =0 if $E = \log p$.

  (augmented) data distribution

  model distribution

$$\nabla_\theta \mathbb{E}_{\hat{p}(x)}[\log p_\theta(x)] = -\mathbb{E}_{\hat{p}(x)p_\theta(z|x)}[\nabla_\theta E_\theta(x, z)] + \mathbb{E}_{p_\theta(x,z)}[\nabla_\theta E_\theta(x, z)].$$

Bayesian Inference

generation

- Generation: harder than BayesNets! Rely on MCMC or train a generator.

Markov Random Fields

- Learning: harder than BayesNets! Even $p_\theta(x, z)$ is unavailable.

$$\nabla_\theta \mathbb{E}_{\hat{p}(x)}[\log p_\theta(x)] = -\mathbb{E}_{\hat{p}(x)p_\theta(z|x)}[\nabla_\theta E_\theta(x, z)] + \mathbb{E}_{p_\theta(x,z)}[\nabla_\theta E_\theta(x, z)].$$

Bayesian Inference

Generation

  - [HS83]: use Gibbs sampling for both inference and generation (for Boltzmann machine).

$$E(\mathbf{v}, \mathbf{h}; \theta) = -\frac{1}{2}\mathbf{v}^\top \mathbf{L}\mathbf{v} - \frac{1}{2}\mathbf{h}^\top \mathbf{J}\mathbf{h} - \mathbf{v}^\top \mathbf{W}\mathbf{h},$$

$$p(h_j = 1|\mathbf{v}, \mathbf{h}_{-j}) = \sigma\Big(\sum_{i=1}^{D} W_{ij} v_i + \sum_{m=1\backslash j}^{P} J_{jm} h_j\Big),$$

$$p(v_i = 1|\mathbf{h}, \mathbf{v}_{-i}) = \sigma\Big(\sum_{j=1}^{P} W_{ij} h_j + \sum_{k=1\backslash i}^{D} L_{ik} v_j\Big),$$

$$\begin{aligned}
\Delta\mathbf{W} &= \alpha\left(\mathrm{E}_{P_{\text{data}}}[\mathbf{v}\mathbf{h}^\top] - \mathrm{E}_{P_{\text{model}}}[\mathbf{v}\mathbf{h}^\top]\right), \\
\Delta\mathbf{L} &= \alpha\left(\mathrm{E}_{P_{\text{data}}}[\mathbf{v}\mathbf{v}^\top] - \mathrm{E}_{P_{\text{model}}}[\mathbf{v}\mathbf{v}^\top]\right), \\
\Delta\mathbf{J} &= \alpha\left(\mathrm{E}_{P_{\text{data}}}[\mathbf{h}\mathbf{h}^\top] - \mathrm{E}_{P_{\text{model}}}[\mathbf{h}\mathbf{h}^\top]\right),
\end{aligned}$$

(Clipped from [SH09])

Markov Random Fields

- Learning: harder than BayesNets! Even $p_\theta(x, z)$ is unavailable.

$$\nabla_\theta \mathbb{E}_{\hat{p}(x)}[\log p_\theta(x)] = -\mathbb{E}_{\hat{p}(x)p_\theta(z|x)}[\nabla_\theta E_\theta(x, z)] + \mathbb{E}_{p_\theta(x,z)}[\nabla_\theta E_\theta(x, z)].$$

Bayesian Inference — Generation

  - [Hin02]: estimation with k-step MCMC approximates the gradient of k-step *Contrastive Divergence* (CD-k).

    (for Products of Experts (no lat. var.))

$$\mathrm{CD}_k := \mathrm{KL}(P^0||P_\theta^\infty) - \mathrm{KL}(P_\theta^k||P_\theta^\infty),$$
$$P^0(x) = \hat{p}(x), P_\theta^k(x) := P^0(x)P_\theta\left(x^{(k)}\middle|x\right).$$

k-step transition of MCMC from data to model.

$$-\frac{\partial}{\partial \boldsymbol{\theta}_m}(P^0 \parallel P_\theta^\infty - P_\theta^1 \parallel P_\theta^\infty) = \left\langle \frac{\partial \log f_{\theta_m}}{\partial \boldsymbol{\theta}_m} \right\rangle_{P^0} - \left\langle \frac{\partial \log f_{\theta_m}}{\partial \boldsymbol{\theta}_m} \right\rangle_{P_\theta^1}$$
$$+ \frac{\partial P_\theta^1}{\partial \boldsymbol{\theta}_m} \frac{\partial (P_\theta^1 \parallel P_\theta^\infty)}{\partial P_\theta^1}.$$

Markov Random Fields

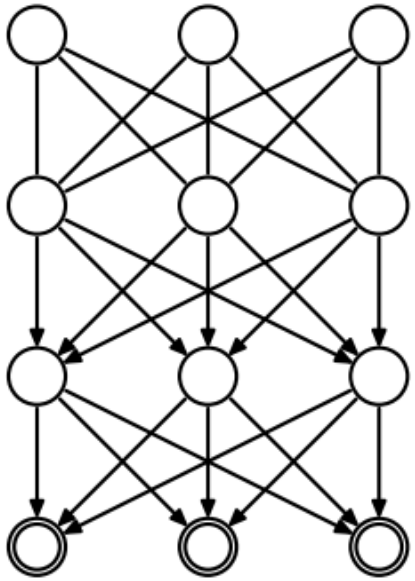
- Learning: harder than BayesNets! Even $p_\theta(x, z)$ is unavailable.

$$\nabla_\theta \mathbb{E}_{\hat{p}(x)}[\log p_\theta(x)] = -\mathbb{E}_{\hat{p}(x)p_\theta(z|x)}[\nabla_\theta E_\theta(x, z)] + \mathbb{E}_{p_\theta(x,z)}[\nabla_\theta E_\theta(x, z)].$$

Bayesian Inference

Generation

  - [HOT06]: training a stack of RBMs layer-wise corresponds to a DBN.



Algorithm 1 (Recursive greedy learning procedure for the deep belief network):

1: Fit the parameters $W^{(1)}$ of the first-layer RBM to data.
2: Fix the parameter vector $W^{(1)}$, and use samples $\mathbf{h}^{(1)}$ from $Q(\mathbf{h}^{(1)}|\mathbf{v}) = P(\mathbf{h}^{(1)}|\mathbf{v}, W^{(1)})$ as the data for training the next layer of binary features with an RBM.
3: Fix the parameters $W^{(2)}$ that define the second layer of features, and use the samples $\mathbf{h}^{(2)}$ from $Q(\mathbf{h}^{(2)}|\mathbf{h}^{(1)}) = P(\mathbf{h}^{(2)}|\mathbf{h}^{(1)}, W^{(2)})$ as the data for training the third layer of binary features.
4: Proceed recursively for the next layers.

[Sal15]

Markov Random Fields

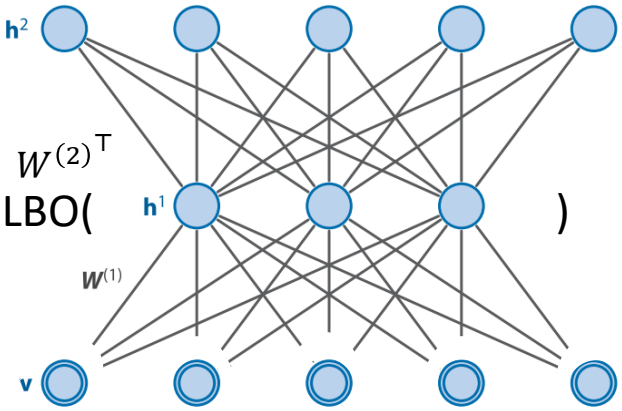
- Learning: harder than BayesNets! Even $p_\theta(x, z)$ is unavailable.
  - [HOT06]: training a stack of RBMs layer-wise corresponds to a DBN.
    - Interpretation: after training the first RBM, training the second RBM is to:

$$\max_{W^{(2)}} \mathbb{E}_{p_{\mathrm{RBM},W^{(1)}}(h^{(1)}|v)} \left[\log p_{\mathrm{RBM},W^{(2)}}(h^{(1)})\right],$$

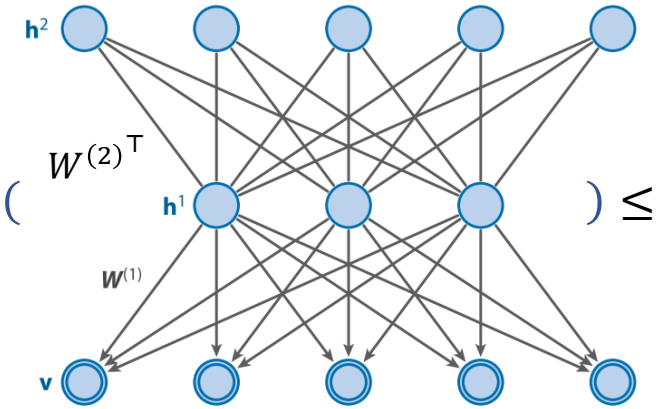
which is equivalent to

$$\max_{W^{(2)}} \mathbb{E}_{q(h^{(1)}|v)} \left[\log p_{\mathrm{RBM},W^{(2)}}(h^{(1)}) + \log p_{\mathrm{SBN},W^{(1)}}(v|h^{(1)})\right] + \mathbb{H}\left[q(h^{(1)}|v)\right].$$

$p_{\mathrm{RBM},W^{(1)}}(h^{(1)}|v)$

If it were a DBM, ELBO(

)

$= \mathrm{ELBO}($ 

$) \leq \log p_{\mathrm{b}}(v).$

$$= \mathbb{E}_{q(h^{(1)}|v)} \left[\log p_{\mathrm{RBM},W^{(2)}}(h^{(1)}) + \log p_{\mathrm{SBN},W^{(1)}}(v|h^{(1)})\right] + \mathbb{H}\left[q(h^{(1)}|v)\right]$$
$$+ \mathbb{E}_{q(h^{(1)}|v)} \left[\log p_{\mathrm{RBM},W^{(1)}}(h^{(1)})\right] + \log Z_{W^{(2)}} - \log Z_{W^{(1)},W^{(2)}}.$$

Markov Random Fields

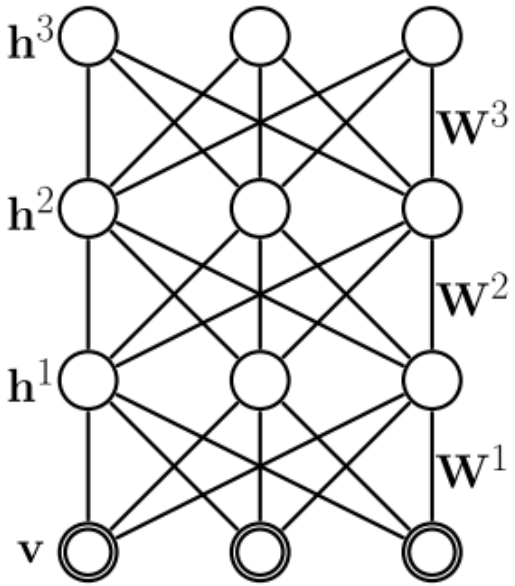
- Learning: harder than BayesNets! Even $p_\theta(x, z)$ is unavailable.

$$\nabla_\theta \mathbb{E}_{\hat{p}(x)}[\log p_\theta(x)] = -\mathbb{E}_{\hat{p}(x)p_\theta(z|x)}[\nabla_\theta E_\theta(x, z)] + \mathbb{E}_{p_\theta(x,z)}[\nabla_\theta E_\theta(x, z)].$$

Bayesian Inference

Generation

  - [SH09]: Deep Boltzmann Machine.



- Initialize the weights by running DBN in both directions.
- Fine-tune with standard training methods of a Boltzmann machine.

Markov Random Fields

Deep-Nets-Based EBMs:

No latent variable; $E_\theta(x)$ is modeled by a neural network.

$$\nabla_\theta \mathbb{E}_{\hat{p}(x)}[\log p_\theta(x)] = -\mathbb{E}_{\hat{p}(x)}[\nabla_\theta E_\theta(x)] + \mathbb{E}_{p_\theta(x')}[\nabla_\theta E_\theta(x')].$$

- [KB16]: learn a generator

$$x \sim q_\phi(x) \Leftrightarrow z \sim q(z), x = g_\phi(z),$$

to mimic the generation from $p_\theta(x)$:

$$\arg\min_\phi \mathrm{KL}(q_\phi, p_\theta) = \arg\min_\phi \mathbb{E}_{q(z)}\left[E_\theta\left(g_\phi(z)\right)\right] - \underbrace{\mathbb{H}[q_\phi]}_{\text{approx. by batch normalization Gaussian}}$$

Markov Random Fields

Deep-Nets-Based EBMs:

No latent variable; $E_\theta(x)$ is modeled by a neural network.

$$\nabla_\theta \mathbb{E}_{\hat{p}(x)}[\log p_\theta(x)] = -\mathbb{E}_{\hat{p}(x)}[\nabla_\theta E_\theta(x)] + \mathbb{E}_{p_\theta(x')}[\nabla_\theta E_\theta(x')].$$

- [ZML16]: interpret the discriminator $D: \mathcal{X} \to \mathbb{R}^+$ as an energy function.

$$\min_D \mathbb{E}_{\hat{p}(x)p(z)} \quad \mathcal{L}_D(x,z) = D(x) + [m - D(G(z))]^+$$
$$\min_G \mathbb{E}_{p(z)} \quad \mathcal{L}_G(z) = D(G(z))$$

  - Theoretical guarantee: for Nash equilibrium state, $p_{G^*}(x) = \hat{p}(x)$ a.e.
  - Interpretation as EBM:
    - Obj. of $D = \int_x \left(p_{data}(x)D(x) + p_{G^*}(x)\,[m - D(x)]^+ \right) \mathrm{d}x.$ Similar gradient to $\mathbb{E}_{\hat{p}}[\log p_\theta]$.
    - Obj. of $G = \mathbb{E}_{p_G}[D] = -\mathrm{KL}(p_G, p_\theta) - \mathbb{H}[p_G]$.

Markov Random Fields

Deep-Nets-Based EBMs:

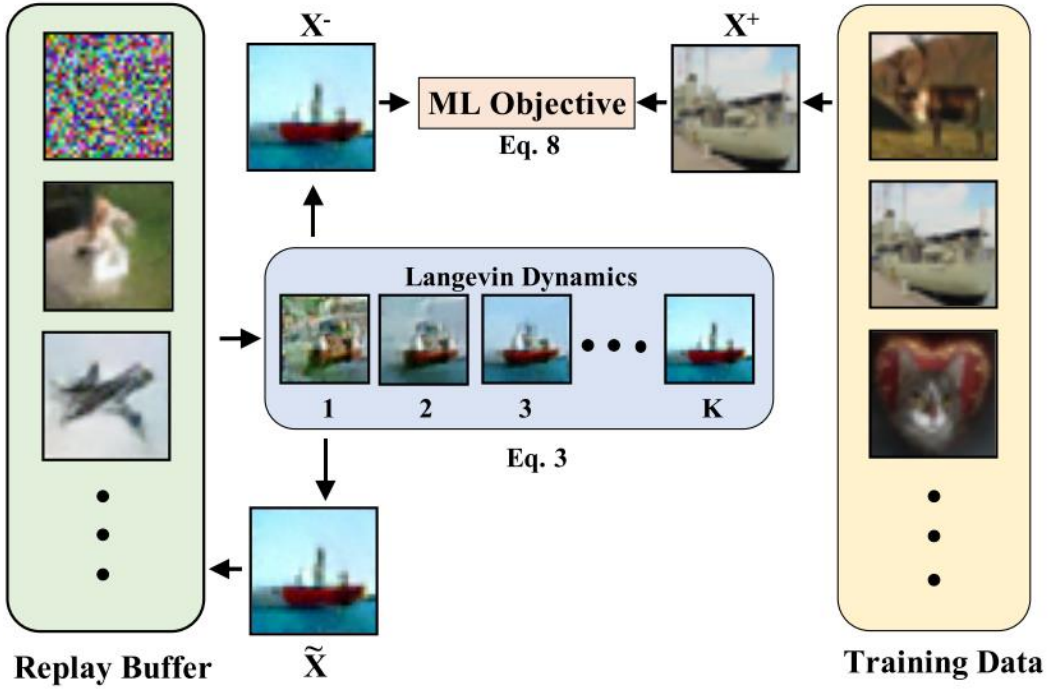
No latent variable; $E_\theta(x)$ is modeled by a neural network.

$$\nabla_\theta \mathbb{E}_{\hat{p}(x)}[\log p_\theta(x)] = -\mathbb{E}_{\hat{p}(x)}[\nabla_\theta E_\theta(x)] + \mathbb{E}_{p_\theta(x')}[\nabla_\theta E_\theta(x')].$$

- [DM19]: estimate $\mathbb{E}_{p_\theta}[\cdot]$ using samples drawn by the Langevin Dynamics

$$x^{(k+1)} = x^{(k)} - \varepsilon \nabla_x E_\theta\left(x^{(k)}\right) + \mathcal{N}(0, 2\varepsilon).$$

  - Replay buffer for initializing the LD chain.
  - L_2-regularization on the energy function.



Markov Random Fields

Deep-Nets-Based EBMs:

- [DM19]

ImageNet32x32 Generation

Model	Inception	FID
CIFAR-10 Unconditional		
PixelCNN (Van Oord et al., 2016)	4.60	65.93
PixelIQN (Ostrovski et al., 2018)	5.29	49.46
EBM (single)	6.02	40.58
DCGAN (Radford et al., 2016)	6.40	37.11
WGAN + GP (Gulrajani et al., 2017)	6.50	36.4
EBM (10 historical ensemble)	6.78	38.2
SNGAN (Miyato et al., 2018)	**8.22**	21.7
CIFAR-10 Conditional		
Improved GAN	8.09	-
EBM (single)	8.30	37.9
Spectral Normalization GAN	**8.59**	25.5
ImageNet 32x32 Conditional		
PixelCNN	8.33	33.27
PixelIQN	10.18	22.99
EBM (single)	**18.22**	**14.31**
ImageNet 128x128 Conditional		
ACGAN (Odena et al., 2017)	28.5	-
EBM* (single)	28.6	43.7
SNGAN	**36.8**	**27.62**

Topics

Online Inference

Online Inference

Update est. of $p(z|x_{1:n})$ to approx. $p\left(z\middle|x_{1:(n+1)}\right)$ in $O(1)$ computation.

- Online nature of Bayes' rule: $p\left(z\middle|x_{1:(n+1)}\right) \propto p(z|x_{1:n})p(x_{n+1}|z)$.
- Variational inference
  - Streaming Variational Bayes [BBW+13]:
    Let q_{n+1} approx. $q_n(z)p(x_{n+1}|z)$ (i.e., inference with prior $q_n(z)$).
  - Stochastic-optimization-based method [HBB10]:
    1) solve global param. $\hat{\lambda}$ with n-replicated x_{n+1};
    2) $\lambda_{n+1} = (1-\varepsilon_{n+1})\lambda_n + \varepsilon_{n+1}\hat{\lambda}$ with Robbins-Monro sequence $\{\varepsilon_n\}_n$.
  - Online Bayesian passive-aggressive [SZ14]: regularize posterior with supervision.
    - Online PA: $z_{n+1} = \arg\min_z \frac{1}{2}\|z - z_n\|^2 + \lambda\ell(z; x_{n+1}, y_{n+1})$.
    - Online BayesPA: $q_{n+1} = \arg\min_q \mathrm{KL}(q, q_n) - \mathbb{E}_q[\log p(x_{n+1}|z)] + \lambda\ell(q; x_{n+1}, y_{n+1})$.
  - $q(z)$ needs to be explicit.

Online Inference

Update est. of $p(z|x_{1:n})$ to approx. $p\left(z\middle|x_{1:(n+1)}\right)$ in $O(1)$ computation.

- Online nature of Bayes' rule: $p\left(z\middle|x_{1:(n+1)}\right) \propto p(z|x_{1:n})p(x_{n+1}|z)$.
- MCMC:
  - Importance-sampling-based [Cho02]:

    if $\{z^{(i)}\}_i \sim p(z|x_{1:n})$, then weighted particles

    $$\left\{z^{(i)}: \frac{p\left(z^{(i)}\middle|x_{1:(n+1)}\right)}{p\left(z^{(i)}\middle|x_{1:n}\right)} = p\left(x_{n+1}\middle|z^{(i)}\right)\right\}_i \sim p\left(z\middle|x_{1:(n+1)}\right).$$

    - *Resample* the particles according to the weight;
    - *Rejuvenate* by an MCMC to avoid particle degeneracy.
  - Particle MCMC [ADH10]:
    - Update particles by a transition proposal $q(z_{n+1}|z_n, x_{n+1})$;
    - Adjust the particles by an MH step.

Topics

Asymmetry of KL

Asymmetry of KL

To approximate distribution p with distribution q (typically parameterized as q_θ),

- Forward KL / Inclusive KL:

$$\arg\min_q \mathrm{KL}(p, q) = \arg\min_q \mathbb{E}_p \left[\log\left(\frac{p}{q}\right)\right]$$
$$= \arg\min_q \mathbb{E}_p[\log q].$$

  - Used in MLE.
  - $q(z) \neq 0$ wherever $p(z) \neq 0$ (otherwise huge penalty) [TOB16, Hus15]. $\mathrm{supp}(q)$ includes $\mathrm{supp}(p)$.
  - Explains unrealistic or blurry generation of MLE-trained autoregressive models and ELBO-trained VAEs.

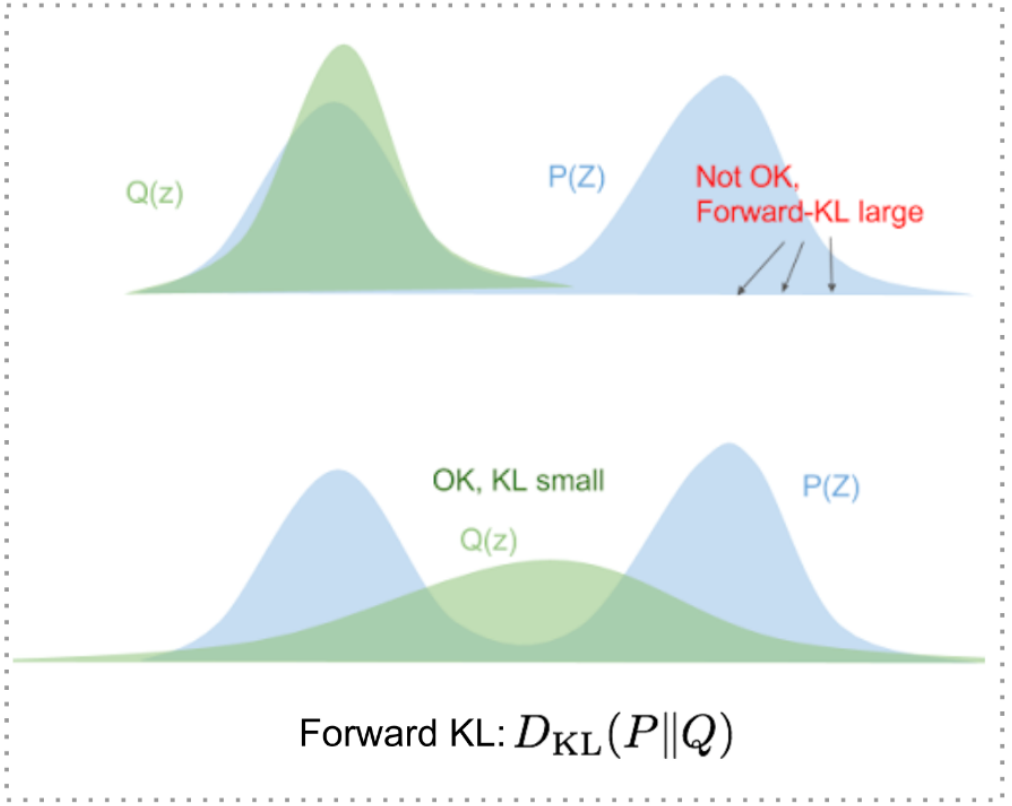


Figure: [https://blog.evjang.com/2016/08/variational-bayes.html], [https://lilianweng.github.io/lil-log/2018/08/12/from-autoencoder-to-beta-vae.html]

Asymmetry of KL

To approximate distribution p with distribution q (typically parameterized as q_θ),

- Forward KL / Inclusive KL:

  $\arg\min_q \mathrm{KL}(p, q)$.

  - Moment seeking behavior:

    Let $q_\phi(z) = \exp\left(\phi^\top T(z) - A(\phi)\right) h(z)$.

    $\arg\min_\phi \mathrm{KL}\left(p, q_\phi\right)$

    $= \arg\min_\phi \phi^\top \mathbb{E}_p[T(z)] - A(\phi)$.

    grad. obj.$= \mathbb{E}_p[T(z)] - \mathbb{E}_{q_\phi}[T(z)]$.

  - Intractable gradient $\nabla_\phi \mathrm{KL}\left(p, q_\phi\right)$ with unnormalized $p \propto \tilde{p}$.

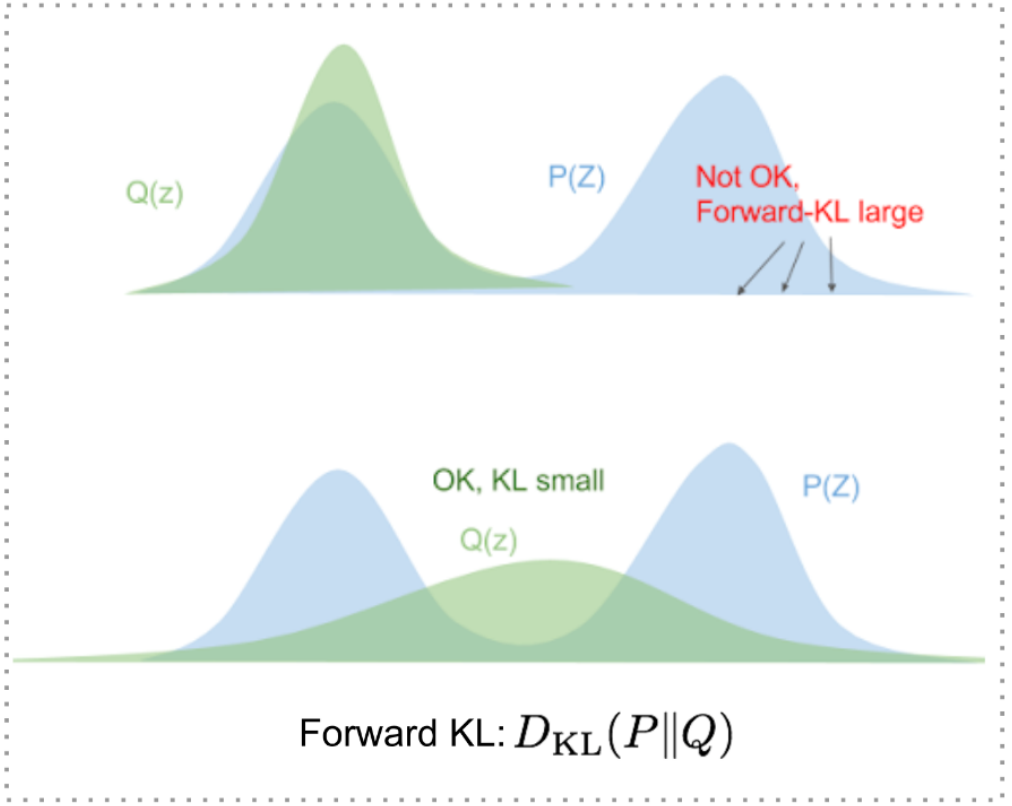


Figure: [https://blog.evjang.com/2016/08/variational-bayes.html],
[https://lilianweng.github.io/lil-log/2018/08/12/from-autoencoder-to-beta-vae.html]

Asymmetry of KL

To approximate distribution p with distribution q (typically parameterized as q_θ),

- Reversed KL / Exclusive KL:

$$\arg\min_q \mathrm{KL}(q,p) = \arg\min_q \mathbb{E}_q\left[\log\left(\frac{q}{p}\right)\right].$$

  - Tractable gradient $\nabla_\phi \mathrm{KL}(q_\phi, p)$ with unnormalized $p \propto \tilde{p}$:

    $\nabla_\phi \mathrm{KL}(q_\phi, p) = \mathbb{E}_{q_\phi}\left[(\nabla_\phi \log q_\phi)\log\frac{q_\phi}{\tilde{p}}\right]$.

    (since $\nabla_\phi \mathbb{E}_{q_\phi}[f] = \mathbb{E}_{q_\phi}[f\,\nabla_\phi \log q_\phi]$, and $\nabla_\phi \mathbb{E}_{q_\phi}[C] = \nabla_\phi C = 0$)

  - Used in variational inference (except for Expectation Propagation).

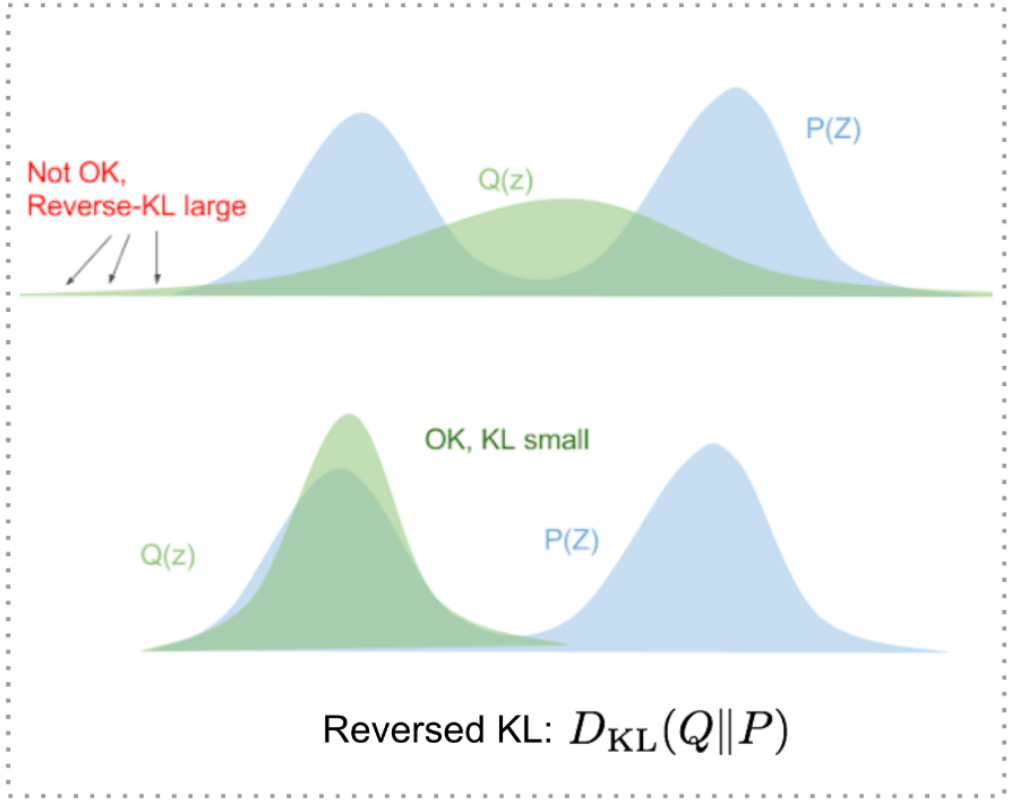


Figure: [https://blog.evjang.com/2016/08/variational-bayes.html], [https://lilianweng.github.io/lil-log/2018/08/12/from-autoencoder-to-beta-vae.html]

Asymmetry of KL

To approximate distribution p with distribution q (typically parameterized as q_θ),

- Reversed KL / Exclusive KL:

$$\arg\min_q \mathrm{KL}(q,p) = \arg\min_q \mathbb{E}_q\left[\log\left(\frac{q}{p}\right)\right].$$

  - $q(z) = 0$ wherever $p(z) = 0$ (otherwise huge penalty) [TOB16, Hus15]. $\mathrm{supp}(q)$ tends to exclude $\mathrm{supp}(p)$.
  - Mode seeking behavior.
  - Jensen-Shannon divergence behaves more like Reversed KL [Hus15].
  - Explains underestimated variance of variational inference and realistic but mode-collapsed generation of GANs.

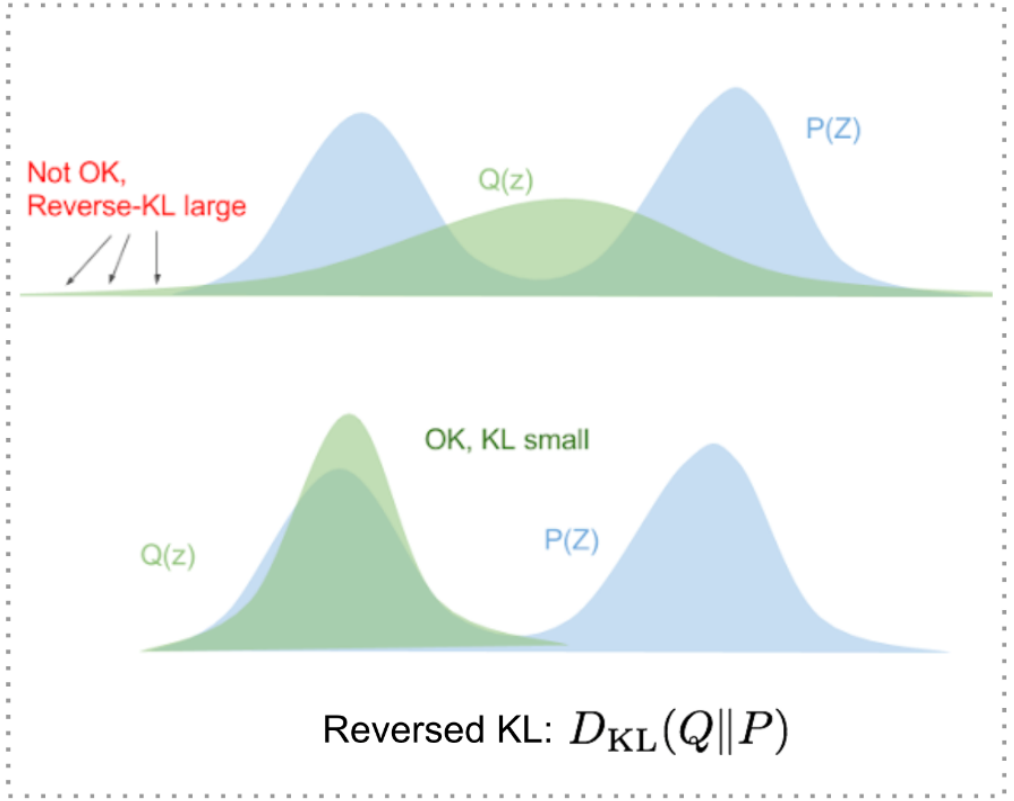


Figure: [https://blog.evjang.com/2016/08/variational-bayes.html], [https://lilianweng.github.io/lil-log/2018/08/12/from-autoencoder-to-beta-vae.html]

Asymmetry of KL

- Adversarial Symmetric Variational Autoencoder [PWH+17]
  - Original VAE objective:

$$\mathbb{E}_{q(x)}[\mathrm{ELBO}(x)] = -\mathrm{KL}\Big(q_\phi(z,x), p_\theta(z,x)\Big) + \mathbb{E}_{q(x)}[\log q(x)].$$

$:= \hat{p}(x)$

    - For inference (i.e., q_ϕ): mode-seeking/collapsing.
    - For learning (i.e., p_θ): over-covering, unrealistic/blurry generation.

$$\mathbb{E}_{\hat{p}(x)}[\mathrm{ELBO}(x)] \le \mathbb{E}_{\hat{p}}[\log p_\theta(x)] = -\mathrm{KL}(\hat{p}, p_\theta) + \mathrm{const.}$$

  - Symmetrized objective:

$$\begin{aligned}\mathcal{L}_x + \mathcal{L}_z := \quad & \mathbb{E}_{q(x)}[\log p_\theta(x)] - \mathrm{KL}\Big(q_\phi(z,x), p_\theta(z,x)\Big) \\ & + \mathbb{E}_{p_\theta(z)}\big[\log q_\phi(z)\big] - \mathrm{KL}\Big(p_\theta(z,x), q_\phi(z,x)\Big).\end{aligned}$$

  - Optimize by adversarial density ratio estimation [MNG17] and reparameterization:

$$\mathcal{L}_x = \mathbb{E}_{q_\phi(z,x)}\left[\log \frac{p_\theta(x)p_\theta(z)}{q_\phi(z,x)} + \log p_\theta(x|z)\right].$$

    Similar optimization for $\mathcal{L}_z$.

Asymmetry of KL

- Dual Discriminator Generative Adversarial Nets [NLVP17]
  - The JS divergence

$$\mathrm{JS}(q_\phi, \hat{p}) = \frac{1}{2}\mathrm{KL}\left(q_\phi, \frac{q_\phi + \hat{p}}{2}\right) + \frac{1}{2}\mathrm{KL}\left(\hat{p}, \frac{q_\phi + \hat{p}}{2}\right)$$

$$q_\phi \coloneqq (G_\phi)_{\#}(p(z))$$

    behaves similarly [Hus15] as the Reversed KL, $\mathrm{KL}(q_\phi, \hat{p})$, thus mode-collapse.
  - Symmetrized objective:

$$\mathcal{L}(\phi) = \alpha(\log\alpha - 1) + \beta(\log\beta - 1) + \alpha\mathrm{KL}(\hat{p}, q_\phi) + \beta\mathrm{KL}(q_\phi, \hat{p}).$$

  - Optimize by adversarial training with two discriminators:

$$\min_\phi \max_{D_1, D_2} \mathcal{J}(\phi, D_1, D_2) \coloneqq \alpha\mathbb{E}_{\hat{p}(x)}[\log D_1(x)] + \mathbb{E}_{p(z)}\left[-D_1\left(G_\phi(z)\right)\right] + \mathbb{E}_{\hat{p}(x)}[-D_2(x)] + \beta\,\mathbb{E}_{p(z)}\left[\log D_2\left(G_\phi(z)\right)\right].$$

$$\mathcal{J}(\phi, D_1^*, D_2^*) = \mathcal{L}(\phi).$$

Asymmetry of KL

- Alpha-divergence [Ama85]: $D_\alpha(p,q) \coloneqq \frac{1}{\alpha(\alpha-1)}\left(1-\int p(z)^\alpha q(z)^{1-\alpha}\,\mathrm{d}z\right)$.
  - Exclusive KL divergence: $D_0(p,q) = \mathrm{KL}(q,p)$.
  - Hellinger distance: $D_{1/2}(p,q) = 4\mathrm{Hel}^2(q,p) = 2\int\left(\sqrt{p(z)}-\sqrt{q(z)}\right)^2\,\mathrm{d}z$.
  - Inclusive KL divergence: $D_1(p,q) = \mathrm{KL}(p,q)$.
- Variational inference with alpha-divergence:
  - Alpha free energy [HLR+16]: $\mathcal{L}_\alpha(q) \coloneqq -\frac{1}{\alpha}\sum_n \log \mathbb{E}_q\left[\left(\frac{p\left(y^{(n)}\middle|z,x^{(n)}\right)p(z)^{\frac{1}{N}}}{q(z)^{\frac{1}{N}}}\right)^\alpha\right]$.
  - Minimization in the EP style (q in exponential family): [HLR+16].
  - Minimization using variational dropout: [LG17].

Asymmetry of KL

- Variational inference with alpha-divergence:
  - Alpha free energy [HLR+16]: $\mathcal{L}_\alpha(q) \coloneqq -\frac{1}{\alpha}\sum_n \log \mathbb{E}_q\left[\left(p\left(y^{(n)}\middle|z,x^{(n)}\right)p(z)^{\frac{1}{N}}/q(z)^{\frac{1}{N}}\right)^\alpha\right]$.
  - Minimization using variational dropout [LG17]:
    - With $q(z) \coloneqq \frac{1}{Z_q}\tilde{q}(z)\left(\tilde{q}(z)/p(z)\right)^{\frac{\alpha}{N-\alpha}}$:

$$\mathcal{L}_\alpha(q) = R_\beta\left(\tilde{q}(z),p(z)\right) - \frac{1}{\alpha}\sum_n \log \mathbb{E}_{\tilde{q}}\left[p\left(y^{(n)}\middle|z,x^{(n)}\right)^\alpha\right], \beta = \frac{N}{N-\alpha}.$$

Renyi divergence

    - For $\frac{\alpha}{N} \to 0$: $q \to \tilde{q}$, $R_\beta\left(\tilde{q}(z),p(z)\right) \to \mathrm{KL}\left(\tilde{q}(z),p(z)\right)$, and $\mathcal{L}_\alpha(q) \to$

$$\mathrm{KL}\left(q(z),p(z)\right) - \frac{1}{\alpha}\sum_n \log \mathbb{E}_q\left[p\left(y^{(n)}\middle|z,x^{(n)}\right)^\alpha\right] =: \tilde{\mathcal{L}}_\alpha(q).$$

    - Variational Dropout Inference:

$$z \sim q(z) \Leftrightarrow z_i = b_i \odot (M_i + \epsilon_i) + (1-b_i)\odot \epsilon_i, b_i \sim \mathrm{Bern}(p_i), \epsilon_i \sim \mathcal{N}(0,\delta^2).$$

$$\tilde{\mathcal{L}}_\alpha(q) \approx \sum_i \frac{p_i}{2}\|M_i\|_F^2 - \frac{1}{\alpha}\mathrm{LogSumExp}_n\left(-\frac{\alpha\tau}{2}\left\|y^{(n)} - \hat{y}^{(n)}\right\|_2^2\right) + \frac{ND}{2}\log\tau\,.$$

Topics

Bayesian Reinforcement Learning

(far from complete)

Bayesian Reinforcement Learning

- Model-based Bayesian RL:

  update model with observations using Bayes' rule.
  - POMDP, Bayesian MDP, Bayesian Bandits, BEETLE...

Bayesian Reinforcement Learning

- Model-free Bayesian RL
  - Policy Gradient:

$\theta^* = \arg\max_\theta \{J(\theta) \coloneqq \mathbb{E}_{s_0,a_0,\dots}[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)]\}$, where $a_t \sim \pi_\theta(a|s_t)$.

  - Entropy Regularized Policy Gradient:
    - Model the policy by parameter θ and treat $\theta \sim q(\theta)$ as a r.v. [LRLP17]:

$$q^*(\theta) = \arg\max_q \mathbb{E}_{q(\theta)}[J(\theta)] - \alpha\, \mathrm{KL}(q, q_0).$$

$$\propto q_0(\theta) \exp\left(\frac{1}{\lambda} J(\theta)\right).$$

posterior

prior

likelihood

    - Bayesian “inference” by SVGD.
    - Helps exploration; increases data efficiency.

Bayesian Reinforcement Learning

- Model-free Bayesian RL
  - Entropy Regularized Policy Gradient:
    - Directly regularize the policy [HTAL17]:

$$\pi^* = \arg\max_{\pi} \sum_t \mathbb{E}_{(s_t,a_t)\sim p(s_t)\pi(a_t|s_t)}\left[r(s_t,a_t) + \alpha\mathbb{H}[\pi(\cdot\,|s_t)]\right].$$

      - Soft Q-Iteration: using soft Bellman Eq.
      - Soft Q-learning: $J_Q(\theta) = \mathbb{E}_{\mathbf{s}_t\sim q_{\mathbf{s}_t},\mathbf{a}_t\sim q_{\mathbf{a}_t}}\left[\frac{1}{2}\left(\hat{Q}^{\bar{\theta}}_{\text{soft}}(\mathbf{s}_t,\mathbf{a}_t) - Q^{\theta}_{\text{soft}}(\mathbf{s}_t,\mathbf{a}_t)\right)^2\right]$

Target Q-value

      - Actor-critic with amortized SVGD: Bayesian inference interpretation.

$$\pi^*_{\text{MaxEnt}}(\mathbf{a}_t|\mathbf{s}_t) = \exp\left(\tfrac{1}{\alpha}(Q^*_{\text{soft}}(\mathbf{s}_t,\mathbf{a}_t) - V^*_{\text{soft}}(\mathbf{s}_t))\right)$$

$$J_\pi(\phi;\mathbf{s}_t) = \mathrm{D}_{\mathrm{KL}}\left(\pi^\phi(\cdot\,|\mathbf{s}_t)\,\middle\|\,\exp\left(\frac{1}{\alpha}\left(Q^\theta_{\text{soft}}(\mathbf{s}_t,\cdot) - V^\theta_{\text{soft}}\right)\right)\right).$$

      - Wasserstein gradient flow version [ZCLC18]:
        - Use a generator $g_\phi(a|s)$ (amortization) that minimizes the KL for every s.

Topics

Causality

Clipped from the slides “Causality” (2018) of:
Ruifei Cui (Radboud University Nijmegen)

Causality

- Correlation unnecessarily means Causation!

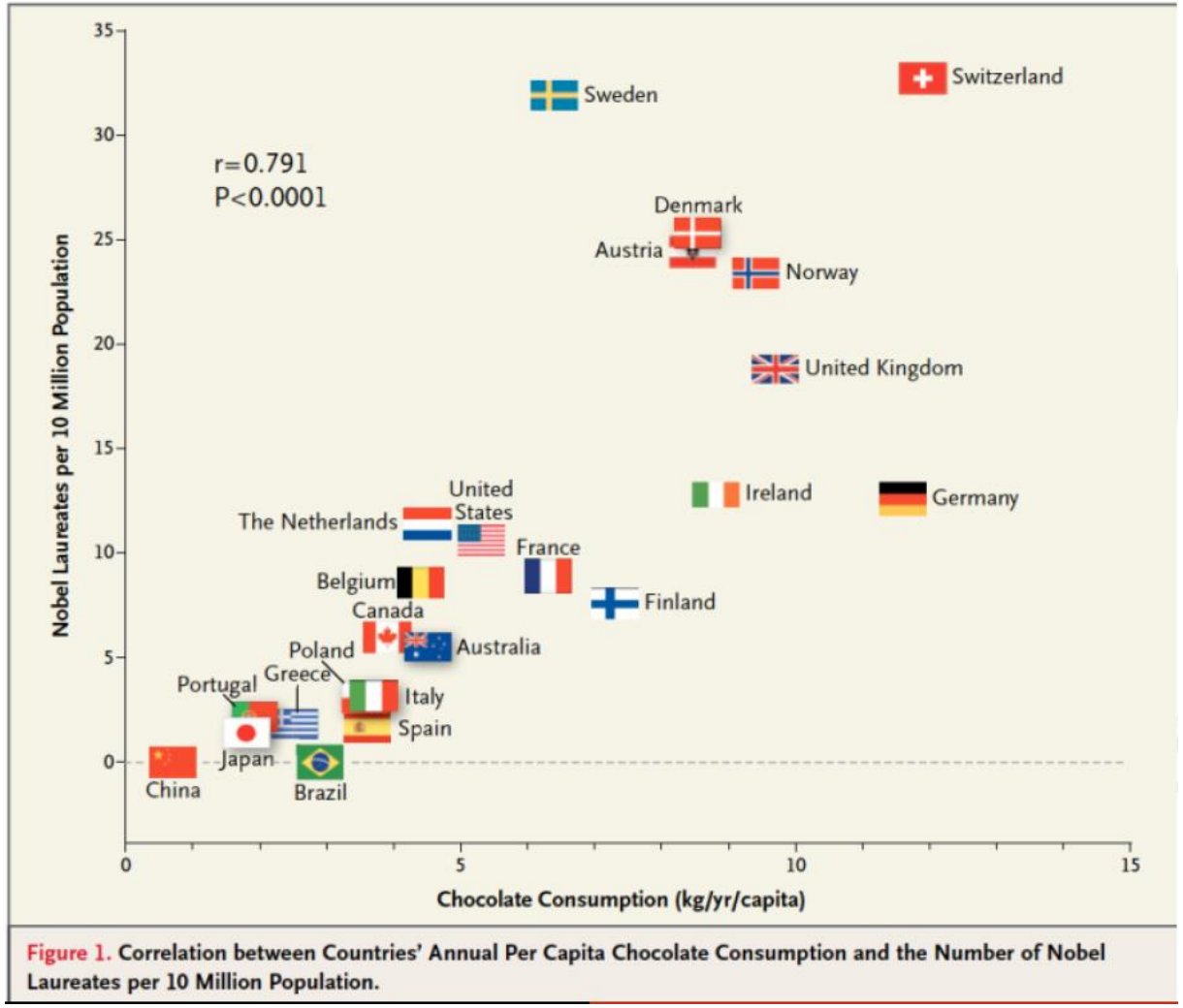


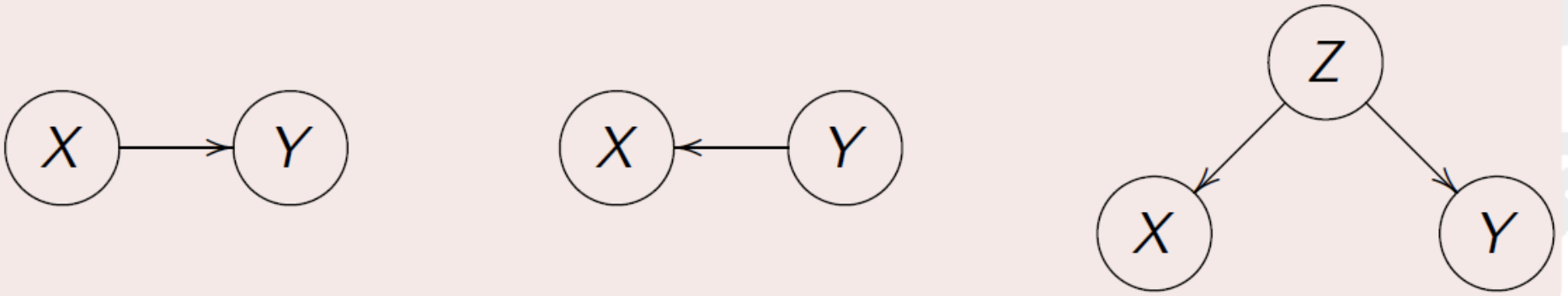
Figure 1. Correlation between Countries' Annual Per Capita Chocolate Consumption and the Number of Nobel Laureates per 10 Million Population.

Credit: Ruifei Cui

Causality

- Correlation unnecessarily means Causation!
  - Correlation is symmetric. Causality is not.
    - Markov Random Fields do not convey causality information.
  - Causality $\Longrightarrow$ Correlation

Three cases that incur correlation



Credit: Ruifei Cui

Causality

Probabilistic Inference (traditional machine learning)

- Models the distribution of the data
- Focuses on predicting consequences of observations

Causal Inference

- Models the mechanism that generates the data
- Also allows to predict results of interventions

The fundamental difference is that Probabilistic Inference talks about observing while Causal Inference talks about intervening.

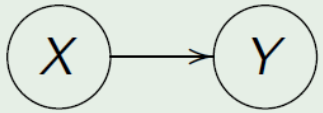
Credit: Ruifei Cui

Causality

Definition (Interventional Distribution)

The resulting distribution of Y after setting $X = x$ is called the interventional distribution, denoted by $p(Y|\mathrm{do}(X = x))$. It is different from $p(Y|X = x)$ in general.

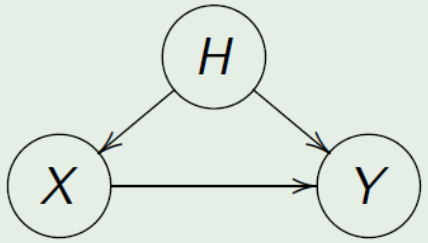
Example



$$p(y|\mathrm{do}(X = x)) = p(y|X = x)$$

Yes!

Example



$$p(y|\mathrm{do}(X = x)) = \int p(h)p(y|x,h)dh \neq p(y|X = x) = \int p(h|x)p(y|x,h)dh$$

No!

X is **set** to the value x. (x does not provide information on H).

X may be caused by H, so x provides information on H, which in turn also affects Y.

Credit: Ruifei Cui

Causality

Theorem (Back-door Criterion (Pearl, 2000))

For an ordered pair of variables (X, Y) and a set of variables $\boldsymbol{H}$ in a DAG $\mathcal{G}$ where $X, Y \notin \boldsymbol{H}$, if

- *no vertex in $\boldsymbol{H}$ is a descendant of X;*
- *$\boldsymbol{H}$ blocks all the back-door paths between X and Y (i.e., the path that starts an arrowhead at X);*

then the interventional distribution can be obtained by adjusting for $\boldsymbol{H}$:

$$p(y|\mathrm{do}\,(X = x)) = \int p(\boldsymbol{h})p(y|x, \boldsymbol{h})d\boldsymbol{h}.$$

For the special case $\boldsymbol{H} = \emptyset$, this reduces to

$$p(y|\mathrm{do}\,(X = x)) = p(y|X = x).$$

Credit: Ruifei Cui

Causality

Definition (Causal Effect)

The causal effect of X on Y is defined as

- X is binary: $\mathbb{E}(Y|\mathrm{do}(X=1)) - \mathbb{E}(Y|\mathrm{do}(X=0))$
- X is continuous: $\frac{\partial}{\partial x}\mathbb{E}(Y|\mathrm{do}(X=x))$

Credit: Ruifei Cui

Causality

Simpson's Paradox

You are investigating the effectiveness of a drug against a deadly disease. You are given access to data collected by health insurance companies about their customers, based on which you have the following table.

	Recovery	No recovery	Total	Recovery rate
Drug	20	20	40	...%
No drug	16	24	40	...%
Total	36	44	80	

- The recovery rates for customers with drug and without drug are 50% and 40% respectively.
- If you were a doctor, would you recommend your patients to take the drug?

Credit: Ruifei Cui

Causality

Simpson's Paradox

Upon closer inspection of the data, you notice something peculiar when you group patients according to gender.

Males	Recovery	No recovery	Total	Recovery rate
Drug	18	12	30	...%
No drug	7	3	10	...%
Total	25	15	40	

Females	Recovery	No recovery	Total	Recovery rate
Drug	2	8	10	...%
No drug	9	21	30	...%
Total	11	29	40	

- For males, with drug: 60%; no drug: 70%.
- For females, with drug: 20%; no drug: 30%.
- Would you recommend your patients to take the drug?

Credit: Ruifei Cui

Causality

- The probability of recovery ($R = 1$) is higher for patients that took the drug ($D = 1$):

$$p(R = 1|D = 1) > p(R = 1|D = 0)$$

- For both male and female patients, the relation is opposite:

$$p(R = 1|D = 1, G = male) < p(R = 1|D = 0, G = male),$$
$$p(R = 1|D = 1, G = female) < p(R = 1|D = 0, G = female)$$

Credit: Ruifei Cui

Causality

1. The research question of interest should be if

$$p(R = 1|\mathrm{do}\,(D = 1)) > p(R = 1|\mathrm{do}\,(D = 0)),$$

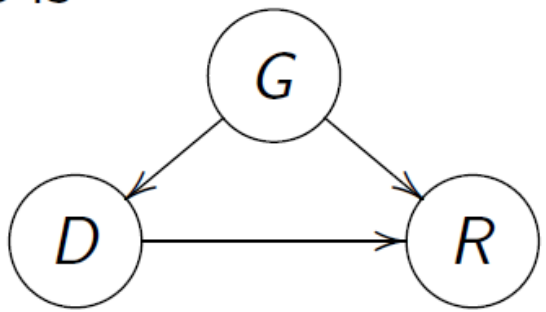
D is SET to the value. Not caused by G.

rather than if

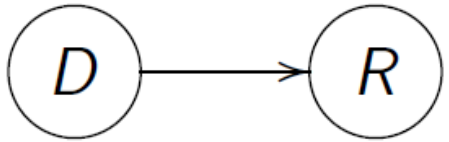
$$p(R = 1|D = 1) > p(R = 1|D = 0).$$

D gives information about G, which also affects R.

2. The true causal structure is



rather than

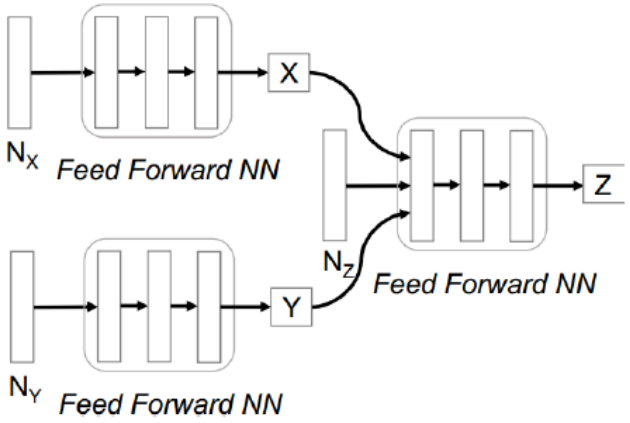


Credit: Ruifei Cui

Causality

Causal Generative Adversarial Networks (CausalGAN)

- CausalGAN[2] aims to learn a causal implicit generative model (CiGM) with adversarial training: models that allow sampling from not only the true observational but also the interventional distributions.
- The key idea is to design the generator according to a causal graph. For example, given the graph $X \rightarrow Z \leftarrow Y$, we have the generator:



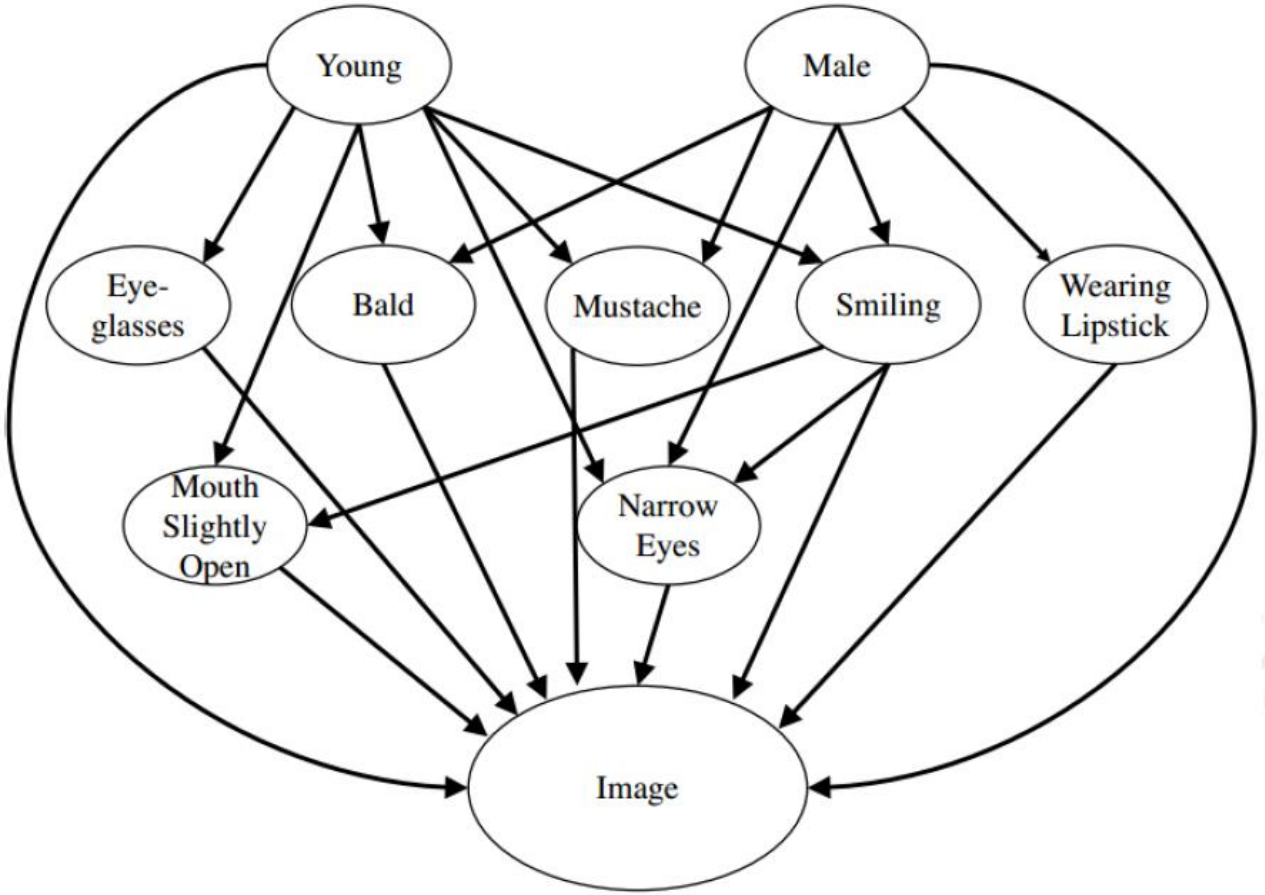
[2]Murat et al., "CausalGAN." ICLR, 2018.

Credit: Ruifei Cui

Causality

An Example of CausalGAN

The generator is designed according to the following causal graph:



Credit: Ruifei Cui

Causality

An Example of CausalGAN

Resulting images generated by the trained model:
Top row: Intervene Mustache=1; Bottom: Condition Mustache=1

Conditional GANs [MO14, OOS17] tend to behave like this.

- Top row contains both males and females because intervening on 'Mustache = 1' does not change the distribution of 'Male'.
- Bottom row only contains males because conditioning on 'Mustache = 1' tells the system 'Male = 1'.

Credit: Ruifei Cui

Topics not Covered

- Gaussian Process
- Bayesian Optimization
- Bayesian Nonparametrics
- Semi-Supervised Learning
- Classical ABCs: message passing, belief propagation
- Dynamic models (e.g., Hidden Markov Models)
- ...

Future Work

Future Work

- Decrease auto-correlation in MCMC explicitly.
- High-dimensional degeneracy issue of particle-base VI.
- Online variational inference with more flexible q.
- Online MCMC with more effective transition.

Thanks!

References

References

- Bayesian Inference: Variational Inference
  - Explicit Parametric VI: model-agnostic variational distribution.
    - [GHB12] Gershman, S., Hoffman, M., & Blei, D. (2012). Nonparametric variational inference. arXiv preprint arXiv:1206.4665.
    - [HBWP13] Hoffman, M. D., Blei, D. M., Wang, C., & Paisley, J. (2013). Stochastic variational inference. *The Journal of Machine Learning Research*, *14*(1), 1303-1347.
    - [KW14] Kingma, D.P., & Welling, M. (2014). Auto-encoding variational Bayes. In *Proceedings of the International Conference on Learning Representations*.
    - [RMW14] Rezende, D. J., Mohamed, S., & Wierstra, D. (2014). Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the International Conference on Machine Learning.*
    - [RGB14] Ranganath, R., Gerrish, S., & Blei, D. (2014). Black box variational inference. In *Artificial Intelligence and Statistics* (pp. 814-822).

References

- Bayesian Inference: Variational Inference
  - Explicit Parametric VI: model-agnostic variational distribution.
    - [RM15] Rezende, D.J., & Mohamed, S. (2015). Variational inference with normalizing flows. In *Proceedings of the International Conference on Machine Learning* (pp. 1530-1538).
    - [KSJ+16] Kingma, D.P., Salimans, T., Jozefowicz, R., Chen, X., Sutskever, I., & Welling, M. (2016). Improved variational inference with inverse autoregressive flow. In *Advances in neural information processing systems* (pp. 4743-4751).

References

- Bayesian Inference: Variational Inference
  - Explicit Parametric VI: expectation propagation
    - [Min01] Minka, T. P. (2001). Expectation propagation for approximate Bayesian inference. In *Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence*(pp. 362-369). Morgan Kaufmann Publishers Inc..
    - [LHT15] Li, Y., Hernández-Lobato, J. M., & Turner, R. E. (2015). Stochastic expectation propagation. In *Advances in neural information processing systems* (pp. 2323-2331).
    - [HA15] Hernández-Lobato, J. M., & Adams, R. (2015). Probabilistic backpropagation for scalable learning of Bayesian neural networks. In *International Conference on Machine Learning* (pp. 1861-1869).

References

- Bayesian Inference: Variational Inference
  - Implicit Parametric VI: density ratio estimation
    - [MSJ+15] Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I., & Frey, B. (2016). Adversarial Autoencoders. In *Proceedings of the International Conference on Learning Representations*.
    - [MNG17] Mescheder, L., Nowozin, S., & Geiger, A. (2017). Adversarial variational Bayes: Unifying variational autoencoders and generative adversarial networks. In *Proceedings of the International Conference on Machine Learning* (pp. 2391-2400).
    - [Hus17] Huszár, F. (2017). Variational inference using implicit distributions. *arXiv preprint* arXiv:1702.08235.
    - [TRB17] Tran, D., Ranganath, R., & Blei, D. (2017). Hierarchical implicit models and likelihood-free variational inference. In *Advances in Neural Information Processing Systems* (pp. 5523-5533).
    - [SSZ18a] Shi, J., Sun, S., & Zhu, J. (2018). Kernel Implicit Variational Inference. In *Proceedings of the International Conference on Learning Representations*.

References

- Bayesian Inference: Variational Inference
  - Implicit Parametric VI: gradient estimation
    - [VLBM08] Vincent, P., Larochelle, H., Bengio, Y., & Manzagol, P. A. (2008). Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning* (pp. 1096-1103). ACM.
    - [LT18] Li, Y., & Turner, R. E. (2018). Gradient estimators for implicit models. In *Proceedings of the International Conference on Learning Representations*.
    - [SSZ18b] Shi, J., Sun, S., & Zhu, J. (2018). A spectral approach to gradient estimation for implicit distributions. In *Proceedings of the 35th International Conference on Machine Learning* (pp. 4651-4660).

References

- Bayesian Inference: Variational Inference
  - Particle-Based VI: methods
    - [LW16] Liu, Q., & Wang, D. (2016). Stein variational gradient descent: A general purpose Bayesian inference algorithm. In *Advances In Neural Information Processing Systems* (pp. 2378-2386).
    - [CZ17] Chen, C., & Zhang, R. (2017). Particle optimization in stochastic gradient MCMC. *arXiv preprint arXiv:1711.10927*.
    - [LZ18] Liu, C., & Zhu, J. (2018). Riemannian Stein Variational Gradient Descent for Bayesian Inference. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence* (pp. 3627-3634).
    - [CMG+18] Chen, W. Y., Mackey, L., Gorham, J., Briol, F. X., & Oates, C. J. (2018). Stein points. *arXiv preprint arXiv:1803.10161*.
    - [FCSS18] Futami, F., Cui, Z., Sato, I., & Sugiyama, M. (2018). Frank-Wolfe Stein sampling. *arXiv preprint arXiv:1805.07912*.

References

- Bayesian Inference: Variational Inference
  - Particle-Based VI: methods
    - [CZW+18] Chen, C., Zhang, R., Wang, W., Li, B., & Chen, L. (2018). A unified particle-optimization framework for scalable Bayesian sampling. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*.
    - [LZC+19] Liu, C., Zhuo, J., Cheng, P., Zhang, R., Zhu, J., & Carin, L. (2019). Understanding and Accelerating Particle-Based Variational Inference. In *Proceedings of the 36th International Conference on Machine Learning* (pp. 4082-4092).
    - [LZZ19] Liu, C., Zhuo, J., & Zhu, J. (2019). Understanding MCMC Dynamics as Flows on the Wasserstein Space. In *Proceedings of the 36th International Conference on Machine Learning* (pp. 4093-4103).

References

- Bayesian Inference: Variational Inference
  - Particle-Based VI: amortized methods
    - [FWL17] Feng, Y., Wang, D., & Liu, Q. (2017). Learning to Draw Samples with Amortized Stein Variational Gradient Descent. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*.
    - [PGH+17] Pu, Y., Gan, Z., Henao, R., Li, C., Han, S., & Carin, L. (2017). VAE learning via Stein variational gradient descent. In *Advances in Neural Information Processing Systems* (pp. 4236-4245).

References

- Bayesian Inference: Variational Inference
  - Particle-Based VI: theories
    - [Liu17] Liu, Q. (2017). Stein variational gradient descent as gradient flow. In *Advances in neural information processing systems* (pp. 3115-3123).
    - [CMG+18] Chen, W. Y., Mackey, L., Gorham, J., Briol, F. X., & Oates, C. J. (2018). Stein points. *arXiv preprint arXiv:1803.10161*.
    - [FCSS18] Futami, F., Cui, Z., Sato, I., & Sugiyama, M. (2018). Frank-Wolfe Stein sampling. *arXiv preprint arXiv:1805.07912*.
    - [ZZC18] Zhang, J., Zhang, R., & Chen, C. (2018). Stochastic particle-optimization sampling and the non-asymptotic convergence theory. *arXiv preprint arXiv:1809.01293*.
    - [LZC+19] Liu, C., Zhuo, J., Cheng, P., Zhang, R., Zhu, J., & Carin, L. (2019). Understanding and Accelerating Particle-Based Variational Inference. In *Proceedings of the 36th International Conference on Machine Learning* (pp. 4082-4092).

References

- Bayesian Inference: Variational Inference
  - Particle-Based VI: literatures on Wasserstein space
    - [JKO98] Jordan, R., Kinderlehrer, D., & Otto, F. (1998). The variational formulation of the Fokker-Planck equation. *SIAM journal on mathematical analysis*, *29*(1), 1-17.
    - [BB00] Benamou, J. D., & Brenier, Y. (2000). A computational fluid mechanics solution to the Monge-Kantorovich mass transfer problem. *Numerische Mathematik*, *84*(3), 375-393.
    - [Ott01] Otto, F. (2001). The geometry of dissipative evolution equations: the porous medium equation. *Taylor & Francis*.
    - [Vil08] Villani, C. (2008). *Optimal transport: old and new* (Vol. 338). Springer Science & Business Media.
    - [AGS08] Ambrosio, L., Gigli, N., & Savaré, G. (2008). *Gradient flows: in metric spaces and in the space of probability measures*. Springer Science & Business Media.

References

- Bayesian Inference: MCMC
  - Classical MCMC
    - [MRR+53] Metropolis, N., Rosenbluth, A. W., Rosenbluth, M.N., Teller, A.H., & Teller, E. (1953). Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6), pp.1087-1092.
    - [Has70] Hastings, W. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1), pp.97-109.
    - [GG87] Geman, S., & Geman, D. (1987). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. In *Readings in computer vision* (pp. 564-584).
    - [Nea01] Neal, R. M. (2001). Annealed importance sampling. *Statistics and computing*, *11*(2), 125-139.
    - [ADDJ03] Andrieu, C., De Freitas, N., Doucet, A., & Jordan, M. I. (2003). An introduction to MCMC for machine learning. *Machine learning*, *50*(1-2), 5-43.

References

- Bayesian Inference: MCMC
  - Dynamics-Based MCMC: full-batch
    - [Lan08] Langevin, P. (1908). Sur la théorie du mouvement Brownien. *Compt. Rendus*, *146*, 530-533.
    - [DKPR87] Duane, S., Kennedy, A.D., Pendleton, B.J., Roweth, D. (1987). Hybrid Monte Carlo. *Physics letters B*, 195(2), pp.216-222.
    - [RT96] Roberts, G. O., & Tweedie, R. L. (1996). Exponential convergence of Langevin distributions and their discrete approximations. *Bernoulli*, *2*(4), 341-363.
    - [RS02] Roberts, G.O., & Stramer, O. (2002). Langevin diffusions and Metropolis-Hastings algorithms. *Methodology and computing in applied probability*, 4(4), pp.337-357.
    - [Nea11] Neal, R.M. (2011). MCMC using Hamiltonian dynamics. Handbook of Markov chain Monte Carlo, 2(11), p.2.
    - [ZWC+16] Zhang, Y., Wang, X., Chen, C., Henao, R., Fan, K., & Carin, L. (2016). Towards unifying Hamiltonian Monte Carlo and slice sampling. In *Advances in Neural Information Processing Systems* (pp. 1741-1749).

References

- Bayesian Inference: MCMC
  - Dynamics-Based MCMC: full-batch
    - [TRGT17] Tripuraneni, N., Rowland, M., Ghahramani, Z., & Turner, R. (2017, August). Magnetic Hamiltonian Monte Carlo. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70* (pp. 3453-3461).
    - [Bet17] Betancourt, M. (2017). A conceptual introduction to Hamiltonian Monte Carlo. *arXiv preprint arXiv:1701.02434*.
    - [CB17] Cheng, X., & Bartlett, P. (2017). Convergence of Langevin MCMC in KL-divergence. *arXiv preprint arXiv:1705.09048*.

References

- Bayesian Inference: MCMC
  - Dynamics-Based MCMC: full-batch (manifold support)
    - [GC11] Girolami, M., & Calderhead, B. (2011). Riemann manifold Langevin and Hamiltonian Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *73*(2), 123-214.
    - [BSU12] Brubaker, M., Salzmann, M., & Urtasun, R. (2012, March). A family of MCMC methods on implicitly defined manifolds. In *Artificial intelligence and statistics* (pp. 161-172).
    - [BG13] Byrne, S., & Girolami, M. (2013). Geodesic Monte Carlo on embedded manifolds. *Scandinavian Journal of Statistics*, *40*(4), 825-845.
    - [LSSG15] Lan, S., Stathopoulos, V., Shahbaba, B., & Girolami, M. (2015). Markov chain Monte Carlo from Lagrangian dynamics. *Journal of Computational and Graphical Statistics*, *24*(2), 357-378.

References

- Bayesian Inference: MCMC
  - Dynamics-Based MCMC: stochastic gradient
    - [WT11] Welling, M., & Teh, Y. W. (2011). Bayesian learning via stochastic gradient Langevin dynamics. In *Proceedings of the International Conference on Machine Learning* (pp. 681-688).
    - [CFG14] Chen, T., Fox, E., & Guestrin, C. (2014). Stochastic gradient Hamiltonian Monte Carlo. In *Proceedings of the International conference on machine learning* (pp. 1683-1691).
    - [DFB+14] Ding, N., Fang, Y., Babbush, R., Chen, C., Skeel, R. D., & Neven, H. (2014). Bayesian sampling using stochastic gradient thermostats. In *Advances in neural information processing systems* (pp. 3203-3211).
    - [Bet15] Betancourt, M. (2015). The fundamental incompatibility of scalable Hamiltonian Monte Carlo and naive data subsampling. In *International Conference on Machine Learning* (pp. 533-540).

References

- Bayesian Inference: MCMC
  - Dynamics-Based MCMC: stochastic gradient
    - [TTV16] Teh, Y. W., Thiery, A. H., & Vollmer, S. J. (2016). Consistency and fluctuations for stochastic gradient Langevin dynamics. *The Journal of Machine Learning Research*, *17*(1), 193-225.
    - [LPH+16] Lu, X., Perrone, V., Hasenclever, L., Teh, Y. W., & Vollmer, S. J. (2016). Relativistic Monte Carlo. *arXiv preprint arXiv:1609.04388*.
    - [ZCG+17] Zhang, Y., Chen, C., Gan, Z., Henao, R., & Carin, L. (2017, August). Stochastic gradient monomial Gamma sampler. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70* (pp. 3996-4005).

References

- Bayesian Inference: MCMC
  - Dynamics-Based MCMC: stochastic gradient (manifold support)
    - [PT13] Patterson, S., & Teh, Y.W. (2013). Stochastic gradient Riemannian Langevin dynamics on the probability simplex. In *Advances in neural information processing systems* (pp. 3102-3110).
    - [MCF15] Ma, Y. A., Chen, T., & Fox, E. (2015). A complete recipe for stochastic gradient MCMC. In *Advances in Neural Information Processing Systems* (pp. 2917-2925).
    - [LZS16] Liu, C., Zhu, J., & Song, Y. (2016). Stochastic Gradient Geodesic MCMC Methods. In *Advances in Neural Information Processing Systems* (pp. 3009-3017).

References

- Bayesian Inference: MCMC
  - Dynamics-Based MCMC: general theory
    - [MCF15] Ma, Y. A., Chen, T., & Fox, E. (2015). A complete recipe for stochastic gradient MCMC. In *Advances in Neural Information Processing Systems* (pp. 2917-2925).
    - [CDC15] Chen, C., Ding, N., & Carin, L. (2015). On the convergence of stochastic gradient MCMC algorithms with high-order integrators. In *Advances in Neural Information Processing Systems* (pp. 2278-2286).
    - [LZZ19] Liu, C., Zhuo, J., & Zhu, J. (2019). Understanding MCMC Dynamics as Flows on the Wasserstein Space. In *Proceedings of the 36th International Conference on Machine Learning* (pp. 4093-4103).
  - Dynamics-Based MCMC: amortized method
    - [LTL17] Li, Y., Turner, R.E., & Liu, Q. (2017). Approximate inference with amortised MCMC. *arXiv preprint arXiv:1702.08343*.

References

- Bayesian Models
  - Bayesian Networks: Sigmoid Belief Networks
    - [Nea92] Neal, R. M. (1992). Connectionist learning of belief networks. *Artificial intelligence*, 56(1), 71-113.
    - [SJJ96] Saul, L. K., Jaakkola, T., & Jordan, M. I. (1996). Mean field theory for sigmoid belief networks. *Journal of artificial intelligence research*, 4, 61-76.
    - [HDFN95] Hinton, G. E., Dayan, P., Frey, B. J., & Neal, R. M. (1995). The “wake-sleep” algorithm for unsupervised neural networks. *Science*, 268(5214), 1158-1161.
    - [MG14] Mnih, A., & Gregor, K. (2014). Neural variational inference and learning in belief networks. In *Proceedings of the International Conference on Machine Learning*.

References

- Bayesian Models
  - Bayesian Networks: Topic Models
    - [BNJ03] Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3(Jan), pp.993-1022.
    - [GS04] Griffiths, T.L., & Steyvers, M. (2004). Finding scientific topics. *Proceedings of the National academy of Sciences*, 101 (suppl 1), pp.5228-5235.
    - [SG07] Steyvers, M., & Griffiths, T. (2007). Probabilistic topic models. *Handbook of latent semantic analysis*, *427*(7), 424-440.
    - [MB08] Mcauliffe, J.D., & Blei, D.M. (2008). Supervised topic models. In *Advances in neural information processing systems* (pp. 121-128).
    - [PT13] Patterson, S., & Teh, Y.W. (2013). Stochastic gradient Riemannian Langevin dynamics on the probability simplex. In *Advances in neural information processing systems* (pp. 3102-3110).

References

- Bayesian Models
  - Bayesian Networks: Topic Models
    - [ZAX12] Zhu, J., Ahmed, A., & Xing, E. P. (2012). MedLDA: maximum margin supervised topic models. *Journal of Machine Learning Research*, *13*(Aug), 2237-2278.
    - [ZCX14] Zhu, J., Chen, N., & Xing, E. P. (2014). Bayesian inference with posterior regularization and applications to infinite latent SVMs. *The Journal of Machine Learning Research*, *15*(1), 1799-1847.
    - [LARS14] Li, A. Q., Ahmed, A., Ravi, S., & Smola, A.J. (2014). Reducing the sampling complexity of topic models. In *Proceedings of the ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 891-900).
    - [YGH+15] Yuan, J., Gao, F., Ho, Q., Dai, W., Wei, J., Zheng, X., Xing, E.P., Liu, T. Y., & Ma, W. Y. (2015). LightLDA: Big topic models on modest computer clusters. In *Proceedings of the 24th International Conference on World Wide Web* (pp. 1351-1361).
    - [CLZC16] Chen, J., Li, K., Zhu, J., & Chen, W. (2016). WarpLDA: a cache efficient o(1) algorithm for latent Dirichlet allocation. *Proceedings of the VLDB Endowment*, 9(10), pp.744-755.

References

- Bayesian Models
  - Bayesian Networks: Variational Auto-Encoders
    - [KW14] Kingma, D.P., & Welling, M. (2014). Auto-encoding variational Bayes. In *Proceedings of the International Conference on Learning Representations*.
    - [GDG+15] Gregor, K., Danihelka, I., Graves, A., Rezende, D. J., & Wierstra, D. (2015). DRAW: A recurrent neural network for image generation. In *Proceedings of the 32nd International Conference on Machine Learning*.
    - [BGS15] Burda, Y., Grosse, R., & Salakhutdinov, R. (2015). Importance weighted autoencoders. *arXiv preprint arXiv:1509.00519*.
    - [DFD+18] Davidson, T.R., Falorsi, L., De Cao, N., Kipf, T., & Tomczak, J.M. (2018). Hyperspherical variational auto-encoders. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*.
    - [MSJ+15] Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I., & Frey, B. (2016). Adversarial Autoencoders. In *Proceedings of the International Conference on Learning Representations*.

References

- Bayesian Models
  - Bayesian Networks: Variational Auto-Encoders
    - [MNG17] Mescheder, L., Nowozin, S., & Geiger, A. (2017). Adversarial variational Bayes: Unifying variational autoencoders and generative adversarial networks. In *Proceedings of the International Conference on Machine Learning* (pp. 2391-2400).
    - [TBGS17] Tolstikhin, I., Bousquet, O., Gelly, S., & Schoelkopf, B. (2017). Wasserstein Auto-Encoders. *arXiv preprint arXiv:1711.01558*.
    - [PWH+17] Pu, Y., Wang, W., Henao, R., Chen, L., Gan, Z., Li, C., & Carin, L. (2017). Adversarial symmetric variational autoencoder. In *Advances in Neural Information Processing Systems* (pp. 4330-4339).
    - [KSDV18] Kocaoglu, Murat, Snyder, C., Dimakis, A. G., & Vishwanath, S. (2018). CausalGAN: Learning causal implicit generative models with adversarial training. In *Proceedings of the International Conference on Learning Representations*.
    - [LWZZ18] Li, C., Welling, M., Zhu, J., & Zhang, B. (2018). Graphical generative adversarial networks. In *Advances in Neural Information Processing Systems* (pp. 6069-6080).

References

- Bayesian Models
  - Bayesian Neural Networks
    - [Nea95] Neal, R. M. (1995). *Bayesian Learning for Neural Networks* (Doctoral dissertation, University of Toronto).
    - [BCKW15] Blundell, C., Cornebise, J., Kavukcuoglu, K., & Wierstra, D. (2015, June). Weight Uncertainty in Neural Network. In *International Conference on Machine Learning* (pp. 1613-1622).
    - [HA15] Hernández-Lobato, J. M., & Adams, R. (2015). Probabilistic backpropagation for scalable learning of Bayesian neural networks. In *International Conference on Machine Learning* (pp. 1861-1869).
    - [GG16] Gal, Y., & Ghahramani, Z. (2016). Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of the International Conference on Machine Learning* (pp. 1050-1059).
    - [LG17] Li, Y., & Gal, Y. (2017). Dropout inference in Bayesian neural networks with alpha-divergences. In *Proceedings of the International Conference on Machine Learning* (pp. 2052-2061).

References

- Bayesian Models
  - Markov Random Fields
    - [Isi25] Ising, E. (1925). Beitrag zur theorie des ferromagnetismus. *Zeitschrift für Physik A Hadrons and Nuclei*, 31(1), 253-258.
    - [Hop82] Hopfield, J. J. (1982, April). Neural networks and physical systems with emergent collective computational abilities. *Proc. Natl. Acad. Sci. USA*, vol. 79 no. 8, pp. 2554-2558.
    - [HS83] Hinton, G., & Sejnowski, T. (1983). Optimal perceptual inference. In *IEEE Conference on Computer Vision and Pattern Recognition*.
    - [Smo86] Smolensky, P. (1986). Information processing in dynamical systems: Foundations of harmony theory. In *Parallel Distributed Processing*, volume 1, chapter 6, pages 194-281. MIT Press.
    - [Hin02] Hinton, G. E. (2002). Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8), 1771-1800.
    - [HOT06] Hinton, G. E., Osindero, S., & Teh, Y. W. (2006). A fast learning algorithm for deep belief nets. *Neural computation*, 18(7), 1527-1554.

References

- Bayesian Models
  - Markov Random Fields
    - [LCH+06] LeCun, Y., Chopra, S., Hadsell, R., Ranzato, M., & Huang, F. (2006). A tutorial on energy-based learning. *Predicting structured data*, *1*(0).
    - [SH09] Salakhutdinov, R., & Hinton, G. (2009, April). Deep Boltzmann machines. *In AISTATS* (pp. 448-455).
    - [Sal15] Salakhutdinov, R. (2015). Learning deep generative models. *Annual Review of Statistics and Its Application*, 2, 361-385.
    - [KB16] Kim, T., & Bengio, Y. (2016). Deep directed generative models with energy-based probability estimation. *arXiv preprint arXiv:1606.03439*.
    - [ZML16] Zhao, J., Mathieu, M., & LeCun, Y. (2016). Energy-based generative adversarial network. *arXiv preprint arXiv:1609.03126*.
    - [DM19] Du, Y., & Mordatch, I. (2019). Implicit generation and generalization in energy-based models. *arXiv preprint arXiv:1903.08689*.

References

- Topics
  - Online Inference: variational inference
    - [HBB10] Hoffman, M., Bach, F. R., & Blei, D. M. (2010). Online learning for latent Dirichlet allocation. In *Advances in Neural Information Processing Systems* (pp. 856-864).
    - [BBW+13] Broderick, T., Boyd, N., Wibisono, A., Wilson, A. C., & Jordan, M. I. (2013). Streaming variational Bayes. In *Advances in Neural Information Processing Systems* (pp. 1727-1735).
    - [SZ14] Shi, T., & Zhu, J. (2014, January). Online Bayesian Passive-Aggressive Learning. In *International Conference on Machine Learning* (pp. 378-386).
  - Online Inference: MCMC
    - [Cho02] Chopin, N. (2002). A sequential particle filter method for static models. *Biometrika*, *89*(3), 539-552.
    - [ADH10] Andrieu, C., Doucet, A., & Holenstein, R. (2010). Particle Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *72*(3), 269-342.

References

- Topics
  - Bayesian Reinforcement Learning
    - [LRLP17] Liu, Y., Ramachandran, P., Liu, Q., & Peng, J. (2017). Stein variational policy gradient. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*.
    - [HTAL17] Haarnoja, T., Tang, H., Abbeel, P., & Levine, S. (2017, August). Reinforcement learning with deep energy-based policies. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70* (pp. 1352-1361).
    - [ZCLC18] Zhang, R., Chen, C., Li, C., & Carin, L. (2018, July). Policy Optimization as Wasserstein Gradient Flows. In *International Conference on Machine Learning* (pp. 5741-5750).

References

- Others
  - Bayesian Models
    - [KYD+18] Kim, T., Yoon, J., Dia, O., Kim, S., Bengio, Y., & Ahn, S. (2018). Bayesian model-agnostic meta-learning. In *Advances in Neural Information Processing Systems* (pp. 7332-7342).
    - [LST15] Lake, B.M., Salakhutdinov, R., & Tenenbaum, J.B. (2015). Human-level concept learning through probabilistic program induction. *Science*, 350(6266), pp. 1332-1338.
  - General Generative Models
    - [TOB16] Theis, L., van den Oord, A., & Bethge, M. (2016). A note on the evaluation of generative models. In *International Conference on Learning Representations (ICLR 2016)* (pp. 1-10).
    - [Hus15] Huszár, F. (2015). How (not) to train your generative model: Scheduled sampling, likelihood, adversary?. *arXiv preprint arXiv:1511.05101.*

References

- Others
  - Generative Adversarial Networks
    - [GPM+14] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial nets. In *Advances in neural information processing systems* (pp. 2672-2680).
    - [MO14] Mirza, M., & Osindero, S. (2014). Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*.
    - [OOS17] Odena, A., Olah, C. & Shlens, J. (2017). Conditional Image Synthesis with Auxiliary Classifier GANs. *Proceedings of the 34th International Conference on Machine Learning, in PMLR 70:2642-2651.*
    - [NLVP17] Nguyen, T., Le, T., Vu, H., & Phung, D. (2017). Dual discriminator generative adversarial nets. In *Advances in Neural Information Processing Systems* (pp. 2670-2680).

References

- Others
  - Related References
    - [Ama85] Amari, S. I. (1985). *Differential-geometrical methods in statistics*. Springer, New York.
    - [Wil92] Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, *8*(3-4), 229-256.
    - [HV93] Hinton, G., & Van Camp, D. (1993). Keeping neural networks simple by minimizing the description length of the weights. In *in Proc. of the 6th Ann. ACM Conf. on Computational Learning Theory*.
    - [NJ01] Ng, A. Y., & Jordan, M. I. (2002). On discriminative vs. generative classifiers: A comparison of logistic regression and naive Bayes. In *Advances in neural information processing systems* (pp. 841-848).
    - [Set06] Sethna, J. (2006). *Statistical mechanics: entropy, order parameters, and complexity* (Vol. 14). Oxford University Press.

References

- Others
  - Related References
    - [SHK+14] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, *15*(1), 1929-1958.
    - [HCD+16] Houthooft, R., Chen, X., Duan, Y., Schulman, J., De Turck, F., & Abbeel, P. (2016). VIME: Variational information maximizing exploration. In *Advances in Neural Information Processing Systems* (pp. 1109-1117).
    - [HLR+16] Hernandez-Lobato, J. M., Li, Y., Rowland, M., Bui, T., Hernandez-Lobato, D. & Turner, R. (2016). Black-Box Alpha Divergence Minimization. *Proceedings of The 33rd International Conference on Machine Learning*, in PMLR 48:1511-1520.
    - [SCZ+17] Shi, J., Chen, J., Zhu, J., Sun, S., Luo, Y., Gu, Y., & Zhou, Y. (2017). ZhuSuan: A library for Bayesian deep learning. *arXiv preprint arXiv:1709.05870*.

The End